MODELING AND CONTROL OF WHEELED MOBILE ROBOTS

Ph.D. Dissertation

CMU-RI-TR-88-20

Patrick Fred Muir

Department of Electrical and Computer Engineering
And
The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213

August, 1988

Advisor:
Professor Charles P. Neuman

Committee:
Dr. Hans P. Moravec
Professor Bruce H. Krogh
Professor Matthew T. Mason

Table of Contents

Page

List of Figures

List of Tables

Acknowledgments

I acknowledge my graduate advisor Professor Charles P. Neuman and the Head of the Mobile Robot Laboratory Dr. Hans Moravec for providing technical feedback and encouragement during the last four years. From a computer science standpoint, Hans has encouraged creative thinking and has been a storehouse of broad scientific knowledge. From an engineering standpoint, Professor Neuman has stressed the fundamentals and insisted on impeccable manuscripts. I hope that these two complementary philosophies are reflected in my own work through the conceptualization of useful ideas founded in fundamental principles and their dissemination through timely publication. I also thank Dr. Ralph Hollis of the IBM Thomas J. Watson Research Center for intensifying my interest in robotics by sharing his enthusiasm in the area.

I am grateful to several CMU colleagues who have colored my college experience with differing points of view the last ten years: David Basile, Michael Blackwell, Kevin Dowling, Alberto Elfes, Dai Feng, Timothy Greattinger, Dr. Pradeep Khosla, Andrew Lee, Larry Mathies, Dr. John Murray, Michael Pfeifer, Dong Hun Shin, Stephen Thomas, and Gregory Toto.

Abstract[1]

The accurate model-based servo-control of *wheeled mobile robots* (WMRs) relies upon the formulation of realistic kinematic and dynamic models. We[2] identify *six* special WMR characteristics (closed-chains, higher-pair joints, unactuated and unsensed joints, friction, and pulse-width modulation) that require methodologies for modeling and control beyond those conventionally applied to stationary manipulators. Then, we develop methodologies for the kinematic and dynamic modeling of robotic mechanisms incorporating these special characteristics. We introduce *instantaneously coincident coordinate systems* and the *wheel Jacobian* to resolve WMR kinematic modeling. We introduce the concepts of *force/torque propagation* and *frictional coupling* at a joint to formulate a powerful unifying dynamic modeling framework. We compute the inverse and forward kinematic and dynamic solutions for model-based WMR servo-control and simulation.

We demonstrate the applicability of (kinematics-based) *resolved motion rate* and (dynamics-based) *resolved acceleration* servo-control methodologies to WMRs through computer simulation evaluation studies. We exemplify our modeling and servo-control methodologies through *Uranus*, a three degree-of-freedom (DOF) WMR, and *Bicsun-Bicas*, a two DOF WMR. Our results show that resolved motion rate servo-control is adequate for general-purpose applications of Uranus. In contrast, the mechanically simpler Bicsun-Bicas requires the computationaly complex resolved acceleration servo-control to compensate for the significant coupling and nonlinear components in its dynamic model. We recommend Bicsun-Bicas with resolved acceleration servo-control for general-purpose indoor applications because it is mechanically simple, capable of tracking any spatial x-y path, and if a turret is added, provides onboard manipulators, sensors, or docking instruments with three DOFs.

---

1 This research has been supported by the office of Naval Research under Contract N00014-81-K-0503.

2 For expository convenience, the author has chosen to use such personal pronouns as "we" and "our".

Chapter 1
Introduction

1.1 Overview and Motivation

Over the past twenty years, as robotics has become a scientific discipline, research and development have concentrated on *stationary* robotic manipulators [Brady82, Luh83], primarily because of their industrial applications. Less effort has been directed toward mobile robots. Although legged [Raibert83] and treaded [Iwamoto83] locomotion have been studied, the overwhelming majority of the mobile robots which have been built and evaluated utilize wheels for locomotion. *Wheeled mobile robots* (WMRs) are more energy efficient than legged or treaded robots on hard, smooth surfaces [Bekker60, Bekker69]; and will potentially be the first mobile robots to find widespread application in industry, because of the hard, smooth plant floors in existing industrial environments. WMRs require fewer and simpler parts and are thus easier to build than legged or treaded mobile robots. Wheel control is less complex than the actuation of multi-joint legs, and wheels cause minimal surface damage in comparison with treads [Holland83].

The WMR literature documents investigations which have concentrated on the application of mobile platforms to perform intelligent tasks [Nilsson84, Giralt79, Kanayama81], rather than on the development of methodologies for modeling, designing, and controlling the mobility subsystem. Improved mechanical designs and mobility servo-controllers will enable the application of WMRs to an expanding number of practical tasks. In this dissertation, we build the foundations for the design of high performance servo-controllers for WMRs by developing systematic kinematic and dynamic modeling methodologies.

Our development parallels the stationary manipulator modeling and control research [Paul81a]. We have identified four goals for our WMR research: kinematic modeling, dynamic modeling, model-based servo-controller design, and computer simulation for model implementation and servo-controller evaluation. For each of these WMR research areas, we: survey prior research, identify the research issues, conceive and develop methodologies as required, and illustrate these methodologies by enumerating systematic procedures and providing prototypical examples. Although some concepts and techniques from stationary manipulator kinematic and dynamic modeling apply to WMRs, several problems arise due to the special characteristics of WMRs (discussed in Section 1.5). We study and solve each of these problems in our WMR model development. In contrast to our modeling activities, we show that the stationary manipulator model-based servo-control designs and the computer simulation techniques documented in the literature are directly applicable to WMRs once the kinematic and dynamic models have been formulated. As a result, the majority of our effort, and the bulk of this dissertation, is directed towards kinematic and dynamic modeling activities.

Until recently, the majority of robotics control research efforts were directed toward the design of feedback algorithms capable of controlling the highly complex, coupled and nonlinear dynamics of robots. The advent of model-based Cartesian-space servo-control algorithms for robots (e.g., resolved motion rate control [Whitney69] and resolved acceleration control [Luh80a]) represent a milestone in robot control and has changed the direction of robotics control research. Since the viability of model-based Cartesian-space robot control has been demonstrated, research directed toward the formulation of accurate robot models has been required to support the controller design. The amount of effort dedicated to WMR kinematic and dynamic modeling in this dissertation is evidence of that change.

In the remainder of this chapter we introduce: past, present and future WMRs in Section 1.2, WMR control systems in Section 1.3, kinematic and dynamic WMR models in Section 1.4, the special characteristics of WMRs which distinguish them from stationary manipulators in Section 1.5 and the notation and terminology for WMR analysis in Section 1.6. In Section 1.7, we expound

upon our dissertation goals and summarize our contribtions. In Section 1.8, we outline the remainder of the dissertation.

1.2 Wheeled Mobile Robots

WMRs are inherently different from stationary robotic manipulators and legged or treaded mobile robots. Before delving into the intricacies of our study, we must define a wheeled mobile robot. The *Robot Institute of America* defines a robot as "*A programmable, multifunction manipulator designed to move material, parts, tools, or specialized devices through variable programmed motions for the performance of a variety of tasks*" [Holland83]. We are interested in modeling and controlling a robot which is capable of locomotion upon wheels. We thus introduce an operational definition of a WMR to specify the range of robots to which our modeling and servo-control studies apply.

> *Wheeled Mobile Robot* - A robot capable of locomotion on a surface solely through the actuation of wheel assemblies mounted on the robot and in contact with the surface. A wheel assembly is a device which provides or allows relative motion between its mount and a surface on which it is intended to have a *single* point of *rolling* contact.

Each wheel (conventional, omnidirectional or ball wheel) and all links between the robot body and the wheel constitute a wheel assembly.

In Appendix 1, we develop a nomenclature and symbolic representation for describing WMRs. In Appendix 2, we apply our nomenclature and symbolic representation rules to survey past and present WMRs[1] to view the spectrum of wheel configurations which are likely to be confronted in our work. We are interested in the kinematic structure of each WMR (i.e., the placement of wheels, steering links, actuators and sensors) because the kinematic structure is the essential information required to formulate *symbolically* the kinematic and dynamic models and servo-control algorithms. Two WMRs with the same kinematic structure have identical symbolic kinematic and dynamic models and

---

[1] With the exception of the omnidirectional treaded vehicle, the hybrid spider drive (when walking), the hybrid locomotion vehicle (when climbing) and the triangle wheel step climber (when climbing steps), the mobile robots reviewed in Appendix 2 satisfy our definition of a WMR.

model-based servo-control algorithms. Once the numerical values of the kinematic, dynamic, and frictional parameters are known, the models and servo-control algorithms for each specific WMR can then be customized and systematically organized for computational efficiency [Murray86].

From our WMR survey in Appendix 2, we specify WMR modeling requirements. A WMR model must allow any number and combination of conventional, omnidirectional or ball wheels. The wheels can be mounted at any position and orientation with respect to the main body provided that each touches the surface-of-travel. Even though each wheel can be mounted at the end of an articulated linkage, we will deal with zero or one steering link per wheel. Finally, there may be coupling between wheels (e.g., two wheels may steer together as on the Stanford Cart). With these observations, we develop in Chapter 2 a methodology for WMR kinematic modeling. In Chapter 3, we detail a WMR dynamic modeling framework.

Our research has been conducted within the Mobile Robot Laboratory in the Robotics Institute of Carnegie Mellon University. The Mobile Robot Laboratory has been studying WMRs for six years through the design, construction and evaluation of a sequence of three WMR prototypes. The first WMR, *Pluto* [Moravec83, Moravec86, Muir84a] (pictured in Figure A2.5), having three independently steered and driven wheels, illuminated control problems for resolution. During the servo-controller design and implementation process, the difficulties of controlling such a complex WMR became evident and the need for WMR modeling and servo-control methodologies was established. The second prototype, *Neptune* [Podnar84] (pictured in Figure A2.3), which is driven and steered like a tricycle, is applied for vision and sonar ranging research studies. The most recent WMR, *Uranus* [Moravec86] (pictured in Figure A2.9), is currently under construction. Uranus possesses four omnidirectional wheels [Ilon75, Adams84] mounted symmetrically at the four corners of a rectangular wheel base providing three degree-of-freedom (DOF) locomotion. An onboard microprocessor will be interfaced to the wheel actuators (brushless DC motors on each wheel), the wheel sensors (optical shaft encoders on each wheel), and an off-board mainframe computer for the communication of high-level commands. The construction of Uranus has not progressed sufficiently to rely upon its operation for experimental studies in this dissertation. Uranus and a

two DOF WMR of our design (*Bicsun-Bicas* studied in Chapter 7) are utilized in computer simulation as prototypical WMRs for displaying our modeling and control methodologies.

Current and futuristic applications of WMRs motivate our research. WMRs are presently applied to accomplish tasks requiring motions over distances much larger than the workspace of stationary manipulators [Schachter83]. Automatic guided vehicles (AGVs) [Derry82] transport parts and materials within factories by following paths marked by reflective tape, paint, or buried wire. As teleoperated machines, they are applied to the remote manipulation of hazardous substances [Fischetti85] such as radioactive materials in nuclear power plants and explosives for modern bomb squads. WMRs are suited for repetitive tasks such as security monitoring [Wilson85] for prisons and warehouses. Smaller, cheaper WMRs are marketed as personal robots [Bell85, Marrs85] for hobbyists and educators. WMRs are applied as testbeds for research [Moravec83, Nilsson84] in control, artificial intelligence, sensor interpretation, and computer vision. WMRs also find application in space and undersea exploration, agricultural machinery, military, and mobility for the disabled.

As the design, construction, control and maintenance of WMRs become better understood and production costs decrease, WMRs will be applied to a growing list of applications. The first natural application will be mobile manipulation to provide the capabilities of a stationary manipulator over an increased workspace. WMRs are ideally suited for such agricultural applications as plowing, sowing and harvesting because these tasks require repetitive motion over large distances along simple paths. Warehouses are also well-structured environments in which WMRs will find application for autonomous storage, sorting and retrieval of products and materials. Private homes are much less structured; however, because of the marketing potential, we envision significant efforts towards developing WMRs as household maids. Autonomous robots, including WMRs, which are the viable alternative to risky manned space flight, will eliminate the time lag problems inherent with interplanetary communication for teloperated mechanisms. Manned military vehicles may someday also be replaced by autonomous WMRs.

1.3 WMR Control

WMR control can be structured in one of two frameworks: a *horizontal* hierarchy or a *vertical* hierarchy of functional modules. The horizontal hierarchy [Coles75, Giralt79, Moravec80, Kanayama81, Elfes83, Nilsson84, Crowley85, Krogh86] has been the traditional framework of mobile robot research. A *horizontal* hierarchy consists of a *sequential* series of modules through which information is passed. Information from the sensors is processed by one module and passed sequentially to succeeding modules. The final module computes commands which are communicated to the WMR actuators. The horizontal hierarchy has been criticized for its brittle nature; i.e., if one module fails to accomplish its task, the entire system fails. The recently introduced *vertical* hierarchy [Brooks86] consists of *parallel* modules, all of which process simultaneously the sensory information to produce the actuator commands. Higher levels in the hierarchy can override the actuator commands of the lower levels. Simple reflex WMR motions are thus realized by the action of computationally-simple, low level modules and long term goals are pursued by the action of computationally-complex, high level modules. The vertical hierarchy can recover from the failure of a module at one level by overriding its operation at a different level. Since both approaches to WMR control consist of an organized connection of functional modules, the modules may be developed independently of the hierarchical implementation.

Individual WMR control research activities typically focus on a particular functional module. Sensor interpretation, including computer vision, sonar sensing and laser ranging, is a popular area of research. Edge detection, stereo matching, and feature detection are a sample of the problems tackled by computer vision researchers [Thorpe83, Moravec84, Thorpe84, Thorpe85, Wallace85, Matthies86]. The conventional task in sonar sensing [Moravec87, Elfes86] is interpreting the sonar data to map the outline of the robot's environment. World modeling [Nilsson84, Crowley85] is the process of applying interpreted sensory data to construct a computer model of the robot's environment. The world model may be symbolic [Nilsson84], geometric [Crowley85], or a grid of occupancy probabilities [Moravec88b], depending upon its subsequent applications. A trajectory planning/obstacle avoidance module [Krogh84, Crowley85, Moravec80, Wallace84] applies the world model

to specify the desired robot position trajectory. To move the robot along a specified trajectory requires a Cartesian-space servo-controller module [Daniel84, Muir85]. This dissertation addresses the design and evaluation of the WMR servo-controller.

We define *servo-controller* as the algorithmic process which computes in *real-time* the command values for the robot actuators to cause the robot to produce desired motions. To fulfill the *real-time* computational requirement, the servo-controller must execute much faster than the fastest robot motions to be controlled.

Although the servo-control of stationary manipulators is widely documented in the literature [Whitney69, Dubowsky79, Luh80a, Luh80b, Paul81a, Paul81b, Brady82, Luh83, Tourassis85, Khosla87, Neuman87a], few WMR servo-control studies have been undertaken. The WMR *control* designs found in the literature are primarily trajectory planning methods of determining the desired WMR motion [Coles75, Giralt79, Moravec80, Kanayama81, Elfes83, Nilsson84, Crowley85, Brooks86, Krogh86]. The sparse literature on WMR *servo-control* is aimed at specific WMR designs [Daniel84, Hongo85, Muir87a], or based upon overly simplified models of the robot motion [Schmidt71, Wallace85]. We formulate WMR modeling and servo-control designs incorporating the complex kinematics and dynamics of WMRs.

Cartesian-space servo-control is the established methodology of stationary manipulator model-based servo-control. Our contribution in the area of WMR servo-control is the recognition and demonstration that Cartesian-space servo-control methodologies are directly applicable to WMRs. Resolved motion rate servo-control [Whitney69] is a Cartesian-space method which utilizes the kinematic robot model. Resolved acceleration servo-control [Luh80a] is a Cartesian-space methodology which utilizes the complete dynamic robot model. Both methods apply the inverse (kinematic or dynamic) model of the robot in the feedforward control path to cancel the robot's inherent motion characteristics. Feedback control signals are then added to servo errors due to unmodeled dynamics or disturbances to zero. Since the inverse (kinematic or dynamic) model of the robot is the keystone of Cartesian-space model-based servo-control, it is paramount to develop accurate kinematic and dynamic robot models.

1.4 WMR Modeling

The majority of our research efforts in this dissertation are directed towards the kinematic and dynamic modeling of WMRs [Neuman87b]. *Kinematics* is the study of the geometry of motion. In the context of WMRs, we are interested in determining the motion of the robot from the geometry of the constraints imposed by the motion of the wheels. Our kinematic analysis is based upon the assignment of coordinate axes within the WMR and its environment, and the application of (4×4) matrices to express transformations between coordinate systems. The parameters required for kinematic modeling are the distances and angles between coordinate systems. The variables of the kinematic model are the accelerations and velocities of the WMR body, steering links and wheels, and the positions of the steering links.

The key feature of our kinematic methodology is the introduction of *instantaneously coincident coordinate systems.* The application of instantaneously coincident coordinate systems allows the formulation of the position, velocity and acceleration equations-of-motion directly from homogeneous transformation matrices. We develop a matrix coordinate transformation algebra for this purpose. We organize the velocity equations-of-motion of each WMR wheel to introduce wheel Jacobian matrices which we then apply to characterize WMR mobility, actuation and sensing, design resolved motion rate servo-controllers, compute the WMR position by dead reckoning and detect wheel slippage. Instantaneously coincident coordinate systems are essential components of our dynamic modeling methodology.

A *dynamic* WMR model describes the relationships between the WMR motion and the actuator forces/torques. Our dynamic modeling methodology is based upon (6×6) matrix-vector computations. Each force/torque source is modeled within its natural coordinate system. The WMR dynamic model is computed by propagating the effects of all of the force/torque sources to a common coordinate system. Dry friction is modeled at each joint to couple forces, torques between WMR components. The parameters of our dynamic model are the distances and angles between coordinate systems, component masses and inertias, and dry and viscous frictional coefficients of the joints. The

variables of the dynamic model are the velocities and accelerations of the WMR body, wheels and steering links, and the positions of the steering links and the actuator forces/torques.

Our dynamic modeling methodology is the most interesting, and the most widely applicable contribution of this dissertation. The methodology unifies the modeling of robotic mechanisms, including: stationary manipulators, multi-manipulators, WMRs, legged mobile robots and robotics hands, incorporating the effects of inertial, viscous frictional, gravitational, actuation and environmental forces/torques; and stiction, Coulomb and rolling friction. Our dynamic methodology is extendible in that it may be applied naturally to model such phenomena such as link or joint flexibility. Our dynamic methodology is modular: each force/torque and dry frictional phenomenon is modeled independently; the force/torque equations-of-motion and the kinematic transformations are built systematically from the force/torque and frictional modules independently; and finally, the dynamic equations-of-motions are assembled systematically by combining the force/torque and kinematic components and are solved by classical methods. Our dynamic methodology is based physically upon the propagation of forces/torques from their sources within the robotic mechanism through the intervening links and joints to the main body. The matrix-vector format of our dynamic methodology allows insightful graphical methods of representation and parallel computational structures for fast execution. The dynamic WMR model is required to apply the resolved acceleration servo-control methodology.

1.5 WMR Special Characteristics

Even though the methodologies for modeling and controlling stationary manipulators are applicable to WMRs, there are inherent differences which cannot be addressed with these methodologies. We have identified the following *six* special characteristics of WMRs:

(1) WMRs contain multiple *closed-chains* [Orin81]; whereas stationary manipulators form closed-chains only when in contact with stationary objects.

(2) The contact between a wheel and a planar surface is a *higher-pair*; whereas stationary manipulators contain only lower-pair joints [Angeles82, Shigley69, Shigley80].

(3) Only some of the degrees-of-freedom (DOFs) of a wheel on a WMR are actuated; whereas all of the DOFs of each joint of a stationary manipulator are actuated.

(4) Only some of the DOFs of a wheel on a WMR have position or velocity sensors; whereas all of the DOFs of each joint of a stationary manipulator have both position and velocity sensors.

(5) Friction within a WMR is significant at the joints including the wheel point-of-contact with the floor; whereas friction is oftentimes neglected or incorporated by overly simplified models for stationary manipulators.

(6) Pulse-width modulation is desired for efficient WMR motor actuation [Muir84b, Muir85] because onboard power supplies are limited; whereas conventional voltage control is applied to stationary manipulators because power economy is not a key issue.

Although these six characteristics are distinct, they are closely interrelated. The first four WMR characteristics receive special consideration in our formulation of the WMR kinematic model [Muir86]. The kinematic WMR model and friction are the vital components of our WMR dynamic modeling methodology. Finally, the kinematic and dynamic models and pulse-width modulation are required for WMR servo-controller design. We discuss in turn each of the aforementioned special WMR characteristics. We draw analogies with modeling and control methodologies for stationary manipulators documented in the literature.

Conventional stationary manipulators are open-chain mechanisms because these robots consist of a serial arrangement (i.e., a *chain*) of links connected by joints which do not form a complete circuit (i.e., the chain is *open*). Each wheel of a WMR may be considered as an open-chain; however, since each wheel contacts a common surface-of-travel (i.e., the floor), the individual chains are closed through the surface-of-travel and the WMR is a *closed-chain*. Open-chains, such as stationary manipulators, allow the serial computation of

kinematic and dynamic models. The coordinate transformation **A** matrices describing the relative position and orientation of two adjacent links of a stationary manipulator may be cascaded *serially* to compute the forward and inverse position solutions [Paul81a]. Similarly, the two-stage *recursive* Newton-Euler dynamic modeling methodology for stationary manipulators allows the *serial* computation of the joint forces/torques from the manipulator motions [Luh80b]. Because WMRs are closed-chains the kinematic and dynamic equations-of-motion must be computed in *parallel*.

A WMR mechanically consists of a main body which is able to move upon the surface-of-travel by means of one or more wheels. Such a *simple* closed-chain mechanical structure is sketched schematically in Figure 1.1. Orin and Oh [Orin81] describe a simple closed-chain mechanism as "*one in which the removal of a particular member of the system breaks all closed-chains*". The particular member which, when removed, breaks all closed-chains is either the main body or the environment.

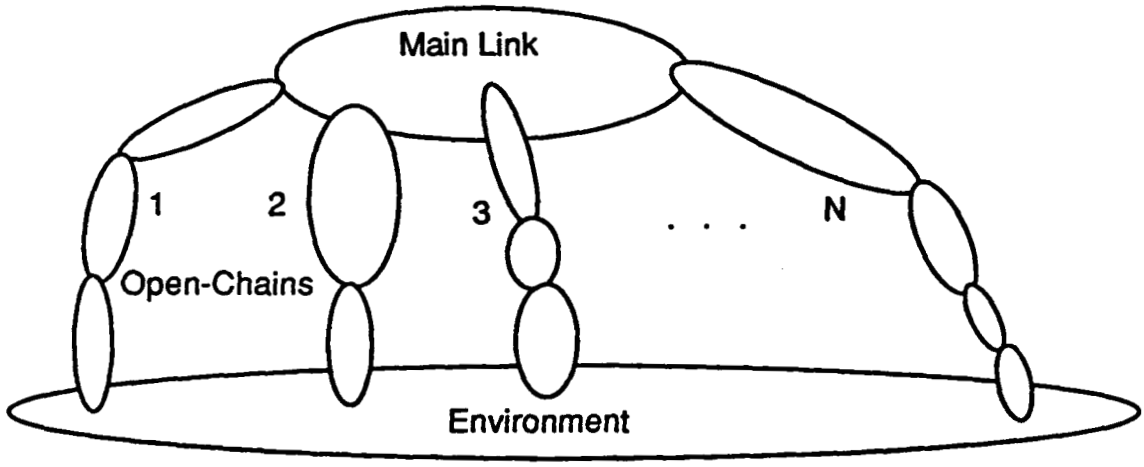


Figure 1.1: A Simple Closed-Chain Robotic Mechanism

The mechanical system in Figure 1.1 consists of a main body in contact with N open-chains of bodies. Each pair of adjoining bodies contact at a joint, and the distal rigid body of each open-chain contacts the environment (i.e., a body external to the system). The mechanical system in Figure 1.1 models WMRs and a multitude of robotic mechanisms. The robotic mechanisms, and the roles played by the main body, the open-chains and the environment are exemplified in Table 1.1.

Table 1.1: Simple Closed-Chain Models of Robotic Mechanisms			
Robotic Mechanism	Main Link	Open-Chains	Environment
m DOF Manipulator	End-Effector	Manipulator Links	Stationary Base
Multi-Manipulator	Workpiece	Manipulators	Stationary Base
WMR	WMR Body	Wheels	Surface-of-travel
Legged Robot	Robot Torso	Legs	Surface-of-travel
Robot Hand	Grasped Object	Fingers	Palm of the Hand

For example:

- An *m DOF robotic manipulator* may be modeled by a single open-chain of m rigid links; the end-effector playing the role of the main link and the stationary base playing the role of the environment;
- A *multi-manipulator system* may be modeled by one open-chain for each manipulator; the workpiece playing the role of the main link and the common stationary base playing the role of the environment;
- A *WMR* may be modeled by one open-chain for each wheel; the WMR body playing the role of the main link and the surface-of-travel playing the role of the environment;
- A *legged robot* may be modeled by one open-chain for each leg; the robot torso playing the role of the main link and the surface-of-travel playing the role of the environment; and
- A *robotic hand* may be modeled by one open-chain for each finger; the grasped object playing the role of the main link and the palm of the hand (which may be the end-effector of a robotic manipulator) playing the role of the environment.

All stationary manipulator joints are either *prismatic* (providing translational motion) or *revolute* (providing rotational motion), and thus are *lower-pairs*. The WMR joint between each wheel and the surface-of-travel is a *higher-pair*. A lower-pair allows a common surface contact between adjacent links; whereas a higher-pair allows point or line contact [Beggs66]. Lower-pairs impose *holonomic* (positional) constraints between two adjacent links; i.e., the position and orientation of one link can be computed from the position and

orientation of the adjacent link, and the position of the joint. Higher-pairs provide non-holonomic (velocity) constraints between two adjacent links; i.e., the velocities of one link can be computed from the velocities of the adjacent link and the velocities of the joint, but these velocity constraints cannot be integrated to compute the position of the link from the positions of the adjacent link and the joint.

To control the motion of an open-chain, all of the joints must be actuated and sensed. In contrast, a closed-chain mechanism may be adequately controlled with some joints *unactuated*, and its motions may be adequately discerned with some joints *unsensed*. Moreover, the WMR higher-pair wheel joints do not allow the actuation and sensing of the rotational θ DOF of each wheel about the point-of-contact with the surface-of-travel. The actuated joints need not correspond to the sensed joints; in fact, it is advantageous to sense unactuated joints which are less susceptible to wheel slippage.

Friction is a non-mechanical property of two substances (i.e., solids, gases or liquids) in contact [Kragelskii65]. Consequently, frictional forces/torques between two substances cannot be modeled exactly by a function of time and the kinematic variables (i.e, the positions, velocities, and accelerations) of the substances. Frictional phenomena originate from the properties of the surface texture, substance composition, temperature, and the kinematic variables. There are, however, accepted approximate models for the frictional forces/torques between two solid bodies (dry friction) and between a solid body and a gas or liquid (viscous friction) which are functions of the kinematic variables and the (normal) forces/torques.

Dry friction plays a significant role in creating the motion of WMRs. WMR mobility stems from the dry translational friction between the wheels and the surface-of-travel. Lack of sufficient friction at the wheel point-of-contact leads to wheel slippage, a problem not encountered in stationary manipulator operation. Dry friction at the wheel bearings also has a significant effect. For example, a WMR traveling in a straight-line trajectory at a constant speed without wheel slip (a typical trajectory segment) only expends energy to overcome bearing, viscous, and rolling frictions. For a WMR with the kinematic structure of *Uranus*, roller bearing frictions in the *omnidirectional* wheels dissipate as much as 80% of the total available energy [Daniel84]. This is in direct contrast to manipulator

bearing friction which is oftentimes negligible in comparison with the inertial, centripetal and Coriolis, and gravitational forces/torques [Paul81a].

Pulse-width modulation (PWM) is a power efficient method for actuating motors. The conventional method of actuating motors is voltage actuation; i.e., the voltage input to the motor is adjusted by the servo-control algorithm. In contrast, pulse-width modulation maintains the magnitude of the voltage to the motor at a constant value. The length of time in each sampling period during which the voltage is turned-on is adjusted by the servo-control algorithm when applying pulse-width modulation. We are not concerned about power efficiency for stationary manipulator control because the power supply (which conventionally is of relatively large weight and volume) is located offboard. We apply pulse-width modulation for WMRs because the power supply which must be carried onboard is limited.

1.6 Notation and Terminology

We utilize the following notation throughout our development. Lower case letters denote scalars (e.g.,m), lower case bold letters denote vectors (e.g.,**v**), upper case letters denote coordinate systems (e.g.,A), upper case italics letters denote bodies (e.g.,*A*), and upper case bold letters denote matrices (e.g.,**R**). We denote *three*-vectors and (3×3) matrices by outlining (e.g., $\mathbf{v}$ and $\mathbf{R}$) to distinguish them from their six-dimensional counterparts. Pre-superscripts denote reference coordinate systems. For example, $^{A}\mathbf{v}$ is the vector **v** in the A coordinate system. The pre-superscript may be omitted if the coordinate system is transparent from the context. Post-subscripts denote coordinate systems, bodies, or components of a vector or matrix, as indicated in each particular case.

We denote precisely each variable within our development with a reference coordinate system and sufficient subscripts for easy identification. As we progress through our analyses, the mathematical expressions representing WMR models become increasing complex, and the precise notation becomes invaluable. Our notation is not as cumbersome as that of Kane [Kane85] who utilizes an additional pre-subscript and thereby denotes the coordinate system in which differentiations of the variables are performed. We forgo this extra

notation by differentiating variables exclusively within the reference coordinate system defined by the pre-superscript.

We manipulate forces along the three Cartesian axes f_x, f_y, and f_z and torques about the axes τ_x, τ_y, and τ_z homogeneously in our dynamics methodology by creating the *force/torque* six-vector:

$$\mathbf{f} = \begin{pmatrix} f_x & f_y & f_z & \tau_x & \tau_y & \tau_z \end{pmatrix}^T.$$

Linear and angular positions, velocities and accelerations are similarly placed in six-vectors:

$$\mathbf{p} = \begin{pmatrix} d_x & d_y & d_z & \theta_x & \theta_y & \theta_z \end{pmatrix}^T, \qquad \mathbf{v} = \begin{pmatrix} v_x & v_y & v_z & \omega_x & \omega_y & \omega_z \end{pmatrix}^T,$$

and

$$\mathbf{a} = \begin{pmatrix} a_x & a_y & a_z & \alpha_x & \alpha_y & \alpha_z \end{pmatrix}^T$$

where the angles θ_x, θ_y, and θ_z are measured about the x, y, and z axes, respectively, according to the roll-pitch-yaw convention [Paul81a].

Coordinate systems play a key role in kinematic and dynamic modeling. Each coordinate system is *fixed to a body* within the system so that the motion of the coordinate system is precisely that of the body to which it is fixed. The coordinate system need not be located within the mass or enclosed volume of the body. In fact, the coordinate system may be located anywhere in space, but its motion is the motion of the body as though there where a rigid massless rod connecting the coordinate system to the body. Positions ${}^A\mathbf{p}_B$, velocities ${}^A\mathbf{v}_B$, and accelerations ${}^A\mathbf{a}_B$ denote the motion of coordinate system B relative to coordinate system A. To specify the position, velocity or acceleration of body *C*, we must specify the position ${}^A\mathbf{p}_{C_1}$, velocity ${}^A\mathbf{v}_{C_1}$, or acceleration ${}^A\mathbf{a}_{C_1}$ of coordinate system C_1, fixed to body *C*, relative to coordinate system A.

The number of *degrees-of-freedom* (DOFs) of a mechanism is determined by the number and type of the joints and the associated kinematic constraints. Specification of the DOFs in the robotics literature is mechanism dependent. For open-chain manipulators, the established convention is to set the number of DOFs equal to the number of joints. The number of DOFs of the *end-effector* of a manipulator is the number of scalar parameters required to

describe the position and orientation of the end-effector. At singularities in the workspace or for redundant manipulators, the number of DOFs of the end-effector is less than the number of DOFs of the manipulator. We define similarly the number of DOFs of a WMR as the number of scalar parameters required to describe the motion (i.e., velocity) of the main link of the WMR. The maximum number of DOFs of a WMR is thus *three* (x-translation, y-translation and θ-rotation), and the number of WMR DOFs at any instant may depend upon the steering angles of the wheels.

1.7 Dissertation Goals and Contributions

Our ambitous goals for this dissertation are fourfold. First, we seek to model the *kinematics* of WMRs. We propose to formulate a methodolgy for computing the system of kinematic equations-of-motion which describe the relative positions, velocities and accelerations of the WMR components. Second, we desire to model the *dynamics* of WMRs by formulating a methodology for computing the system of dynamic equations-of-motion which describe the relationships between the forces/torques acting within the WMR and the positions, velocities, and accelerations of the WMR components. We propose to formulate the WMR kinematic and dynamic modeling methodologies for the broad range of WMRs summarized in Section 1.2 utilizing conventional, steered conventional, omnidirectional and ball wheels and incorporating the special WMR characteristics detailed in Section 1.5. Third, we apply our kinematic and dynamic WMR models to design WMR *servo-controllers*. We believe that Cartesian-space stationary manipulator servo-control methodologies such as the resolved motion rate [Whitney69] and resolved acceleration [Luh80a] algorithms are directly applicable to WMRs. Fourth and finally, we wish to *simulate* WMRs under servo-control through computer implementation of the WMR kinematic and dynamic models to evaluate the performance of the WMR servo-controllers.

In the course of our four-year study, we have attained all four of our dissertation goals. We have formulated a WMR kinematic modeling methodology based upon the computation of wheel Jacobian matrices. Our introduction of instantaneously coincident coordinate systems provides the key for adapting existing stationary manipulator kinematic modeling methodologies

to WMRs. We have shown that our WMR kinematic model is applicable to mobility, actuation and sensing characterization, mechanical design, dead reckoning and wheel slip detection in addition to its primary servo-control and simulation applications.

We have introduced a dynamic modeling framework based upon the novel concepts of force/torque propagation through links and frictional coupling through joints. Our dynamic modeling framework is the first documented which incorporates higher-pair joints, simple closed-chains, unactuated and unsensed joints and dry friction in an integrated, modular, systematic framework. Our dynamic modeling framework applies to stationary manipulators, multi-manipulators, WMRs, legged mobile robots, and robotic hands, and therby has potentially widespread applications for future robot design, control and simulation applications.

We have shown that resolved motion rate and resolved acceleration servo-control methodologies are directly applicable to non-redundant WMRs. These servo-controllers can be implemented with sampling periods which are proportional to the weight of the WMR. A 0.2 second sampling period is thus sufficient for adequately controlling a 90 Kg (200 lb) WMR. Such a large sampling period (as compared to manipulator servo-control sampling periods) is welcome for real-time applications because it provides for implementations on small, inexpensive microprocessors.

We have simulated WMRs under servo-control through the application of MACSYMA. We consider two case studies: Uranus, a three DOF WMR constructed in the Mobile Robot Laboratory and Bicsun-Bicas, a two DOF WMR of our design which has the kinematic structure of a wheel chair. We have found that a servo-controller based entirely upon the kinematic model is adequate for control of Uranus. In contrast, the servo-controller design must incorporate the dynamic model for adequate control of Bicsun-Bicas. We conclude that the dynamic coupling between the two dependent WMR axes of motion and the nonlinear variations of the internal forces/torques due to the motion of the castors require dynamic compensation to enable control. The kinematically simpler, two DOF WMR is thus conceptually, and computationally more complex to control.

Our research has addressed the areas of kinematic and dynamic modeling, mobility characterization, mechanical design, servo-control and simulation. Through this comprehensive study we have concluded that the Bicsun-Bicas design is superior to others for general-purpose trajectory following applications on hard smooth surfaces such as indoor parts and materials transport and manipulation. We detail the advantages of this structure in Chapter 8.

1.8 Outline of the Dissertation

In this dissertation, we advance the understanding of WMRs, from the motivation of the kinematic and dynamic modeling methodologies, through their application to WMR servo-control, to computer simulation and controller evaluation studies.

In Chapter 2, we develop our methodology for modeling the kinematics of WMRs. Coordinate systems are assigned to prescribed positions on the the robot. We compute transformation matrices to characterize the translations and rotations between coordinate systems. We develop a matrix coordinate transformation algebra to compute the position, velocity, and acceleration relationships between coordinate systems. We apply the axioms and corollaries of this algebra to transform positions, velocities, and accelerations, specified in one coordinate system to another coordinate system, and develop the wheel Jacobian matrix to relate the motions of a wheel to the motions of the WMR body. The Jacobians of all of the wheels on a WMR are applied to compute the actuated inverse and sensed forward velocity solutions.

In Chapter 3, we detail our dynamic WMR modeling methodology. We formulate a unifying framework for modeling the dynamics of any simple closed-chain mechanical system of rigid links incorporating the special characteristics of WMRs. We apply classical methods for the solution of systems of linear algebraic equations to compute the forward and actuated inverse dynamic solutions for simulation and servo-control applications.

In Chapter 4, we design model-based WMR servo-controllers. We apply the conventional algorithms for stationary manipulator servo-control to WMR

servo-control. We demonstrate that stationary manipulator servo-controller designs are directly applicable to WMR control if accurate kinematic and dynamic WMR models are constructed. We develop and apply actuator command conversion algorithms to interface resolved motion rate and resolved acceleration servo-controllers to pulse-width modulated wheel motors.

In Chapter 5, we describe the computer simulation of WMR dynamics. We apply the forward dynamic WMR solution to compute the WMR accelerations from the wheel actuator torques in each simulation time step. A fourth-order Runge-Kutta integration routine with adaptive step size is programmed to provide accurate simulation performance. We describe and program simulation variables and servo-controller performance measures into our computer simulation package for subsequent WMR servo-controller evaluation.

Chapters 6 and 7 are the culmination of the WMR modeling and control dissertation. The kinematic and dynamic models in Chapters 2 and 3, the servo-controller designs in Chapter 4, the simulation program and servo-controller evaluation methodology in Chapter 5 are applied to two specific prototype WMRs. Through our two case studies, we exemplify and demonstrate the applicability of the procedures developed in this dissertation. The unified applications of the modeling and control methodologies produces a coherent, in-depth understanding of the operation of the WMRs. The knowledge thereby obtained is supported through simulation experiments.

In Chapter 6, we undertake the modeling and servo-control of the prototype three DOF WMR Uranus. The kinematic and dynamic models are developed and solved according to the methodologies developed in Chapters 2 and 3. Kinematic and dynamics-based servo-controllers are designed from the models of Uranus. Finally, Uranus is simulated under model-based servo-control to evaluate the relative performances of resolved motion rate and resolved acceleration servo-controller designs. In Chapter 7, Bicsun-Bicas, a prototype two DOF WMR of our design is similarly modeled, controlled and evaluated. We conclude that Uranus may be adequately controlled for most applications with resolved motion rate servo-control, but that resolved acceleration servo-control is required to stabilize Bicsun-Bicas because of the inherent differences in the dynamics of the two WMRs.

We advance our conclusions, enumerate the contributions, and suggest areas of future research in Chapter 8.

Fifteen appendices are included at the end of this dissertation to support topics related to our WMR modeling and control research. We introduce in Appendix 1 a nomenclature and symbolic representation for WMRs. In Appendix 2, we survey past and present WMRs. We describe in Appendix 3 the differing wheel types and compile their Jacobian matrices. In Appendix 4, we summarize background information required for the solution of systems of linear algebraic equations. We document in Appendices 5 and 6 the detailed matrix computations required for the actuated inverse and sensed forward velocity solutions. In Appendices 7 through 10, we apply the WMR kinematic equations-of-motion to WMR mobility characterization, mechanical design, dead reckoning, and wheel slip detection. We model in Appendix 11 the kinematics of six exemplary WMRs. In Appendix 12, we compile the (6×6) kinematic matrices required for dynamic modeling. In Appendix 13, we describe how flexible links may be modeled and incorporated in our dynamic WMR model. In Appendix 14, we demonstrate the graphical representation of our dynamic modeling framework. Finally, in Appendix 15, we apply our dynamic modeling framework to two mechanically simple systems: a double pendulum and a biped in the frontal plane, and show agreement with the dynamic models of these two systems documented in the literature. Our bibliography follows the appendices.

Chapter 2
Kinematic Modeling of WMRs

2.1 Introduction

A kinematic model of a robotic mechanism lays the foundation for dynamic modeling and model-based servo-control. Kinematic modeling is the formulation of the constraints between the positions, velocities and accelerations of the components of a mechanism. A WMR kinematic model defines the constraints between the positions, velocities, and accelerations of the WMR body, wheels and steering links. Our WMR kinematic modeling methodology forms the basis for our design of resolved motion rate WMR control algorithms in Chapter 4.

Our kinematic analysis of WMRs parallels the development of kinematics for stationary manipulators [Paul81a, Brady82, McInnis86]. In this chapter, we apply and extend manipulator kinematic modeling methodology to incorporate the special WMR characteristics discussed in Section 1.5; namely higher-pair joints, closed-link chains and unactuated and unsensed joints. We summarize manipulator kinematic modeling methodology in Section 2.2. We then apply and extend this methodology in Section 2.3 to develop a WMR kinematic modeling methodology. In Section 2.4, we solve the kinematic model and interpret the solutions. We enumerate our step-by-step WMR kinematic modeling procedure in Section 2.5 and summarize the development in Section 2.6. We illustrate the application of our kinematic modeling methodology in four appendices. The kinematic model is applied to mobility characterization (in Appendix 7), mechanical design (in Appendix 8), dead reckoning (in Appendix 9), and wheel slip detection (in Appendix 10).

2.2 Manipulator Kinematic Modeling Methodologies

A standard method for modeling the kinematics of stationary robotic manipulators begins by applying the Denavit-Hartenberg convention [Denavit55] to assign coordinate axes to each of the robot joints. *Successive* coordinate systems on the robot are related by (4×4) homogeneous transformation **A**-matrices. The **A**-matrices are specified completely by four characteristic parameters (two displacements and two rotations) between consecutive coordinate systems. Each **A**-matrix describes both the shape and size of a robot link, and the translation (for a prismatic joint) or rotation (for a rotational joint) of the associated joint.

The position and orientation in base coordinates of the end-effector of a stationary manipulator is found by cascading the **A**-matrices *from* the base link *to* the end-effector [Pieper68]. Velocity relationships are found by differentiating the position relationships [Denavit65]. Velocities of the individual joints are related to the velocities of the end-effector by the manipulator Jacobian matrix [Paul81a] in the *forward velocity solution.* The inverse Jacobian matrix is applied in the *inverse velocity solution* to compute the velocities of the joint variables from the velocities of the end-effector. Accelerations of the individual joints are related to the accelerations of the end-effector by differentiating the velocity equations-of-motion.

2.3 A WMR Kinematic Modeling Methodology

2.3.1 Overview

In this section, we apply and extend standard robotic methodology (summarized in Section 2.2) to model the *kinematics* of WMRs [Muir87b]. The novel aspects are our treatment of the higher-pair joint between each wheel and the floor, the development of a transformation matrix algebra, and the formulation of a wheel Jacobian matrix.

We begin our development by enumerating our kinematic modeling assumptions in Section 2.3.2. In Section 2.3.3, we apply the Sheth-Uicker convention [Sheth71] to assign coordinate systems to the WMR body, wheels and steering links to facilitate kinematic modeling. The Sheth-Uicker convention

allows us to model the *higher-pair* relationship between each wheel on a WMR and the floor. To model the higher-pair joints at the point-of-contact between each wheel and the floor, we define *instantaneously coincident coordinate systems* (in Section 2.3.4).

In Section 2.3.5, we assign homogeneous (4×4) transformation matrices to relate coordinate systems. We also present a *matrix coordinate transformation algebra* to formulate the kinematic equations-of-motion of a WMR. All of the kinematics in this chapter are derived by straightforward application of the axioms and corollaries of our matrix coordinate transformation algebra. The WMR characteristic kinematic parameters required to compute the transformation matrices are enumerated in Section 2.3.6. In Section 2.3.7, we demonstrate that transforming the coordinates of a point between coordinate systems is equivalent to finding a path in a position transformation graph.

In Section 2.3.8, we formulate the velocity kinematics. The relationships between the wheel velocities and the WMR body velocities are linear. We thus develop a *wheel Jacobian matrix* to calculate the vector of WMR body velocities from the vector of wheel velocities. We formulate the wheel acceleration kinematics in Section 2.3.9. In Section 2.3.10, we transform velocities and accelerations from the WMR body coordinate system to the floor coordinate system. Finally, in Section 2.3.11, we present our concluding remarks.

2.3.2 WMR Kinematic Modeling Assumptions

We begin by enumerating our kinematic modeling assumptions to constrain the class of mobile robots to which our kinematic modeling methodology applies. To include all existing and foreseeable WMRs, we would have to generalize our methodology and thereby complicate the modeling of the overwhelming majority of WMRs. We introduce the practical assumptions in Table 2.1 to make the modeling problem tractable.

We discuss our assumptions in turn. Assumption 1 states that the dynamics of such WMR components as flexible suspension mechanisms and tires are negligible. We make this assumption to apply rigid body mechanics to kinematic modeling. We recognize that flexible structures may play a significant

role in the kinematic analysis of WMRs. A dynamic analysis to determine the changes in kinematic structure due to forces/torques acting on flexible components is required to model these components. Such an analysis is appropriate for WMRs even though it has not conventionally been addressed for stationary open-chain manipulators because WMRs are inherently *closed-chain* mechanisms. Flexible components, that allow compliance in the multiple closed-chains of a WMR, lead to a consistent kinematic model. Without compliant structures, there cannot be a consistent kinematic model for WMRs in the presence of surface irregularities, inexact component dimensions and inexact control actuation [Muir84a]. We introduce a dynamic model of flexible components in Appendix 13 which is compatible with our dynamic modeling methodology in Chapter 3. A simultaneous kinematic and dynamic analysis of WMRs is thus a natural continuation of our research.

Table 2.1: WMR Kinematic Modeling Assumptions

Design Assumptions

(1) The WMR does not contain flexible parts.
(2) There is zero or one steering link per wheel.
(3) All steering axes are perpendicular to the surface-of-travel.

Operational Assumptions

(4) The WMR moves on a planar surface.
(5) The translational friction at the point-of-contact between a wheel and the surface-of-travel is large enough so that no translational slip may occur.
(6) The rotational friction at the point-of-contact between a wheel and the surface-of-travel is small enough so that rotational slip may occur.

We introduce Assumptions 2 and 3 to reduce the range of WMRs that our methodology must address, by limiting the complexity of our kinematic model. WMRs which have more than one link per wheel can be analyzed by our

methodology if only one steering link is allowed to move. We require that all non-steering links must be stationary, as if they are extensions of the robot body or wheel mounts. By constraining the steering links to be perpendicular to the surface-of-travel in Assumption 3, we reduce all motions to a plane[1]. We thus constrain all component motions to a rotation about the normal to the surface-of-travel, and two translations in a plane parallel to the surface.

Assumption 4 neglects irregularities in the actual surface on which a WMR travels. Even though this assumption restricts the range of practical applications, environments which do not satisfy this assumption (e.g., rough, bumpy or rocky surfaces) do not lend themselves to energy efficient wheeled vehicle travel [Bekker69].

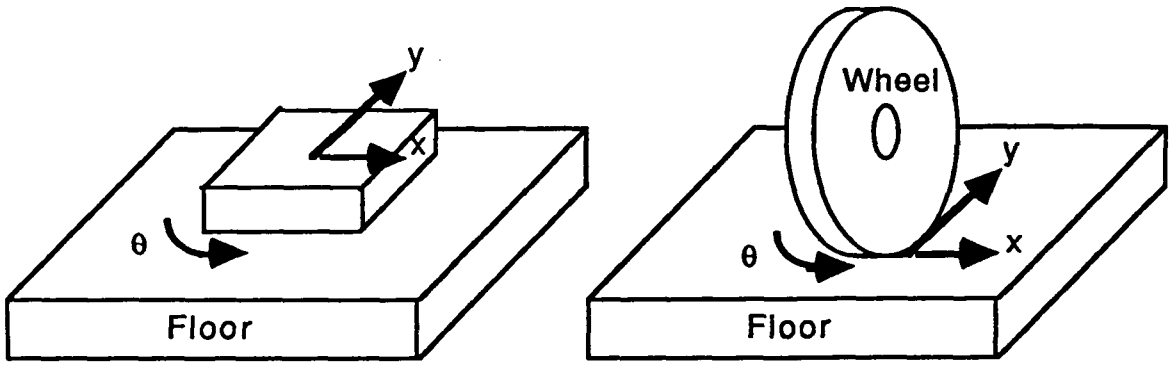
Assumption 5 ensures the applicability of the theoretical kinematic properties of a wheel in rolling contact [Beggs66, Shigley69] for the two translational DOFs. This assumption is realistic for dry surfaces as demonstrated by the success of braking mechanisms on automobiles. Automobiles also illustrate the practicality of Assumption 6. The wheels must rotate (i.e., slip) about their points-of-contact to navigate a turn. Since WMRs also rely on rotational wheel slip, we include Assumption 6.

2.3.3 WMR Kinematic Modeling Coordinate Systems

Coordinate system assignment is the first step in the kinematic modeling of a stationary manipulator [Paul81a]. Lower-pair mechanisms (such as revolute and prismatic joints) function with two surfaces in relative motion. In contrast, the wheels of a WMR are higher-pairs which function ideally by point contact. Because the **A**-Matrices which model manipulators depend upon the relative position and orientation of two successive joints, the Denavit-Hartenberg convention [Denavit55] leads to ambiguous assignments of coordinate transformation matrices in multiple closed-chains [Sheth71] which are inherent in WMRs. The ambiguity arises in deciding the joint ordering when a single link connects more than two joints.

---

1 For our WMR dynamic modeling in Chapter 3, we detail kinematic computations to model six DOF motion.



Block on Floor - An Example of Planar-Pair Motions

Analogous Wheel Planar-Pair Motions

Figure 2.1: Planar-Pair Model of a Wheel

We apply the Sheth-Uicker convention [Sheth71] to assign coordinate systems and model each wheel as a planar-pair at the point-of-contact. This convention allows the modeling of the higher-pair wheel motion and eliminates ambiguities in coordinate transformation matrices. The planar-pair allows three DOFs as shown in Figure 2.1: x and y translation, and rotation about the point-of-contact. The Sheth-Uicker convention is ideal for modeling ball wheels; the angular velocities of the wheel are converted directly into translational velocities along the surface. The planar-pair motions must be constrained to include wheels which do not allow three DOFs. For example, the coordinate system assigned at the point-of-contact of a conventional wheel is aligned with the y-axis parallel to the wheel. The wheel model is completed by constraining the x-component of the wheel velocity to zero to satisfy Assumption 5 (in Section 2.3.2) and avoid translational slip. In Appendix 3, we detail the operation of conventional, steered conventional, omnidirectional, and ball wheels.

We assign coordinate systems at both ends of each *link* of the WMR. The links of the closed-chain of a WMR are the floor, the robot body and the steering links. The *joints* are: a revolute-pair at each steering axis, a planar-pair to model each wheel, and a planar-pair to model the robot body. When the joint variables are zero, the coordinate systems of the two links which share the joint coincide. The Sheth-Uicker convention separates the *constant* shape and size parameters from the *variable* wheel joint parameters, and simplifies the subsequent model formulation. In Table 2.2, we summarize our coordinate system assignments for a WMR having N wheels. Placement of the coordinate

systems is illustrated in Figure 2.2 for the pictorial view of a WMR. For a WMR with N wheels, we assign 3N+1 coordinate systems to the robot and one stationary reference coordinate system. There are also N+1 instantaneously coincident coordinate systems (described in Section 2.3.4) which need not be assigned explicitly.

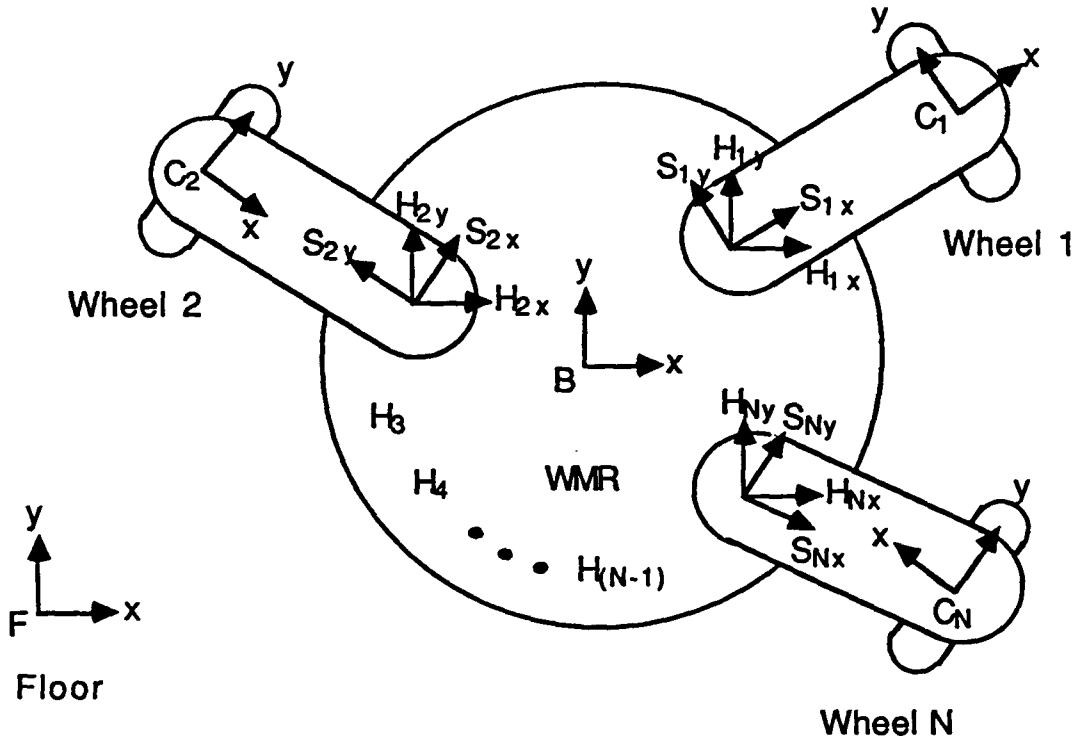


Figure 2.2: WMR Kinematic Modeling Coordinate System Assignments

The *floor* coordinate system F is stationary relative to the surface-of-travel and serves as the reference coordinate system for robot motions. The *body* coordinate system B is assigned to the robot body so that the position of the WMR is the displacement from the floor coordinate system to the body coordinate system. The *hip* coordinate system H_i is assigned at the point on the robot body which intersects the steering axis of wheel i. The *steering* coordinate system S_i is assigned at the same point along the steering axis of wheel i, but is fixed relative to the steering link. We assign a *contact point* coordinate system C_i at the point-of-contact between each wheel and the floor.

Table 2.2: WMR Kinematic Modeling Coordinate System Assignments

F *Floor*: Stationary reference coordinate system with the z-axis orthogonal to the surface-of-travel.

B *Body* Coordinate system which moves with the WMR body, with the z-axis orthogonal to the surface-of-travel.

H_i *Hip* (for steered wheels i): Coordinate system which moves with the WMR body, with the z-axis coincident with the axis of steering joint i.

S_i *Steering* (for steered wheels i): Coordinate system which moves with steering link i, with the z-axis coincident with the z-axis of H_i, and the origin coincident with the origin of H_i.

C_i *Contact Point* (for i=1,...,N): Coordinate system which moves with the wheel i point-of-contact, with the origin at the point-of-contact between the wheel and the surface-of-travel; the y-axis is parallel to the wheel (if the wheel has a preferred orientation; if not, the y-axis is assigned arbitrarily) and the x-y plane is tangent to the surface-of-travel.

$\overline{B}$ *Instantaneously Coincident Body:* Coordinate system coincident with the B coordinate system and stationary relative to the F coordinate system. The coordinate system need not be assigned explicitly.

$\overline{C}_i$ *Instantaneously Coincident Contact Point* (for i=1,...,N): Coordinate system coincident with the C_i coordinate system and stationary relative to the F coordinate system. The coordinate system need not be assigned explicitly.

Coordinate system assignments are not unique. There is freedom to assign the coordinate systems at positions and orientations which lead to convenient structures of the kinematic model. For example, all of the hip coordinate systems may be assigned parallel to the robot coordinate system resulting in *sparse* robot-hip transformation matrices and thus simplifying the model (as exemplified in Chapters 6 and 7 and Appendix 11). Alternatively, the

x-axes of the hip coordinate systems can be aligned with the zero position of the steering joint position encoders to express the hip-steering transformation in terms of the actual steering angle.

For stationary serial link manipulators, all joints are one-dimensional lower-pairs: prismatic joints allow z motion and revolute joints allow θ motion. In contrast, WMRs have three-dimensional higher-pair wheel-to-floor and body-to-floor joints allowing simultaneous x, y and θ motions. In Section 2.3.4, we introduce instantaneously coincident coordinate systems to model these higher-pair joints.

2.3.4 Instantaneously Coincident Coordinate Systems

To introduce the concept of instantaneously coincident coordinate systems, we consider the one-dimensional example of a ball rolling in a straight line on a flat surface. The position of the ball is depicted by the point b in Figure 2.3.

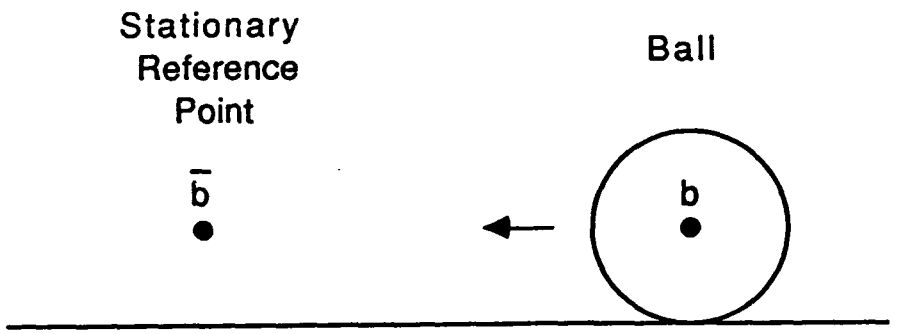


Figure 2.3: Ball in Motion Before Instantaneous Coincidence

The ball is moving from right to left with velocity v_b and acceleration a_b. The stationary reference point $\bar{b}$ lies in the path of the moving ball. At the *instant* the ball (point b) and the reference (point $\bar{b}$) *coincide* in Figure 2.4, we observe that: (1) The position of the ball relative to the reference point ${}^{\bar{b}}p_b$ is zero; and (2) The velocity ${}^{\bar{b}}v_b$ and acceleration ${}^{\bar{b}}a_b$ of the ball relative to the reference point are non-zero. We call the point $\bar{b}$ an *instantaneously coincident* reference point for the moving ball at the instant shown in Figure 2.4.

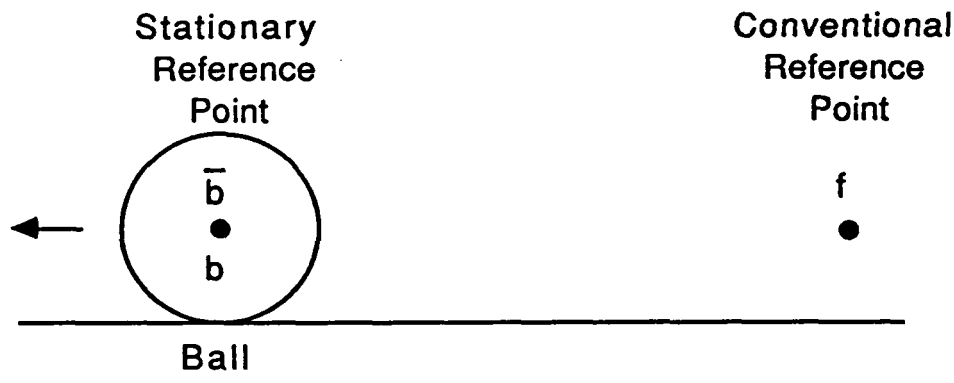


Figure 2.4: Ball in Motion at Instantaneous Coincidence

We continuously assign an instantaneously coincident reference point $\bar{b}$ during the motion of the ball to generalize our observations for all time t. The position of the ball relative to its instantaneously coincident reference point is zero (i.e., ${}^{\bar{b}}p_b(t)=0$), and the velocity and acceleration of the ball relative to its instantaneously coincident reference point are non-zero (i.e., ${}^{\bar{b}}v_b(t)\neq 0$ and ${}^{\bar{b}}a_b(t)\neq 0$). In the framework of instantaneously coincident reference points, we emphasize that we *cannot* differentiate the position (velocity) equation-of-motion to obtain the velocity (acceleration) equation-of-motion.

The stationary reference point f in Figure 2.4 is a conventional reference point whose position is fixed. Since both reference points f and $\bar{b}$ are stationary, the velocity (acceleration) of the ball relative to the point f is equal to the velocity (acceleration) of the ball relative to the point $\bar{b}$ in this one-dimensional example. Consequently, it is not advantageous to introduce instantaneously coincident references in the one-dimensional example. The practical need for instantaneously coincident coordinate systems arises in the multi-dimensional example depicted in Figure 2.5.

The coordinate system B is moving in three-dimensions: x, y, and θ. The coordinate systems $\bar{B}$ and F are stationary; $\bar{B}$ is an instantaneously coincident coordinate system and F is a conventional reference coordinate system. We make the analogous observations. The position of the moving coordinate system B relative to its instantaneously coincident coordinate system $\bar{B}$ is zero (i.e., ${}^{\bar{B}}\mathbf{p}_B=0$). The position of the moving coordinate system B relative to the conventional reference coordinate system F is non-zero (i.e., ${}^{F}\mathbf{p}_B\neq 0$). The non-zero velocity ${}^{\bar{B}}\mathbf{v}_B$ (acceleration ${}^{\bar{B}}\mathbf{a}_B$) of the moving coordinate system relative to

the instantaneously coincident coordinate system is not equal to the velocity ${}^{F}\mathbf{v}_{B}$ (acceleration ${}^{F}\mathbf{a}_{B}$) of the moving coordinate system relative to the conventional reference coordinate system. Instead, the velocity (acceleration) of the moving coordinate system relative to the conventional reference coordinate system F depends upon the position and orientation of the moving coordinate system *relative* to the reference coordinate system. *The motivation for assigning instantaneously coincident coordinate systems is that the velocities (accelerations) of a multi-dimensional moving coordinate system can be computed or specified independently of the position of the moving coordinate system.* The instantaneously coincident coordinate system is a conceptual tool which enables us to compute the velocities and accelerations of a moving coordinate system relative to its instantaneous current position and orientation.

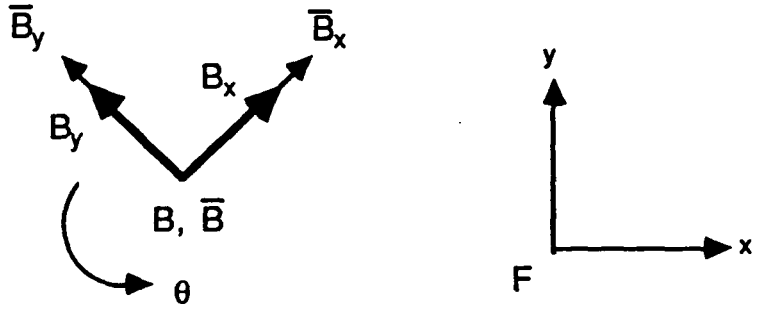


Figure 2.5: Coordinate System B in Motion

In the preceding discussion, we recognize that the moving coordinate system may play the role of the robot body coordinate system B, and the conventional reference coordinate system F may play the role of the floor coordinate system. We assign an *instantaneously coincident body* coordinate system $\bar{B}$ (in Table 2.2) at the same position and orientation in space as the body coordinate system B. We define the instantaneously coincident body coordinate system to be stationary relative to the floor coordinate system F. By design, the position and orientation of the body coordinate system B and the instantaneously coincident body coordinate system $\bar{B}$ are identical, but (in general) the relative velocities and accelerations between the two coordinate systems are non-zero. When the body coordinate system moves relative to the floor coordinate system, we assign a different instantaneously coincident

coordinate system for each time instant. The instantaneously coincident body coordinate system facilitates the specification of WMR velocities (accelerations) independently of the WMR position. Similarly, the *instantaneously coincident contact point* coordinate system $\overline{C}_i$ (in Table 2.2) coincides with the contact point coordinate system C_i and is stationary relative to the floor coordinate system.

2.3.5 Matrix Coordinate Transformation Algebra

Homogeneous (4 × 4) transformation matrices are defined to express the relative positions and orientations of coordinate systems [Paul81a]. The homogeneous transformation matrix ${}^{A}\Pi_{B}$ transforms the coordinates of the point ${}^{B}\mathbf{r}$ in coordinate system B to its corresponding coordinates ${}^{A}\mathbf{r}$ in coordinate system A:

$$ {}^{A}\mathbf{r} = {}^{A}\Pi_{B}\, {}^{B}\mathbf{r} . \qquad (2.3.1) $$

Vectors denoting points in space, such as ${}^{A}\mathbf{r}$ in (2.3.1), consist of three Cartesian coordinates and a scale factor as the fourth element:

$$ {}^{A}\mathbf{r} = \begin{pmatrix} {}^{A}r_x \\ {}^{A}r_y \\ {}^{A}r_z \\ 1 \end{pmatrix} . \qquad (2.3.2) $$

We always use a scale factor of unity. Transformation matrices contain the (3 × 3) rotational matrix $(\mathbf{n} \;\; \mathbf{o} \;\; \mathbf{a})$, and the (3 × 1) translational vector $\mathbf{p}$ [Paul81a]:

$$ {}^{A}\Pi_{B} = \begin{pmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{pmatrix} . \qquad (2.3.3) $$

The three vector components **n**, **o**, and **a** of the rotational matrix in (2.3.3) express the orientation of the x, y, and z-axes, respectively, of the B coordinate system relative to the A coordinate system and are thus orthonormal. The three components p_x, p_y, and p_z of the translational vector **p** express the

displacement of the origin of the B coordinate system relative to the origin of the A coordinate system along the x, y, and z-axes of the A coordinate system, respectively.

These celebrated properties guarantee that the inverse of a transformation matrix always has the special form:

$$ {}^{A}\Pi_{B}^{-1} = \begin{pmatrix} n_x & n_y & n_z & -(\mathbf{p}\cdot\mathbf{n}) \\ o_x & o_y & o_z & -(\mathbf{p}\cdot\mathbf{o}) \\ a_x & a_y & a_z & -(\mathbf{p}\cdot\mathbf{a}) \\ 0 & 0 & 0 & 1 \end{pmatrix}. \qquad (2.3.4) $$

Before we define the transformation matrices between the coordinate systems of our WMR model, we compile in Table 2.3 our nomenclature for rotational and translational displacements, velocities and accelerations. Since any two coordinate systems A and B in our WMR kinematic model are located at non-zero x, y and z-coordinates relative to each other, the transformation matrix must therefore contain the translations ${}^{A}d_{Bx}$, ${}^{A}d_{By}$ and ${}^{A}d_{Bz}$. We assign all kinematic modeling coordinate systems with the z-axes perpendicular to the surface-of-travel so that all rotations between coordinate systems are about the z-axis. A transformation matrix in our WMR kinematic model thus embodies a rotation ${}^{A}\theta_{B}$ about the z-axis of coordinate system A and the translations ${}^{A}d_{Bx}$, ${}^{A}d_{By}$ and ${}^{A}d_{Bz}$ along the respective coordinate axes:

$$ {}^{A}\Pi_{B} = \begin{pmatrix} \cos{}^{A}\theta_{B} & -\sin{}^{A}\theta_{B} & 0 & {}^{A}d_{Bx} \\ \sin{}^{A}\theta_{B} & \cos{}^{A}\theta_{B} & 0 & {}^{A}d_{By} \\ 0 & 0 & 1 & {}^{A}d_{Bz} \\ 0 & 0 & 0 & 1 \end{pmatrix}. \qquad (2.3.5) $$

For zero rotational and translational displacements, the coordinate transformation matrix in (2.3.5) reduces to the identity matrix. By applying the inverse in (2.3.4) to the transformation matrix in (2.3.5), we obtain

$$ {}^{A}\Pi_{B}^{-1} = \begin{pmatrix} \cos{}^{A}\theta_{B} & \sin{}^{A}\theta_{B} & 0 & -{}^{A}d_{Bx}\cos{}^{A}\theta_{B} - {}^{A}d_{By}\sin{}^{A}\theta_{B} \\ -\sin{}^{A}\theta_{B} & \cos{}^{A}\theta_{B} & 0 & {}^{A}d_{Bx}\sin{}^{A}\theta_{B} - {}^{A}d_{By}\cos{}^{A}\theta_{B} \\ 0 & 0 & 1 & -{}^{A}d_{Bz} \\ 0 & 0 & 0 & 1 \end{pmatrix}. \qquad (2.3.6) $$

Table 2.3: Scalar Rotational and Translational Variables

Displacements

$^{A}\theta_{B}$: The rotational displacement about the z-axis of the A coordinate system between the x-axis of the A coordinate system and the x-axis of the B coordinate system (counterclockwise by convention).

$^{A}d_{Bj}$: (for $j \in [x, y, z]$) : The translational displacement along the j-axis of the A coordinate system between the origin of the A coordinate system and the origin of the B coordinate system.

Velocities

$^{A}\omega_{B}$: The rotational velocity $^{A}\dot{\theta}_{B}$ about the z-axis of the A coordinate system between the x-axis of the A coordinate system and the x-axis of the B coordinate system.

$^{A}v_{Bj}$: (for $j \in [x, y]$) : The translational velocity $^{A}\dot{d}_{Bj}$ along the j-axis of the A coordinate system between the origin of the A coordinate system and the origin of the B coordinate system. Since all motion is in the x-y plane, the z-component $^{A}\dot{d}_{Bz}$ of the translational velocity is zero.

Accelerations

$^{A}\alpha_{B}$: The rotational acceleration $^{A}\ddot{\theta}_{B} = {}^{A}\dot{\omega}_{B}$ about the z-axis of the A coordinate system between the x-axis of the A coordinate system and the x-axis of the B coordinate system.

$^{A}a_{Bj}$: (for $j \in [x, y]$) : The translational acceleration $^{A}\ddot{d}_{Bj} = {}^{A}\dot{v}_{Bj}$ along the j-axis of the A coordinate system between the origin of the A coordinate system and the origin of the B coordinate system. Since all motion is parallel to the x-y plane, the z-component $^{A}\ddot{d}_{Bz}$ of the translational acceleration is zero.

In Section 2.3.8, we differentiate[2] the transformation matrix in (2.3.5) componentwise to compute robot velocities:

$$ {}^{A}\dot{\Pi}_{B} = \begin{pmatrix} -{}^{A}\omega_{B}\sin{}^{A}\theta_{B} & -{}^{A}\omega_{B}\cos{}^{A}\theta_{B} & 0 & {}^{A}v_{Bx} \\ {}^{A}\omega_{B}\cos{}^{A}\theta_{B} & -{}^{A}\omega_{B}\sin{}^{A}\theta_{B} & 0 & {}^{A}v_{By} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \qquad (2.3.7) $$

and in Section 2.3.9, we differentiate the transformation matrix in (2.3.7) componentwise to compute robot accelerations:

$$ {}^{A}\ddot{\Pi}_{B} = \begin{pmatrix} -{}^{A}\alpha_{B}\sin{}^{A}\theta_{B} - {}^{A}\omega_{B}^{2}\cos{}^{A}\theta_{B} & -{}^{A}\alpha_{B}\cos{}^{A}\theta_{B} + {}^{A}\omega_{B}^{2}\sin{}^{A}\theta_{B} & 0 & {}^{A}a_{Bx} \\ {}^{A}\alpha_{B}\cos{}^{A}\theta_{B} - {}^{A}\omega_{B}^{2}\sin{}^{A}\theta_{B} & -{}^{A}\alpha_{B}\sin{}^{A}\theta_{B} - {}^{A}\omega_{B}^{2}\cos{}^{A}\theta_{B} & 0 & {}^{A}a_{By} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}. \qquad (2.3.8) $$

The Sheth-Uicker assignment of coordinate systems results in *constant* and *variable* transformation matrices between coordinate systems. The transformation matrix between coordinate systems fixed at two different positions on the same link is constant. Transformation matrices relating the position and orientation of coordinate systems on different links include joint variables and thus are variable. Constant and variable transformation matrices are denoted by ${}^{A}\mathbf{T}_{B}$ and ${}^{A}\Phi_{B}$, respectively [Sheth71]. In Tables 2.4 and 2.5, we compile the constant and variable transformation matrices in our WMR kinematic model, respectively. The *constant* transformation matrices are the floor-$\bar{\text{b}}$ody transformation (${}^{F}\mathbf{T}_{\bar{B}}$), the floor-$\bar{\text{c}}$ontact transformation (${}^{F}\mathbf{T}_{\bar{C}_i}$), the body-hip transformation (${}^{B}\mathbf{T}_{H_i}$) for steered wheels, the steering-contact transformation (${}^{S_i}\mathbf{T}_{C_i}$) for steered wheels and the body-contact transformation (${}^{B}\mathbf{T}_{C_i}$) for non-steered wheels. Since the instantaneously coincident coordinate systems $\bar{B}$ and $\bar{C}_i$ are stationary relative to the floor coordinate system, all transformation matrices between the floor coordinate system and the instantaneously coincident coordinate systems are constant. The *variable* transformation

---

2 All differentiation is computed within the reference coordinate system of the pre-superscript; i.e., ${}^{A}\dot{\Pi}_{B} = \frac{{}^{A}d}{dt}{}^{A}\Pi_{B}$.

matrices are the body-body transformation (${}^{\bar{B}}\Phi_B$), the hip-steering transformation (${}^{H_i}\Phi_{S_i}$) and the contact-contact transformation (${}^{\bar{C}_i}\Phi_{C_i}$). The transformation matrix from a coordinate system to its instantaneously coincident counterpart (or visa-versa) is variable because there is relative motion.

We compile the first and second time-derivatives of the variable transformation matrices in Tables 2.6 and 2.7, respectively. The matrix derivatives involving instantaneously coincident coordinate systems (i.e., ${}^{\bar{B}}\dot{\Phi}_B$, ${}^{\bar{C}_i}\dot{\Phi}_{C_i}$, ${}^{\bar{B}}\ddot{\Phi}_B$, and ${}^{\bar{C}_i}\ddot{\Phi}_{C_i}$) are formed by differentiating and simplifying the elements of the transformation matrices ${}^{\bar{R}}\Phi_R$ and ${}^{\bar{C}_i}\Phi_{C_i}$, respectively, by substituting ${}^{\bar{B}}\theta_B=0$ and ${}^{\bar{C}_i}\theta_{C_i}=0$. Because of the simplifying substitutions, the second time-derivative of a transformation matrix involving an instantaneously coincident coordinate system cannot be obtained by differentiating the first time-derivative. The time-derivatives of constant transformation matrices are zero.

The kinematics of stationary manipulators are modeled by exploiting the properties of transformation matrices [Denavit65]. We formalize the manipulation of transformation matrices in the presence of instantaneously coincident coordinate systems by defining a *matrix coordinate transformation algebra*. The algebra consists of a set of operands and a set of operations which may be applied to the operands. The operands of matrix coordinate transformation algebra are transformation matrices and their first and second time-derivatives. The operations are listed in Table 2.8 as seven axioms. In Table 2.8, A, B, and X are coordinate systems and Π denotes either a constant T transformation matrix or a variable Φ transformation matrix. Matrix coordinate transformation algebra allows the direct calculation of the relative positions, velocities and accelerations of robot coordinate systems (including instantaneously coincident coordinate systems).

Table 2.4: Constant Transformation Matrices

Floor-$\bar{B}$ody Transformation:

$$ {}^{F}T_{\bar{B}} = \begin{pmatrix} \cos{}^{F}\theta_{\bar{B}} & -\sin{}^{F}\theta_{\bar{B}} & 0 & {}^{F}d_{\bar{B}x} \\ \sin{}^{F}\theta_{\bar{B}} & \cos{}^{F}\theta_{\bar{B}} & 0 & {}^{F}d_{\bar{B}y} \\ 0 & 0 & 1 & {}^{F}d_{\bar{B}z} \\ 0 & 0 & 0 & 1 \end{pmatrix} $$

Floor-$\bar{C}$ontact Transformation:

$$ {}^{F}T_{\bar{C}_i} = \begin{pmatrix} \cos{}^{F}\theta_{\bar{C}_i} & -\sin{}^{F}\theta_{\bar{C}_i} & 0 & {}^{F}d_{\bar{C}_ix} \\ \sin{}^{F}\theta_{\bar{C}_i} & \cos{}^{F}\theta_{\bar{C}_i} & 0 & {}^{F}d_{\bar{C}_iy} \\ 0 & 0 & 1 & {}^{F}d_{\bar{C}_iz} \\ 0 & 0 & 0 & 1 \end{pmatrix} $$

Body-Hip Transformation:

$$ {}^{B}T_{H_i} = \begin{pmatrix} \cos{}^{B}\theta_{H_i} & -\sin{}^{B}\theta_{H_i} & 0 & {}^{B}d_{H_ix} \\ \sin{}^{B}\theta_{H_i} & \cos{}^{B}\theta_{H_i} & 0 & {}^{B}d_{H_iy} \\ 0 & 0 & 1 & {}^{B}d_{H_iz} \\ 0 & 0 & 0 & 1 \end{pmatrix} $$

Steering-Contact Transformation:

$$ {}^{S_i}T_{C_i} = \begin{pmatrix} \cos{}^{S_i}\theta_{C_i} & -\sin{}^{S_i}\theta_{C_i} & 0 & {}^{S_i}d_{C_ix} \\ \sin{}^{S_i}\theta_{C_i} & \cos{}^{S_i}\theta_{C_i} & 0 & {}^{S_i}d_{C_iy} \\ 0 & 0 & 1 & {}^{S_i}d_{C_iz} \\ 0 & 0 & 0 & 1 \end{pmatrix} $$

Body-Contact Transformation:

$$ {}^{B}T_{C_i} = \begin{pmatrix} \cos{}^{B}\theta_{C_i} & -\sin{}^{B}\theta_{C_i} & 0 & {}^{B}d_{C_ix} \\ \sin{}^{B}\theta_{C_i} & \cos{}^{B}\theta_{C_i} & 0 & {}^{B}d_{C_iy} \\ 0 & 0 & 1 & {}^{B}d_{C_iz} \\ 0 & 0 & 0 & 1 \end{pmatrix} $$

Table 2.5: Variable Transformation Matrices

$\bar{B}$ody-Body Transformation:

$$ {}^{\bar{B}}\Phi_{B} = \begin{pmatrix} \cos{}^{\bar{B}}\theta_{B} & -\sin{}^{\bar{B}}\theta_{B} & 0 & {}^{\bar{B}}d_{Bx} \\ \sin{}^{\bar{B}}\theta_{B} & \cos{}^{\bar{B}}\theta_{B} & 0 & {}^{\bar{B}}d_{By} \\ 0 & 0 & 1 & {}^{\bar{B}}d_{Bz} \\ 0 & 0 & 0 & 1 \end{pmatrix} $$

Hip-Steering Transformation:

$$ {}^{H_i}\Phi_{S_i} = \begin{pmatrix} \cos{}^{H_i}\theta_{S_i} & -\sin{}^{H_i}\theta_{S_i} & 0 & 0 \\ \sin{}^{H_i}\theta_{S_i} & \cos{}^{H_i}\theta_{S_i} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} $$

$\bar{C}$ontact-Contact Transformation:

$$ {}^{\bar{C}_i}\Phi_{C_i} = \begin{pmatrix} \cos{}^{\bar{C}_i}\theta_{C_i} & -\sin{}^{\bar{C}_i}\theta_{C_i} & 0 & {}^{\bar{C}_i}d_{C_ix} \\ \sin{}^{\bar{C}_i}\theta_{C_i} & \cos{}^{\bar{C}_i}\theta_{C_i} & 0 & {}^{\bar{C}_i}d_{C_iy} \\ 0 & 0 & 1 & {}^{\bar{C}_i}d_{C_iz} \\ 0 & 0 & 0 & 1 \end{pmatrix} $$

Table 2.6: Transformation Matrix Time-Derivatives

$\bar{B}$ody-Body:

$$ {}^{\bar{B}}\dot{\Phi}_B = \begin{pmatrix} 0 & -{}^{\bar{B}}\omega_B & 0 & {}^{\bar{B}}v_{Bx} \\ {}^{\bar{B}}\omega_B & 0 & 0 & {}^{\bar{B}}v_{By} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} $$

Hip-Steering:

$$ {}^{Hi}\dot{\Phi}_{Si} = \begin{pmatrix} -{}^{Hi}\omega_{Si}\sin{}^{Hi}\theta_{Si} & -{}^{Hi}\omega_{Si}\cos{}^{Hi}\theta_{Si} & 0 & 0 \\ {}^{Hi}\omega_{Si}\cos{}^{Hi}\theta_{Si} & -{}^{Hi}\omega_{Si}\sin{}^{Hi}\theta_{Si} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} $$

$\bar{C}$ontact-Contact:

$$ {}^{\bar{C}i}\dot{\Phi}_{Ci} = \begin{pmatrix} 0 & -{}^{\bar{C}i}\omega_{Ci} & 0 & {}^{\bar{C}i}v_{Cix} \\ {}^{\bar{C}i}\omega_{Ci} & 0 & 0 & {}^{\bar{C}i}v_{Ciy} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} $$

Table 2.7: Transformation Matrix Second Time-Derivatives

$\bar{B}$ody-Body:

$$ {}^{\bar{B}}\ddot{\Phi}_B = \begin{pmatrix} -{}^{\bar{B}}\omega_B^2 & -{}^{\bar{B}}\alpha_B & 0 & {}^{\bar{B}}a_{Bx} \\ {}^{\bar{B}}\alpha_B & -{}^{\bar{B}}\omega_B^2 & 0 & {}^{\bar{B}}a_{By} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} $$

Hip-Steering:

$$ {}^{H_i}\ddot{\Phi}_{S_i} = \begin{pmatrix} -{}^{H_i}\alpha_{S_i}\sin{}^{H_i}\theta_{S_i} - {}^{H_i}\omega_{S_i}^2\cos{}^{H_i}\theta_{S_i} & -{}^{H_i}\alpha_{S_i}\cos{}^{H_i}\theta_{S_i} + {}^{H_i}\omega_{S_i}^2\sin{}^{H_i}\theta_{S_i} & 0 & 0 \\ {}^{H_i}\alpha_{S_i}\cos{}^{H_i}\theta_{S_i} - {}^{H_i}\omega_{S_i}^2\sin{}^{H_i}\theta_{S_i} & -{}^{H_i}\alpha_{S_i}\sin{}^{H_i}\theta_{S_i} - {}^{H_i}\omega_{S_i}^2\cos{}^{H_i}\theta_{S_i} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} $$

$\bar{C}$ontact-Contact:

$$ {}^{\bar{C}_i}\ddot{\Phi}_{C_i} = \begin{pmatrix} -{}^{\bar{C}_i}\omega_{C_i}^2 & -{}^{\bar{C}_i}\alpha_{C_i} & 0 & {}^{\bar{C}_i}a_{C_ix} \\ {}^{\bar{C}_i}\alpha_{C_i} & -{}^{\bar{C}_i}\omega_{C_i}^2 & 0 & {}^{\bar{C}_i}a_{C_iy} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} $$

Table 2.8: Matrix Coordinate Transformation Algebra Axioms

Identity:	${}^{A}\Pi_{B} = \mathbf{I}$	for B=A or B=$\bar{A}$
Cascade:	${}^{A}\Pi_{B} = {}^{A}\Pi_{X}\,{}^{X}\Pi_{B}$	
Inversion:	${}^{A}\Pi_{B} = {}^{B}\Pi_{A}^{-1}$	
Zero-Velocity:	${}^{A}\dot{\Pi}_{B} = \mathbf{0}$	for B=A or Π=**T**
Velocity:	${}^{A}\dot{\Pi}_{B} = {}^{A}\dot{\Pi}_{X}\,{}^{X}\Pi_{B} + {}^{A}\Pi_{X}\,{}^{X}\dot{\Pi}_{B}$	
Zero-Acceleration:	${}^{A}\ddot{\Pi}_{B} = \mathbf{0}$	for B=A or Π=**T**
Acceleration:	${}^{A}\ddot{\Pi}_{B} = {}^{A}\ddot{\Pi}_{X}\,{}^{X}\Pi_{B} + 2\,{}^{A}\dot{\Pi}_{X}\,{}^{X}\dot{\Pi}_{B} + {}^{A}\Pi_{X}\,{}^{X}\ddot{\Pi}_{B}$	

The *identity axiom* is self-evident since neither rotations nor translations are required to transform a coordinate system to itself or to its instantaneously coincident coordinate system. The *cascade axiom* specifies the order in which transformation matrices are multiplied: the coordinate transformation matrix from the reference system to the destination is the cascade of two coordinate transformation matrices, the first is from the reference system to an intermediate coordinate system, and the second is from the intermediate coordinate system to the destination. The *inversion axiom* states that the coordinate transformation matrix from a reference coordinate system to a destination coordination system is the inverse of the coordinate transformation matrix from the destination coordinate system to the reference coordinate system.

Just as the multiplication of transformation matrices is specified by the cascade axiom, time-differentiation of transformation matrices is specified by the four velocity and acceleration axioms. Specifically, we cannot differentiate both sides of a matrix transformation equation. For example, if we were to differentiate both sides of the equation ${}^{A}\Pi_{\bar{A}} = \mathbf{I}$, we would obtain the incorrect result that ${}^{A}\dot{\Pi}_{\bar{A}} = \mathbf{0}$ because the velocities between a coordinate system and its instantaneously coincident counterpart are (in general) non-zero. The *zero-*

velocity axiom states that the relative velocities between a coordinate system A and itself (B=A) or another coordinate system assigned to the same link (Π=**T**) are zero. This is because two coordinate systems assigned to the same link are stationary relative to the link and each other. Similarly, the *zero-acceleration* axiom states that the relative accelerations between a coordinate system A and itself (B=A) or another coordinate system assigned to the same link (Π=**T**) are zero. The *velocity* axiom specifies how the time-derivative of a transformation matrix may be expressed in terms of the two cascaded transformation matrices and their time-derivatives. Finally, the *acceleration* axiom specifies how the second time-derivative of a transformation matrix may be expressed in terms of the two cascaded transformation matrices and their first and second time-derivatives.

The matrix coordinate transformation axioms in Table 2.8 lead to the *corollaries* listed in Table 2.9 which we apply to the kinematic modeling of WMRs. We develop the *instantaneous coincidence corollary* by applying the identity and cascade axioms. The instantaneous coincidence corollary simplifies transformation matrix expressions by eliminating the instantaneously coincident coordinate systems. The *cascade position corollary* calculates the transformation matrix from a reference coordinate system to a destination coordinate system which may be kinematically separated from the reference system by a number of cascaded intermediate coordinate systems. The cascade position corollary, which is derived by repeated applications of the cascade axiom, is the foundation of position kinematics (in Section 2.3.7). The *cascade velocity corollary* is derived by repeated applications of the velocity axiom and the cascade axiom. The *cascade acceleration corollary* is derived by repeated applications of the cascade, velocity and acceleration axioms. In Sections 2.3.8 and 2.3.9, we apply the cascade velocity and the cascade acceleration corollaries to relate the linear and angular velocities and accelerations between coordinate systems. Throughout Section 2.3.8, we apply the axioms and corollaries of the matrix coordinate transformation algebra to derive the wheel Jacobian matrix.

Table 2.9: Matrix Coordinate Transformation Algebra Corollaries

Instantaneous Coincidence:	${}^{\bar{A}}\Pi_B = {}^{A}\Pi_{\bar{B}} = {}^{\bar{A}}\Pi_{\bar{B}} = {}^{A}\Pi_B$
Cascade Position:	${}^{A}\Pi_Z = {}^{A}\Pi_B \, {}^{B}\Pi_C \, {}^{C}\Pi_D \ldots {}^{Y}\Pi_Z$
Cascade Velocity:	${}^{A}\dot{\Pi}_Z = {}^{A}\dot{\Pi}_B \, {}^{B}\Pi_Z + {}^{A}\Pi_B \, {}^{B}\dot{\Pi}_C \, {}^{C}\Pi_Z + \ldots + {}^{A}\Pi_Y \, {}^{Y}\dot{\Pi}_Z$
Cascade Acceleration:	${}^{A}\ddot{\Pi}_Z = {}^{A}\ddot{\Pi}_B \, {}^{B}\Pi_Z + {}^{A}\Pi_B \, {}^{B}\ddot{\Pi}_C \, {}^{C}\Pi_Z + \ldots + {}^{A}\Pi_Y \, {}^{Y}\ddot{\Pi}_Z$
	$+ 2 \, {}^{A}\dot{\Pi}_B \left[{}^{B}\dot{\Pi}_C \, {}^{C}\Pi_Z + {}^{B}\Pi_C \, {}^{C}\dot{\Pi}_D \, {}^{D}\Pi_Z + \ldots + {}^{B}\Pi_Y \, {}^{Y}\dot{\Pi}_Z \right]$
	$+ 2 \, {}^{A}\Pi_B \, {}^{B}\dot{\Pi}_C \left[{}^{C}\dot{\Pi}_D \, {}^{D}\Pi_Z + \ldots + {}^{C}\Pi_Y \, {}^{Y}\dot{\Pi}_Z \right]$
	$+ \ldots + 2 \, {}^{A}\Pi_X \, {}^{X}\dot{\Pi}_Y \, {}^{Y}\dot{\Pi}_Z$

2.3.6 WMR Characteristic Kinematic Parameters

We obtain the characteristic kinematic parameters from measurements or design specifications. Specifically, we require the three translational displacements (e.g., ${}^{A}d_{Bx}$, ${}^{A}d_{By}$, ${}^{A}d_{Bz}$) and the angle (e.g., ${}^{A}\theta_B$) between adjacent coordinates systems (e.g., A and B) to compute the following three transformation matrices according to Tables 2.4 and 2.5 for each steered wheel i: (1) Body-Hip: ${}^{B}\mathbf{T}_{H_i}$, (2) Hip-Steering: ${}^{H_i}\Phi_{S_i}$, (3) Steering-Contact: ${}^{S_i}\mathbf{T}_{C_i}$. For each non-steered wheel i, we compute the Body-Contact transformation matrix ${}^{B}\mathbf{T}_{C_i}$. We also require the wheel radius r_{wi}, and the roller radius r_{ri} and roller angle η_i (for omnidirectional wheels) for each wheel i=1,...,N.

2.3.7 WMR Position Transformation Graph

We apply the transformation matrices and the matrix coordinate transformation algebra (in Section 2.3.5) to compute position kinematics. We compute the position of the point ${}^{A}\mathbf{r}$ relative to the A coordinate system from the position of the point ${}^{Z}\mathbf{r}$ relative to the Z coordinate system by applying the transformation matrix in (2.3.1):

$$^{A}r = {}^{A}\Pi_{Z}\, {}^{Z}r\,. \tag{2.3.9}$$

When the transformation matrix ${}^{A}\Pi_{Z}$ is not known directly, we apply the cascade position corollary to calculate ${}^{A}\Pi_{Z}$ from known transformation matrices:

$$^{A}\Pi_{Z} = {}^{A}\Pi_{B}\, {}^{B}\Pi_{C}\, {}^{C}\Pi_{D} \ldots {}^{Y}\Pi_{Z}\,. \tag{2.3.10}$$

We apply position transformation graphs to determine whether there is a complete set of known transformation matrices which can be cascaded to create ${}^{A}\Pi_{Z}$. In Figure 2.6, we display a position transformation graph of a WMR.

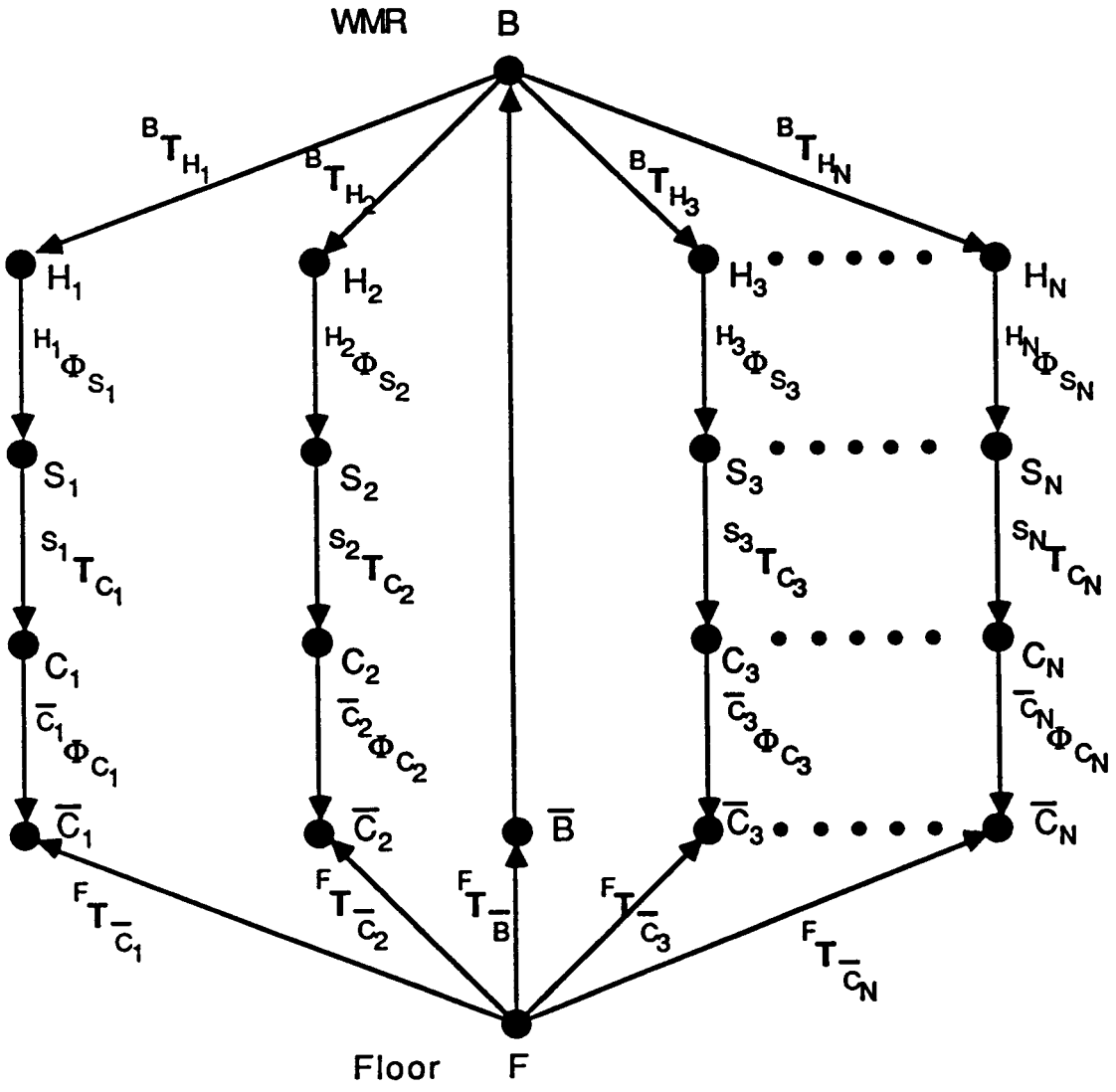


Figure 2.6: Position Transformation Graph of a WMR

The origin of each coordinate system is represented by a dot, and transformations between coordinate systems are depicted by directed arrows. The transformation in the direction opposing an arrow is calculated by applying the inversion axiom. Finding a cascade of transformations to calculate a desired transformation matrix (e.g. ${}^{F}\Pi_{S_1}$) is thus equivalent to finding a path from the reference coordinate system of the desired transformation (F) to the destination coordinate system (S_1). The matrices to be cascaded are listed by traversing the path in order. Each transformation in the path which is traversed from the tail to the head of an arrow is listed as the matrix itself, while transformations traversed from the head to the tail are listed as the inverse of the matrix.

For example, a point located at position **r** relative to the steering coordinate system S_1 in Figure 2.3.7 is transformed to its position relative to the floor coordinate system F according to:

$$ {}^{F}\mathbf{r} = {}^{F}\Pi_{S_1}\,{}^{S_1}\mathbf{r}\,, \tag{2.3.11} $$

where

$$ {}^{F}\Pi_{S_1} = {}^{F}\mathbf{T}_{\bar{B}}\,{}^{\bar{B}}\Phi_{B}\,{}^{B}\mathbf{T}_{H_1}\,{}^{H_1}\Phi_{S_1}\,. \tag{2.3.12} $$

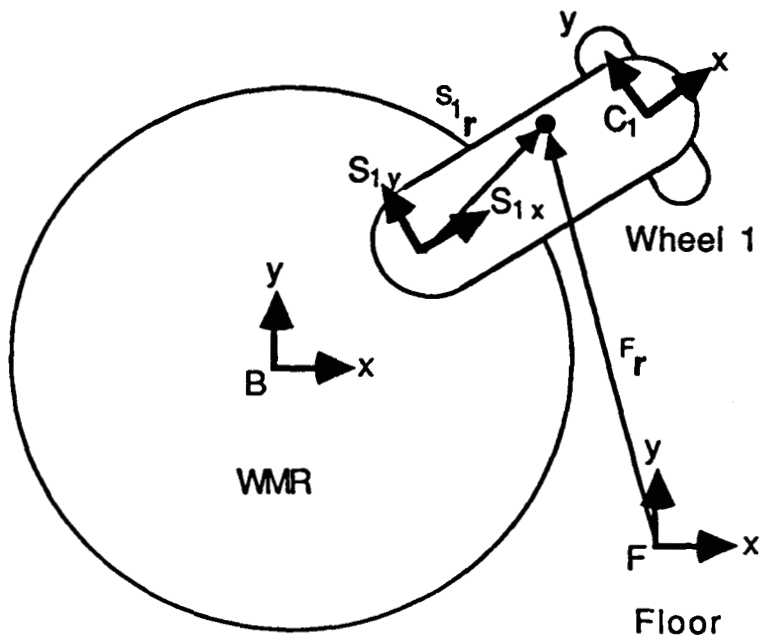


Figure 2.7: Point Fixed with the Steering Link

In this example, the reference coordinate system is the floor coordinate system F and the destination coordinate system is the steering coordinate system S_1. There are multiple paths between any two coordinate systems in Figure 2.6 because WMRs are closed-chain structures. In practice, the number of feasible paths is reduced because some of the transformation matrices are unknown. For example, we may seek to compute the desired transformation matrix in (2.3.11) as:

$$^{F}\Pi_{S_1} = {}^{F}\mathbf{T}_{\bar{C}_1}\, {}^{\bar{C}_1}\Phi_{C_1}\, {}^{C_1}\mathbf{T}_{S_1} \tag{2.3.13}$$

but the transformation matrix $^{F}\mathbf{T}_{\bar{C}_1}$ from the floor to the wheel contact point is typically unknown.

2.3.8 The Wheel Jacobian

We relate the velocities of the WMR by applying the matrix coordinate transformation algebra axioms and the cascade velocity corollary. In this section, we apply our methodology to compute the velocities of the WMR body relative to the instantaneously coincident body coordinate system from the velocities of a wheel. The wheel velocities (as described in Appendix 3) are the steering velocity ω_S, the wheel velocity about its axle ω_{wx}, the rotational slip velocity ω_{wz}, the roller velocities ω_r (for omnidirectional wheels) and the rotational velocity ω_{wy} (for ball wheels). We thus introduce the *wheel Jacobian matrix* to compute the robot velocity vector from the wheel velocity vector.

We differentiate the point transformation in (2.3.9) with respect to time to compute the velocity of the point **r** fixed in the A coordinate system:

$$^{A}\dot{\mathbf{r}} = {}^{A}\dot{\Pi}_{Z}\, {}^{Z}\mathbf{r}\,. \tag{2.3.14}$$

When the matrix $^{A}\dot{\Pi}_{Z}$ is not known directly, we apply the cascade velocity corollary to compute $^{A}\dot{\Pi}_{Z}$ from known transformation matrices and transformation matrix time-derivatives according to:

$$^{A}\dot{\Pi}_{Z} = {}^{A}\dot{\Pi}_{B}\, {}^{B}\Pi_{Z} + {}^{A}\Pi_{B}\, {}^{B}\dot{\Pi}_{C}\, {}^{C}\Pi_{Z} + \ldots + {}^{A}\Pi_{Y}\, {}^{Y}\dot{\Pi}_{Z}\,. \tag{2.3.15}$$

For example, (2.3.11) relates the position **r** of a point in the steering coordinate system S_1 to its position in the floor coordinate system F. We compute the velocity of the point **r** relative to the floor coordinate system by differentiating (2.3.11):

$$ {}^{F}\dot{\mathbf{r}} = {}^{F}\dot{\Pi}_{S_1}\, {}^{S_1}\mathbf{r} \,. \qquad (2.3.16) $$

Since the vector ${}^{S_1}\mathbf{r}$ is constant, its time-derivative is zero. We apply the cascade velocity corollary and the WMR transformation graph (in Figure 2.6) to obtain an expression for the unknown transformation matrix derivative in (2.3.16):

$$ {}^{F}\dot{\Pi}_{S_1} = {}^{F}\dot{\mathbf{T}}_{\bar{B}}\, {}^{\bar{B}}\Pi_{S_1} + {}^{F}\mathbf{T}_{\bar{B}}\, {}^{\bar{B}}\dot{\Phi}_{B}\, {}^{B}\Pi_{S_1} + {}^{F}\Pi_{B}\, {}^{B}\dot{\mathbf{T}}_{H_1}\, {}^{H_1}\Phi_{S_1} + {}^{F}\Pi_{H_1}\, {}^{H_1}\dot{\Phi}_{S_1} \,. \qquad (2.3.17) $$

We simplify (2.3.17) to require only known transformation matrices and transformation matrix derivatives:

$$
\begin{aligned}
{}^{F}\dot{\Pi}_{S_1} &= {}^{F}\mathbf{T}_{\bar{B}}\, {}^{\bar{B}}\dot{\Phi}_{B}\, {}^{B}\Pi_{S_1} + {}^{F}\Pi_{H_1}\, {}^{H_1}\dot{\Phi}_{S_1} && \text{Zero-Velocity Axiom} \\
&= {}^{F}\mathbf{T}_{\bar{B}}\, {}^{\bar{B}}\dot{\Phi}_{B}\, {}^{B}\mathbf{T}_{H_1}\, {}^{H_1}\Phi_{S_1} + {}^{F}\mathbf{T}_{\bar{B}}\, {}^{\bar{B}}\Phi_{B}\, {}^{B}\mathbf{T}_{H_1}\, {}^{H_1}\dot{\Phi}_{S_1} && \text{Cascade Corollary} \\
&= {}^{F}\mathbf{T}_{\bar{B}}\, {}^{\bar{B}}\dot{\Phi}_{B}\, {}^{B}\mathbf{T}_{H_1}\, {}^{H_1}\Phi_{S_1} + {}^{F}\mathbf{T}_{\bar{B}}\, {}^{B}\mathbf{T}_{H_1}\, {}^{H_1}\dot{\Phi}_{S_1} && \text{Identity Axiom} \\
&= {}^{F}\Pi_{B}\, {}^{\bar{B}}\dot{\Phi}_{B}\, {}^{B}\mathbf{T}_{H_1}\, {}^{H_1}\Phi_{S_1} + {}^{F}\Pi_{B}\, {}^{B}\mathbf{T}_{H_1}\, {}^{H_1}\dot{\Phi}_{S_1} && \text{Instantaneous Coincidence}
\end{aligned}
\qquad (2.3.18)
$$

In (2.3.18), the robot velocity (in ${}^{\bar{B}}\dot{\Phi}_{B}$) is computed in the sensed forward solution (in Section 2.4.3), the steering position (in ${}^{H_1}\Phi_{S_1}$) and velocity (in ${}^{H_1}\dot{\Phi}_{S_1}$) are sensed, the WMR body position (in ${}^{F}\Pi_{B}$) is computed by dead reckoning (in Appendix 9), and the body-to-hip transformation (${}^{B}\mathbf{T}_{H_1}$) is specified by the mechanical design. The right-hand side of (2.3.18) is thus known. We substitute (2.3.18) into (2.3.16) to compute the velocity of the point **r** relative to the floor coordinate system.

We formulate analogously the equations-of-motion to model the velocities of the robot in terms of the velocities of a wheel. We begin our development by applying the cascade velocity corollary to write the matrix equation in (2.3.19) with the unknown dependent variables (i.e., WMR body

velocities, ${}^{\bar{B}}\dot{\Phi}_B$) on the left-hand side, and the independent variables (i.e., the wheel i velocities, ${}^{H_i}\dot{\Phi}_{S_i}$ and ${}^{\bar{C}_i}\dot{\Phi}_{C_i}$) on the right-hand side:

$${}^{\bar{B}}\dot{\Pi}_B = {}^{F}T_{\bar{B}}^{-1}\,{}^{F}T_{\bar{C}_i}\,{}^{\bar{C}_i}\dot{\Phi}_{C_i}\,{}^{S_i}T_{C_i}^{-1}\,{}^{H_i}\Phi_{S_i}^{-1}\,{}^{B}T_{H_i}^{-1} + {}^{F}T_{\bar{B}}^{-1}\,{}^{F}T_{\bar{C}_i}\,{}^{\bar{C}_i}\Phi_{C_i}\,{}^{S_i}T_{C_i}^{-1}\,{}^{H_i}\dot{\Phi}_{S_i}^{-1}\,{}^{B}T_{H_i}^{-1}. \qquad (2.3.19)$$

The position transformation graph in Figure 2.6 is utilized to determine the order in which to cascade the transformation matrices; the inversion axiom is applied when an arrow in the transformation graph is traversed from head-to-tail and the zero-velocity axiom is applied to eliminate the matrices which multiply the derivatives of constant T matrices. Since the position of the wheel contact point relative to the floor is typically unknown, we apply the cascade position corollary to write an alternative expression for the floor-contact transformation matrix:

$${}^{F}T_{\bar{C}_i} = {}^{F}T_{\bar{B}}\,{}^{\bar{B}}\Phi_B\,{}^{B}T_{H_i}\,{}^{H_i}\Phi_{S_i}\,{}^{S_i}T_{C_i}\,{}^{\bar{C}_i}\Phi_{C_i}^{-1}. \qquad (2.3.20)$$

We substitute (2.3.20) into (2.3.19) to obtain:

$${}^{\bar{B}}\dot{\Pi}_B = {}^{\bar{B}}\Phi_B\,{}^{B}T_{H_i}\,{}^{H_i}\Phi_{S_i}\,{}^{S_i}T_{C_i}\,{}^{\bar{C}_i}\Phi_{C_i}^{-1}\,{}^{\bar{C}_i}\dot{\Phi}_{C_i}\,{}^{S_i}T_{C_i}^{-1}\,{}^{H_i}\Phi_{S_i}^{-1}\,{}^{B}T_{H_i}^{-1} + {}^{\bar{B}}\Phi_B\,{}^{B}T_{H_i}\,{}^{H_i}\Phi_{S_i}\,{}^{H_i}\dot{\Phi}_{S_i}^{-1}\,{}^{B}T_{H_i}^{-1}. \qquad (2.3.21)$$

and apply the identity axiom to simplify (2.3.21):

$${}^{\bar{B}}\dot{\Pi}_B = {}^{B}T_{H_i}\,{}^{H_i}\Phi_{S_i}\,{}^{S_i}T_{C_i}\,{}^{\bar{C}_i}\dot{\Phi}_{C_i}\,{}^{S_i}T_{C_i}^{-1}\,{}^{H_i}\Phi_{S_i}^{-1}\,{}^{B}T_{H_i}^{-1} + {}^{B}T_{H_i}\,{}^{H_i}\Phi_{S_i}\,{}^{H_i}\dot{\Phi}_{S_i}^{-1}\,{}^{B}T_{H_i}^{-1} \qquad (2.3.22)$$

We next apply Tables 2.4, 2.5, and 2.6 to write the transformation matrices and the transformation matrix derivatives and multiply the result to obtain:

$$\begin{pmatrix} 0 & -{}^{\bar{B}}\omega_B & 0 & {}^{\bar{B}}v_{Bx} \\ {}^{\bar{B}}\omega_B & 0 & 0 & {}^{\bar{B}}v_{By} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} =$$

$$\begin{pmatrix} 0 & -{}^{\bar{C}_i}\omega_{C_i} & 0 & {}^{\bar{C}_i}\omega_{C_i}\,{}^{B}d_{C_iy} + {}^{\bar{C}_i}v_{C_ix}\cos{}^{B}\theta_{C_i} - {}^{\bar{C}_i}v_{C_iy}\sin{}^{B}\theta_{C_i} \\ {}^{\bar{C}_i}\omega_{C_i} & 0 & 0 & -{}^{\bar{C}_i}\omega_{C_i}\,{}^{B}d_{C_ix} + {}^{\bar{C}_i}v_{C_ix}\sin{}^{B}\theta_{C_i} + {}^{\bar{C}_i}v_{C_iy}\cos{}^{B}\theta_{C_i} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$+\begin{pmatrix} 0 & {}^{H_i}\omega_{S_i} & 0 & -{}^{H_i}\omega_{S_i}\,{}^{B}d_{H_iy} \\ -{}^{H_i}\omega_{S_i} & 0 & 0 & {}^{H_i}\omega_{S_i}\,{}^{B}d_{H_ix} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}. \qquad (2.3.23)$$

To simplify the notation in (2.3.23), we have made the following substitutions:

$$\begin{aligned} {}^{B}\theta_{H_i} + {}^{H_i}\theta_{S_i} + {}^{S_i}\theta_{C_i} &= {}^{B}\theta_{C_i} \\ {}^{S_i}d_{C_ix}\cos({}^{B}\theta_{H_i} + {}^{H_i}\theta_{S_i}) - {}^{S_i}d_{C_iy}\sin({}^{B}\theta_{H_i} + {}^{H_i}\theta_{S_i}) + {}^{B}d_{H_ix} &= {}^{B}d_{C_ix} \qquad (2.3.24) \\ {}^{S_i}d_{C_ix}\sin({}^{B}\theta_{H_i} + {}^{H_i}\theta_{S_i}) + {}^{S_i}d_{C_iy}\cos({}^{B}\theta_{H_i} + {}^{H_i}\theta_{S_i}) + {}^{B}d_{H_iy} &= {}^{B}d_{C_iy} \end{aligned}$$

Upon equating the elements in (2.3.23), we obtain the body velocities:

$${}^{\bar{B}}\dot{\mathbf{p}}_B = \begin{pmatrix} {}^{\bar{B}}v_{Bx} \\ {}^{\bar{B}}v_{By} \\ {}^{\bar{B}}\omega_B \end{pmatrix} = \begin{pmatrix} \cos{}^{B}\theta_{C_i} & -\sin{}^{B}\theta_{C_i} & {}^{B}d_{C_iy} & -{}^{B}d_{H_iy} \\ \sin{}^{B}\theta_{C_i} & \cos{}^{B}\theta_{C_i} & -{}^{B}d_{C_ix} & {}^{B}d_{H_ix} \\ 0 & 0 & 1 & -1 \end{pmatrix} \begin{pmatrix} {}^{\bar{C}_i}v_{C_ix} \\ {}^{\bar{C}_i}v_{C_iy} \\ {}^{\bar{C}_i}\omega_{C_i} \\ {}^{H_i}\omega_{S_i} \end{pmatrix} = \hat{\mathbf{J}}_i\,\dot{\hat{\mathbf{q}}}_i \qquad (2.3.25)$$

where i=1,...,N is the wheel index, ${}^{\bar{B}}\dot{\mathbf{p}}_B$ is the vector of WMR body velocities in the $\bar{\text{b}}$ody frame, $\hat{\mathbf{J}}_i$ is the pseudo-Jacobian matrix for wheel i, and $\dot{\hat{\mathbf{q}}}_i$ is the pseudo-velocity vector for wheel i. We define the number of wheel variables of wheel i to be w_i. The physical velocity vector $\dot{\mathbf{q}}_i$ of typical wheels does not contain the four component velocities in (2.3.25). Typical wheels posses fewer than four wheel variables and thus fewer than four elements in the velocity vector. Furthermore, since all physical wheel motions are rotations about physical wheel axes, the wheel velocity vector contains the angular velocities of the wheels rather than the linear velocities of the point-of-contact along the

surface-of-travel. We relate the (4×1) pseudo-velocity vector to the $(w_i \times 1)$ physical velocity vector by the $(4 \times w_i)$ wheel matrix $\mathbf{W}_i$:

$$\dot{\hat{\mathbf{q}}}_i = \mathbf{W}_i \dot{\mathbf{q}}_i . \tag{2.3.26}$$

We substitute (2.3.26) into (2.3.25) to compute the body velocities from the wheel velocity vector:

$$^{\bar{B}}\dot{\mathbf{p}}_B = \hat{\mathbf{J}}_i \mathbf{W}_i \dot{\mathbf{q}}_i = \mathbf{J}_i \dot{\mathbf{q}}_i . \tag{2.3.27}$$

The product $\mathbf{J}_i = (\hat{\mathbf{J}}_i \mathbf{W}_i)$ is the $(3 \times w_i)$ wheel Jacobian matrix of wheel i. The rank of the wheel Jacobian matrix indicates the number of DOFs of the wheel. A wheel having fewer DOFs than wheel variables is *redundant*. The Jacobian matrix of a redundant wheel has dependent columns. We thus formulate the following *rank test* [Strang86] as a computational method to determine whether a wheel is non-redundant:

Non-Redundant Wheel Criterion

$$\text{Rank}\,(\mathbf{J}_i) = w_i \tag{2.3.28}$$

Three wheel types have been utilized in the WMR designs surveyed in Appendix 2: non-steered conventional wheels, steered conventional wheels and omnidirectional wheels. The wheel Jacobian matrices for these wheels and the ball wheel are detailed in Appendix 3. We utilize (2.3.27) in Section 2.4 to develop the forward and inverse WMR velocity solutions.

2.3.9 Wheel Acceleration Computations

We apply the cascade acceleration corollary (in Table 2.9) to compute the WMR accelerations. Since the development parallels that of the velocity kinematics in Section 2.3.8, we omit the computational details and concentrate on interpreting the results. As noted in Section 2.3.5, we cannot formulate the acceleration equations-of-motion by differentiating the results of Section 2.3.8

because differentiation of both sides of a transformation matrix equation is not an allowable operation in our matrix coordinate transformation algebra. This is in contrast to the acceleration kinematics of mechanisms containing only lower-pairs (e.g., stationary manipulators) which are formulated by differentiating velocity kinematics.

The acceleration of the point **r** fixed relative to the moving coordinate system Z is transformed to the A coordinate system according to:

$$^{A}\ddot{\mathbf{r}} = {}^{A}\ddot{\Pi}_{Z}\,{}^{Z}\mathbf{r}\,. \tag{2.3.29}$$

We apply the cascade acceleration corollary to compute the second time-derivative of the transformation matrix ${}^{A}\ddot{\Pi}_{Z}$. By applying the cascade acceleration corollary, the component accelerations of the WMR body (${}^{\bar{B}}a_{Bx}$, ${}^{\bar{B}}a_{By}$ and ${}^{\bar{B}}\alpha_{B}$) are related to the wheel accelerations (${}^{Hi}\alpha_{Si}$, ${}^{\bar{C}i}a_{Cix}$, ${}^{\bar{C}i}a_{Ciy}$, and ${}^{\bar{C}i}\alpha_{Ci}$) as the cascade velocity corollary (in Section 2.3.8) relates the WMR body velocities to the wheel velocities. In the notation of (2.3.24), the body accelerations are:

$$\begin{pmatrix} {}^{\bar{B}}a_{Bx} \\ {}^{\bar{B}}a_{By} \\ {}^{\bar{B}}\alpha_{B} \end{pmatrix} = \begin{pmatrix} \cos{}^{B}\theta_{Ci} & -\sin{}^{B}\theta_{Ci} & {}^{B}d_{Ciy} & -{}^{B}d_{Hiy} \\ \sin{}^{B}\theta_{Ci} & \cos{}^{B}\theta_{Ci} & -{}^{B}d_{Cix} & {}^{B}d_{Hix} \\ 0 & 0 & 1 & -1 \end{pmatrix} \begin{pmatrix} {}^{\bar{C}i}a_{Cix} \\ {}^{\bar{C}i}a_{Ciy} \\ {}^{\bar{C}i}\alpha_{Ci} \\ {}^{Hi}\alpha_{Si} \end{pmatrix}$$

$$+ \begin{pmatrix} {}^{B}d_{Cix} & {}^{B}d_{Hix} & {}^{B}d_{Hix} \\ {}^{B}d_{Ciy} & {}^{B}d_{Hiy} & {}^{B}d_{Hiy} \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} {}^{\bar{C}i}\omega_{Q}^{2} \\ -2\,{}^{\bar{C}i}\omega_{Ci}\,{}^{Hi}\omega_{Si} \\ {}^{Hi}\omega_{Si}^{2} \end{pmatrix}. \tag{2.3.30}$$

The robot accelerations in (2.3.30) are composed of three components: the *self*-accelerations (${}^{\bar{C}i}a_{Cix}$, ${}^{\bar{C}i}a_{Ciy}$, ${}^{\bar{C}i}\alpha_{Ci}$ and ${}^{Hi}\alpha_{Si}$); the *centripetal* accelerations (${}^{\bar{C}i}\omega_{Ci}^{2}$ and ${}^{Hi}\omega_{Si}^{2}$) having squared velocities; and the *Coriolis* accelerations (${}^{\bar{C}i}\omega_{Ci}$ ${}^{Hi}\omega_{Si}$) having products of different velocities.

2.3.10 WMR-to-Floor Motion Transformations

We equate the components in the matrix equation in (2.3.15) to compute the translational ${}^{A}v_{Zx}$, and ${}^{A}v_{Zy}$ and rotational ${}^{A}\omega_{Z}$ velocities[3] of coordinate system Z relative to coordinate system A. We apply this methodology to the practical problem of transforming velocities of the WMR body from body coordinates $\bar{B}$ to floor coordinates F. We assume that the floor-body transformation matrix ${}^{F}T_{B}$ (i.e., the position and orientation of the WMR body relative to the floor) and the matrix ${}^{\bar{B}}\dot{\Phi}_{B}$ (i.e., the velocities of the WMR body relative to its current position and orientation) are known. The velocities to be computed (i.e, the velocities of the WMR body relative to the floor) are the components of the matrix ${}^{F}\dot{\Pi}_{B}$. We apply the cascade velocity corollary (in Section 2.3.5) and the WMR position transformation graph (in Section 2.3.7) to write the matrix equation

$$ {}^{F}\dot{\Pi}_{B} = {}^{F}\dot{T}_{\bar{B}}\,{}^{\bar{B}}\Phi_{B} + {}^{F}T_{\bar{B}}\,{}^{\bar{B}}\dot{\Phi}_{B} \qquad (2.3.31) $$

in terms of known matrices. To simplify (2.3.31), we apply the zero-velocity axiom and the instantaneous coincidence corollary:

$$ {}^{F}\dot{\Pi}_{B} = {}^{F}T_{B}\,{}^{\bar{B}}\dot{\Phi}_{B}. \qquad (2.3.32) $$

We expand each matrix in (2.3.32) into scalar components: the matrix derivative ${}^{F}\dot{\Pi}_{B}$ according to (2.3.7), the transformation matrix ${}^{F}T_{B}$ according to (2.3.5), and the transformation matrix derivative ${}^{\bar{B}}\dot{\Phi}_{B}$ according to Table 2.6. Upon multiplying, we obtain:

$$ \begin{pmatrix} -{}^{F}\omega_{B}\sin{}^{F}\theta_{B} & -{}^{F}\omega_{B}\cos{}^{F}\theta_{B} & 0 & {}^{F}v_{Bx} \\ {}^{F}\omega_{B}\cos{}^{F}\theta_{B} & -{}^{F}\omega_{B}\sin{}^{F}\theta_{B} & 0 & {}^{F}v_{By} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} = $$

---

[3] Because of our kinematic modeling coordinate system assignments (in Section 2.3.3), there are no translational velocities along the z-axis or angular velocities about the x and y-axes.

$$\begin{pmatrix} -{}^{\bar{B}}\omega_B \sin{}^F\theta_B & -{}^{\bar{B}}\omega_B \cos{}^F\theta_B & 0 & {}^{\bar{B}}v_{Bx}\cos{}^F\theta_B - {}^{\bar{B}}v_{By}\sin{}^F\theta_B \\ {}^{\bar{B}}\omega_B \cos{}^F\theta_B & -{}^{\bar{B}}\omega_B \sin{}^F\theta_B & 0 & {}^{\bar{B}}v_{Bx}\sin{}^F\theta_B + {}^{\bar{B}}v_{By}\cos{}^F\theta_B \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}. \qquad (2.3.33)$$

We obtain the angular velocity of the WMR body ${}^F\omega_B$ from elements (1,1) and (2,1) and read the translational velocities ${}^Fv_{Bx}$ and ${}^Fv_{By}$ directly from elements (1,4) and (2,4) of (2.3.33), respectively. We find that:

$${}^F\dot{\mathbf{p}}_B = \begin{pmatrix} {}^Fv_{Bx} \\ {}^Fv_{By} \\ {}^F\omega_B \end{pmatrix} = \begin{pmatrix} \cos{}^F\theta_B & -\sin{}^F\theta_B & 0 \\ \sin{}^F\theta_B & \cos{}^F\theta_B & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} {}^{\bar{B}}v_{Bx} \\ {}^{\bar{B}}v_{By} \\ {}^{\bar{B}}\omega_B \end{pmatrix} = \mathbf{M}\, {}^{\bar{B}}\dot{\mathbf{p}}_B. \qquad (2.3.34)$$

In (2.3.34), we observe that the angular velocity of the WMR body is equal in both coordinate systems; whereas the translational velocities in the floor coordinate system depend upon the WMR orientation. The matrix **M** is the (3 × 3) *motion matrix* which depends upon the WMR body position ${}^F\mathbf{p}_B$. We apply the motion matrix to dead reckoning in Appendix 9 and to servo-control in Chapter 4 for WMRs.

Transforming WMR body accelerations from $\bar{\text{b}}$ody coordinates to floor coordinates is analogous to transforming WMR body velocities. We find that the body accelerations are transformed from the $\bar{\text{b}}$ody to the floor coordinate system by the motion matrix **M** that transforms the velocities in (2.3.34):

$${}^F\ddot{\mathbf{p}}_B = \begin{pmatrix} {}^Fa_{Bx} \\ {}^Fa_{By} \\ {}^F\alpha_B \end{pmatrix} = \begin{pmatrix} \cos{}^F\theta_B & -\sin{}^F\theta_B & 0 \\ \sin{}^F\theta_B & \cos{}^F\theta_B & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} {}^{\bar{B}}a_{Bx} \\ {}^{\bar{B}}a_{By} \\ {}^{\bar{B}}\alpha_B \end{pmatrix} = \mathbf{M}\, {}^{\bar{B}}\ddot{\mathbf{p}}_B. \qquad (2.3.35)$$

2.3.11 Concluding Remarks

We have formulated the position, velocity and acceleration kinematics of a WMR. To accomplish this task, we have incorporated the special WMR characteristics discussed in Section 1.5; namely higher-pair joints, closed-chains and unactuated and unsensed joints. We have introduced the

instantaneously coincident coordinate system as a tool for modeling the higher-pair wheel-to-floor joint. The application of instantaneously coincident coordinate systems leads to the development of a matrix coordinate transformation algebra which is the foundation of all kinematic equations-of-motion in this chapter.

The *non-redundant wheel criterion* in (2.3.28) is a test on the wheel Jacobian matrix to determine whether a wheel has as many DOFs as wheel variables. We apply this criterion in Appendix 7 to reveal disadvantages of redundant wheels. Our wheel velocity and acceleration equations-of-motion in Sections 2.3.8 and 2.3.9 may be applied to the dynamic modeling, mechanical design, wheel slip detection and servo-control of a WMR. In Section 2.4, we combine the velocity equations-of-motion of all of the wheels on a WMR and compute the forward and inverse solutions which are also required for dynamic modeling, wheel slip detection, and servo-control applications.

2.4 Forward and Inverse Kinematic Solutions

2.4.1 Overview

Since WMRs are multiple closed-chains, the forward and inverse solutions are obtained by solving *simultaneously* the kinematic equations-of-motion of all of the wheels. In this section, we combine the kinematic equations-of-motion of all of the wheels on a WMR to form the composite robot equation. We then proceed to solve the composite robot equation by addressing two classical kinematic modeling problems: the inverse solution (in Section 2.4.2) and the forward solution (in Section 2.4.3).

The *actuated inverse solution* (in Section 2.4.2) computes the actuated wheel velocities from the WMR body velocities. For WMR control, we solve only for the velocities of the actuated wheel variables. The solution for *all* of the wheel velocities is a special case which may be obtained by assuming that all of the wheel variables are actuated. The *sensed forward solution* (in Section 2.4.3) computes the WMR body velocities from the sensed wheel velocities and positions. Since a WMR consists of closed kinematic chains, we show in

Section 2.4.3 that it is not required to sense all of the wheel positions and velocities, and in practice, it is difficult to do so.

In Section 2.3.8, we developed the wheel Jacobian matrix $\mathbf{J}_i$ by applying velocity kinematics to compute the WMR body velocity vector $\dot{\mathbf{p}}$ from the wheel velocity vector $\dot{\mathbf{q}}_i$:

$$\dot{\mathbf{p}} = \mathbf{J}_i \dot{\mathbf{q}}_i \qquad \text{for } i = 1,...,N, \qquad (2.4.1)$$

where i is the wheel index, N is the total number of wheels, $\dot{\mathbf{p}}$ is the vector of WMR body velocities, $\mathbf{J}_i$ is the $(3 \times w_i)$ Jacobian matrix for wheel i, w_i is the number of variables for wheel i, and $\dot{\mathbf{q}}_i$ is the $(w_i \times 1)$ vector of wheel velocities. The 3N wheel equations in (2.4.1) must be solved simultaneously to characterize the WMR motion. We combine the wheel equations to form the *composite robot equation*:

$$\begin{pmatrix} \mathbf{I}_1 \\ \mathbf{I}_2 \\ \vdots \\ \mathbf{I}_N \end{pmatrix} \dot{\mathbf{p}} = \begin{pmatrix} \mathbf{J}_1 & 0 & \dots & 0 \\ 0 & \mathbf{J}_2 & . & \vdots \\ \vdots & . & . & 0 \\ 0 & \dots & 0 & \mathbf{J}_N \end{pmatrix} \begin{pmatrix} \dot{\mathbf{q}}_1 \\ \dot{\mathbf{q}}_2 \\ \vdots \\ \dot{\mathbf{q}}_N \end{pmatrix} \qquad (2.4.2)$$

or

$$\mathbf{A}_0 \dot{\mathbf{p}} = \mathbf{B}_0 \dot{\mathbf{q}} \qquad (2.4.3)$$

where $\mathbf{I}_i$ for i=1,...,N are (3×3) identity matrices, $\mathbf{A}_0$ is a $(3N \times 3)$ matrix, $\mathbf{B}_0$ is a $(3N \times w)$ block diagonal matrix, $w=w_1+w_2+...+w_N$ is the total number of wheel variables and $\dot{\mathbf{q}}$ is the composite wheel velocity vector. The composite robot equation in (2.4.3) is a set of linear algebraic equations of the form $\mathbf{Ax}=\mathbf{By}$ which is amenable to solution by classical techniques. In Appendix 4, we review the solutions of this system of linear algebraic equations. Having formulated the matrix equation in (2.4.3) to model the WMR motion, we proceed to investigate the solution for the wheel velocity vector $\dot{\mathbf{q}}$ in Section 2.4.2 and the WMR body velocity vector $\dot{\mathbf{p}}$ in Section 2.4.3.

2.4.2 Actuated Inverse Velocity Solution

We compute the actuated inverse velocity solution by solving for the actuated wheel velocities in (2.4.3). Because of the closed-chains in WMRs, we need not actuate all of the wheel variables. To separate the actuated and unactuated wheel variables, we partition the wheel equation in (2.4.1) into two components:

$$\dot{\mathbf{p}} = \mathbf{J}_{ia}\,\dot{\mathbf{q}}_{ia} + \mathbf{J}_{iu}\,\dot{\mathbf{q}}_{iu} \qquad \text{for } i = 1,\ldots,N\,. \qquad (2.4.4)$$

The "a" subscripts denote the actuated components and the "u" subscripts denote the unactuated components. We let a_i denote the number of actuated variables, and u_i denote the number of unactuated variables for wheel i (i.e., $a_i+u_i=w_i$). We let $a=a_1+a_2+\ldots+a_N$ be the total number of actuated wheel variables and $u=u_1+u_2+\ldots+u_N$ be the total number of unactuated wheel variables. We combine the partitioned wheel equations in (2.4.4) to rewrite the composite robot equation in (2.4.2) as

$$\begin{pmatrix} \mathbf{I}_1 \\ \mathbf{I}_2 \\ \vdots \\ \mathbf{I}_N \end{pmatrix} \dot{\mathbf{p}} = \begin{pmatrix} \mathbf{J}_{1a} & 0 & \ldots & 0 & \mathbf{J}_{1u} & 0 & \ldots & 0 \\ 0 & \mathbf{J}_{2a} & . & \vdots & 0 & \mathbf{J}_{2u} & . & \vdots \\ \vdots & . & . & 0 & \vdots & . & . & 0 \\ 0 & \ldots & 0 & \mathbf{J}_{Na} & 0 & \ldots & 0 & \mathbf{J}_{Nu} \end{pmatrix} \begin{pmatrix} \dot{\mathbf{q}}_{1a} \\ \dot{\mathbf{q}}_{2a} \\ \vdots \\ \dot{\mathbf{q}}_{Na} \\ \dot{\mathbf{q}}_{1u} \\ \dot{\mathbf{q}}_{2u} \\ \vdots \\ \dot{\mathbf{q}}_{Nu} \end{pmatrix} \qquad (2.4.5)$$

or

$$\mathbf{A}_0\,\dot{\mathbf{p}} = \mathbf{B}_{0p}\,\dot{\mathbf{q}}_p\,. \qquad (2.4.6)$$

The (3N × w) matrix $\mathbf{B}_{0p}$ and the (w × 1) vector $\dot{\mathbf{q}}_p$ are the *partitioned counterparts of the matrix* $\mathbf{B}_0$ and the vector $\dot{\mathbf{q}}$ in (2.4.2). The soluble motion criterion in Appendix 7 indicates the condition (i.e., rank $[\mathbf{B}_0]$ = w) under which the least-squares solution may be practically applied to compute the inverse solution. In the remainder of this section, we assume that the least-squares

solution (in Appendix 4) is applicable and that all of the *normal* matrices (e.g., $\mathbf{B}_{0p}^T \mathbf{B}_{0p}$) encountered in its application are computable. We apply the least-squares solution to compute the vector of wheel variables from the WMR body velocity vector:

$$\dot{\mathbf{q}}_p = (\mathbf{B}_{0p}^T \mathbf{B}_{0p})^{-1} \mathbf{B}_{0p}^T \mathbf{A}_0 \dot{\mathbf{p}}. \tag{2.4.7}$$

In Appendix 5, we compute the vector of *actuated* wheel velocities $\dot{\mathbf{q}}_a = [\dot{\mathbf{q}}_{1a}^T \ldots \dot{\mathbf{q}}_{Na}^T]^T$ in (2.4.5) as:

Actuated Inverse Velocity Solution

$$\dot{\mathbf{q}}_a = \begin{pmatrix} [\mathbf{J}_{1a}^T \Delta(\mathbf{J}_{1u}) \mathbf{J}_{1a}]^{-1} \mathbf{J}_{1a}^T \Delta(\mathbf{J}_{1u}) \\ [\mathbf{J}_{2a}^T \Delta(\mathbf{J}_{2u}) \mathbf{J}_{2a}]^{-1} \mathbf{J}_{2a}^T \Delta(\mathbf{J}_{2u}) \\ \vdots \\ [\mathbf{J}_{Na}^T \Delta(\mathbf{J}_{Nu}) \mathbf{J}_{Na}]^{-1} \mathbf{J}_{Na}^T \Delta(\mathbf{J}_{Nu}) \end{pmatrix} \dot{\mathbf{p}} = \mathbf{J}_a \dot{\mathbf{p}} \tag{2.4.8}$$

where the matrix function $\Delta(\mathbf{J}_{iu})$ is defined in Appendix 4. Each $(a_i \times 3)$ block row of the matrix on the right-hand side of (2.4.8), corresponding to the actuated velocities $\dot{\mathbf{q}}_{ia}$, involves only the Jacobian matrix of wheel i. The inverse solution for each wheel is thus independent of the kinematic equations of all of the remaining (N-1) wheels. When wheel i is non-redundant with three-DOFs and all three wheel variables are actuated, each block row of (2.4.8) simplifies to

$$\mathbf{J}_i \dot{\mathbf{q}}_{ia} = \dot{\mathbf{p}}. \tag{2.4.9}$$

We may therefore assume that all of the wheel variables of all of the non-redundant wheels having three-DOFs are actuated, solve the system of w_i linear algebraic equations in (2.4.9) to compute the wheel velocities, and extract the actuated velocities for WMR control. This approach requires approximately one-tenth of the arithmetic operations required for the direct application of (2.4.8).

2.4.3 Sensed Forward Velocity Solution

The sensed forward solution computes the WMR body velocity vector $\dot{\mathbf{p}}$ in (2.4.3) from the sensed wheel positions $\mathbf{q}_s$ and velocities $\dot{\mathbf{q}}_s$. The development of the sensed forward solution parallels the actuated inverse solution in Section 2.4.2. The first step is to separate the sensed and not-sensed wheel velocities and write (2.4.1) as:

$$\dot{\mathbf{p}} = \mathbf{J}_{is}\,\dot{\mathbf{q}}_{is} + \mathbf{J}_{in}\,\dot{\mathbf{q}}_{in} \qquad \text{for } i = 1,\ldots,N\,. \tag{2.4.10}$$

The subscripts "s" and "n" denote the sensed and not-sensed quantities, respectively. The numbers of sensed and not-sensed variables of wheel i are s_i and n_i, respectively (i.e., $s_i+n_i=w_i$). We assume that both the position and velocity of a sensed wheel variable are available. We combine the wheel equations in (2.4.10) for i=1,...,N to form the partitioned robot sensing equation, with all of the unknown robot and wheel positions and velocities on the left-hand side:

$$\begin{pmatrix} \mathbf{I}_1 & -\mathbf{J}_{1n} & 0 & \ldots & 0 \\ \mathbf{I}_2 & 0 & -\mathbf{J}_{2n} & . & : \\ : & : & . & . & 0 \\ \mathbf{I}_N & 0 & \ldots & 0 & -\mathbf{J}_{Nn} \end{pmatrix} \begin{pmatrix} \dot{\mathbf{p}} \\ \dot{\mathbf{q}}_{1n} \\ \dot{\mathbf{q}}_{2n} \\ : \\ \dot{\mathbf{q}}_{Nn} \end{pmatrix} = \begin{pmatrix} \mathbf{J}_{1s} & 0 & \ldots & 0 \\ : & \mathbf{J}_{2s} & . & 0 \\ : & . & . & 0 \\ 0 & \ldots & 0 & \mathbf{J}_{Ns} \end{pmatrix} \begin{pmatrix} \dot{\mathbf{q}}_{1s} \\ \dot{\mathbf{q}}_{2s} \\ : \\ \dot{\mathbf{q}}_{Ns} \end{pmatrix} \tag{2.4.11}$$

or

$$\mathbf{A}_n\,\dot{\mathbf{p}}_n = \mathbf{B}_s\,\dot{\mathbf{q}}_s\,. \tag{2.4.12}$$

We define the total number of sensed wheel velocities to be $s=s_1+\ldots+s_N$ and the total number of not-sensed wheel variables to be $n=n_1+\ldots+n_N$. Thereby, $\mathbf{A}_n$ is $(3N \times [3+n])$, $\dot{\mathbf{p}}_n$ is $([3+n] \times 1)$, $\mathbf{B}_s$ is $(3N \times s)$ and $\dot{\mathbf{q}}_s$ is $(s \times 1)$. We apply the least-squares solution (in Appendix 4) to compute the vector of WMR body and not-sensed wheel velocities $\dot{\mathbf{p}}_n$ from the sensed wheel velocity vector $\dot{\mathbf{q}}_s$:

$$\dot{\mathbf{p}}_n = (\mathbf{A}_n^T\,\mathbf{A}_n)^{-1}\,\mathbf{A}_n^T\,\mathbf{B}_s\,\dot{\mathbf{q}}_s\,. \tag{2.4.13}$$

In Appendix 7, we develop the adequate sensing criterion in (A7.14) which indicates the conditions under which the sensed forward velocity solution in (2.4.13) is applicable. In the remainder of this section, we assume that the sensed forward velocity solution applies and that all normal matrices, such as $(\mathbf{A}_n^T \mathbf{A}_n)$ in (2.4.13), have full rank.

In contrast to the actuated inverse velocity solution, the least-squares forward velocity solution need not produce a zero error because of sensor noise and wheel slippage. In the presence of these error sources, we cannot compute the exact velocity of the WMR body. Our least-squares solution does provide an optimal solution by minimizing the sum of the squared errors in the velocity components. Our least-squares forward solution may thus be applied practically (in Appendix 9) to dead reckoning for a WMR in the presence of sensor noise and wheel slippage.

In Appendix 6, we solve (2.4.11) for the WMR body velocities $\dot{\mathbf{p}}$ to obtain:

Sensed Forward Velocity Solution

$$\dot{\mathbf{p}} = [\Delta(\mathbf{J}_{1n}) + \Delta(\mathbf{J}_{2n}) + \ldots + \Delta(\mathbf{J}_{Nn})]^{-1} [\Delta(\mathbf{J}_{1n})\mathbf{J}_{1s} \quad \Delta(\mathbf{J}_{2n})\mathbf{J}_{2s} \ldots \Delta(\mathbf{J}_{Nn})\mathbf{J}_{Ns}] \dot{\mathbf{q}}_s$$

or

$$\dot{\mathbf{p}} = \mathbf{J}_s \dot{\mathbf{q}}_s . \qquad (2.4.14)$$

A wheel without sensed variables does not contribute any columns $\Delta(\mathbf{J}_{in})\mathbf{J}_{is}$ to (2.4.14). Furthermore, if three independent wheel variables are not sensed, the matrix $\Delta(\mathbf{J}_{in})$ is zero. We may thus eliminate the kinematic equations-of-motion of any wheel which has three not-sensed DOFs in the computation of the sensed forward velocity solution. We note in Appendix 3 that the Jacobian matrix of a steered wheel depends upon the steering angle. Consequently, if any wheel variables of a steered wheel are sensed, the steering angle must also be sensed so that $\mathbf{J}_{in}$ and $\mathbf{J}_{is}$ are computable. Since the matrix $[\Delta(\mathbf{J}_{1n}) + \Delta(\mathbf{J}_{2n}) + \ldots + \Delta(\mathbf{J}_{Nn})]$ is (3×3), solving the system of linear

algebraic equations in (2.4.14) for the WMR body velocities $\dot{\mathbf{p}}$ is *not* a computational burden.

2.4.4 Concluding Remarks

We have combined the equations-of-motion of each wheel on a WMR to formulate and solve the composite robot equation. The actuated inverse velocity solution in (2.4.2) computes the actuated wheel velocities from the WMR body velocity vector and is applicable when the *soluble motion criterion* in (A7.1) of Appendix 7 is satisfied. We have shown that the actuated inverse velocity solution is computed independently for each wheel on a WMR.

The sensed forward velocity solution in (2.4.14) is the least-squares solution of the WMR body velocities in terms of the sensed wheel velocities and is applicable when the *adequate sensing criterion* in (A7.14) of Appendix 7 is satisfied. The least-squares forward velocity solution, which minimizes the sum of the squared errors in the velocity components, is the optimal solution of the WMR body velocities in the presence of sensor noise and wheel slippage. We have found that the sensed forward velocity solution may be simplified by eliminating the equations-of-motion of wheels having three not-sensed DOFs because they do not affect the solution. If any variables of a steered wheel are sensed, the steering angle must also be sensed. In Appendix 9, we apply these computations to dead reckoning for WMR control.

We extend our kinematic solution methodology in Appendix 7 to characterize the solutions of the composite robot equation and their implications for WMR mobility, actuation and sensing. A summary of the results follows. We have developed the *mobility characterization tree* in Figure A7.2 to characterize the motion properties of a WMR. The implications of the mobility characterization tree are summarized by the following insights. If the *soluble motion criterion* in (A7.1) is satisfied, the actuated inverse velocity solution, actuation and sensing trees, and the WMR DOF computation in (A7.4) are applicable. The *three DOF motion criterion* in (A7.2) indicates whether the WMR kinematic structure allows three DOF motion. If the kinematic structure does not allow three DOF motion, the *kinematic motion constraints* are computed according to (A7.3). The *number of WMR DOFs* are then computed from (A7.4).

The implications of the *actuation characterization tree* in Figure A7.3 are summarized by three criteria. The *adequate actuation criterion* in (A7.8) indicates whether the number and placement of the actuators is adequate for producing all motions allowed by the mobility structure. If the adequate actuation criterion is not satisfied, some robot DOFs are uncontrollable. The *robust actuation criterion* in (A7.10) determines whether the actuation structure is robust; i.e., actuator conflict cannot occur in the presence of actuator tracking errors. If the actuation structure is adequate but not robust, some actuator motions are dependent. The *actuator coupling criterion* in (A7.9) computes these actuator dependencies which must be satisfied to avoid actuator conflict and forced wheel slip.

The *sensing characterization tree* in Figure A7.4 indicates properties of the sensing structure of a WMR. The *adequate sensing criterion* in (A7.14) indicates whether the number and placement of the wheel sensors is adequate for discerning all WMR motions allowed by the mobility structure. The *robust sensing criterion* in (A7.15) indicates whether the sensing structure is such that the computation of the WMR body position from wheel sensor measurements is minimally sensitive to wheel slip and sensor noise. The *wheel slip criterion* in (A7.16), which provides a computational algorithm for detecting wheel slip in robust sensing structures, is developed further in Appendix 10.

2.5 Step-by-Step WMR Kinematic Modeling Procedure

We summarize the development of this chapter with a step-by-step enumeration of our WMR kinematic modeling methodology to facilitate engineering applications:

(1) **Make a sketch of the WMR.** Show the relative positioning of the wheels and the steering links. The sketch need not be to scale. A top and a side view are typically sufficient.

(2) **Assign the coordinate systems.** Assign the robot, hip, steering, contact point and floor coordinate systems according to the conventions introduced in Table 2.2.

(3) **Develop the (4×4) coordinate transformation matrices.** Formulate the body-hip, hip-steering, steering-contact and body-contact transformation matrices according to Tables 2.4 and 2.5.

(4) **Formulate the wheel Jacobian matrices.** Formulate the wheel Jacobian matrix, which relates wheel velocities to WMR body velocities, by substituting components of the transformation matrices into the symbolic wheel Jacobian matrix templates compiled in Appendix 3.

(5) **Formulate the wheel acceleration matrices.** Formulate the wheel acceleration matrix-vector equations-of-motion by substituting components of the transformation matrices into the wheel acceleration equation in (2.3.30). The wheel matrix (in Appendix 3) for a specific wheel may then be applied to write the acceleration equations in terms of physical wheel variables.

(6) **Compute the Actuated Inverse Velocity Solution.** Compute the actuated inverse velocity solution by partitioning the wheel Jacobian matrices into actuated and unactuated submatrices, and then applying (2.4.8).

(7) **Compute the Sensed Forward Velocity Solution.** Compute the sensed forward velocity solution by partitioning the wheel Jacobian matrices into sensed and unsensed submatrices, and then applying (2.4.14).

In Chapters 6 and 7 and Appendix 11, we apply our kinematic methodology to eight exemplary WMRs. For each WMR, we follow our step-by-step procedure to compute the actuated inverse and sensed forward velocity solutions, where applicable, and characterize their mobility, actuation and sensing structures.

2.6 Conclusions

We have developed and illustrated a methodology for the kinematic modeling of WMRs. We have found that the established kinematic modeling methodology for stationary manipulators is not applicable to WMRs because of

the higher-pair wheel-to-floor joints, the multiple closed-chains formed by multiple wheels, and the unactuated and unsensed wheel variables. Our development in this chapter spans the kinematic analysis of WMRs, including:

- Assignment of coordinate systems (in Sections 2.3.3 and 2.3.4) ;
- Formulation of the transformation matrices (in Section 2.3.5);
- Formulation of the kinematic model (in Section 2.5);
- Solutions of the kinematic model (in Section 2.4);

and our companion developments in the appendices span the following related WMR kinematic issues:

- Modeling of ball, omnidirectional, and conventional wheels (in Appendix 3);
- Characterization of WMR mobility, actuation, and sensing (in Appendix 7);
- Applications to mechanical design (in Appendix 8), dead reckoning (in Appendix 9), and wheel slip detection (in Appendix 10); and
- Kinematic modeling of six exemplary WMRs (in Appendix 11).

We conclude this chapter by summarizing our development and highlighting the significant results and implications.

We begin modeling a WMR by sketching the mechanical structure. We assign (in Section 2.3.3) one *body* coordinate system, and a *hip*, *steering*, and *contact* coordinate system for each wheel. We apply the Sheth-Uicker convention to coordinate system assignment and transformation matrix computation because it allows the modeling of the higher-pair wheel contact point motion and provides unambiguous transformation matrix labeling for the multiple closed-chains formed by the wheels.

We model each wheel (conventional, steered-conventional, omnidirectional or ball wheel) as a planar-pair which allows three DOFs: x-translation, y-translation, and θ-rotation. A conventional wheel attains y-translational motion by rolling contact. The translation in the x-direction and the θ-rotation about the point-of-contact occur when the wheel slips. We model the rotational slip as a wheel DOF because relatively small forces are required; furthermore, the majority of all WMRs rely on this DOF. We do not consider the x-translational wheel slip a DOF because relatively large forces are necessary.

Omnidirectional wheels rely on rotational wheel slip, whereas ball wheels do not.

By inspection of the sketch, we write the body-to-hip, hip-to-steering, steering-to-contact, and body-to-contact transformation matrices for each wheel in the format of Tables 2.4 and 2.5. Under the assumption of no translational wheel slip, the wheel rotations define the motion of the wheel contact point coordinate system with respect to a stationary coordinate system at the same position and orientation on the floor. The coordinate system fixed with respect to the floor is important because we reference the velocities of the wheel contact point to this *instantaneously coincident coordinate system.* The rotational velocity of a wheel about its axle is thus proportional to the translational velocity of the contact point coordinate system with respect to the instantaneously coincident wheel contact-point coordinate system. Similarly, there is an instantaneously coincident body coordinate system to reference the velocities of the body coordinate system. We assign instantaneously coincident coordinate systems because of the higher-pair wheel contact points.

For each wheel we develop (in Section 2.3.8) a Jacobian matrix to specify the WMR body velocities (in the instantaneously coincident body coordinate system: ${}^{\overline{B}}v_{Bx}$, ${}^{\overline{B}}v_{By}$, ${}^{\overline{B}}\omega_{B}$) as linear combinations of the wheel velocities (e.g., the steering velocity, the rotational velocity about the wheel axle, the rotational slip velocity, and the roller velocities for omnidirectional wheels). We write the Jacobian matrix for a wheel by substituting elements of the coordinate transformation matrices, wheel and roller radii and roller orientation angles into the symbolic Jacobian matrix templates of Appendix 3. For a steered wheel, the Jacobian matrix depends explicitly on the steering angle.

Our research has illuminated the following important wheel properties. A $(3 \times w_i)$ Jacobian matrix $\mathbf{J}_i$ is associated with a wheel having w_i wheel variables. A Jacobian matrix having rank w_i satisfies the non-redundant wheel criterion in (2.3.28), the wheel has w_i DOFs and all wheel variables are independent. If the rank of the Jacobian matrix is less than w_i, the wheel is redundant having fewer than w_i DOFs, and some of the wheel variables are dependent. For example, any conventional wheel which is steered about an axis that intersects the wheel contact point, or is oriented perpendicularly to the line from the steering axis to the contact point, is redundant. We have noted the following disadvantages of

redundant wheels (without wheel couplings). The actuated inverse and sensed forward velocity solutions do not apply. We cannot characterize the actuation and sensing structure of WMRs with redundant wheels because the actuation and sensing characterization trees are created by applying the actuated inverse solution. We also cannot determine the number of DOFs of a WMR with redundant wheels (and no wheel couplings) because the DOF specification in (A7.4) is not computable.

Since the actuated inverse velocity solution is not applicable, we cannot control such a WMR by computing the actuator velocities from the desired WMR body velocities. Steering the WMR by computing the steering angle of a redundant wheel is an ad-hoc approach since a steering angle cannot be controlled instantaneously. We point-out that some existing WMRs having redundant steered conventional wheels (e.g., Neptune and the Stanford Cart) are controlled in this manner with some success. Since our survey (in Appendix 2) and examples (in Appendix 11) show that WMRs have been designed with redundant wheels, we infer that the implications of redundant wheels were not previously well-understood.

We have formed the *composite robot equation* (in Section 2.4.1) by adjoining the equations-of-motion of all of the wheels. Linear positional couplings between wheel variables (e.g., steering angles and wheel axle angles) can be incorporated into our model by making the appropriate substitutions in the composite robot equation, as demonstrated in Appendix 11.7 for the Stanford Cart. We have solved the composite robot equation and interpreted properties of the solutions to illuminate the mobility characteristics of WMRs.

We have computed two solutions of the composite robot equation: the actuated inverse and sensed forward velocity solutions. In the *actuated inverse velocity solution* in (2.4.8), we compute the actuated wheel velocities from the desired WMR body velocities. The actuated inverse velocity solution is applicable for WMRs satisfying the soluble motion criterion (in A7.1). In the *sensed forward velocity solution* in (2.4.14), we compute the WMR body velocities from the sensed wheel velocities. The adequate sensing criterion (in A7.14) indicates whether the sensed forward velocity solution is applicable for a specific WMR. If there are no wheel couplings, the composite robot equation in

(2.4.2) need not be formed explicitly because the actuated inverse and sensed forward velocity solutions are expressed in terms of the wheel Jacobian matrices. The computations required for the actuated inverse and sensed forward velocity solutions are additions, multiplications and the solution of (at most) three *linear* algebraic equations.

WMR kinematics play fundamental roles in servo-control, mobility characterization, mechanical design, dead reckoning, wheel slip detection and trajectory feasibility determination. We comment briefly on each application area in turn.

In Chapter 4, we apply kinematics to *servo-control.* The few documented WMR control systems are wheel level control systems [Daniel84, Hongo85], which do not incorporate a dynamic model of the WMR. The documented designs are tailored to the specific WMR being controlled. We detail a kinematics-based Cartesian-space servo-controller (in Section 4.2) for WMRs requiring the sensed forward and actuated inverse velocity solutions.

The composite robot equation in (2.4.3) may have zero, one, or an infinite number of solutions corresponding to three WMR mobility characterizations: overdetermined, determined, and undetermined, respectively. The *mobility characterization tree* (in Figure A7.2) enables us to determine the mobility characteristics of a WMR by indicating tests to be conducted on the composite robot equation. It is both impractical and unnecessary to actuate and sense every wheel variable on a WMR because of the closed-chains. A subset of the wheel variables is thus actuated, and a subset (not necessarily the same subset) is sensed. Even though a specific WMR may allow three DOF motion, we must guarantee that the wheel actuators can actuate all three DOFs, and that the sensors can discern all three DOFs. We apply the *actuation and sensing characterization trees* (in Figures A7.3 and A7.4, respectively) to address these issues.

Just as the mobility characterization tree enables us to determine the motion characteristics of an existing WMR, we may apply the tree to design WMRs to possess such desired characteristics as two or three DOFs. We may design a WMR with any specified workspace constraints (i.e., set of allowable motions) by proper choice and placement of the wheels. In Table A8.1, we list

the design criteria for a robust omnidirectional WMR. The actuation characterization tree may be applied to design a WMR to have a robust actuation structure, thus avoiding actuator conflict, as shown for Uranus in Appendix 11.4. Similarly, the sensing characterization tree may be applied to design a WMR with a robust sensing structure to minimize the adverse effects of wheel slip on the computation of the WMR position. We have noted that the set of actuated wheel variables and sensed wheel variables cannot coincide if both robust actuation and robust sensing are desired. To illustrate our kinematic modeling methodology, we model two *three* DOF WMRs: Uranus (tetroas-whemor in Chapter 6) and the Unimation robot (troas-whemor in Appendix 11.2). In Chapter 8, we suggest that three DOF WMRs are applicable for use with an on-board manipulator. The mobility of the base extends the workspace of the manipulator. The majority of practical applications (i.e., parts, tools, and materials transport) require only two DOFs. We conclude that a WMR having two diametrically opposed driven wheels (bicas-polycsun-whemor) is ideal for this general purpose application because of the simplicity of its mechanical design and kinematic model.

Dead reckoning is the real-time computation of the robot position from wheel sensor measurements. In Appendix 9, we develop a dead reckoning update algorithm by integrating the WMR body velocity computed by the sensed forward velocity solution.

We have uncovered three methods of dealing with wheel slip: design the actuation structure to avoid slip, design the sensing structure to detect slip, and minimize the errors in the computed WMR body position due to slip. We model rotational wheel slip for both conventional and omnidirectional wheels because many WMR designs rely on this DOF. We wish to avoid, detect or minimize the adverse effects of the unmodeled translational wheel slip. One approach to eliminating wheel slip is to actuate all of the wheels, such as with the four-wheel drive on an automobile. Since this can lead to actuator conflict, we must design wheel couplings to ensure that the actuator coupling criterion is satisfied, as with Uranus (in Appendix 11.4). This solution does not guarantee zero wheel slip, but if slip does occur, all wheels must slip in unison which is unlikely. Since a robust sensing structure allows us to detect wheel slip, we design the sensing structure to satisfy the robust sensing criterion and detect wheel slip by the

method of Appendix 10. In this way, we are able to detect the onset of wheel slip and notify the robot processor that an absolute method of robot positioning (e.g., robot vision) should be applied before continuing. This method will fail in the unlikely case that all wheels slip in unison. The least-squares sensed forward velocity solution (in Section 2.4.3) is less sensitive to wheel slippage if the sensing structure is designed to be robust. If wheel slip does occur, and no absolute positioning method is available, the adverse effects can be reduced by applying the least-squares sensed forward velocity solution.

The kinematic motion constraints (computed in Appendix 7) may be applied to assess the feasibility of candidate WMR trajectories. The mobility characterization tree provides a means of computing the constraints on the velocities of the WMR from the composite robot equation. These constraint equations may be applied to determine whether a proposed trajectory, computed by the motion planner, is executable, and if not, how to adjust the trajectory to be within the WMRs kinematic motion capabilities.

In Chapter 3, we apply our kinematic methodology to the dynamic modeling of WMRs. In analogy with manipulator dynamic modeling, our WMR kinematic methodology serves as the foundation upon which to formulate WMR dynamic models. In contrast to manipulator dynamics, we must resolve the special WMR characteristics of closed-chains, higher-pair joints, friction and unactuated and unsensed joints.

Chapter 3
Dynamic Modeling of WMRs

3.1 Introduction

An accurate mathematical model of a complex robotic mechanism, such as a WMR, is paramount for integrated electrical, mechanical and servo-controller design. Enhancing the accuracy of the WMR model improves performance, increasing productivity and extending capabilities. Such has been the case with stationary robotic manipulators. A *kinematic* robot model, describing the relative motions (i.e., the positions, velocities and accelerations) between the joints and the end-effector, leads to servo-controller designs which provide satisfactory performance for such simple tasks as teleoperation at low speed and small payloads [Whitney69]. A *dynamic* robot model, relating the joint and the end-effector masses/inertias, forces/torques, and motions, leads to servo-controller designs which provide accurate performance for such tasks as pick and place operations and Cartesian-space path control [Luh80a]. Prior to this dissertation, WMR models were kinematic and consequently WMR servo-controllers were kinematics-based [Daniel84, Hongo85]. In Chapter 2, we have developed a WMR kinematic modeling methodology. In analogy with stationary manipulator servo-control, WMR performance will improve significantly with the formulation of dynamic WMR models.

In this chapter, we formulate dynamic models of WMRs to pave the way for the design of high-accuracy, dynamics-based WMR servo-controllers. The dynamic model of a robotic mechanism (e.g., a WMR) describes the temporal evolution of the outputs (e.g., the WMR positions, velocities, and accelerations) in response to the system inputs (e.g., the wheel actuator torques). In Section 3.2, we highlight the established stationary manipulator dynamic modeling

formulations. We conclude that none of the established formulations are naturally applicable to accurate dynamic modeling incorporating all of the special WMR characteristics discussed in Section 1.5: (1) closed-chains; (2) higher-pair joints; (3) unactuated joints; (4) unsensed joints; (5) friction; and (6) pulse-width modulation. The requirement for a WMR dynamic modeling methodology provides the impetus for our development of a novel dynamic modeling framework which fulfills the WMR modeling requirements. We present, in Section 3.3, our dynamic modeling framework which is applicable to a broad spectrum of multi-link robotic mechanisms including stationary manipulators, multi-manipulators, WMRs, legged mobile robots and robotic hands. We apply Newtonian dynamics to WMRs, enhancing this dynamic modeling approach, as required, to accommodate the six aforementioned special characteristics of WMRs.

Our dynamic modeling framework allows the independent formulation of the force/torque equations-of-motion and the kinematic transformation equations. In Section 3.4, we detail the companion kinematic transformations required for the dynamic model. We combine the force/torque equations-of-motion and the kinematic transformations to form the WMR dynamic model. In Section 3.5, we apply classical methods to solve this system of linear algebraic equations. We compute the actuator forces/torques from the WMR accelerations, velocities and positions in the actuated inverse dynamic solution; and the WMR accelerations from the actuator forces/torques in the forward dynamic solution.

In Section 3.6, we apply our dynamic modeling framework to WMRs. Our development culminates in a *sixteen*-step WMR dynamic modeling procedure. In Section 3.7, we conclude the chapter by summarizing the dynamic modeling framework. We have relegated interesting issues related to our dynamic modeling framework to Appendices 13-15. In Appendix 13, we illustrate the extension of our dynamics framework to model flexible links. We display in Appendix 14 a graphical representation of our system dynamic model. Finally, in Appendix 15, we apply our dynamics framework to model a planar double pendulum and a biped in the frontal plane to show agreement with dynamic models of these robotic mechanisms available in the literature.

3.2 Manipulator Dynamic Modeling Methodologies

The Lagrange [Hollerbach80, Paul81a] and Newton-Euler [Luh80b, McInnis86] formulations are the two dynamics methodologies most widely applied to stationary manipulator modeling. Both model open-chains containing lower-pair joints. When dry friction is incorporated [Paul81a], the frictional force/torque is added to the actuator force/torque for each joint. This dry friction modeling procedure does *not* generalize to chains containing *unactuated* joints, since the dry friction at the unactuated joints does not affect the computed actuator forces/torques. The dependence of the dry friction on the normal force is also neglected in this dry friction model. Viscous friction is incorporated for the actuators, but not for the robot links [Pfeifer84]. Finally, application of these dynamics formulations to robot servo-control requires that *all* of the joint positions and velocities be sensed. Consequently, neither the Lagrange nor the Newton-Euler formulations are naturally applicable to WMR modeling.

Extensions to the Lagrange and Newton-Euler dynamics formulations allow the modeling of a broader range of mechanisms. Kane [Kane61] has applied partial rates of change to develop a dynamics formulation for nonholonomic systems. Wittenburg [Wittenburg77] developed a methodology for the dynamic modeling of closed-chains by reducing the closed-chain structure to a tree-structure. The dynamic model of the tree structure is written by open-chain methodologies and the kinematic constraints at the ignored joints are introduced to complete the dynamics of the closed-chain structure. Draganoiu [Draganoiu82] and Luh [Luh85] apply similar methods to model stationary manipulators having closed-chain structures. Orin and Oh [Orin81] model a closed-chain mechanism as a set of open-chains, each with a six-dimensional matrix-vector force/torque model. Orin and Oh then apply linear programming to compute the inverse solution which optimizes a weighted combination of energy consumption and load balancing for a hexapod locomotion vehicle.

Although some of these established formulations model nonholonomic constraints and closed-chains, none address the problem of unactuated and unsensed joints, and none are naturally applicable for incorporating the effects of dry friction (i.e., stiction and Coulomb and rolling friction at the wheel point-of-contact and at the bearings) and viscous friction. The lack of an adequate

methodology for the integrated dynamic modeling of all six of the special WMR characteristics provides the impetus for our development of a *unifying dynamics framework* (introduced in the next section) which applies to a broad class of multi-link robotic mechanisms.

3.3 A Unifying Dynamic Modeling Framework

3.3.1 Overview

We develop a unifying dynamics framework which satisfies the WMR dynamic modeling requirements discussed in Section 1.5 and applies to a broad class of multi-link robotic mechanisms. We apply Newtonian dynamics, D'Alembert's principle, six-dimensional matrix-vector force/torque modeling, and instantaneously coincident coordinate systems (introduced in Chapter 2) to formulate the dynamic model of a WMR. Our approach is to construct the conceptually and computationally complex WMR dynamic model from conceptually and computationally simpler component force/torque models by conceptually simple force/torque manipulation methods.

At any time instant, a robotic mechanism of rigid links (e.g., a WMR) may be viewed as a massless structural frame to which forces and torques are applied at disparate points. Motion of the mechanism is represented by changing inertial forces/torques according to D'Alembert's principle. The six-dimensional sum of all forces/torques on the structure is thus zero at any time instant. In Section 3.3.2, we model the force/torque sources which act on robotic mechanisms. We utilize the six-vector notation introduced in Section 1.6 to model inertial, gravitational, actuation, viscous friction, and environmental contact forces/torques. Each force/torque model consists of a six-vector template of forces/torques and a *natural* coordinate system in which the model of the force/torque vector is conceptually the simplest. In our modeling paradigm, each force/torque source is applied to the point on the mechanical structure corresponding to the origin of its natural coordinate system. For example, the inertial torques on a rigid link are modeled most simply by Euler's equation applied in a coordinate system located at the center-of-mass of the link and aligned with its principal axes [Landau76].

Forces/torques applied by different point sources interact by *propagating* through the structural frame (as described in Section 3.3.3). The structural frame consists of massless links interconnected by joints. The force/torque propagation properties of each link are described by kinematic positional parameters within the *link Jacobian matrix*. The force/torque propagation properties of each joint are described by dry frictional coupling parameters within a *joint coupling matrix*. We model dry friction (i.e., stiction, Coulomb friction and rolling friction) as a force (torque) coupling phenomenon because dry friction couples normal forces (torques) along one axis to frictional forces along another axis at a joint: i.e., if the normal force (torque) is zero, there is no dry frictional force (torque). This contrasts with the conventional view of dry friction as a source of forces/torques at a joint. In Section 3.3.4, we compute the force/torque equations-of-motion of the robotic mechanism by summing the forces/torques within a common coordinate system on the structure (the center of mass of the main link of the system) to zero according to Newton's equilibrium law.

3.3.2 Six-Vector Force/Torque Modeling

3.3.2.1 Force/Torque Sources

The forces/torques acting on the links within a robotic mechanism originate from the following sources:

- Inertial (in Section 3.3.2.2);
- Gravitational (in Section 3.3.2.3);
- Actuation (in Section 3.3.2.4);
- Viscous friction (in Section 3.3.2.5); and
- Environmental contact (in Section 3.3.2.6).

The dynamic model of each of these forces/torques and their corresponding natural coordinate systems are detailed in the indicated section. We utilize the conceptually simple dynamic models of these forces/torques as the modular building blocks for the systematic formulation of the dynamic model of a robotic mechanism.

3.3.2.2 Inertial Forces/Torques

The *inertial forces* acting on rigid link *A* are proportional to the translational accelerations of the link referred to an *inertial* (Newtonian [Landau76]) coordinate system located at the *center-of-mass*[1] of the link:

$$^{M(A)}f_{iA} = m_A \, ^{\bar{M}(A)}a_{M(A)} \; . \qquad (3.3.1)$$

In (3.3.1), M(*A*) is a coordinate system fixed to link A at its center-of-mass, $\bar{M}(A)$ is an inertial coordinate system instantaneously coincident with the center-of-mass of link *A*, $^{M(A)}f_{iA} = [^{M(A)}f_{iAx} \; ^{M(A)}f_{iAy} \; ^{M(A)}f_{iAz}]^T$ is the three-vector of inertial forces acting on rigid link *A*; m_A is the mass of link *A*; and $^{\bar{M}(A)}a_{M(A)} = [^{\bar{M}(A)}a_{M(A)x} \; ^{\bar{M}(A)}a_{M(A)y} \; ^{\bar{M}(A)}a_{M(A)z}]^T$ is the acceleration three-vector of link *A*.

The inertial *torques* acting on a rigid link, which are a function of the angular velocities and accelerations of the link, are decoupled from the inertial forces in (3.3.1) when referred to a coordinate system located at the center-of-mass of the link. The inertial torque computation for link *A* is simplified in the natural coordinate system M($\bar{A}$) which is located at the center-of-mass and aligned with the *principal axes* [2]

$$^{M(A)}\tau_{iA} = {}^{M(A)}\mathbf{I}_A \, ^{\bar{M}(A)}\alpha_{M(A)} + {}^{\bar{M}(A)}\omega_{M(A)} \times {}^{M(A)}\mathbf{I}_A \, ^{\bar{M}(A)}\omega_{M(A)} \; . \qquad (3.3.2)$$

In (3.3.2), $^{M(A)}\tau_{iA} = [^{M(A)}\tau_{iAx} \; ^{M(A)}\tau_{iAy} \; ^{M(A)}\tau_{iAz}]^T$ is the three-vector inertial torque acting on link *A*; $^{M(A)}\mathbf{I}_A = \mathbf{Diag}\left({}^{M(A)}I_{Ax} \; {}^{M(A)}I_{Ay} \; {}^{M(A)}I_{Az} \right)$ is the constant (3×3) diagonal *inertia matrix*[3] of link *A*; $^{\bar{M}(A)}\alpha_{M(A)} = [^{\bar{M}(A)}\alpha_{M(A)x} \; ^{\bar{M}(A)}\alpha_{M(A)y} \; ^{\bar{M}(A)}\alpha_{M(A)z}]^T$ is the three-vector angular acceleration of link *A*; $^{M(A)}\omega_{M(A)} =$

---

1 The center-of-mass of a homogeneous simple rigid link (i.e., a rectangular block, cylinder or sphere) is located at the geometric center of the link.

2 The principal axes of a homogeneous simple rigid link are aligned with the geometric axes of symmetry. For an arbitrary rigid link, there is a set of principal axes at any point fixed to the link.

3 The inertia matrix of a rigid link about a coordinate system describes the distribution of mass about the axes of the coordinate system. The inertia matrix is constant relative to a coordinate system fixed to the rigid link (e.g., M(*A*)) and is diagonal about the principal axes of the link.

$[{}^{\overline{M(A)}}\omega_{M(A)x}\ {}^{\overline{M(A)}}\omega_{M(A)y}\ {}^{\overline{M(A)}}\omega_{M(A)z}]^T$ is the three-vector angular velocity of link *A*; and the × denotes the vector cross-product. We combine (3.3.1) and (3.3.2) to form the six-vector inertial force/torque acting on link *A* at the natural center-of-mass coordinate system M(*A*) as:

$$ {}^{M(A)}\mathbf{f}_{iA} = [{}^{M(A)}\mathbf{f}_{iA}^T \quad {}^{M(A)}\tau_{iA}^T]^T . \tag{3.3.3} $$

The inertial forces/torques in (3.3.3) are of three types: *self*-inertial forces/torques which are proportional to accelerations (a_x, a_y, a_z, α_x, α_y, and α_z), *centripetal* forces/torques which are proportional to products of different velocities ($\omega_x\omega_y$, $\omega_x\omega_z$, and $\omega_y\omega_z$), and *Coriolis* forces/torques which are proportional to squares of velocities (ω_x^2, ω_y^2, and ω_z^2). The Coriolis forces/torques in our inertial force/torque model in (3.3.3) are zero. However, by computing in Section 3.4 the angular velocities in (3.3.3) from velocities of the main robot link and the velocities of the joints within the robotic mechanism, nonzero Coriolis forces/torques appear in our dynamic model.

3.3.2.3 Gravitational Forces

Newton's gravitational law expresses the gravitational forces/torques ${}^{G(A)}\mathbf{f}_{gA}$ acting on rigid link *A* as the product of the mass of link *A* and the gravitational constant g=9.8 meters per second:

$$ {}^{G(A)}\mathbf{f}_{gA} = -m_A\, g\, \mathbf{e}_i \tag{3.3.4} $$

where the six-vector $\mathbf{e}_i$ is $[001000]^T$ and the natural coordinate system for the gravitational law G(*A*) is located at the center-of-mass of link *A* with its z-axis aligned opposite to the gravitational field.

3.3.2.4 Actuation Forces/Torques

The force (torque) of an actuator applied to a rigid link equals the force (torque) produced by the actuator in direct-drive applications. Gearing may be introduced to couple the actuator and rigid link. The linear (angular) velocity of

the link about the actuator axis is then equal to $1/\beta$ times the linear (angular) velocity of the actuator, where β is the *gear ratio*. The actuation force/torque six-vector ${}^{P(A)}\mathbf{f}_{aA}$ applied to rigid link *A* along (or about) the x, y, or z axis of the natural point-of-application coordinate system P(*A*) is:

$$ {}^{P}\mathbf{f}_{aA} = -{}^{P}\mathbf{f}_{aB} = \begin{cases} \beta\, f\, \mathbf{e}_i & \text{for a linear actuator} \\ \beta\, \tau\, \mathbf{e}_i & \text{for a rotational actuator .} \end{cases} \qquad (3.3.5) $$

where f (τ) is the force (torque) exerted by the actuator, and ${}^{P}\mathbf{f}_{aB}$ is the equal and opposite reactional force/torque six-vector (according to Newton's action-reaction law) which acts on the adjacent link B. In general, the assignment of the links *A* and *B* in (3.3.5) is arbitrary. We apply the convention of modeling the actuator force/torque in (3.3.5) on link *A* which is kinematically *furthest* from the main link of the robotic mechanism, and the reactional force/torque on link *B* which is kinematically *closest* to the main link.

3.3.2.5 Viscous Friction Forces/Torques

Viscous frictional forces/torques are exerted on a link which is moving through a gaseous or liquid medium such as air or water. The viscous friction acting on link *A* is a function of the surface area of the link as viewed from the direction-of-motion, the shape of the link, and the surface texture. Since many links have approximately regular shapes (i.e, rectangular boxes, cylinders or spheres), a coordinate system may be fixed to the link having axes perpendicular to sides of equal area. We apply this natural *center-of-volume* coordinate system V(*A*) to model viscous friction.

The viscous frictional force/torque exerted by air on a rigid link moving with one DOF is conventionally modeled as being proportional to (a power of) the velocity of the rigid link [Landau76]. We propose the following natural extension of this model to approximate the viscous frictional forces/torques exerted on a rigid link moving with six DOFs:

$$ {}^{V(A)}\mathbf{f}_{vA} = {}^{V(A)}\mathbf{V}_A \; {}^{\overline{V}(A)}\mathbf{v}_{V(A)} . \qquad (3.3.6) $$

In (3.3.6), ${}^{V(A)}\mathbf{f}_{vA}$ is the viscous frictional force/torque six-vector, ${}^{\bar{V}(A)}\mathbf{v}_{V(A)}$ is the velocity six-vector, and we approximate the (6×6) viscous friction matrix ${}^{V(A)}\mathbf{V}_A$ by the diagonal matrix:

$$ {}^{V(A)}\mathbf{V}_A = \mathbf{Diag}\left(\begin{array}{cccccc} \mu_{vAx} & \mu_{vAy} & \mu_{vAz} & \varepsilon_{vAx} & \varepsilon_{vAy} & \varepsilon_{vAz} \end{array} \right). \qquad (3.3.7) $$

In (3.3.7), μ_{vAi} is the translational viscous frictional coefficient along the i-th axis, and ε_{vAi} is the rotational viscous frictional coefficient about the i-th axis. Because of the coupling of forces/torques between axes, the viscous friction matrix for links of complex shape is not diagonal. To simplify our WMR dynamic modeling, we assume that the diagonal matrix in (3.3.7) is appropriate to model the viscous frictional forces/torques on WMR links. Viscous frictional forces/torques on a rigid link in air are oftentimes negligible. Viscous frictional forces/torques are significant in underwater applications, but not in outer space applications.

3.3.2.6 Environmental Contact Forces/Torques

When link *A* of a robotic mechanism contacts body *E* external to the mechanism (such as the distal link of one of the open-chains in Figure 1.1) the external body may exert *environmental contact forces/torques* ${}^{C(E,A)}\mathbf{f}_{eA}$ on the contacting link at the contact point coordinate system C(*E*,*A*) fixed to body *E*:

$$ {}^{C(E,A)}\mathbf{f}_{eA} = \left(\begin{array}{cccccc} f_{eAx} & f_{eAy} & f_{eAz} & \tau_{eAx} & \tau_{eAy} & \tau_{eAz} \end{array} \right)^T. \qquad (3.3.8) $$

The natural coordinate system C(*E*,*A*) for modeling the environmental forces/torques exerted on link *A* is located at the contact point. If there is line or surface contact, we assign the contact point coordinate system at the geometric center of the contact distance or area, respectively.

3.3.3 Matrix-Vector Force/Torque Propagations

3.3.3.1 Force/Torque Propagations

In reality, some forces/torques are applied to a link at a point (e.g., actuation and environmental contact forces/torques), and some are distributed over the mass (e.g., inertial and gravitational forces/torques) or surface of the

link (e.g., viscous frictional forces/torques). For our engineering analysis in Section 3.3.2, we model all of the forces/torques on a link as being applied at a specific point: the origin of the *natural* coordinate system. We choose to model the forces/torques in the natural coordinate systems because the models are conceptually simplest. We may model a force/torque in *any* coordinate system; but the components of the force/torque vector depend upon the location and orientation of the coordinate system. For example, we may model a force/torque acting on link *A* at two distinct coordinate systems A_1 and A_2. The force/torque vectors modeled within the coordinate systems A_1 and A_2 are then denoted by ${}^{A1}\mathbf{f}_A$ and ${}^{A2}\mathbf{f}_A$, respectively. The force/torque vector ${}^{A1}\mathbf{f}_A$ applied within coordinate system A_1 thus has the identical effect on link *A* as the force/torque vector ${}^{A2}\mathbf{f}_A$ applied within coordinate system A_2. We refer to the method of computing the force/torque vector ${}^{A2}\mathbf{f}_A$ from the force/torque vector ${}^{A1}\mathbf{f}_A$ as *force/torque propagation*. In analogy with the propagation of light and sound waves through a medium, force/torque sources at a point within a robotic mechanism propagate their effects throughout the system.

To formulate the dynamic model of a robotic mechanism, we apply (in Section 3.3.3.2) Newton's equilibrium law, summing the forces/torques acting on the robotic mechanism *in a common coordinate system*. Since (in Section 3.3.2) we model each force/torque within its own natural coordinate system, we develop in this section a methodology for propagating the forces/torques from the natural coordinate systems to kinematically distant coordinate systems. We may thereby propagate the forces/torques to any common coordinate system for application of Newton's equilibrium law.

In Figure 3.1, we depict a mechanical open-chain consisting of a series of rigid links *A* to *M* connected by joints. We assign two coordinate systems to each link; one at each joint. For example, coordinate systems A_1 and A_2 are fixed to link *A*, and B_1 and B_2 are fixed to link *B*. The two coordinate systems at each joint (e.g., A_2 and B_1) are located at the same position and orientation in space except for translations and rotations of the joint variables according to the Sheth-Uicker convention [Sheth71]. When all joint variables are zero, the two coordinate systems at each joint coincide. Only two dimensions are shown in Figure 3.1. We make a direct correspondence between the open-chain in Figure 3.1 and each open-chain in the simple closed-chain robotic mechanism

depicted in Figure 1.1, with link *A* playing the role of the distal link and link *M* playing the role of the main link of the robotic mechanism.

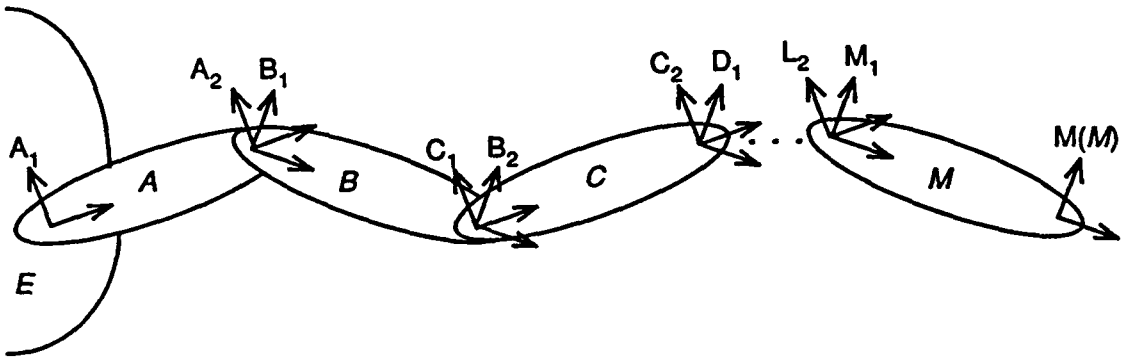


Figure 3.1: An Open-Chain of Rigid Links

Our objective is to propagate the forces/torques originating in coordinate system A_1 to coordinate system M(*M*). We begin in Section 3.3.3.2 by propagating the forces/torques through link *A* from coordinate system A_1 to coordinate system A_2. In Section 3.3.3.3, we propagate the forces/torques through the joint from link *A* to link *B* in coordinate system B_1. We generalize these results in Section 3.3.3.4 to propagate directly forces/torques on link *A* in coordinate system A_1 to forces/torques on link *M* in coordinate system M(*M*). In Section 3.3.3.5, we consider the special requirements of propagating gravitational forces/torques.

3.3.3.2 Link Force/Torque Propagation

In this section, we develop a method for propagating the forces/torques through link *A* from coordinate system A_1 to coordinate system A_2. We equate the *instantaneous powers* in the two coordinate systems to obtain the solution. Our methodology parallels the developments in the literature [Paul81a] which equate the virtual work in the two coordinate systems. We introduce the instantaneous power approach as an alternative to the manipulation of differential displacement vectors. Instantaneous power in one dimension is the product of force and velocity. In six dimensions, we compute the instantaneous power ${}^{A_1}p_A$ associated with a force/torque ${}^{A_1}\mathbf{f}_A$ acting on link *A* moving with velocity ${}^{A_1}\bar{\mathbf{v}}_{A_1}$ as the inner product

$$ {}^{A_1}p_A = {}^{A_1}f_A^T \, {}^{\bar{A}_1}v_{A_1} . \qquad (3.3.9) $$

Our objective is to compute the equivalent force/torque vector ${}^{A_2}f_A$ in coordinate system A_2 ; i.e., to propagate the force/torque vector from coordinate system A_1 to coordinate system A_2. Since energy is conserved within a rigid link, the time rate-of-change of energy (i.e., the power) is independent of our choice of coordinate systems, and we equate the instantaneous power computed in coordinate system A_1 to the instantaneous power computed in coordinate system A_2:

$$ {}^{A_1}f_A^T \, {}^{\bar{A}_1}v_{A_1} = {}^{A_2}f_A^T \, {}^{\bar{A}_2}v_{A_2} . \qquad (3.3.10) $$

The velocity ${}^{\bar{A}_1}v_{A_1}$ is related *linearly* to the velocity ${}^{\bar{A}_2}v_{A_2}$ by the (6×6) link Jacobian matrix ${}^{A_1}L_{A_2}$ (derived in Section 3.4.2):

$$ {}^{\bar{A}_1}v_{A_1} = {}^{A_1}L_{A_2} \, {}^{\bar{A}_2}v_{A_2} . \qquad (3.3.11) $$

The link Jacobian matrix ${}^{A_1}L_{A_2}$ is computed by substituting the elements of the position vector ${}^{A_2}p_{A_1}$ into the template for the link Jacobian in Appendix 12. We substitute the velocity transformation in (3.3.11) into the instantaneous power balance in (3.3.10) to obtain

$$ {}^{A_1}f_A^T \, {}^{A_1}L_{A_2} \, {}^{\bar{A}_2}v_{A_2} = {}^{A_2}f_A^T \, {}^{\bar{A}_2}v_{A_2} . \qquad (3.3.12) $$

Since (3.3.12) must hold for all velocities ${}^{\bar{A}_2}v_{A_2}$, we find, by transposing both sides, that

$$ {}^{A_2}f_A = {}^{A_1}L_{A_2}^T \, {}^{A_1}f_A . \qquad (3.3.13) $$

Forces/torques are thus propagated through links by applying the transpose of the link Jacobian matrix according to (3.3.13). The link Jacobian matrix ${}^{A_1}L_{A_2}$ is dependent only upon the position/orientation ${}^{A_2}p_{A_1}$ and is thus independent of the motion (i.e., velocities and accelerations) of the coordinate systems A_1 and A_2. We note that (3.3.13) holds for any two coordinate systems, specifically:

$$ {}^{B_1}\mathbf{f}_A = {}^{A_2}\mathbf{L}_{B_1}^{T}\, {}^{A_2}\mathbf{f}_A \,. \qquad (3.3.14) $$

By substituting (3.3.14) into (3.3.13), we arrive at the *cascade* property of link Jacobian matrices:

$$ {}^{A_1}\mathbf{L}_{B_1} = {}^{A_1}\mathbf{L}_{A_2}\, {}^{A_2}\mathbf{L}_{B_1} \,. \qquad (3.3.15) $$

The link Jacobian cascade property in (3.3.15) generalizes to any number of intermediate coordinate systems.

We focus on link *A* in Figure 3.2 to provide an intuitive understanding of force/torque propagation through a rigid link.

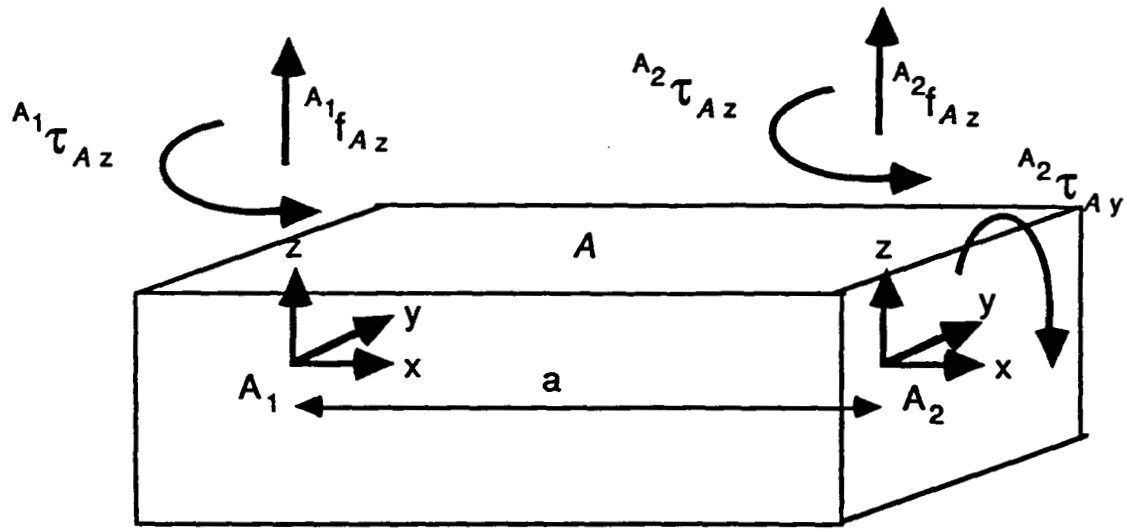


Figure 3.2: A Rigid Link

In Figure 3.2, coordinate systems A_1 and A_2 are parallel and separated by the distance a. In general, coordinate systems A_1 and A_2 need not be parallel, but this simplifies the example. The force/torque vector propagated to coordinate system A_2 is computed from the transposed link Jacobian matrix and the force/torque vector applied at coordinate system A_1 according to (3.3.13):

$$\begin{pmatrix} {}^{A2}f_{Ax} \\ {}^{A2}f_{Ay} \\ {}^{A2}f_{Az} \\ {}^{A2}\tau_{Ax} \\ {}^{A2}\tau_{Ay} \\ {}^{A2}\tau_{Az} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & a & 0 & 1 & 0 \\ 0 & -a & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} {}^{A1}f_{Ax} \\ {}^{A1}f_{Ay} \\ {}^{A1}f_{Az} \\ {}^{A1}\tau_{Ax} \\ {}^{A1}\tau_{Ay} \\ {}^{A1}\tau_{Az} \end{pmatrix}. \qquad (3.3.16)$$

We make the following correspondences between Figure 3.2 and (3.3.16). The scalar force ${}^{A1}f_{Az}$ applied along the z-axis of coordinate system A_1 produces an equal force ${}^{A2}f_{Az}$ along the z-axis of coordinate system A_2, and a torque ${}^{A2}\tau_{Ay}$ about the y-axis equal to the applied force multiplied by the moment arm a. The *direct* force propagation is denoted in the transposed link Jacobian in (3.3.16) by the unit element in the diagonal (3,3) position, and the *indirect* force-to-torque propagation is denoted by the (5,3) off-diagonal element. The scalar torque ${}^{A1}\tau_{Az}$ applied about the z-axis of coordinate system A_1 produces an equal torque ${}^{A2}\tau_{Az}$ about the z-axis of coordinate system A_2. The direct torque propagation is denoted in (3.3.16) by the unit (6,6) diagonal element. The transposed link Jacobian matrix ${}^{A2}\mathbf{L}^{T}_{A_1}$ is a nonlinear function of the position/orientation ${}^{A1}\mathbf{p}_{A_2}$ of coordinate system A_2 relative to coordinate system A_1. The upper-right (3×3) submatrix of the transposed link Jacobian is zero because torques do not propagate to forces. The upper-left (3×3) submatrix is equal to the lower-right (3×3) submatrix and is a nonlinear function of the orientation angles of coordinate system A_2 relative to coordinate system A_1. In the example in (3.3.16), these two submatrices are identity matrices because the two coordinate systems A_1 and A_2 in Figure 3.2 are *parallel*. *Propagation of forces to forces and torques to torques thus depends upon the relative orientation of the coordinate systems.* The lower-left (3×3) submatrix is a nonlinear function of the positions and orientation angles. *Propagation of forces to torques depends upon both the relative position and orientation of the coordinate systems.*

3.3.3.3 Joint Force/Torque Propagation

In this section, we propagate forces/torques through a joint from link *A* to link *B* at coordinate system B_1 which is located at the joint between the two links. Our goal is to compute the forces/torques ${}^{B1}\mathbf{f}_B$ on link *B* from the forces/torques ${}^{B1}\mathbf{f}_A$ on link *A* at the joint. The characteristics of force (torque) propagation along (about) axes of a joint coordinate system differ depending whether or not the axes are joint DOFs.

Forces (torques) propagate *directly* through a joint from link *A* to link *B* along (about) *axes which are not joint DOFs*. Force (torque) propagation from link *A* to link *B* along (about) axes which are not joint DOFs is thus identical to force (torque) propagation though a rigid link. The force (torque) ${}^{B1}(f,\tau)_{Bi}$ propagated to link B along (about) joint non-DOF axis i is therefore equal to the force (torque) ${}^{B1}(f,\tau)_{Ai}$ acting on link *A* along (about) axis i:

$$ {}^{B1}(f,\tau)_{Bi} = {}^{B1}(f,\tau)_{Ai} \ . \qquad (3.3.17) $$

Forces (torques) propagate *indirectly* through a joint from link *A* to link *B* along (about) *axes which are joint DOFs*. Force (torque) propagation from link *A* to link *B* along (about) axes which are joint DOFs depends upon the dry friction characteristics of the joint, including: stiction, Coulomb friction, and rolling friction. In this section, we model these dry frictions. We show that the dry frictional force (torque) ${}^{B1}(f,\tau)_{Bi}$ propagated to link B along (about) joint DOF axis i is proportional to the normal force (torque) ${}^{B1}(f,\tau)_{Aj}$ acting on link *A* along (about) the joint non-DOF axis j:

$$ {}^{B1}(f,\tau)_{Bi} = \zeta_{ij} \ {}^{B1}(f,\tau)_{Aj} \ . \qquad (3.3.18) $$

The proportionality factor ζ_{ij} is a *nonlinear* function of the normal force (torque), the state of motion of the joint along (about) axis i (i.e., moving or not moving), and the sign of the velocity along (about) axis i. The proportionality factor ζ_{ij} differs for the three dry frictions: stiction, Coulomb friction and rolling friction. We indicate the type of dry friction by its subscript: ζ_{sij} for stiction, ζ_{cij} for Coulomb friction and ζ_{rij} for rolling friction.

Stiction is a coupling phenomenon which occurs at the onset of relative motion between two adjoining links *A* and *B*. We refer to the plane which is tangent to both links at the contact point (or line or surface of contact) as the *plane-of-contact*. If link *A* exerts an increasing force (torque) on link *B* parallel (perpendicular) to the plane-of-contact along the i-th axis, link *B* exerts increasing stictional forces (torques) on link *A* along the i-th axis to prevent relative motion between the two links until the stictional threshold force (torque) is exceeded. Once the stictional threshold force (torque) is exceeded, slip occurs, and there is relative motion between the two links. The stictional threshold for the i-th axis is the product of the stictional constant μ_{sABi} and the normal force (torque) along the j-th axis. Stiction does *not* affect links in relative motion. Even if the force (torque) applied to a moving link is reduced below the stictional threshold, forces (torques) are not coupled due to stiction.

We thus model the stictional force (torque) coupling proportionality factor between links *A* and *B* as:

$$\zeta_{sij} = \begin{cases} 1 & \text{if } i=j;\ {}^{B1}v_{A2i}=0 \text{ and } \left|{}^{B1}f_{Ai}\right| \le \mu_{sABi}\left|{}^{B1}f_{Aj}\right| \\ 0 & \text{otherwise .} \end{cases} \qquad (3.3.19)$$

Stiction thus couples forces/torques along a single axis (i=j). When the relative velocity ${}^{B1}v_{A2i}$ between the two links is zero and the magnitude of the force/torque ${}^{B1}f_{Ai}$ along the i-th axis is below the stictional threshold (i.e., the product of the stictional constant μ_{sABi} and the magnitude of the normal force/torque ${}^{B1}f_{Aj}$ along the j-th axis), the stictional force/torque equals the applied force/torque. When the magnitude of the force/torque along the i-th joint axis exceeds the stictional threshold, or when relative motion occurs, the stictional coefficient ζ_{sij} becomes zero and the stictional force/torque vanishes.

When the force/torque exerted by link *B* on link *A* along a joint axis exceeds the stictional threshold, slip occurs and *Coulomb frictional* force/torque coupling appears. We model the Coulomb frictional force/torque coupling along the i-th axis from link *B* to link *A* at the joint (according to Coulomb's Law) as opposing the relative motion with a magnitude proportional to the normal force in the j-th direction:

$$\zeta_{cij} = \begin{cases} -\mu_{cABij}\, \mathrm{sgn}({}^{B1}\mathbf{f}_{Aj}\, {}^{B1}\mathbf{v}_{A2i}) & \text{if } i \neq j \text{ and } {}^{B1}\mathbf{v}_{A2i} \neq 0 \\ 0 & \text{otherwise.} \end{cases} \qquad (3.3.20)$$

In (3.3.20), μ_{cABij} is a Coulomb frictional constant, and the scalar *signum* function sgn(x) denotes the algebraic sign of the scalar argument x. When there is a nonzero relative joint velocity ${}^{B1}\mathbf{v}_{A2i}$ between two links *A* and *B* along the i-th axis, a force/torque due to Coulomb friction appears as the negative of the product of the Coulomb frictional constant and the magnitude of the normal force. In (3.3.20), we express the sign of the Coulomb friction as the negative of the sign of the relative velocity.

Rolling friction occurs when link *A* (e.g., a wheel) is in rolling[4] contact with link *B*. Although two such rigid links are modeled ideally as having point contact, in actuality the area-of-contact is non-zero. Points on the surface of link *A* near the center of the area-of-contact must compress to meet the surface of link *B*, and points around the periphery of the area-of-contact must expand. This is a consequence of one or both of the links (e.g., a wheel or the surface-of-travel) being compliant (i.e., not strictly rigid) and is the source of rolling friction. The motion of the surface points within the area-of-contact as described above is subject to stictional and Coulomb frictional force/torque couplings. The net effect is a rolling frictional torque which we model approximately in (3.3.21) as having a magnitude proportional to the normal force on the rolling link and a direction opposing the direction-of-rotation of the rolling link.

$$\zeta_{rij} = \begin{cases} -\mu_{rABij}\, sgn({}^{B1}\mathbf{f}_{Aj}\, {}^{B1}\mathbf{v}_{A2i}) & \text{if the i-th axis is a rolling joint axis } {}^{B1}\mathbf{v}_{A2i} \neq 0 \text{ and } j \leq 3 < i \\ 0 & \text{otherwise.} \end{cases} \qquad (3.3.21)$$

In (3.3.21), μ_{rABij} is the rolling frictional constant which couples normal forces along the j-th axis to rolling frictional torques along the i-th axis. Rolling friction modeled in (3.3.21) is similar to Coulomb friction modeled in (3.3.20),

---

4 Rolling between two rigid links in relative motion occurs when the point-of-contact on one of the links is stationary relative to the point-of-contact on the other link.

except that rolling friction only couples normal *forces* to frictional *torques* (for $j \le 3 < i$); whereas Coulomb friction may couple normal forces or torques to frictional forces or torques. Both Coulomb and rolling friction may not occur simultaneously along the same joint axis because Coulomb friction appears when slip occurs, but slip cannot occur during rolling contact.

The torque caused by rolling friction is modeled conventionally in a coordinate system located at the *center-of-rotation*[5] of the rolling link (e.g., the axle of a wheel). We model rolling friction in (3.3.21) in the contact point coordinate system because the normal force which causes the rolling friction is applied at the contact point.

The dry friction modeled by the proportionality factor ζ_{ij} in (3.3.18) depends upon the axes i and j. For i=j, a non-zero factor ζ_{ij} models stiction; i.e., stiction couples forces (torques) along (about) axis i directly from link *A* to link *B*. For $i \neq j$, a non-zero factor ζ_{ij} may model Coulomb or rolling friction. Coulomb friction couples *normal* forces/torques on link *A* to forces/torques on link *B*. For $4 \le i \le 6$ and $1 \le j \le 3$, ζ_{ij} models rolling friction which couples *forces* on link *A* to *torques* on link *B*.

We combine the joint coupling characteristics along (about) all six axes to compute the six-vector force/torque ${}^{B1}\mathbf{f}_B$ propagated to link B from the six-vector force/torque ${}^{B1}\mathbf{f}_A$ applied by link A:

$$ {}^{B1}\mathbf{f}_B = {}^{B1}\mathbf{C}_{BA}\, {}^{B1}\mathbf{f}_A \,. \qquad (3.3.22) $$

In (3.3.22), the (6×6) *coupling matrix* ${}^{B1}\mathbf{C}_{BA} = [k_{ij}]$ from link *A* to link *B* at coordinate system B_1 consists of the components:

$$ k_{ij} = \begin{cases} 1 & \text{if i=j and the i-th axis is not a joint DOF} \\ \zeta_{ij} & \text{otherwise.} \end{cases} \qquad (3.3.23) $$

The joint coupling matrices of practical joints are *sparse* for three reasons: (1) Some of the dry frictional coefficients ζ_{ij} are negligible; (2) All of the

---

5 The center-of-rotation of a moving link is the point fixed to the link which is stationary relative to a reference coordinate system.

elements in each row corresponding to a joint non-DOF axis are zero, except the diagonal element which is unity; and (3) All of the elements in each column corresponding to a joint DOF axis are zero, except the diagonal element which is ζ_{sij}. The second reason is evident because no dry frictional forces (torques) act along (about) an axis for which no motion may occur; forces (torques) along (about) these axes propagate directly. The third reason is evident because axes which do not propagate forces/torques directly cannot act as normal forces/torques for propagating dry frictional forces/torques. For Uranus (described in Chapter 6), as for typical robotic mechanisms, the densest coupling matrix contains five non-zero dry frictional coefficients and three unity coefficients (i.e., the densest coupling matrix is 8/36 or twenty-two percent full).

From the definitions of the frictional coefficients in (3.3.21), (3.3.22) and (3.3.23), $\zeta_{ijAB}=\zeta_{ijBA}$ for stictional forces/torques and $\zeta_{ijAB}=-\zeta_{ijBA}$ for Coulomb and rolling frictional forces/torques. The joint coupling matrix ${}^{B1}\mathbf{C}_{AB}$ from link *B* to link *A* is thus computed from the joint coupling matrix ${}^{B1}\mathbf{C}_{BA}$ from link *A* to link *B* by inverting the off-diagonal elements:

$$ {}^{B1}\mathbf{C}_{AB} = {}^{B1}\mathbf{C}_{BA} \bullet \mathbf{C}_{mask} \tag{3.3.24} $$

where the $\bullet$ operator multiplies conformable matrices *element-by-element*, and the (6×6) coupling mask $\mathbf{C}_{mask}$ is:

$$ \mathbf{C}_{mask} = \begin{pmatrix} 1 & -1 & -1 & -1 & -1 & -1 \\ -1 & 1 & -1 & -1 & -1 & -1 \\ -1 & -1 & 1 & -1 & -1 & -1 \\ -1 & -1 & -1 & 1 & -1 & -1 \\ -1 & -1 & -1 & -1 & 1 & -1 \\ -1 & -1 & -1 & -1 & -1 & 1 \end{pmatrix}. \tag{3.3.25} $$

The joint coupling matrix computation in (3.3.24) is required in Section 3.3.4 to generate the force/torque equations-of-motion.

We focus on the joint between links *A* an *B* shown in Figure 3.3 to provide an intuitive understanding of frictional coupling at a joint.

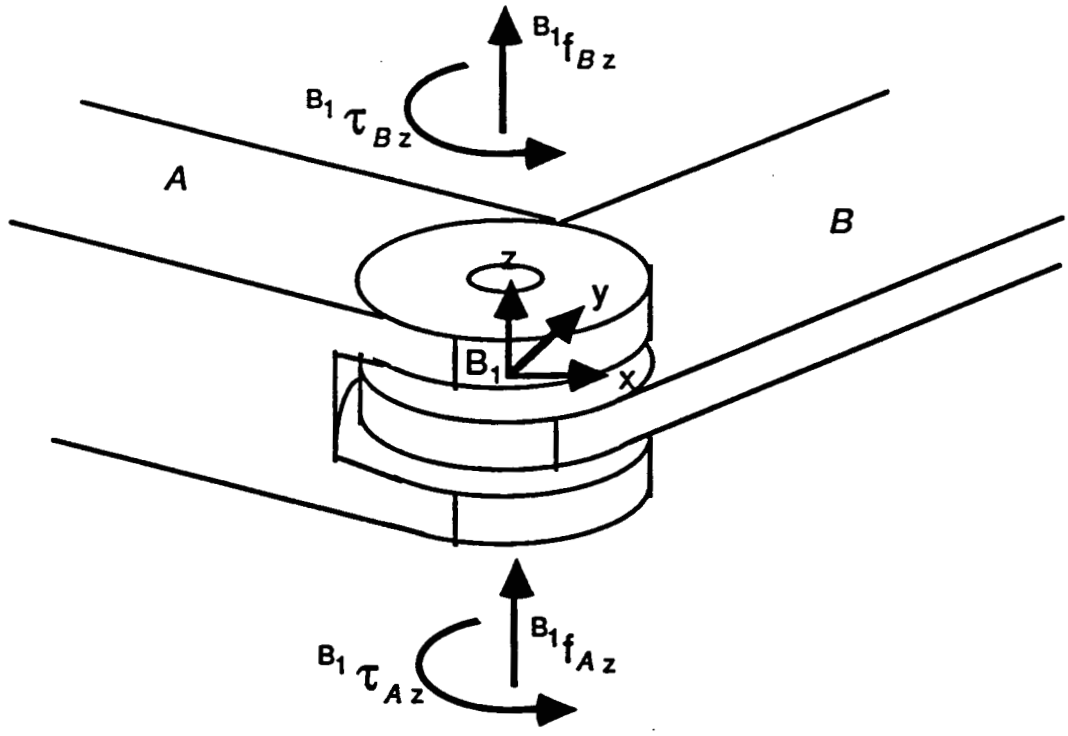


Figure 3.3: The Joint Between Links *A* and *B*

In Figure 3.3, coordinate system B_1 is assigned with its z-axis aligned with the joint axis between links *A* and *B*. The force/torque vector propagated to link *B* is computed from the joint coupling matrix and the force/torque vector applied to link *A* according to:

$$\begin{pmatrix} {}^{B_1}f_{Bx} \\ {}^{B_1}f_{By} \\ {}^{B_1}f_{Bz} \\ {}^{B_1}\tau_{Bx} \\ {}^{B_1}\tau_{By} \\ {}^{B_1}\tau_{Bz} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & \zeta_c & 0 & 0 & \zeta_s \end{pmatrix} \begin{pmatrix} {}^{B_1}f_{Ax} \\ {}^{B_1}f_{Ay} \\ {}^{B_1}f_{Az} \\ {}^{B_1}\tau_{Ax} \\ {}^{B_1}\tau_{Ay} \\ {}^{B_1}\tau_{Az} \end{pmatrix} \qquad (3.3.26)$$

where:

$$\zeta_s = \begin{cases} 1 & \text{if } {}^{B_1}\omega_{A_2z}=0; \\ & \text{and } |{}^{B_1}\tau_{Az}| \le \mu_s |{}^{B_1}f_{Az}| \\ 0 & \text{otherwise} \end{cases} \qquad \zeta_c = \begin{cases} -\mu_c \text{sgn}({}^{B_1}\omega_{A_2z}\, {}^{B_1}f_{Az}) & \text{if } {}^{B_1}\omega_{A_2z} \neq 0 \\ 0 & \text{otherwise .} \end{cases}$$

We make the following correspondences between Figure 3.3 and (3.3.26). The scalar force ${}^{B_1}f_{Az}$ applied to link *A* along the z-axis (a non-DOF axis) of coordinate system B_1 produces an equal force ${}^{B_1}f_{Bz}$ on link *B* along the z-axis of coordinate system B_1. This direct force propagation is denoted in (3.3.26) by the unit element in the diagonal (3,3) position of the coupling matrix. The scalar torque ${}^{B_1}\tau_{Az}$ applied to link *A* about the z-axis (a DOF axis) of coordinate system B_1 may or may not produce a torque ${}^{B_1}\tau_{Bz}$ on link *B* about the z-axis of coordinate system B_1 depending upon the frictional characteristics of the joint. The stictional coefficient ζ_S, the (6,6) diagonal element of the coupling matrix, models the stictional characteristics of the joint and the Coulomb coefficient ζ_C, the (6,3) off-diagonal element of the coupling matrix, models the Coulomb frictional characteristics of the joint. The joint coupling matrices for two prototype WMRs are formulated in Chapters 6 and 7.

3.3.3.4 The Propagation Matrix

By cascading alternately the link and joint force/torque propagations formulated in Sections 3.3.3.2 and 3.3.3.3, respectively, we may propagate a force/torque vector ${}^{A_1}\mathbf{f}_A$ in coordinate system A_1 to the force/torque vector ${}^{M(M)}\mathbf{f}_M$ in coordinate system M(*M*) in the open-chain depicted in Figure 3.1:

$$ {}^{M(M)}\mathbf{f}_M = \left[{}^{M_1}\mathbf{L}^T_{M(M)} \, {}^{M_1}\mathbf{C}_{ML} \cdots {}^{B_1}\mathbf{L}^T_{C_1} \, {}^{B_1}\mathbf{C}_{BA} \, {}^{A_1}\mathbf{L}^T_{B_1} \right] {}^{A_1}\mathbf{f}_A . \qquad (3.3.27) $$

We call the cascade of the transposed link Jacobian and joint coupling matrices in (3.3.27) the *propagation matrix*:

$$ {}^{M(M)}\mathbf{P}_{A_1} = \left[{}^{M_1}\mathbf{L}^T_{M(M)} \, {}^{M_1}\mathbf{C}_{ML} \cdots {}^{B_1}\mathbf{L}^T_{C_1} \, {}^{B_1}\mathbf{C}_{BA} \, {}^{A_1}\mathbf{L}^T_{B_1} \right] . \qquad (3.3.28) $$

The force/torque propagation matrix in (3.3.28) provides a conceptually simple algorithm for computing the effects on link *M* of a force/torque acting on link *A* when the two links are separated by an open-chain of links and joints.

3.3.3.5 The Link Gravitational Propagation Matrix

All of the natural coordinate systems for the force/torque sources described in Section 3.3.2 are fixed rigidly to links within the robotic mechanism except for the link gravitational coordinate systems required to model gravitational forces. The link gravitational coordinate system for each link is assigned with the z-axis opposing the gravitational field, irrespective of the orientation of the link. In this section, we develop a special algorithm for computing the propagation matrix ${}^{M(L)}\mathbf{P}_{G(L)}$ from the link gravitational coordinate system G(L) of link L to the center-of-mass coordinate system M(L) of the link. With this algorithm, we may then propagate the gravitational forces from the center-of-mass coordinate system to any other coordinate system within the robotic mechanism by cascading transposed link Jacobian and joint coupling matrices as illustrated in Section 3.3.3.4.

To proceed with our development, we must assign the absolute gravitational coordinate system G, with its origin coincident with the origin of the floor coordinate system F (assigned in Table 2.2) and its z-axis opposed to the gravitational field. The origin of the link gravitational coordinate system G(L) thus coincides with the origin of the center-of-mass coordinate system M(L) according to our coordinate system assignments in Section 3.3.2.3. The propagation of gravitational forces from the link gravitational coordinate system G(L) to the center-of-mass coordinate system M(L) is thus dependent upon the relative orientation but independent of the relative *position* of the G(L) and M(L) coordinate systems because of the structure of the transposed link Jacobian matrix (as described in Section 3.3.3.2). The z-axes of the absolute gravitational coordinate system G and the link gravitational coordinate system G(L) are parallel. Propagating gravitational forces from the link gravitational coordinate system G(L) to the center-of-mass coordinate system M(L) is equivalent to propagating gravitational forces from the absolute gravitational coordinate system to the center-of-mass coordinate system if: (1) we ignore the intervening joints between the absolute gravitational coordinate system G and the center-of-mass coordinate system M(L); and (2) zero the torques propagated to the center-of-mass coordinate system M(L) due to the positional displacement of the origin of the absolute gravitational coordinate system G from the origin of the center-of-mass coordinate system M(L). We thus find that the task of

propagating gravitational forces from the link gravitational coordinate system G(L) to the center-of-mass coordinate system M(L) is identical to the task of propagating the gravitational forces from the absolute gravitational coordinate system G to the center-of-mass coordinate system M(L) ignoring the intervening joints and zeroing the positional displacement between these two coordinate systems. The propagation matrix from the link gravitational coordinate system G(L) to the center-of-mass coordinate system M(L) is thus computed by cascading the transposed link Jacobian matrices from the absolute gravitational coordinate system G to the center-of-mass coordinate system M(L), ignoring the intervening joint coupling matrices, and then zeroing the lower-left (3x3) submatrix of the result (to eliminate the effects of the positional displacement):

$$
\begin{aligned}
{}^{M(L)}\mathbf{P}_{G(L)} &= \left[{}^{L_1}\mathbf{L}^{T}_{M(L)}\ {}^{K_1}\mathbf{L}^{T}_{L_1} \cdots {}^{B_1}\mathbf{L}^{T}_{C_1}\ {}^{A_1}\mathbf{L}^{T}_{B_1}\ {}^{G}\mathbf{L}^{T}_{A_1} \right] \bullet \mathbf{P}_{mask} \\
&= \left[{}^{M(L)}\mathbf{L}_{L_1}\ {}^{L_1}\mathbf{L}_{K_1} \cdots {}^{C_1}\mathbf{L}_{B_1}\ {}^{B_1}\mathbf{L}_{A_1}\ {}^{A_1}\mathbf{L}_{G} \right] \bullet \mathbf{P}_{mask} \,. \qquad (3.3.29)
\end{aligned}
$$

In (3.3.29), the $\bullet$ operator multiplies the matrices element-wise, and the (6x6) propagation mask is:

$$
\mathbf{P}_{mask} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}. \qquad (3.3.30)
$$

3.3.4 Matrix-Vector Force/Torque Equations-of-Motion

3.3.4.1 Foundations

Newton's translational and rotational equilibrium laws[6] are the foundations for formulating our force/torque equations-of-motion of the robotic mechanism in Figure 1.1. In Section 3.3.2, we applied D'Alembert's principle to

---

6 Newton's equilibrium law for one-dimensional translational motion is $f_{net}=ma$ where f_{net} is the algebraic sum of the forces acting on the link, m is the mass of the link, and a is the acceleration of the link.

model the dynamic forces/torques as static forces/torques. We thus formulate the force/torque equations-of-motion by equating to zero the sum of the six-vector forces/torques exerted by all sources s on the robotic mechanism within the common coordinate system M(*M*):

$$\sum_{s} {}^{M(M)}\mathbf{f}_{s} = \mathbf{0} \ . \qquad (3.3.31)$$

To apply the equilibrium law in (3.3.31), we require the force/torque models in Section 3.3.2 for each source s: inertial (i), gravitational (g), actuation (a), viscous friction (v), and environmental contact (e); and the force/torque propagation methodology in Section 3.3.3 to propagate the forces/torques from their natural coordinate systems to the common M(*M*) coordinate system.

In Section 3.3.4.2, we apply the equilibrium law in (3.3.31) at the center-of-mass coordinate system M(*M*) of the main link of the robotic mechanism and thereby obtain six *primary* force/torque equations-of-motion. In Section 3.3.4.3, we compute the internal joint forces/torques. In Section 3.3.4.4, we apply the internal forces/torques to compute the *secondary* force/torque equations-of-motion expressing the joint constraints. The primary and secondary force/torque equations-of-motion form the basis of the dynamic model of any simple closed-chain robotic mechanism.

3.3.4.2 Primary Force/Torque Equations-of-Motion

Our objective is to model the motion of the *main link M* of the robotic mechanism in Figure 1.1. We thus select the center-of-mass coordinate system M(*M*) of the main link as the common coordinate system for applying the equilibrium law in (3.3.27). The primary force/torque equations-of-motion are formulated by straightforward application of the equilibrium law:

$$\sum_{X} \sum_{s} \left({}^{M(M)}\mathbf{P}_{N(s,X)} \, {}^{N(s,X)}\mathbf{f}_{sX} \right) = \mathbf{0} \ . \qquad (3.3.32)$$

In (3.3.32), the outer summation extends over all links *X* within the mechanism, the inner summation extends over all sources of forces/torques (i.e., s = i, g, a, v, and e), and N(s,*X*) is the natural coordinate system for source s on link *X*. The matrix-vector models of each force/torque ${}^{N(s,X)}\mathbf{f}_{sX}$ are written

directly (according to our formulation in Section 3.3.2), and each propagation matrix ${}^{M(M)}\mathbf{P}_{N(s,X)}$ is computed according to Section 3.3.3.

3.3.4.3 Internal Force/Torque Computations

The dry frictional proportionality factors introduced in Section 3.3.3.3, which depend upon the internal forces/torques between adjacent links in the mechanism, are required for computing the propagation matrices introduced in Section 3.3.3.4, and the force/torque equations-of-motion in Sections 3.3.4.2 and 3.3.4.4. In this section, we apply our matrix-vector force/torque propagation methodology to the *recursive Newton-Euler dynamics formulation* [Luh80b] to compute the internal forces/torques. The recursive Newton-Euler dynamics formulation computes recursively the net forces/torques on each link of an open-chain robotic manipulator from the end-effector to the base.

We apply this procedure to the open-chain in Figure 3.1 which represents one of the N open-chains within the simple closed-chain robotic mechanism in Figure 1.1 and thereby compute recursively the net forces/torques on each link from the environment *E* to the main link *M*. The *net* forces/torques on each link are then identified as the *internal* forces/torques. We compute the internal forces/torques ${}^{B_1}\mathbf{f}_A$ on link *A* in the joint coordinate system B_1 as the sum of the inertial, gravitational, actuator, viscous frictional and environmental contact forces/torques (i.e., s=i,g,a,v, and e) acting on link *A* propagated from their natural coordinate systems N(s,*A*) to the B_1 coordinate system:

$$ {}^{B_1}\mathbf{f}_A = {}^{A_2}\mathbf{L}_{B_1}^{T} \sum_{s} {}^{A_2}\mathbf{P}_{N(s,A)}\, {}^{N(s,A)}\mathbf{f}_{sA} \,. \qquad (3.3.33) $$

The internal forces/torques ${}^{C_1}\mathbf{f}_B$ on link *B* in the joint coordinate system C_1 are computed similarly as the sum of all forces/torques (i.e., s=i,g,a,v, and e) acting on bodies *A* and *B* propagated from their natural coordinate systems to the C_1 coordinate system:

$$ {}^{C_1}\mathbf{f}_B = {}^{B_2}\mathbf{L}_{C_1}^{T} \sum_{s} \left({}^{B_2}\mathbf{P}_{N(s,A)}\, {}^{N(s,A)}\mathbf{f}_{sA} + {}^{B_2}\mathbf{P}_{N(s,B)}\, {}^{N(s,B)}\mathbf{f}_{sB} \right) . \qquad (3.3.34) $$

To reduce the computational requirements, the internal force/torque computation in (3.3.34) may be performed recursively as:

$$ {}^{C_1}\mathbf{f}_B = {}^{B_2}\mathbf{L}_{C_1}^{T} \left\{ {}^{B_2}\mathbf{P}_{B_1} \, {}^{B_1}\mathbf{f}_A + \sum_{s} \left({}^{B_2}\mathbf{P}_{N(s,B)} \, {}^{N(s,B)}\mathbf{f}_{sB} \right) \right\} . \qquad (3.3.35) $$

By applying the internal forces/torques ${}^{B_1}\mathbf{f}_A$ at the previous joint coordinate system B_1 to the computation of the internal forces/torques at the present joint coordinate system C_1, we may thus compute recursively all of the internal forces/torques that are required for the dry friction models in the open-chain.

3.3.4.4 Secondary Force/Torque Equations-of-Motion

The propagation of forces/torques through a joint from link A to link B is modeled in (3.3.22). Since the assignment of bodies *A* and *B* in (3.3.22) is arbitrary, forces/torques propagate by the same model from link *B* to link *A*:

$$ {}^{B_1}\mathbf{f}_A = {}^{B_1}\mathbf{C}_{AB} \, {}^{B_1}\mathbf{f}_B . \qquad (3.3.36) $$

where the coupling matrix ${}^{B_1}\mathbf{C}_{AB}$ is computed by negating the off-diagonal elements of the coupling matrix ${}^{B_1}\mathbf{C}_{BA}$ as discussed in Section 3.3.3.3. We substitute (3.3.36) into (3.3.22) to obtain the six-vector force/torque equation-of-motion at the joint:

$$ \left[\mathbf{I} - {}^{B_1}\mathbf{C}_{BA} \, {}^{B_1}\mathbf{C}_{AB} \right] {}^{B_1}\mathbf{f}_B = \mathbf{0} , \qquad (3.3.37) $$

where $\mathbf{I}$ is the (6×6) identity matrix, $\mathbf{0}$ is the zero six-vector, and the internal force/torque vector ${}^{B_1}\mathbf{f}_B$ is computed according to the recursive procedure outlined in Section 3.3.4.3. Of the six equations in (3.3.37), we obtain $(6\text{-}d_{AB})$ null equations (i.e., equations that reduce to 0=0) and d_{AB} *secondary force/torque equations-of-motion,* where d_{AB} is the number of DOFs of the joint between links *A* and *B*. We formulate (3.3.37) for all joints between all pairs of contacting links *A* and *B* in the robotic mechanism to obtain a set of $N_s = \sum_{(AB)} d_{AB}$ secondary force/torque equations-of-motion.

3.3.5 Concluding Remarks

Our dynamic modeling framework has several features which make it an attractive choice for the dynamic modeling of multi-link robotic mechanisms. The formulation allows the modeling of a broad range of physical phenomena including inertial, gravitational, actuation, viscous frictional and environmental contact forces/torques and dry friction couplings. The extension of the formulation to model such phenomena as flexibility is introduced in Appendix 13. The dynamic modeling methodology applies to robotic mechanisms containing a wide variety of structural characteristics including simple closed-chains, and lower and higher-pairs. Consequently, the dynamic modeling formulation applies to a spectrum of conventional robotic mechanisms such as stationary manipulators, multi-manipulators, WMRs, legged robots and robotic hands. Future extensions may enable the modeling of mechanical systems with complex closed-chains.

The dynamic modeling formulation is *modular* such that the force/torque equations-of-motion (in this section) and the kinematic transformations (in Section 3.4) may be formulated independently. These submodules are, in turn, built up from a foundation of modular, conceptually simple, technically proven component models including: Newton's laws, Euler's equations, Coulomb's law, and the link Jacobian matrix. The accuracy and applicability of the dynamic model to a specific robotic mechanism is ensured by the accuracy and applicability of these component models.

Our dynamic modeling framework provides insights into mechanical system dynamics. The formulation emphasizes the duality between forces/torques and linear/angular velocities. The link Jacobian matrix transforms velocities between two coordinate systems while the transpose of the link Jacobian matrix propagates forces/torques between the same two coordinate systems. We have also shown that forces/torques propagate through a joint *directly* along the joint *non-DOF axes*, but *indirectly* along joint *DOF axes* (i.e., those axes which support velocities). One implication to be drawn from this duality of forces/torques and linear/angular velocities is that the independent control of both force (torque) and linear (angular) velocity along (about) the same axis is impossible.

We have recognized the nature of dry friction as a *coupling* phenomena, in contrast to the conventional view of dry friction as a force/torque source. We have introduced the concept of force/torque *propagation*; i.e., computing the effect of a force/torque which originates in one coordinate system within the mechanism at another coordinate system by propagating the force/torque through the intervening links and joints. The propagation of dynamic forces/torques as static forces/torques reinforces D'Alembert's principle. Specifically, the propagation of forces/torques requires only the *positions* of the mechanical system components. Force/torque modeling is thus accomplished at any instant by taking a *snapshot* of the present system configuration and propagating forces/torques through it. We have introduced the paradigm of a dynamic robotic mechanism at any instant as consisting of a massless structural frame and a set of forces/torques acting upon it at disparate points. The modularity of the dynamic modeling framework thus improves our understanding by partitioning the complex dynamic model of a robotic mechanism into independently formulated force/torque, kinematic, and dry friction models which are easy to comprehend separately.

The matrix-vector structure of the dynamic modeling computations provides several advantages. Because matrix-vector and matrix-matrix multiplications are representable by *directed graphs*, we present in Appendix 14 the dynamic model of a robotic mechanism in a visualizable form to enhance understanding. A graph representation of the dynamic system allows the application of *graph-theoretic techniques* to the analysis of robotic mechanisms. The graphical representation emphasizes the *parallelism* in the computations of the dynamic model. The inherent parallelism in the dynamic model suggests that dynamics-based servo-control and simulation of complex robotic mechanisms is possible in real-time through the application of *array processors*. In fact, realizing the equivalence of the graphical representation of robotic mechanisms and neural networks [Lippman87] introduces the possibility of executing dynamics-based servo-control and simulations of robotic mechanisms at the speed of light on existing or future analog *neural networks*.

3.4 Companion Kinematic Transformations

3.4.1 Overview

In Section 3.3.2, we modeled the force/torque sources in their natural coordinate systems. The resulting dynamic force/torque equations-of-motion (formulated in Section 3.3.4) are dependent upon velocities and accelerations of robotic links in instantaneously coincident coordinate systems which are neither specified nor sensed in actual implementations. The force/torque equations-of-motion developed in Section 3.3.4 are thereby dependent on the actuator forces/torques τ_a, the environmental contact forces/torques $\mathbf{f}_e$, and the velocities ${}^{\overline{X}}\mathbf{v}_X$ and accelerations ${}^{\overline{X}}\mathbf{a}_X$ of all of the links of the robotic mechanism:

$$\mathbf{f}_1(\tau_a, \mathbf{f}_e, {}^{\overline{X}}\mathbf{a}_X, {}^{\overline{X}}\mathbf{v}_X) = \mathbf{0}. \tag{3.4.1}$$

We compute in this section the unknown and unsensed velocities ${}^{\overline{X}}\mathbf{v}_X$ and accelerations ${}^{\overline{X}}\mathbf{a}_X$ from known or sensed velocities and accelerations.

We conceptualize a robotic mechanism as the simple closed-chain shown in Figure 1.1 consisting of a main link and N open-chains each having several links. We are required to compute only the open-chain link velocities ${}^{\overline{X}}\mathbf{v}_X$ and accelerations ${}^{\overline{X}}\mathbf{a}_X$ since the velocities ${}^{\overline{M(M)}}\mathbf{v}_{M(M)}$ and accelerations ${}^{\overline{M(M)}}\mathbf{a}_{M(M)}$ of the main link are either specified (as in the actuated inverse dynamic solution) or computed (as in the forward dynamic solution). Our goals in this section are then to compute the velocities and accelerations of the open-chain links from the velocities and accelerations of the main link and the sensed open-chain joint velocities.

In Section 3.4.2, we compute the velocities ${}^{\overline{X}}\mathbf{v}_X$ of each open-chain link *from* the velocities ${}^{\overline{M(M)}}\mathbf{v}_{M(M)}$ of main link and the velocities of the intervening joints ${}^{Y}\mathbf{v}_X$:

$${}^{\overline{X}}\mathbf{v}_X = \mathbf{f}_2({}^{\overline{M(M)}}\mathbf{v}_{M(M)}, {}^{Y}\mathbf{v}_X). \tag{3.4.2}$$

The velocities of the intervening joints are computed from the velocities of the WMR main link coordinate system by applying the actuated inverse velocity solution in (2.4.8) and assuming that all of the joints are actuated:

$$^{Y}\mathbf{v}_{X} = \mathbf{f}_3(^{\overline{M(M)}}\mathbf{v}_{M(M)}) . \tag{3.4.3}$$

The velocities of the WMR main link are then computed from the sensed joint velocities by applying the sensed forward velocity solution in (2.4.14):

$$^{\overline{M(M)}}\mathbf{v}_{M(M)} = \mathbf{f}_4(\mathbf{v}_s) . \tag{3.4.4}$$

In Section 3.4.3, we compute the accelerations of the links $^{\overline{X}}\mathbf{a}_{X}$ from the velocities and accelerations of main link and the velocities and accelerations of the intervening joints and the velocities of the intervening links:

$$^{\overline{X}}\mathbf{a}_{X} = \mathbf{f}_5(^{\overline{M(M)}}\mathbf{a}_{M(M)}, {}^{Y}\mathbf{a}_{X}, {}^{Y}\mathbf{v}_{X}, {}^{\overline{X}}\mathbf{v}_{X}) . \tag{3.4.5}$$

The accelerations of the joints $^{Y}\mathbf{a}_{X}$ are computed by the inverse acceleration solution in analogy with the inverse velocity solution in (3.4.3):

$$^{Y}\mathbf{a}_{X} = \mathbf{f}_6(^{\overline{M(M)}}\mathbf{a}_{M(M)}) . \tag{3.4.6}$$

By substituting the individual kinematic transformations $\mathbf{f}_2$ to $\mathbf{f}_6$ in (3.4.2) to (3.4.6) into the force/torque equations-of-motion in (3.4.1), we formulate the dynamic equations-of-motion (i.e., the dynamic model) of the robotic mechanism which is directly applicable to servo-control and simulation applications:

$$\mathbf{f}_7(\tau_a, \mathbf{f}_e, \mathbf{v}_s, {}^{\overline{M(M)}}\mathbf{a}_{M(M)}) = \mathbf{0} . \tag{3.4.7}$$

3.4.2 Six-Vector Velocity Transformations

We consider the open-chain of rigid links in Figure 3.1 to develop a velocity transformation methodology. Our goal is to compute the velocities $^{\overline{A_1}}\mathbf{v}_{A_1}$ of coordinate system A_1 from the velocities $^{\overline{M(M)}}\mathbf{v}_{M(M)}$ of the main link and the

velocities ${}^{Y}\mathbf{v}_{X}$ of the intervening joints for X,Y = A_2,B_1; B_2,C_1; ...; L_2,M_1. To solve this *velocity transformation* problem, we first consider the simpler problem of transforming velocities from coordinate system B_1 to coordinate system A_2. We thus compute the velocities ${}^{\overline{A_2}}\mathbf{v}_{A_2}$ of coordinate system A_2 from the velocities ${}^{\overline{B_1}}\mathbf{v}_{B_1}$ of coordinate system B_1 and the velocities ${}^{B_1}\mathbf{v}_{A_2}$ of the joint between bodies *A* and *B*. In Figure 3.4, we illustrate the three translational joint displacements ${}^{B_2}\mathbf{p}_{I_1}=[d_x\ 0\ 0]^T$, ${}^{I_1}\mathbf{p}_{I_2}=[0\ d_y\ 0]^T$, ${}^{I_2}\mathbf{p}_{I_3}=[0\ 0\ d_z]^T$, and the three rotational joint displacements ${}^{I_3}\theta_{I_4}=[\theta_x\ 0\ 0]^T$, ${}^{I_4}\theta_{I_5}=[0\ \theta_y\ 0]^T$, ${}^{I_5}\theta_{A_2}=[0\ 0\ \theta_z]^T$ between coordinate systems B_2 and A_1 according to the roll-pitch-yaw convention [Paul81a].

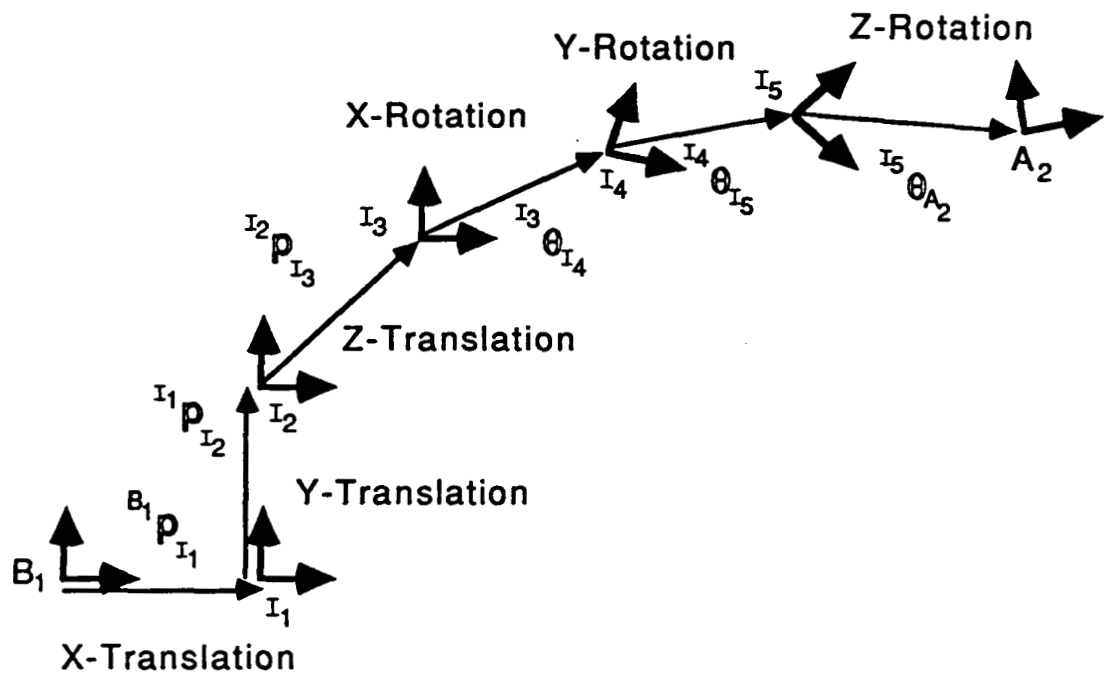


Figure 3.4: Intermediate Coordinate Systems for Six Dimensional Motion

In Figure 3.4, the five intermediate coordinate systems I_1 to I_5 decompose the six dimensional joint transformation into six independent one dimensional transformations. The origins of the final four coordinate systems I_3, I_4, I_5 and A_2, are coincident; for clarity, we have displaced them in Figure 3.4. The link coordinate system B_1 and each of the six joint displacements may undergo motions which contribute to the motion of the A_2 coordinate system. In

this section, we first transform the translational velocities, then the rotational velocities, and finally we combine the transformed translational and rotational velocities into a six-vector.

The translational velocities ${}^{\bar{B}_1}v_{B_1}$ of coordinate system B_1 and the three translational joint velocities ${}^{B_1}v_{A_{2x}}=\dot{d}_x$, ${}^{B_1}v_{A_{2y}}=\dot{d}_y$, and ${}^{B_1}v_{A_{2z}}=\dot{d}_z$ contribute to the translational velocity of coordinate system A_2. Multipling by the (3×3) rotational matrix ${}^{\bar{A}_2}R_{\bar{O}}$ for $O=B_1$, I_1, and I_2 rotates each of these translational velocities to the A_2 coordinate system in (3.4.8).

The rotational velocities ${}^{\bar{B}_1}\omega_{B_1}$ of coordinate system B_1 and the three rotational joint velocities ${}^{B_1}\omega_{A_{2x}}=\dot{\theta}_x$, ${}^{B_1}\omega_{A_{2y}}=\dot{\theta}_y$, ${}^{B_1}\omega_{A_{2z}}=\dot{\theta}_z$ also contribute to the translational velocity of the coordinate system A_2. Each of these rotational velocities is crossed with the three-vector displacement ${}^{O}p_{A_2}$ of the O coordinate system from the A_2 coordinate system (i.e., the moment arm), and then rotated to the A_2 coordinate system by multiplying by the rotation matrix ${}^{\bar{A}_2}R_{\bar{O}}$ for $O=I_3$, I_4, and I_5.

The translational velocity of coordinate system A_2 is thus the superposition of the translational and rotational velocity contributions:

$$
\begin{aligned}
{}^{\bar{A}_2}v_{A_2} &= {}^{\bar{A}_2}R_{\bar{B}_1}\,[{}^{\bar{B}_1}v_{B_1}+({}^{\bar{B}_1}\omega_{B_1}\times{}^{B_1}p_{A_2})] + {}^{\bar{A}_2}R_{\bar{I}_1}\,{}^{B_1}v_{I_1} + {}^{\bar{A}_2}R_{\bar{I}_2}\,{}^{I_1}v_{I_2} + {}^{\bar{A}_2}R_{\bar{I}_3}\,{}^{I_2}v_{I_3} \\
&\quad + {}^{\bar{A}_2}R_{\bar{I}_3}\,({}^{I_3}\omega_{I_4}\times{}^{I_3}p_{A_2}) + {}^{\bar{A}_2}R_{\bar{I}_4}\,({}^{I_4}\omega_{I_5}\times{}^{I_4}p_{A_2}) + {}^{\bar{A}_2}R_{\bar{I}_5}\,({}^{I_5}\omega_{A_2}\times{}^{I_5}p_{A_2})\,. \qquad (3.4.8)
\end{aligned}
$$

Since the position vectors ${}^{I_3}p_{A_2}$, ${}^{I_4}p_{A_2}$, and ${}^{I_5}p_{A_2}$ in Figure 3.4 are zero, and the rotation matrices ${}^{\bar{A}_2}R_{\bar{I}_1}$, ${}^{\bar{A}_2}R_{\bar{I}_2}$, and ${}^{\bar{A}_2}R_{\bar{I}_3}$ are equal, we simplify (3.4.8) to:

$$
{}^{\bar{A}_2}v_{A_2} = {}^{\bar{A}_2}R_{\bar{B}_1}\,[{}^{\bar{B}_1}v_{B_1}+({}^{\bar{B}_1}\omega_{B_1}\times{}^{B_1}p_{A_2})] + {}^{\bar{A}_2}R_{\bar{B}_1}\,{}^{B_1}v_{A_2}\,, \qquad (3.4.9)
$$

where the translational joint velocity three-vector ${}^{B_1}v_{A_2}=({}^{B_1}v_{I_1}+{}^{I_1}v_{I_2}+{}^{I_2}v_{I_3})=[\dot{d}_x\ \dot{d}_y\ \dot{d}_z]^T$. The translational link velocities ${}^{\bar{B}_1}v_{B_1}$ and the translational joint velocities ${}^{B_1}v_{A_2}$ do not contribute to the rotational velocity ${}^{\bar{A}_2}\omega_{A_2}$ of coordinate system A_2. The rotational velocities ${}^{\bar{B}_1}\omega_{B_1}$ of coordinate system B_1 and the three rotational

joint velocities ${}^{B_1}\omega_{A_{2x}}$, ${}^{B_1}\omega_{A_{2y}}$, and ${}^{B_1}\omega_{A_{2z}}$, each multiplied by the appropriate rotational matrix, superimpose to create the rotational velocity ${}^{\bar{A}_2}\omega_{A_2}$ of coordinate system A_2:

$$ {}^{\bar{A}_2}\omega_{A_2} = {}^{\bar{A}_2}R_{\bar{B}_1}\ {}^{\bar{B}_1}\omega_{B_1} + {}^{\bar{A}_2}R_{\bar{I}_3}\ {}^{I_3}\omega_{I_4} + {}^{\bar{A}_2}R_{\bar{I}_4}\ {}^{I_4}\omega_{I_5} + {}^{\bar{A}_2}R_{\bar{I}_5}\ {}^{I_5}\omega_{A_2}. \qquad (3.4.10) $$

In parallel with the translational joint velocities, we combine the rotational joint velocities into the rotational joint velocity three-vector ${}^{B_1}\omega_{A_2}$ and rewrite (3.4.10) as

$$ {}^{\bar{A}_2}\omega_{A_2} = {}^{\bar{A}_2}R_{\bar{B}_1}\ {}^{\bar{B}_1}\omega_{B_1} + {}^{\bar{A}_2}R_{*}\ {}^{B_1}\omega_{A_2}, \qquad (3.4.11) $$

where the rotational matrix ${}^{\bar{A}_2}R_*$ is formed by zeroing the rotations θ_y and θ_z in the first column and θ_z in the second column of ${}^{\bar{A}_2}R_{\bar{B}_1}$. In (3.4.9) and (3.4.11), we compute the translational and rotational velocities of coordinate system A_2 from the translational and rotational velocities of coordinate system B_1 and the intervening joints. These linear three-vector relationships may be written compactly as the six-vector:

$$ {}^{\bar{A}_2}V_{A_2} = {}^{A_2}L_{B_1}\ {}^{\bar{B}_1}V_{B_1} + {}^{A_2}J_{B_1}\ {}^{B_1}V_{A_2}. \qquad (3.4.12) $$

In (3.4.12), the (6x6) *link Jacobian* matrix ${}^{A_2}L_{B_1}$ is the matrix coefficient of the link velocities ${}^{\bar{B}_1}V_{B_1}$ and the (6x6) *joint Jacobian* matrix ${}^{A_2}J_{B_1}$ is the matrix coefficient of the joint velocities ${}^{B_1}V_{A_2}$. In Appendix 12, we expand both the link Jacobian ${}^{A_2}L_{B_1}$ and joint Jacobian ${}^{A_2}J_{B_1}$ matrices into scalar components which are nonlinear (trigonometric) functions of the position six-vector ${}^{B_1}p_{A_2}$. We apply the transpose of the link Jacobian matrix in Section 3.3.3.2 to propagate forces/torques between coordinate systems.

We apply the velocity transformation in (3.4.8) to transform velocities in coordinate system A_2 to coordinate system A_1 by allowing coordinate systems A_1 and A_2 to play the roles in (3.4.12) of coordinate systems A_2 and B_1, respectively, and zeroing the joint velocities ${}^{B_1}V_{A_2}$ (since there is no joint between coordinate systems A_1 and A_2):

$$^{\bar{A}_1}\mathbf{v}_{A_1} = {}^{A_1}\mathbf{L}_{A_2}\,{}^{\bar{A}_2}\mathbf{v}_{A_2}. \qquad (3.4.13)$$

The joint velocity transformation in (3.4.13) and the link velocity transformation in (3.4.12) apply to any two adjacent coordinate systems in the open-chain. We thus cascade the link and joint velocity transformations repeatedly to compute the velocities of coordinate system A_1 as a *linear* function of the velocities of the main link coordinate system $^{\bar{M}(M)}\mathbf{v}_{M(M)}$ and the velocities $^{Y_1}\mathbf{v}_{X_2}$ of the intervening joints for X,Y = A,B; B,C; ...; L,M:

$$^{\bar{A}_1}\mathbf{v}_{A_1} = {}^{A_1}\mathbf{L}_{M(M)}\,{}^{\bar{M}(M)}\mathbf{v}_{M(M)} + \sum_{X,Y}\left({}^{A_1}\mathbf{L}_{X_2}\,{}^{X_2}\mathbf{J}_{Y_1}\,{}^{Y_1}\mathbf{v}_{X_2}\right). \qquad (3.4.14)$$

The main link velocities $^{\bar{M}(M)}\mathbf{v}_{M(M)}$ and joint velocities $^{Y_1}\mathbf{v}_{X_2}$ in (3.4.14) are called the *partial velocities* of the resultant velocity $^{\bar{A}_1}\mathbf{v}_{A_1}$ [Kane85].

The closed-form solution in (3.4.14) is inefficient for computing all of the velocities in an open-chain. Let us consider computing, in turn, the velocity vectors $^{\bar{X}}\mathbf{v}_X$ for $X=M_1,L_2,\ldots,A_1$ of each coordinate system from the main link to the environment. Application of the closed-form velocity transformation in (3.4.14) to the computation of each successive velocity vector requires the computation of the previous velocity vector. We avoid recomputing the transformation of the previous velocity vector in the computation of the succeeding velocity vector by implementing *recursively* the link and joint velocity transformations in (3.4.12) and (3.4.13) directly. We compute the velocity vector in each coordinate system in turn, from the main link to the environment, by applying the link velocity transformation in (3.4.13) if there is no intervening joint between the present coordinate system and the previous coordinate system, and the joint velocity transformation in (3.4.12) if there is an intervening joint. The recursive implementation parallels the recursive Newton-Euler formulation [Luh80b] for computing recursively the velocities of each link of an open-chain robotic manipulator from the base to the end-effector. The closed-form velocity transformation in (3.4.14) is valuable for formulating closed-form dynamic models to extract physical insights into the dynamics of the robotic mechanism.

3.4.3 Six-Vector Acceleration Transformations

The open-chain link accelerations referred to instantaneously coincident coordinate systems, as required by the force/torque equations-of-motion in Section 3.3.4, are neither specified nor sensed in actual implementations. In this section, we thus compute the open-chain link accelerations (within their respective instantaneously coincident coordinate systems). Our goal is to compute the accelerations ${}^{\bar{A}_1}\mathbf{a}_{A_1}$ of coordinate system A_1 in Figure 3.3.1 from the velocities ${}^{\bar{M}(M)}\mathbf{v}_{M(M)}$ and accelerations ${}^{\bar{M}(M)}\mathbf{a}_{M(M)}$ of the main link coordinate system M(*M*) and the velocities ${}^{Y_1}\mathbf{v}_{X_2}$ and accelerations ${}^{Y_1}\mathbf{a}_{X_2}$ of the intervening joints for X,Y = A,B; B,C; ...; L,M.

To solve this *acceleration transformation* problem, we first consider the simpler problem of transforming accelerations from coordinate system B_1 to coordinate system A_2. We thus compute the accelerations ${}^{\bar{A}_2}\mathbf{a}_{A_2}$ of coordinate system A_2 from the velocities ${}^{\bar{B}_1}\mathbf{v}_{B_1}$ and accelerations ${}^{\bar{B}_1}\mathbf{a}_{B_1}$ of coordinate system B_1 and the velocities ${}^{B_1}\mathbf{v}_{A_2}$ and accelerations ${}^{B_1}\mathbf{a}_{A_2}$ of the intervening joint.

We differentiate the translational velocity ${}^{\bar{A}_2}\mathbf{v}_{A_2}$ in (3.4.9) and rotational velocity ${}^{\bar{A}_2}\boldsymbol{\omega}_{A_2}$ in (3.4.11) within the A_2 coordinate system (i.e., $\frac{{}^{A_2}d}{dt}$) to compute the translational acceleration ${}^{\bar{A}_2}\mathbf{a}_{A_2}$ and the rotational acceleration ${}^{\bar{A}_2}\boldsymbol{\alpha}_{A_2}$, respectively. To implement the differentiation, we apply the following *five* fundamental rules in which **h** is a three-vector and U, V, X, Y, and Z are coordinate systems.

Rule 1.
$$\frac{{}^{X}d}{dt}\,{}^{X}\mathbf{h}_y = {}^{X}\dot{\mathbf{h}}_y \tag{3.4.15}$$

The first rule is the notational convention of this dissertation; a dotted vector denotes differentiation within the reference coordinate system of the vector (i.e., the X coordinate system).

Rule 2.
$${}^{\bar{X}}\mathbf{h}_Z = {}^{\bar{X}}\mathbf{R}_{\bar{Y}}\,{}^{\bar{Y}}\mathbf{h}_Z \tag{3.4.16}$$

The second rule formalizes vector transformations in relatively stationary coordinate systems. For two vectors, defined in coordinate systems $\bar{X}$ and $\bar{Y}$ which are relatively stationary, the vector defined in coordinate system $\bar{X}$ is equal to the vector defined in coordinate system $\bar{Y}$ multiplied by the (3×3) rotation matrix ${}^{\bar{X}}R_{\bar{Y}}$.

Rule 3.
$$\frac{{}^{\bar{X}}d}{dt}\,{}^{\bar{X}}R_{\bar{Y}}\,{}^{U}h_V = {}^{\bar{X}}R_{\bar{Y}}\frac{{}^{\bar{Y}}d}{dt}\,{}^{U}h_V \qquad (3.4.17)$$

The third rule states that the operations of *rotation* between two relatively stationary coordinate systems and *differentiation* within the two coordinate systems commute.

Rule 4.
$$\frac{{}^{\bar{X}}d}{dt}\,{}^{X}h_y = \frac{{}^{X}d}{dt}\,{}^{X}h_y + \left({}^{\bar{X}}\omega_X \times {}^{X}h_y\right) \qquad (3.4.18)$$

The fourth rule is the classical expression for computing the derivative of the vector ${}^{X}h_y$ within the stationary coordinate system $\bar{X}$ from the derivative of the vector with respect to the moving coordinate system X and the rotational velocity ${}^{\bar{X}}\omega_X$ of the moving coordinate system [Landau76].

Rule 5.
$$\frac{{}^{X}d}{dt}\left({}^{Y}h_Z \times {}^{U}h_V\right) = \left(\frac{{}^{X}d}{dt}\,{}^{Y}h_Z \times {}^{U}h_V\right) + \left({}^{Y}h_Z \times \frac{{}^{X}d}{dt}\,{}^{U}h_V\right) \qquad (3.4.19)$$

Finally, the fifth rule is a simplification rule for the differentiation of vector cross-products [Beyer82].

We differentiate (3.4.9) within the A_2 coordinate system and apply the foregoing five rules to obtain the translational accelerations:

$${}^{\bar{A}_2}a_{A_2} = {}^{\bar{A}_2}R_{\bar{B}_1}\left\{{}^{\bar{B}_1}a_{B_1} + {}^{B_1}a_{A_2} + 2({}^{\bar{B}_1}\omega_{B_1} \times {}^{B_1}v_{A_2}) + ({}^{\bar{B}_1}\alpha_{B_1} \times {}^{B_1}p_{A_2}) + [{}^{\bar{B}_1}\omega_{B_1} \times ({}^{\bar{B}_1}\omega_{B_1} \times {}^{B_1}p_{A_2})]\right\}. \qquad (3.4.20)$$

We differentiate (3.4.11) within the A_2 coordinate system and apply rules one through four to obtain the rotational accelerations:

$${}^{\bar{A}_2}\alpha_{A_2} = {}^{\bar{A}_2}R_{\bar{B}_1}\,{}^{\bar{B}_1}\alpha_{B_1} + {}^{\bar{A}_2}R_{\cdot}\,{}^{\bar{B}_1}\alpha_{A_2} + {}^{\bar{A}_2}R_{\cdot}\,({}^{\bar{B}_1}\omega_{B_1} \times {}^{B_1}\omega_{A_2}). \qquad (3.4.21)$$

The three-vector relationships in (3.4.20) and (3.4.21) may be written compactly as the six-vector:

$$\overline{A_2}\mathbf{a}_{A_2} = {}^{A_2}\mathbf{L}_{B_1}\,\overline{B_1}\mathbf{a}_{B_1} + {}^{A_2}\mathbf{J}_{B_1}\,{}^{B_1}\mathbf{a}_{A_2} + {}^{A_2}\mathbf{R}_{B_1}\,\overline{B_1}\omega^{r}_{A_2} + {}^{A_2}\mathbf{S}_{B_1}\,\overline{B_1}\omega^{2}_{B_1} + {}^{A_2}\mathbf{T}_{B_1}\,\overline{B_1}\omega^{t}_{A_2}. \tag{3.4.22}$$

In (3.4.22), the (6×6) *rotational velocity product* matrix ${}^{A_2}\mathbf{R}_{B_1}$, the (6×6) *squared velocity product* matrix ${}^{A_2}\mathbf{S}_{B_1}$, and the (6×6) *translational velocity product* matrix ${}^{A_2}\mathbf{T}_{B_1}$ are computed from the templates in Appendix 12 and the position six-vector ${}^{B_1}\mathbf{p}_{A_2}$. The rotational velocity product six-vector $\overline{B_1}\omega^{r}_{A_2}$, the squared velocity product six-vector $\overline{B_1}\omega^{2}_{B_1}$, and the translational velocity product six-vector $\overline{B_1}\omega^{t}_{A_2}$ are:

$$\overline{B_1}\omega^{r}_{A_2} = \left[\omega_{1x}\omega_{2y} \quad \omega_{1x}\omega_{2z} \quad \omega_{1y}\omega_{2x} \quad \omega_{1y}\omega_{2z} \quad \omega_{1z}\omega_{2x} \quad \omega_{1z}\omega_{2y}\right]^T, \tag{3.4.23}$$

$$\overline{B_1}\omega^{2}_{B_1} = \left[\omega_x^2 \quad \omega_y^2 \quad \omega_z^2 \quad \omega_x\omega_y \quad \omega_x\omega_z \quad \omega_y\omega_z\right]^T, \tag{3.4.24}$$

and $$\overline{B_1}\omega^{t}_{A_2} = \left[\omega_{1x}v_{2y} \quad \omega_{1x}v_{1z} \quad \omega_{1y}v_{2x} \quad \omega_{1y}v_{2z} \quad \omega_{1z}v_{2x} \quad \omega_{1z}v_{2y}\right]^T, \tag{3.4.25}$$

where $\omega_{1i} = \overline{B_1}\omega_{B_1 i}$, $\omega_{2i} = {}^{B_1}\omega_{A_2 i}$ and $v_{2i} = {}^{B_1}v_{A_2 i}$ for i=x,y,z.

We apply the acceleration transformation in (3.4.22) to transform accelerations in coordinate system A_2 to coordinate system A_1 by allowing coordinate systems A_1 and A_2 to play the roles of coordinate systems A_2 and B_1, respectively, and zeroing the joint velocities ${}^{B_1}\mathbf{v}_{A_2}$ and accelerations ${}^{B_1}\mathbf{a}_{A_2}$ (since there is no joint between coordinate systems A_1 and A_2):

$$\overline{A_1}\mathbf{a}_{A_1} = {}^{A_1}\mathbf{L}_{A_2}\,\overline{A_2}\mathbf{a}_{A_2} + {}^{A_1}\mathbf{S}_{A_2}\,\overline{A_2}\omega^{2}_{A_1}. \tag{3.4.26}$$

The joint acceleration transformation in (3.4.22) and the link acceleration transformation in (3.4.26) apply to any two adjacent coordinate systems in the open-chain. We thus cascade the link and joint acceleration transformations repeatedly to compute the accelerations ${}^{\bar{A}_1}\mathbf{a}_{A_1}$ of coordinate system A_1 as a *linear* function of the accelerations ${}^{\bar{M}(M)}\mathbf{a}_{M(M)}$ of the main link and the accelerations ${}^{Y_1}\mathbf{a}_{X_2}$ of the *intervening* joints, and a nonlinear function of the velocities ${}^{\bar{M}(M)}\mathbf{v}_{M(M)}$ of the main link and the velocities ${}^{Y_1}\mathbf{v}_{X_2}$ of the intervening joints for X,Y = A,B; B,C; ...; L,M:

$$ {}^{\bar{A}_1}\mathbf{a}_{A_1} = {}^{A_1}\mathbf{L}_{M(M)}\, {}^{\bar{M}(M)}\mathbf{a}_{M(M)} + \sum_{X,Y} \left({}^{A_1}\mathbf{L}_{X_1}\, {}^{X_1}\mathbf{S}_{X_2}\, \bar{X}_2 \omega_{X_1}^2 + {}^{A_1}\mathbf{h}_{M(M)} \right), \qquad (3.4.27) $$

where

$$ {}^{A_1}\mathbf{h}_{M(M)} = {}^{A_1}\mathbf{L}_{Y_2} \left({}^{Y_2}\mathbf{J}_{X_1}\, {}^{X_1}\mathbf{a}_{Y_2} + {}^{Y_2}\mathbf{R}_{X_1}\, \bar{X}_1 \omega_{Y_2}^r + {}^{Y_2}\mathbf{S}_{X_1}\, \bar{X}_1 \omega_{X_1}^2 + {}^{Y_2}\mathbf{T}_{X_1}\, \bar{X}_1 \omega_{Y_2}^t \right). $$

In parallel with the closed-form velocity transformation in (3.4.14), the closed-form acceleration transformation in (3.4.27) is inefficient for computing all of the accelerations in an open-chain. The application of the closed-form acceleration transformation in (3.4.27) to the computation of each successive acceleration vector requires the computations of the previous acceleration vector. We avoid recomputing the transformation of the previous acceleration vector in the computation of the succeeding acceleration vector by implementing *recursively* the link and joint acceleration transformations in (3.4.22) and (3.4.26). We compute the acceleration vector in each coordinate system in turn, from the main link to the environment, by applying the link acceleration transformation in (3.4.26) if there is no intervening joint between the present coordinate system and the previous coordinate system, and the joint acceleration transformation in (3.4.22) if there is an intervening joint. The recursive implementation parallels the recursive Newton-Euler formulation [Luh80b] for computing the accelerations of each link of an open-chain robotic manipulator from the base to the end-effector. The closed-form acceleration transformation in (3.4.27) is valuable for formulating closed-form dynamic

models to illuminate physical insights into the dynamics of the robotic mechanism.

3.4.4 Concluding Remarks

The dynamic model of a robotic mechanism is formulated by computing independently the force/toque equations-of-motion in Section 3.3.4 and the kinematic transformations in this section. The kinematic transformations are substituted into the force/torque equations-of-motion to formulate a dynamic model which depends only upon the accelerations of the main link and the sensed joint positions and velocities. As discussed in Section 3.5, the dynamic model is then solved for the actuator forces/torques (in servo-control applications) and for the WMR body accelerations (in simulation applications).

3.5 Forward and Inverse Dynamic Solutions

3.5.1 Overview

We solve the dynamic WMR model for servo-controller design and computer simulation applications. For WMR servo-controller design, we require the solution of the WMR model to compute the actuator torques from the desired WMR motion in the *actuated inverse dynamic solution.* For WMR computer simulation, we require the solution of the WMR model to compute the WMR body accelerations from the wheel actuator forces/torques and the positions and velocities of the sensed joints in the *forward dynamic solution.*

The $6+N_s$ dynamic equations-of-motion in (3.4.1), which model a robotic mechanism, are *linear* in the a actuator forces/torques τ_a, the 6N environmental forces/torques $\mathbf{f_e}$ and the six accelerations of the main link ${}^{\bar{M}(M)}\mathbf{a}_{M(M)}$; i.e., the dynamic equations-of-motion are:

$$\mathbf{A}\,\tau_a + \mathbf{B}\,\mathbf{f_e} + \mathbf{C}\,{}^{\bar{M}(M)}\mathbf{a}_{M(M)} = \mathbf{b}\,, \qquad (3.5.1)$$

where **A** is $[(6+N_s)\times a]$, **B** is $[(6+N_s)\times 6N]$ and **C** is $[(6+N_s)\times 6]$, and **b** is a $(6+N_s)$-vector. In Sections 3.5.2 and 3.5.3, we enumerate the numbers of unknowns

and computational procedures for the actuated inverse and forward dynamic solutions of the WMR dynamic equations-of-motion in (3.5.1).

In this section, we apply the classical solution criteria and methodologies for systems of linear algebraic equations [Brogan85, Cohn58] to solve the dynamic WMR model. We apply the *weighted minimum-norm* solution to compute the actuated inverse and the forward dynamic solutions. The weighted minimum-norm actuated inverse solution minimizes a weighted combination of actuator torques to conserve power or avoid wheel slip. These are critical issues for an autonomous WMR which must carry its power supply onboard and relies upon dead reckoning for WMR position estimates. The minimum-norm solution is also required to handle the undetermined nature of the constraint forces which exist in a robotic mechanism which contacts the environment at more than three points (such as the normal forces on the wheels of a WMR having more than three wheels).

We introduce a *delay-by-one sampling period* technique which allows us to *decouple* the computation of the dry friction coefficients (which are nonlinear in the internal forces/torques) from the solution of the dynamic equations-of-motion. We thereby compute the frictional coefficients at the present sampling period from the internal forces/torques computed at the previous sampling period. By choosing the sampling period of the dynamics implementation (e.g., the servo-controller or computer simulation) sufficiently small, delay-by-one introduces negligible computational degradation. This requirement is not restrictive since the sampling period of a robotic mechanism to be controlled must be chosen to be much smaller than the time constants of the mechanism for digital servo-controller implementation.

We first compute the actuated inverse dynamic solution in Section 3.5.2. We next compute the forward dynamic solution in Section 3.5.3. We then introduce the delay-by-one sampling period algorithm for frictional coefficient computation in Section 3.5.4. Finally, in Section 3.5.5, we advance our concluding remarks.

3.5.2 Actuated Inverse Dynamic Solution

For servo-control of a robotic mechanism, we require the actuated inverse dynamic solution; i.e., the solution of the actuator forces/torques τ_a to drive the main link along a desired trajectory represented by ${}^{\bar{M}(M)}\mathbf{a}_{M(M)}$. We must also compute the environmental forces/torques to obtain, in turn, the internal forces/torques and the frictional coefficients. We arrange the dynamic equations-of-motion in (3.5.1) in the classical form:

$$\mathbf{A}_i \mathbf{x}_i = \mathbf{b}_i \tag{3.5.2}$$

in which the matrix $\mathbf{A}_i = [\mathbf{A}\ \mathbf{B}]$, the unknown vector $\mathbf{x}_i = [\tau_a^T, \mathbf{f}_e^T]^T$ and the vector $\mathbf{b}_i = \mathbf{b} - \mathbf{C}\,{}^{\bar{M}(M)}\mathbf{a}_{M(M)}$.

For WMR applications, the number of unknown actuator forces/torques and environmental forces/torques (6N+a) exceeds the number of independent equations ($6+N_s$). For Uranus (in Chapter 6): N=4, a=4, and N_s=20; and for Bicsun-Bicas (in Chapter 7): N=4, a=2, and N_s=18. In the dynamic models of these exemplary WMRs, there are *two* more unknowns than equations. We thus apply the *minimum-norm solution* [Brogan85].

We begin by applying Gauss elimination to reduce the computational complexity of the minimum norm solution. We solve independently each of the N_s secondary equations for N_s of the 6N environmental contact forces/torques from the N normal environmental contact forces along the z-axis and the ($5N-N_s$) remaining environmental contact forces/torques. We substitute the N_s computed environmental contact forces/torques into the six primary dynamic equations-of-motion. This step reduces the dynamic system model to six equations in ($6N+a-N_s$) unknown actuator forces/torques and environmental forces/torques $\mathbf{f}_e^*$:

$$\mathbf{A}_i^* \mathbf{x}_i^* = \mathbf{b}_i^*, \tag{3.5.3}$$

where the reduced unknown vector is $\dot{x}_i = [\tau_a^T, \dot{f}_e^T]^T$. We then apply the weighted minimum-norm solution[7]:

$$\dot{x}_i = W^{-1}\dot{A}_i^T [\dot{A}_i W^{-1}\dot{A}_i^T]^{-1} \dot{b}_i \qquad (3.5.4)$$

to solve the undetermined system of linear algebraic equations in (3.5.3) and thereby compute the actuator forces/torques. The *diagonal* weighting matrix W is:

$$W = \begin{pmatrix} W_a & 0 \\ 0 & W_e \end{pmatrix}, \qquad (3.5.5)$$

where the weights on the environmental forces/torques (the diagonal elements of W_e) are set equal to unity and the weights on the actuator forces/torques (the diagonal elements of W_a) are chosen to minimize power consumption or avoid wheel slip. For the two prototype WMRs studied in Chapters 6 and 7, we partition (3.5.4) into subsystems of three or fewer equations, and apply the weighted minimum-norm solution to each subsystem to limit the computational complexity to the solution of three or fewer simultaneous linear algebraic equations.

3.5.3 Forward Dynamic Solution

To simulate the robotic mechanism, we solve the dynamic system model in (3.5.1) for the accelerations of the main link from the actuator forces/torques. We then compute the velocity and position trajectories of the main link by integrating the accelerations as described in Chapter 5.

The unknowns in the dynamic equations-of-motion in (3.5.1) for the forward dynamic solution are the six main link accelerations $\overline{{}^{M(M)}a}_{M(M)}$ and the 6N environmental contact forces f_e. The actuator forces/torques τ_a are specified as inputs to the simulation. In parallel with the actuated inverse dynamic

---

7 The weighted minimum norm solution of (3.5.3) is the vector x which minimizes the weighted sum-of-squares $x^T W x$ subject to the linear algebraic equation constraint Ax=b. Straightforward application of Lagrange multipliers leads directly to (3.5.4).

solution, we arrange the dynamic equations-of-motion in (3.5.1) into the classical form:

$$\mathbf{A}_f \mathbf{x}_f = \mathbf{b}_f \tag{3.5.6}$$

in which the matrix $\mathbf{A}_f = [\mathbf{C}\ \mathbf{B}]$, the vector $\mathbf{x}_f = [{}^{\bar{M}(M)}\mathbf{a}_{M(M)},\ \mathbf{f}_e^T]^T$, and the vector $\mathbf{b}_f = \mathbf{b} - \mathbf{A}\,\tau_a^T$.

For WMR simulation, the number of unknown main link accelerations and environmental forces/torques (6N+6) exceeds the number of independent equations ($6+N_s$). We again apply the *weighted minimum-norm solution* [Brogan85]. In parallel with the inverse dynamic solution, we apply Gauss elimination to reduce the ($6+N_s$) equations in the dynamic model to six equations in six unknown main link accelerations and ($6N-N_s$) unknown environmental forces/torques $\mathbf{f}_e^*$:

$$\mathbf{A}_f^* \mathbf{x}_f^* = \mathbf{b}_f^*, \tag{3.5.7}$$

where the reduced unknown vector $\mathbf{x}_f^* = [\tau_a^T,\ \mathbf{f}_e^{*T}]^T$. We next apply the weighted minimum-norm solution in (3.5.4) with an *identity* weighting matrix to compute the main link accelerations and the environmental forces/torques. For any set of actuator forces/torques there is a unique set of system link accelerations and environmental contact forces/torques because actual mechanical systems are deterministic. There must always be exactly one solution to the dynamic system model in (3.5.7), and the unknown environmental forces/torques must depend upon the unknown main link accelerations. Consequently, the minimum-norm solution of (3.5.8) must be independent of the weighting matrix **W**. We thus specify the weighting matrix as the identity matrix and compute the forward solution as:

$$\mathbf{x}_f^* = \mathbf{A}_f^{*T} [\mathbf{A}_f^* \mathbf{A}_f^{*T}]^{-1} \mathbf{b}_f^*. \tag{3.5.8}$$

For the two prototype WMRs studied in Chapters 6 and 7, we partition (3.5.8) into subsystems of three or fewer equations, and apply the weighted minimum-

norm solution to each subsystem to limit the computational complexity to the solution of three or fewer simultaneous linear algebraic equations.

3.5.4 Computing the Internal Forces/Torques

The environmental forces/torques are required to compute the internal forces/torques in Section 3.3.4.3. The internal forces/torques are required for the computation of the dry friction coefficients, which are in turn required to compute the force/torque equations-of-motion in Sections 3.3.4.2 and 3.3.4.4. The force/torque equations-of-motion and the kinematic transformations in Section 3.4 are then combined to formulate the dynamic equations-of-motion from which we compute the actuated inverse and forward dynamic solutions. However, as discussed in Sections 3.5.2 and 3.5.3, the actuated inverse or the forward dynamic solution is required to compute the environmental forces/torques. This realization suggests that the internal forces/torques should be solved simultaneously with the actuated inverse and forward dynamic solutions. Unfortunately, the nonlinear nature of dry friction precludes the application of linear methodologies to compute the internal forces/torques simultaneously with the dynamic model.

We resolve the dilemma by noting that the actuated inverse and forward solutions of the dynamic model will be implemented on a digital computer which executes the solution program once every sampling period T for servo-control or simulation applications. We thus propose to compute the actuated inverse or forward dynamic solution for the *present sampling period* from the internal forces/torques from the *previous sampling period*; and then compute the internal forces/torques for the present sampling period for application to the following sampling period. The errors introduced by this computational procedure are due to the changes in the internal forces/torques from one sampling period to the next. We require that the sampling period be small enough so that the changes in the internal forces/torques between sampling periods are negligible. We fulfill this requirement in our servo-controller designs in Chapters 6 and 7 by exercising the engineering practice of constraining the sampling period to be less than ten percent of the WMR time constant.

3.5.5 Concluding Remarks

We have applied the weighted minimum-norm solution to compute the actuated inverse and forward WMR dynamic solutions for servo-control (in Chapter 4) and computer simulation (in Chapter 5) applications. We then introduced a delay-by-one sampling period technique to compute the internal forces/torques by decoupling the solution of the linear dynamic equations-of-motion from the computation of the nonlinear dry frictional coefficients.

3.6 WMR Dynamic Modeling Procedure

In this chapter, we have developed a framework for the dynamic modeling of a robotic mechanism having the closed-chain structure shown in Figure 1.1. In this section, we enumerate a step-by-step procedure to model the dynamics of a WMR. We make correspondences between the WMR components and the system links in Figure 1.1, assign coordinate systems to each of the WMR links and then follow the modeling framework detailed in Sections 3.3 to 3.5.

We begin by identifying the following five WMR components which correspond to the links of the robotic mechanism depicted in Figure 1.1:

(1) WMR main body,
(2) Steering links (for steered wheels),
(3) Wheels,
(4) Rollers (for omnidirectional wheels), and
(5) Load.

The *WMR main body* consists of the structural and electrical components which move rigidly with the main section of the WMR. The WMR main body typically includes batteries, computers and electronics boards, the main supporting structure, the stators of the steering motors (for actuated steered wheels), and the stators of the wheel motors (for actuated nonsteered wheels). The armature of the steering motors and the stator of wheel motors (for steered actuated wheels) are integral components of the *steering link* since they move rigidly with the steering link. The armatures of the wheel motors are integral components of the *wheels*. All *rollers* on an omnidirectional wheel, except the

roller in contact with the surface-of-travel, are components of the wheel. The single roller in contact with the surface-of-travel is modeled as a separate rigid link. The WMR *load* which may represent parts or materials sits upon the WMR body.

The WMR main body plays the role of the main link in the simple closed-chain robotic mechanism in Figure 1.1. Each wheel assembly, consisting of: a wheel (for each conventional nonsteered wheel), a steering link and a wheel (for each steered conventional wheel), or a wheel and a roller (for each omnidirectional wheel), plays the role of one of the N open-chains in Figure 1.1. Steered omnidirectional wheels and ball wheels may be modeled similarly. The load corresponds to an open-chain which does not contact the environment (i.e., the coupling matrix between the load and the environment is the null matrix).

The assignment of coordinate systems throughout a WMR for subsequent dynamic modeling is motivated by the component force/torque models in Section 3.3.2. To model the WMR forces/torques and force/torque couplings, we must assign the following four natural coordinate systems for each WMR link *A*:

- Center-of-mass M(*A*) for inertial forces/torques;
- Center-of-volume V(*A*) for viscous frictional forces/torques;
- Point-of-application P(*A*) for actuator forces/torques; and
- Contact point C(*A*,*B*) for environmental contact forces/torques for all links *B* which contact link *A*.

The gravitational coordinate system G(*A*) is also required for each WMR link, but need not be assigned explicitly since its location is determined by the location of the center-of-mass coordinate system as discussed in Section 3.3.3.5. The coordinate system assignments for kinematic modeling in Table 2.2 are utilized for dynamic modeling. We compile in Table 3.1 a set of additional coordinate systems which, together with those coordinate systems assigned for kinematic modeling in Table 2.2, are required for WMR dynamic modeling.

Table 3.1: Additional Dynamic WMR Modeling Coordinate Systems

G	*Gravitational*: Stationary coordinate system fixed with the Earth, with the z-axis opposed to the gravitational field.
B	*Body*: Coordinate system fixed with the WMR main body, with the origin at the center-of-mass of the WMR body, and aligned with the principal axes.
B_t	*Body Top*: Coordinate system fixed with the WMR main body, with the origin at the geometric center of the contact area between the WMR body and the load, and the z-axis perpendicular to the contact area.
L	*Load*: Coordinate system fixed with the load, with the origin at the center-of-mass of the load, and the axes aligned with the principal axes.
L_b	*Load Bottom*: Coordinate system fixed with the load, and coincident with B_t.
S_{mi}	*Steering Center-of-Mass* (for steered wheel i): Coordinate system for steered wheels, fixed with steering link i, with the origin at the center-of-mass of steering link i, and the axes aligned with the principal axes.
W_i	*Wheel* (for i=1,...,N): Coordinate system fixed with wheel i, with the origin at the geometric center of the wheel, and the x-axis aligned with the axle.
A_i	*Axle* (for i=1,...,N): Coordinate system fixed with steering link i if there is one, otherwise fixed with the WMR main body; with the origin and x-axis coincident with the origin and x-axis of W_i.
R_i	*Roller* (for omnidirectional wheel i): Coordinate system for omnidirectional wheels, fixed with the roller on wheel i which is in contact with the surface-of-travel, with the origin at the geometric center of the roller, and the x-axis aligned with the roller axle.
P_i	*Pivot* (for omnidirectional wheel i): Coordinate system fixed with omnidirectional wheel i; with the origin and the x-axis coincident with the origin and x-axis of R_i.

In practice, some of the natural coordinate systems required for dynamic modeling coincide, reducing the number of coordinate systems to be assigned. For example, the center-of-mass and the center-of-volume of each link often coincide, and the point-of-application and a contact point coordinate system typically coincide. Table 3.2 displays the correspondence between the WMR coordinate systems assigned in Tables 2.2 and 3.1 and the natural coordinate systems required to model the force/torques on each WMR link.

Table 3.2: Natural Coordinate Systems for WMR Links

A	M(A)	G(A)	P(A)	V(A)	B_1	C(A,B_1)	B_2	C(A,B_2)
WMR Main Link	B	B	-	B	*Steering Link i or Wheel i*	H_i A_i	*Load*	B_t
Steering Link i	S_{mi}	S_{mi}	S_i	S_{mi}	*WMR Main Link*	S_i	*Wheel i*	A_i
Wheel i	W_i	A_i	W_i	W_i	*Steering Link i or Main Link*	W_i W_i	*Environment or Roller i*	C_i P_i
Roller i	R_i	P_i	-	R_i	*Wheel i*	R_i	*Environment*	C_i
Load	L	L	-	L	*Main Link*	L_b	-	-

In Tables 2.2 and 3.1, we have defined the coordinate systems utilized in Table 3.2. With these coordinate system assignments, we apply the following step-by-step WMR dynamic modeling procedure.

(1) Sketch the WMR and show the relative locations of each component (i.e., the load, main body, steering links, wheels and rollers).

(2) Assign the coordinate systems to each WMR component according to the conventions specified in Tables 2.2 and Table 3.1, and draw each coordinate system in the appropriate sketch made in Step 1.

(3) Formulate the position six-vectors ${}^{A}\mathbf{p}_{B}$ between all pertinent adjacent coordinate systems A and B in the WMR by inspection of the sketches.

(4) Formulate the force/torque six-vectors for each link in the WMR; i.e., the inertial force/torque six-vector in (3.3.3), gravitational force/torque six-vector in (3.3.4), actuation force/torque six-vector in (3.3.5), viscous friction force/torque six-vector in (3.3.6), and environmental contact force/torque six-vector in (3.3.8)

(5) Formulate the coupling matrix for each joint as described in Section 3.3.3.3.

(6) Formulate the link Jacobian matrices as described in Section 3.3.3.2.

(7) Compute the propagation matrices as described in Section 3.3.3.4.

(8) Formulate the six primary force/torque equations-of-motion developed in Section 3.3.4.2.

(9) Formulate the internal force/torque equations developed in Section 3.3.4.3.

(10) Formulate the secondary force/torque equations-of-motion developed in Section 3.3.4.4.

(11) Transform the instantaneously coincident velocities to main link velocities and joint velocities as described in Section 3.4.2.

(12) Transform the instantaneously coincident accelerations to main link accelerations and joint velocities and accelerations as described in Section 3.4.3.

(13) Compute the inverse velocity solution by assuming that all of the wheel velocities are actuated in (2.4.8).

(14) Compute the sensed forward velocity solution in (2.4.14).

(15) Compute the inverse acceleration solution in analogy with the inverse velocity solution in (2.4.8).

(16) Substitute the kinematic transformations computed in Steps 11 to 15 into the force/torque equations-of-motion formulated in Steps 8 to 10.

The dynamic model of the WMR is computed *symbolically* according to the sixteen step procedure enumerated above. We perform the required symbolic computations with the aid of the mathematical computation program MACSYMA [Symbolics85]. In Chapters 6 and 7, we program MACSYMA to formulate the dynamic models of two WMRs. We define the kinematic position six-vectors, the force/torque six-vectors, and the coupling matrices within a MACSYMA batch file. Then, MACSYMA automatically structures and computes the six primary and N_s secondary *symbolic* dynamic equations-of-motion.

3.7 Summary

The dynamic model of the robotic mechanism in Figure 1.1 characterizes the relationships between the forces/torques and motions of the system. The *application* of the dynamic model dictates the required *accuracy* of the model which, in turn, dictates which physical phenomena to model depending upon their relative significance. An investigation into the relative significance of the forces/torques in our dynamic model for a particular system or application is beyond the scope of this dissertation. Identification of the characteristic parameters which model each phenomena is similarly beyond the scope of this dissertation. We have operated under the assumption that the physically significant phenomena within the system to be modeled have been determined, and that all kinematic, dynamic and frictional characteristic parameters have been identified.

In this chapter, we have developed the *modules* to model the dynamics of the simple closed-chain robotic mechanism in Figure 1.1 and have applied these concepts to enumerate a step-by-step procedure for the dynamic modeling of a WMR. Servo-control algorithms emanating from our dynamic WMR model may be optimized for efficient computation by eliminating all scalar additions and multiplications by zero and multiplications by plus and minus one before implementation. Moreover, the servo-control algorithms may be amenable to execution on an array processor because the dynamic model is founded upon matrix-vector products. In this case, the dynamic model is appropriate for direct implementation as illustrated in Appendix 14. Having formulated methodologies for the kinematic and dynamic modeling of WMRs, we turn our attention in Chapter 4 to WMR model-based servo-controller design.

Chapter 4
Model-Based WMR Servo-Controller Design

4.1 Introduction

In Chapters 2 and 3, we formulated kinematic and dynamic WMR models. In this chapter, we incorporate these models in the design of accurate WMR *servo-controllers*. The task of the servo-controller depicted in Figure 4.1 is to compute in *real-time* the actuator command signals to drive the WMR body along the reference trajectory. At each sampling instant, the trajectory planner communicates the current reference positions, velocities, and accelerations to the servo-controller. Within each sampling period, the servo-controller computes the command signals from the reference trajectory, sensory feedback measurements, and a model of the WMR motion. The servo-controller communicates the command signals to the WMR actuators, converting the command signals to either a voltage or PWM signal for direct hardware application. The novel aspects of this chapter are the application of documented stationary manipulator servo-controller designs [Whitney69, Luh80a, Paul81b] to WMRs through the implementation of the WMR kinematic and dynamic models formulated in Chapters 2 and 3 and the design of command conversion algorithms.

We continue the philosophy of our modeling activities by building upon analogies with stationary manipulator servo-controller designs. Over the past twenty years, stationary manipulator servo-control systems have improved progressively: from independent joint-space control [Paul81a], to kinematics-based Cartesian-space control [Whitney69], to dynamics-based Cartesian-space control [Luh80a], to robust dynamics-based control [Tourassis85, Neuman87a] and adaptive control algorithms [Dubowsky79]. Current WMR

servo-controller designs are primarily independent joint-space servo-controllers [Muir85]. Although Cartesian-space kinematics-based servo-controllers have been documented in the literature [Hongo85, Daniel84], dynamics-based Cartesian-space WMR servo-controllers have not. We anticipate that WMR controllers will evolve in the pattern of stationary manipulator controllers by incorporating progressively enhanced kinematic and dynamic models.

In Chapters 6 and 7, we design and evaluate resolved motion rate and resolved acceleration servo-controllers for Uranus and Bicsun-Bicas. In this chapter, we detail the resolved motion rate and resolved acceleration servo-controller methodologies. In Chapter 5, we detail our WMR simulation methodology for servo-controller evaluation.

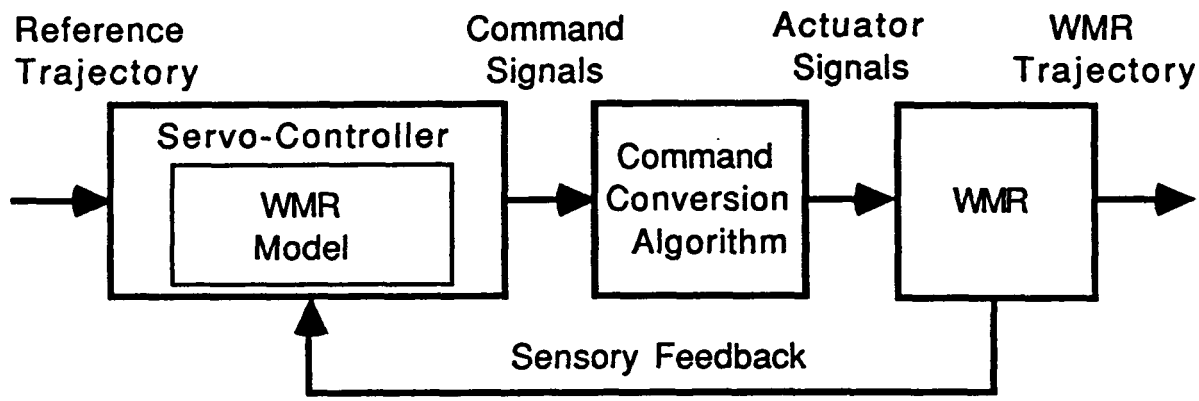


Figure 4.1: A WMR Servo-Controller

The detailed formulation of accurate WMR kinematic and dynamic models in Chapters 2 and 3 paves the way for the application of the aforementioned servo-control methodologies to WMRs. The design of kinematics-based Cartesian-space servo-controllers for stationary manipulators is accomplished with the resolved motion rate control algorithm [Whitney69] which utilizes the actuated inverse and sensed forward velocity solutions. In Section 4.2, we apply the resolved motion rate approach to the design of a WMR servo-controller. The design of dynamics-based Cartesian-space servo-controllers for stationary robotic manipulators is accomplished with the resolved acceleration approach [Luh80a, Khosla87] which applies the actuated inverse dynamic solution to compute the actuator forces/torques in the feedforward path of the servo-controller. The feedforward actuator forces/torques *decouple* the

complex nonlinear robot dynamics from the feedback control operation [Luh80a]. A linear state-variable feedback control algorithm is then formulated (by introducing stable poles) to zero the deviations in the manipulator trajectory caused by modeling errors, unmodeled dynamics, and unmodeled disturbances. The nominal feedforward and feedback control signals are then summed to compute the commanded actuator torques.

Since the command signals computed by the resolved motion rate approach are actuator velocities and the command signals computed by the resolved acceleration approach are actuator torques, we require command conversion algorithms to apply these signals to voltage controlled DC motors. Pulse-width modulation of the applied motor voltage is desirable to take advantage of the power efficiency of PWM [Muir85]. In Section 4.4, we develop four command conversion algorithms: velocity-to-voltage, velocity-to-pulse-width, torque-to-voltage, and torque-to-pulse-width. We summarize our servo-controller designs in Section 4.5.

4.2 Resolved Motion Rate WMR Servo-Control

The resolved motion rate servo-controller, which is a Cartesian-space kinematics-based servo-controller design, allows the specification of reference trajectories in the coordinates of the task [Whitney69] and is thus more desirable than joint-space control for coordinated motion of a robotic mechanism. The motivation for the WMR resolved motion rate servo-controller in Figure 4.2 is straightforward. The feedforward path utilizes the inverse velocity solution to decouple kinematically the three WMR axes of motion. The feedback path computes the WMR body position from wheel sensor measurements. The WMR body position error is then applied to the feedforward path to servo-out the position errors.

We describe the operation of the resolved motion rate WMR servo-controller illustrated in Figure 4.2. We assume that each wheel position ${}^{A}\theta_{W}$ is measured by a shaft encoder and that wheel velocity estimates ${}^{A}\tilde{\omega}_{W}$ are computed by backward differencing or pulse timing [Rangan82]. We include the sensed forward velocity solution in (2.4.14) in the feedback path to estimate the WMR body velocities ${}^{B}\tilde{\bar{v}}_{B}$. The WMR body velocities are transformed to the floor

coordinate system by applying the motion matrix in (2.3.34). Dead reckoning (described in Appendix 9) is applied to estimate the WMR body position ${}^{F}\tilde{\mathbf{p}}_{B}$ from the WMR body velocities ${}^{F}\tilde{\mathbf{v}}_{B}$. We then subtract the estimated WMR body position ${}^{F}\tilde{\mathbf{p}}_{B}$ from the reference WMR body position ${}^{F}\mathbf{p}_{rB}$ (which is provided by the trajectory planner) to compute the WMR body position error ${}^{F}\mathbf{e}_{B}$. Each component ${}^{F}e_{Bi}$ for i=x,y,θ of the position error ${}^{F}\mathbf{e}_{B}$ is filtered by the PSD (proportional+summation+difference) compensator to compute the control signal ${}^{F}u_{Bi}$ at the n-th sampling instant:

$$ {}^{F}u_{Bi}(nT) = k_{pi}\,{}^{F}e_{Bi}(nT) + k_{si}\sum_{j=1}^{n}{}^{F}e_{Bi}(jT) + k_{di}\left({}^{F}e_{Bi}(nT)-{}^{F}e_{Bi}[(n-1)T]\right) \qquad (4.2.1) $$

where the position k_{pi}, summation k_{si}, and difference gains k_{di} are specified independently for each axis i=x,y,θ, and T is the constant sampling period.

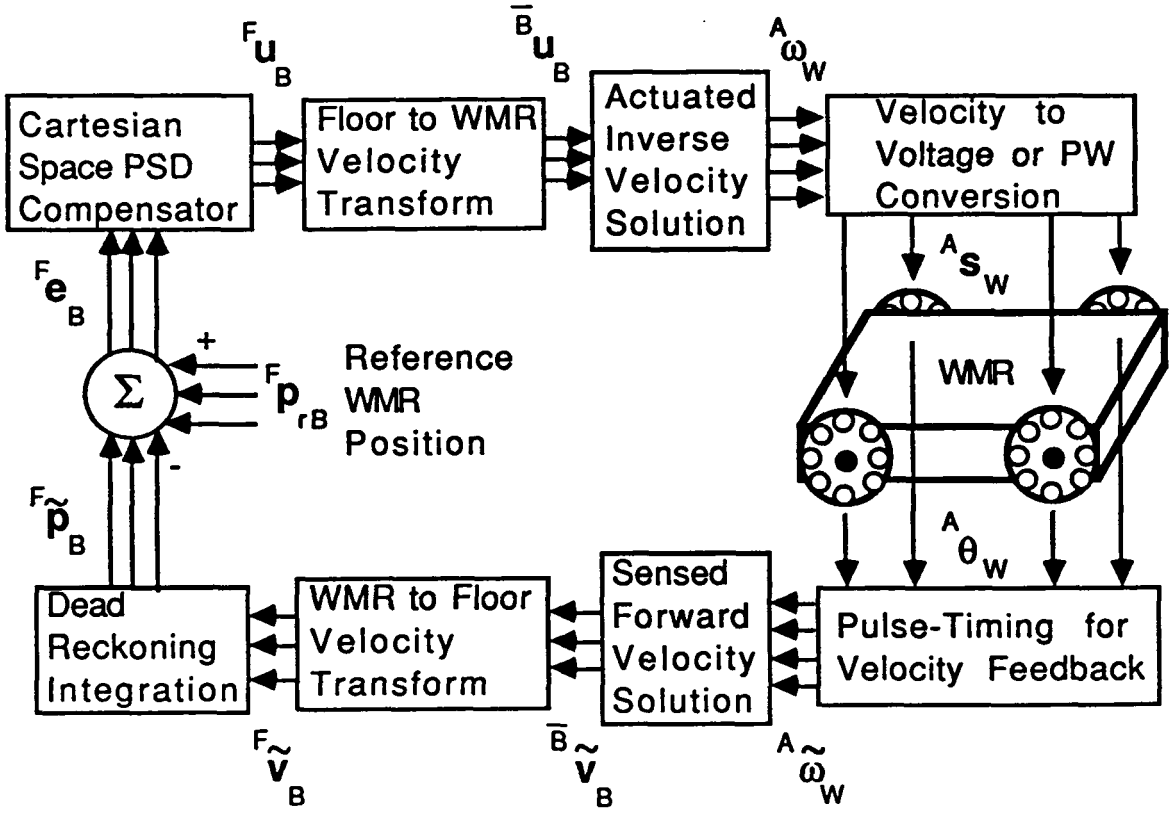


Figure 4.2: Resolved Motion Rate WMR Servo-Control

The control signal ${}^{F}\mathbf{u}_{B}$ is then transformed to the WMR body coordinate system by the inverse of the motion matrix in (2.3.34). The resulting control signal ${}^{B}\bar{\mathbf{u}}_{B}$ is treated as the desired WMR velocity. This WMR velocity is transformed into actuator velocities by applying the actuated inverse velocity solution in (2.4.8). The resulting wheel velocities ${}^{A}\omega_{W}$ are then converted by the velocity-to-voltage command conversion in (4.4.1) or the velocity-to-pulse-width command conversion in (4.4.2) to actuate the wheel motors utilizing voltage or PWM control, respectively.

We experimentally select the proportional and difference gains k_{pi} and k_{di} independently for each axis of motion. We first choose the gains for the x-axis of motion by conducting simulated step response experiments (which are described in Chapter 5). The position gain k_{px} with zero difference gain, $k_{dx}=0$, is adjusted to provide a fast response. The difference gain is then increased to eliminate overshoot in the x-axis. The gains for the y-axis motion are set equal to the x-axis gains to provide for homogeneous response characteristics along both translational axes. The proportional and difference gains for the rotational motion θ are chosen in a similar manner by simulating a step θ trajectory. The summation component of the PSD algorithm in (4.2.1) introduces overshoot in the step responses of the two prototype WMRs evaluated in Chapters 6 and 7 without providing commensurate reductions in the steady-state error. The summation gains are therefore chosen to be zero. We apply this gain selection procedure in Chapter 6 to design a resolved motion rate servo-controller for Uranus.

Since the feedback for the resolved motion rate servo-controller is in Cartesian space, this controller is capable of compensating for WMR body position errors. Since, however, the controller is based entirely upon kinematic information, this controller cannot compensate for dynamic or frictional forces/torques as is shown in Chapters 6 and 7.

4.3 Resolved Acceleration WMR Servo-Control

To compensate for tracking errors introduced by dynamic forces/torques and friction, we apply the WMR dynamic model (developed in Chapter 3) to design a resolved acceleration WMR servo-controller. The resolved

acceleration WMR servo-controller shown in Figure 4.3 is a Cartesian-space dynamics-based servo-controller design. The feedback path is identical to that for the resolved motion rate WMR servo-controller, utilizing the sensed forward velocity solution to estimate the WMR body velocity from the sensed wheel velocities and dead reckoning to estimate the WMR body position. The WMR body position error ${}^{F}e_{B}$ and velocity error ${}^{F}\dot{e}_{B}$ are computed and multiplied by the position and velocity gains k_p and k_d, respectively, and the acceleration references ${}^{F}a_{rB}$ are fed-forward. In contrast to the resolved motion rate servo-controller gains, we select equal gains for all axes of motion in the resolved acceleration controller.

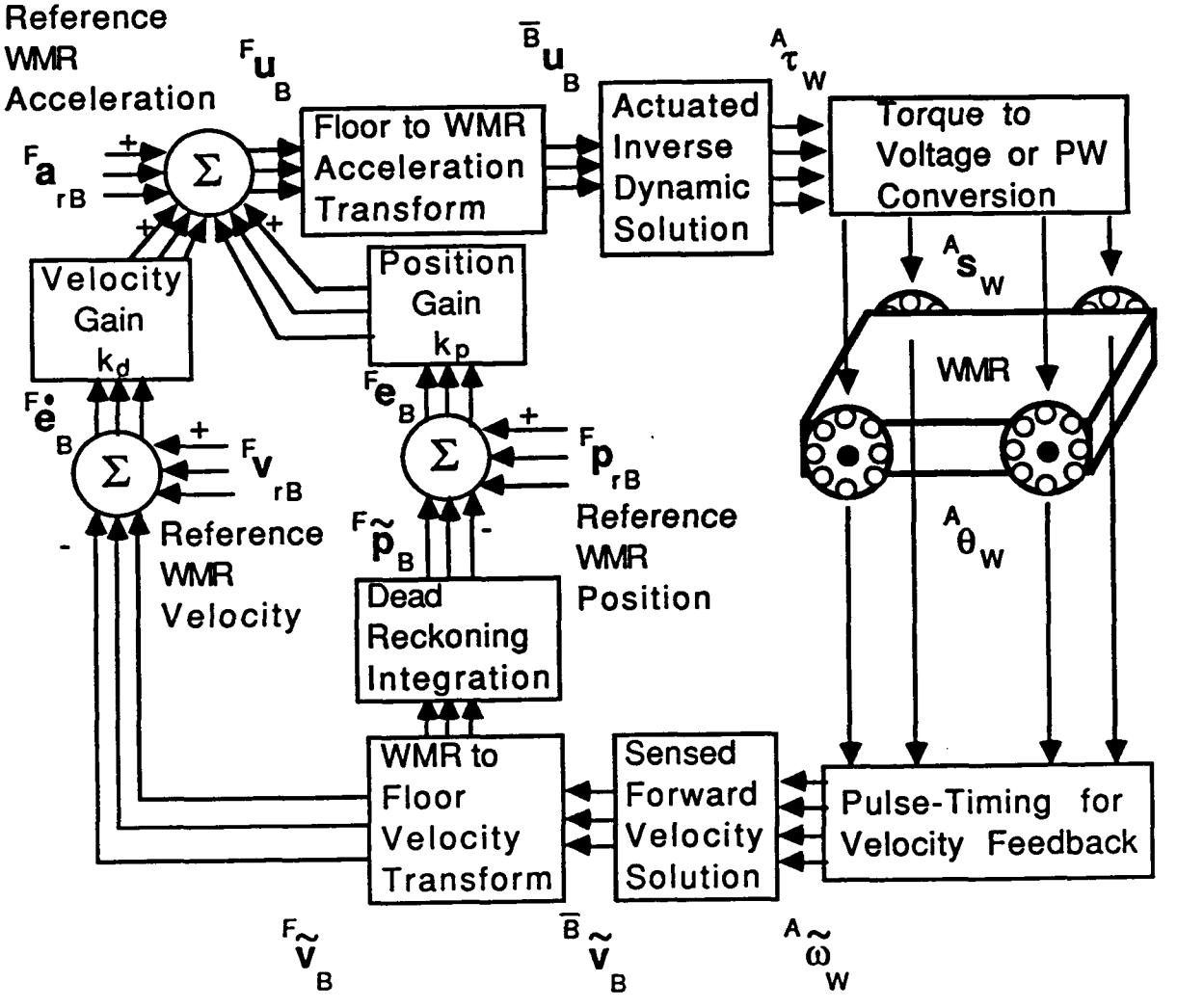


Figure 4.3: Resolved Acceleration WMR Servo-Control

The position and velocity gains k_p and k_d are chosen to provide a critically damped response. The x-axis of motion is adjusted first with the aid of the WMR step response simulation described in Chapter 5. The position gain is increased until no further reduction in settling time is obtained, and the velocity gain is chosen as $k_v=2\sqrt{k_p}$ to provide critical damping [Tourassis85].

The WMR reference position, velocity, and acceleration trajectories are provided by the trajectory planner. The sum ${}^{F}\mathbf{u}_B$ of the position and velocity errors and the reference acceleration is transformed to the WMR body coordinate system by the inverse of the motion matrix in (2.3.34) and interpreted as the desired WMR accelerations. These WMR accelerations ${}^{B}\dot{\mathbf{u}}_B$ are applied to compute the actuated inverse dynamic solution (described in Section 3.5). The resulting actuator torque commands are then converted to motor voltages or PWM according to the torque-to-voltage command conversion in (4.4.4) or the torque-to-pulse-width command conversion in (4.4.5).

The actuated inverse dynamic WMR solution is applied in the feedforward path of the resolved acceleration servo-controller to cancel the inherent dynamics of the WMR. The WMR body will thus follow the reference feedforward acceleration trajectory exactly, if the dynamic model is exact, there are no unmodeled forces/torques, and there are no unmodeled external disturbances. Since, in practice, we cannot guarantee that the dynamic model is exact and that there are no unmodeled forces/torques or external disturbances, the feedback position and velocity paths are incorporated in the servo-controller to null the system tracking errors caused by these effects. The WMR position and velocity errors are then summed with the reference acceleration trajectory.

The resolved acceleration servo-controller design is more computationally complex than resolved motion rate servo-controller design (as quantified in Chapters 6 and 7) and requires the trajectory planner to compute reference positions, velocities, and accelerations. Knowledge of the dynamic and frictional parameters is also required to implement resolved acceleration control. The advantages are that the resolved acceleration approach can compensate for dynamic and frictional forces/torques, and the design of the gains is straightforward. We design and evaluate resolved acceleration servo-controllers for Uranus in Chapter 6 and Bicsun-Bicas in Chapter 7.

4.4 Actuator Command Conversions

The wheel actuators of the prototype WMR designs in Chapters 6 and 7 are voltage controlled brushless DC motors. However, the actuator commands computed by the resolved motion rate WMR servo-controller in Section 4.2 are motor velocities and the actuator commands computed by the resolved acceleration WMR servo-controllers in Section 4.3 are motor torques. We therefore include a conversion algorithm within our servo-control systems to interface the servo-controller commands to the applied voltage to the motors. We desire pulse-width modulation (PWM) control to take advantage of the efficiency of the PWM [Muir85].

We consider the two types of motor control: voltage and PWM. For voltage control, the voltage v_m is applied to the DC motor for the entire duration of the sampling period T. For PWM control, as shown in Figure 4.4, the maximum voltage v_{max} is applied to the DC motor for the length of time equal to the pulse width s which cannot exceed the sampling period. To accommodate both kinematics and dynamics-based servo-controllers, and voltage and PWM control, we develop *four* motor command conversion algorithms: (1) velocity-to-voltage, (2) velocity-to-pulse-width, (3) torque-to-voltage, and (4) torque-to-pulse-width.

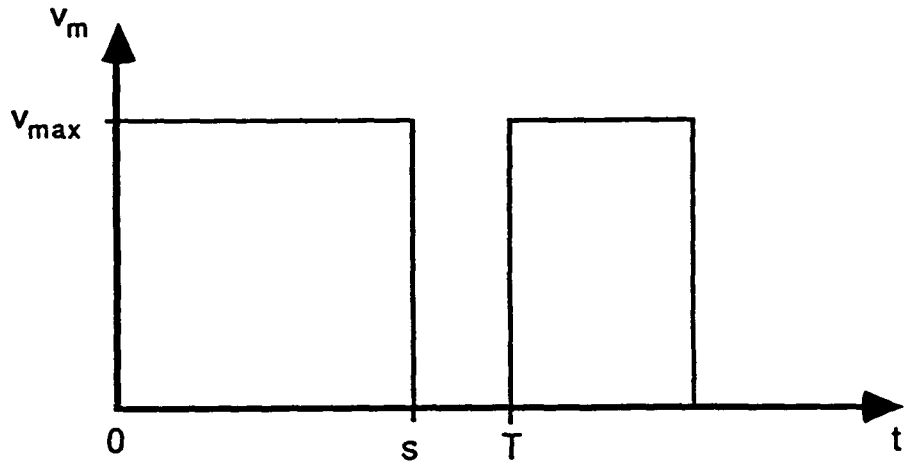


Figure 4.4: Pulse-Width Modulation

To apply the output of a kinematics-based servo-controller to a WMR with voltage controlled DC motors requires a velocity-to-voltage conversion

algorithm. We have discovered, through the computer simulations described in Chapter 5, that adequate servo-control is achieved by the one-to-one command conversion:

Velocity to Voltage Command Conversion

$$v_m = k_1 \omega_c \qquad (4.4.1)$$

where ω_c is the computed actuator angular velocity, and the unity gain k_1=1volt-second.

We develop a voltage-to-pulse-width algorithm by equating the average voltage in one sampling period for the voltage controlled and PWM scenarios. The average voltage in one sampling period for voltage control is equal to the applied voltage v_m and for PWM is (v_{max}s/T). Equating the average voltage in the two cases and solving for the pulse-width yields the voltage-to-pulse-width algorithm:

Velocity to Pulse-Width Command Conversion

$$s = \left(\frac{T\, k_1}{v_{max}}\right)\omega_c . \qquad (4.4.2)$$

The commanded pulse-width s in (4.4.2) must be limited to a maximum value of T remain within the present sampling period.

We apply Kirchhoff's voltage law to the conventional model of an armature-controlled DC motor [Kuo82] to form the foundation of the torque-to-voltage computations:

$$v_m = R_m i_m + L_m \frac{di_m}{dt} + k_b \omega_m . \qquad (4.4.3)$$

In (4.4.3), i_m is the armature current, R_m is the armature resistance, L_m is the armature inductance, k_b is the motor back emf constant, and ω_m is the motor angular velocity. To simplify the development, we assume negligible armature inductance (i.e., L_m=0). This assumption is equivalent to assuming that the

electrical time constant of the motor is much smaller than the mechanical time constant. Since the effective time constants of our prototype WMRs in Chapters 6 and 7 are 2 seconds, this assumption is realistic.

We develop a torque-to-voltage conversion algorithm to apply the resolved acceleration servo-controllers with voltage controlled DC motors. The torque generated by a DC motor is $\tau_m = k_t i_m$ where k_t is the motor torque constant. By substituting the armature current into the armature circuit equation in (4.4.3), we obtain the torque-to-voltage conversion algorithm:

Torque to Voltage Command Conversion

$$v_m = \left(\frac{R}{k_t}\right)\tau_m + k_b\omega_m . \qquad (4.4.4)$$

We develop a torque-to-pulse-width algorithm to implement PWM with resolved acceleration servo-control. The torque-to-pulse-width algorithm is computed by substituting the motor voltage in (4.4.4) into the velocity-to-voltage algorithm in (4.4.1) and the result into (4.4.2):

Torque to Pulse-Width Command Conversion

$$s = \left(\frac{T\ R}{v_{max}\ k_t}\right)\tau_m + \left(\frac{T\ k_b\ \omega_m}{v_{max}}\right). \qquad (4.4.5)$$

These four command conversion algorithms are computationally simple and thus require negligible computation in comparison with the resolved motion rate and resolved acceleration servo-control algorithms of Sections 4.2 and 4.3.

4.5 Summary

In this chapter, we have applied stationary manipulator servo-control approaches to WMR servo-control. Kinematics-based resolved motion rate WMR servo-control (described in Section 4.2) compensates for WMR tracking errors, and has low computational requirements, but does not compensate for dynamic or frictional forces/torques. The more computationally complex

dynamics-based resolved acceleration WMR servo-controller (described in Section 4.3) compensates for dynamic and frictional forces/torques, and requires knowledge of the dynamic and frictional parameters, and the position, velocity and acceleration reference trajectories. If adequate WMR control is attainable without compensation of dynamic and frictional forces/torques, we suggest that the computationally simpler resolved motion rate servo-controller is appropriate for WMR control. If dynamic and frictional forces/torques are significant for a specific WMR, resolved acceleration servo-control is required.

In Chapters 6 and 7, we evaluate the resolved motion rate and resolved acceleration WMR servo-controllers and the four actuator command conversion algorithms for the two prototype WMRs Uranus and Bicsun-Bicas. Since WMRs are closed-chain structures, all components are kinematically coupled under the assumption of no translational wheel slip. Consequently, when any component undergoes motion, all components move, and the WMR time constant, that of the main body of the WMR, is large (2 seconds for our prototype WMRs in Chapters 6 and 7). We exercise the rule-of-thumb for choosing the servo-controller sampling period to be one-tenth of the WMR time constant. We show through computer simulations in Chapters 6 and 7 that this choice of sampling period is sufficient for adequate WMR servo-control.

Chapter 5
Computer Simulation for WMR Servo-Controller Evaluation

5.1 Introduction

We simulate WMR kinematics and dynamics to evaluate the relative performance of model-based WMR servo-controller designs since an operational WMR is not available for hardware servo-controller implementation and evaluation. WMR kinematics and dynamics simulation, moreover, is less costly and more flexible than hardware implementation. Computer simulation packages, such as VAST (Versatile Arm Simulation Tool) [Pfeifer84], are available for stationary manipulator dynamics simulation. VAST (written in the C programming language) simulates the closed-form Lagrangian-Euler dynamic model of a stationary manipulator [Bejczy74]. We require the simulation of WMR dynamic models formulated symbolically according to our dynamics framework in Chapter 3 by the symbolic manipulation program MACSYMA [Symbolics85]. The application of a dynamics simulation package, such as VAST, to our WMR servo-control studies thus presents the problem of adapting the simulation package to accept dynamic models formulated symbolically according to our dynamics framework in MACSYMA. We have not attempted the major software engineering effort of adapting VAST. Instead, we have programmed a dynamic WMR numerical simulation package within MACSYMA. The heart of the WMR simulation package is the *fourth-order* Runge-Kutta integration routine from VAST which we have re-coded for numerical execution in MACSYMA.

For WMR dynamic simulation, we apply a Vax[1] 11/8800 computer to mimic both the dynamics of the WMR and the computations of the WMR servo-controller. In Section 5.2, we describe the functionality of the WMR dynamic simulator. The simulation parameters (discussed in Section 5.3) are such parameters as the trajectory type and WMR load which can be changed from simulation to simulation. By changing the simulation parameters and observing the resulting changes in the simulated WMR motion, we evaluate the relative performance of WMR servo-controllers. In Section 5.4, we describe the performance measures applied to evaluate servo-controllers. Finally, in Section 5.5, we summarize our WMR simulation methodology.

We couple our WMR simulation package with the servo-controllers designed in Chapter 4 to evaluate their performance in Chapters 6 and 7 on two prototype WMRs. The servo-controller designs and the WMR simulator are founded upon the kinematic and dynamic WMR models developed in Chapters 2 and 3.

5.2 Simulator Functionality

There are *four* computational components of WMR computer simulation: the trajectory planner, the servo-controller algorithm, the forward dynamic solution and the integration routine. The execution sequence of these components is illustrated in Figure 5.1.

Two functional loops are represented in Figure 5.1. The *outer* loop is executed once each sampling period. The *inner* loop is executed once each integration step. At the beginning of each sampling period, the reference WMR trajectory (the three dimensional position, velocity and acceleration) is computed[2] by the trajectory planner. The actuator commands are then computed by the servo-controller algorithm. The inner loop is executed several times (typically ten or more) to simulate the WMR motion for each sampling period. Each pass through the inner loop is an integration step. In each

---

1 Vax is a trademark of the Digital Equipment Corporation.

2 In the actual program, the entire WMR reference trajectory is computed and stored in a table before the simulation begins. The reference points for each sampling period are found by indexing into the table.

integration step, we compute the WMR body three-dimensional acceleration by the forward dynamic solution (described in Section 3.5.2) and then apply the Runga-Kutta integration routine twice to compute the three-dimensional WMR body velocity and position. We apply the fourth-order Runge-Kutta integration routine from VAST[3]. The integration routine adapts the step size to ensure uniformly accurate results during periods of quickly varying and smooth motions. The Collatz ratio is set to 0.25 according to the guidelines set by [Pfeifer84]. After enough integration steps have been executed to complete one sampling period, we store the WMR body positions, velocities and accelerations for future reference. We also compute performance indices (detailed in Section 5.4) which are required for our servo-controller evaluation studies.

The actuation and floor reactional forces/torques within our simulation are monitored to ensure realistic conditions. Since actual wheel actuators have an upper torque limit, we must ensure that the numerical values of the actuator torques within our simulation never exceed the maximum. When the value of the computed actuator torque does exceed the maximum, we limit our simulated torque values to the maximum value. A similarly unrealistic condition arises when the floor reactional forces on any of the wheels becomes negative. This represents the anomalous effect of the floor pulling down on the wheels, and occurs when the WMR accelerates too quickly (i.e., the WMR does a wheelie). When this negative floor reactional force condition occurs, there is no easy and realistic solution. We thus incorporate a provision to invalidate a simulation in which this condition occurs. None of the simulation experiments with our prototype WMRs in Chapters 6 and 7 encounter negative floor reactional forces.

The dynamic models in Chapter 3 and the servo-control algorithms in Chapter 4 are applied directly to our WMR simulation experiments except for the following two notable exceptions; the exclusion of stiction in the simulation and dead reckoning in the servo-controller. The secondary equation-of-motion written for a joint DOF which has stiction becomes trivial (i.e., 0=0) when the stiction is acting. The number of independent equations-of-motion in the dynamic model is reduced by one, and the solution of the model must then be recomputed symbolically. It is prohibitive computationally to incorporate

---

[3] The forward dynamic solution is actually computed four times in each integration step of the fourth-order Runge-Kutta algorithm.

numerous forward and actuated inverse dynamic solutions corresponding to each combination of sticking and non-sticking joints in our simulation. We therefore neglect stiction in our simulation experiments. Secondly, the dead reckoning integration required in the feedback path of the servo-controllers in Chapter 4 is redundant in the simulation and computationally expensive. The simulation program computes the position of the WMR from the body accelerations. The dead reckoning integration for a servo-controller computes the position of the WMR from the sensed wheel velocities. Since our intent is to compare servo-control algorithms, and not dead reckoning algorithms, we utilize the simulated WMR position and velocity within the feedback path of the servo-controller. This allows us to isolate the performance of each servo-controller from that of the dead reckoning computations. In actual application, care must be taken to ensure accurate dead reckoning, since the tracking performance of the servo-controller is limited by its own estimate of the WMR position.

The simulation program consists of approximately 1000 lines of MACSYMA program code. We run our simulations on a Vax 11/8800 under the Unix[4] operating system. We typically simulate 10 seconds of real-time during one WMR experiment. The actual run time of a simulation varies according to the load on the time-shared mainframe Vax and the particular simulation parameters (described in Section 5.3). Typical run times vary between 20 and 200 cpu minutes.

A WMR computer simulation is accurate to the extent that the simulated WMR motions correspond to the actual motions of the WMR. If the WMR were operational, there would be little reason for the simulation. Since the WMR is not operational, we cannot verify the computer simulation with reality. WMR simulation alone is not our goal. Rather, our purpose is to evaluate the *relative* performance characteristics of different WMR servo-controllers and not absolute performance specifications. We thereby answer general questions about WMR servo-control (e.g., Is a kinematics-based servo-controller sufficient for three DOF WMR tracking?) rather than absolute performance (e.g., What is the tracking accuracy, in centimeters, of a particular WMR/servo-controller combination?).

---

4 Unix is a Trademark of AT&T Bell Laboratories.

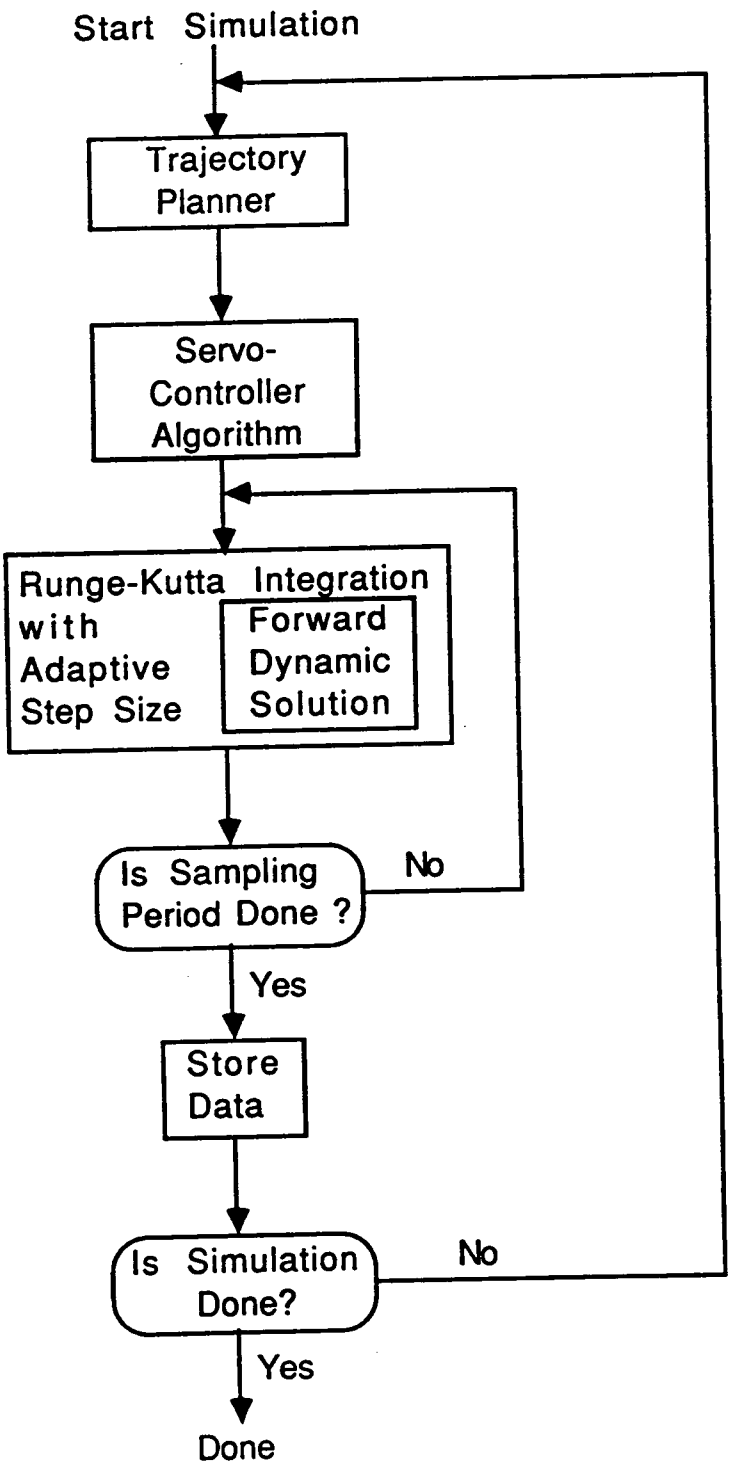


Figure 5.1: Flowchart of Simulator Functionality

We have strived to ensure the accuracy of our simulations. The dynamic model developed in Chapter 3 is based upon conceptually proven component models, including: Newton's laws, Euler's equations, Coulomb's law and the link Jacobian. Parameter identification is beyond the scope of this dissertation; however, we have utilized actual measurement and design specifications to obtain the kinematic and dynamic parameters of the prototype WMR Uranus for our simulations described in Chapter 6. We have checked the forward and actuated inverse dynamic solutions by substituting symbolically the actuated inverse solution into the forward dynamic solution. The results indicate mathematical consistency by simplifying to a trivial identity (i.e., 1=1). We set all frictions to zero for this consistency check to make the computations tractable.

We also checked the simulation code by applying the actuated inverse dynamic solution, playing the role of an ideal feedforward servo-controller, in place of the servo-control algorithm in the simulation. We ran the simulation with this ideal feedforward servo-controller executing *every integration step* (rather than every sampling period) to minimize discretization errors. The resulting WMR trajectory is within 3.3×10^{-7} meters of the reference trajectory throughout the simulation. We thereby estimate that our simulations are accurate to six decimal digits. This accuracy is sufficient for our servo-controller studies.

We attribute the nonzero tracking error of our ideal feedforward simulation test to the numerical errors of floating point computations and numerical integration. The floating point numerical computations within the MACSYMA simulator are limited by the computer hardware to 8 digits (32 bits). The exact rational representation of numbers cannot be applied to all computations within the MACSYMA simulation code. The computation time for a higher-precision floating point implementation is prohibitive for simulation applications. Runge-Kutta integration is typically applied to computer simulation applications because of its inherent accuracy; however, no numerical integration algorithm is exact for the simulation of nonlinear, time-varying robot dynamics.

5.3 Simulation Parameters

Our MACSYMA simulator is designed to evaluate the relative performance of a small number of servo-controllers (two in Chapters 6 and 7) on a small number of representative WMRs (two in Chapters 6 and 7). The simulator and each servo-control algorithm is hand customized for minimum execution time for each WMR. The design and development of the simulator is not a contribution of this dissertation, but a tool for demonstrating the application of the dynamic models introduced in Chapter 3 for the servo-control of WMRs designed in Chapter 4. We have thereby traded-off ease of future application of the simulation program for a short development time. We do suggest that our simulator embodies a successful feasibility experiment, showing that WMR dynamic simulation is possible, incorporating the salient characteristics of WMRs enumerated in Section 1.5 within the MACSYMA symbolic manipulation environment. Our simulation code thus lays the foundation for the development of a general-purpose dynamic simulation package for simple closed-chain robotic mechanisms.

We have incorporated in the simulator the ability to select parameters for each simulation. Our simulation parameters are:

- Actuation type (voltage or PWM);
- Sampling period (in seconds);
- Task time (in seconds);
- Base frictional constant (dimensionless);
- Load mass (in kilograms);
- Precision of the sensor measurements and actuator commands (in bits); and
- Reference trajectory type (step, spline or circle).

In Chapters 6 and 7, we customize our simulation program for the two prototype WMRs (Uranus and Bicsun-Bicas) and for each of the two servo-control algorithms (resolved motion rate and resolved acceleration). The two WMRs can be combined with the two servo-controllers and any combination of the aforementioned simulation parameters. Each simulation parameter has an associated nominal value which represents a typical value of that parameter for actual implementation. We describe each of these simulation parameters and their nominal values in turn.

We can specify the actuation of the WMR motors to be either voltage or PWM. The appropriate actuator command conversion algorithm in Section 4.4 is then applied to the simulation. Since PWM control is an area of current research, the nominal value of the actuation parameter is voltage control. The sampling period is a simulation parameter with a nominal value of 0.2 seconds, one-tenth of the time constant of the prototype WMRs. The duration of a simulation experiment is specifiable with a nominal value of 10 seconds. This allows a sufficiently interesting simulation experiment length without an overwhelming amount of accumulated data.

We have assigned all dry frictional constants to be multiples of a single base frictional value to simplify the manipulation of frictional coefficients. The base frictional value is a simulation parameter with a nominal value of 0.01. We have assigned the Coulomb and rolling frictional coefficients to be 0.01 and 0.001 times the base frictional value, respectively. The base friction of the simulation and the base friction of the servo-controller may be adjusted independently to observe the effects of friction and the effects of inaccurate estimates of the friction on the performance of the servo-controllers. The WMR load mass, and the load mass value applied to the servo-control algorithms are both simulation parameters with nominal zero values. The simulated load may thus be adjusted to observe the sensitivity of the servo-controller performance to inaccurate knowledge about the WMR load. The precision of the sensor measurements and actuator command values is a simulation parameter with a nominal value of 16 bits. The numerical precision of these values is adjusted by truncating the computed values to the appropriate number of digits. Sensitivity to finite precision is a practical concern since actual shaft encoders have resolutions of 8 to 16 bits; digital-to-analog interfaces for motor actuation similarly have a limited precision.

The time profile of the *WMR velocity along each WMR trajectory* is diagrammed in Figure 5.2. A trapezoidal velocity profile is implemented to provide smooth transitions from the initial rest position to a constant velocity and from the constant velocity back to rest. The total distance along the trajectory d_{total} is the area enclosed by the trapezoid, and the acceleration along the trajectory is the slope of the velocity plot. In Figure 5.2, the WMR velocity along the trajectory undergoes an initial acceleration a_{ramp} from rest for the time period

t_{acc}. This is followed by a period of constant velocity v_{max}. The WMR completes the motion with a deceleration period from $(t_{final}-t_{acc})$ to t_{final} with acceleration of $-a_{ramp}$.

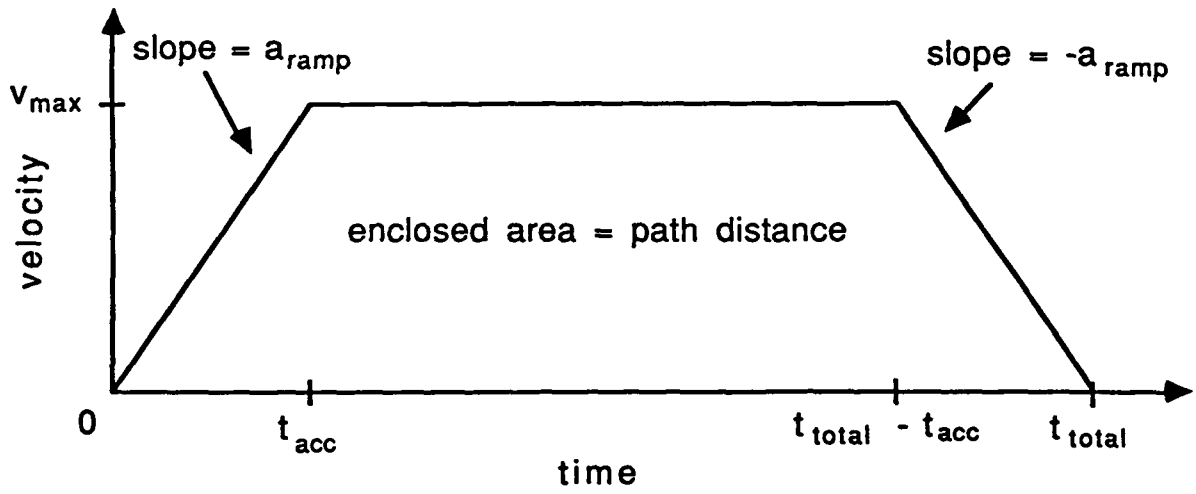


Figure 5.2: WMR Trajectory Velocity Profile

The spatial x-y reference WMR trajectory may be specified as either a step, spline or circle as shown in Figures 5.3 to 5.5, respectively. For each of the trajectories (step, spline and circle), we specify the final time t_{final}, acceleration time t_{acc}, and total distance d_{total}. The ramp acceleration a_{ramp} and maximum velocity v_{max} are then computed from plane geometry. The time progression of the WMR along the trajectory is thereby computed. The θ-rotational WMR trajectory is a simulation parameter which is assigned independently a trapezoidal time-velocity profile as shown in Figure 5.2. There are two rotational trajectories utilized in our simulation experiments: a constant orientation (i.e., t_{final}=10 seconds, t_{acc}=0 seconds and d_{total}=0 radians), and a standard rotation (i.e., t_{final}=10 seconds , t_{acc}=0.3t_{final}, and d_{total}=1.5 radians).

To execute the step trajectory in Figure 5.3, we initially position the WMR at (0,0) with an orientation angle of 0 radian. The actual motion of the WMR towards the reference line illuminates the step-response characteristics of the WMR/servo-controller combination. The characteristic parameters of the step trajectory are: t_{final}=10 seconds, t_{acc}=T and d_{total}=1.0 meter, where T is the constant sampling period.

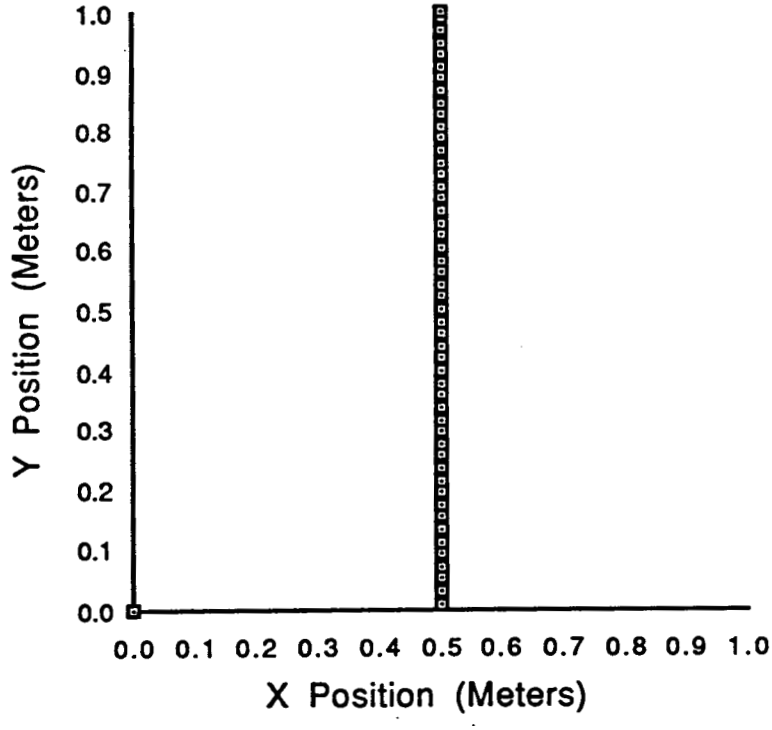


Figure 5.3: Step Trajectory

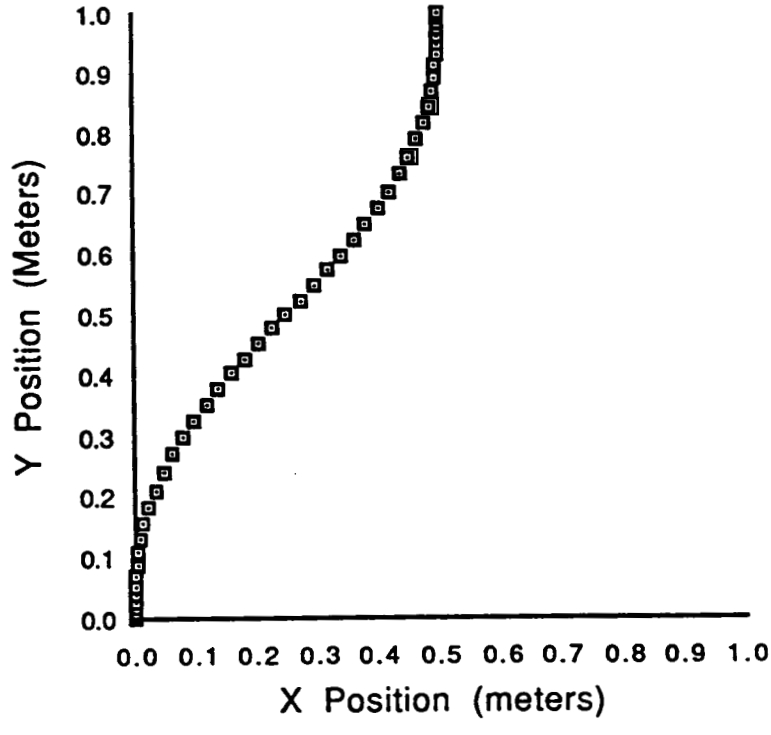


Figure 5.4: Spline Trajectory

The spline trajectory in Figure 5.4 is a fifth-order polynomial function with initial position (0,0), final position (0.5,1.0), and infinite first and second initial and final spatial derivatives. The WMR velocity along the trajectory has the trapezoidal shape shown in Figure 5.2 with the following parameter values: t_{final}=10 seconds, t_{acc}=0.3t_{final} and d_{total}=distance along the spline in meters. The motion of the WMR in response to the spline reference trajectory illuminates the tracking characteristics of the WMR/servo-controller combination.

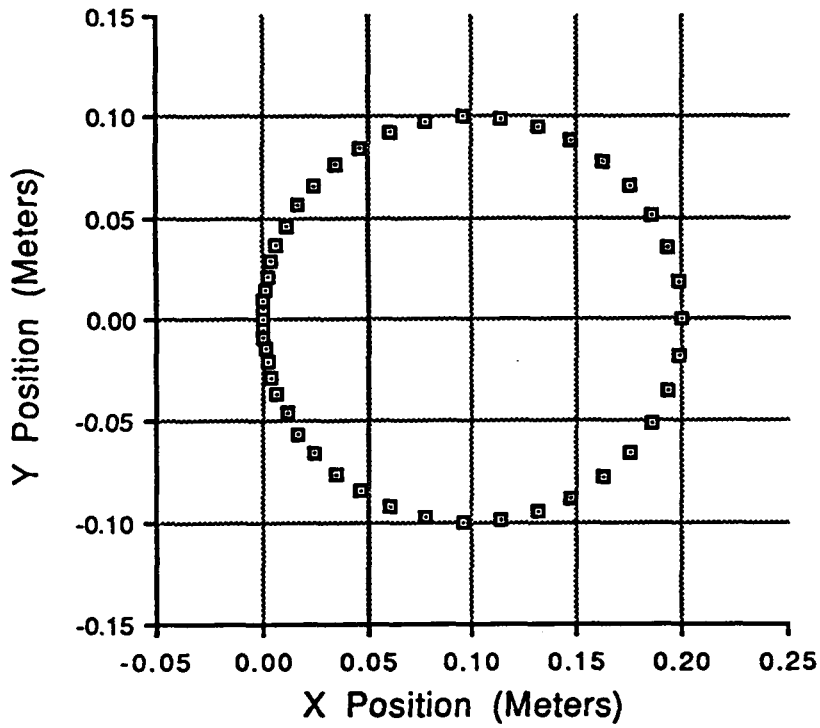


Figure 5.5: Circle Trajectory

The circle trajectory is a geometric circle of radius 0.1 meter and center (0.1,0). The WMR is initiated at (0,0) and proceeds clockwise around the circle. The WMR velocity along the trajectory has the trapezoidal shape shown in Figure 5.2 with the following parameter values: t_{final}=10 seconds, t_{acc}=0.3t_{final} and d_{total}=0.2π meters. The motion of the WMR in response to the circle reference trajectory illuminates the temporal lag in the tracking characteristics of the WMR/servo-controller combination. In contrast to the spline trajectory, which requires small, slow variations of the WMR direction of motion, the circle trajectory requires larger, continuous variations of the WMR direction of motion from 0 to 2π radians. A WMR temporal lag for the spline trajectory results in little

deviation from the spatial path because the variations in the direction of motion are small. For the circle trajectory, any WMR temporal lag is graphically apparent as deviations from the reference WMR path due to the larger continuous variations in the direction of motion.

5.4 Servo-Controller Performance Measures

We store the WMR body positions, velocities and accelerations at each sampling instant so that we may plot and observe the results after the simulation is completed. We evaluate the relative performance of the WMR servo-controllers from these data and practical considerations. The performance measures according to which we evaluate the two servo-controllers are:

- Number of computations;
- Maximum spatial error (in meters);
- Maximum rotational error (in radians);
- Accumulated error (in meter-seconds);
- Overshoot (in meters);
- Settling time (in seconds);
- Steady-state spatial error (in meters);
- Number of actuation saturations; and
- *A-priori* information required.

The number of computations, which is the number of scalar multiplications and additions required to implement the servo-control algorithm, indicates the computational complexity of the control algorithm. We tabulate (in Chapters 6 and 7) the computations required both when friction is incorporated and neglected in the servo-controller algorithm. The maximum spatial error is the maximum deviation of the WMR from the reference trajectory in the x-y plane of the floor. The maximum rotational error is similarly the maximum deviation of the WMR orientation from the reference orientation. The maximum spatial and rotational errors are appropriate performance measures for the spline and circle trajectories. The accumulated error is the square root of the sum of the squares of the spatial errors in the x and y directions summed throughout the simulation experiment and multiplied by the sampling period. We normalize the accumulated error by the sampling period to compare accumulated error values of simulations having differing sampling periods. The overshoot, settling time (95%), and steady-state spatial error are appropriate performance measures for

the step trajectory. The number of actuator saturations gives an indication of the smoothness of the servo-control. The *a priori* information required for the design of a servo-controller indicates the computational complexity and engineering effort of the servo-controller design process.

5.5 Summary

We have implemented a MACSYMA-based WMR simulator which utilizes the kinematic and dynamic WMR models formulated in Chapters 2 and 3. The simulation parameters: actuation type, sampling period, task time, base frictional constant, load mass, numerical precision and reference trajectory type can be adjusted from simulation to simulation. WMR/servo-controller combinations are evaluated (in Chapters 6 and 7) according to nine performance measures: number of computations, maximum spatial error, maximum rotational error, accumulated error, overshoot, settling time, steady-state spatial error, number of command saturations and *a-priori* information.

In Chapters 6 and 7, we evaluate four WMR/servo-controller combinations: Uranus and Bicsun-Bicas each with resolved motion rate and resolved acceleration servo-controllers. For each WMR/servo-controller combination, we begin our evaluation with an experiment having all simulation parameters set to their nominal values. The effect of each simulation parameter on the WMR/servo-controller performance is then illuminated by systematically varying one or more of the simulation parameters from their nominal values and observing the change in the performance measures.

An ideal WMR development project incorporates WMR simulation as a precursor to mechanical construction. Proposed WMR mechanical designs are simulated and proposed servo-controller designs are evaluated interactively. The simulation allows the flexibility of changing easily mechanical and servo-controller parameters to enhance the combined performance of the WMR/servo-controller system. The ultimate performance of the WMR can then be assessed before effort and time are expended on construction. In this scenario, it would be beneficial to utilize the exact machine code (e.g., C code) in the simulated servo-controller to be utilized on the WMR, thus avoiding the step of converting the simulated servo-controller code to the target WMR servo-control processor.

Similarly, a computer-aided design (CAD) package would be interfaced directly to the simulator to develop the WMR kinematic and dynamic models directly from the design database. Mechanical design changes would then be reflected automatically in the WMR/servo-controller simulation.

Chapter 6
Case Study I: Uranus, A Three DOF WMR Prototype

6.1 Introduction

In Chapters 6 and 7, we illustrate our WMR modeling and control methodologies through *case studies* of two prototype WMRs. In this chapter, we focus upon the three DOF WMR *Uranus.* Uranus is a omnidirectional WMR being designed and constructed in the Mobile Robot Laboratory of the Robotics Institute of Carnegie Mellon University [Moravec86]. Uranus has the kinematic structure of the Wheelon wheelchair [Alvema82], consisting of four omnidirectional wheels mounted at the corners of a rectangular body as pictured in Figure A2.10. According to our nomenclature in Appendix 1, Uranus has a *tetroas* structure; i.e., four (tetra) omnidirectional (o), actuated (a), and sensed (s) wheels. The symbolic representation of Uranus is diagrammed in Figure A2.9. Each *omidirectional* wheel, pictured in Figures A3.1 and A3.2, consists of a central hub with twelve free-running rollers mounted at 45° angles around the hub periphery. The profile of the rollers is such that the silhouette of the omnidirectional wheel is circular. The four wheels are actuated by brushless DC servo-motors and are sensed by optical shaft encoders. The rollers are neither actuated nor sensed. Uranus is an ideal prototype WMR for autonomous mobile transport applications on smooth, hard surfaces.

To model symbolically the kinematics of Uranus in Section 6.2, we apply the development of Chapter 2. Then, in Section 6.3, we apply our development of Chapter 3 to model symbolically the dynamics of Uranus. The symbolic kinematic equations-of-motion in Section 6.2 are incorporated with the symbolic dynamic equations-of-motion in Section 6.3 to formulate the composite WMR

dynamic model for simulation and control applications. In Section 6.4, we tabulate the numerical parameter values utilized to simulate Uranus' operation. We design model-based servo-controllers for Uranus in Section 6.5 by applying the WMR controller design methodology of Chapter 4. In Section 6.6, we compile the results of the simulation experiments which we conducted for servo-controller evaluation studies (as described in Chapter 5). We interpret the simulation results in Section 6.7 to evaluate the relative performance of the servo-controllers. Finally, in Section 6.8, we advance our conclusions and Uranus servo-controller recommendations.

6.2 Kinematic Modeling

6.2.1 Overview

We model the kinematics of Uranus to illustrate the application of our kinematic modeling procedure in Chapter 2. We begin in Section 6.2.2 by assigning coordinate systems to Uranus. In Section 6.2.3, we compute the homogeneous coordinate transformation matrices which transform positions between coordinate systems. Then, in Section 6.2.4, we formulate the wheel Jacobians to compute the wheel velocity and acceleration equations-of-motion, which together form the kinematic model. We characterize the mobility of Uranus in Section 6.2.5 by applying the mobility, actuation and sensing characterization trees in Appendix 7. In Section 6.2.6, we compute the kinematic solutions; specifically, the complete inverse velocity solution, the actuated inverse velocity solution, the complete inverse acceleration solution, and the sensed forward velocity solution. We apply our kinematic model to compute a translational wheel slip detection algorithm in Section 6.2.7 and a dead reckoning algorithm in Section 6.2.8. In Section 6.2.9, we summarize the salient features of the kinematic modeling of Uranus.

6.2.2 Coordinate System Assignments

The Uranus coordinate systems are assigned in Figure 6.1 according to our kinematic modeling coordinate system assignments in Table 2.2.

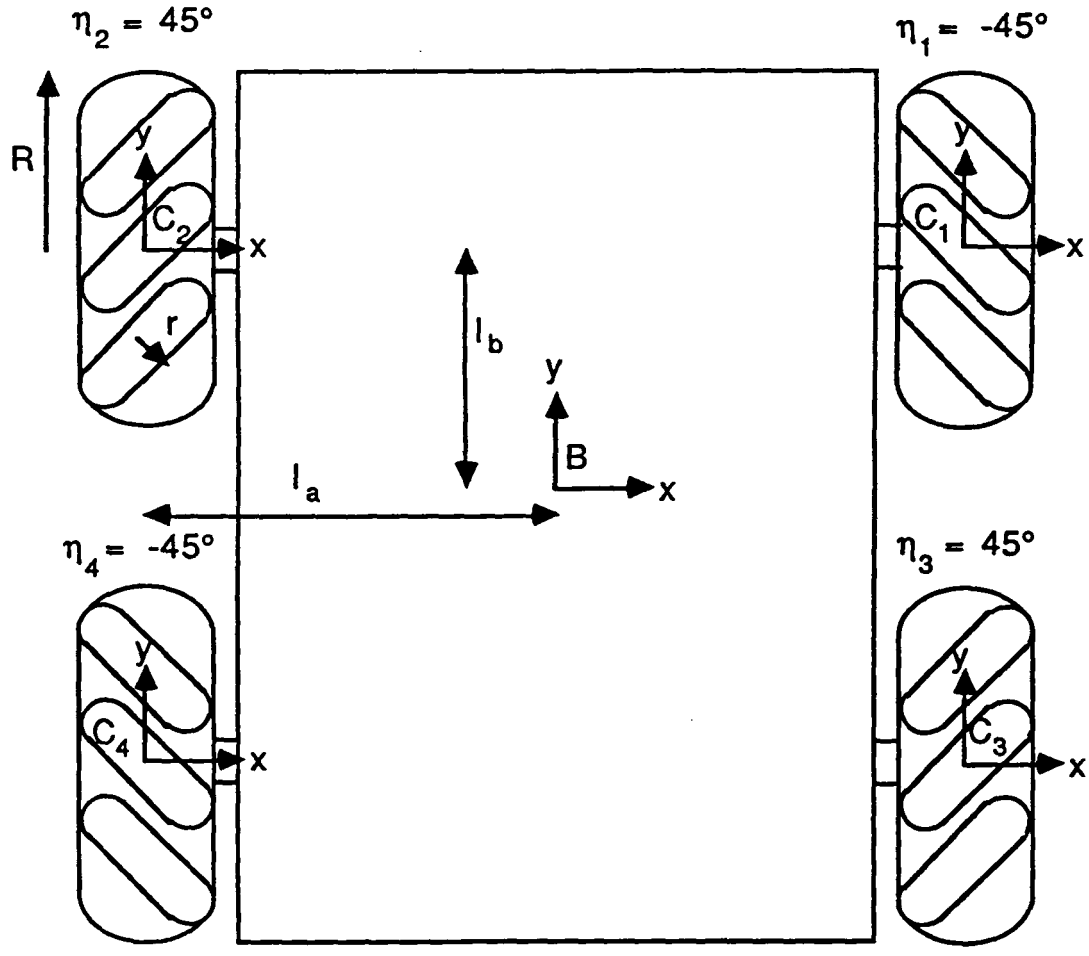
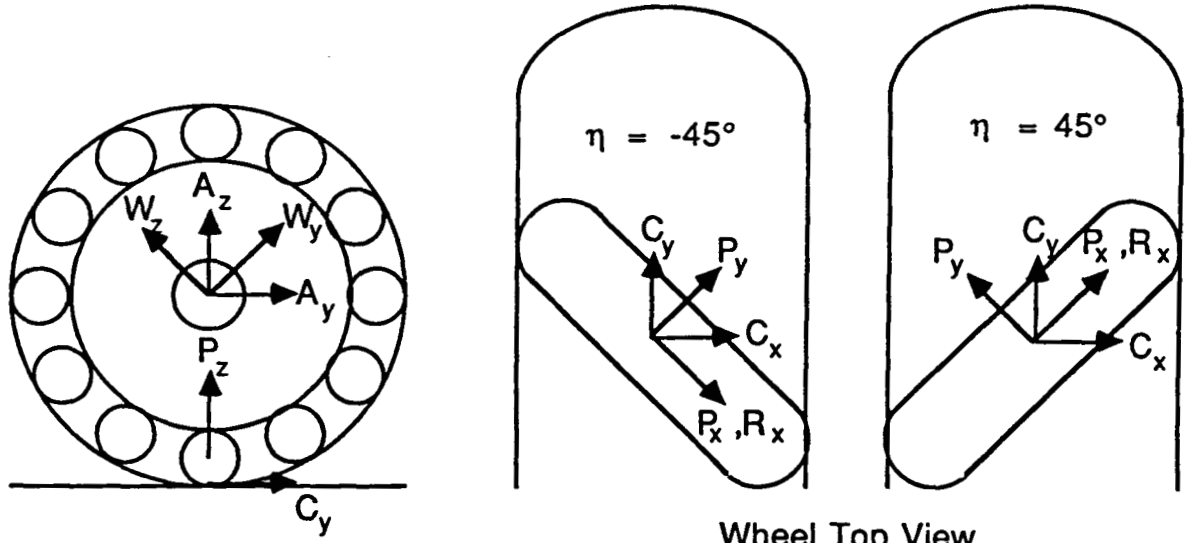


Figure 6.1: Uranus Kinematic Coordinate System Assignments

The body coordinate system B is assigned at the geometric center of Uranus and the contact point coordinate systems C_i for i=1,...,4 are assign at the wheel points-of-contact. Since there are no steering links, the hip and steering coordinate systems are not required. The wheel and roller radii assignments are $R_1=R_2=R_3=R_4=R$ and $r_1=r_2=r_3=r_4=r$, respectively; and the roller angles are $\eta_1=\eta_4=-45°$, and $\eta_2=\eta_3=45°$. The axle, wheel, pivot and roller coordinate systems A_i, W_i, P_i, and R_i for i=1,...,4, which are required for formulating the wheel equations-of-motion and the dynamic model as specified in Table 3.1, are shown in Figure 6.2.



Wheel Side View

Wheel Top View
Showing Roller Touching Floor

Figure 6.2: Additional Uranus Coordinate System Assignments

6.2.3 Coordinate Transformation Matrices

The coordinate transformation matrices required for the kinematic modeling of Uranus are:

$$ {}^{B}T_{C_1} = \begin{pmatrix} 1 & 0 & 0 & l_a \\ 0 & 1 & 0 & l_b \\ 0 & 0 & 1 & -l_c \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad {}^{B}T_{C_2} = \begin{pmatrix} 1 & 0 & 0 & -l_a \\ 0 & 1 & 0 & l_b \\ 0 & 0 & 1 & -l_c \\ 0 & 0 & 0 & 1 \end{pmatrix} $$

$$ {}^{B}T_{C_3} = \begin{pmatrix} 1 & 0 & 0 & -l_a \\ 0 & 1 & 0 & -l_b \\ 0 & 0 & 1 & -l_c \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad {}^{B}T_{C_4} = \begin{pmatrix} 1 & 0 & 0 & l_a \\ 0 & 1 & 0 & -l_b \\ 0 & 0 & 1 & -l_c \\ 0 & 0 & 0 & 1 \end{pmatrix}. $$

6.2.4 Wheel Equations-of-Motion

From the coordinate transformation matrices and the Jacobian matrix template for omnidirectional wheels in (A3.7) we write the velocity equation-of-motion for each wheel:

$$\dot{\mathbf{p}} = \begin{pmatrix} 0 & -r\sqrt{2}/2 & l_b \\ R & -r\sqrt{2}/2 & -l_a \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} {}^{A_1}\omega_{W_1x} \\ {}^{P_1}\omega_{R_1x} \\ {}^{\bar{C}_1}\omega_{C_1z} \end{pmatrix} = \mathbf{J}_1 \dot{\mathbf{q}}_1 \qquad (6.2.1)$$

$$\dot{\mathbf{p}} = \begin{pmatrix} 0 & r\sqrt{2}/2 & l_b \\ R & -r\sqrt{2}/2 & l_a \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} {}^{A_2}\omega_{W_2x} \\ {}^{P_2}\omega_{R_2x} \\ {}^{\bar{C}_2}\omega_{C_2z} \end{pmatrix} = \mathbf{J}_2 \dot{\mathbf{q}}_2 \qquad (6.2.2)$$

$$\dot{\mathbf{p}} = \begin{pmatrix} 0 & -r\sqrt{2}/2 & -l_b \\ R & -r\sqrt{2}/2 & l_a \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} {}^{A_3}\omega_{W_3x} \\ {}^{P_3}\omega_{R_3x} \\ {}^{\bar{C}_3}\omega_{C_3z} \end{pmatrix} = \mathbf{J}_3 \dot{\mathbf{q}}_3 \qquad (6.2.3)$$

$$\dot{\mathbf{p}} = \begin{pmatrix} 0 & r\sqrt{2}/2 & -l_b \\ R & -r\sqrt{2}/2 & -l_a \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} {}^{A_4}\omega_{W_4x} \\ {}^{P_4}\omega_{R_4x} \\ {}^{\bar{C}_4}\omega_{C_4z} \end{pmatrix} = \mathbf{J}_4 \dot{\mathbf{q}}_4 \qquad (6.2.4)$$

The WMR body velocity three-vector is $\dot{\mathbf{p}} = \left[{}^{\bar{B}}v_{Bx} \quad {}^{\bar{B}}v_{By} \quad {}^{\bar{B}}\omega_{Bz}\right]^T$.

We apply the wheel Jacobians $\mathbf{J}_i$ for i=1,...,4 in (6.2.1)-(6.2.4) and the wheel acceleration equation in (2.3.30) to compute the wheel acceleration equations-of-motion:

$$\ddot{\mathbf{p}} = \mathbf{J}_1 \ddot{\mathbf{q}}_1 + \left[l_a \quad l_b \quad 0\right]^T {}^{\bar{C}_1}\omega_{C_1z}^2 \qquad (6.2.5)$$

$$\ddot{\mathbf{p}} = \mathbf{J}_2 \ddot{\mathbf{q}}_2 + \left[-l_a \quad l_b \quad 0\right]^T {}^{\bar{C}_2}\omega_{C_2z}^2 \qquad (6.2.6)$$

$$\ddot{\mathbf{p}} = \mathbf{J}_3 \ddot{\mathbf{q}}_3 + \left[-l_a \quad -l_b \quad 0\right]^T {}^{\bar{C}_3}\omega_{C_3z}^2 \qquad (6.2.7)$$

$$\ddot{\mathbf{p}} = \mathbf{J}_4 \ddot{\mathbf{q}}_4 + \left[l_a \quad -l_b \quad 0\right]^T {}^{\bar{C}_4}\omega_{C_4z}^2 \qquad (6.2.8)$$

The WMR body acceleration three-vector is $\ddot{\mathbf{p}} = \left[{}^{\bar{B}}a_{Bx} \quad {}^{\bar{B}}a_{By} \quad {}^{\bar{B}}\alpha_{Bz}\right]^T$.

6.2.5 Mobility Characteristics

Since the determinants of all of the four wheel Jacobians in (6.2.1)-(6.2.4) are nonzero, the soluble motion criterion in (A7.1) is satisfied, and the actuated inverse velocity solution in (2.4.8) is applicable. Furthermore, the three DOF criterion in (A7.2) is satisfied and the kinematic structure is capable of three DOF motion.

The adequate actuation criterion in (A7.8) yields: $\det[\mathbf{A}_a^T\mathbf{A}_a]=64(l_a+l_b)^2$. The actuators are thus able to provide motion in all three DOFs. We find that the robust actuation criterion in (A7.10) is not satisfied because the actuation structure is overdetermined. The actuation structure is thus not robust and actuator conflict may occur. Our 1984 recognition of the non-robust nature of Pluto's actuation structure [Muir84a] inaugurated our dissertation research. The sensed and actuated wheel variables are identical, thus the adequate sensing criterion in (A7.14) is identical to the adequate actuation criterion. Since the robot sensing equation in (A7.13) is also overdetermined, the sensing structure is robust which allows the detection of translational wheel slip by the method of Appendix 10. The sensed forward velocity solution in (2.4.14) is therefore applicable.

6.2.6 Kinematic Solutions

The complete inverse velocity solution is required for dynamic modeling. We compute the complete inverse velocity solution in (2.4.9) from (6.2.1)-(6.2.4):

$$\begin{pmatrix} {}^{A_1}\omega_{W_1x} \\ {}^{P_1}\omega_{R_1x} \\ {}^{\bar{C}_1}\omega_{C_1z} \end{pmatrix} = \begin{pmatrix} -1/R & 1/R & (l_a+l_b)/R \\ -\sqrt{2}/r & 0 & \sqrt{2}l_b/r \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} {}^{\bar{B}}v_{Bx} \\ {}^{\bar{B}}v_{By} \\ {}^{\bar{B}}\omega_{Bz} \end{pmatrix} = \mathbf{J}_1^{-1}\dot{\mathbf{p}} \qquad (6.2.9)$$

$$\begin{pmatrix} {}^{A_2}\omega_{W_2x} \\ {}^{P_2}\omega_{R_2x} \\ {}^{\bar{C}_2}\omega_{C_2z} \end{pmatrix} = \begin{pmatrix} 1/R & 1/R & -(l_a+l_b)/R \\ \sqrt{2}/r & 0 & -\sqrt{2}l_b/r \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} {}^{\bar{B}}v_{Bx} \\ {}^{\bar{B}}v_{By} \\ {}^{\bar{B}}\omega_{Bz} \end{pmatrix} = \mathbf{J}_2^{-1}\dot{\mathbf{p}} \qquad (6.2.10)$$

$$\begin{pmatrix} {}^{A_3}\omega_{W_{3x}} \\ {}^{P_3}\omega_{R_{3x}} \\ {}^{\bar{C}_3}\omega_{C_{3z}} \end{pmatrix} = \begin{pmatrix} -1/R & 1/R & -(l_a+l_b)/R \\ -\sqrt{2}/r & 0 & -\sqrt{2}l_b/r \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} {}^{\bar{B}}v_{Bx} \\ {}^{\bar{B}}v_{By} \\ {}^{\bar{B}}\omega_{Bz} \end{pmatrix} = \mathbf{J}_3^{-1}\dot{\mathbf{p}} \qquad (6.2.11)$$

$$\begin{pmatrix} {}^{A_4}\omega_{W_{4x}} \\ {}^{P_4}\omega_{R_{4x}} \\ {}^{\bar{C}_4}\omega_{C_{4z}} \end{pmatrix} = \begin{pmatrix} 1/R & 1/R & (l_a+l_b)/R \\ \sqrt{2}/r & 0 & \sqrt{2}l_b/r \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} {}^{\bar{B}}v_{Bx} \\ {}^{\bar{B}}v_{By} \\ {}^{\bar{B}}\omega_{Bz} \end{pmatrix} = \mathbf{J}_4^{-1}\dot{\mathbf{p}} \qquad (6.2.12)$$

The actuated inverse velocity solution is obtained from the complete inverse velocity solution in (6.2.9)-(6.2.12) by extracting the actuated wheel velocities. This approach requires approximately *one-tenth* of the computation of the direct application of the actuated inverse velocity solution in (2.4.8). The actuated inverse velocity solution, required for servo-controller design, is:

$$\begin{pmatrix} {}^{A_1}\omega_{W_{1x}} \\ {}^{A_2}\omega_{W_{2x}} \\ {}^{A_3}\omega_{W_{3x}} \\ {}^{A_4}\omega_{W_{4x}} \end{pmatrix} = \frac{1}{R}\begin{pmatrix} -1 & 1 & l_a+l_b \\ 1 & 1 & -l_a-l_b \\ -1 & 1 & -l_a-l_b \\ 1 & 1 & l_a+l_b \end{pmatrix} \begin{pmatrix} {}^{\bar{B}}v_{Bx} \\ {}^{\bar{B}}v_{By} \\ {}^{\bar{B}}\omega_{Bz} \end{pmatrix} \qquad (6.2.13)$$

The complete inverse acceleration solution is required for dynamic modeling. We compute the complete inverse acceleration solution by solving independently each of the wheel acceleration equations-of-motion in (6.2.5)-(6.2.8), applying the inverse wheel Jacobian matrices in (6.2.9)-(6.2.12):

$$\ddot{\mathbf{q}}_1 = \mathbf{J}_1^{-1}\left\{\ddot{\mathbf{p}} - [l_a \;\; l_b \;\; 0]^T\, {}^{\bar{C}_1}\omega_{C_{1z}}^2\right\} \qquad (6.2.14)$$

$$\ddot{\mathbf{q}}_2 = \mathbf{J}_2^{-1}\left\{\ddot{\mathbf{p}} - [-l_a \;\; l_b \;\; 0]^T\, {}^{\bar{C}_2}\omega_{C_{2z}}^2\right\} \qquad (6.2.15)$$

$$\ddot{\mathbf{q}}_3 = \mathbf{J}_3^{-1}\left\{\ddot{\mathbf{p}} - [-l_a \;\; -l_b \;\; 0]^T\, {}^{\bar{C}_3}\omega_{C_{3z}}^2\right\} \qquad (6.2.16)$$

$$\ddot{\mathbf{q}}_4 = \mathbf{J}_4^{-1}\left\{\ddot{\mathbf{p}} - [l_a \;\; -l_b \;\; 0]^T\, {}^{\bar{C}_4}\omega_{C_{4z}}^2\right\} \qquad (6.2.17)$$

The wheel acceleration three-vector is $\ddot{\mathbf{q}}_i = \left[{}^{A_i}\alpha_{W_{ix}} \quad {}^{P_i}\alpha_{R_{ix}} \quad {}^{\bar{C}_i}\omega_{C_{iz}}\right]^T$ for i=1,...,4.

The sensed forward velocity solution is required for the dead reckoning computations within the feedback path of model-based WMR servo-controllers. We apply the least-squares sensed forward velocity solution in (2.4.14) to obtain:

$$\begin{pmatrix} {}^{\bar{B}}v_{Bx} \\ {}^{\bar{B}}v_{By} \\ {}^{\bar{B}}\omega_{Bz} \end{pmatrix} = \frac{R}{4(l_a+l_b)} \begin{pmatrix} -(l_a+l_b) & (l_a+l_b) & -(l_a+l_b) & (l_a+l_b) \\ (l_a+l_b) & (l_a+l_b) & (l_a+l_b) & (l_a+l_b) \\ 1 & -1 & -1 & 1 \end{pmatrix} \begin{pmatrix} {}^{A_1}\omega_{W_1x} \\ {}^{A_2}\omega_{W_2x} \\ {}^{A_3}\omega_{W_3x} \\ {}^{A_4}\omega_{W_4x} \end{pmatrix} \quad (6.2.18)$$

6.2.7 Translational Wheel Slip Detection

Since the sensing structure of Uranus is overdetermined, the kinematic equations-of-motion are inconsistent in the presence of translational wheel slip. The error in the least-squares forward velocity solution is then non-zero. We detect translational wheel slip by the method of Appendix 10. The first step is to substitute the wheel Jacobian matrices into the robot sensing equation in (A7.13). We then form the least-squares error in (A10.1) from the robot sensing equation. The least-squares error is applied to the wheel slip detection algorithm in (A10.2) to obtain the translational wheel slip detection algorithm for Uranus:

$$\text{If } \frac{R}{2}\left|{}^{A_1}\omega_{W_1x} + {}^{A_2}\omega_{W_2x} - {}^{A_3}\omega_{W_3x} - {}^{A_4}\omega_{W_4x}\right| > e_t\,, \quad (6.2.19)$$

translational wheel slip has occurred. The scalar error threshold e_t is experimentally determined from the worst case sensor noise error.

6.2.8 Dead Reckoning

We apply the dead reckoning update formula in (A9.4) to Uranus. Upon substituting the sensed forward velocity solution in (6.2.18) into (A9.4), expanding, and simplifying, we obtain algorithms for the direct computation of the current orientation ${}^{F}\theta_B(nT)$, and the recursive computation of the current translational positions ${}^{F}d_{Bx}(nT)$ and ${}^{F}d_{By}(nT)$ of Uranus:

$$ {}^{F}\theta_{B}(nT) = \frac{R}{4(l_a+l_b)}\theta(nT) + {}^{F}\theta_{B}(0) - \frac{R}{4(l_a+l_b)}\theta(0) \qquad (6.2.20) $$

and

$$ \begin{pmatrix} {}^{F}d_{Bx}(nT) \\ {}^{F}d_{By}(nT) \end{pmatrix} = \begin{pmatrix} {}^{F}d_{Bx}[(n-1)T] \\ {}^{F}d_{By}[(n-1)T] \end{pmatrix} + \frac{TR}{8}\begin{pmatrix} -p & m & -p & m \\ m & p & m & p \end{pmatrix}\left\{\dot{q}_s[(n-1)T]+\dot{q}_s(nT)\right\} \qquad (6.2.21) $$

where $\theta(\bullet) = \left\{{}^{A_1}\theta_{W_1x}(\bullet) - {}^{A_2}\theta_{W_2x}(\bullet) - {}^{A_3}\theta_{W_3x}(\bullet) + {}^{A_4}\theta_{W_4x}(\bullet)\right\}$,

$p = \left\{\cos{}^{F}\theta_{B}(nT) + \sin{}^{F}\theta_{B}(nT)\right\}$ and $m = \left\{\cos{}^{F}\theta_{B}(nT) - \sin{}^{F}\theta_{B}(nT)\right\}$

The direct computation of the WMR orientation indicates that (6.2.20) is a holonomic constraint [Landau76]. Since the WMR translations cannot be computed directly, (6.2.21) are non-holonomic constraints. Errors in the recursively computed WMR translations due to finite precision and sensor noise will accumulate; whereas, the direct computation of the holonomic orientation constraint in (6.2.20) will not accumulate these errors.

6.2.9 Concluding Remarks

In this section, we have shown that the four omnidirectional wheels on Uranus each possesses three DOFs thereby providing Uranus with three DOFs (x, y and θ) in the plane. The actuation structure is adequate but actuator conflict may occur. The positioning of the omnidirectional wheels on the right and left hand sides of the WMR is important. We demonstrate in Appendix A11.4 that Uranus would have inadequate actuation if the right and left hand wheels were interchanged. Also, in Appendix A11.4, we illustrate how Uranus may be converted into a robust actuation structure so that only three motors are required and no actuator conflict may occur. The sensing structure of Uranus is robust, thus allowing the detection of translational wheel slip by the algorithm in (6.2.19). The kinematic solutions in Section 6.2.6, the translational wheel slip detection algorithm in (6.2.19), and the dead reckoning algorithm in (6.2.20)-(6.2.21) have low computational requirements which allow real-time implementation.

We apply the Uranus kinematic solutions in (6.2.9)-(6.2.-18) to dynamic modeling in Section 6.3, servo-controller design in Section 6.5, and the simulation experiments in Section 6.6. The coordinate systems assigned in Section 6.2.2 are a subset of the coordinate systems assigned in Section 6.3 for dynamic modeling. We apply the complete inverse velocity solution in Section 6.2.6 to compute the frictional coefficients for dynamic simulation. We incorporate the actuated inverse velocity solution in the feedforward path, and the sensed forward velocity solution and dead reckoning algorithm in the feedback path of the resolved motion rate servo-controller in Section 6.5. For application to resolved acceleration servo-control and simulation, we include our complete inverse acceleration solution in Section 6.2.6 to eliminate the wheel joint accelerations form the Uranus dynamic model in Section 6.3.

6.3 Dynamic Modeling

6.3.1 Overview

We model the dynamics of Uranus to illustrate the application of our dynamic modeling procedure in Chapter 3. We assume: negligible roller masses and inertias; no wheel translational slip; and negligible viscous friction on the load, wheels and rollers. We make correspondences between the components of Uranus pictured in Figure A2.10 and the simple closed-chain in Figure 1.1. We identify the rectangular WMR body as the main link, the floor as the environmental body, the hub of each omnidirectional wheel as the proximal link of each of the four open-chains, the roller which contacts the floor as the distal link of each open-chain, and the load as an additional single link open-chain which does not contact the floor. We sketch these components in the plane in Figures 6.1 and 6.2 and in three-dimensions in Figure 6.3.

We begin in Section 6.3.2 by assigning coordinate systems (in addition to the coordinate systems assigned for kinematic modeling in Section 6.2) for Uranus dynamic modeling. In Section 6.3.3, we formulate the components required for dynamic modeling: position six-vectors, force/torque source six-vectors, companion (6×6) kinematic matrices, (6×6) joint coupling matrices, dry frictional coefficients, and the (6×6) propagation matrices. We apply these components in Section 6.3.4 to formulate the primary, internal and secondary

force/torque equations-of-motion. In Section 6.3.5, we compute the six-vector velocity and acceleration transformations which we then combine in Section 6.3.6 with the force/torque equations-of-motion to compute the Uranus dynamic model. In Section 6.3.7, we solve the dynamic model to compute the actuated inverse and forward dynamic solutions. Finally, in Section 6.3.8, we advance our concluding remarks.

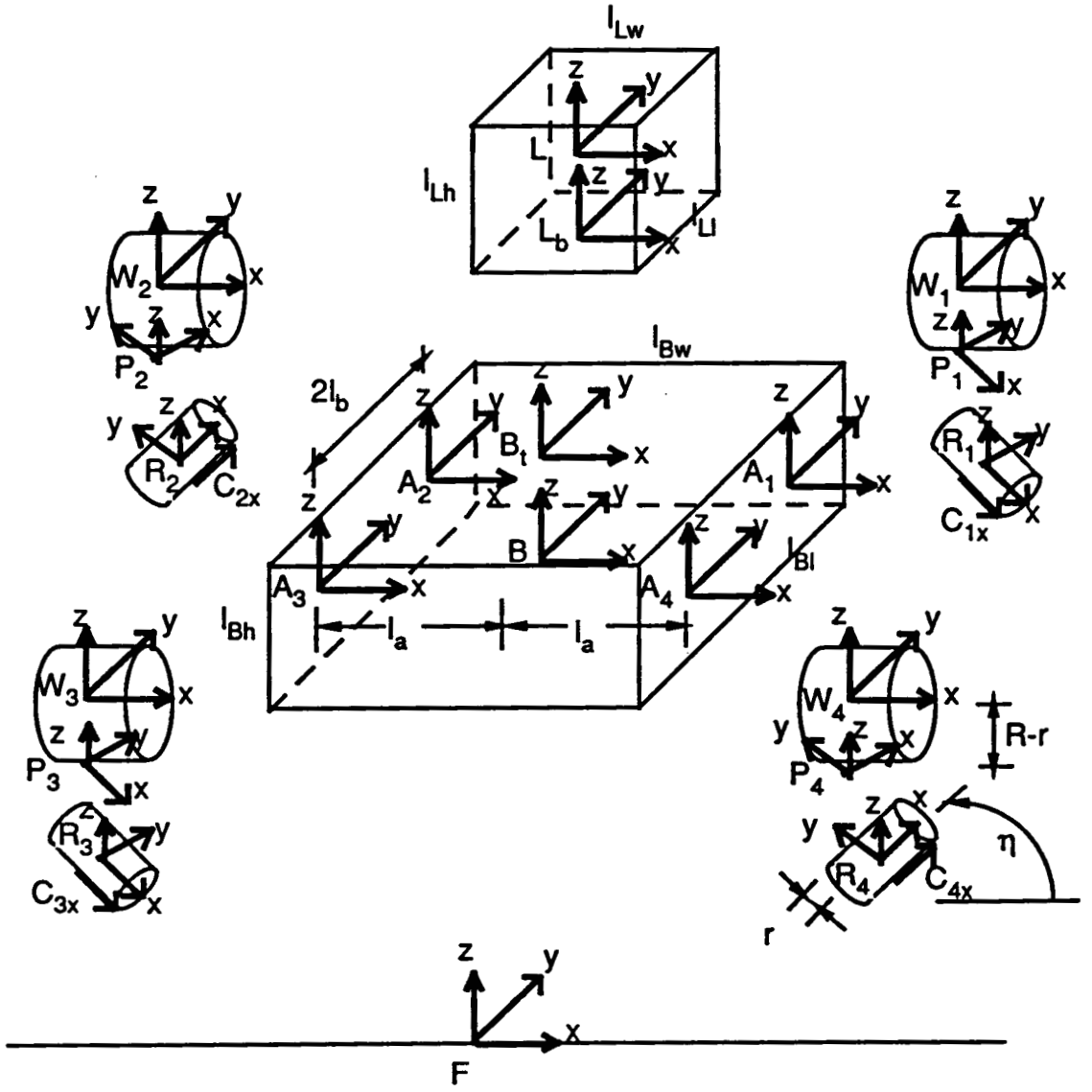


Figure 6.3: Uranus with all Coordinate Systems Assigned

6.3.2 Coordinate System Assignments

We assign and draw the body B, axle A_i, wheel W_i, pivot P_i, roller R_i, contact-point C_i (for i=1,...,4), and floor F coordinate systems in Figures 6.1-6.3. The *ten* gravitational coordinate systems, G(*B*) and G(*L*), and G(W_i) and G(R_i) for i=1,...,4, which are located at the origins of the B, L, W_i, and R_i coordinate systems and aligned with the gravitational field are not be explicitly assigned.

6.3.3 Dynamic Model Components

The *thirty-two* position six-vectors of Uranus, formulated by inspection of Figure 6.3, are:

$$^{F}\mathbf{p}_{B} = \left({}^{F}d_{Bx} \;\; {}^{F}d_{By} \;\; l_c \;\; 0 \;\; 0 \;\; {}^{F}\theta_{Bz} \right)^T \qquad {}^{B}\mathbf{p}_{B_t} = \left(0 \;\; 0 \;\; l_{Bh}/2 \;\; 0 \;\; 0 \;\; 0 \right)^T$$

$$^{B_1}\mathbf{p}_{L_b} = \left(0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \right)^T \qquad {}^{L_b}\mathbf{p}_{L} = \left(0 \;\; 0 \;\; l_{Lh}/2 \;\; 0 \;\; 0 \;\; 0 \right)^T$$

$$^{B}\mathbf{p}_{A_1} = \left(l_a \;\; l_b \;\; (R-l_c) \;\; 0 \;\; 0 \;\; 0 \right)^T \qquad {}^{B}\mathbf{p}_{A_2} = \left(-l_a \;\; l_b \;\; (R-l_c) \;\; 0 \;\; 0 \;\; 0 \right)^T$$

$$^{B}\mathbf{p}_{A_3} = \left(-l_a \;\; -l_b \;\; (R-l_c) \;\; 0 \;\; 0 \;\; 0 \right)^T \qquad {}^{B}\mathbf{p}_{A_4} = \left(l_a \;\; -l_b \;\; (R-l_c) \;\; 0 \;\; 0 \;\; 0 \right)^T$$

$$^{A_i}\mathbf{p}_{W_i} = \left(0 \;\; 0 \;\; 0 \;\; {}^{A_i}\theta_{W_ix} \;\; 0 \;\; 0 \right)^T \qquad {}^{W_i}\mathbf{p}_{Q_i} = \left(0 \;\; 0 \;\; 0 \;\; -{}^{A_i}\theta_{W_ix} \;\; 0 \;\; 0 \right)^T$$

$$^{Q_i}\mathbf{p}_{P_i} = \left(0 \;\; 0 \;\; (r-R) \;\; 0 \;\; 0 \;\; h_i \right)^T \qquad {}^{P_i}\mathbf{p}_{R_i} = \left(0 \;\; 0 \;\; 0 \;\; {}^{P_i}\theta_{R_ix} \;\; 0 \;\; 0 \right)^T$$

$$^{R_i}\mathbf{p}_{U_i} = \left(0 \;\; 0 \;\; 0 \;\; -{}^{P_i}\theta_{R_ix} \;\; 0 \;\; 0 \right)^T \quad \text{and} \quad {}^{U_i}\mathbf{p}_{C_i} = \left(0 \;\; 0 \;\; -r \;\; 0 \;\; 0 \;\; 0 \right)^T$$

for i=1,...,4. The *intermediate* coordinate systems Q_i and U_i are introduced to ease the determination of the position vectors (with roll-pitch-yaw rotations) by inspection of Figures 6.1-6.3. Coordinate system Q_i is coincident with the A_i coordinate system and fixed with wheel W_i, and coordinate system U_i is coincident with the P_i coordinate system and fixed with the roller R_i. The position and orientation of the WMR body relative to the reference floor coordinate system are ${}^{F}d_{Bx}$, ${}^{F}d_{By}$, and ${}^{F}\theta_{Bz}$, the wheel angles are ${}^{A_i}\theta_{W_ix}$ and the roller angles are ${}^{P_i}\theta_{R_ix}$ for i=1,...,4.

The *twenty-five* force/torque six-vectors for Uranus, formulated according to Sections 3.3.2.2 to 3.3.2.6, are:

$$ {}^{B}\mathbf{f}_{iB} = \begin{pmatrix} -m_B {}^{\bar{B}}a_{Bx} \\ -m_B {}^{\bar{B}}a_{By} \\ 0 \\ 0 \\ 0 \\ -I_{Bz} {}^{\bar{B}}\alpha_{Bz} \end{pmatrix} \qquad {}^{L}\mathbf{f}_{iL} = \begin{pmatrix} -m_L {}^{\bar{L}}a_{Lx} \\ -m_L {}^{\bar{L}}a_{Ly} \\ 0 \\ 0 \\ 0 \\ -I_{Lz} {}^{\bar{L}}\alpha_{Lz} \end{pmatrix} \qquad {}^{B}\mathbf{f}_{vB} = \begin{pmatrix} -\mu_{vBx} {}^{\bar{B}}v_{Bx} \\ -\mu_{vBy} {}^{\bar{B}}v_{By} \\ 0 \\ 0 \\ 0 \\ -\varepsilon_{vBz} {}^{\bar{B}}\omega_{Bz} \end{pmatrix} $$

$$ {}^{W_i}\mathbf{f}_{iW_i} = \begin{pmatrix} -m_W {}^{\bar{W}_i}a_{W_ix} \\ -m_W {}^{\bar{W}_i}a_{W_iy} \\ -m_W {}^{\bar{W}_i}a_{W_iz} \\ -I_{W_ix} {}^{\bar{W}_i}\alpha_{W_ix} \\ -I_{W_iy} {}^{\bar{W}_i}\alpha_{W_iy} - (I_{W_ix} - I_{W_iy}) {}^{\bar{W}_i}\omega_{W_ix} {}^{\bar{W}_i}\omega_{W_iz} \\ -I_{W_iy} {}^{\bar{W}_i}\alpha_{W_iz} - (I_{W_iy} - I_{W_ix}) {}^{\bar{W}_i}\omega_{W_ix} {}^{\bar{W}_i}\omega_{W_iy} \end{pmatrix} \qquad {}^{L}\mathbf{f}_{gL} = \begin{pmatrix} 0 \\ 0 \\ -m_L g \\ 0 \\ 0 \\ 0 \end{pmatrix} $$

$$ {}^{B}\mathbf{f}_{gB} = \begin{pmatrix} 0 & 0 & -m_B g & 0 & 0 & 0 \end{pmatrix}^T \qquad {}^{G(W_i)}\mathbf{f}_{gW_i} = \begin{pmatrix} 0 & 0 & -m_W g & 0 & 0 & 0 \end{pmatrix}^T $$

$$ {}^{C_i}\mathbf{f}_{eR_i} = \begin{pmatrix} f_{xi} & f_{yi} & f_{zi} & \tau_{xi} & \tau_{yi} & \tau_{zi} \end{pmatrix}^T \qquad \text{and} \qquad {}^{W_i}\mathbf{f}_{aW_i} = -{}^{A_i}\mathbf{f}_{aB} = \begin{pmatrix} 0 & 0 & 0 & \beta\tau_i & 0 & 0 \end{pmatrix}^T $$

for i=1,...,4. The inertial forces/torques are formulated according to (3.3.3), the gravitational forces/torques according to (3.3.4), the actuation forces/torques according to (3.3.5), the viscous friction forces/torques according to (3.3.6), and the environmental forces/torques according to (3.3.8). Since each wheel is symmetric about its axle, $I_{W_iz} = I_{W_iy}$ in the inertial force/torque six-vector ${}^{W_i}\mathbf{f}_{iW_i}$.

The companion kinematic matrices are required for the force/torque equations-of-motion in Section 6.3.4 and the velocity and acceleration transformations in Section 6.3.5. Each (6×6) kinematic matrix ${}^{A}\mathbf{X}_{B}$, for

X=L,J,R,S, and **T**, is formulated by substituting the elements of the position six-vector ${}^{B}\mathbf{p}_{A}$ into the corresponding kinematic matrix template in Appendix 12. The *fifty-eight* Uranus companion kinematic matrices are:

$$ {}^{B_t}\mathbf{L}_{B} = \begin{pmatrix} 1 & 0 & 0 & 0 & l_{Bh}/2 & 0 \\ 0 & 1 & 0 & -l_{Bh}/2 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \qquad {}^{L}\mathbf{L}_{L_b} = \begin{pmatrix} 1 & 0 & 0 & 0 & l_{Lh}/2 & 0 \\ 0 & 1 & 0 & -l_{Lh}/2 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} $$

$$ {}^{L_b}\mathbf{L}_{B_t} = {}^{L_b}\mathbf{J}_{B_t} = \mathbf{I} $$

$$ {}^{B}\mathbf{L}_{F} = \begin{pmatrix} \cos{}^{F}\theta_{Bz} & \sin{}^{F}\theta_{Bz} & 0 & -l_c\sin{}^{F}\theta_{Bz} & l_c\cos{}^{F}\theta_{Bz} & {}^{F}d_{Bx}\sin{}^{F}\theta_{Bz} - {}^{F}d_{By}\cos{}^{F}\theta_{Bz} \\ -\sin{}^{F}\theta_{Bz} & \cos{}^{F}\theta_{Bz} & 0 & -l_c\cos{}^{F}\theta_{Bz} & -l_c\sin{}^{F}\theta_{Bz} & {}^{F}d_{By}\sin{}^{F}\theta_{Bz} + {}^{F}d_{Bx}\cos{}^{F}\theta_{Bz} \\ 0 & 0 & 1 & {}^{F}d_{By} & -{}^{F}d_{Bx} & 0 \\ 0 & 0 & 0 & \cos{}^{F}\theta_{Bz} & \sin{}^{F}\theta_{Bz} & 0 \\ 0 & 0 & 0 & -\sin{}^{F}\theta_{Bz} & \cos{}^{F}\theta_{Bz} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} $$

$$ {}^{A_1}\mathbf{L}_{B} = \begin{pmatrix} 1 & 0 & 0 & 0 & R-l_c & -l_b \\ 0 & 1 & 0 & l_c-R & 0 & l_a \\ 0 & 0 & 1 & l_b & -l_a & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \qquad {}^{A_2}\mathbf{L}_{B} = \begin{pmatrix} 1 & 0 & 0 & 0 & R-l_c & -l_b \\ 0 & 1 & 0 & l_c-R & 0 & -l_a \\ 0 & 0 & 1 & l_b & l_a & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} $$

$$ {}^{A_3}\mathbf{L}_{B} = \begin{pmatrix} 1 & 0 & 0 & 0 & R-l_c & l_b \\ 0 & 1 & 0 & l_c-R & 0 & -l_a \\ 0 & 0 & 1 & -l_b & l_a & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \qquad {}^{A_4}\mathbf{L}_{B} = \begin{pmatrix} 1 & 0 & 0 & 0 & R-l_c & l_b \\ 0 & 1 & 0 & l_c-R & 0 & l_a \\ 0 & 0 & 1 & -l_b & -l_a & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} $$

$$
{}^{W_i}L_{A_i} = {}^{W_i}J_{A_i} = \begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 \\
0 & \cos^{A_i}\theta_{W_ix} & \sin^{A_i}\theta_{W_ix} & 0 & 0 & 0 \\
0 & -\sin^{A_i}\theta_{W_ix} & \cos^{A_i}\theta_{W_ix} & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & \cos^{A_i}\theta_{W_ix} & \sin^{A_i}\theta_{W_ix} \\
0 & 0 & 0 & 0 & -\sin^{A_i}\theta_{W_ix} & \cos^{A_i}\theta_{W_ix}
\end{pmatrix}
$$

$$
{}^{Q_i}L_{W_i} = \begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 \\
0 & \cos^{A_i}\theta_{W_ix} & -\sin^{A_i}\theta_{W_ix} & 0 & 0 & 0 \\
0 & \sin^{A_i}\theta_{W_ix} & \cos^{A_i}\theta_{W_ix} & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & \cos^{A_i}\theta_{W_ix} & -\sin^{A_i}\theta_{W_ix} \\
0 & 0 & 0 & 0 & \sin^{A_i}\theta_{W_ix} & \cos^{A_i}\theta_{W_ix}
\end{pmatrix}
$$

$$
{}^{P_i}L_{Q_i} = \begin{pmatrix}
\sqrt{2}/2 & -\sqrt{2}/2 & 0 & \sqrt{2}(r-R)/2 & \sqrt{2}(r-R)/2 & 0 \\
\sqrt{2}/2 & \sqrt{2}/2 & 0 & -\sqrt{2}(r-R)/2 & \sqrt{2}(r-R)/2 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & \sqrt{2}/2 & -\sqrt{2}/2 & 0 \\
0 & 0 & 0 & \sqrt{2}/2 & \sqrt{2}/2 & 0 \\
0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix}
$$

$$
{}^{R_i}L_{P_i} = \begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 \\
0 & \cos^{P_i}\theta_{R_ix} & \sin^{P_i}\theta_{R_ix} & 0 & 0 & 0 \\
0 & -\sin^{P_i}\theta_{R_ix} & \cos^{P_i}\theta_{R_ix} & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & \cos^{P_i}\theta_{R_ix} & \sin^{P_i}\theta_{R_ix} \\
0 & 0 & 0 & 0 & -\sin^{P_i}\theta_{R_ix} & \cos^{P_i}\theta_{R_ix}
\end{pmatrix}
$$

$$
{}^{U_i}\mathbf{L}_{R_i} = \begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 \\
0 & \cos^{P_i}\theta_{R_ix} & -\sin^{P_i}\theta_{R_ix} & 0 & 0 & 0 \\
0 & \sin^{P_i}\theta_{R_ix} & \cos^{P_i}\theta_{R_ix} & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & \cos^{P_i}\theta_{R_ix} & -\sin^{P_i}\theta_{R_ix} \\
0 & 0 & 0 & 0 & \sin^{P_i}\theta_{R_ix} & \cos^{P_i}\theta_{R_ix}
\end{pmatrix}
$$

$$
{}^{C_i}\mathbf{L}_{U_i} = \begin{pmatrix}
1 & 0 & 0 & 0 & -r & 0 \\
0 & 1 & 0 & r & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix}
\qquad
{}^{L_b}\mathbf{R}_{B_t} = \begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & 1 \\
0 & 1 & 0 & 0 & -1 & 0 \\
-1 & 0 & 1 & 0 & 0 & 0
\end{pmatrix}
$$

$$
{}^{W_i}\mathbf{R}_{A_i} = \begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & 1 \\
-\sin^{A_i}\theta_{W_ix} & \cos^{A_i}\theta_{W_ix} & \sin^{A_i}\theta_{W_ix} & 0 & -\cos^{A_i}\theta_{W_ix} & 0 \\
-\cos^{A_i}\theta_{W_ix} & -\sin^{A_i}\theta_{W_ix} & \cos^{A_i}\theta_{W_ix} & 0 & \sin^{A_i}\theta_{W_ix} & 0
\end{pmatrix}
$$

$$
{}^{B_t}\mathbf{S}_{B} = \begin{pmatrix}
0 & 0 & 0 & 0 & l_{Bh}/2 & 0 \\
0 & 0 & 0 & 0 & 0 & l_{Bh}/2 \\
-l_{Bh}/2 & -l_{Bh}/2 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}
\qquad
{}^{L}\mathbf{S}_{L_b} = \begin{pmatrix}
0 & 0 & 0 & 0 & l_{Lh}/2 & 0 \\
0 & 0 & 0 & 0 & 0 & l_{Lh}/2 \\
-l_{Lh}/2 & -l_{Lh}/2 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}
$$

$$
{}^{A_1}S_B=\begin{pmatrix} 0 & -l_a & -l_a & -l_b & R-l_c & 0 \\ -l_b & 0 & -l_b & l_a & 0 & R-l_c \\ l_c-R & l_c-R & 0 & 0 & l_a & l_b \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}
\qquad
{}^{A_2}S_B=\begin{pmatrix} 0 & l_a & l_a & -l_b & R-l_c & 0 \\ -l_b & 0 & -l_b & -l_a & 0 & R-l_c \\ l_c-R & l_c-R & 0 & 0 & -l_a & l_b \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}
$$

$$
{}^{A_3}S_B=\begin{pmatrix} 0 & l_a & l_a & l_b & R-l_c & 0 \\ l_b & 0 & l_b & -l_a & 0 & R-l_c \\ l_c-R & l_c-R & 0 & 0 & -l_a & -l_b \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}
\qquad
{}^{A_4}S_B=\begin{pmatrix} 0 & -l_a & -l_a & l_b & R-l_c & 0 \\ l_b & 0 & l_b & l_a & 0 & R-l_c \\ l_c-R & l_c-R & 0 & 0 & l_a & -l_b \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}
$$

$$
{}^{L_b}T_{B_t}=\begin{pmatrix} 0 & 0 & 0 & 2 & 0 & -2 \\ 0 & -2 & 0 & 0 & 2 & 0 \\ 2 & 0 & -2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}
\qquad
{}^{L_b}S_{B_t}={}^{W_i}S_{A_i}=\mathbf{0}
$$

$$
{}^{W_i}T_{A_i}=\begin{pmatrix} 0 & 0 & 0 & 2 & 0 & -2 \\ 2\sin^{A_i}\theta_{W_ix} & -2\cos^{A_i}\theta_{W_ix} & -2\sin^{A_i}\theta_{W_ix} & 0 & 2\cos^{A_i}\theta_{W_ix} & 0 \\ 2\cos^{A_i}\theta_{W_ix} & 2\sin^{A_i}\theta_{W_ix} & -2\cos^{A_i}\theta_{W_ix} & 0 & -2\sin^{A_i}\theta_{W_ix} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}
$$

for i=1,...,4.

In the absence of translational wheel slippage, the *thirteen* coupling matrices are:

$$ {}^{C_i}\mathbf{C}_{R_iF}=\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ \zeta_{cai} & 0 & \zeta_{ri} & \zeta_{sai} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \zeta_{cci} & 0 & 0 & \zeta_{sci} \end{pmatrix} \qquad {}^{P_i}\mathbf{C}_{W_iR_i}=\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & \zeta_{cri} & \zeta_{sri} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} $$

$$ {}^{A_i}\mathbf{C}_{BW_i}=\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & \zeta_{cwi} & \zeta_{swi} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \qquad \text{and} \qquad {}^{B_t}\mathbf{C}_{BL}=\mathbf{I} \quad \text{for } i=1,\ldots,4. $$

The coupling matrix between the WMR body and the load ${}^{B_t}\mathbf{C}_{BL}$ is the identity matrix because we assume that the load cannot slide.

The *thirty-six* frictional coefficients required for the coupling matrices are formulated as:

$$ \zeta_{ri}=\begin{cases} -\mu_r \operatorname{sgn}({}^{P_i}\omega_{R_ix}) & \text{if } {}^{P_i}\omega_{R_ix}\neq 0 \\ 0 & \text{otherwise} \end{cases} $$

$$ \zeta_{sci}=\begin{cases} 1 & \text{if } {}^{C_i}\omega_{C_iz}=0; \\ & \text{and } |{}^{C_i}\tau_{Fx}|\leq\mu_{sc}|{}^{C_i}f_{Fz}| \\ 0 & \text{otherwise} \end{cases} \qquad \zeta_{cci}=\begin{cases} -\mu_{cc}\operatorname{sgn}({}^{\bar{B}_i}\omega_{C_iz}\,{}^{C_i}f_{Fz}) & \text{if } {}^{\bar{B}_i}\omega_{C_iz}\neq 0 \\ 0 & \text{otherwise} \end{cases} $$

$$ \zeta_{sai}=\begin{cases} 1 & \text{if } {}^{P_i}\omega_{R_ix}=0; \\ & \text{and } |{}^{P_i}\tau_{R_ix}|\leq\mu_{sa}|{}^{P_i}f_{R_ix}| \\ 0 & \text{otherwise} \end{cases} \qquad \zeta_{cai}=\begin{cases} -\mu_{ca}\operatorname{sgn}({}^{P_i}\omega_{R_ix}\,{}^{P_i}f_{R_ix}) & \text{if } {}^{P_i}\omega_{R_ix}\neq 0 \\ 0 & \text{otherwise} \end{cases} $$

$$ \zeta_{sri}=\begin{cases} 1 & \text{if } {}^{P_i}\omega_{R_ix}=0; \\ & \text{and } |{}^{P_i}\tau_{R_ix}|\leq\mu_{sr}|{}^{P_i}f_{R_iz}| \\ 0 & \text{otherwise} \end{cases} \qquad \zeta_{cri}=\begin{cases} -\mu_{cr}\operatorname{sgn}({}^{P_i}\omega_{R_ix}\,{}^{P_i}f_{R_iz}) & \text{if } {}^{P_i}\omega_{R_ix}\neq 0 \\ 0 & \text{otherwise} \end{cases} $$

$$ \zeta_{swi}=\begin{cases} 1 & \text{if } {}^{A_i}\omega_{W_ix}=0; \text{ and} \\ & |{}^{A_i}\tau_{W_ix}|\leq\mu_{sw}|{}^{A_i}f_{W_iz}| \\ 0 & \text{otherwise} \end{cases} \qquad \zeta_{cwi}=\begin{cases} -\mu_{cw}\operatorname{sgn}({}^{A_i}\omega_{W_ix}\,{}^{A_i}f_{W_iz}) & \text{if } {}^{A_i}\omega_{W_ix}\neq 0 \\ 0 & \text{otherwise} \end{cases} $$

for i=1,...,4. The stictional coefficients are formulated according to (3.3.19), the Coulomb frictional coefficients according to (3.3.20) and the rolling frictional

coefficients according to (3.3.21). The rolling (r) friction constant is μ_r, and the stiction (s) and Coulomb (c) friction constants are μ_{sc} and μ_{cc} for the wheel contact (c) friction, μ_{sa} and μ_{ca} for the axial (a) roller friction, μ_{sr} and μ_{cr} for the roller (r) bearing friction, and μ_{sw} and μ_{cw} for the wheel (w) bearing friction. We assume that the axial frictions and all stictional constants are zero (i.e., $\mu_{ca}=\mu_{sc}=\mu_{sa}=\mu_{sr}=\mu_{sw}=\mu_s=0$) to reduce the computational complexity of solving the dynamic model in Section 6.3.7. For simplicity, we also assume that all Coulomb frictional constants are equal (i.e., $\mu_{cc}=\mu_{cr}=\mu_{cw}=\mu_c$).

We apply the link Jacobian cascade property in (3.3.15) to eliminate the intermediate coordinate systems U_i and Q_i for i=1,...,4 from the computation of the propagation matrices:

$$ {}^{C_i}\mathbf{L}_{R_i} = {}^{C_i}\mathbf{L}_{U_i}\,{}^{U_i}\mathbf{L}_{R_i} \tag{6.3.1} $$

$$ {}^{P_i}\mathbf{L}_{W_i} = {}^{P_i}\mathbf{L}_{Q_i}\,{}^{Q_i}\mathbf{L}_{W_i} \tag{6.3.2} $$

The four wheel gravitational propagation matrices are formulated according to (3.3.27) for i=1,...,4 as:

$$ {}^{W_i}\mathbf{P}_{G(W_i)} = \left({}^{B}\mathbf{L}_{F}^{T}\,{}^{A_i}\mathbf{L}_{B}^{T}\,{}^{W_i}\mathbf{L}_{A_i}^{T}\right)^{T} * \mathbf{P}_{mask}\,. \tag{6.3.3} $$

The gravitational propagation mask $\mathbf{P}_{mask}$ is defined in (3.3.28). The remaining *twenty-five* propagation matrices required for Uranus dynamic modeling are formulated by applying the propagation cascade in (3.3.26) for i=1,...,4 to obtain:

$$ {}^{P_i}\mathbf{P}_{C_i} = {}^{R_i}\mathbf{L}_{P_i}^{T}\,{}^{C_i}\mathbf{L}_{R_i}^{T}\,{}^{C_i}\mathbf{C}_{R_iF} \tag{6.3.4} $$

$$ {}^{A_i}\mathbf{P}_{P_i} = {}^{W_i}\mathbf{L}_{A_i}^{T}\,{}^{P_i}\mathbf{L}_{W_i}^{T}\,{}^{P_i}\mathbf{C}_{W_iR_i} \tag{6.3.5} $$

$$ {}^{A_i}\mathbf{P}_{W_i} = {}^{W_i}\mathbf{L}_{A_i}^{T} \tag{6.3.6} $$

$$ {}^{A_i}\mathbf{P}_{G(W_i)} = {}^{W_i}\mathbf{L}_{A_i}^{T}\,{}^{W_i}\mathbf{P}_{G(W_i)} \tag{6.3.7} $$

$$ {}^{B}P_{L} = {}^{B_t}L_{B}^{T}\,{}^{B_t}C_{BL}\,{}^{L_b}L_{B_t}^{T}\,{}^{L}L_{L_b}^{T} \tag{6.3.8} $$

$$ {}^{B}P_{W_i} = {}^{A_i}L_{B}^{T}\,{}^{A_i}C_{BW_i}\,{}^{W_i}L_{A_i}^{T} \tag{6.3.9} $$

$$ {}^{B}P_{A_i} = {}^{A_i}L_{B}^{T} \tag{6.3.10} $$

$$ {}^{B}P_{C_i} = {}^{B}P_{W_i}\,{}^{P_i}L_{W_i}^{T}\,{}^{P_i}C_{W_iR_i}\,{}^{R_i}L_{P_i}^{T}\,{}^{R_i}P_{C_i} \tag{6.3.11} $$

$$ {}^{B}P_{G(W_i)} = {}^{B}P_{W_i}\,{}^{W_i}P_{G(W_i)} \tag{6.3.12} $$

6.3.4 Force/Torque Equations-of-Motion

We formulate the primary force/torque equations-of-motion by propagating all of the force/torque sources in Section 6.3.3 with the propagation matrices in Section 6.3.7 to the body coordinate system and summing the resulting matrices to zero according to (3.3.28):

$$
\begin{aligned}
&{}^{B}P_{W_1}\,{}^{W_1}f_{iW_1} + {}^{B}P_{G(W_1)}\,{}^{G(W_1)}f_{gW_1} + {}^{B}P_{C_1}\,{}^{C_1}f_{eR_1} + {}^{B}P_{A_1}\,{}^{A_1}f_{aB} + {}^{B}P_{W_1}\,{}^{W_1}f_{aW_1} \\
&\quad + {}^{B}P_{W_2}\,{}^{W_2}f_{iW_2} + {}^{B}P_{G(W_2)}\,{}^{G(W_2)}f_{gW_2} + {}^{B}P_{C_2}\,{}^{C_2}f_{eR_2} + {}^{B}P_{A_2}\,{}^{A_2}f_{aB} + {}^{B}P_{W_2}\,{}^{W_2}f_{aW_2} \\
&\quad + {}^{B}P_{W_3}\,{}^{W_3}f_{iW_3} + {}^{B}P_{G(W_3)}\,{}^{G(W_3)}f_{gW_3} + {}^{B}P_{C_3}\,{}^{C_3}f_{eR_3} + {}^{B}P_{A_3}\,{}^{A_3}f_{aB} + {}^{B}P_{W_3}\,{}^{W_3}f_{aW_3} \\
&\quad + {}^{B}P_{W_4}\,{}^{W_4}f_{iW_4} + {}^{B}P_{G(W_4)}\,{}^{G(W_4)}f_{gW_4} + {}^{B}P_{C_4}\,{}^{C_4}f_{eR_4} + {}^{B}P_{A_4}\,{}^{A_4}f_{aB} + {}^{B}P_{W_4}\,{}^{W_4}f_{aW_4} \\
&\quad + {}^{B}P_{L}\,{}^{L}f_{iL} + {}^{B}P_{L}\,{}^{L}f_{gL} + {}^{B}f_{iB} + {}^{B}f_{gB} + {}^{B}f_{vB} = 0
\end{aligned}
\tag{6.3.13}
$$

We apply MACSYMA to compute symbolically this set of *six* primary force/torque equations-of-motion.

The internal forces/torques at each of the joints are required to compute the secondary force/torque equations-of-motion and the frictional coefficients. We compute recursively the internal forces/torques according to (3.3.31) to obtain:

$$ {}^{C_i}f_{F} = {}^{C_i}f_{eR_i} \tag{6.3.14} $$

$$ {}^{P_i}\mathbf{f}_{R_i} = {}^{P_i}\mathbf{P}_{C_i}\, {}^{C_i}\mathbf{f}_F \tag{6.3.15} $$

$$ {}^{A_i}\mathbf{f}_{W_i} = {}^{A_i}\mathbf{P}_{P_i}\, {}^{P_i}\mathbf{f}_{R_i} + \left\{ {}^{A_i}\mathbf{P}_{W_i}\, {}^{W_i}\mathbf{f}_{iW_i} + {}^{A_i}\mathbf{P}_{G(W_i)}\, {}^{G(W_i)}\mathbf{f}_{gW_i} + {}^{A_i}\mathbf{P}_{W_i}\, {}^{W_i}\mathbf{f}_{aW_i} \right\} \tag{6.3.16} $$

We apply the joint coupling matrices in Section 6.3.3 and the internal forces/torques in (6.3.14)-(6.3.16) to formulate the secondary force/torque equations-of-motion for i=1,...,4 according to (3.3.33):

$$ \left(\mathbf{I} - {}^{C_i}\mathbf{C}_{FR_i}\, {}^{C_i}\mathbf{C}_{R_iF}\right) {}^{C_i}\mathbf{f}_F = \mathbf{0} \tag{6.3.17} $$

$$ \left(\mathbf{I} - {}^{P_i}\mathbf{C}_{R_iW_i}\, {}^{P_i}\mathbf{C}_{W_iR_i}\right) {}^{P_i}\mathbf{f}_{R_i} = \mathbf{0} \tag{6.3.18} $$

$$ \left(\mathbf{I} - {}^{A_i}\mathbf{C}_{W_iB}\, {}^{A_i}\mathbf{C}_{BW_i}\right) {}^{A_i}\mathbf{f}_{W_i} = \mathbf{0} \tag{6.3.19} $$

The required coupling matrices are computed from the coupling matrices in Section 6.3.3 by applying the coupling matrix mask $\mathbf{C}_{mask}$ in (3.3.25) according to (3.3.24):

$$ {}^{C_i}\mathbf{C}_{FR_i} = {}^{C_i}\mathbf{C}_{R_iF} * \mathbf{C}_{mask} \tag{6.3.20} $$

$$ {}^{P_i}\mathbf{C}_{R_iW_i} = {}^{P_i}\mathbf{C}_{W_iR_i} * \mathbf{C}_{mask} \tag{6.3.21} $$

$$ {}^{A_i}\mathbf{C}_{W_iB} = {}^{A_i}\mathbf{C}_{BW_i} * \mathbf{C}_{mask} \tag{6.3.22} $$

Of the *seventy-two* scalar equations in (6.3.17)-(6.3.19), *fifty-two* are trivial (i.e., 0=0). The *twenty* nontrivial equations are: *three* at each of the contact-points (C_i for i=1,...,4) corresponding to the x, y and z-rotational wheel DOFs at the contact points (higher-pair joints), and *one* at each of the eight joints (R_i and W_i for i=1,...,4) corresponding to the x-rotational joint DOF axes (lower-pair joints).

6.3.5 Companion Kinematic Transformations

We transform the velocities in the inertial and viscous frictional components of the force/torque equations-of-motion in (6.3.13)-(6.3.19) from the natural instantaneously coincident coordinate systems to the WMR body and joint velocities. We first define the joint velocity six-vectors:

$$ {}^{B_t}\mathbf{v}_{L_b} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}^T \quad \text{and} \quad {}^{A_i}\mathbf{v}_{W_i} = \begin{pmatrix} 0 & 0 & 0 & {}^{A_i}\omega_{W_ix} & 0 & 0 \end{pmatrix}^T \qquad \text{for i=1,...,4} $$

and the WMR body velocity six-vector

$$\overline{{}^{B}v}_{B} = \left(\begin{array}{cccccc} \overline{{}^{B}v}_{Bx} & \overline{{}^{B}v}_{By} & 0 & 0 & 0 & \overline{{}^{B}\omega}_{Bz} \end{array} \right)^{T}$$

The velocity six-vector ${}^{B_t}v_{L_b}$ between the WMR body and the load is zero because we assume that the load cannot slide upon the body. The wheel i rotational velocity about its axle ${}^{A_i}\omega_{W_ix}$ is the only nonzero component of the velocity six-vector ${}^{A_i}v_{W_i}$ between the axle and the wheel, and the WMR body velocity six-vector $\overline{{}^{B}v}_{B}$ has only nonzero components in the x and y-translational and z-rotational axes corresponding to the three WMR DOFs. Transformations of the roller and the contact point velocities are not required because they are not utilized in the force/torque equations-of-motion. We compute recursively the velocities of the Uranus coordinate systems according to the velocity transformations in (3.4.8) and (3.4.9) to obtain:

$$\overline{{}^{B_t}v}_{B_t} = {}^{B_t}L_{B}\,\overline{{}^{B}v}_{B} \tag{6.3.23}$$

$$\overline{{}^{L_b}v}_{L_b} = {}^{L_b}L_{B_t}\,\overline{{}^{B_t}v}_{B_t} + {}^{L_b}J_{B_t}\,{}^{B_t}v_{L_b} \tag{6.3.24}$$

$$\overline{{}^{L}v}_{L} = {}^{L}L_{L_b}\,\overline{{}^{L_b}v}_{L_b} \tag{6.3.25}$$

and for i=1,...,4

$$\overline{{}^{A_i}v}_{A_i} = {}^{A_i}L_{B}\,\overline{{}^{B}v}_{B} \tag{6.3.26}$$

$$\overline{{}^{W_i}v}_{W_i} = {}^{W_i}L_{A_i}\,\overline{{}^{A_i}v}_{A_i} + {}^{W_i}J_{A_i}\,{}^{A_i}v_{W_i}. \tag{6.3.27}$$

In parallel with the velocity transformations, we define the joint acceleration six-vectors

$${}^{B_t}a_{L_b} = \left(\begin{array}{cccccc} 0 & 0 & 0 & 0 & 0 & 0 \end{array} \right)^{T} \text{ and } {}^{A_i}a_{W_i} = \left(\begin{array}{cccccc} 0 & 0 & 0 & {}^{A_i}\alpha_{W_ix} & 0 & 0 \end{array} \right)^{T},$$

the WMR body acceleration six-vector

$$\overline{{}^{B}a}_{B} = \left(\begin{array}{cccccc} \overline{{}^{B}a}_{Bx} & \overline{{}^{B}a}_{By} & 0 & 0 & 0 & \overline{{}^{B}\alpha}_{Bz} \end{array} \right)^{T},$$

the rotational velocity product vectors according to (3.4.23)

$$\bar{A}_i\omega^r_{W_i}=\left(\bar{A}\omega_{Ax}\,{}^{A}\omega_{Wy} \quad \bar{A}\omega_{Ax}\,{}^{A}\omega_{Wz} \quad \bar{A}\omega_{Ay}\,{}^{A}\omega_{Wx} \quad \bar{A}\omega_{Ay}\,{}^{A}\omega_{Wz} \quad \bar{A}\omega_{Az}\,{}^{A}\omega_{Wx} \quad \bar{A}\omega_{Az}\,{}^{A}\omega_{Wy}\right)^T,$$

$$\bar{B}_t\omega^r_{L_b}=\begin{pmatrix}0 & 0 & 0 & 0 & 0 & 0\end{pmatrix}^T,$$

the squared velocity product vectors according to (3.4.24)

$$\bar{B}\omega^2_B=\begin{pmatrix}0 & 0 & \bar{B}\omega^2_{Bz} & 0 & 0 & 0\end{pmatrix}^T,$$

$$\bar{A}_i\omega^2_{A_i}=\left(\bar{A}\omega^2_{Ax} \quad \bar{A}\omega^2_{Ay} \quad \bar{A}\omega^2_{Az} \quad \bar{A}\omega_{Ax}\,\bar{A}\omega_{Ay} \quad \bar{A}\omega_{Ax}\,\bar{A}\omega_{Az} \quad \bar{A}\omega_{Ay}\,\bar{A}\omega_{Az}\right)^T,$$

$$\bar{L}_b\omega^2_{L_b}=\left(\bar{L}_b\omega^2_{L_bx} \quad \bar{L}_b\omega^2_{L_by} \quad \bar{L}_b\omega^2_{L_bz} \quad \bar{L}_b\omega_{L_bx}\,\bar{L}_b\omega_{L_by} \quad \bar{L}_b\omega_{L_bx}\,\bar{L}_b\omega_{L_bz} \quad \bar{L}_b\omega_{L_by}\,\bar{L}_b\omega_{L_bz}\right)^T,$$

$$\bar{B}_t\omega^2_{B_t}=\left(\bar{B}_t\omega^2_{B_tx} \quad \bar{B}_t\omega^2_{B_ty} \quad \bar{B}_t\omega^2_{B_tz} \quad \bar{B}_t\omega_{B_tx}\,\bar{B}_t\omega_{B_ty} \quad \bar{B}_t\omega_{B_tx}\,\bar{B}_t\omega_{B_tz} \quad \bar{B}_t\omega_{B_ty}\,\bar{B}_t\omega_{B_tz}\right)^T,$$

and the translational velocity product vectors according to (3.4.25)

$$\bar{A}_i\omega^t_{W_i}=\bar{B}_t\omega^t_{L_b}=\begin{pmatrix}0 & 0 & 0 & 0 & 0 & 0\end{pmatrix}^T$$

We then transform recursively the accelerations according to (3.4.12) and (3.4.13) to obtain:

$$\bar{B}_t\mathbf{a}_{B_t}={}^{B_t}\mathbf{L}_B\,\bar{B}\mathbf{a}_B+{}^{B_t}\mathbf{S}_B\,\bar{B}\omega^2_B \tag{6.3.28}$$

$$\bar{L}_b\mathbf{a}_{L_b}={}^{L_b}\mathbf{L}_{B_t}\,\bar{B}_t\mathbf{a}_{B_t}+{}^{L_b}\mathbf{J}_{B_t}\,{}^{B_t}\mathbf{a}_{L_b}+{}^{L_b}\mathbf{R}_{B_t}\,\bar{B}_t\omega^r_{L_b}+{}^{L_b}\mathbf{S}_{B_t}\,\bar{B}_t\omega^2_{B_t}+{}^{L_b}\mathbf{T}_{B_t}\,\bar{B}_t\omega^t_{L_b} \tag{6.3.29}$$

$$\bar{L}\mathbf{a}_L={}^{L}\mathbf{L}_{L_b}\,\bar{L}_b\mathbf{a}_{L_b}+{}^{L}\mathbf{S}_{L_b}\,\bar{L}_b\omega^2_{L_b} \tag{6.3.30}$$

and for i=1,...,4

$$\bar{A_i}a_{A_i} = {}^{A_i}L_B\,\bar{B}a_B + {}^{A_i}S_B\,\bar{B}\omega_B^2 \qquad (6.3.31)$$

$$\bar{W_i}a_{W_i} = {}^{W_i}L_{A_i}\,\bar{A_i}a_{A_i} + {}^{W_i}J_{A_i}\,{}^{A_i}a_{W_i} + {}^{W_i}R_{A_i}\,\bar{A_i}\omega_{W_i}^r + {}^{W_i}S_{A_i}\,\bar{A_i}\omega_{A_i}^2 + {}^{W_i}T_{A_i}\,\bar{A_i}\omega_{W_i}^t \qquad (6.3.32)$$

6.3.6 Dynamic Equations-of-Motion

In Section 6.3.4, we computed the six primary force/torque equations-of-motion and the N_s=20 secondary force/torque equations-of-motion. We program MACSYMA to compute and substitute the link velocities and accelerations (in Section 6.3.5) referenced to instantaneously coincident coordinate systems into the *twenty-six* force/torque equations-of-motion to obtain the dynamic model of Uranus displayed in Tables 6.1 and 6.2. For these symbolic modeling computations, MACSYMA required 16.82 cpu minutes of execution time on a time-shared Vax 11/8800 computer under the Unix operating system to generate automatically the Uranus dynamic model.

The six *primary* equations-of-motion (P1)-(P6) are displayed in Table 6.1. The three *translational equilibrium equations* (P1)-(P3) have a straightforward physical interpretation. The x and y-translational equations (P1) and (P2) sum the viscous frictional, inertial, and environmental contact forces in the x and y-directions, respectively, to zero. The z-translational equation (P3) sums the gravitational and normal environmental contact forces to zero. The *rotational equilibrium equations* (P4)-(P6) are complex functions of the actuator, viscous frictional, inertial, and environmental contact forces/torques and the frictional coefficients.

The twenty *secondary* equations-of-motion (S1)-(S20) are displayed in Table 6.2: equations (S1)-(S12) are formulated at the wheel contact points, equations (S13)-(S16) are formulated at the roller axles, and equations (S17)-(S20) are formulated at the wheel axles. The complexity of the secondary equations-of-motion increases as we ascend from the floor contact joint towards the main body. Each secondary equation-of-motion corresponding to a joint axis with stiction; i.e., (S1)-(S4) and (S9)-(S20), becomes null if stiction occurs (i.e., when the stictional coefficient is unity).

Table 6.1: Uranus Primary Dynamic Equations-of-Motion

P1: $-\mu_{vBx}\,{}^{\bar{B}}v_{Bx} - (4\,m_W + m_B + m_L)\,{}^{\bar{B}}a_{Bx}$
$+ \sqrt{2}/2\,(-f_{y4} + f_{x4} + f_{y3} + f_{x3} - f_{y2} + f_{x2} + f_{y1} + f_{x1}) = 0$

P2: $-\mu_{vBy}\,{}^{\bar{B}}v_{By} - (4\,m_W + m_B + m_L)\,{}^{\bar{B}}a_{By}$
$+ \sqrt{2}/2\,(f_{y4} + f_{x4} + f_{y3} - f_{x3} + f_{y2} + f_{x2} + f_{y1} - f_{x1}) = 0$

P3: $-4\,g\,m_W - g\,m_B - g\,m_L + f_{z4} + f_{z3} + f_{z2} + f_{z1} = 0$

P4: $\sqrt{2}/2\,\{-\zeta_{sa1}\,\zeta_{sw1}\,\zeta_{sr1}\,\tau_{x1} + [(1-\zeta_{sw1})\,R + \zeta_{sw1}\,(1-\zeta_{sr1})\,r - l_c]\,f_{y1}$
$+ \beta(1-\zeta_{sw1})\,\tau_1 + \beta(1-\zeta_{sw2})\,\tau_2 + \beta(1-\zeta_{sw3})\,\tau_3 + \beta(1-\zeta_{sw4})\,\tau_4$
$- \zeta_{sa2}\,\zeta_{sr2}\,\zeta_{sw2}\,\tau_{x2} + [(1-\zeta_{sw2})\,R + (1-\zeta_{sr2})\,\zeta_{sw2}\,r - l_c]\,f_{y2}$
$- \zeta_{sa3}\,\zeta_{sr3}\,\zeta_{sw3}\,\tau_{x3} + [(1-\zeta_{sw3})\,R + (1-\zeta_{sr3})\,\zeta_{sw3}\,r - l_c]\,f_{y3}$
$- \zeta_{sa4}\,\zeta_{sr4}\,\zeta_{sw4}\,\tau_{x4} + [(1-\zeta_{sw4})\,R + (1-\zeta_{sr4})\,\zeta_{sw4}\,r - l_c]\,f_{y4}$
$+[(\zeta_{sw1}-1)R+l_c-\zeta_{ca1}\zeta_{sw1}\zeta_{sr1}]f_{x1} + [(1-\zeta_{sw2})R-l_c-\zeta_{ca2}\zeta_{sr2}\zeta_{sw2}]f_{x2}$
$+[(\zeta_{sw3}-1)R+l_c-\zeta_{ca3}\zeta_{sr3}\zeta_{sw3}]f_{x3} + [(1-\zeta_{sw4})R-l_c-\zeta_{ca4}\zeta_{sr4}\zeta_{sw4}]f_{x4}\}$
$- [\sqrt{2}/2\,(\zeta_{r1}\,\zeta_{sr1} + \zeta_{cr1})\,\zeta_{sw1} + l_b + \zeta_{cw1}]\,f_{z1} + I_{Wx}\,\zeta_{sw1}\,{}^{A_1}\alpha_{W_{1x}}$
$- [\sqrt{2}/2\,(\zeta_{r2}\,\zeta_{sr2} + \zeta_{cr2})\,\zeta_{sw2} + l_b + \zeta_{cw2}]\,f_{z2} + I_{Wx}\,\zeta_{sw2}\,{}^{A_2}\alpha_{W_{2x}}$
$- [\sqrt{2}/2\,(\zeta_{r3}\,\zeta_{sr3} + \zeta_{cr3})\,\zeta_{sw3} - l_b + \zeta_{cw3}]\,f_{z3} + I_{Wx}\,\zeta_{sw3}\,{}^{A_3}\alpha_{W_{3x}}$
$- [\sqrt{2}/2\,(\zeta_{r4}\,\zeta_{sr4} + \zeta_{cr4})\,\zeta_{sw4} - l_b + \zeta_{cw4}]\,f_{z4} + I_{Wx}\,\zeta_{sw4}\,{}^{A_4}\alpha_{W_{4x}}$
$+ 4\,m_W\,(l_c - R)\,{}^{\bar{B}}a_{By} + (\zeta_{cw4} + \zeta_{cw3} + \zeta_{cw2} + \zeta_{cw1})\,g\,m_W = 0$

P5: $\sqrt{2}/2\,\{\zeta_{sa1}\zeta_{sr1}\tau_{x1} + (l_c+\zeta_{ca1}\zeta_{sr1})f_{x1} + [(\zeta_{sr1}-1)r+l_c]f_{y1} - \zeta_{sa2}\zeta_{sr2}\tau_{x2}$
$+(l_c-\zeta_{ca2}\zeta_{sr2})f_{x2} - [(\zeta_{sr2}-1)r+l_c]f_{y2} + \zeta_{sa3}\zeta_{sr3}\tau_{x3} + (l_c+\zeta_{ca3}\zeta_{sr3})f_{x3}$
$+[(\zeta_{sr3}-1)r+l_c]f_{y3} - \zeta_{sa4}\zeta_{sr4}\tau_{x4} + (l_c-\zeta_{ca4}\zeta_{sr4})f_{x4} - [(\zeta_{sr4}-1)r+l_c]f_{y4}\}$
$+[\sqrt{2}/2(\zeta_{r1}\zeta_{sr1}+\zeta_{cr1})+l_a]f_{z1} + [-\sqrt{2}/2(\zeta_{r2}\,\zeta_{sr2}+\zeta_{cr2})-l_a]f_{z2}$
$+[\sqrt{2}/2(\zeta_{r3}\zeta_{sr3}+\zeta_{cr3})-l_a]f_{z3} + [-\sqrt{2}/2(\zeta_{r4}\zeta_{sr4}+\zeta_{cr4})+l_a]f_{z4}$
$+4m_W(R-l_c)\,{}^{\bar{B}}a_{Bx} + (I_{Wx}-2I_{Wy})\,({}^{A_4}\omega_{W_{4x}}+{}^{A_3}\omega_{W_{3x}}+{}^{A_2}\omega_{W_{2x}}+{}^{A_1}\omega_{W_{1x}})\,{}^{\bar{B}}\omega_{Bz} = 0$

P6: $\sqrt{2}/2\{-(l_b+l_a)f_{x1}+(l_a-l_b)f_{y1}-(l_b+l_a)f_{x2}+(l_b-l_a)f_{y2}+(l_b+l_a)f_{x3}+(l_b-l_a)f_{y3}$
$+(l_b+l_a)f_{x4}+(l_a-l_b)f_{y4}\}-\varepsilon_{vBz}\,{}^{\bar{B}}\omega_{Bz}-[4(l_b^2+l_a^2)m_W+4I_{Wy}+I_{Bz}+I_{Lz}]\,{}^{\bar{B}}\alpha_{Bz}$
$+\zeta_{sc1}\tau_{z1}+\zeta_{cc1}f_{z1}+\zeta_{sc2}\tau_{z2}+\zeta_{cc4}f_{z4}+\zeta_{cc2}f_{z2}+\zeta_{sc3}\tau_{z3}+\zeta_{cc3}f_{z3}+\zeta_{sc4}\tau_{z4} = 0$

Table 6.2: Uranus Secondary Dynamic Equations-of-Motion

S1-S4: $(1 - \zeta_{sai})\,(\tau_{xi} + \zeta_{ri}\, f_{zi} + \zeta_{cai}\, f_{xi}) = 0$ for i=1,...,4

S5-S8: $\tau_{yi} = 0$ for i=1,...,4

S9-S12: $(1 - \zeta_{sci})\,(\tau_{zi} + \zeta_{cci}\, f_{zi}) = 0$ for i=1,...,4

S13-S16: $(1 - \zeta_{sri})\,[\zeta_{sai}\,\tau_{xi} + r\, f_{yi} + (\zeta_{ri} + \zeta_{cri})\, f_{zi} + \zeta_{cai}\, f_{xi}] = 0$ for i=1,...,4

S17: $(1 - \zeta_{sw1})\,\{\beta\,\tau_1 + \sqrt{2}/2\,\zeta_{sa1}\,\zeta_{sr1}\,\tau_{x1} + \sqrt{2}/2\,(\zeta_{ca1}\,\zeta_{sr1} - R)\, f_{x1}$
$- g\,\zeta_{cw1}\, m_W + \sqrt{2}/2\,(R + r + \zeta_{sr1}\, r)\, f_{y1}$
$+ [\sqrt{2}/2\,(\zeta_{r1}\,\zeta_{sr1} + \zeta_{cr1}) + \zeta_{cw1}]\, f_{z1} - I_{Wx}{}^{A_1}\alpha_{W_1x}\} = 0$

S18: $(1 - \zeta_{sw2})\,\{\beta\,\tau_2 + \sqrt{2}/2\,\zeta_{sa2}\,\zeta_{sr2}\,\tau_{x2} + \sqrt{2}/2\,(R + \zeta_{ca2}\,\zeta_{sr2})\, f_{x2}$
$- g\,\zeta_{cw2}\, m_W + \sqrt{2}/2\,(R - r + \zeta_{sr2}\, r)\, f_{y2}$
$+ [\sqrt{2}/2\,(\zeta_{r2}\,\zeta_{sr2} + \zeta_{cr2}) + \zeta_{cw2}]\, f_{z2} - I_{Wx}{}^{A_2}\alpha_{W_2x}\} = 0$

S19: $(1 - \zeta_{sw3})\,\{\beta\,\tau_3 + \sqrt{2}/2\,\zeta_{sa3}\,\zeta_{sr3}\,\tau_{x3} + \sqrt{2}/2(-R+\zeta_{ca3}\,\zeta_{sr3})f_{x3}$
$- g\,\zeta_{cw3}\, m_W + \sqrt{2}/2\,(R - r + \zeta_{sr3}\, r)\, f_{y3}$
$+ [\sqrt{2}/2\,(\zeta_{r3}\,\zeta_{sr3} + \zeta_{cr3}) + \zeta_{cw3}]\, f_{z3} - I_{Wx}{}^{A_3}\alpha_{W_3x}\} = 0$

S20: $(1 - \zeta_{sw4})\,\{\beta\,\tau_4 + \sqrt{2}/2\,\zeta_{sa4}\,\zeta_{sr4}\,\tau_{x4} + \sqrt{2}/2\,(R + \zeta_{ca4}\,\zeta_{sr4})\, f_{x4}$
$- g\,\zeta_{cw4}\, m_W + \sqrt{2}/2\,(R - r + \zeta_{sr4}\, r)\, f_{y4}$
$+ [\sqrt{2}/2\,(\zeta_{r4}\,\zeta_{sr4} + \zeta_{cr4}) + \zeta_{cw4}]\, f_{z4} - I_{Wx}{}^{A_4}\alpha_{W_4x}\} = 0$

6.3.7 Dynamic Solutions

The complexity and size of the Uranus dynamic model in Tables 6.1 and 6.2 precludes symbolic solution by MACSYMA. We therefore compute numerically the actuated inverse and forward dynamic solutions of Uranus. For both solutions, we reduce the system of twenty-six equations to six equations by applying MACSYMA to solve independently each of the secondary dynamic equations-of-motion for distinct floor reactional forces/torques. We solve (S1)-(S4) for τ_{xi}, (S5)-(S8) for τ_{yi}, (S9)-(S12) for τ_{zi}, (S13)-(S16) for f_{yi}, and (S17)-(S20) for f_{xi}; for i=1,...,4. We then substitute these floor reactional forces/torques into the six primary dynamic equations-of-motion. We call the resulting six equations the *reduced* dynamic model.

We compute the *forward dynamic solution* by solving the reduced dynamic model for the unknown WMR body accelerations ${}^{\bar{B}}a_{Bx}$, ${}^{\bar{B}}a_{By}$, and ${}^{\bar{B}}\alpha_{Bz}$ and the unknown floor reactional forces f_{z1}, f_{z2}, f_{z3}, and f_{z4}, as described in Section 3.5.2. We apply the minimum-norm solution [Brogan85] to solve the system of *six* independent linear algebraic equations in *seven* unknowns. Since the x and y-translational and z-rotational equilibrium equations of the reduced dynamic model correspond to the x, y and z WMR DOF, we program MACSYMA to apply independently Gauss elimination to this system of three equations to compute the unknown WMR body accelerations symbolically in terms of the unknown floor reactional forces. We program MACSYMA to substitute the symbolic solutions into the remaining three equations (i.e., the z-translational and x and y-rotational equilibrium equations of the reduced dynamic model) to obtain a system of three equations in four unknown floor reactional forces. We apply the the minimum norm solution to solve numerically these three equations and compute the floor reactional forces. We compute the numerical values of the WMR body accelerations by substituting the floor reactional forces into the symbolic solution for the WMR body accelerations.

We compute the *actuated inverse dynamic solution* by solving the reduced dynamic model for the unknown actuator torques τ_1, τ_2, τ_3, and τ_4 and the unknown floor reactional forces f_{z1}, f_{z2}, f_{z3}, and f_{z4}, as described in Section 3.5.3. We apply the minimum-norm solution to solve the system of *six* independent linear algebraic equations in *eight* unknowns. In parallel with the

forward dynamic solution, we apply the *minimum norm* solution to compute symbolically the x and y-translational and z-rotational equilibrium equations of the reduced dynamic model for the four actuation torques. We compute numerically the floor reactional forces as in the forward dynamic solution. We then substitute the numerical floor reactional forces into the symbolic solution for the actuator torques to compute numerically the actuator torques.

In both the actuated inverse and forward solutions, the four unknown floor reactional forces are computed from a set of three independent equations according to the minimum norm solution. This solution correctly incorporates the *indeterminate* nature of the four-point contact to a plane. The redundant nature of the actuation structure is also illuminated by the *minimum norm solution* of the four wheel actuator torques *from three equations*. The determined nature of the motion of the WMR is indicated through the determined solution of the three independent equations for the three WMR body accelerations. From the physical point-of-view, this means that: for any combination of actuator torques there is a unique set of WMR body accelerations, but for any set of WMR body accelerations there may be more than one set of actuator torques.

6.3.8 Concluding Remarks

We have computed in this section the dynamic model of Uranus. In the following sections, we apply the kinematic model (in Section 6.2) and the dynamic model to design servo-controllers for Uranus and simulate the combine WMR/servo-controller operation to evaluate the servo-controllers.

6.4 Modeling Parameter Values

The numerical values of the kinematic, dynamic, and frictional parameters applied for the Uranus simulation and control experiments are listed in Table 6.3. The numerical values of these parameters were acquired by direct measurement and indirect computation, and from component specification sheets.

Table 6.3: Uranus Numerical Modeling Parameter Values

Name	Value	Units	Meaning
l_{Bh}	0.3048	m	body height
l_{Lh}	0.3048	m	load height
l_a	0.2667	m	half body width
l_b	0.2667	m	half body length
l_c	0.2270	m	body height above floor
R	0.1127	m	wheel radius
r	0.0159	m	roller radius
m_B	90.72	Kg	body mass
m_L	0	Kg	nominal load mass
m_W	0.3629	Kg	wheel mass
I_{Bz}	4.835	Kg m^2	body inertia about z-axis
I_{Lz}	$m_L l_{Lh}^2/6$	Kg m^2	load inertia about z-axis
I_{Wx}	0.02305	Kg m^2	wheel inertia about x-axis
I_{Wy}	0.02480	Kg m^2	wheel inertia about y-axis
g	9.8	m s^{-2}	gravitational constant
β	4		gear ratio
μ_{vBx}	45.36	Kg s^{-1}	body viscous friction along x-axis
μ_{vBy}	45.36	Kg s^{-1}	body viscous friction along y-axis
ε_{vBz}	2.417	Kg m s^{-1}	body viscous friction about z-axis
bf	0.01		base friction constant
μ_r	0.001*bf		rolling friction constant
μ_c	0.01*bf		Coulomb friction constant
μ_s	0		stiction constant
R_m	2.24	ohms	motor resistance
k_b	0.459	volt s^{-1}	motor back emf constant
k_t	0.459	N m amp^{-1}	motor torque constant
v_{max}	22.4	volt	maximum motor voltage

6.5 Servo-Controller Designs

In this section, we design the two model-based servo-controllers detailed in Chapter 4 for Uranus: resolved motion rate and resolved acceleration servo-controllers. We design the resolved motion rate servo-controller by applying the WMR servo-controller design methodology specified in Section 4.2. We have formulated in Section 6.2 the computational components of the resolved motion rate servo-controller in Figure 4.2: the actuated inverse and sensed forward velocity solutions in Section 6.2.6, and the dead reckoning integration in Section 6.2.8. We developed the velocity-to-voltage and velocity-to-pulse-width command conversion algorithms in Section 4.4. To complete the design, we must specify the controller gains k_{pi} and k_{di} for i=x,y,z. Following the guidelines specified in Section 4.2, we arrive at the servo-controller gains for the resolved motion rate control of Uranus:

$$k_{px} = 6\ s^{-1},\ k_{dx} = 8\ s^{-1},\quad k_{py} = 6\ s^{-1},\quad k_{dy} = 8\ s^{-1},\quad k_{pz} = 6\ s^{-1},\quad k_{dz} = 6\ s^{-1}.$$

We evaluate the performance of this resolved motion rate servo-controller design in Section 6.7.

We design the resolved acceleration servo-controller by applying the WMR servo-controller design methodology specified in Section 4.3. We have formulated the computational components of the resolved acceleration servo-controller in Sections 6.2 and 6.3: the actuated inverse dynamic solution in Section 6.3.7, the sensed forward velocity solution in Section 6.2.6, and the dead-reckoning integration in Section 6.2.8. We developed the torque-to-voltage and torque-to-pulse-width command conversion algorithms in Section 4.4. To complete the design, we must specify the controller gains k_{pi} and k_{di} for i=x,y,z. Following the guidelines specified in Section 4.3, we arrive at the servo-controller gains for the resolved acceleration servo-control of Uranus:

$$k_{px} = 20\ s^{-2},\ k_{dx} = 8.94\ s^{-1},\ k_{py} = 20\ s^{-2},\ k_{dy} = 8.94\ s^{-1},\ k_{pz} = 20\ s^{-2},\ k_{dz} = 8.94\ s^{-1}.$$

We evaluate the performance of this resolved acceleration servo-controller design in Section 6.7.

6.6 Simulation Experiments

6.6.1 Overview

We conduct simulation experiments to evaluate the performance of the kinematics-based resolved motion rate and dynamics-based resolved acceleration servo-controllers for Uranus. One set of experiments is undertaken for each servo-controller. Each set of experiments consists of one simulation with all simulation parameters assigned to their nominal values. The remaining experiments change one or more of the simulation parameters from their nominal values to study their effects on servo-controller performance. In this section, we present the experimental data; in Section 6.7, we interpret the data and evaluate the servo-controllers.

6.6 2 Resolved Motion Rate Servo-Controller Experiments

We tabulate in Table 6.4 the experiments (numbered uk) and corresponding simulation parameters for Uranus under resolved motion rate servo-control. The columns in Table 6.4 correspond to the simulation parameters detailed in Section 5.3.

Table 6.4: Uranus Resolved Motion Rate Simulation Experiments

Experiment Number	Command Type	Sampling Period (s)	Task Time (s)	Base Friction	Load Mass (Kg)	I/O Precision (bits)	Trajectory Type
uk1	voltage	0.2	10.0	0.01	0	16	step
uk2	voltage	0.2	10.0	0	0	16	step
uk3	voltage	0.2	10.0	0.01	0	16	spline
uk4	voltage	0.2	10.0	0	0	16	spline
uk5	voltage	0.2	10.0	0.01	45	16	spline
uk6	voltage	0.2	5.0	0.01	0	16	spline
uk7	pwm	0.2	10.0	0.01	0	16	spline
uk8	voltage	0.2	10.0	0.01	0	8	spline
uk9	voltage	0.2	10.0	0.01	0	16	spline*
uk10	voltage	0.2	10.0	0.01	0	16	circle
uk11	voltage	0.04	10.0	0.01	0	16	spline
uk12	pwm	0.2	5.0	0.01	45	8	spline*
uk13	voltage	0.2	10.0	0.1	0	16	spline

A * in the *trajectory type* column indicates that a simultaneous z-rotational reference trajectory ending at 1.5 radians was applied. The results of the

simulation experiments are tabulated in Table 6.5. The experiment numbers indicate the correspondence between these results and the experimental conditions specified in Table 6.4. The columns of Table 6.5 correspond to the performance measures detailed in Section 5.4.

Table 6.5: Uranus Resolved Motion Rate Simulation Results

Experiment Number	Adds/ Multiplies	Max. x-y Error (m)	Max. θ Error (rad)	Accum. Error (m s)	Settling Time (s)	Steady S. x-y Error (m)	Command Saturates	Simulation Time (min)
uk1	25/25	-	0.000	0.618	1.6	0.026	4	194.1
uk2	25/25	-	0.000	0.603	1.6	0.025	4	15.7
uk3	25/25	0.061	0.000	0.408	-	-	0	204.3
uk4	25/25	0.056	0.000	0.384	-	-	0	16.3
uk5	25/25	0.063	0.000	0.422	-	-	0	204.3
uk6	25/25	0.115	0.000	0.386	-	-	0	106.2
uk7	25/25	0.061	0.001	0.407	-	-	0	157.8
uk8	25/25	0.057	0.005	0.381	-	-	0	179.3
uk9	25/25	0.070	0.068	0.665	-	-	0	52.9
uk10	25/25	0.034	0.000	0.234	-	-	0	219.4
uk11	25/25	0.106	0.000	0.711	-	-	8	236.2
uk12	25/25	0.135	0.143	0.650	-	-	0	43.5
uk13	25/25	0.105	0.000	0.729	-	-	8	198.1

The *overshoot* performance measure is not included in Table 6.5 because overshoot never occurs in any of the simulation experiments. The simulation times in the last column are in cpu minutes.

We plot the nominal step, spline and circle trajectories for Uranus under resolved motion rate servo-control in Figures 6.4, 6.5 and 6.6, respectively. In the figures, the reference WMR trajectory is marked by open squares and the actual WMR trajectory is marked by filled diamonds.

6.6.3 Resolved Acceleration Servo-Controller Experiments

We tabulate in Table 6.6 the experiments (numbered ud) and corresponding simulation parameters for Uranus under resolved acceleration servo-control. The results of the resolved acceleration simulation experiments are tabulated in Table 6.7. We plot the nominal step, spline and circle trajectories for Uranus under resolved acceleration servo-control in Figures 6.7, 6.8 and 6.9, respectively.

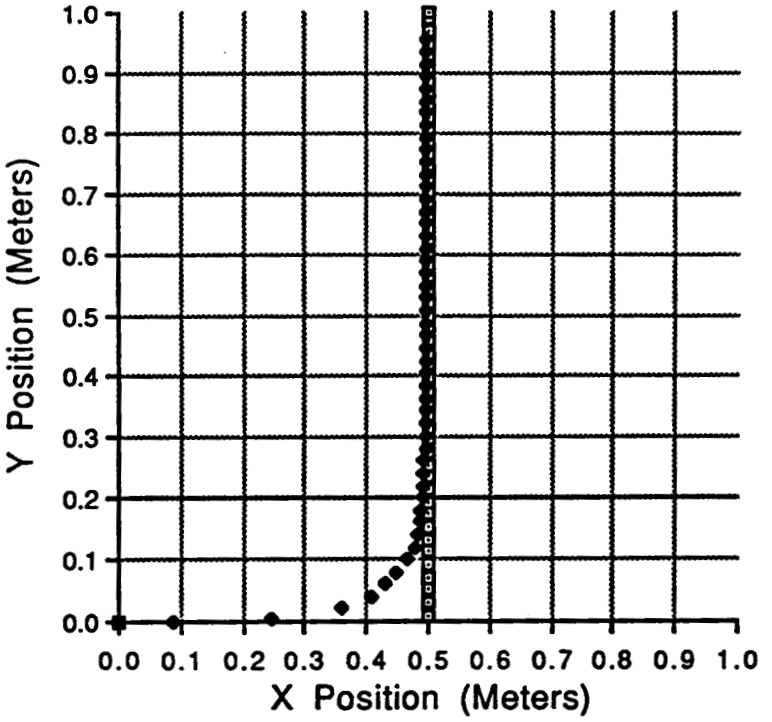


Figure 6.4: Uranus Resolved Motion Rate Control On Nominal Step Reference Trajectory (Experiment uk1)

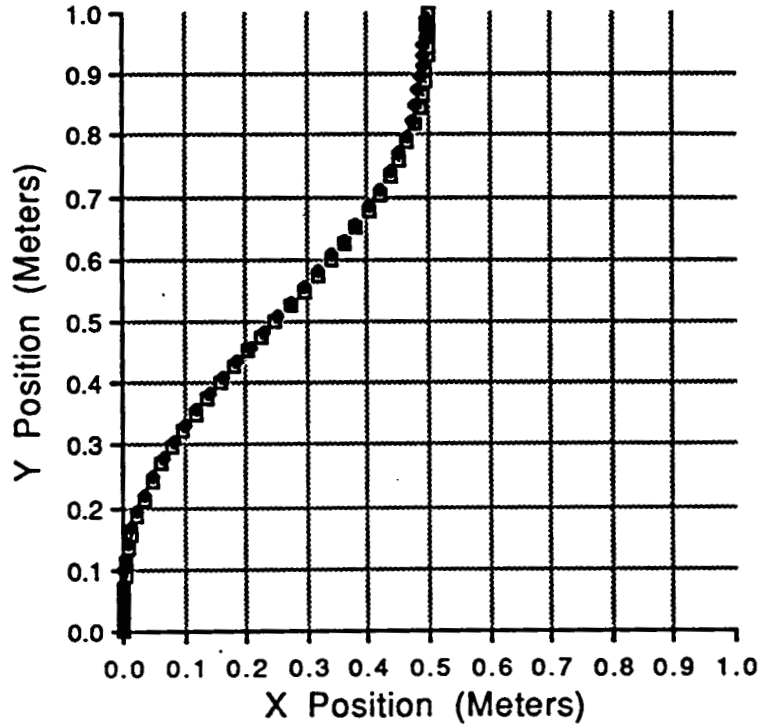


Figure 6.5: Uranus Resolved Motion Rate Control On Nominal Spline Reference Trajectory (Experiment uk3)

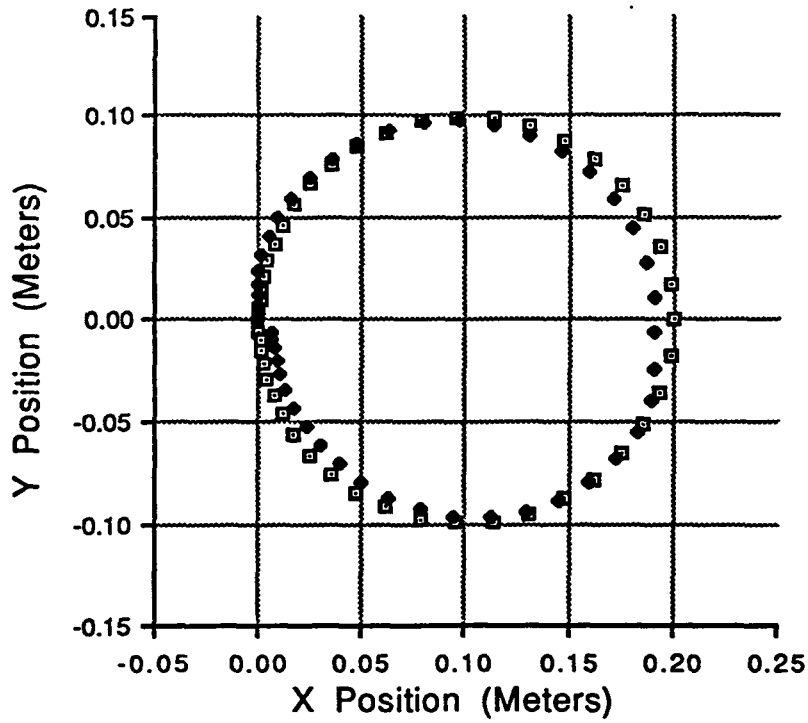


Figure 6.6: Uranus Resolved Motion Rate Control On Nominal Circle Reference Trajectory (Experiment uk10)

Table 6.6: Uranus Resolved Acceleration Simulation Experiments

Experiment Number	Command Type	Sampling Period (s)	Task Time (s)	Base Friction Sim./Con.	Load Mass (Kg) Sim./Con.	I/O Precision (bits)	Trajectory Type
ud1	voltage	0.2	10.0	0.01/0.01	0/0	16	step
ud2	voltage	0.2	10.0	0/0	0/0	16	step
ud3	voltage	0.2	10.0	0.01/0	0/0	16	step
ud4	voltage	0.2	10.0	0.01/0.01	0/0	16	spline
ud5	voltage	0.2	10.0	0/0	0/0	16	spline
ud6	voltage	0.2	10.0	0.01/0	0/0	16	spline
ud7	voltage	0.2	10.0	0.01/0.01	45/0	16	spline
ud8	voltage	0.2	10.0	0.01/0.01	0/45	16	spline
ud9	voltage	0.2	5.0	0.01/0.01	0/0	16	spline
ud10	pwm	0.2	10.0	0.01/0.01	0/0	16	spline
ud11	voltage	0.2	10.0	0.01/0.01	0/0	8	spline
ud12	voltage	0.2	10.0	0.01/0.01	0/0	16	spline*
ud13	voltage	0.2	10.0	0.01/0.01	0/0	16	circle
ud14	voltage	0.04	10.0	0.01/0.01	0/0	16	spline
ud15	pwm	0.2	5.0	0.01/0	45/0	8	spline*
ud16	voltage	0.2	10.0	0.1/0	0/0	16	spline

Table 6.7: Uranus Resolved Acceleration Simulation Results

Experiment Number	Adds/ Multiplies	Max. x-y Error (m)	Max. θ Error (rad)	Accum. Error	Settling Time (s)	Steady S. x-y Error (m)	Command Saturates	Simulation Time (min)
ud1	181/271	-	0.000	0.273	1.0	0.000	6	104.4
ud2	50/54	-	0.000	0.269	1.0	0.000	6	15.0
ud3	50/54	-	0.000	0.304	1.2	0.003	6	168.9
ud4	181/271	0.004	0.001	0.020	-	-	0	88.7
ud5	50/54	0.004	0.000	0.018	-	-	0	16.5
ud6	50/54	0.008	0.000	0.034	-	-	0	210.1
ud7	181/271	0.004	0.000	0.017	-	-	0	91.8
ud8	181/271	0.004	0.001	0.020	-	-	0	85.6
ud9	181/271	0.013	0.000	0.034	-	-	0	45.7
ud10	181/271	0.013	0.003	0.093	-	-	0	153.3
ud11	181/271	0.008	0.003	0.040	-	-	0	98.0
ud12	181/271	0.004	0.007	0.042	-	-	0	52.1
ud13	181/271	0.007	0.003	0.037	-	-	0	72.6
ud14	181/271	0.003	0.001	0.016	-	-	0	104.6
ud15	50/54	0.024	0.074	0.276	-	-	0	46.3
ud16	181/271	0.042	0.001	0.292	-	-	0	191.9

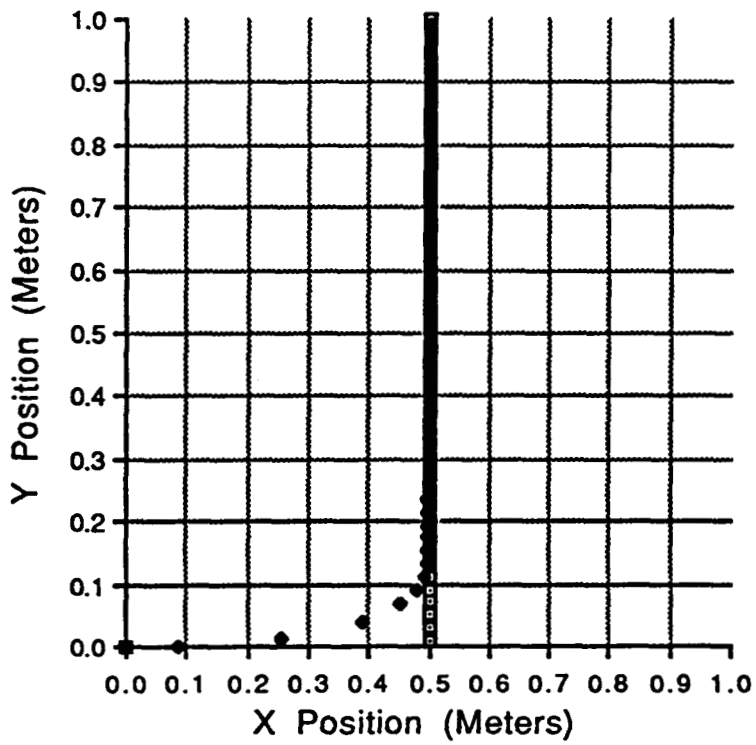


Figure 6.7: Uranus Resolved Acceleration Control On Nominal Step Reference Trajectory (Experiment ud1)

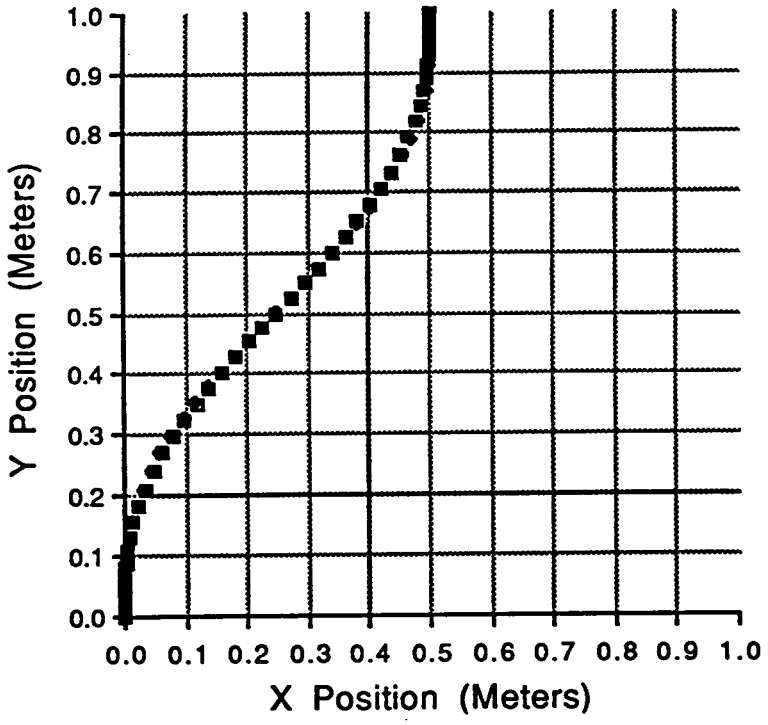


Figure 6.8: Uranus Resolved Acceleration Control On Nominal Spline Reference Trajectory (Experiment ud4)

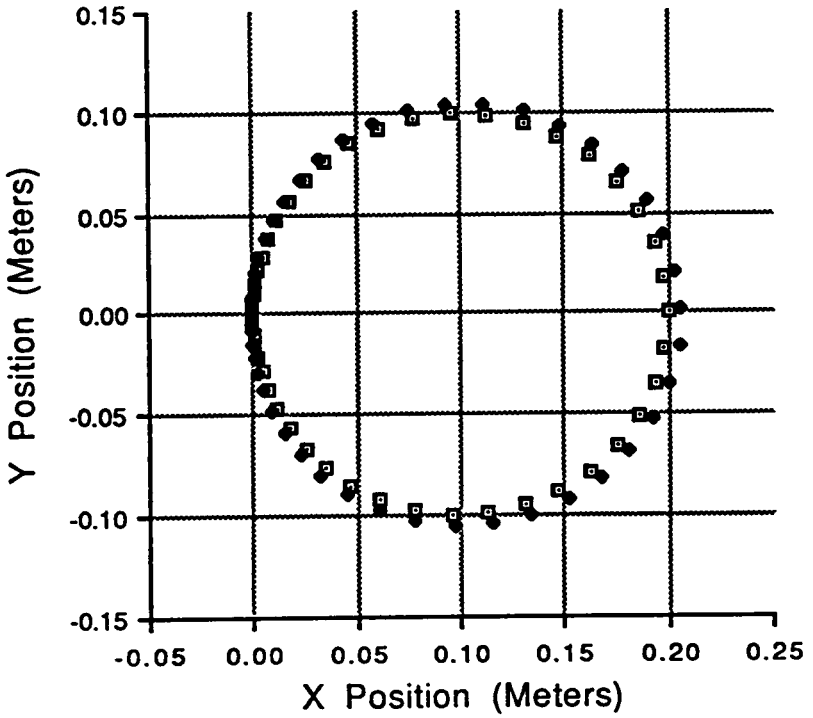


Figure 6.9: Uranus Resolved Acceleration Control On Nominal Circle Reference Trajectory (Experiment ud13)

6.6.4 Concluding Remarks

We evaluate the performance of the resolved motion rate and resolved acceleration servo-controllers by interpreting in Section 6.7 the results of the simulation experiments in Sections 6.6.2 and 6.6.3. In this concluding section, we interpret the simulation times of the experiments (listed in Tables 6.5 and 6.7) to describe how the MACSYMA numerical computational load varies with the simulation parameters.

We observe that the simulation times for Uranus under resolved motion rate servo-control are approximately twice the simulation times for resolved acceleration servo-control when *friction* is simulated (e.g., experiment uk1 requires 194.1 minutes vs. experiment ud1 which requires 104.4 minutes; and experiment uk3 requires 204.3 minutes vs. experiment ud4 which requires 88.7 minutes). This contrasts dramatically with the computational complexities of the servo-control algorithms because the resolved motion rate servo-controller requires only 11% of the computations required by the resolved acceleration servo-controller. When friction is zeroed, the simulation experiments of the two servo-controllers require the same time (e.g., experiment uk2 requires 15.7 minutes vs. experiment ud2 which requires 15.0 minutes; and experiment uk4 requires 16.3 minutes vs. experiment ud5 which requires 16.5 minutes). We also observe that the simulation times are 6 to 12 times larger for experiments with friction than for experiments without friction. From these findings, we conclude that the dominating factor in determining the simulation time of an experiment is the *smoothness* of the simulated WMR under servo-control. Since we have modeled friction as a hard nonlinearity causing instantaneous jumps in forces/torques, inclusion of friction in the simulation adds a high frequency component to the WMR dynamics. The step size of the integration routine within the simulator reduces with quickly-varying high frequency signals. The more integration steps which must be executed, the longer will be the simulation time. The resolved acceleration servo-controller attempts to compensate for the WMR friction variations, producing smoother simulations and shorter simulation times than does the resolved motion rate servo-controller. Our conclusion is substantiated by the simulation times of resolved acceleration control experiments ud3 and ud6 (168.9 minutes and 210.1 minutes, respectively) which approach those of the respective resolved motion rate experiments uk1

(194.1 minutes) and uk3 (204.3 minutes) when frictional compensation within the servo-controller is zeroed; i.e., the WMR is simulated with friction but the resolved acceleration servo-controller frictional parameters are set equal to zero.

6.7 Servo-Controller Evaluations

6.7.1 Overview

We interpret the results of the simulation experiments highlighted in Section 6.6 to evaluate the performance of Uranus under resolved motion rate and resolved acceleration servo-control. We establish our criteria for acceptable WMR/servo-control performance for general-purpose applications. WMRs show potential for applications which require the transport of materials, parts, tools, manipulators, sensors or docking devices over distances which are much greater than the working volumes of stationary manipulators. We direct our performance evaluations towards mobile operation over hard, smooth surfaces where wheeled locomotion is the most power efficient. The requirements of such tasks are controlled motion along spatial (x-y) trajectories. The rate of travel and the orientation of the vehicle along the spatial trajectory are typically unimportant features of these tasks.

We thereby deem the performance of a WMR/servo-controller combination *acceptable* if the simulation experimental results show the ability to approach quickly and follow closely a typical WMR x-y spatial trajectory. A *quick approach* to a reference trajectory will be evidenced by a reduction of at least e^{-1} in the spatial position error within one WMR time constant; i.e., a conventional first-order exponential error decay. We apply the nominal step experiment (i.e., the step trajectory with all simulation parameters set equal to their nominal values) to evaluate this response criterion. To *follow closely* an x-y spatial trajectory, the maximum spatial deviation form the path during trajectory tracking must be less than 5% of the largest WMR dimension (i.e., height, length or width). We apply the nominal spline experiment to evaluate this tracking criterion. We emphasize that our acceptability criertion is based upon spatial deviations from the path and not temporal errors so that time lags in the tracking performance are not as degrading as displacements from the reference path. It

is possible to reduce temporal tracking errors to within allowable limits by time-scaling the reference trajectory before execution [Graettinger88]. Our maximum spatial error performance measure introduced in Chapter 5 must be resolved into two components: spatial deviation from the path and temporal lag to evaluate the acceptability of a WMR/servo-controller combination.

According to our criterion for acceptable WMR/servo-control performance, Uranus under servo-control will show acceptable performance if the 95% settling time during the nominal step experiment is less than 6 seconds and the maximum deviation from the path component of the maximum spatial error during the nominal spline experiment less than 2.7 centimeters.

For each servo-controller, we interpret the response and tracking characteristics to determine whether the performance is acceptable. We then discuss the effects of friction, load variations, finite precision, simultaneous rotation, pulse-width modulation, and sampling period variations on WMR/servo-controller performance to determine the sensitivity of the performance to these realistic operating conditions.

6.7.2 Resolved Motion Rate Servo-Controller Evaluation

The nominal step Experiment uk1 (in Figure 6.4) is the focal point for evaluating the Uranus resolved motion rate servo-controller response characteristics. The spatial response resembles a first-order exponential rise to the reference line trajectory. We observe a settling time of 1.6 s with zero overshoot and a 2.6 cm steady-state error on the approach side of the reference trajectory. The trajectory approach is thus within the 6 s limit for acceptable settling time. The steady-state spatial error consists of two components: a small (approximately 0.5 cm) constant deviation from the path (on the approach side) of the WMR trajectory from the reference due to the uncompensated dry friction, and a larger (approximately 2 cm) trajectory temporal lag component caused by the finite response time between when a position reference is introduced and when the WMR position reaches the reference position. During the design phase, we noted that the steady-state errors decrease as the controller gains are increased.

The nominal step experiment experiences four command saturations because of the initial abrupt change in reference trajectory embodied in a step. These saturations occur during the first sampling period of the simulation when there is the largest error between the reference and actual trajectories. Most of the spline and circle tracking experiments do not experience command saturations because the initial WMR position lies on the reference trajectory.

The nominal spline Experiment uk3 (in Figure 6.5) and nominal circle Experiment uk10 (in Figure 6.6) are representative of typical curved WMR trajectories and are the basis for discussing the Uranus resolved motion rate tracking characteristics. Uranus resolved motion rate servo-control tracks the nominal spline and circle trajectories with maximum spatial errors of 6.1 cm and 3.4 cm, respectively. These maximum errors are dependent upon the speeds and accelerations of the WMR along the trajectories as demonstrated by Experiment uk6. In Experiment uk6, the total task time is reduced to 5 s which increases the reference velocities and accelerations and thereby increases the maximum spatial error to 11.5 cm. We must limit the maximum WMR reference velocities and accelerations by time-scalling the reference trajectory to avoid such large tracking errors. We observe in Figure 6.5 that although the maximum spatial error for the nominal spline is 6.1 cm, the maximum deviation from the path component of that error is less than 1 cm and the remaining component (approximately 5.5 cm) is temporal lag error. The performance is thus within the 2.7 cm maximum for acceptable tracking error.

The nominal circle experiment, which embodies a quickly changing trajectory, is valuable for demonstrating the lag in the WMR trajectory. The WMR temporal lag is manifested in deviations from the path. The position deviations from the path for the nominal circle trajectory thus appear larger than the deviations from the path for the nominal spline trajectory even though the maximum spatial error is smaller. The positional errors at the end of the circle trajectory are due to the uncompensated dry frictions.

We zero the simulated friction in the step Experiment uk2 and spline Experiment uk4 and compare the results to the nominal step and spline responses. There are slight decreases in maximum spatial error and accumulated error in both cases. In spline Experiment uk13, we set the simulated friction to ten times the nominal value. The tracking results show a

large 10.5 cm maximum spatial tracking error. Further increases in WMR friction are likely to be accompanied by similar increases in tracking errors. We conclude that the base friction within the WMR must be kept less than 0.1 through careful mechanical design to enable the practical application of resolved motion rate servo-control to Uranus.

In Experiment uk5, the simulation operates with a 45 Kg mass on Uranus (one-half the mass of the WMR). The maximum spatial error and the accumulated error increase slightly relative to the nominal spline trajectory experiment but the performance remains acceptable for most WMR applications. The resolved motion rate servo-controller is relatively insensitive to load variations.

The precision of actuator commands and feedback signals are limited to 8 bits in Experiment uk8. The results show a small decrease in the maximum spatial error and accumulated error relative to the nominal spline trajectory. We can therefore apply inexpensive microprocessors with 8 bit data busses to the implementation of resolved motion rate servo-control. The decrease in tracking errors may indicate that there are no more than 8 bits of significant precision in our WMR simulations.

We add a simultaneous rotational trajectory to Experiment uk9 and observe the maximum spatial error increase to 7 cm. During the design process, we found that this tracking error due to a rotational reference decreased as the servo-controller gains are increased. We attribute these tracking errors to the uncompensated forces/torques which are coupled from the rotation of the WMR body to the translational equations-of-motion. As a result, Uranus under resolved motion rate servo control provides good trajectory tracking of x-y trajectories, but degraded performance with a simultaneous rotational reference (i.e., three simultaneous DOFs).

Experiment uk7 utilizes PWM commands to the wheel motors. The results are virtually identical to the nominal spline trajectory with voltage control. Since our pulse period is large, we might have expected degraded performance. The results uphold the premise that PWM control is appropriate as long as the pulse period is much smaller than (i.e., at least ten times smaller than) the time

constant of the system under control. The pulse period is 0.2 s and the time constant of the WMR is 2 s.

We have chosen our nominal sampling period (0.2 s) to be one-tenth the time constant of Uranus (2 s). The time constant of Uranus is estimated as the ratio of the mass of the WMR to the viscous frictional coefficient. The resulting nominal step and spline experiments display an adequate level of performance for many practical applications. We reduce the sampling period to 0.04 s in uk11 while keeping the servo-controller gains unchanged. The large 10.6 cm spatial tracking error is attributed to the inappropriate controller gains. The resolved motion rate servo-controller must be redesigned for operation at different sampling periods.

Experiment uk12 is a worst case scenario. Each of the simulation variables (command type, task time, load mass and I/O precision) is chosen to maximize the tracking error. We notice that the tracking errors are compounded by the individual error sources and the resulting maximum spatial error 13.5 cm is unacceptably large. The maximum spatial error of the worst case scenerio Experiment uk12 exceeds the sum of the maximum spatial error of the nominal spline experiment uk3 and the maximum spatial errors of the pulse-width modulation, reduced task time, additional load and 8 bit precision Experiments uk5 through uk9.

6.7.3 Resolved Acceleration Servo-Controller Evaluation

The nominal step Experiment ud1 is the basis for evaluating the Uranus resolved acceleration servo-controller response characteristics. The nominal step response plotted in Figure 6.7 displays the classical characteristics of a linear first-order system. We observe an acceptably short settling time (1 s) with zero overshoot and zero steady-state error. In practical terms this means that for straight line WMR reference trajectories: all WMR position errors will be reduced by 95% in 1 second, 99.75% in 2 seconds, and so on. Six command saturations occur in the nominal step experiment due to the large initial positional errors.

The nominal spline Experiment ud4 (in Figure 6.8) and nominal circle Experiment ud13 (in Figure 6.9) represent typical curved WMR trajectories and

are the basis for evaluating the Uranus resolved acceleration tracking characteristics. Uranus resolved acceleration servo-control tracks the nominal spline and circle trajectories with maximum spatial errors of 0.4 cm and 0.7 cm, respectively, and thus provides adequate tracking performance. These error limits are dependent upon the speeds and accelerations of the WMR along the trajectories as demonstrated by Experiment ud9. In Experiment ud9, the total experiment time is reduced to 5 s which increases the reference velocities and accelerations and thereby increases the maximum spatial error to 1.3 cm. We thus may limit the maximum WMR tracking errors by choosing the reference trajectory appropriately.

We zero the simulated friction in the step Experiment ud2 and spline Experiment ud5 and compare the results to the nominal step and spline responses in Experiments ud1 and ud4. Since there is virtually no response differences between the zero-friction and nominal compensated friction cases, we conclude that the friction compensation in the nominal trajectories is sufficient. We return the WMR frictions to their nominal values but utilize inaccurate zero frictional values in the servo-controller for step Experiment ud3 and spline Experiment ud6. We find an increase in the settling time and accumulated error of the step trajectory and increases in the maximum spatial error and accumulated error for the spline trajectory. Performance thus degrades when the servo-controller does not compensate for frictional forces/torques. This behavior is emphasized in spline Experiment ud16 when the simulated friction is ten times the nominal value and the servo-controller assumes no friction. The tracking results show a 4.2 cm maximum spatial tracking error. Further increases in WMR friction are likely to be accompanied by corresponding increases in tracking errors.

By assuming no friction in the servo-controller, we reduce the total number of additions and multiplications from 452 to 104, a 77% decrease in computational requirements. This represents a marked decrease in controller complexity and required *a priori* knowledge since we need not identify the frictional coefficients for the servo-control algorithm. If the WMR frictions are excessive, the accompanying increases in tracking errors may grow to be unacceptable.

We vary the load in Experiments ud7 and ud8. In Experiment ud7, the simulation operates with a 45 Kg mass on Uranus (one-half the mass of the WMR) while the servo-controller assumes zero mass. In Experiment ud8, the simulation operates with zero load mass while the servo-controller assumes a 45 Kg mass. The results are virtually unchanged from the nominal spline trajectory experiment. The resolved acceleration servo-controller is thus insensitive to load variations.

The precision of actuator commands and feedback signals are limited to 8 bits in Experiment ud11. The results show a doubling of maximum spatial error and accumulated error over the nominal spline trajectory. But the 0.7 cm maximum spatial error of this experiment is tolerable for practical implementations.

We add a simultaneous rotational trajectory to Experiment ud12 and observe no significant degradation of performance. Reference rotations cause no significant spatial tracking errors because feedforward compensation for rotational references is an inherent property of resolved acceleration servo-control.

The application of PWM motor commands in Experiment ud10 causes an increase in the maximum spatial error to 1.3 cm and similar increases in the other error measures. In contrast to resolved motion rate servo-control, PWM adversely effects the performance of resolved acceleration servo-control. The resulting 1.3 cm maximum spatial error and 0.003 radian maximum rotational error, however, are tolerable for many practical applications.

We have chosen our nominal sampling period (0.2 s) to be one-tenth the time constant of Uranus (2 s). The resulting nominal step and spline experiments display an adequate level of performance for most practical applications. In Experiment ud14, we decrease the sampling period to 0.04 s without changing the controller gains causing an improvement in tracking characteristics. The maximum spatial error decreases to 0.3 cm. To achieve this reduction, we have traded 80% of the sampling period for a 25% decrease in maximum spatial error. For many practical applications this marginal decrease in tracking errors is not justified due to the large amount of computation time which must be sacrificed. Moreover, this experiment demonstrates that the

sampling period of resolved acceleration servo-controllers may be adjusted without redesigning the controller gains.

Our worst case scenario in Experiment ud15 shows that the spatial error of the nominal spline Experiment ud4 and the spatial errors of the uncompensated friction Experiment ud6, load variation Experiment ud7, fast trajectory Experiment ud9, PWM Experiment ud10, 8 bit I/O precision Experiment ud11 and simultaneous rotation Experiment ud12 add, producing a 2.4 cm maximum spatial error. In contrast to the worst case resolved motion rate experiment which produced unacceptably large maximum spatial errors of 13.5 cm, the worst case resolved acceleration experiment is within acceptable spatial tracking limits.

6.7.4 Concluding Remarks

We have found that both the resolved motion rate and resolved acceleration servo-controllers are adequate for Uranus control based upon our performance measures introduced in Section 5.4. To summarize our servo-controller evaluations, we compute qualitatively in Table 6.8 the pertinant findings.

Table 6.8: Uranus Servo-Controller Performance Evaluations

	Resolved Motion Rate	Resolved Acceleration
Settling Time	Fast	Faster
Steady-State Errors	Small	None
Spatial Tracking Errors	Moderate	Small
Load Variations	Insensitive	Insensitive
8-Bit I/O Precision	Accurate	Accurate
Simultaneous Rotation	Degrades Performance	Accurate
Pulse-Width Modulation	Insensitive	Degrades Slightly
Reference Computations	Minimal	Extensive
Dynamic Model	Not Required	Required
Computational Requirements	Low	Extensive
A priori Knowledge	Minimal	Extensive

Both the resolved motion rate and resolved acceleration Uranus servo-controllers provide adequate performance for general-purpose indoor applications based upon our criteria specified in Section 6.7.1. In balance, we choose the resolved motion rate Uranus servo-controller because it requires little *a priori* knowledge and has low computational requirements.

6.8 Conclusions and Recommendations

In this chapter, we have applied successfully the development of this dissertation to formulate the kinematic and dynamic models of Uranus. We then applied the models to design and evaluate servo-controllers through simulation experiments.

Uranus is a general-purpose three DOF WMR with the capability to servo simultaneously to independent x, y and θ trajectories. Three DOF WMR operation is required for the simultaneous operation of an onboard manipulator or for docking maneuvers which require independent differential motions along all three axes. For many applications (such as parts and materials transport), we need not control simultaneously the WMR orientation and translation. These differing WMR applications have differing control requirements. Analogously, the resolved motion rate and resolved acceleration servo-controllers have differing advantages. We must therefore match the servo-controller to the application.

We recommend that a resolved motion rate servo-controller be applied to such applications as mobile transport of parts, tools and materials, which may be divided into translation and rotation tasks. These tasks may tolerate small tracking errors while taking advantage of the design and computational simplicity of resolved motion rate servo-control. We recognize that Uranus under resolved motion rate servo-control will perform adequately for most of the presently concieved WMR tasks. Resolved acceleration servo-control should be applied to high-accuracy applications requiring simultaneous rotations which are intolerant of steady-state errors or degradations in tracking performance due to simultaneous rotations.

Actuator conflict with Pluto [Muir84a] was so severe that adequate control was not possible. Even though the potential for actuator conflict exists with Uranus because of its overdetermined actuation structure, we have not observed actuator conflict in any of the resolved motion rate or resolved acceleration servo-controller simulation experiments.

In Chapter 7, we model and control Bicsun-Bicas. Our study of two different WMRs (Uranus having three DOFs and Bicsun-Bicas having two DOFs) is designed to provide the opportunity to compare and contrast the findings and arrive at generally applicable recommendations. We compare and interpret physically our findings in the concluding section of Chapter 7.

Chapter 7
Case Study II: Bicsun-Bicas, A Two DOF WMR Prototype

7.1 Introduction

To continue our case studies, we apply in this chapter our WMR modeling and control methodologies to the two DOF WMR prototype *Bicsun-Bicas*. Bicsun-Bicas is a four-wheeled WMR of our design having the kinematic structure of a conventional wheel chair depicted in Figure 7.1. The name Bicsun-Bicas (derived from our WMR nomenclature in Appendix 1) refers to a WMR consisting of two (Bi), conventional (c), steered (s), unactuated (u) and not-sensed (n) wheels and two conventional, actuated (a), and sensed (s) wheels. Bicsun-Bicas has two free-wheeling castors and two diametrically opposed drive wheels. We assume that the robot dimensions, masses, inertias, and motors are identical to Uranus to allow a comparative evaluation of the two WMRs. Our two case studies allow us to compare and contrast the kinematic and dynamic models and servo-controllers of Uranus (in Chapter 6) and Bicsun-Bicas. Uranus provides three DOF locomotion by four actuated omnidirectional wheels; whereas, Bicsun-Bicas provides two DOF locomotion by two actuated conventional non-steered wheels and two unactuated steered wheels (i.e., two castors).

Throughout this chapter, we follow the format of the companion case study of Uranus in Chapter 6. In Sections 7.2 and 7.3, we model the kinematics and dynamics of Bicsun-Bicas. We tabulate the numerical values of the modeling parameters in Section 7.4 and design WMR servo-controllers in Section 7.5. In Section 7.6, we describe, tabulate, and plot the results of our simulation experiments. Then, in Section 7.7, we evaluate and compare the

performances of the servo-controllers with the Uranus servo-controllers. Finally, in Section 7.8, we advance our conclusions and explain physically the similarities and differences between the modeling and control of Uranus and Bicsun-Bicas.

7.2 Kinematic Modeling

7.2.1 Overview

We model the kinematics of Bicsun-Bicas to illustrate the application of our kinematic modeling procedure in Chapter 2. We begin in Section 7.2.2 by assigning coordinate systems to Bicsun-Bicas. In Section 7.2.3, we compute the homogeneous coordinate transformation matrices. Then in Section 7.2.4, we formulate the wheel Jacobians to compute the wheel velocity and acceleration equations-of-motion. We characterize the mobility of Bicsun-Bicas in Section 7.2.5 by applying the mobility, actuation and sensing characterization trees in Appendix 7. In Section 7.2.6, we compute the kinematic solutions; specifically, the complete inverse velocity solution, the actuated inverse velocity solution, the complete inverse acceleration solution, and the sensed forward velocity solution. We address translational wheel slip detection in Section 7.2.7 and formulate a dead reckoning algorithm in Section 7.2.8. In Section 7.2.9, we conclude the kinematic modeling of Bicsun-Bicas.

7.2.2 Coordinate System Assignments

The Bicsun-Bicas coordinate systems are assigned in Figure 7.1 according to our kinematic modeling coordinate system assignments defined in Table 2.2. The body coordinate system B is fixed to the WMR body. There is a contact point coordinate system C_j for j=1 and 2 assigned to each of the non-steered conventional wheels and hip H_k, steering S_k and contact point C_k coordinate systems for k=3 and 4 assigned to each of the steered conventional wheels. The wheel and caster radii are $R_1=R_2=R$, and $r_3=r_4=r$, respectively. The axle, and wheel coordinate systems A_i and W_i, for i=1,...,4, which are required for formulating the wheel equations-of-motion and the dynamic model as specified in Table 3.1, are shown in Figure 7.2.

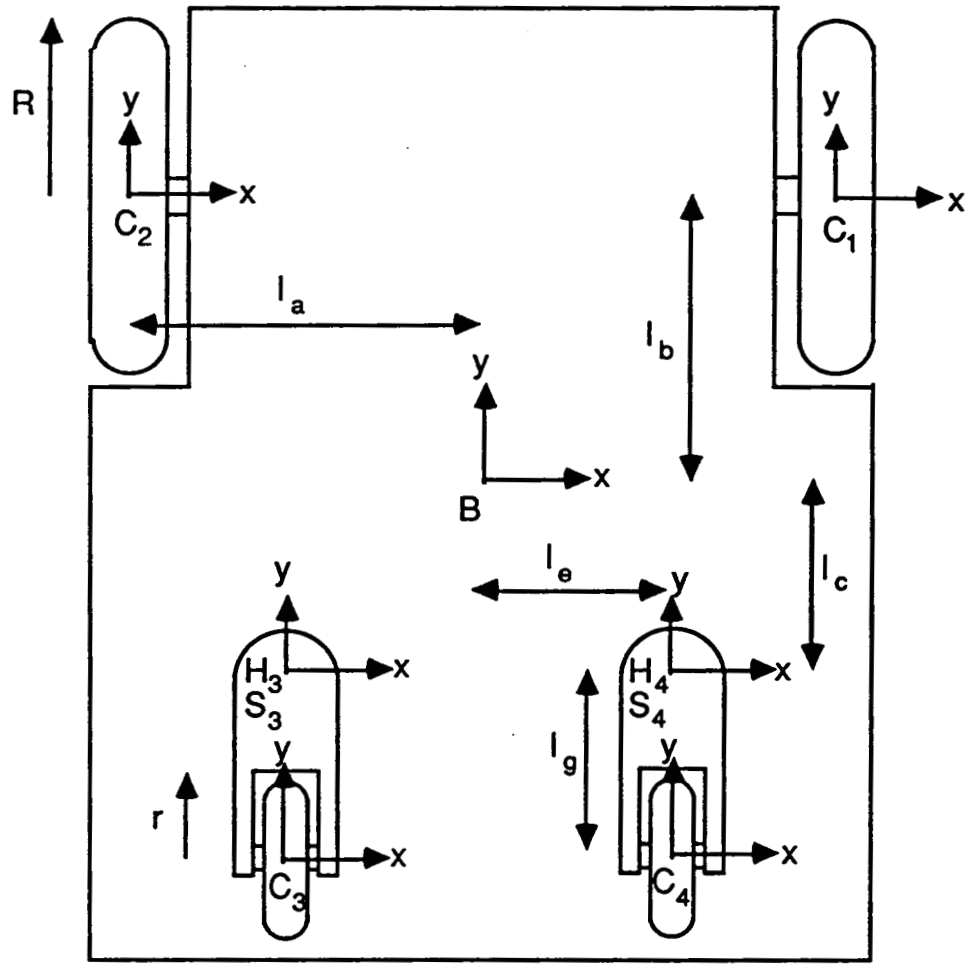


Figure 7.1: Bicsun-Bicas Kinematic Coordinate System Assignments

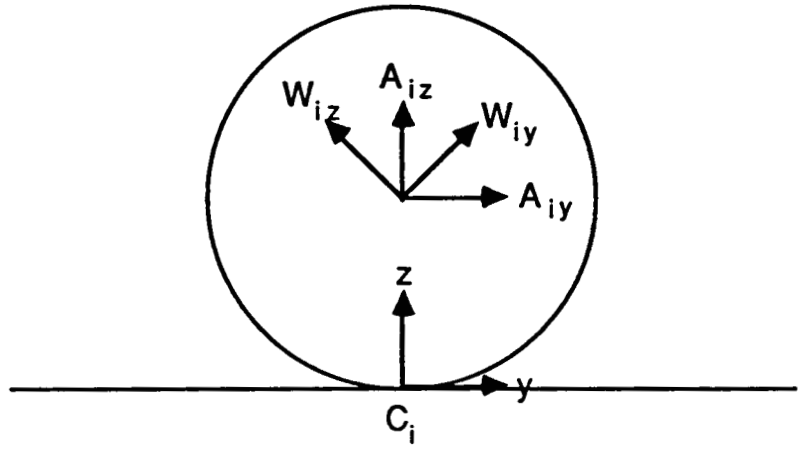


Figure 7.2: Additional Bicsun-Bicas Coordinate System Assignments

7.2.3 Coordinate Transformation Matrices

The *eight* coordinate transformation matrices required for the kinematic modeling of Bicsun-Bicas are:

$$ {}^{B}T_{C_1} = \begin{pmatrix} 1 & 0 & 0 & l_a \\ 0 & 1 & 0 & l_b \\ 0 & 0 & 1 & -l_d\text{-}R \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad {}^{B}T_{C_2} = \begin{pmatrix} 1 & 0 & 0 & -l_a \\ 0 & 1 & 0 & l_b \\ 0 & 0 & 1 & -l_d\text{-}R \\ 0 & 0 & 0 & 1 \end{pmatrix} $$

$$ {}^{B}T_{H_3} = \begin{pmatrix} 1 & 0 & 0 & -l_e \\ 0 & 1 & 0 & -l_c \\ 0 & 0 & 1 & -l_f \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad {}^{B}T_{H_4} = \begin{pmatrix} 1 & 0 & 0 & l_e \\ 0 & 1 & 0 & -l_c \\ 0 & 0 & 1 & -l_f \\ 0 & 0 & 0 & 1 \end{pmatrix} $$

$$ {}^{H_3}\Phi_{S_3} = \begin{pmatrix} \cos{}^{H_3}\theta_{S_3z} & -\sin{}^{H_3}\theta_{S_3z} & 0 & 0 \\ \sin{}^{H_3}\theta_{S_3z} & \cos{}^{H_3}\theta_{S_3z} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad {}^{S_3}T_{C_3} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -l_g \\ 0 & 0 & 1 & l_f\text{-}l_d\text{-}R \\ 0 & 0 & 0 & 1 \end{pmatrix} $$

$$ {}^{H_4}\Phi_{S_4} = \begin{pmatrix} \cos{}^{H_4}\theta_{S_4z} & -\sin{}^{H_4}\theta_{S_4z} & 0 & 0 \\ \sin{}^{H_4}\theta_{S_4z} & \cos{}^{H_4}\theta_{S_4z} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad {}^{S_4}T_{C_4} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -l_g \\ 0 & 0 & 1 & l_f\text{-}l_d\text{-}R \\ 0 & 0 & 0 & 1 \end{pmatrix}. $$

7.2.4 Wheel Equations-of-Motion

The Jacobian matrix template for non-steered conventional wheels in (A3.2) allows us to write the velocity equation-of-motion for wheels one and two:

$$ \dot{\mathbf{p}} = \begin{pmatrix} 0 & l_b \\ -R & -l_a \\ 0 & 1 \end{pmatrix} \begin{pmatrix} {}^{A_1}\omega_{W_1x} \\ {}^{\bar{C}_1}\omega_{C_1z} \end{pmatrix} = \mathbf{J}_1 \dot{\mathbf{q}}_1 \qquad (7.2.1) $$

$$\dot{\mathbf{p}} = \begin{pmatrix} 0 & l_b \\ -R & l_a \\ 0 & 1 \end{pmatrix} \begin{pmatrix} {}^{A_2}\omega_{W_{2x}} \\ \bar{C}_2 \omega_{C_{2z}} \end{pmatrix} = \mathbf{J}_2 \dot{\mathbf{q}}_2 \tag{7.2.2}$$

The Jacobian matrix template for steered conventional wheels in (A3.4) allows us to write the velocity equation-of-motion for wheels three and four:

$$\dot{\mathbf{p}} = \begin{pmatrix} r \sin^{H_3}\theta_{S_{3z}} & -l_g \cos^{H_3}\theta_{S_{3z}} - l_c & l_c \\ -r \cos^{H_3}\theta_{S_{3z}} & -l_g \sin^{H_3}\theta_{S_{3z}} + l_e & -l_e \\ 0 & 1 & -1 \end{pmatrix} \begin{pmatrix} {}^{A_3}\omega_{W_{3x}} \\ \bar{C}_3 \omega_{C_{3z}} \\ {}^{H_3}\omega_{S_{3z}} \end{pmatrix} = \mathbf{J}_3 \dot{\mathbf{q}}_3 \tag{7.2.3}$$

$$\dot{\mathbf{p}} = \begin{pmatrix} r \sin^{H_4}\theta_{S_{4z}} & -l_g \cos^{H_4}\theta_{S_{4z}} - l_c & l_c \\ -r \cos^{H_4}\theta_{S_{4z}} & -l_g \sin^{H_4}\theta_{S_{4z}} - l_e & l_e \\ 0 & 1 & -1 \end{pmatrix} \begin{pmatrix} {}^{A_4}\omega_{W_{4x}} \\ \bar{C}_4 \omega_{C_{4z}} \\ {}^{H_4}\omega_{S_{4z}} \end{pmatrix} = \mathbf{J}_4 \dot{\mathbf{q}}_4 \tag{7.2.4}$$

The WMR body velocity three-vector is $\dot{\mathbf{p}} = \begin{bmatrix} {}^{B}v_{Bx} & {}^{B}v_{By} & {}^{B}\omega_{Bz} \end{bmatrix}^T$.

We apply the wheel Jacobians $\mathbf{J}_i$ for i=1,...,4 in (7.2.1)-(7.2.4) and the wheel acceleration equation in (2.3.30) to compute the wheel acceleration equations-of-motion:

$$\ddot{\mathbf{p}} = \mathbf{J}_1 \ddot{\mathbf{q}}_1 + \begin{bmatrix} l_a & l_b & 0 \end{bmatrix}^T \bar{C}_1 \omega_{C_{1z}}^2 \tag{7.2.5}$$

$$\ddot{\mathbf{p}} = \mathbf{J}_2 \ddot{\mathbf{q}}_2 + \begin{bmatrix} -l_a & l_b & 0 \end{bmatrix}^T \bar{C}_2 \omega_{C_{2z}}^2 \tag{7.2.6}$$

$$\ddot{\mathbf{p}} = \mathbf{J}_3 \ddot{\mathbf{q}}_3 + \begin{pmatrix} l_g \sin^{H_3}\theta_{S_{3z}} - l_e & -l_e & -l_e \\ -l_g \cos^{H_3}\theta_{S_{3z}} - l_c & -l_c & -l_c \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \bar{C}_3 \omega_{C_3}^2 \\ -2\, \bar{C}_3 \omega_{C_3}\, {}^{H_3}\omega_{S_3} \\ {}^{H_3}\omega_{S_3}^2 \end{pmatrix} \tag{7.2.7}$$

$$\ddot{\mathbf{p}} = \mathbf{J}_4 \ddot{\mathbf{q}}_4 + \begin{pmatrix} {}^{H_4}l_g \sin\theta_{S_4 z} + l_e & l_e & l_e \\ -{}^{H_4}l_g \cos\theta_{S_4 z} - l_c & -l_c & -l_c \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \bar{c}_4 \omega_{C_4}^2 \\ -2\,\bar{c}_4 \omega_{C_4}\,{}^{H_4}\omega_{S_4} \\ {}^{H_4}\omega_{S_4}^2 \end{pmatrix} \qquad (7.2.8)$$

The WMR body acceleration three-vector is $\ddot{\mathbf{p}} = \left[{}^{\bar{B}}a_{Bx} \quad {}^{\bar{B}}a_{By} \quad {}^{\bar{B}}\alpha_{Bz} \right]^T$.

7.2.5 Mobility Characteristics

The soluble motion and the three DOF motion criteria in (A7.1) and (A7.2) are not satisfied by Bicsun-Bicas because non-steered conventional wheels one and two are degenerate. Wheels three and four allow three DOFs and thus impose no kinematic motion constraints. The kinematic motion constraints specified in (A7.3) imposed by wheels one and two are identical:

$$ {}^{\bar{B}}v_{Bx} = l_b \, {}^{\bar{B}}\omega_{Bz} \qquad (7.2.9)$$

The x-translational WMR velocity ${}^{\bar{B}}v_{Bx}$ and the z-rotational WMR velocity ${}^{\bar{B}}\omega_{Bz}$ are thus linearly dependent upon each other but independent of the y-translational velocity ${}^{\bar{B}}v_{By}$. Bicsun-Bicas therefore has two DOFs: y-translation and either x-translation or z-rotation. The adequate actuation and adequate sensing criteria in (A7.8) and (A7.14) are satisfied indicating that we may control and discern both WMR DOFs. The robust actuation criterion in (A7.10) is satisfied indicating that no actuator conflict may occur, and the robust sensing criterion in (A7.15) is not satisfied indicating that translational wheel slip cannot be detected by the method of Appendix A10.

7.2.6 Kinematic Solutions

The complete inverse velocity solution is required for dynamic modeling in Section 7.3. We compute the complete inverse velocity solution by assuming that all of the wheel velocities are actuated in the actuated inverse velocity solution in (2.4.8):

$$\begin{pmatrix} {}^{A_1}\omega_{W_{1x}} \\ \bar{C}_1\omega_{C_{1z}} \end{pmatrix} = \frac{1}{R(l_b^2+1)} \begin{pmatrix} -l_a l_b & -l_b^2-1 & -l_a \\ Rl_b & 0 & R \end{pmatrix} \begin{pmatrix} \bar{B}v_{Bx} \\ \bar{B}v_{By} \\ \bar{B}\omega_{Bz} \end{pmatrix} = (J_1^T J_1)^{-1} J_1^T \dot{p} \qquad (7.2.10)$$

$$\begin{pmatrix} {}^{A_2}\omega_{W_{2x}} \\ \bar{C}_2\omega_{C_{2z}} \end{pmatrix} = \frac{1}{R(l_b^2+1)} \begin{pmatrix} l_a l_b & -l_b^2-1 & l_a \\ Rl_b & 0 & R \end{pmatrix} \begin{pmatrix} \bar{B}v_{Bx} \\ \bar{B}v_{By} \\ \bar{B}\omega_{Bz} \end{pmatrix} = (J_2^T J_2)^{-1} J_2^T \dot{p} \qquad (7.2.11)$$

$$\begin{pmatrix} {}^{A_3}\omega_{W_{3x}} \\ {}^{P_3}\omega_{R_{3x}} \\ \bar{C}_3\omega_{C_{3z}} \end{pmatrix} = \frac{1}{rl_g} \begin{pmatrix} l_g \sin^{H_3}\theta_{S_{3z}} & -l_g \cos^{H_3}\theta_{S_{3z}} & l_c l_g \sin^{H_3}\theta_{S_{3z}} + l_e l_g \cos^{H_3}\theta_{S_{3z}} \\ -r\cos^{H_3}\theta_{S_{3z}} & -r\sin^{H_3}\theta_{S_{3z}} & l_e r\sin^{H_3}\theta_{S_{3z}} - l_c r\cos^{H_3}\theta_{S_{3z}} \\ -r\cos^{H_3}\theta_{S_{3z}} & -r\sin^{H_3}\theta_{S_{3z}} & -l_g r + l_e r\sin^{H_3}\theta_{S_{3z}} - l_c r\cos^{H_3}\theta_{S_{3z}} \end{pmatrix} \dot{p} = J_3^{-1} \dot{p}$$

(7.2.12)

$$\begin{pmatrix} {}^{A_4}\omega_{W_{4x}} \\ {}^{P_4}\omega_{R_{4x}} \\ \bar{C}_4\omega_{C_{4z}} \end{pmatrix} = \frac{1}{rl_g} \begin{pmatrix} l_g \sin^{H_4}\theta_{S_{4z}} & -l_g \cos^{H_4}\theta_{S_{4z}} & l_c l_g \sin^{H_4}\theta_{S_{4z}} - l_e l_g \cos^{H_4}\theta_{S_{4z}} \\ -r\cos^{H_4}\theta_{S_{4z}} & -r\sin^{H_4}\theta_{S_{4z}} & -l_e r\sin^{H_4}\theta_{S_{4z}} - l_c r\cos^{H_4}\theta_{S_{4z}} \\ -r\cos^{H_4}\theta_{S_{4z}} & -r\sin^{H_4}\theta_{S_{4z}} & -l_g r - l_e r\sin^{H_4}\theta_{S_{4z}} - l_c r\cos^{H_4}\theta_{S_{4z}} \end{pmatrix} \dot{p} = J_4^{-1} \dot{p}$$

(7.2.13)

Since we are minimizing the sum of linear and angular velocities, we have applied the *weighted* least-squares solution with the diagonal weighting matrix **W=diag(1, 1, 1m^2)** to maintain consistent units in (7.2.10) and (7.2.11). Consequently, the 1 in the factor (l_b^2+1) has units of square meters. Because Jacobians three and four are nonsingular, the least squares inverse velocity solutions for wheels three and four reduce to the simplified form shown in (7.2.12) and (7.2.13). The Uranus wheel Jacobians in Chapter 6 are also nonsingular and the simplified form of the inverse velocity solution applies; thus the weighted least-squares solution was not required.

The actuated inverse velocity solution is obtained from the complete inverse velocity solution in (7.2.10)-(7.2.13) by extracting only the actuated

wheel velocities. The actuated inverse velocity solution, required for servo-controller design in Section 7.4, is:

$$\begin{pmatrix} {}^{A_1}\omega_{W_1x} \\ {}^{A_2}\omega_{W_2x} \end{pmatrix} = \frac{1}{R(l_b^2+1)} \begin{pmatrix} -l_a l_b & -l_b^2-1 & -l_a \\ l_a l_b & -l_b^2-1 & l_a \end{pmatrix} \begin{pmatrix} {}^{\bar{B}}v_{Bx} \\ {}^{\bar{B}}v_{By} \\ {}^{\bar{B}}\omega_{Bz} \end{pmatrix} \tag{7.2.14}$$

The complete inverse acceleration solution is required for dynamic modeling in Section 7.3. We compute the complete inverse acceleration solution by solving each of the wheel acceleration equations-of-motion in (7.2.5)-(7.2.8) independently, applying the inverse wheel Jacobian matrices in (7.2.10)-(7.2.13). Since the masses and inertias of the castors are negligible, we do not model the the inertial forces/torques on wheels three and four. We thus compute the complete inverse acceleration solution for only wheels one and two.

$$\ddot{\mathbf{q}}_1 = (\mathbf{J}_1^T \mathbf{J}_1)^{-1} \mathbf{J}_1^T \{\ddot{\mathbf{p}} - [l_a \;\; l_b \;\; 0]^T \bar{C}_1 \omega_{C_1z}^2\} \tag{7.2.15}$$

$$\ddot{\mathbf{q}}_2 = (\mathbf{J}_2^T \mathbf{J}_2)^{-1} \mathbf{J}_2^T \{\ddot{\mathbf{p}} - [-l_a \;\; l_b \;\; 0]^T \bar{C}_2 \omega_{C_2z}^2\} \tag{7.2.16}$$

The wheel acceleration vector is $\ddot{\mathbf{q}}_j = [{}^{A_j}\alpha_{W_jx} \quad \bar{C}_j\omega_{C_jz}]^T$ for j=1 and 2.

We apply the least-squares sensed forward velocity solution in (2.4.14) to obtain:

$$\begin{pmatrix} {}^{\bar{B}}v_{Bx} \\ {}^{\bar{B}}v_{By} \\ {}^{\bar{B}}\omega_{Bz} \end{pmatrix} = \frac{R}{2l_a} \begin{pmatrix} -l_b & l_b \\ -l_a & -l_a \\ -1 & 1 \end{pmatrix} \begin{pmatrix} {}^{A_1}\omega_{W_1x} \\ {}^{A_2}\omega_{W_2x} \end{pmatrix} \tag{7.2.17}$$

The sensed forward velocity solution is required in Section 7.4 for the dead reckoning computations within the feedback path of model-based WMR servo-controllers.

7.2.7 Translational Wheel Slip Detection

Since the sensing structure of Bicsun-Bicas is adequate but not robust, the kinematic model is always consistent even in the presence of translational

wheel slip. Although we cannot detect translational wheel slip by the method of Appendix 10, we may design Bicsun-Bicas to have a robust sensing structure by installing sensors on the castors.

7.2.8 Dead Reckoning

We apply the dead reckoning update algorithm in (A9.4) to Bicsun-Bicas. Upon substituting the sensed forward velocity solution in (7.2.7) into (A9.4), expanding, and simplifying, we obtain algorithms for the direct computation of the current orientation ${}^{F}\theta_{Bz}(nT)$, and the recursive computation of the current translational positions ${}^{F}d_{Bx}(nT)$ and ${}^{F}d_{By}(nT)$ of Bicsun-Bicas:

$$ {}^{F}\theta_{Bz}(nT) = \frac{R}{2l_a}\theta(nT) + {}^{F}\theta_{Bz}(0) - \frac{R}{2l_a}\theta(0) \qquad (7.2.18) $$

and

$$ \begin{pmatrix} {}^{F}d_{Bx}(nT) \\ {}^{F}d_{By}(nT) \end{pmatrix} = \begin{pmatrix} {}^{F}d_{Bx}[(n-1)T] \\ {}^{F}d_{By}[(n-1)T] \end{pmatrix} \qquad (7.2.19) $$

$$ + \frac{TR}{4l_a} \begin{pmatrix} -l_b\cos{}^{F}\theta_{Bz}+l_a\sin{}^{F}\theta_{Bz} & l_b\cos{}^{F}\theta_{Bz}+l_a\sin{}^{F}\theta_{Bz} \\ -l_b\sin{}^{F}\theta_{Bz}-l_a\cos{}^{F}\theta_{Bz} & l_b\sin{}^{F}\theta_{Bz}-l_a\cos{}^{F}\theta_{Bz} \end{pmatrix} \left\{ \dot{\mathbf{q}}_s[(n-1)T] + \dot{\mathbf{q}}_s(nT) \right\} $$

where $\theta(\bullet)=\left\{ {}^{A_2}\theta_{W_{2x}}(\bullet) - {}^{A_1}\theta_{W_{1x}}(\bullet) \right\}$ and $\dot{\mathbf{q}}_s(t)=\begin{pmatrix} {}^{A_1}\omega_{W_{1x}}(t) & {}^{A_2}\omega_{W_{2x}}(t) \end{pmatrix}$.

The direct computation of the WMR orientation indicates that (7.2.18) is a holonomic constraint [Landau76]. Since the WMR translations cannot be computed directly, (7.2.19) are non-holonomic constraints. Errors in the recursively computed WMR translations due to finite precision and sensor noise will accumulate; whereas the direct computation of the holonomic orientation constraint in (7.2.18) will not accumulate these errors.

7.2.9 Concluding Remarks

The kinematics of Bicsun-Bicas differ from the kinematics of Uranus because of the two *degenerate* non-steered conventional wheels. Non-steered

conventional wheels have two wheel velocities which are linearly related to the three WMR body velocities by the (3x2) wheel Jacobian in (A3.2). Because there is not always a solution for the two wheel variables from the three WMR body velocities, wheels one and two are degenerate. The degenerate wheels constrain the possible motions of Bicsun-Bicas according to the motion constraint in (7.2.9). A WMR has a maximum of three DOFs. Because of the scalar kinematic constraint in (7.2.9), Bicsun-Bicas has two DOFs. Only two WMR body velocities may be chosen independently to specify its motion.

7.3 Dynamic Modeling

7.3.1 Overview

We model the dynamics of Bicsun-Bicas to illustrate the application of our dynamic modeling procedure formulated in Chapter 3. As in Chapter 6, we assume: negligible castor masses and inertias; no wheel translational slip; and negligible viscous friction on the load, wheels and casters. We make correspondences between the components of Bicsun-Bicas and the simple closed-chain in Figure 1.1. We identify the rectangular WMR body as the main link, the floor as the environmental body, the conventional non-steered wheels one and two as open-chains each consisting of one link, and the conventional steered wheels three and four as open-chains each consisting of two links: the steering link as the proximal link and the wheel as the distal link of each open-chain. The load is an additional single link open-chain which does not contact the floor. We sketch these components in the plane in Figures 7.1 and 7.2 and in three-dimensions in Figure 7.3.

We begin in Section 7.3.2 by assigning coordinate systems, in addition to the coordinate systems assigned for kinematic modeling in Section 7.2, for the dynamic modeling of Bicsun-Bicas. In Section 7.3.3, we formulate the components required for dynamic modeling: position six-vectors, force/torque source six-vectors, companion (6x6) kinematic matrices, (6x6) joint coupling matrices, dry frictional coefficients, and the (6x6) propagation matrices. From these components in Section 7.3.4, we formulate the primary, internal and secondary force/torque equations-of-motion. In Section 7.3.5, we compute the six-vector velocity and acceleration transformations which we then combine

with the force/torque equations-of-motion to compute in Section 7.3.6 the Bicsun-Bicas dynamic model. We solve in Section 7.3.7 the dynamic model to compute the actuated inverse and forward dynamic solutions. We then advance our concluding remarks in Section 7.3.8 and compare the dynamic models of Bicsun-Bicas and Uranus.

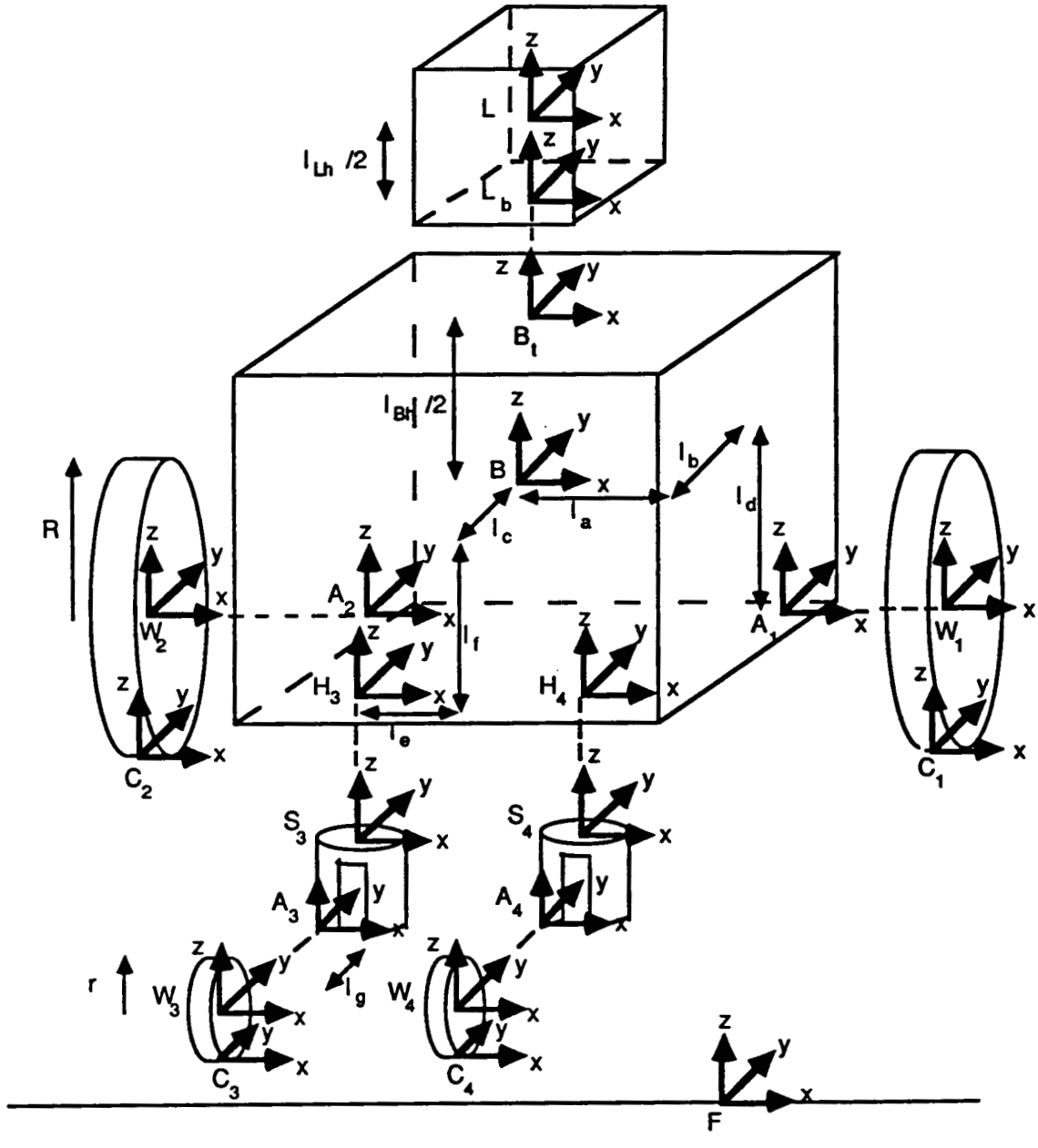


Figure 7.3: Bicsun-Bicas with all Coordinate Systems Assigned

7.3.2 Coordinate System Assignments

We assign and draw the body B, axle A_i, wheel W_i, hip H_k, steering S_k, contact-point C_i (for i=1,...,4 and k=3 and 4), and floor F coordinate systems in Figures 7.1-7.3. The *four* gravitational coordinate systems, G(*B*), G(*L*), and G(W_j) for j=1 and 2, located at the origins of the B, L, and W_j coordinate systems and aligned with the gravitational field, are not explicitly assigned.

For the dynamic modeling of Bicsun-Bicas, we formulate the required matrices and vectors and the matrix-vector equations as prescribed in Chapter 3, and then apply MACSYMA to compute and simplify the complex matrix-vector expressions.

7.3.3 Dynamic Model Components

The *twenty-four* position six-vectors of Bicsun-Bicas are:

$${}^{F}p_{B} = \left({}^{F}d_{Bx} \;\; {}^{F}d_{By} \;\; l_d+R \;\; 0 \;\; 0 \;\; {}^{F}\theta_{Bz} \right)^T \qquad {}^{B}p_{B_t} = \left(0 \;\; 0 \;\; l_{Bh}/2 \;\; 0 \;\; 0 \;\; 0 \right)^T$$

$${}^{B_t}p_{L_b} = \left(0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \right)^T \qquad {}^{L_b}p_{L} = \left(0 \;\; 0 \;\; l_{Lh}/2 \;\; 0 \;\; 0 \;\; 0 \right)^T$$

$${}^{B}p_{A_1} = \left(l_a \;\; l_b \;\; -l_d \;\; 0 \;\; 0 \;\; 0 \right)^T \qquad {}^{B}p_{A_2} = \left(-l_a \;\; l_b \;\; -l_d \;\; 0 \;\; 0 \;\; 0 \right)^T$$

$${}^{A_i}p_{W_i} = \left(0 \;\; 0 \;\; 0 \;\; {}^{A_i}\theta_{W_ix} \;\; 0 \;\; 0 \right)^T \qquad {}^{W_i}p_{Q_i} = \left(0 \;\; 0 \;\; 0 \;\; -{}^{A_i}\theta_{W_ix} \;\; 0 \;\; 0 \right)^T$$

$${}^{Q_j}p_{C_j} = \left(0 \;\; 0 \;\; -R \;\; 0 \;\; 0 \;\; 0 \right)^T \qquad {}^{Q_k}p_{C_k} = \left(0 \;\; 0 \;\; -r \;\; 0 \;\; 0 \;\; 0 \right)^T$$

$${}^{B}p_{H_3} = \left(-l_e \;\; -l_c \;\; -l_f \;\; 0 \;\; 0 \;\; 0 \right)^T \qquad {}^{B}p_{H_4} = \left(l_e \;\; -l_c \;\; -l_f \;\; 0 \;\; 0 \;\; 0 \right)^T$$

$${}^{S_k}p_{A_k} = \left(0 \;\; -l_g \;\; (-l_d-R+l_f+r) \;\; 0 \;\; 0 \;\; 0 \right)^T \qquad {}^{H_k}p_{S_k} = \left(0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; {}^{H_k}\theta_{S_kz} \right)^T$$

for i=1,...,4, j=1 and 2, and k=3 and 4. The *intermediate* coordinate systems Q_i for i=1,...,4 are introduced to ease the determination of the position vectors (with roll-pitch-yaw rotations) by inspection of Figures 7.1-7.3. Coordinate system Q_i is coincident with the A_i coordinate system and fixed with wheel W_i. The position and orientation of the WMR body relative to the reference floor coordinate

system are ${}^{F}d_{Bx}$, ${}^{F}d_{By}$, and ${}^{F}\theta_{Bz}$, the wheel angles are ${}^{A_i}\theta_{W_ix}$ and the steering angles are ${}^{H_k}\theta_{S_kx}$ for i=1,...,4 and k=3 and 4.

The *seventeen* force/torque six-vectors for Bicsun-Bicas are:

$${}^{B}\mathbf{f}_{iB} = \begin{pmatrix} -m_B {}^{\bar{B}}a_{Bx} \\ -m_B {}^{\bar{B}}a_{By} \\ 0 \\ 0 \\ 0 \\ -I_{Bz} {}^{\bar{B}}\alpha_{Bz} \end{pmatrix} \qquad {}^{L}\mathbf{f}_{iL} = \begin{pmatrix} -m_L {}^{\bar{L}}a_{Lx} \\ -m_L {}^{\bar{L}}a_{Ly} \\ 0 \\ 0 \\ 0 \\ -I_{Lz} {}^{\bar{L}}\alpha_{Lz} \end{pmatrix} \qquad {}^{B}\mathbf{f}_{vB} = \begin{pmatrix} -\mu_{vBx} {}^{\bar{B}}v_{Bx} \\ -\mu_{vBy} {}^{\bar{B}}v_{By} \\ 0 \\ 0 \\ 0 \\ -\varepsilon_{vBz} {}^{\bar{B}}\omega_{Bz} \end{pmatrix}$$

$${}^{W_i}\mathbf{f}_{iW_j} = \begin{pmatrix} -m_W {}^{\bar{W}_j}a_{W_jx} \\ -m_W {}^{\bar{W}_j}a_{W_jy} \\ -m_W {}^{\bar{W}_j}a_{W_jz} \\ -I_{Wx} {}^{\bar{W}_j}\alpha_{W_jx} \\ -I_{Wy} {}^{\bar{W}_j}\alpha_{W_jy} - (I_{Wx} - I_{Wy}) {}^{\bar{W}_j}\omega_{W_jx} {}^{\bar{W}_j}\omega_{W_jz} \\ -I_{Wy} {}^{\bar{W}_j}\alpha_{W_jz} - (I_{Wy} - I_{Wx}) {}^{\bar{W}_j}\omega_{W_jx} {}^{\bar{W}_j}\omega_{W_jy} \end{pmatrix} \qquad {}^{L}\mathbf{f}_{gL} = \begin{pmatrix} 0 \\ 0 \\ -m_L g \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$${}^{B}\mathbf{f}_{gB} = \begin{pmatrix} 0 & 0 & -m_B g & 0 & 0 & 0 \end{pmatrix}^T \qquad {}^{G(W_j)}\mathbf{f}_{gW_j} = \begin{pmatrix} 0 & 0 & -m_W g & 0 & 0 & 0 \end{pmatrix}^T$$

$${}^{C_i}\mathbf{f}_{eW_i} = \begin{pmatrix} f_{xi} & f_{yi} & f_{zi} & \tau_{xi} & \tau_{yi} & \tau_{zi} \end{pmatrix}^T \quad \text{and} \quad {}^{W_j}\mathbf{f}_{aW_j} = -{}^{A_j}\mathbf{f}_{aB} = \begin{pmatrix} 0 & 0 & 0 & \beta\tau_j & 0 & 0 \end{pmatrix}^T$$

for i=1,...,4 and j=1 and 2. The inertial forces/torques are formulated according to (3.3.3), the gravitational forces/torques according to (3.3.4), the actuation forces/torques according to (3.3.5), the viscous friction forces/torques according to (3.3.6), and the environmental forces/torques according to (3.3.8). Since each wheel is symmetric about its axle, $I_{Wz} = I_{Wy}$ in the inertial force/torque six-vector ${}^{W_j}\mathbf{f}_{iW_j}$.

The companion kinematic matrices are required for the force/torque equations-of-motion in Sections 7.3.4 and the kinematic transformations in Section 7.3.5. Each (6×6) kinematic matrix ${}^{A}\mathbf{X}_{B}$, for **X=L,J,R,S**, and **T**, is formulated by substituting the elements of the position six-vector ${}^{B}\mathbf{p}_{A}$ into the corresponding kinematic matrix template in Appendix 12. The *forty-two* Bicsun-Bicas companion kinematic matrices are:

$$ {}^{B_t}\mathbf{L}_{B} = \begin{pmatrix} 1 & 0 & 0 & 0 & l_{Bh}/2 & 0 \\ 0 & 1 & 0 & -l_{Bh}/2 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \qquad {}^{L}\mathbf{L}_{L_b} = \begin{pmatrix} 1 & 0 & 0 & 0 & l_{Lh}/2 & 0 \\ 0 & 1 & 0 & -l_{Lh}/2 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} $$

$$ {}^{L_b}\mathbf{L}_{B_t} = {}^{L_b}\mathbf{J}_{B_t} = \mathbf{I} $$

$$ {}^{B}\mathbf{L}_{F} = \begin{pmatrix} \cos\theta_z & \sin\theta_z & 0 & -(R+l_d)\sin\theta_z & (R+l_d)\cos\theta_z & {}^{F}d_{Bx}\sin\theta_z - {}^{F}d_{By}\cos\theta_z \\ -\sin\theta_z & \cos\theta_z & 0 & -(R+l_d)\cos\theta_z & -(R+l_d)\sin\theta_z & {}^{F}d_{By}\sin\theta_z + {}^{F}d_{Bx}\cos\theta_z \\ 0 & 0 & 1 & {}^{F}d_{By} & -{}^{F}d_{Bx} & 0 \\ 0 & 0 & 0 & \cos\theta_z & \sin\theta_z & 0 \\ 0 & 0 & 0 & -\sin\theta_z & \cos\theta_z & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} $$

where $\theta_z = {}^{F}\theta_{Bz}$.

$$ {}^{A_1}\mathbf{L}_{B} = \begin{pmatrix} 1 & 0 & 0 & 0 & -l_d & -l_b \\ 0 & 1 & 0 & l_d & 0 & l_a \\ 0 & 0 & 1 & l_b & -l_a & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \qquad {}^{A_2}\mathbf{L}_{B} = \begin{pmatrix} 1 & 0 & 0 & 0 & -l_d & -l_b \\ 0 & 1 & 0 & l_d & 0 & -l_a \\ 0 & 0 & 1 & l_b & l_a & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} $$

$$
{}^{W_i}L_{A_i} = {}^{W_i}J_{A_i} = \begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 \\
0 & \cos^{A_i}\theta_{W_ix} & \sin^{A_i}\theta_{W_ix} & 0 & 0 & 0 \\
0 & -\sin^{A_i}\theta_{W_ix} & \cos^{A_i}\theta_{W_ix} & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & \cos^{A_i}\theta_{W_ix} & \sin^{A_i}\theta_{W_ix} \\
0 & 0 & 0 & 0 & -\sin^{A_i}\theta_{W_ix} & \cos^{A_i}\theta_{W_ix}
\end{pmatrix}
$$

$$
{}^{Q_i}L_{W_i} = \begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 \\
0 & \cos^{A_i}\theta_{W_ix} & -\sin^{A_i}\theta_{W_ix} & 0 & 0 & 0 \\
0 & \sin^{A_i}\theta_{W_ix} & \cos^{A_i}\theta_{W_ix} & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & \cos^{A_i}\theta_{W_ix} & -\sin^{A_i}\theta_{W_ix} \\
0 & 0 & 0 & 0 & \sin^{A_i}\theta_{W_ix} & \cos^{A_i}\theta_{W_ix}
\end{pmatrix}
$$

$$
{}^{C_j}L_{Q_j} = \begin{pmatrix}
1 & 0 & 0 & 0 & -R & 0 \\
0 & 1 & 0 & R & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix}
\qquad
{}^{C_k}L_{Q_k} = \begin{pmatrix}
1 & 0 & 0 & 0 & -r & 0 \\
0 & 1 & 0 & r & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix}
$$

$$
{}^{H_3}L_B = \begin{pmatrix}
1 & 0 & 0 & 0 & -l_f & l_c \\
0 & 1 & 0 & l_f & 0 & -l_e \\
0 & 0 & 1 & -l_c & l_e & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix}
\qquad
{}^{H_4}L_B = \begin{pmatrix}
1 & 0 & 0 & 0 & -l_f & l_c \\
0 & 1 & 0 & l_f & 0 & l_e \\
0 & 0 & 1 & -l_c & -l_e & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix}
$$

$$
{}^{A_k}L_{S_k} = \begin{pmatrix}
1 & 0 & 0 & 0 & -R+r+l_f-l_d & l_g \\
0 & 1 & 0 & R-r-l_f+l_d & 0 & 0 \\
0 & 0 & 1 & -l_g & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix}
$$

$$ {}^{S_k}L_{H_k} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \cos^{H_k}\theta_{S_kz} & \sin^{H_k}\theta_{S_kz} & 0 & 0 & 0 \\ 0 & -\sin^{H_k}\theta_{S_kz} & \cos^{H_k}\theta_{S_kz} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \cos^{H_k}\theta_{S_kz} & \sin^{H_k}\theta_{S_kz} \\ 0 & 0 & 0 & 0 & -\sin^{H_k}\theta_{S_kz} & \cos^{H_k}\theta_{S_kz} \end{pmatrix} $$

$$ {}^{W_j}R_{A_j} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 \\ -\sin^{A_j}\theta_{W_jx} & \cos^{A_j}\theta_{W_jx} & \sin^{A_j}\theta_{W_jx} & 0 & -\cos^{A_j}\theta_{W_jx} & 0 \\ -\cos^{A_j}\theta_{W_jx} & -\sin^{A_j}\theta_{W_jx} & \cos^{A_j}\theta_{W_jx} & 0 & \sin^{A_j}\theta_{W_jx} & 0 \end{pmatrix} $$

$$ {}^{L_b}R_{B_t} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 \\ 0 & 1 & 0 & 0 & -1 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} $$

$$ {}^{B_t}S_B = \begin{pmatrix} 0 & 0 & 0 & 0 & l_{Bh}/2 & 0 \\ 0 & 0 & 0 & 0 & 0 & l_{Bh}/2 \\ -l_{Bh}/2 & -l_{Bh}/2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} $$

$$ {}^{A_1}S_B = \begin{pmatrix} 0 & -l_a & -l_a & -l_b & -l_d & 0 \\ -l_b & 0 & -l_b & l_a & 0 & -l_d \\ l_d & l_d & 0 & 0 & l_a & l_b \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} $$

$$ {}^{A_2}S_B = \begin{pmatrix} 0 & l_a & l_a & -l_b & -l_d & 0 \\ -l_b & 0 & -l_b & -l_a & 0 & -l_d \\ l_d & l_d & 0 & 0 & -l_a & l_b \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} $$

$$ {}^{L}S_{L_b} = \begin{pmatrix} 0 & 0 & 0 & 0 & l_{Lh}/2 & 0 \\ 0 & 0 & 0 & 0 & 0 & l_{Lh}/2 \\ -l_{Lh}/2 & -l_{Lh}/2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} $$

$$ {}^{L_b}T_{B_t} = \begin{pmatrix} 0 & 0 & 0 & 2 & 0 & -2 \\ 0 & -2 & 0 & 0 & 2 & 0 \\ 2 & 0 & -2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} $$

$${}^{L_b}S_{B_t} = {}^{W_i}S_{A_j} = \mathbf{0}$$

$${}^{W_j}T_{A_j} = \begin{pmatrix} 0 & 0 & 0 & 2 & 0 & -2 \\ 2\sin^{A_j}\theta_{W_jx} & -2\cos^{A_j}\theta_{W_jx} & -2\sin^{A_j}\theta_{W_jx} & 0 & 2\cos^{A_j}\theta_{W_jx} & 0 \\ 2\cos^{A_j}\theta_{W_jx} & 2\sin^{A_j}\theta_{W_jx} & -2\cos^{A_j}\theta_{W_jx} & 0 & -2\sin^{A_j}\theta_{W_jx} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

for i=1,...,4; j=1 and 2; and k=3 and 4.

In the absence of translational wheel slippage, the *eleven* coupling matrices are:

$${}^{C_i}C_{W_iF} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & \zeta_{ri} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \zeta_{cci} & 0 & 0 & 0 \end{pmatrix} \qquad {}^{A_j}C_{BW_j} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & \zeta_{cwi} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$${}^{A_k}C_{S_kW_k} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & \zeta_{cwj} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \qquad {}^{H_k}C_{BS_k} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & \zeta_{csk} & 0 & 0 & 0 \end{pmatrix}$$

for i=1,...,4; j=1 and 2; and k=3 and 4. The coupling matrix ${}^{B_t}C_{BL}$ between the WMR body and the load is the identity matrix because we assume that the load cannot slide.

The Coulomb frictional coefficients are formulated according to (3.3.20) and the rolling frictional coefficients according to (3.3.21). The rolling (r) friction constant is μ_r, and the Coulomb (c) frictional constants are μ_{cc} for the wheel contact (c) friction, μ_{cw} for the wheel (w) bearing friction, and μ_{cs} for the steering (s) bearing friction. We assume no stiction to avoid computational complexities

when solving the dynamic model. For simplicity, we assume that all Coulomb frictional constants are equal (i.e., $\mu_{cc}=\mu_{cw}=\mu_{cs}=\mu_c$). The *fourteen* frictional coefficients required for the coupling matrices in Section 7.3.5 are:

$$\zeta_{ri} = \begin{cases} -\mu_r \text{sgn}({}^{A_i}\omega_{W_{ix}}) & \text{if } {}^{A_i}\omega_{W_{ix}} \neq 0 \\ 0 & \text{otherwise} \end{cases} \qquad \text{for } i=1,\ldots,4$$

$$\zeta_{cci} = \begin{cases} -\mu_{cc} \text{sgn}({}^{\bar{C}_i}\omega_{C_{iz}} {}^{C_i}f_{Fz}) & \text{if } {}^{\bar{C}_i}\omega_{C_{iz}} \neq 0 \\ 0 & \text{otherwise} \end{cases} \qquad \text{for } i=1,\ldots,4$$

$$\zeta_{cwi} = \begin{cases} -\mu_{cw} \text{sgn}({}^{A_i}\omega_{W_{ix}} {}^{A_i}f_{W_{ix}}) & \text{if } {}^{A_i}\omega_{W_{ix}} \neq 0 \\ 0 & \text{otherwise} \end{cases} \qquad \text{for } i=1,\ldots,4$$

$$\zeta_{csk} = \begin{cases} -\mu_{cs} \text{sgn}({}^{H_k}\omega_{S_{kx}} {}^{H_k}f_{S_{kz}}) & \text{if } {}^{H_k}\omega_{S_{kx}} \neq 0 \\ 0 & \text{otherwise} \end{cases} \qquad \text{for } k=3 \text{ and } 4$$

To compute the propagation matrices, we apply the link Jacobian cascade property in (3.3.15) to eliminate the intermediate coordinate systems Q_i for i=1,...,4 from our analysis:

$${}^{C_i}\mathbf{L}_{W_i} = {}^{C_i}\mathbf{L}_{Q_i} {}^{Q_i}\mathbf{L}_{W_i} \qquad \text{for } i=1,\ldots,4 \qquad (7.3.1)$$

The gravitational propagation matrices for wheels one and two are formulated according to (3.3.27) as:

$${}^{W_j}\mathbf{P}_{G(W_j)} = \left({}^{B}\mathbf{L}_F^T \, {}^{A_j}\mathbf{L}_B^T \, {}^{W_j}\mathbf{L}_{A_j}^T\right)^T * \mathbf{P}_{mask} \qquad \text{for } j=1 \text{ and } 2 \qquad (7.3.2)$$

The gravitational propagation mask $\mathbf{P}_{mask}$ is defined in (3.3.28). The remaining *seventeen* propagation matrices required for Bicsun-Bicas dynamic modeling are formulated by applying the propagation cascade in (3.3.26) with the link Jacobians and coupling matrices formulated in this section to obtain:

$${}^{W_i}\mathbf{P}_{C_i} = {}^{C_i}\mathbf{L}_{W_i}^T \qquad \text{for } i=1,\ldots,4 \qquad (7.3.3)$$

$${}^{S_k}\mathbf{P}_{A_k} = {}^{A_k}\mathbf{L}_{S_k}^T \qquad \text{for } k=3 \text{ and } 4 \qquad (7.3.4)$$

$${}^{B}\mathbf{P}_{L} = {}^{B_t}\mathbf{L}_B^T \, {}^{B_t}\mathbf{C}_{BL} \, {}^{L_b}\mathbf{L}_{B_t}^T \, {}^{L}\mathbf{L}_{L_b}^T \qquad (7.3.5)$$

$$ {}^{B}\mathbf{P}_{W_j} = {}^{A_j}\mathbf{L}_B^T \, {}^{A_j}\mathbf{C}_{BW_j} \, {}^{W_j}\mathbf{L}_{A_j}^T \qquad \text{for j=1 and 2} \qquad (7.3.6) $$

$$ {}^{B}\mathbf{P}_{A_j} = {}^{A_j}\mathbf{L}_B^T \qquad \text{for j=1 and 2} \qquad (7.3.7) $$

$$ {}^{B}\mathbf{P}_{C_j} = {}^{A_j}\mathbf{L}_B^T \, {}^{A_j}\mathbf{C}_{BW_j} \, {}^{W_j}\mathbf{L}_{A_j}^T \, {}^{C_j}\mathbf{L}_{W_j}^T \, {}^{C_j}\mathbf{C}_{W_j F} \qquad \text{for j=1 and 2} \qquad (7.3.8) $$

$$ {}^{B}\mathbf{P}_{C_k} = {}^{H_k}\mathbf{L}_B^T \, {}^{H_k}\mathbf{C}_{BS_k} \, {}^{S_k}\mathbf{L}_{H_k}^T \, {}^{A_k}\mathbf{L}_{S_k}^T \, {}^{A_k}\mathbf{C}_{S_k W_k} \, {}^{W_k}\mathbf{L}_{A_k}^T \, {}^{C_k}\mathbf{L}_{W_k}^T \, {}^{C_k}\mathbf{C}_{W_k F} $$

$$ \text{for k=3 and 4} \qquad (7.3.9) $$

$$ {}^{B}\mathbf{P}_{G(W_j)} = {}^{B}\mathbf{P}_{W_j} \, {}^{W_j}\mathbf{P}_{G(W_j)} \qquad \text{for j=1 and 2} \qquad (7.3.10) $$

7.3.4 Force/Torque Equations-of-Motion

We formulate the six primary force/torque equations-of-motion by propagating all of the force/torque sources in Section 7.3.3 with the propagation matrices in Section 7.3.3 to the WMR body coordinate system and then summing these propagated force/torque sources to zero according to (3.3.28):

$$ {}^{B}\mathbf{P}_{W_1} \, {}^{W_1}\mathbf{f}_{iW_1} + {}^{B}\mathbf{P}_{G(W_1)} \, {}^{G(W_1)}\mathbf{f}_{gW_1} + {}^{B}\mathbf{P}_{C_1} \, {}^{C_1}\mathbf{f}_{eW_1} + {}^{B}\mathbf{P}_{A_1} \, {}^{A_1}\mathbf{f}_{aB} + {}^{B}\mathbf{P}_{W_1} \, {}^{W_1}\mathbf{f}_{aW_1} $$

$$ + {}^{B}\mathbf{P}_{W_2} \, {}^{W_2}\mathbf{f}_{iW_2} + {}^{B}\mathbf{P}_{G(W_2)} \, {}^{G(W_2)}\mathbf{f}_{gW_2} + {}^{B}\mathbf{P}_{C_2} \, {}^{C_2}\mathbf{f}_{eW_2} + {}^{B}\mathbf{P}_{A_2} \, {}^{A_2}\mathbf{f}_{aB} + {}^{B}\mathbf{P}_{W_2} \, {}^{W_2}\mathbf{f}_{aW_2} $$

$$ + {}^{B}\mathbf{P}_{C_3} \, {}^{C_3}\mathbf{f}_{eW_3} + {}^{B}\mathbf{P}_{C_4} \, {}^{C_4}\mathbf{f}_{eW_4} + {}^{B}\mathbf{P}_{L} \, {}^{L}\mathbf{f}_{iL} + {}^{B}\mathbf{P}_{L} \, {}^{L}\mathbf{f}_{gL} + {}^{B}\mathbf{f}_{iB} + {}^{B}\mathbf{f}_{gB} + {}^{B}\mathbf{f}_{vB} = \mathbf{0} \qquad (7.3.11) $$

We apply MACSYMA to compute symbolically this set of six primary force/torque equations-of-motion.

The internal forces/torques at each of the joints are computed recursively according to (3.3.31) to obtain:

$$ {}^{C_i}\mathbf{f}_F = {}^{C_i}\mathbf{f}_{eW_i} \qquad \text{for i=1,...,4} \qquad (7.3.12) $$

$$ {}^{A_j}\mathbf{f}_{W_j} = {}^{W_j}\mathbf{L}_{A_j}^T \left({}^{W_j}\mathbf{P}_{C_j} \, {}^{C_j}\mathbf{f}_F + \left\{ {}^{W_j}\mathbf{f}_{iW_j} + {}^{W_j}\mathbf{P}_{G(W_j)} \, {}^{G(W_j)}\mathbf{f}_{gW_j} + {}^{W_j}\mathbf{f}_{aW_j} \right\} \right) $$

$$ \text{for j=1 and 2} \qquad (7.3.13) $$

$$ {}^{A_k}\mathbf{f}_{W_k} = {}^{W_k}\mathbf{L}_{A_k}^T \, {}^{W_k}\mathbf{P}_{C_k} \, {}^{C_k}\mathbf{f}_F \qquad \text{for k=3 and 4} \qquad (7.3.14) $$

$$ {}^{H_k}\mathbf{f}_{S_k} = {}^{S_k}\mathbf{L}_{H_k}^{T}\ {}^{S_k}\mathbf{P}_{A_k}\ {}^{A_k}\mathbf{f}_{W_k} \qquad \text{for k=3 and 4} \qquad (7.3.15) $$

We apply the joint coupling matrices in Section 7.3.3 and the internal forces/torques to formulate the secondary force/torque equations-of-motion according to (3.3.33) to obtain:

$$ (\mathbf{I} - {}^{C_i}\mathbf{C}_{FW_i}\ {}^{C_i}\mathbf{C}_{W_iF})\ {}^{C_i}\mathbf{f}_F = \mathbf{0} \qquad \text{for i=1,...,4} \qquad (7.3.16) $$

$$ (\mathbf{I} - {}^{A_j}\mathbf{C}_{W_jB}\ {}^{A_j}\mathbf{C}_{BW_j})\ {}^{A_j}\mathbf{f}_{W_j} = \mathbf{0} \qquad \text{for j=1 and 2} \qquad (7.3.17) $$

$$ (\mathbf{I} - {}^{A_k}\mathbf{C}_{W_kS_k}\ {}^{A_k}\mathbf{C}_{S_kW_k})\ {}^{A_k}\mathbf{f}_{W_k} = \mathbf{0} \qquad \text{for k=3 and 4} \qquad (7.3.18) $$

$$ (\mathbf{I} - {}^{H_k}\mathbf{C}_{S_kB}\ {}^{H_k}\mathbf{C}_{BS_k})\ {}^{H_k}\mathbf{f}_{S_k} = \mathbf{0} \qquad \text{for k=3 and 4} \qquad (7.3.19) $$

The required coupling matrices are computed from the coupling matrices in Section 7.3.3 by applying the coupling matrix mask $\mathbf{C}_{mask}$ in (3.3.25) according to (3.3.24):

$$ {}^{C_i}\mathbf{C}_{FW_i} = {}^{C_i}\mathbf{C}_{W_iF} * \mathbf{C}_{mask} \qquad \text{for i=1,...,4} \qquad (7.3.20) $$

$$ {}^{A_j}\mathbf{C}_{W_jB} = {}^{A_j}\mathbf{C}_{BW_j} * \mathbf{C}_{mask} \qquad \text{for j=1 and 2} \qquad (7.3.21) $$

$$ {}^{A_k}\mathbf{C}_{W_jS_k} = {}^{A_k}\mathbf{C}_{S_kW_k} * \mathbf{C}_{mask} \qquad \text{for k=3 and 4} \qquad (7.3.22) $$

$$ {}^{H_k}\mathbf{C}_{S_kB} = {}^{H_k}\mathbf{C}_{BS_k} * \mathbf{C}_{mask} \qquad \text{for k=3 and 4} \qquad (7.3.23) $$

Of the *sixty* scalar equations in (7.3.16)-(7.3.19), *forty-two* are trivial (i.e., 0=0) and *eighteen* are nontrivial: *three* at each of the contact-points (C_i for i=1,...,4) corresponding to the x, y and z-rotational joint DOF axes, *one* at each of the wheel joints (W_i for i=1,...,4) corresponding to the x-rotational joint DOF axes, and *one* at each of the steering joints (S_k for k=3 and 4) corresponding to the z-rotational joint DOF axes.

7.3.5 Companion Kinematic Transformations

We transform the velocities in the inertial and viscous frictional components of the force/torque equations-of-motion from the natural

instantaneously coincident coordinate systems to the WMR body and joint velocities. We first define the joint velocity six-vectors

$$ {}^{B_t}v_{L_b} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}^T \quad \text{and} \quad {}^{A_j}v_{W_j} = \begin{pmatrix} 0 & 0 & 0 & {}^{A_j}\omega_{W_jx} & 0 & 0 \end{pmatrix}^T $$

for j=1 and 2 and the WMR body velocity six-vector

$$ \bar{}^{B}v_B = \begin{pmatrix} \bar{}^{B}v_{Bx} & \bar{}^{B}v_{By} & 0 & 0 & 0 & \bar{}^{B}\omega_{Bz} \end{pmatrix}^T $$

Transformations of the castor velocities and the contact point velocities are not required because they are not utilized in the force/torque equations-of-motion. We compute recursively the velocities of each of the Bicsun-Bicas coordinate systems according to the velocity transformations in (3.4.8) and (3.4.9) to obtain:

$$ \bar{}^{B_t}v_{B_t} = {}^{B_t}L_B \, \bar{}^{B}v_B \tag{7.3.24} $$

$$ \bar{}^{L_b}v_{L_b} = {}^{L_b}L_{B_t} \, \bar{}^{B_t}v_{B_t} + {}^{L_b}J_{B_t} \, {}^{B_t}v_{L_b} \tag{7.3.25} $$

$$ \bar{}^{L}v_{L} = {}^{L}L_{L_b} \, \bar{}^{L_b}v_{L_b} \tag{7.3.26} $$

and for j=1 and 2:

$$ \bar{}^{A_j}v_{A_j} = {}^{A_j}L_B \, \bar{}^{B}v_B \tag{7.3.27} $$

$$ \bar{}^{W_j}v_{W_j} = {}^{W_j}L_{A_j} \, \bar{}^{A_j}v_{A_j} + {}^{W_j}J_{A_j} \, {}^{A_j}v_{W_j}. \tag{7.3.328} $$

In parallel with the velocity transformations, we define the joint acceleration six-vectors for j=1 and 2:

$$ {}^{B_t}a_{L_b} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}^T \quad \text{and} \quad {}^{A_j}a_{W_j} = \begin{pmatrix} 0 & 0 & 0 & {}^{A_j}\alpha_{W_jx} & 0 & 0 \end{pmatrix}^T $$

the WMR body acceleration six-vector

$$ \bar{}^{B}a_B = \begin{pmatrix} \bar{}^{B}a_{Bx} & \bar{}^{B}a_{By} & 0 & 0 & 0 & \bar{}^{B}\alpha_{Bz} \end{pmatrix}^T $$

the rotational velocity product vectors according to (3.4.23) for j=1 and 2:

$$\bar{A}_j\omega^r_{W_j}=\begin{pmatrix}0 & 0 & \bar{A}_j\omega_{A_jy}{}^{A_j}\omega_{W_jx} & 0 & \bar{A}_j\omega_{A_jz}{}^{A_j}\omega_{W_jx} & 0\end{pmatrix}^T$$

$$\bar{B}_t\omega^r_{L_b}=\begin{pmatrix}0 & 0 & 0 & 0 & 0 & 0\end{pmatrix}^T$$

the squared velocity product vectors according to (3.4.24) for j=1 and 2:

$$\bar{B}\omega^2_B=\begin{pmatrix}0 & 0 & \bar{B}\omega^2_{Bz} & 0 & 0 & 0\end{pmatrix}^T$$

$$\bar{A}_j\omega^2_{A_j}=\begin{pmatrix}\bar{A}_j\omega^2_{A_jx} & \bar{A}_j\omega^2_{A_jy} & \bar{A}_j\omega^2_{A_jz} & \bar{A}_j\omega_{A_jx}\bar{A}_j\omega_{A_jy} & \bar{A}_j\omega_{A_jx}\bar{A}_j\omega_{A_jz} & \bar{A}_j\omega_{A_jy}\bar{A}_j\omega_{A_jz}\end{pmatrix}^T$$

$$\bar{L}_b\omega^2_{L_b}=\begin{pmatrix}\bar{L}_b\omega^2_{L_bx} & \bar{L}_b\omega^2_{L_by} & \bar{L}_b\omega^2_{L_bz} & \bar{L}_b\omega_{L_bx}\bar{L}_b\omega_{L_by} & \bar{L}_b\omega_{L_bx}\bar{L}_b\omega_{L_bz} & \bar{L}_b\omega_{L_by}\bar{L}_b\omega_{L_bz}\end{pmatrix}^T$$

$$\bar{B}_t\omega^2_{B_t}=\begin{pmatrix}\bar{B}_t\omega^2_{B_tx} & \bar{B}_t\omega^2_{B_ty} & \bar{B}_t\omega^2_{B_tz} & \bar{B}_t\omega_{B_tx}\bar{B}_t\omega_{B_ty} & \bar{B}_t\omega_{B_tx}\bar{B}_t\omega_{B_tz} & \bar{B}_t\omega_{B_ty}\bar{B}_t\omega_{B_tz}\end{pmatrix}^T$$

and the translational velocity product vectors according to (3.4.25) for j=1 and 2:

$$\bar{A}_j\omega^t_{W_j}=\bar{B}_t\omega^t_{L_b}=\begin{pmatrix}0 & 0 & 0 & 0 & 0 & 0\end{pmatrix}^T$$

We then transform recursively the accelerations according to (3.4.12) and (3.4.13) to obtain:

$$\bar{B}_t\mathbf{a}_{B_t}={}^{B_t}\mathbf{L}_B\,\bar{B}\mathbf{a}_B+{}^{B_t}\mathbf{S}_B\,\bar{B}\omega^2_B \tag{7.3.29}$$

$$\bar{L}_b\mathbf{a}_{L_b}={}^{L_b}\mathbf{L}_{B_t}\,\bar{B}_t\mathbf{a}_{B_t}+{}^{L_b}\mathbf{J}_{B_t}\,{}^{B_t}\mathbf{a}_{L_b}+{}^{L_b}\mathbf{R}_{B_t}\,\bar{B}_t\omega^r_{L_b}+{}^{L_b}\mathbf{S}_{B_t}\,\bar{B}_t\omega^2_{B_t}+{}^{L_b}\mathbf{T}_{B_t}\,\bar{B}_t\omega^t_{L_b} \tag{7.3.30}$$

$$\bar{L}\mathbf{a}_L={}^{L}\mathbf{L}_{L_b}\,\bar{L}_b\mathbf{a}_{L_b}+{}^{L}\mathbf{S}_{L_b}\,\bar{L}_b\omega^2_{L_b} \tag{7.3.31}$$

and for j=1 and 2:

$$\bar{}^{A_j}a_{A_j} = {}^{A_j}L_B \, \bar{}^{B}a_B + {}^{A_j}S_B \, \bar{}^{B}\omega_B^2 \qquad (7.3.32)$$

$$\bar{}^{W_j}a_{W_j} = {}^{W_j}L_{A_j} \, \bar{}^{A_j}a_{A_j} + {}^{W_j}J_{A_j} \, {}^{A_j}a_{W_j} + {}^{W_j}R_{A_j} \, \bar{}^{A_j}\omega_{W_j}^{r} + {}^{W_j}S_{A_j} \, \bar{}^{A_j}\omega_{A_j}^{2} + {}^{W_j}T_{A_j} \, \bar{}^{A_j}\omega_{W_j}^{t} \qquad (7.3.33)$$

7.3.6 Dynamic Equations-of-Motion

In Section 7.3.4, we have computed the six primary force/torque equations-of-motion and the N_s=18 secondary force/torque equations-of-motion. We program MACSYMA to compute and substitute the link velocities and accelerations (in Section 7.3.5), referenced to instantaneously coincident coordinate systems, into the *twenty-four* force/torque equations-of-motion to obtain the dynamic model of Bicsun-Bicas displayed in Tables 7.1 and 7.2.

The six *primary* equations-of-motion (P1)-(P6) are displayed in Table 7.1. The three *translational equilibrium equations* (P1)-(P3) have a straightforward physical interpretation. The translational x and y equations (P1) and (P2) sum the viscous frictional, inertial, and environmental contact forces in the x and y directions, respectively, to zero. The z equation (P3) sums the gravitational and normal environmental contact forces to zero. The *rotational equilibrium equations* (P4)-(P6) are complex functions of the frictional coefficients, and the actuator, viscous frictional, inertial, and environmental contact forces/torques.

The eighteen *secondary* equations-of-motion (S1)-(S18) are displayed in Table 7.2: equations (S1)-(S12) are formulated at the wheel contact points, equations (S13)-(S16) are formulated at the wheel axles, and equations (S17)-(S18) are formulated at the steering axles. The complexity of the secondary equations-of-motion increases as we ascend from the floor contact joints towards the main body.

Table 7.1: Bicsun-Bicas Primary Dynamic Equations-of-Motion

P1: $-\sin({}^{H3}\theta_{S3z})\, f_{y3} + \cos({}^{H3}\theta_{S3z})\, f_{x3} - \sin({}^{H4}\theta_{S4z})\, f_{y4} + \cos({}^{H4}\theta_{S4z})\, f_{x4}$
$+ 2\, l_b\, m_W\, {}^{\overline{B}}\alpha_{Bz} - \mu_{vBx}\, {}^{\overline{B}}v_{Bx} - (2\, m_W + m_L + m_B)\, {}^{\overline{B}}a_{Bx} + f_{x1} + f_{x2} = 0$

P2: $\sin({}^{H3}\theta_{S3z})\, f_{x3} + \cos({}^{H3}\theta_{S3z})\, f_{y3} + \sin({}^{H4}\theta_{S4z})\, f_{x4} + \cos({}^{H4}\theta_{S4z})\, f_{y4}$
$+ 2\, l_b\, m_W\, {}^{\overline{B}}\omega^2_{Bz} - \mu_{vBy}\, {}^{\overline{B}}v_{By} - (2\, m_W + m_L + m_B)\, {}^{\overline{B}}a_{By} + f_{y1} + f_{y2} = 0$

P3: $-(2\, m_W + m_L + m_B)\, g + f_{z1} + f_{z2} + f_{z3} + f_{z4} = 0$

P4: $-2\, l_d\, m_W\, {}^{\overline{B}}a_{By} + (l_{Bh}+l_{Lh})\, m_L\, {}^{\overline{B}}a_{By}/2 + 2\, l_b\, l_d\, m_W\, {}^{\overline{B}}\omega^2_{Bz} - \beta\, \tau_1 - \beta\, \tau_2$
$+ (R+l_d)\sin({}^{H4}\theta_{S4z})f_{x4} + (R+l_d)\sin({}^{H3}\theta_{S3z})f_{x3} - (2l_b+\zeta_{cw2}+\zeta_{cw1})m_W\, g$
$+ (R-r+l_d)\cos({}^{H4}\theta_{S4z})\, f_{y4} + (\zeta_{cw4}-l_g)\cos({}^{H4}\theta_{S4z})\, f_{z4}$
$+ (R-r+l_d)\cos({}^{H3}\theta_{S3z})\, f_{y3} + (\zeta_{cw3}-l_g)\cos({}^{H3}\theta_{S3z})\, f_{z3}$
$+ l_d\, f_{y1} + (l_b+\zeta_{cw1})\, f_{z1} + l_d\, f_{y2} + (l_b+\zeta_{cw2})\, f_{z2} - l_c\, f_{z3} - l_c\, f_{z4} = 0$

P5: $2\, l_d\, m_W\, {}^{\overline{B}}a_{Bx} - (l_{Lh} + l_{Bh})\, m_L\, {}^{\overline{B}}a_{Bx}/2 - 2\, l_b\, l_d\, m_W\, {}^{\overline{B}}\alpha_{Bz}$
$+ (2I_{Wy}-I_{Wx})\, {}^{\overline{B}}\omega_{Bz}\, {}^{A2}\omega_{W2x} + (2I_{Wy}-I_{Wx})\, {}^{\overline{B}}\omega_{Bz}\, {}^{A1}\omega_{W1x}$
$-(R+l_d)\cos({}^{H3}\theta_{S3z})f_{x3} + (R-r+l_d)\sin({}^{H3}\theta_{S3z})f_{y3} + (\zeta_{cw3}-l_g)\sin({}^{H3}\theta_{S3z})f_{z3}$
$-(R+l_d)\cos({}^{H4}\theta_{S4z})f_{x4} + (R-r+l_d)\sin({}^{H4}\theta_{S4z})f_{y4} + (\zeta_{cw4}-l_g)\sin({}^{H4}\theta_{S4z})f_{z4}$
$- R\, f_{x1} - R\, f_{x2} - l_d\, f_{x1} - l_a\, f_{z1} - l_d\, f_{x2} + l_a\, f_{z2} + l_e\, f_{z3} - l_e\, f_{z4} = 0$

P6: $-2(l_b^2 + l_a^2)\, m_W\, {}^{\overline{B}}\alpha_{Bz} - (2I_{Wy}+I_{Lz}+I_{Bz})\, {}^{\overline{B}}\alpha_{Bz} + 2\, l_b\, m_W\, {}^{\overline{B}}a_{Bx} - \varepsilon_{vBz}\, {}^{\overline{B}}\omega_{Bz}$
$+ [l_c\cos({}^{H3}\theta_{S3z}) - l_e\sin({}^{H3}\theta_{S3z})]f_{x3} - [l_c\sin({}^{H3}\theta_{S3z}) - l_e\cos({}^{H3}\theta_{S3z})]\, f_{y3}$
$+ [l_e\sin({}^{H4}\theta_{S4z}) + l_c\cos({}^{H4}\theta_{S4z})]\, f_{x4} + [l_e\cos({}^{H4}\theta_{S4z}) - l_c\sin({}^{H4}\theta_{S4z})]\, f_{y4}$
$- l_b f_{x1} + l_a f_{y1} + \zeta_{cc1} f_{z1} - l_b f_{x2} - l_a f_{y2} + \zeta_{cc2} f_{z2} + \zeta_{cs3} f_{z3} + \zeta_{cs4} f_{z4} = 0$

Table 7.2: Bicsun-Bicas Secondary Dynamic Equations-of-Motion

S1-S4:	$\tau_{xi} + \zeta_{ri}\, f_{zi} = 0$	for i=1,...,4
S5-S8:	$\tau_{yi} = 0$	for i=1,...,4
S9-S12:	$\tau_{zi} + \zeta_{cci}\, f_{zi} = 0$	for i=1,...,4
S13-S14:	$\beta\, \tau_j + R\, f_{yj} - \zeta_{cwj}\, m_W\, g + \zeta_{rj}\, f_{zj} + \zeta_{cwj}\, f_{zj} - I_{Wx}\, {}^{Aj}\alpha_{Wjx} = 0$	for j=1 and 2
S15-S16:	$r\, f_{yk} + \zeta_{rk}\, f_{zk} + \zeta_{cwk}\, f_{zk} = 0$	for k=3 and 4
S17-S18:	$I_g\, f_{xk} + \zeta_{csk}\, f_{zk} + \zeta_{cck}\, f_{zk} = 0$	for k=3 and 4

In Table 7.3, we compare and contrast the dynamic models of Uranus in Tables 6.1 and 6.2 and Bicsun-Bicas in Tables 7.1 and 7.2. The force/torque coupling between the Bicsun-Bicas axes-of-motion is evident in the primary dynamic equations-of-motion in Table 7.1. The coupling between the x-translational and z-rotational axes, which we computed as a kinematic motion constraint in (7.2.9), is manifested in the coupling torque ($2\, I_b\, m_W\, {}^{\overline{B}}a_{Bx}$) in (P6). There are *eleven* coupling terms in the Bicsun-Bicas dynamic model, compared with only *three* in the Uranus dynamic model. There are also 32 nonlinear components in the Bicsun-Bicas primary dynamic equations-of-motion, including the terms containing the trigonometric functions of the steering angles $\sin({}^{Hk}\theta_{Skz})$ and $\cos({}^{Hk}\theta_{Skz})$ for k=3 and 4 and the squared velocities such as ${}^{\overline{B}}\omega^2_{Bz}$. The Uranus dynamic model contains only four nonlinear terms. Overall, there are six times as many coupling and nonlinear terms in the Bicsun-Bicas dynamic model as in the Uranus dynamic model even though there is less than

one-half the number of computations (198 computations for Bicsun-Bicas vs. 423 for Uranus). The numbers of computations for the two dynamic models is nearly equal when friction is neglected (164 for Uranus and 146 for Bicsun-Bicas). *We conclude that the computational complexity of a WMR dynamic model without friction is directly related to the number of wheels, and the computational complexity with friction is directly related to both the number of wheels and the number of frictional coefficients modeled.* Moreover, four times the cpu minutes are required to model the more complex Uranus (16.82 minutes) than Bicsun-Bicas (4.30 minutes).

Table 7.3: WMR Dynamic Model Comparison

	Uranus	Bicsun-Bicas
DOFs	3	2
Wheels	4	4
Frictional Coefficients	36	14
Number of Equations	26	24
Computations No Friction	146	164
Computations w/Friction	423	198
Coupling Terms	3	11
Nonlinear Terms	4	32
Modeling Time	16.82 minutes	4.30 minutes

7.3.7 Dynamic Solutions

The complexity of the Bicsun-Bicas dynamic model in Tables 7.1 and 7.2 precludes symbolic solution by MACSYMA. We therefore compute numerically the actuated inverse and forward dynamic solutions of Bicsun-Bicas. In both solutions, we reduce the system of *twenty-four* equations to six equations by solving independently each of the secondary dynamic equations-of-motion for distinct elements of the floor reactional forces/torques. We solve (S1)-(S4) for τ_{xi}, (S5)-(S8) for τ_{yi}, (S9)-(S12) for τ_{zi}, (S13)-(S16) for f_{yi}, and (S17)-(S18) for f_{xk}; for i=1,...,4 and k=3 and 4. We then substitute these eighteen floor reactional forces/torques into the six primary dynamic equations-of-motion. We

call the resulting six equations the *reduced* dynamic model. By inspection of the reduced dynamic model, we determine that the floor reactional forces f_{x1} and f_{x2} of Bicsun-Bicas are dependent variables. The reduced dynamic model is thus linear in the three WMR body accelerations ${}^{\bar{B}}a_{Bx}$, ${}^{\bar{B}}a_{By}$, and ${}^{\bar{B}}\alpha_{Bz}$, the five independent floor reactional forces f_{z1}, f_{z2}, f_{z3}, f_{z4}, and f_{x1} and the two actuator torques τ_1 and τ_2.

We compute the forward dynamic solution (as described in Section 3.5.2) by solving the reduced dynamic model for the unknown WMR body accelerations ${}^{\bar{B}}a_{Bx}$, ${}^{\bar{B}}a_{By}$, and ${}^{\bar{B}}\alpha_{Bz}$ and the unknown floor reactional forces f_{z1}, f_{z2}, f_{z3}, f_{z4}, and f_{x1}. For the forward dynamic solution, the reduced dynamic model is a set of six independent linear algebraic equations in eight unknowns. We apply Gauss elimination to the y-translational and z-rotational equilibrium equations of the reduced dynamic model to compute the unknown WMR body accelerations ${}^{\bar{B}}a_{Bx}$, ${}^{\bar{B}}a_{By}$ symbolically in terms of the unknown floor reactional forces. By differentiating the kinematic motion constraint in (7.2.9), we compute the z-rotational WMR body acceleration as the x-translational acceleration divided by the length l_b. We then substitute the symbolic solutions into the remaining four equations. We apply Gauss elimination to compute the floor reactional force f_{x1}, leaving three equations in four unknown floor reactional forces f_{z1}, f_{z2}, f_{z3}, and f_{z4}. We apply the minimum norm solution to solve numerically these three equations and compute the floor reactional forces. We then compute the numerical values of the WMR body accelerations by substituting the numerical values of the floor reactional forces into the symbolic solution for the WMR body accelerations.

We compute the actuated inverse dynamic solution (as described in Section 3.5.3) by solving the reduced dynamic model for the unknown actuator torques τ_1 and τ_2, and the unknown floor reactional forces f_{z1}, f_{z2}, f_{z3}, f_{z4}, and f_{x1}. We apply the minimum-norm solution to the reduced dynamic model consisting of six independent linear algebraic equations in seven unknowns. We apply Gauss elimination to solve symbolically the y-translational and z-rotational equilibrium equations of the reduced dynamic model for the two actuator torques. We compute numerically the floor reactional forces as in the forward dynamic solution. We then substitute the numerical floor reactional forces into

the symbolic solution for the actuator torques to compute numerically the actuator torques.

In both the actuated inverse and forward solutions, the four unknown z-axis floor reactional forces are computed from a set of three independent equations according to the minimum norm solution. This solution correctly incorporates the indeterminate nature of the four-point contact to a plane. The determined nature of the actuation structure of Bicsun-Bicas is also illuminated by the direct solution of the two wheel actuator torques from two equations. The determined nature of the motion of the WMR is indicated through the direct solution of two independent equations for the two independent WMR body accelerations. In physical terms this means that: for any combination of actuator torques there is a unique set of WMR body accelerations, and for any set of WMR body accelerations which satisfy the kinematic motion constraint in (7.2.9) there is a unique set of actuator torques.

7.3.8 Concluding Remarks

We have computed the dynamic model of Bicsun-Bicas. We found that the dynamic model of Bicsun-Bicas differs inherently from that of Uranus in that there are six times as many coupling and nonlinear terms due to the kinematic motion constraint and the castor steering angles. In contrast, the Bicsun-Bicas dynamic model is only half as computationaly complex as that of Uranus. In the following sections, we apply the kinematic modeling in Section 7.2 and the dynamic model in this section to design servo-controllers for Bicsun-Bicas and simulate the combined WMR/servo-controller operation to evaluate the servo-controllers.

7.4 Modeling Parameter Values

The numerical values of the kinematic, dynamic, and frictional parameters for the Bicsun-Bicas simulation experiments are displayed in Table 7.4. To the extent practical, the numerical values of these parameters are chosen to correspond to the numerical values of the Uranus parameters (in Table 6.3) to enable performance comparisons between WMRs.

Table 7.4: Bicsun-Bicas Numerical Modeling Parameter Values

Name	Value	Units	Meaning
l_{Bh}	0.3048	m	body height
l_{Lh}	0.3048	m	load height
l_a	0.2667	m	half body width
l_b	0.2667	m	half body length
l_c	0.2667	m	caster y-displacement
l_d	0.1524	m	wheel z-displacement
l_e	0.2667	m	caster x-displacement
l_f	0.1524	m	caster z-displacement
l_g	0.0254	m	caster steering link length
R	0.1127	m	wheel radius
r	0.0381	m	roller radius
m_B	90.72	Kg	body mass
m_L	0	Kg	nominal load mass
m_W	0.3629	Kg	wheel mass
I_{Bz}	4.835	Kg m^2	body inertia about z-axis
I_{Lz}	$m_L l_{Lh}^2/6$	Kg m^2	load inertia about z-axis
I_{Wx}	0.02305	Kg m^2	wheel inertia about x-axis
I_{Wy}	0.02480	kg m^2	wheel inertia about y-axis
g	9.8	m s^{-2}	gravitational constant
β	4		gear ratio
μ_{vBx}	45.36	kg s^{-1}	body viscous friction along x-axis
μ_{vBy}	45.36	kg s^{-1}	body viscous friction along y-axis
ε_{vBz}	2.417	kg m s^{-1}	body viscous friction about z-axis
bf	0.01		base friction constant
μ_r	0.001 bf		rolling friction constant
μ_c	0.01 bf		Coulomb friction constant
R_m	2.24	ohms	motor resistance
k_b	0.459	volt s^{-1}	motor back emf constant
k_t	0.459	N m amp^{-1}	motor torque constant
v_{max}	22.4	volt	maximum motor voltage

7.5 Servo-Controller Designs

In this section, we address the two model-based servo-controllers detailed in Chapter 4 for Bicsun-Bicas: resolved motion rate and resolved acceleration servo-control. As noted in Section 7.2.5, the x-translational and z-rotational Bicsun-Bicas axes of motion are dependent, resulting in significant coupling forces/torques between these two axes. There are also significant nonlinear terms in the Bicsun-Bicas dynamic model due to the steering angles on wheels three and four as described in Section 7.3.6. We were thus unable to design a *stable* resolved motion rate servo-controller for Bicsun-Bicas. Since resolved motion rate servo-control does not compensate for such coupling and nonlinear forces/torques, this servo-controller is incapable of stable operation of Bicsun-Bicas. In contrast, the Uranus dynamic model is relatively free of coupling and nonlinear terms and resolved motion rate servo-control was shown in Section 6.7 to be adequate.

We have successfully designed a stable resolved acceleration servo-controller for Bicsun-Bicas by applying the WMR servo-controller design methodology specified in Section 4.3. Since resolved acceleration servo-control compensates for coupling and nonlinear forces/torques, this servo-controller is capable of stable operation of Bicsun-Bicas. The computational components of the resolved acceleration servo-controller have been formulated in this chapter: the sensed forward velocity solution in Section 7.2.6, the dead reckoning algorithm in Section 7.2.8, and the actuated inverse dynamic solution in Section 7.3.7. The torque-to-voltage and torque-to-pulse-width command conversion algorithms have been developed in Section 4.4. We need only specify the controller gains k_{pi} and k_{di} for i=x and y to complete the design. Following the guidelines specified in Section 4.3, we arrive at the following servo-controller gains for the resolved acceleration servo-control of Bicsun-Bicas:

$$k_{px} = 10\ s^{-2}, \qquad k_{dx} = 6.32\ s^{-1}, \quad k_{py} = 10\ s^{-2}, \qquad k_{dy} = 6.32\ s^{-1}\ .$$

7.6 Simulation Experiments

We conduct simulation experiments to evaluate the performance of the resolved acceleration servo-controller for Bicsun-Bicas. Our set of experiments

consists of one simulation with all simulation variables assigned to their nominal values. The remaining experiments change one or more of the simulation variables from their nominal values to study their effects on the servo-controller performance. In this section, we present the experimental data; in Section 7.7, we analyze the data and evaluate the servo-controller performance.

In Table 7.5, we list the experiments and corresponding simulation variables for Bicsun-Bicas under resolved acceleration servo-control. In Table 7.6, we highlight the salient results of the resolved acceleration simulation experiments.

Table 7.5: Bicsun-Bicas Resolved Acceleration Simulation Experiments

Experiment Number	Command Type	Sampling Period (s)	Task Time (s)	Base Friction Sim./Con.	Load Mass (Kg) Sim./Con.	I/O Precision (bits)	Trajectory Type
bd1	voltage	0.2	10.0	0.01/0.01	0/0	16	step
bd2	voltage	0.2	10.0	0/0	0/0	16	step
bd3	voltage	0.2	10.0	0.01/0	0/0	16	step
bd4	voltage	0.2	10.0	0.01/0.01	0/0	16	spline
bd5	voltage	0.2	10.0	0/0	0/0	16	spline
bd6	voltage	0.2	10.0	0.01/0	0/0	16	spline
bd7	voltage	0.2	10.0	0.01/0.01	45/0	16	spline
bd8	voltage	0.2	10.0	0.01/0.01	0/45	16	spline
bd9	voltage	0.2	5.0	0.01/0.01	0/0	16	spline
bd10	pwm	0.2	10.0	0.01/0.01	0/0	16	spline
bd11	voltage	0.2	10.0	0.01/0.01	0/0	8	spline
bd12	voltage	0.2	10.0	0.01/0.01	0/0	16	circle
bd13	voltage	0.04	10.0	0.01/0.01	0/0	16	spline
bd14	pwm	0.2	5.0	0.01/0	45/0	8	spline
bd15	voltage	0.2	10.0	0.1/0	0/0	16	spline

We plot the nominal step (Experiment bd1), spline (Experiment bd4) and circle (Experiment bd12) trajectories for Bicsun-Bicas under resolved acceleration servo-control in Figures 7.4, 7.5 and 7.6, respectively. The unfilled squares indicate the reference trajectory; the solid diamonds portray the actual WMR trajectory.

Table 7.6: Bicsun-Bicas Resolved Acceleration Simulation Results

Experiment Number	Adds/ Multiplies	Max. x-y Error (m)	Accum. Error	Settling Time (s)	Steady S. x-y Error (m)	Command Saturates	Simulation Time (min)
bd1	120/227	-	0.372	1.6	0.000	3	93.7
bd2	24/32	-	0.360	1.6	0.000	3	129.8
bd3	24/32	-	0.372	1.6	0.000	3	92.7
bd4	120/227	0.008	0.026	-	-	0	151.1
bd5	24/32	0.008	0.023	-	-	0	128.6
bd6	24/32	0.008	0.026	-	-	0	143.8
bd7	120/227	0.008	0.027	-	-	0	154.4
bd8	120/227	0.008	0.026	-	-	0	149.8
bd9	120/227	0.026	0.044	-	-	0	72.6
bd10	120/227	0.026	0.130	-	-	0	102.3
bd11	120/227	0.009	0.042	-	-	0	123.7
bd12	120/227	0.011	0.067	-	-	0	109.5
bd13	120/227	0.005	0.025	-	-	0	158.2
bd14	24/32	0.033	0.076	-	-	0	75.6
bd15	120/227	0.030	0.099	-	-	1	122.9

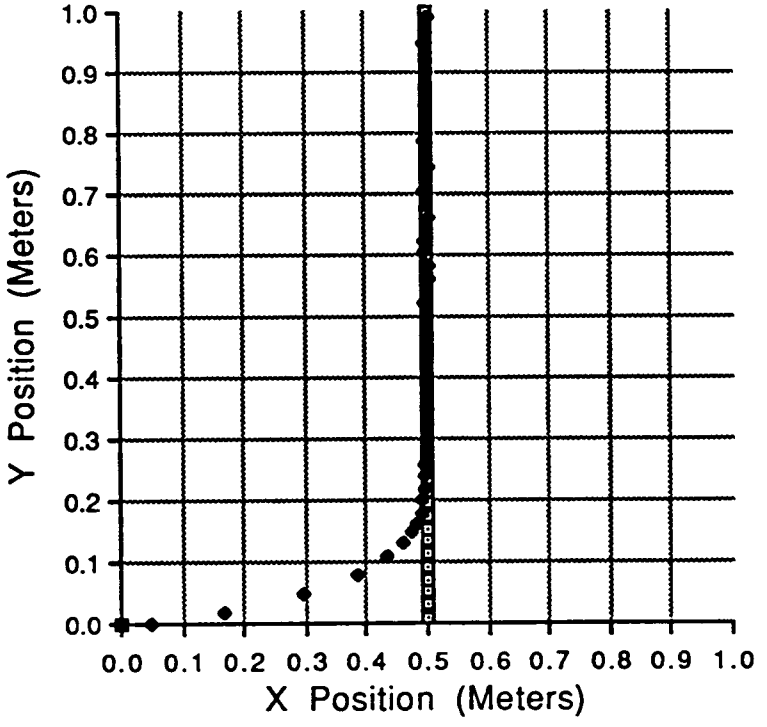


Figure 7.4: Bicsun-Bicas Resolved Acceleration Control On Nominal Step Reference Trajectory (Experiment bd1)

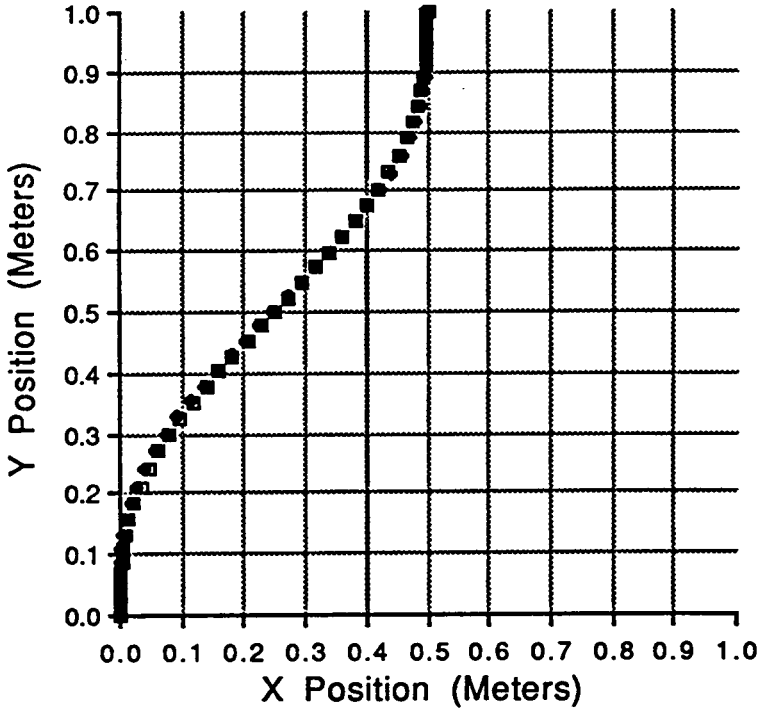


Figure 7.5: Bicsun-Bicas Resolved Acceleration Control On Nominal Spline Reference Trajectory (Experiment bd4)

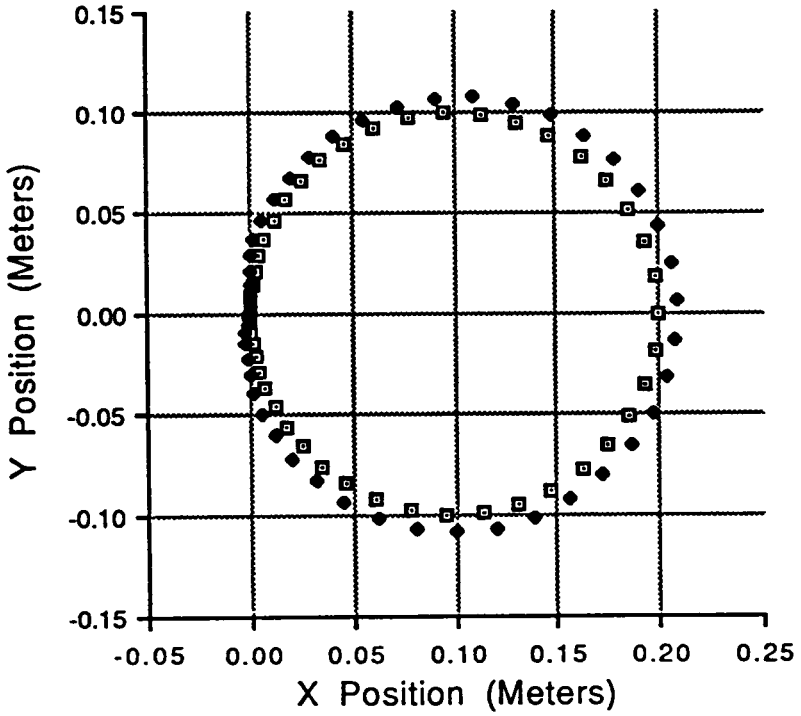


Figure 7.6: Bicsun-Bicas Resolved Acceleration Control On Nominal Circle Reference Trajectory (Experiment bd12)

The simulation times for Bicsun-Bicas (in Table 7.6) under resolved acceleration servo-control are more difficult to interpret than those for Uranus (in Tables 6.5 and 6.7). We suggest that the large number of coupling and nonlinear components in the dynamic model are the source of the simulation time variations. Coupling and nonlinearities within a dynamic system tend to cause oscillations in the response. The large number of coupling and nonlinear terms thus destabilize the WMR/servo-controller system. Without proper dynamic compensation (such as with resolved motion rate servo-control), the WMR is unstable. We may stabilize the system by feedforward dynamic compensation (as with resolved acceleration servo-control). The high frequency components embodied in the step experiment excite the oscillatory modes of the WMR, but friction has an inherent damping effect. When friction is zeroed (in Experiment bd2), the oscillations are enhanced, producing a quickly varying oscillatory response requiring the step size of the simulation to decrease as compared with the nominal step experiment (Experiment bd1) and thereby increase the simulation time. In contrast, the nominal spline tracking experiment (Experiment bd4) has no high frequency content and does not excite the oscillatory modes of Bicsun-Bicas. The simulation times for the spline experiment thus exhibit the trends noted for Uranus. We observe a decrease in simulation time (128.6 minutes in Experiment bd5) when friction is zeroed as compared with the nominal spline experiment with friction (151.1 minutes in Experiment bd4).

The range of simulation times for Uranus (from 15.0 min in Experiment ud2 to 236.2 min in Experiment uk11: a 16 to 1 ratio) is eight times the range of simulation times for Bicsun-Bicas (from 72.6 min in Experiment bd9 to 158.2 min in Experiment bd13: a ratio of 2 to 1). The large range in simulation times for Uranus is due to friction. When no friction is simulated for Uranus, the simulation times are small because the WMR response is smooth. By simulating friction, we introduce high frequency signals in the simulation which require drastically longer simulation times. For Bicsun-Bicas simulations without friction, the significant coupling and nonlinear components in its dynamic model introduce high frequency signals which require moderately long simulation times. Although friction adds high frequency signals to Bicsun-Bicas simulations, it also dampens the high frequency signals from the coupling and nonlinear

components. So, the increase in simulation times when friction is added is not as dramatic as it is with Uranus.

7.7 Servo-Controller Evaluations

We interpret the results of the simulation experiments tabulated in Section 7.6 to evaluate the performance of Bicsun-Bicas under resolved acceleration servo-control. According to our criterion for acceptable WMR/servo-control performance (described in Section 6.7.1), Bicsun-Bicas under servo-control will show acceptable performance if the 95% settling time during the nominal step experiment is less than 6 s and the maximum deviation from the path component of the maximum spatial error during the nominal spline experiment is less than 2.7 cm. These criteria are identical to those for Uranus because we designed Bicsun-Bicas with the same body mass and dimensions as Uranus.

We interpret the response and tracking characteristics to determine whether the performance is acceptable. Then we discuss the effects of friction, load variations, finite precision, simultaneous rotation, pulse-width modulation, and sampling period variations on the combined WMR/servo-controller performance to determine the sensitivity of the performance to these realistic conditions.

The nominal step Experiment bd1 (in Figure 7.4) is the basis for evaluating the Bicsun-Bicas resolved acceleration servo-controller response characteristics. We observe an acceptable settling time of 1.6 s with zero overshoot and zero steady state error. The nominal step experiment experiences three command saturations because of the abrupt change in reference trajectory embodied in a step. We observe small amplitude (0.2 cm) high frequency oscillations (1.25 Hz) during steady-state tracking. We attribute these oscillations to a numerical limit cycle because of their small magnitude and high frequency (1/4 cycle/sampling period). Bicsun Bicas under resolved acceleration servo-control has a settling time equal to that of Uranus under resolved motion rate servo-control but 0.6 s longer than that of Uranus under resolved acceleration servo-control.

The nominal spline Experiment bd4 (in Figure 7.5) and nominal circle Experiment bd12 (in Figure 7.6) are representative of typical curved WMR trajectories. Bicsun-Bicas resolved acceleration servo-control tracks the nominal spline and circle trajectories with maximum spatial errors of 0.8 cm and 1.1 cm, respectively. Since the maximum deviation from the path is a component of, and thus less than, the maximum spatial error, the resolved acceleration servo-controller has acceptable performance characteristics. The maximum error of the circle trajectory is larger than that of the spline because the circle trajectory requires more WMR body rotations for a two DOF WMR than does the spline. The tracking errors are dependent upon the speeds and accelerations of the WMR along the trajectories as demonstrated by Experiment bd9. In spline Experiment bd9, the total experiment time is reduced to 5 s which increases the reference velocities and accelerations and thereby increases the maximum spatial error to 2.6 cm. Bicsun-Bicas under resolved acceleration servo-control has better tracking performance than that of Uranus under resolved motion rate servo-control, but is not as accurate as Uranus under resolved acceleration servo-control.

We zero the friction in the simulation and servo-controller in step Experiment bd2 and spline Experiment bd5 and compare the results with the nominal step and spline responses. In both experiments, the accumulated error is reduced only slightly. In spline Experiment bd15, we set the simulated friction to ten times the nominal value while leaving the servo-controller friction at zero. The tracking results show a 3 cm maximum spatial tracking error. This represents a 275% increase in maximum spatial error, but is acceptable because the maximum deviation from the path is less than 2.7 cm. We conclude that for Bicsun-Bicas resolved acceleration servo-control with a base friction value of 0.1 or less, no dry frictional compensation is required. By assuming no friction in the servo-controller, we reduce the total number of additions and multiplications from 347 to 56; a 84% decrease in computational requirements. Once again, this represents a marked decrease in controller complexity and required *a priori* knowledge because we need not identify the friction coefficients for the servo-control algorithm. Bicsun-Bicas under resolved acceleration servo-control exhibits the insensitivity to friction displayed by Uranus under resolved acceleration servo-control.

We vary the load in Experiments bd7 and bd8. In Experiment bd7, the simulation operates with a 45 Kg load mass (one-half the mass of the WMR) on Bicsun-Bicas while the servo-controller assumes zero load mass. In Experiment bd8, the simulation operates with zero load mass while the servo-controller assumes a 45 Kg load mass. The results are unchanged from the nominal spline trajectory experiment indicating an insensitivity of the performance to load variations. Uranus under both resolved motion rate and resolved acceleration servo-control similarly exhibited an insensitivity to load variations.

The precision of actuator commands and feedback signals are limited to 8 bits in Experiment bd11. The results show an increase in the maximum spatial error and accumulated error relative to the nominal spline trajectory. The resulting maximum spatial error (0.9 cm), which is within our acceptability limits, indicates that small, inexpensive microprocessors with 8 bit I/O busses may be applied to implement the resolved acceleration servo-controller. Uranus under resolved motion rate servo-control shows a decrease in maximum spatial error with 8 bit precision, and Uranus under resolved acceleration servo-control shows an increased maximum spatial error (to 0.7 cm). The performance of both Uranus and Bicsun-Bicas is virtually unaffected when the precision is limited to 8 bits.

In contrast to Uranus, we cannot specify a simultaneous rotation reference trajectory for Bicsun-Bicas because the resolved acceleration servo-controller designed in Section 7.5 operates in the two dimensions x and y. Any reference rotation would be ignored by the servo-controller. The orientation of the WMR is dependent upon its x-translation as described by the motion constraint in (7.2.9).

The application of PWM motor commands in Experiment bd10 increases the maximum spatial error (from 0.8 cm to 2.6 cm). PWM control thus sacrifices tracking accuracy to obtain power efficiency and design simplicity. As with Uranus under resolved acceleration servo-control, the performance decreases when PWM is applied. This is in contrast with Uranus under resolved motion rate servo-control which exhibits no change in performance when PWM is applied.

We have chosen our nominal sampling period (0.2 s) to be one-tenth of the time constant of Bicsun-Bicas (2 s). The resulting nominal step and spline experiments display an acceptable level of performance according to our criteria in Section 6.7.1. We reduce the sampling period to 0.04 s in Experiment bd13 while keeping the servo-controller gains unchanged. The spatial tracking error is thereby reduced from 0.8 cm to 0.5 cm. This represents a slight improvement in tracking performance over the nominal spline experiment. As with Uranus under resolved acceleration servo-control, the performance improves with decreasing sampling period. Uranus under resolved motion rate servo-control exhibits a large increase in maximum spatial error when the sampling period is reduced with no change in controller gains. We conclude that the gains of a resolved motion rate WMR servo-controller are sampling period dependent, whereas those for resolved acceleration servo-control are not.

Our worst case scenario in Experiment bd14 indicates that a fast trajectory, mismatched load and friction values and reduced I/O precision produce a 3.3 cm maximum spatial error. This spatial error is less than what we would expect if the individual error sources were to sum to produce the worst case error. The worst case Bicsun-Bicas error is similar to that of Uranus under resolved acceleration servo-control and much less than that of Uranus under resolved motion rate servo-control. We conclude that simultaneous error sources drastically degrade the performance of the resolved motion rate servo-controller, but simultaneous error sources degrade the performance of the resolved acceleration servo-controller less than the sum of the degradations caused by the individual error sources alone.

7.8 Conclusions and Recommendations

Bicsun-Bicas is a general-purpose two DOF WMR, with the capability to servo to independent x and y (or θ and y) trajectories simultaneously under resolved acceleration servo-control. Because Bicsun-Bicas possesses only two DOFs, two of the WMR axes of motion (x and θ) are dependent. This kinematic motion constraint couples the forces/torques between WMR axes in the dynamic model. The motions of the caster steering angles are a source of nonlinear forces/torques in the WMR dynamic model. Although resolved motion rate

servo-control is adequate for Uranus operation, it is not sufficient for Bicsun-Bicas control because of the significant coupling and nonlinear force/torque components in its dynamic model. The coupling and nonlinearities are not as prevalent in the Uranus dynamic model; consequently, the computationaly and conceptually simpler resolved motion rate servo-controller, which does not compensate for these effects, is adequate. The mechanically simpler WMR Bicsun-Bicas is more difficult to control. This illuminates the trade-off between mechanical and servo-control design efforts and underscores the requirement to incorporate servo-controller design activities within the WMR mechanical and electrical design processes.

Resolved acceleration servo-control provides stable WMR operation. Compensation is not required for small dry frictions (i.e., base frictions less than 0.1) and this reduces the computational requirements of the servo-control algorithm by a factor of six: from 347 to 56 additions and multiplies. Since the time constant of the main body of Bicsun-Bicas is 2 s, a sampling period of 0.2 s has proven to be adequate. There is no difficulty executing the 56 additions and multiplications of the resolved acceleration servo-control algorithm within 0.2 s on conventional microprocessors. The performance of the resolved acceleration servo-controller has been shown to be insensitive to load variations (as much as half the WMR weight), dry friction variations and 8 bit precision. The resolved acceleration servo-controller gains are applicable at different sampling periods and PWM motor control produces a significant, but tolerable, degradation in performance.

We have shown that WMR servo-controller design must be approached on a case-by-case basis. WMR having three DOFs and no steering links (such as Uranus) have minimal coupling and nonlinear components in their dynamic models and allow the application of the computationaly simple resolved motion rate servo-controller design. WMRs with kinematic motion constraints have coupling components and WMRs with steering links have nonlinear components in their dynamic models. If these coupling and nonlinear components are significant (such as with Bicsun-Bicas), resolved motion rate servo-control will be inadequate and a more computationaly complex servo-controller design, such as resolved acceleration, is required.

Chapter 8

Conclusions

8.1 Introduction

We have introduced methodologies for modeling the kinematics and dynamics, and for designing model-based servo-control algorithms for WMRs. We drew upon analogies with the available stationary manipulator literature while incorporating the special characteristics of WMRs: closed-chains, higher-pair joints, unactuated and unsensed joints, friction, and pulse-width modulation actuation. We found that the existing modeling methodologies for stationary manipulators had to be modified and extended considerably for WMR application; but that stationary manipulator computer simulation and Cartesian-space servo-control methodologies are directly applicable. The majority of our research activities and this dissertation have thereby been directed towards WMR modeling. Several spin-off areas of research, such as WMR mobility characterization, dead reckoning and wheel slip detection, which were also illuminated during our studies are documented in the appendices.

There are many aspects of WMR motion and control which we have addressed in this dissertation that are applicable to a broad range of robotic mechanisms and which have as yet undiscovered significance. For example, the concept of force/torque propagation, introduced in Section 3.3.3, has direct physical motivation. Forces/torques do propagate through solids at the speed of sound just as light waves propagate at the speed of light through space. Forces/torques propagating through rigid bodies can thus be modeled by the methodologies of systems engineering and the application of wave phenomena concepts. The foundations of systems engineering in the literature is vast, and the application to robotics has not been fully realized. We may gain new

physical insights into robot motion from such a systems engineering viewpoint For example, let us reconsider the implications of force/torque coupling at a joint with reference to Figure 3.3. We have found that forces/torques propagate directly along joint axes which are not joint DOFs; i.e., joint axes along which the adjoining links are in immobile contact. In systems engineering terminology, we may state that the impedance of the links are matched at the joint and thus the power transfer is high. This is why forces/torques propagate directly along non-DOF axes. In contrast, the impedance of the adjoining links at the joint along joint DOFs is dramatically mismatched and so little power is` transfered. Consequently, no forces/torques are propagated. This joint propagation example is analogous to the matching of amplifier and speaker impedances for electronic audio systems. If impedances can be used to model joints, then they can also be used to model contact between the robot and the environment, such as at the wheel point-of-contact or at the finger tip of a robotic hand. We suggest that many such physical insights into robotic operation are possible and useful for future modeling and control research.

In this final chapter, we conclude our WMR modeling and control study. In Section 8.2, we summarize our developments and highlight the case study results. Then, in Section 8.3, we discuss an enhanced WMR/servo-controller design for general-purpose indoor applications which has become evident through our studies. We enumerate our contributions to WMR kinematic and dynamic modeling and model-based servo-controller design in Section 8.4. Finally, in Section 8.5, we recommend areas of future WMR research which appear promising in light of our results.

8.2 Summary

Our WMR modeling developments in Chapters 2 and 3 have focused on the incorporation of the special WMR characteristics into the WMR kinematic and dynamic models. The *closed-chain structure* of WMRs affects both kinematic and dynamic modeling. Whereas open-chain structures, such as stationary manipulators, allow the sequential solution of the kinematic and dynamic equations-of-motion, closed-chain structures require the simultaneous solution of the equations-of-motion. We applied the least-squares solution to compute the actuated inverse and sensed forward velocity WMR solutions

because of their overdetermined nature; and the minimum norm solution to compute the actuated inverse and forward dynamic WMR solutions because of their undetermined nature. For an open-chain manipulator, the time constant of the smallest link (i.e., the distal link) sets the servo-controller sampling period to be on the order of milliseconds. For a closed-chain WMR, the time constant of the largest link (i.e., the main body) sets the servo-controller sampling period to be on the order of tenths of seconds. We have demonstrated through computer simulation that two prototype WMRs, a three DOF WMR Uranus and a two DOF WMR Bicsun-Bicas, having time constants of 2 seconds can be controlled by servo-controllers having 0.2 second sampling periods. By adding links to stationary manipulators, we may increase its workspace and the number of DOFs. In contrast, the addition of wheels to WMRs imposes constraints on the WMR motion.

The *higher-pair joints* at the point-of-contact between a wheel and the floor cannot be modeled by the Denavit-Hartenberg convention [Denavit55] typically applied to stationary manipulator modeling. We have conceived and applied instantaneously coincident coordinate systems in conjunction with the Sheth-Uicker convention [Sheth71] to resolve this difficulty. Instantaneously coincident coordinate systems are the keystone of our kinematic modeling activities. Instantaneously coincident coordinate systems lead directly to the wheel Jacobian, which relates linearly the velocities of a wheel to the velocities of the WMR body. Instantaneously coincident coordinate systems are also required for the dynamic modeling of inertial and viscous frictional forces/torques of WMRs.

Because some of the joints of a WMR are *not actuated*, compensation for dry frictions at those joints cannot be achieved by the conventional method of adding the frictional force/torque to each joint actuator. The unactuated WMR joints require us to incorporate joint friction in the dynamic model. We developed the joint coupling matrix in conjunction with the concept of force/torque propagation to incorporate joint frictions, including stiction, rolling friction and Coulomb friction along or about any axis. We compute the *actuated* inverse kinematic and dynamic solutions in contrast to the conventional inverse solutions for control applications.

Similarly, some WMR joints are *not sensed*. Consequently, the values of the unsensed joint velocities are unknown for the computation of the dynamic model. We thus apply the *sensed* forward velocity solution in contrast to the conventional forward velocity solution to compute the WMR body velocities from the sensed joint velocities.

Friction is incorporated in our dynamic model within the joint coupling matrices. In contrast to the conventional view of friction as a source of opposing forces/torques at a joint, we have introduced the model of friction as a coupling phenomenon. Rolling, stiction and Coulomb friction couple normal forces/torques at the joint to forces/torques along or about axes which oppose the motion.

We have developed and implemented actuator command conversion algorithms for the application of conventional kinematics and dynamics-based servo-control algorithms to the control of robotic mechanisms through *pulse-width modulation*. We thus obtain the power efficiency of PWM control in conjunction with the conventional robot servo-control algorithms.

We have translated the foregoing conceptual foundations into step-by-step modeling and servo-controller design procedures. This enables the practical engineering of kinematic and dynamic models and servo-control algorithms for a broad range of WMRs and robotic mechanisms. Our WMR kinematic modeling procedure is enumerated in Section 2.5. Beyond WMR applications, our dynamic modeling procedure in Section 3.6 is applicable to a broad spectrum of robotic mechanisms including stationary manipulators, multi-manipulators, WMRs, legged mobile robots and robotic hands. The kinematics-based resolved motion rate servo-controller design in Section 4.2 and the dynamics-based resolved acceleration servo-controller design in Section 4.3 are reduced to gain design procedures once the associated WMR models have been formulated.

We have exemplified our modeling and servo-controller design methodologies in Chapters 6 and 7 by detailing the kinematic and dynamic modeling, servo-controller design, simulation and evaluation of two prototype WMRs: the three DOF WMR Uranus and the two DOF WMR Bicsun-Bicas. We have found that resolved motion rate servo-control is adequate for Uranus

control but not for Bicsun-Bicas control. In contrast to the Uranus dynamic model which contains few coupling and nonlinear components, the Bicsun-Bicas model contains significant coupling components arising from the kinematic constraint between its x and θ axes of motion and nonlinear components arising from its steering angles. Resolved motion rate servo-control, which cannot compensate for such coupling and nonlinear components, was incapable of stable servo-control of Bicsun-Bicas. Resolved acceleration servo-control, which does compensate for coupling and nonlinear components, thus provides adequate control for both Uranus and Bicsun-Bicas.

Our modeling developments and case studies have illuminated an enhanced WMR design which we detail in Section 8.3.

8.3 An Enhanced WMR Implementation

Through our WMR modeling and servo-controller design activities, we have acquired valuable physical insights into WMR design which we apply in this section to recommend a WMR design implementation which is superior for general-purpose indoor applications to the WMRs which we have surveyed in Appendix 2.

WMRs show potential for applications which require the transport of materials, parts, tools, manipulators, sensors or docking devices over distances which are much greater than the working volumes of stationary manipulators. We direct our comments towards mobile operation over hard, smooth surfaces where wheeled locomotion is the most power efficient. The requirements of such tasks are controlled motion along spatial (x-y) trajectories. Typically, the rate of travel and the orientation of the vehicle along the spatial trajectory are not important features of these tasks. To follow an arbitrary spatial x-y trajectory, we require only two DOFs. We conclude that three DOFs are not required for general-purpose applications. A two DOF WMR allows a simpler mechanical design, reducing both construction and hardware costs. The kinematic and dynamic models and servo-control algorithms are also simpler to design and implement.

According to the adequate actuation criterion in (A7.8), a minimum of two actuators is required to control adequately two DOF motion. More than two

actuators may lead to actuator conflict as indicated by the actuation characterization tree in Figure A7.3. The mechanically simplest two DOF WMR with two actuators is from the Bicas family of WMRs: having two diametrically opposed driven non-steered conventional wheels. Such a WMR requires one or more castors (such as on Bicsun-Bicas in Chapter 7) for stability, or the wheels must be mounted at an angle (such as Topo [Helmers83b]). We prefer the application of two conventional castors because WMRs with two angled wheels oscillate about the vertical when in operation and a WMR with one castor is easily toppled by small external disturbances. The commercially-available Labmate WMR [Labmate87] is the most readily obtainable WMR of the Bicas family having four castors.

Because the rate of travel (and tracking lag) is of secondary importance in comparison with spatial deviations from the reference path, we may actuate the WMR to minimize transport time while maintaining its path deviation errors below an acceptable threshold. Algorithms which increase the WMR speed during periods of low path deviation errors and decrease the speed during periods when the path deviation error approaches the threshold can then be developed. If the orientation of manipulators, sensors or docking instruments is pertinent during WMR motion, we may mount a rotating turret on the WMR. A turret, when controlled synchronously with the two DOFs of the base, would give the onboard instruments *three DOFs* (x and y controlled by the motion of the base and θ controlled by the turret motion) without the expense and control complexities of a three DOF wheel base.

We recommend that high quality bearings be incorporated in the WMR construction to minimize joint frictions. We have shown in Chapter 7 that resolved acceleration servo-control is required for such a WMR. By careful mechanical construction, we can reduce the joint frictions and neglect the frictional compensation in the servo-controller algorithm. This, in turn, reduces the computational requirements of the servo-controller and the requirement to identify the frictional parameters. We recommend a single control microprocessor (such as the Motorola 68000 or 68008) having a 16 or 32 bit internal architecture, eight bit external busses and an instruction set which provides hardware multiplies. We recommend position encoders on each steering axis of the castors and tachometers on all four wheels. All sensors

should have eight bits of precision. We have justified an eight bit implementation in Chapter 7 by demonstrating that such a resolved acceleration servo-controller provides acceptable performance with eight bit I/O precision. This sensing structure allows for the robust sensing of the WMR velocities according to the robust sensing criterion in (A7.15). The actuators may be conventional or brushless DC motors, interfaced to the control processor through a PIA (peripheral interface adaptor) and power FETs (field effect transistors) in an H-bridge configuration [Muir84a] to implement PWM. The position encoders and tachometers are interfaced to the processor through PIAs. The reference trajectory is communicated to the control processor from high level processors or a manual control box through a PIA or dual-ported memory. The control processor also communicates the dead reckoned position (computed in Section 7.2.8), wheel slip results (computed as in Section 7.2.7) and WMR velocities (computed according to the sensed forward velocity solution in (2.4.14)) to the high-level processors or to a human operator through PIAs or dual-ported memory.

The servo-control algorithm, dead reckoning, and wheel slip detection are computed in real-time. The sampling period may be chosen to be one-tenth of the WMR time constant, which may be approximated by the WMR mass divided by the viscous frictional coefficient in air. By substituting the value of the viscous frictional coefficient from Chapters 6 and 7 into this approximation, we may compute the sampling period from the WMR weight; i.e., the sampling period may be computed as 2.3×10^{-4} s/N times the WMR weight in Newtons (or 10^{-3} s/lb times the WMR weight in pounds). Because the sampling period is on the order of tenths of seconds, the resolved acceleration servo-control algorithm, dead reckoning, wheel slip detection and the trajectory planner can be computed each sampling period by the control processor. These real-time algorithms should be stored in an EPROM (electrically programmable read only memory) for non-volatile storage. This completes the design of the servo-controller hardware, software, and firmware for a general-purpose indoor WMR, which need not be adjusted once it is operational. Such a WMR could be mass-produced at low cost for wide industrial and household applications.

We proceed to enumerate our contributions to robotic and control engineering.

8.4 Contributions

Our WMR research contributes the following engineering advances in the areas of robot modeling and control:

(1) Conceptualization of *instantaneously coincident coordinate systems* for the modeling of higher-pair joints;

(2) Introduction of the *wheel Jacobian* to organize the kinematic equations-of-motion of a wheel and enable the systematic solution of the kinematic equations-of-motion of a WMR;

(3) Introduction of a modular, extendible, matrix-vector *dynamic modeling framework* which unifies the modeling of a broad spectrum of robotic mechanisms and incorporates the effects of all forces/torques and frictions of interest;

(4) Conceptualization of *force/torque propagation* which models correctly the physical mechanism by which forces/torques are transmitted through rigid bodies;

(5) Conceptualization of *dry friction at a joint as a coupling phenomenon*, in contrast to the conventional view of joint friction as a force/torque source originating at a joint;

(6) Realization and simulation verification that resolved motion rate and resolved acceleration manipulator *servo-controller methodologies* are directly applicable to WMR servo-control;

(7) Realization and simulation verification that the *sampling period of a WMR servo-controller* may be chosen to be one-tenth the time constant of its main body;

(8) Design and simulation verification of velocity-to-voltage, velocity-to-pulse-width, torque-to-voltage, and torque-to-pulse-width command conversion algorithms which allow the application of conventional

model-based servo-controllers to the control of robots having voltage and PWM actuated motors;

(9) Design and simulation evaluation of a resolved motion rate servo-controller for Uranus which provides adequate performance for general-purpose indoor applications;

(10) Design and simulation evaluation of a resolved acceleration servo-controller for Bicsun-Bicas which provides adequate performance for general-purpose indoor applications; and

(11) Detailed description and technical justification of a WMR/servo-control system (in Section 8.3) which is mechanically simple, inexpensive, allows robust actuation and sensing, is capable of following any spatial path on a hard smooth surface, and if a turret is added, provides onboard manipulators, sensors, or docking instruments with three DOFs.

We have translated these engineering contributions into practical methodologies thereby demonstrating their utility. Even though our dissertation is tailored to WMRs, our modeling and servo-control methodologies may be applied to a broad spectrum of robotic mechanisms. The analysis, design and control of mechanisms having closed-chains, higher-pair joints, unactuated and unsensed joints, friction or pulse-width modulation may benefit from our methodologies.

8.5 Recommendations for Future Work

Our WMR modeling and control research has illuminated several areas for future research activities. The first area follows directly from our modeling activities: kinematic, dynamic and frictional parameter identification for WMRs. There are documented methods of stationary manipulator parameter identification [Khosla85]. Perhaps our approach of extending stationary manipulator methodologies will also apply to parameter identification. If this is indeed the case, adaptive WMR servo-control may become a reality through the application of on-line stationary manipulator parameter identification algorithms. Identification procedures are required for the hardware implementation of our servo-control algorithms. We view WMR parameter identification as the most

significant challenge to WMR control research because the successful implementation of model-based WMR servo-control depends largely on the identification of accurate parameter values.

Manipulators onboard WMRs will likely find future applications in remote and hazardous manipulation tasks. The modeling and coordinated control of an onboard manipulator is thus a promising area of future research. The key realization for resolving onboard manipulation is that the combined wheeled base and onboard manipulator is a single robot and must not be considered as two separate components. Our dynamic modeling methodology, which incorporates the characteristics of wheeled and stationary robotics, may be applied to the combined system. The resolved motion rate and resolved acceleration servo-controller methodologies are also directly applicable.

Extensions to our kinematic and dynamic models abound. Computation of the maximum WMR accelerations and velocities from actuator force/torque limits would provide valuable information for trajectory planning algorithms. Our dynamic modeling framework can be extended to model temperature and humidity effects, and materials effects such as stress, strain, fracture, and wear of robotic components. Dynamic robot models incorporating such phenomena may then be applied to determine the performance of robots in adverse environments and during extended periods of operation. The variable structure of the WMR dynamic model when the WMR does a wheelie or contacts another object can be modeled. Such dynamic models would allow the simulation and control of WMRs in the presence of these realistic conditions. The application of linear systems concepts to model forces/torques as signals, and links and joints as impedances, (introduced in Section 8.1) may be the key to future dynamic analyses of robotic mechanisms.

Our dynamic WMR models can be extended to model the effects of wheel slippage. We have assumed no translational wheel slip. If the sensing structure is robust, we can detect, but cannot compensate for, wheel slip. Perhaps motion during wheel slip can be computed by minimizing the system energy. WMR servo-control during translational wheel slip is a related problem with interesting implications. With conventional servo-control algorithms, as the system errors (e.g., position errors) increase, the command values to the actuators increase to decrease the error. When wheel slip occurs and the WMR

position errors grow, increasing the wheel actuator commands will accentuate the existing slip.

Our dynamics framework may be generalized to model complex closed-chains. This problem may be solved by modeling the dynamics of internally closed-chains independently, thus reducing the complex closed-chain structure to a simple closed-chain which may then be modeled by our framework.

Our graphical representation of dynamic models in Appendix 14 suggests that array processors [Rao88] or neural networks [Lippman87] may be applied to the fast computation of robot dynamics. Our WMR simulations required from 15 to 236 cpu minutes to execute on a time-shared computer to simulate ten seconds of WMR motion: a ratio of 90:1 to 1416:1 actual-to-simulated time. Research with parallel processors would be required to implement this idea in hardware. Fast dynamics computation can improve robot accuracies by reducing the servo-controller sampling period and can bring robot dynamics simulations closer to real-time performance.

Appendix 1

A WMR Nomenclature and Symbolic Representation

A1.1 Overview

In this appendix, we introduce a nomenclature and a symbolic representation for describing the *kinematic* structure of WMRs through the application of essential kinematic information. We define *essential kinematic information* as the minimal information required to solve *symbolically* the kinematic equations-of-motion. WMRs with differing numerical dimensions but identical kinematic structures thus have identical symbolic kinematic solutions. For example, the presense of a steering link is considered essential kinematic information because an equation which relates the velocity of a steered wheel to the velocity of the WMR body must depend upon the steering angle. In contrast, the distance between two wheels is not essential kinematic information because knowing the numerical value of the distance is not required to formulate or solve the symbolic kinematic equations-of-motion. The nomenclature provides a convenient literal and verbal representation of the essential kinematic information. The symbolic representation displays pictorially the essential kinematic relations between the WMR body, wheels and steering links using mnemonic symbols. Our desire to compare the kinematic characteristics of WMRs of differing structures has led to these representations. Without simple, straightforward and informative descriptions of the kinematic structure of a WMR, comparisons between WMRs become confusing and awkward. The conventional pictorial representations are mechanical drawings in which characteristics unessential for kinematic analysis complicate understanding. Similarly, the conventional literal descriptions of WMR kinematics are through lengthy verbal explanations. Our symbolic and literal representations of WMRs characterize the kinematic structure of a WMR through simple diagrams or names.

Our symbolic (naming) representation has been devised to be easily drawn (written or spoken) and interpreted, while providing the following information:

- The number of wheels;
- The type of each wheel;
- The steered wheels;
- The relative positioning of the wheels;
- The actuated DOFs of each wheel; and
- The sensed DOFs of each wheel.

Our symbolic representation can be augmented to include functional dependencies between wheels and to define the distances and angles between components (although these characteristics are not considered essential kinematic information). Although functional dependencies are needed for symbolic solutions, it is difficult to incorporate arbitrary functional relations into our representations. Our definition of essential kinematic information is chosen because our ultimate objective is the control of WMRs; consequently, information required for the forward and inverse kinematic solutions is directly applicable to WMR control. For this reason, we specify the DOFs of each wheel which are actuated and sensed. The motion of an unactuated (non-sensed) wheel DOF may constrain the motion of the WMR, whereas the motion of an actuated (sensed) wheel DOF may be computed symbolically from the motion of the WMR body. *Understanding these representations may most easily be accomplished by scanning the rules delineated in Sections A1.2 and A1.3 and then following the examples in Section A1.4.* The reader can then refer back to the rules for a more detailed understanding.

A1.2 Symbolic Representation Rules

The rules for generating and interpreting WMR diagrams follow.

(1) A WMR is depicted by a large circle.

(2) Each *wheel* appears as a small circle within the WMR circle.

(3) Each *steering axis* is portrayed as circle smaller than the associated wheel; a steering link is drawn as a line segment from the steering axis to the respective wheel. If the steering axis intersects the center of the respective wheel, it is depicted as a small circle within and concentric to the wheel circle, and a steering link is not drawn.

(4) The relative positions of the wheel circles (for non-steered wheels) and steering axes (for steered wheels) correspond to the relative positions of the wheels and steering axes on the WMR.

(5) The DOFs of a wheel are indicated by line segments and arcs within the wheel circle drawn in the directions of the translational and rotational DOFs, respectively. The rotational slip DOF of a wheel is implied and no arc is drawn. A conventional wheel has one radial line segment in the direction of travel from the wheel center to the wheel circle. Similarly, an omnidirectional wheel has two radial line segments, and a ball wheel has two radial line segments and an arc (one quarter of a circle) drawn within the wheel circle.

(6) The *actuated* DOFs of each wheel are drawn with an arrowhead appended to the line indicating the wheel DOF.

(7) The *sensed* DOFs of each wheel are drawn with a "T" appended to the line indicting the wheel DOF. A wheel DOF, which is both actuated and sensed, is indicated by a closed arrow (i.e., the combination of a "T" and an arrow).

(8) Functional dependencies between DOFs within or between wheels may be indicated by dashed lines. Dashed lines may also be used to indicate that a component of a WMR cannot be described adequately by our representation (Optional).

A1.3 Nomenclature Rules

Our nomenclature expresses the identical information as the symbolic representation in Section A1.2. For compactness, we limit the amount of positional, actuation and sensing information in the name of the WMR. The rules for creating and interpreting WMR names follow.

(1) The name of the kinematic structure of a **wheeled mobile robot** ends with the suffix *-whemor*. This suffix may be omitted when it is understood that the name is that of a WMR.

(2) *Sets* of one or more wheels of the same functional type are indicated by syllables separated by hyphens.

(3) Two or more wheels of a WMR are of the same functional type if they are of the same wheel type (i.e., conventional, omnidirectional, or ball); are all steered or all not-steered; are all actuated and sensed similarly; and are all placed symmetrically with respect to either the center of the WMR, a line through the WMR center (the major axis), or a line perpendicular to the major axis (the minor axis).

(4) The syllables are ordered from the beginning to the end of the name according to the following *precedence* characteristics which are listed from the most to the least important:

- Symmetry with respect to the robot center;
- Symmetry with respect to the major axis;
- Symmetry with respect to the minor axis;
- Number of wheels;
- Steered wheels;
- Ball wheels;
- Omnidirectional wheels;
- Conventional wheels;
- Actuated wheels; and
- Sensed wheels.

For example, all wheel sets which are symmetric with respect to the WMR center appear first; and if there is more than one wheel set which is symmetric with respect to the WMR center, the set having the largest number of wheels (if there is not a tie) is listed first in the name.

(5) Each syllable representing a set of wheels consists of:

(i) One of the prefixes *uni*, *bi*, *tri*, *tetra*, *penta*, *hexa*, *hepta*, *octa*, *ennea*, *deca*, or *poly* to indicate the number of wheels in the set;

(ii) Followed by one of the letters *c*, *o*, *b*, or *w* to indicate that they are either a conventional, omnidirectional, ball or an unspecified wheel type. For an omnidirectional wheel, the final vowel of the prefix is dropped before adding *o* to make the name pronounceable;

(iii) Followed by an *s*, if the wheels are steered;

(iv) Followed by either an *a* or *u* to indicate that the wheels are actuated or unactuated, respectively. A wheel having more than one DOF and/or a steering axis is considered actuated if the steering angle or any of the DOFs is actuated;

(v) Followed by either an *s* or *n* to indicate that the wheels are sensed or not-sensed, respectively. A wheel having more than one DOF and/or a steering axis is considered sensed if the steering angle or any of the DOFs is sensed.

(6) A kinematic structure of a WMR, which cannot be named adequately according to these rules, is named by prefixing the name which most closely indicates the structure with *pseudo-*.

A1.4 Examples

In Appendix 2, we illustrate the kinematic diagram and name of *twelve* WMRs. The predominant WMR kinematic structure documented in the literature has two parallel conventional wheels, one on each side of the robot (thus, the syllable *bicas*). These WMRs also possess one or two castors for stability. Among the most widely known examples are Shakey [Nilsson84] and Newt [Hollis77] (in Figure A2.1). Shakey has two free-wheeling casters for stability (*bicsun*), whereas Newt utilizes only one (*unicsun*). By mounting the two driven wheels at an acute angle to the floor in their Topo [Helmers83b] WMR, the Androbot Company stabilized the WMR without the use of castors. Even though the acute angle of the wheels cannot be represented in either the symbolic representation or the name, we can infer that the wheels must be angled for stability by assuming the most practical realization. Mobile robots which possess multiple non-steered, driven wheels whose axes are non-colinear must rely on wheel slip if the robot is to navigate turns. Such is the case with the Terregator [Wallace85] (in Figure A2.2) which uses six parallel, non-steered, conventional wheels, three on either side (*hexacas*). The mechanically more complex, steered and driven conventional wheel is utilized on Neptune [Podnar84] (in Figure A2.4), which has a tricycle wheel arrangement; the front

wheel is steered and driven (*unicsas*), while the two rear wheels are at a fixed parallel orientation and are undriven (*bicun*). Pluto [Moravec83] (in Figure A2.4) has three steered and driven wheels (*tricsas*).

The Stanford Cart [Moravec80] (in Figure A2.7) has two steered, undriven wheels in the front (*bicsan*) and two fixed, driven wheels in the back (*bican*). The two steered wheels are coupled to be oriented in the same direction, thus the *pseudo* prefix. The JPL Rover [Lewis73] (in Figure A2.7) is similar to the Stanford Cart except that both the front and back wheel pairs have coupled steering (*pseudo-bicsan-bicsan-whemor*). Kludge [Holland83] has complex functional dependencies between the wheels. This WMR has three conventional wheels that are both steered and driven, as on Pluto. In addition, a chain and gear arrangement is used to equalize all drive velocities and steering angles. To complicate further the arrangement, each wheel is mounted on an actuated link which can be pivoted towards or away from the center of the WMR to adjust its stability properties (*pseudo-tricsas*). Dashed lines are used in the symbolic representation of Kludge to indicate the functional dependencies between steering angles and wheel actuation, and the inability to represent the pivoted link. The hybrid spider drive [Holland83] (in Figure A2.8) utilizes four conventional wheels, two on either side of the robot, each of which is mounted at the end of a three DOF leg linkage (*pseudo-tetracsas*). The hybrid locomotion vehicle [Ichikawa83] (in Figure A2.8) utilizes six steered and driven conventional wheels, each at the end of an actuated vertical leg (*pseudo-hexacsas*). Uranus [Moravec86] (in Figure A2.10) utilizes four omnidirectional wheels positioned at the corners of a rectangular wheelbase (*tetroas*). The Unimation Robot [Carlisle83] (in Figure A2.10) possesses three DOFs using only three actuators and three omnidirectional wheels (*troas*). The most maneuverable wheel is a ball which is actuated to possess three DOFs [Moravec82] (*unibas*).

A class of kinematic structures which may consist of a large number of WMRs is specified by the *poly* prefix. For example, a *polycas-whemor* refers to the class of WMRs which has only conventional non-steered wheels arranged symmetrically with respect to the robot center or its major axis. Similarly a class of WMRs which has a number of wheels whose type is not specified is called *polywas-whemor*. Also, if the actuation and sensing characteristics are not

important for the discussion, the actuation and sensing labels may be omitted, as in *polyw-whemor*. Admittedly, our nomenclature has disadvantages. Names created by these rules may not be easily pronounceable. There is not a one-to-one relationship between WMRs and the names created by our nomenclature. There are examples of WMRs which have several legal names (e.g., wheel sets can always be divided into multiple sets, each having fewer wheels). Furthermore, it is not always possible to determine the symmetry of a WMR from its name (e.g., a *hexac-whemor* may be symmetric with respect to the robot center or the major axis). These disadvantages are the result of our attempt to assign compact names. Most ambiguities in the nomenclature can be eliminated by assuming the practical alternative. For example, a *tric-whemor* must be symmetric with respect to the WMR center and not the major axis, because it would be more practical to name the latter a *bic-unic-whemor.*

Finally, we note that our nomenclature can be extended to other classes of mobile robots. For example, Legged Mobile Robots (LMRs) may be denoted by the suffix *lemor*, and Treaded Mobile Robots (TMRs) may be denoted by the suffix *tremor.*

Appendix 2

Survey of WMRs

In this appendix, we survey the kinematic configurations of existing WMRs. We are interested in determining the types of wheels utilized and the relative placement of the wheels on WMRs. Documentation of WMRs is scattered throughout the robotics, artificial intelligence, control engineering, scientific, industrial, popular and hobbiest literature [Bell85, Croix85, Fischetti85, Johnson84, Schachter83]. We examine documented WMRs to understand the requirements of kinematic and dynamic methodologies for this class of mobile robots.

In Appendix 1, we have introduced a nomenclature and a pictorial representation for describing the kinematic structure of WMRs. The diagramming conventions provide a convenient tool for describing and comparing kinematic structures of WMRs. We apply these rules to develop symbolic representations and kinematic names for the WMRs presented in this survey and refer to these representations as we describe each WMR.

The most common kinematic arrangement of mobile robots documented in the literature has two diametrically opposed wheels (i.e., two parallel conventional wheels, one on each side of the robot). These WMRs also possess one or two castors for stability. Among the most widely known examples are: Shakey [Nilsson84] and Newt [Hollis77] (in Figure A2.1), Jason [Smith73], Hilare [Giralt79], Yamabiko [Kanayama81, Iijima81], ROBART II [Everett85], and RB5X [Marrs85]. The commercially-available Labmate WMR [Labmate87] possesses four castors. By mounting the two driven wheels at an acute angle to the floor in their Topo [Helmers83b] WMR, the Androbot Company stabilized the WMR without the use of castors.

Mobile robots which possess multiple non-steered, driven wheels whose axes are non-colinear must rely on translational wheel slip if the robot is to navigate turns. Such is the case with the RDS Prowler [Rogers84] and the Terregator [Wallace85] (in Figure A2.2), both of which use six parallel, non-steered, conventional wheels, three on each side. Similarly, Gemini

[Helmers85] (in Figure A2.2) utilizes two synchronously driven wheels on each side.

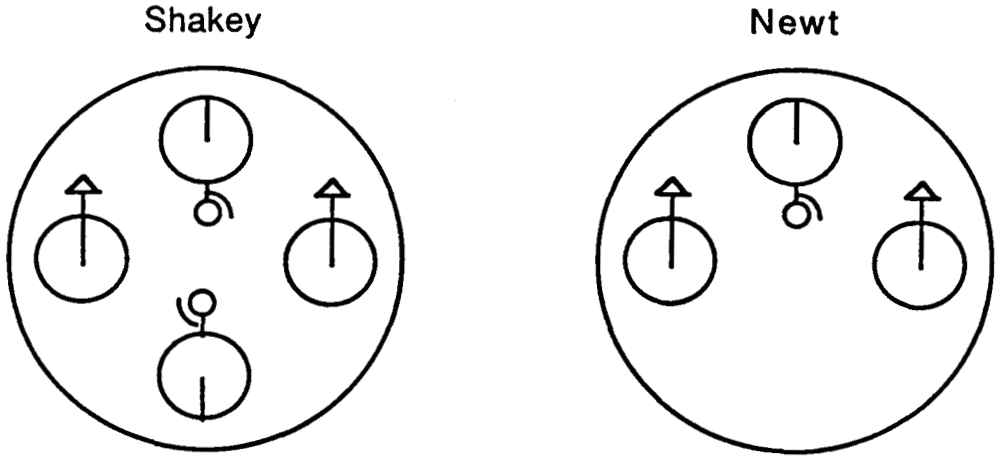


Figure A2.1: Symbolic Representations of Shakey and Newt

The mechanically more complex, steered and driven conventional wheel is utilized on Neptune [Podnar84] (pictured in Figure A2.3 and symbolized in Figure A2.4), Hero-1 [Helmers83a] and Avatar [Balmer82]. These three robots have a tricycle wheel arrangement; the front wheel is steered and driven, while the two rear wheels are at a fixed parallel orientation and are undriven.

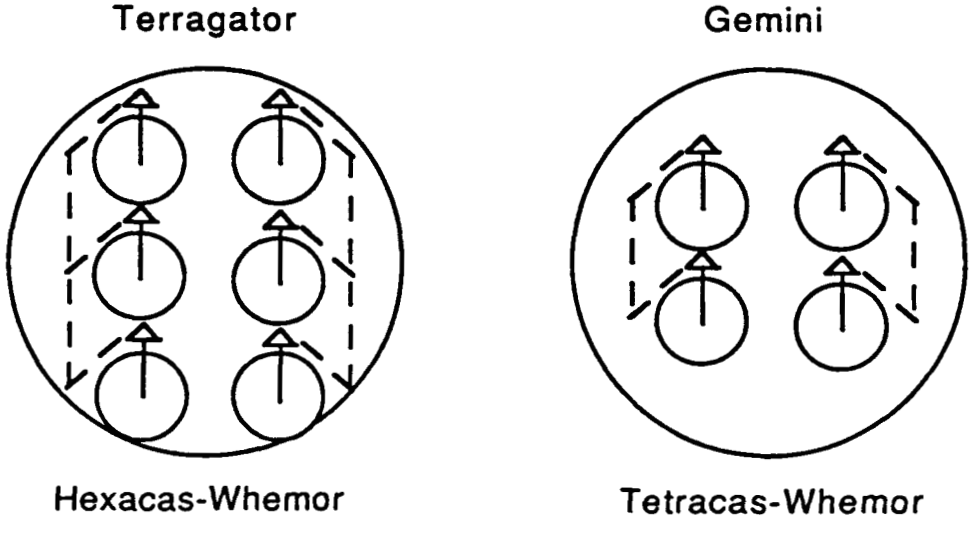


Figure A2.2: Symbolic Representations of Terregator and Gemini

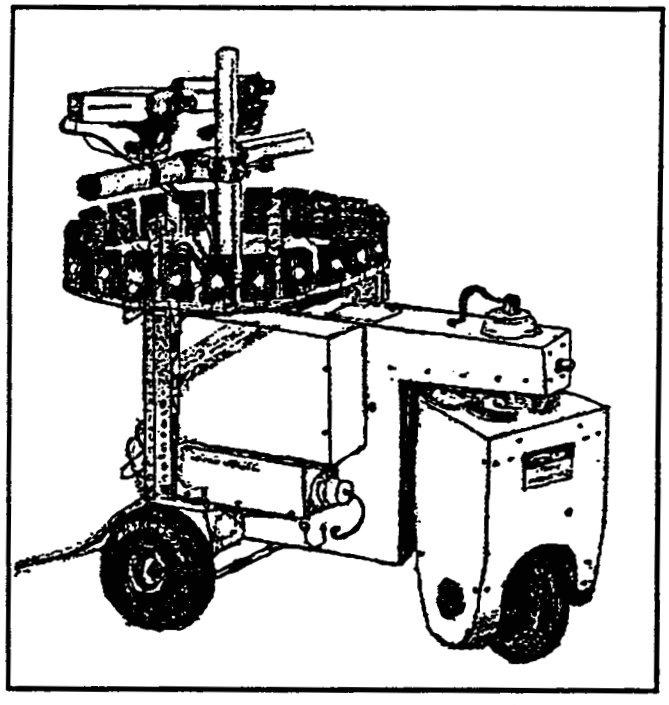


Figure A2.3: Neptune[1]
The Mobile Robot Laboratory's Second WMR (1984)

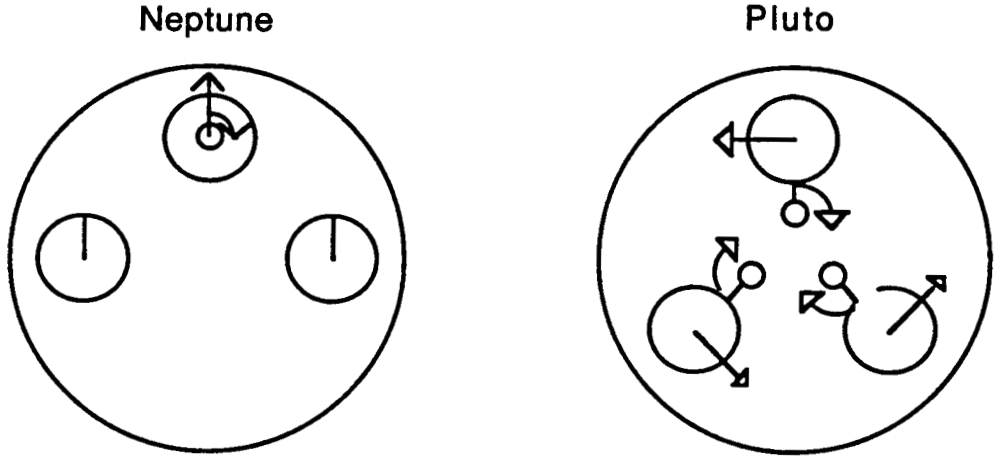


Figure A2.4: Symbolic Representations of Neptune and Pluto

Pluto [Moravec83] (pictured in Figure A2.5 and symbolized in Figure A2.4), also known as the CMU Rover, has three steered and driven wheels. The Stanford Cart [Moravec80] (pictured in Figure A2.6 and symbolized in Figure

---

1 Figures A2.3, A2.5, A2.6, and A2.9 are reprinted from [Moravec88a] by the author's permission.

A2.7) has two steered, undriven wheels in the front and two fixed, driven wheels in the rear. The two front wheels are coupled by an Ackerman steering linkage. An Ackerman steering linkage [McGraw82] approximately ensures the correct wheel angles to avoid wheel slip. Both the front and back wheels of the JPL Rover [Lewis73] (in Figure A2.7) are coupled by Ackerman steering linkages, and all four wheels are driven independently.

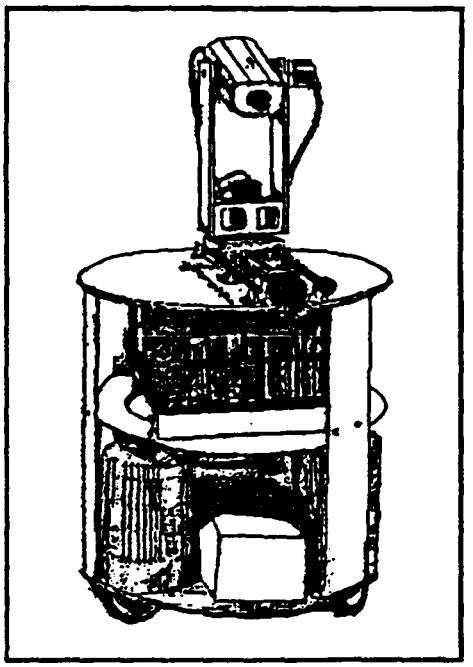


Figure A2.5: Pluto
The Mobile Robot Laboratory's First WMR (1983).
(Now on display in the Boston Computer Museum.)

Kludge [Holland85] is an example of a WMR with complex functional dependencies between the wheels. This robot has three conventional wheels that are both steered and driven. A chain and gear arrangement is used to equalize all drive velocities and steering angles (Synchro-Drive). To complicate further the arrangement, each wheel is mounted on an actuated link which can be pivoted towards or away from the center of the WMR for stability. Kludge's successor K2A [Holland85] embodies the synchro-drive mechanism using concentric shafts instead of chains and does not have any actuated links. The Denning Sentry robot [Wilson85] also utilizes a three-wheel synchronous drive and steer system.

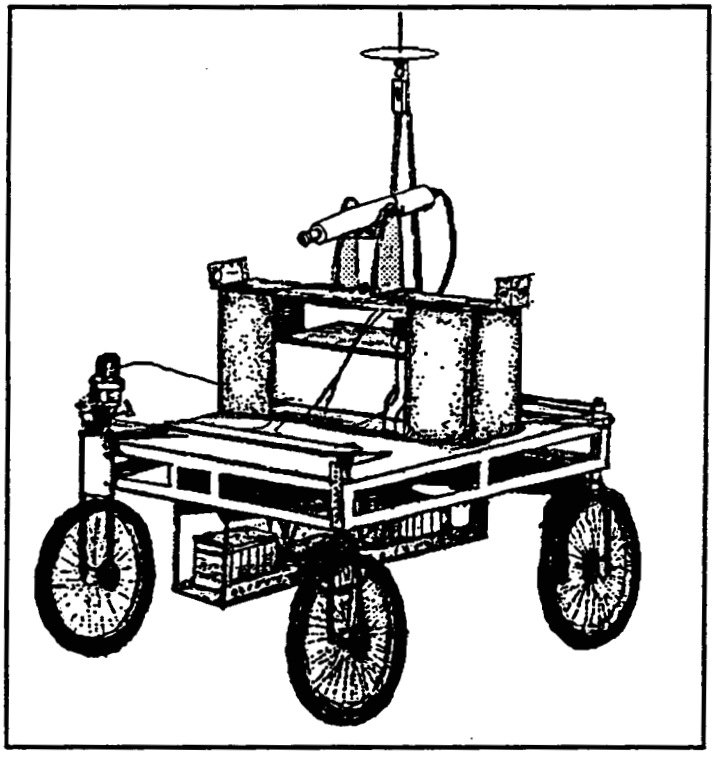


Figure A2.6: The Stanford Cart

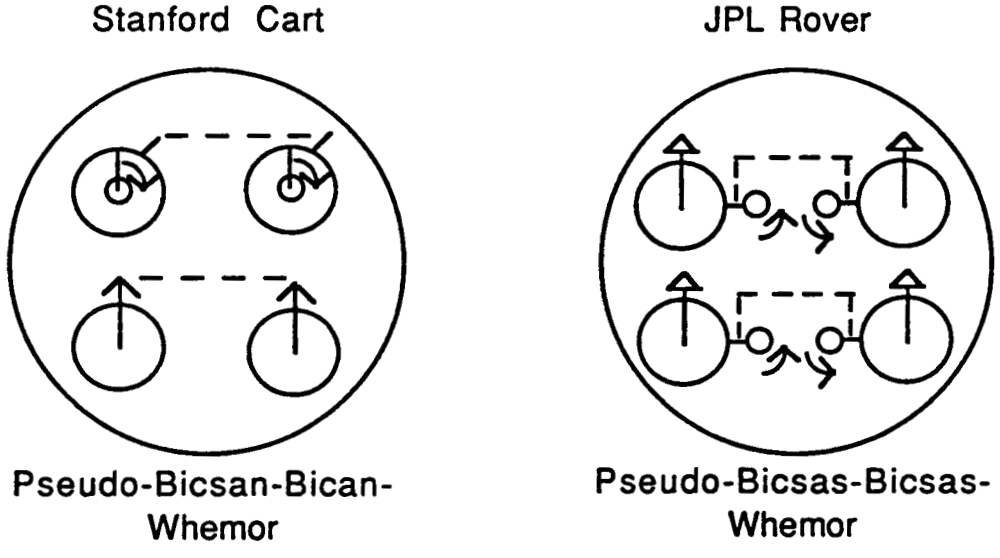
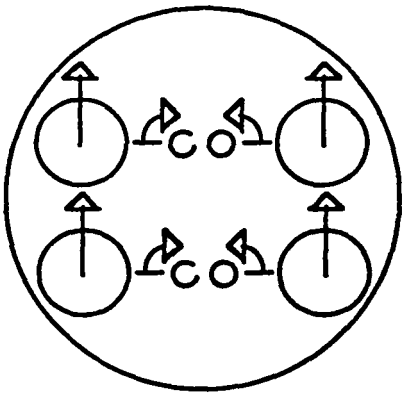


Figure A2.7: Symbolic Representations of the Stanford Cart and the JPL Rover

The hybrid spider drive [Holland83] (in Figure A2.8) utilizes four conventional wheels, two on either side of the robot, each of which is mounted at the end of a three DOF leg linkage. The hybrid locomotion vehicle

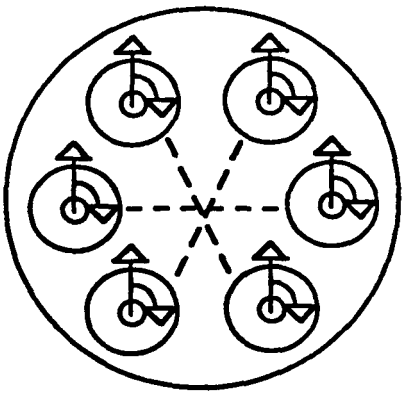
[Ichikawa83] (in Figure A2.8) utilizes six steered and driven conventional wheels, each at the end of an actuated vertical leg.

Hybrid Spider Drive



Pseudo-Tetracsas-Whemor

Hybrid Locomotion Vehicle



Pseudo-Haxacsas-Whemor

Figure A2.8: Symbolic Representations of the Hybrid Spider Drive and the Hybrid Locomotion Vehicle

Equally obscure is the triangle wheel step climber [Whitaker62], which possesses four sets of three wheels mounted at the vertices of equilateral triangles. When a wheel encounters a step, the triangle pivots about its center and the WMR reaches the top of the step by rolling on a different set of wheels.

The recent application of *omnidirectional* wheels (in Appendix 3) has led to novel mobile kinematic configurations. Omnidirectional wheels have been used for powered wheelchairs (e.g., Omni drive [Holland83] and Wheelon [Alvema82]) and ambulatory drive platforms [Wier81]. The later orients the omnidirectional wheels at an acute angle to the floor for stability. Uranus [Moravec86] (pictured in A2.9 and symbolized in Figure A2.10) has a rectangular wheel base with four omnidirectional wheels having rollers at 45° angles to the wheel orientation. The Unimation robot [Carlisle83] (in Figure A2.10) and Fetal1 [Johnson84] have triangular wheel bases and three omnidirectional wheels with rollers at 90° angles to the wheel orientation.

Omnidirectional treads [Bonmartini56, Bradbury80] operate as omnidirectional wheels with the rollers mounted upon tank-like treads. A *ball* wheel (in Appendix 3) is the most maneuverable wheel allowing three DOF motion [Moravec82, Brown77, Jones48]. The first design of Jason [Smith73]

incorporated three ball wheel castors which were later replaced by a single conventional castor. We are unaware of any other documented applications of ball wheels on WMRs.

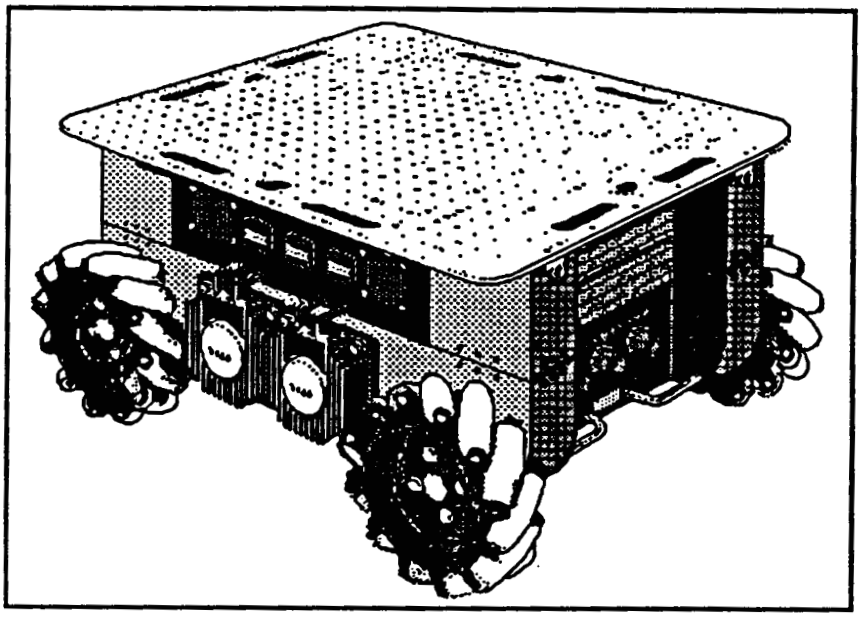


Figure A2.9: Uranus
The Mobile Robot Laboratory's Third WMR (1987)

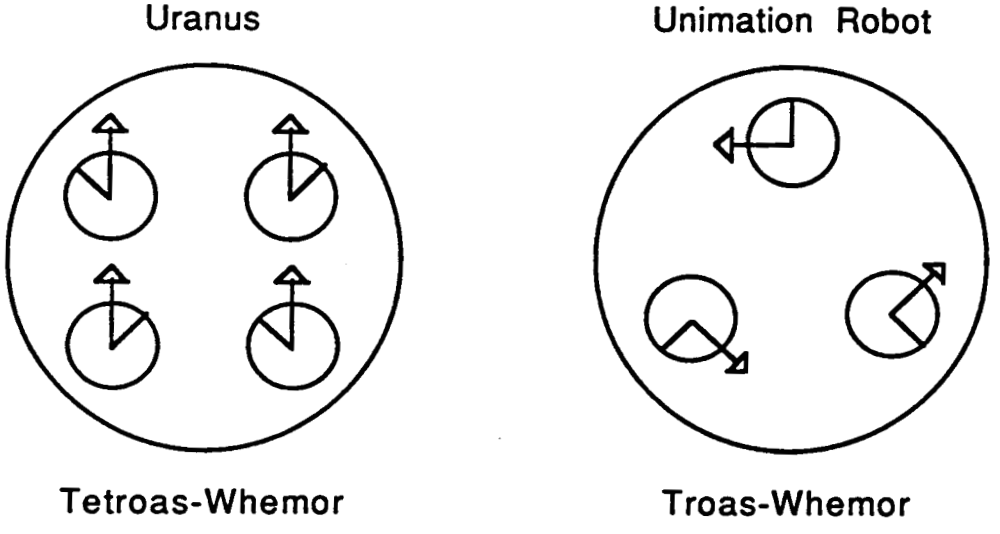


Figure A2.10: Symbolic Representations of Uranus and the Unimation Robot

In Chapter 2, we formulate a kinematics methodology incorporating the variability in the numbers and types of wheels and actuating mechanisms which is evident in this survey.

Appendix 3

Wheels and Wheel Jacobians

A3.1 Overview

Three wheel types are used in WMR designs: conventional, omnidirectional, and ball wheels. In addition, conventional wheels are often mounted on a steering link to provide an additional DOF. Schematic views of these three wheels are shown in Figures A3.1 and A3.2. Figure A3.1 illustrates the side view of the wheels and indicates the wheel velocities. Figure A3.2 illustrates the top view of the wheels and indicates the wheel pseudo-velocities introduced in Chapter 2.

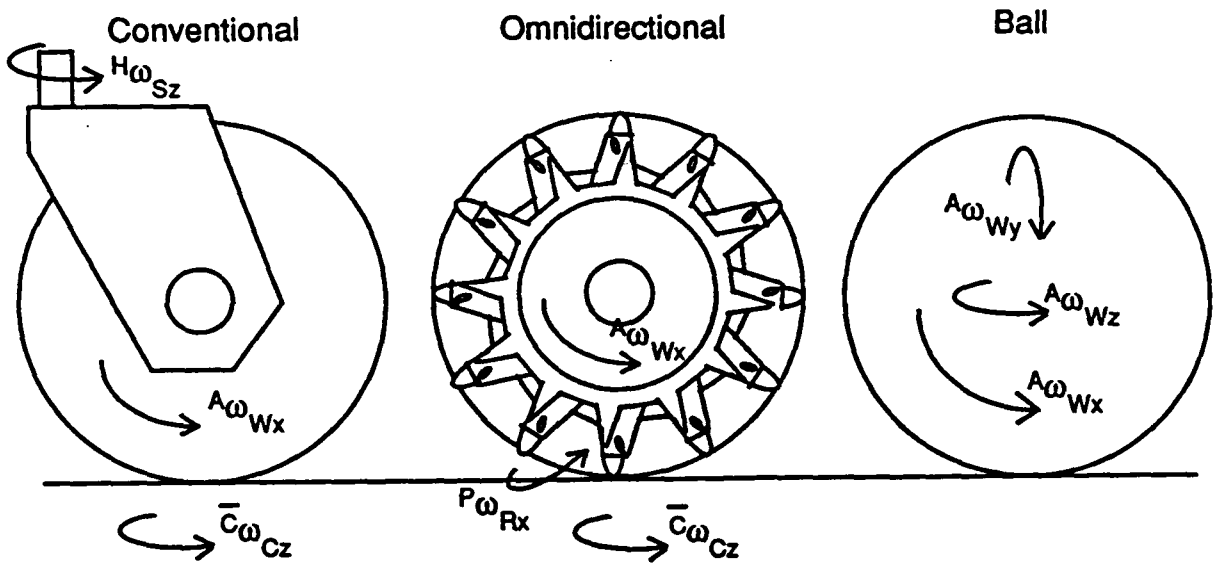


Figure A3.1: Side View Showing Actual Wheel Velocities

In this appendix, we develop the wheel Jacobian matrices for conventional wheels, steered conventional wheels, omnidirectional wheels and ball wheels. The wheel Jacobian matrix (as introduced in Section 2.3.8) relates the velocities of the WMR to the velocities of the wheel. The wheel Jacobian matrix $\mathbf{J}$ in (2.3.27) is the product of the pseudo-Jacobian matrix $\hat{\mathbf{J}}$ and the wheel matrix $\mathbf{W}$. The pseudo-Jacobian matrix in (2.3.25) relates the wheel pseudo-velocities $\dot{\hat{\mathbf{q}}}$ to the robot velocities $^{\bar{B}}\dot{\mathbf{p}}_{B}$, and the wheel matrix relates the

actual wheel velocities $\dot{q}$ to the wheel pseudo-velocities. The pseudo-velocities are the velocities ${}^{\bar{C}}v_{Cx}$, ${}^{\bar{C}}v_{Cy}$, ${}^{\bar{C}}\omega_{Cz}$, and ${}^{H}\omega_{Sz}$ indicated in Figure A3.2 and the actual wheel velocities are the angular velocities of the wheel, rollers, and steering link ${}^{A}\omega_{Wx}$, ${}^{A}\omega_{Wy}$, ${}^{A}\omega_{Wz}$, ${}^{P}\omega_{Rx}$, and ${}^{H}\omega_{Sz}$ about their respective axes indicated in Figures A3.1. With these observations, the wheel matrix for each wheel is written directly from the wheel geometry. The wheel Jacobian matrix is then computed by multiplying the pseudo-Jacobian matrix by the wheel matrix. We consider each of the aforementioned wheels in turn.

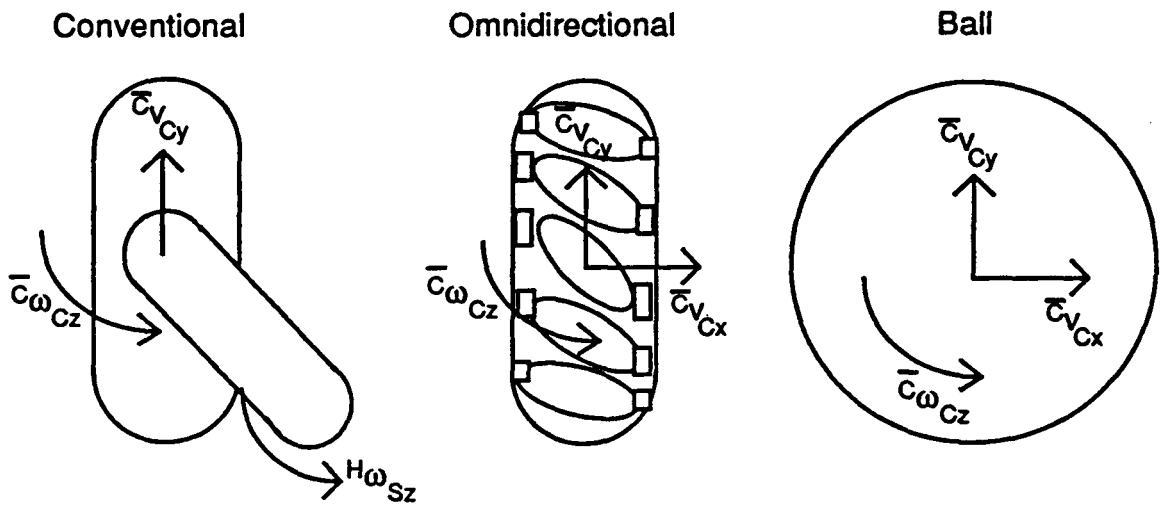


Figure A3.2: Top View Showing Wheel Pseudo-Velocities

A3.2 Conventional Non-Steered Wheel

The *conventional non-steered* wheel having two DOFs is the simplest to construct. It allows travel upon a surface in the direction of the wheel orientation, and rotation about the point-of-contact between the wheel and the floor. We note that the rotational DOF of an actual conventional wheel is slippage, since there is an area-of-contact rather than a single point. Even though we define the rotational slip as a DOF, we do not consider slip transverse to the wheel orientation a DOF, because transverse wheel slip is more difficult to model. The conventional wheel is the most widely used wheel; automobiles, roller skates and bicycles utilize this wheel.

The two DOFs of the conventional non-steered wheel correspond to the two wheel pseudo-velocities ${}^{\bar{C}}v_{Cy}$, and ${}^{\bar{C}}\omega_{Cz}$. The actual wheel velocities are the

angular velocity of the wheel about its axle ${}^{A}\omega_{Wx}$ and the angular velocity of the rotational slip ${}^{\bar{C}}\omega_{Cz}$. These velocities are related by the (4 × 2) wheel matrix **W** in (A3.1):

$$\dot{\mathbf{q}} = \begin{pmatrix} {}^{C}v_{Cx} \\ {}^{\bar{C}}v_{Cy} \\ {}^{\bar{C}}\omega_{Cz} \\ {}^{H}\omega_{Sz} \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ R & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} {}^{A}\omega_{Wx} \\ {}^{\bar{C}}\omega_{Cz} \end{pmatrix} = \mathbf{W}\,\dot{\mathbf{q}} \qquad \text{(A3.1)}$$

In (A3.1), R is the wheel radius.

The (4×2) wheel matrix **W** is premultiplied by the (3×4) pseudo-Jacobian matrix in (2.3.25) to form the (3×2) Jacobian matrix:

Conventional Non-Steered Wheel Jacobian

$$\mathbf{J} = \begin{pmatrix} -R\sin{}^{B}\theta_{Cz} & {}^{B}d_{Cy} \\ R\cos{}^{B}\theta_{Cz} & -{}^{B}d_{Cx} \\ 0 & 1 \end{pmatrix}. \qquad \text{(A3.2)}$$

The conventional non-steered wheel is termed *degenerate* because the Jacobian is not square. Even though the WMR body velocities can always be computed from the wheel velocities, the wheel velocities can be computed from the WMR body velocities only when the WMR is capable of the desired motion.

A3.3 Conventional Steered Wheel

The conventional steered wheel has an additional DOF provided by the steering joint corresponding to the pseudo-velocity ${}^{H}\omega_{Sz}$ (shown in Figure A3.2). The actual steering velocity ${}^{H}\omega_{Sz}$ (shown in Figure A3.1) is equal to the steering pseudo-velocity. The (4×3) wheel matrix **W** and the (3×3) wheel Jacobian matrix **J** are, respectively:

$$\dot{\mathbf{q}} = \begin{pmatrix} {}^{\bar{C}}v_{Cx} \\ {}^{\bar{C}}v_{Cy} \\ {}^{\bar{C}}\omega_{Cz} \\ {}^{H}\omega_{Sz} \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ R & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} {}^{A}\omega_{Wx} \\ {}^{\bar{C}}\omega_{Cz} \\ {}^{H}\omega_{Sz} \end{pmatrix} = \mathbf{W}\,\dot{\mathbf{q}} \qquad \text{(A3.3)}$$

and

Conventional Steered Wheel Jacobian

$$\mathbf{J} = \begin{pmatrix} -R\sin{}^{B}\theta_{Cz} & {}^{B}d_{Cy} & {}^{B}d_{Hy} \\ R\cos{}^{B}\theta_{Cz} & -{}^{B}d_{Cx} & {}^{B}d_{Hx} \\ 0 & 1 & -1 \end{pmatrix}. \qquad \text{(A3.4)}$$

The conventional steered wheel Jacobian matrix is *non-singular* if its determinant is nonzero; i.e., if

$$det(\mathbf{J}) = R\,({}^{S}d_{Cy}\cos{}^{S}\theta_{Cz} - {}^{S}d_{Cx}\sin{}^{S}\theta_{Cz}) \neq 0\,. \qquad \text{(A3.5)}$$

The wheel Jacobian is singular and the conventional steered wheel is degenerate if the steering axis intercepts the wheel point-of-contact (i.e., if ${}^{S}d_{Cx}={}^{S}d_{Cy}=0$) or if the wheel is oriented perpendicularly to the steering link (i.e., if ${}^{S}d_{Cx}\sin{}^{S}\theta_{Cz}={}^{S}d_{Cy}\cos{}^{S}\theta_{Cz}$).

A3.4 Omnidirectional Wheel

The *omnidirectional* wheel [Grabowiecki19, Blumrich74, Ilon75] has three DOFs. One DOF is in the direction of the wheel orientation. The second DOF is provided by motion of rollers mounted around the periphery of the wheel hub. In principle, the roller axles can be mounted at any nonzero angle η with respect to the wheel orientation. The omnidirectional wheel in Figures A3.1 and A3.2 has a roller angle of 45° [Ilon75]. The third DOF is rotational slip about the point-of-contact. It is possible, but not common, to actuate the rollers of an omnidirectional wheel [Holland83] with a complex driving arrangement. When sketching WMRs having omnidirectional wheels, the rollers on the underside of the wheel (i.e., those touching the surface-of-travel) are drawn and not the

rollers which are actually visable from a top view, to facilitate the kinematic analysis.

The omnidirectional wheel DOFs correspond to the pseudo-velocities ${}^{\bar{C}}v_{Cx}$, ${}^{\bar{C}}v_{Cy}$, and ${}^{\bar{C}}\omega_{Cz}$ that are linear combinations of the actual wheel velocities ${}^{A}\omega_{Wx}$, ${}^{P}\omega_{Rx}$, and ${}^{\bar{C}}\omega_{Cz}$:

$$\dot{\hat{q}} = \begin{pmatrix} {}^{\bar{C}}v_{Cx} \\ {}^{\bar{C}}v_{Cy} \\ {}^{\bar{C}}\omega_{Cz} \\ {}^{H}\omega_{Sz} \end{pmatrix} = \begin{pmatrix} 0 & r\sin(\eta) & 0 \\ R & -r\cos(\eta) & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} {}^{A}\omega_{Wx} \\ {}^{P}\omega_{Rx} \\ {}^{\bar{C}}\omega_{Cz} \end{pmatrix} = W\dot{q} \qquad \text{(A3.6)}$$

where R is the wheel radius and r is the roller radius.

The omnidirectional wheel Jacobian matrix is:

Omnidirectional Wheel Jacobian

$$J = \begin{pmatrix} -R\sin{}^{B}\theta_{Cz} & r\sin({}^{B}\theta_{Cz}+\eta) & {}^{B}d_{Cy} \\ R\cos{}^{B}\theta_{Cz} & -r\cos({}^{B}\theta_{Cz}+\eta) & -{}^{B}d_{Cx} \\ 0 & 0 & 1 \end{pmatrix}. \qquad \text{(A3.7)}$$

The omnidirectional wheel is singular; i.e., the determinant of the omnidirectional wheel Jacobian $-R_w R_r \sin(\eta)$ is zero when the rollers are aligned with the wheel (i.e., whenever $\eta=0$).

A3.5 Ball Wheel

The most maneuverable wheel is a *ball* which possesses three DOFs corresponding to rotations about the three normal axes positioned at the wheel center ${}^{\bar{C}}v_{Cx}$, ${}^{\bar{C}}v_{Cy}$, and ${}^{\bar{C}}\omega_{Cz}$. The wheel matrix relating the actual wheel velocities ${}^{A}\omega_{Wx}$, ${}^{A}\omega_{Wy}$, and ${}^{\bar{C}}\omega_{Cz}$ to the pseudo-velocities is:

$$\dot{q} = \begin{pmatrix} {}^{\bar{C}}v_{Cx} \\ {}^{\bar{C}}v_{Cy} \\ {}^{\bar{C}}\omega_{Cz} \\ {}^{H}\omega_{Sz} \end{pmatrix} = \begin{pmatrix} 0 & R & 0 \\ R & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} {}^{A}\omega_{Wx} \\ {}^{A}\omega_{Wy} \\ {}^{\bar{C}}\omega_{Cz} \end{pmatrix} = W\dot{q}. \qquad \text{(A3.8)}$$

The ball wheel Jacobian matrix is:

Ball Wheel Jacobian

$$J = \begin{pmatrix} -R\sin{}^{B}\theta_{Cz} & R\cos{}^{B}\theta_{Cz} & {}^{B}d_{Cy} \\ R\cos{}^{B}\theta_{Cz} & R\sin{}^{B}\theta_{Cz} & -{}^{B}d_{Cx} \\ 0 & 0 & 1 \end{pmatrix}. \qquad \text{(A3.9)}$$

Since the determinant of the ball wheel Jacobian matrix is R^2, the Jacobian matrix in (A3.9) is non-singular for all non-zero wheel radii.

Schemes have been devised for actuating and sensing ball wheels [Moravec82], but we are unaware of any existing implementations. An omnidirectional wheel which is steered about its point-of-contact is kinematically equivalent to a ball wheel, and may be a practical design alternative.

A3.6 Concluding Remarks

In Chapters 6 and 7 and Appendix 11, we apply the wheel Jacobian matrices developed in this appendix to obtain the kinematic equations-of-motions of *eight* WMRs.

Appendix 4

Solution of Ax=By

In this appendix, we extend the standard criteria [Cohn58] for the solution of the system of linear algebraic equations $\mathbf{A}\,\mathbf{x} = \mathbf{b}$, where $\mathbf{A}$ is an $(m \times n)$ matrix, $\mathbf{x}$ is a $(n \times 1)$ vector and $\mathbf{b}$ is a $(m \times 1)$ vector, to the solution of the system of linear algebraic equations

$$\mathbf{A}\,\mathbf{x} = \mathbf{B}\,\mathbf{y} \tag{A4.1}$$

where $\mathbf{B}$ is an $(m \times p)$ matrix and $\mathbf{y}$ is a $(p \times 1)$ vector. Since the composite robot equation (2.4.3) has the form of (A4.1), solutions of (A4.1) are directly applicable to the solution of the composite robot equation.

We apply the method of least-squares [Cohn58] to compute the vector $\mathbf{x}$ for *overdetermined* (i.e., having fewer unknowns than independent equations) and *determined* (i.e., having the same number of unknowns as independent equations) systems of linear algebraic equations:

$$\mathbf{x} = (\mathbf{A}^T\mathbf{A})^{-1}\,\mathbf{A}^T\mathbf{B}\,\mathbf{y}\,. \tag{A4.2}$$

The necessary condition for applying the least-squares solution in (A4.2) is that rank($\mathbf{A}$)=n. There is no unique solution for *undetermined* systems (i.e., systems having fewer independent equations than unknowns).

The residual error of the least-squares method is:

$$\mathbf{A}\,\mathbf{x} - \mathbf{B}\,\mathbf{y} = [\mathbf{A}\,(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T - \mathbf{I}]\,\mathbf{B}\,\mathbf{y} = \Delta(\mathbf{A})\,\mathbf{B}\,\mathbf{y}\,. \tag{A4.3}$$

where, for expository convienience, we define the *Delta* matrix function $\Delta(\mathbf{U})$ as:

$$\Delta(\mathbf{U}) = \begin{cases} -\mathbf{I} & \text{for } \mathbf{U} = \text{null} \\ \mathbf{U}\,(\mathbf{U}^T\mathbf{U})^{-1}\,\mathbf{U}^T - \mathbf{I} & \text{otherwise} \end{cases} \tag{A4.4}$$

where the argument $\mathbf{U}$ is a $(c \times d)$ matrix of rank d.

To characterize WMR motion (in Appendix 7), we must determine whether the least-squares error in (A4.3) is zero for *all* **y**. To do so, we may apply either of the following equivalent tests:

$$\Delta(\mathbf{A})\ \mathbf{B} = \mathbf{0} \tag{A4.5}$$

or

$$\text{rank}[\mathbf{A};\mathbf{B}] = \text{rank}[\mathbf{A}]\ . \tag{A4.6}$$

If either test in (A4.5) or (A4.6) is satisfied, the least-squares error is zero for all **y**. The first test in (A4.5) is apparent from the expression for the least-squares error in (A4.3). The second test in (A4.6) states that if the columns of the matrix **B** lie in the vector space spanned by the columns of the matrix **A**, then the vector **By** must also lie in the vector space spanned by the columns of **A** for all **y**. The vector **By** can then be expressed as a linear combination of the columns of **A** by proper choice (via the least-squares solution) of **x**. Similarly, we may determine whether the least-squares error is zero for a *specific* **y** by applying either of the following two equivalent tests:

$$\Delta(\mathbf{A})\ \mathbf{B}\ \mathbf{y} = \mathbf{0} \tag{A4.7}$$

or

$$\text{rank}[\mathbf{A};\mathbf{By}] = \text{rank}[\mathbf{A}]\ . \tag{A4.8}$$

We depict in Figure A4.1 a *tree* illustrating the nature of all possible solutions for the vector **x** of the system of linear algebraic equations in (A4.1). The tree branches (directed arrows) indicate tests on **A**, **B** and **y** and are numbered for future reference. The leaves (boxes) indicate the corresponding properties of the solution.

As depicted in Figure A4.1, the system of linear algebraic equations in (A4.1) may be determined, overdetermined or undetermined. The top branches, (0) and (1), determine whether the least-squares solution is applicable by testing the rank of the matrix **A**. If the rank of **A** is n (branch (0)), the least-squares solution is applicable and there is a unique solution for some **y**. If the rank of **A** is less than n (branch (1)), the least-squares solution is not applicable indicating that the system is *undetermined* and there is no unique solution for any **y**. An undetermined system has more unknowns than independent equations.

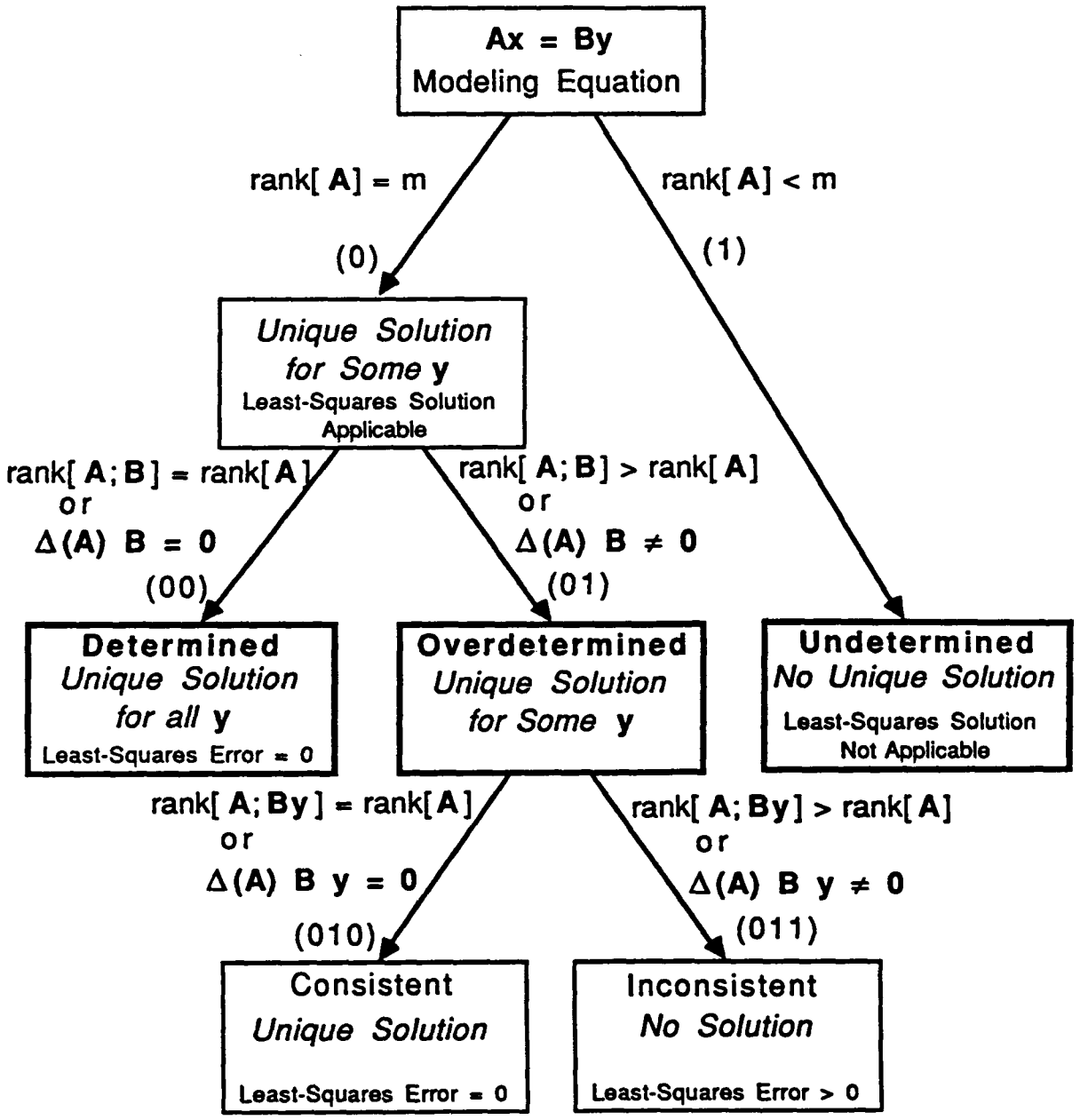


Figure A4.1: The Solution Tree for the Vector x

In a *determined* system, the number of independent equations (less than or equal to m) equals the number of unknowns (n). The least-squares error is zero for all **y** and thus tests (A4.3) and (A4.4) apply at branch (00).

In an *overdetermined* system, the number of independent equations exceeds the number of unknowns. The least-squares error of an overdetermined system is thus non-zero for *some* **y** (branch (01)). Tests (A4.7)

and (A4.8) are applied at branch (010) to determine whether the least-squares error is zero for a specific **y**. If so, the system is consistent and there is a unique solution. If the least-squares error is non-zero for a specific **y** (branch (011)), the system is inconsistent and there is no exact solution. In Appendix 7, we apply the *solution tree* in Figure A4.1 to the composite robot equation in (2.4.3) to characterize WMR mobility.

Appendix 5

Actuated Inverse Velocity Solution Computations

In this appendix, we detail the matrix manipulations leading to the *actuated inverse velocity solution* in Section 2.4.2. We solve the composite partitioned robot equation in (2.4.6)

$$\mathbf{A}_0\dot{\mathbf{p}} = \begin{pmatrix} \mathbf{I}_1 \\ \mathbf{I}_2 \\ \vdots \\ \mathbf{I}_N \end{pmatrix} \dot{\mathbf{p}} = \begin{pmatrix} \mathbf{J}_{1a} & 0 & \dots & 0 & \mathbf{J}_{1u} & 0 & \dots & 0 \\ 0 & \mathbf{J}_{2a} & . & \vdots & 0 & \mathbf{J}_{2u} & . & \vdots \\ \vdots & . & . & 0 & \vdots & . & . & 0 \\ 0 & \dots & 0 & \mathbf{J}_{Na} & 0 & \dots & 0 & \mathbf{J}_{Nu} \end{pmatrix} \begin{pmatrix} \dot{\mathbf{q}}_{1a} \\ \dot{\mathbf{q}}_{2a} \\ \vdots \\ \dot{\mathbf{q}}_{Na} \\ \dot{\mathbf{q}}_{1u} \\ \dot{\mathbf{q}}_{2u} \\ \vdots \\ \dot{\mathbf{q}}_{Nu} \end{pmatrix} = \mathbf{B}_{0p} \begin{pmatrix} \dot{\mathbf{q}}_a \\ \dot{\mathbf{q}}_u \end{pmatrix} \tag{A5.1}$$

to compute the actuated wheel velocities $\dot{\mathbf{q}}_a$ in the least-squares solution in (2.4.7):

$$\begin{pmatrix} \dot{\mathbf{q}}_a \\ \dot{\mathbf{q}}_u \end{pmatrix} = (\mathbf{B}_{0p}^T \mathbf{B}_{0p})^{-1} \mathbf{B}_{0p}^T \mathbf{A}_0 \dot{\mathbf{p}}. \tag{A5.2}$$

We begin by forming the matrix product:

$$(\mathbf{B}_{0p}^T \mathbf{B}_{0p}) = \begin{pmatrix} \mathbf{D}_{aa} & \mathbf{D}_{au} \\ \mathbf{D}_{au}^T & \mathbf{D}_{uu} \end{pmatrix} \tag{A5.3}$$

where

$$\mathbf{D}_{aa} = \begin{pmatrix} \mathbf{J}_{1a}^T\mathbf{J}_{1a} & 0 & \dots & 0 \\ 0 & \mathbf{J}_{2a}^T\mathbf{J}_{2a} & . & \vdots \\ 0 & . & . & 0 \\ 0 & \dots & 0 & \mathbf{J}_{Na}^T\mathbf{J}_{Na} \end{pmatrix}, \quad \mathbf{D}_{au} = \begin{pmatrix} \mathbf{J}_{1a}^T\mathbf{J}_{1u} & 0 & \dots & 0 \\ 0 & \mathbf{J}_{2a}^T\mathbf{J}_{2u} & . & \vdots \\ 0 & . & . & 0 \\ 0 & \dots & 0 & \mathbf{J}_{Na}^T\mathbf{J}_{Nu} \end{pmatrix},$$

and

$$\mathbf{D}_{uu} = \begin{pmatrix} \mathbf{J}_{1u}^T\mathbf{J}_{1u} & 0 & \dots & 0 \\ 0 & \mathbf{J}_{2u}^T\mathbf{J}_{2u} & . & \vdots \\ 0 & . & . & 0 \\ 0 & \dots & 0 & \mathbf{J}_{Nu}^T\mathbf{J}_{Nu} \end{pmatrix}.$$

To compute the actuated wheel velocities, we have written the matrix $(\mathbf{B}_{0p}^T\,\mathbf{B}_{0p})$ in block form with four components, each one a block diagonal matrix. We let the block matrix $\mathbf{X}$ be the inverse of the matrix in (A5.3). To compute the block components of the matrix inverse in terms of the block components of the matrix in (A5.3), we apply the fact that the inverse of a matrix times the matrix itself is the identity matrix; i.e.,

$$\begin{pmatrix} \mathbf{X}_{11} & \mathbf{X}_{12} \\ \mathbf{X}_{21} & \mathbf{X}_{22} \end{pmatrix} \begin{pmatrix} \mathbf{D}_{aa} & \mathbf{D}_{au} \\ \mathbf{D}_{au}^T & \mathbf{D}_{uu} \end{pmatrix} = \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \tag{A5.4}$$

Since we seek only the upper (actuated) components of the wheel velocity vector $\dot{\mathbf{q}}_a$ in (A5.1), we compute only the two components in the top row of the block matrix inverse. We thus separate the solution for the actuated wheel velocities

$$\dot{\mathbf{q}}_a = [\mathbf{X}_{11}\,\mathbf{X}_{12}]\,\mathbf{B}_{0p}^T\,\mathbf{A}_0\,\dot{\mathbf{p}} \tag{A5.5}$$

from the solution for the unactuated ones. We expand (A5.4) to obtain

$$\mathbf{X}_{11}\,\mathbf{D}_{aa} + \mathbf{X}_{12}\,\mathbf{D}_{au}^T = \mathbf{I} \tag{A5.6}$$

and

$$\mathbf{X}_{11}\,\mathbf{D}_{au} + \mathbf{X}_{12}\,\mathbf{D}_{uu} = \mathbf{0}\,. \tag{A5.7}$$

From (A5.6) and (A5.7), we find

$$\mathbf{X}_{12} = -\mathbf{X}_{11}\,\mathbf{D}_{au}\,\mathbf{D}_{uu}^{-1} \tag{A5.8}$$

and

$$\mathbf{X}_{11} = (\mathbf{D}_{aa} - \mathbf{D}_{au}\,\mathbf{D}_{uu}^{-1}\,\mathbf{D}_{au}^{T})^{-1} \tag{A5.9}$$

where

$$\mathbf{D}_{uu}^{-1} = \begin{pmatrix} (\mathbf{J}_{1u}^{T}\mathbf{J}_{1u})^{-1} & 0 & \cdots & 0 \\ 0 & (\mathbf{J}_{2u}^{T}\mathbf{J}_{2u})^{-1} & . & \vdots \\ 0 & . & . & 0 \\ 0 & \cdots & 0 & (\mathbf{J}_{Nu}^{T}\mathbf{J}_{Nu})^{-1} \end{pmatrix}. \tag{A5.10}$$

The block diagonal matrix $\mathbf{X}_{11}$ in (A5.9) is

$$\mathbf{X}_{11} = \begin{pmatrix} -[\mathbf{J}_{1a}^{T}\Delta(\mathbf{J}_{1u})\mathbf{J}_{1a}]^{-1} & 0 & \cdots & 0 \\ 0 & -[\mathbf{J}_{2a}^{T}\Delta(\mathbf{J}_{2u})\mathbf{J}_{2a}]^{-1} & . & \vdots \\ 0 & . & . & 0 \\ 0 & \cdots & 0 & -[\mathbf{J}_{Na}^{T}\Delta(\mathbf{J}_{Nu})\mathbf{J}_{Na}]^{-1} \end{pmatrix} \tag{A5.11}$$

and the block diagonal matrix $\mathbf{X}_{12}$ in (A5.8) is

$$\mathbf{X}_{12} = \begin{pmatrix} \Lambda_1 & 0 & \cdots & 0 \\ 0 & \Lambda_2 & . & \vdots \\ 0 & . & . & 0 \\ 0 & \cdots & 0 & \Lambda_N \end{pmatrix} \tag{A5.12}$$

where

$$\Lambda_i = [\mathbf{J}_{ia}^{T}\,\Delta(\mathbf{J}_{iu})\,\mathbf{J}_{ia}]^{-1}\,\mathbf{J}_{ia}^{T}\,\mathbf{J}_{iu}\,(\mathbf{J}_{iu}^{T}\,\mathbf{J}_{iu}^{T})^{-1}\,. \tag{A5.13}$$

We substitute (A5.12) and (A5.11) into (A5.5) to obtain the actuated wheel velocity vector

$$\dot{q}_a = \begin{pmatrix} [J_{1a}^T \Delta(J_{1u}) J_{1a}]^{-1} J_{1a}^T \Delta(J_{1u}) \\ [J_{2a}^T \Delta(J_{2u}) J_{2a}]^{-1} J_{2a}^T \Delta(J_{2u}) \\ \vdots \\ [J_{Na}^T \Delta(J_{Nu}) J_{Na}]^{-1} J_{Na}^T \Delta(J_{Nu}) \end{pmatrix} \dot{p}. \qquad \text{(A5.14)}$$

Equation (A5.14) is the least-squares solution for the actuated wheel velocity vector. We note that this solution is applicable only when the matrix in (A5.3) is invertible. The conditions under which this solution is applicable are specified by the soluble motion criterion in (A7.1). In Chapter 4, we apply the actuated inverse velocity solution in (A5.14) to design WMR resolved motion rate servo-controllers.

Appendix 6

Sensed Forward Velocity Solution Computations

In this appendix, we detail the matrix manipulations leading to the least-squares *sensed forward velocity solution*. We solve the partitioned robot sensing equation in (2.4.11)

$$\mathbf{A}_n \begin{pmatrix} \dot{\mathbf{p}} \\ \dot{\mathbf{q}}_n \end{pmatrix} = \begin{pmatrix} \mathbf{I}_1 & -\mathbf{J}_{1n} & 0 & \dots & 0 \\ \mathbf{I}_2 & 0 & -\mathbf{J}_{2n} & . & : \\ : & : & . & . & 0 \\ \mathbf{I}_N & 0 & \dots & 0 & -\mathbf{J}_{Nn} \end{pmatrix} \begin{pmatrix} \dot{\mathbf{p}} \\ \dot{\mathbf{q}}_{1n} \\ \dot{\mathbf{q}}_{2n} \\ : \\ \dot{\mathbf{q}}_{Nn} \end{pmatrix}$$

$$= \begin{pmatrix} \mathbf{J}_{1s} & 0 & \dots & 0 \\ : & \mathbf{J}_{2s} & . & 0 \\ : & . & . & 0 \\ 0 & \dots & 0 & \mathbf{J}_{Ns} \end{pmatrix} \begin{pmatrix} \dot{\mathbf{q}}_{1s} \\ \dot{\mathbf{q}}_{2s} \\ : \\ \dot{\mathbf{q}}_{Ns} \end{pmatrix} = \mathbf{B}_s \dot{\mathbf{q}}_s \qquad \text{(A6.1)}$$

to compute the WMR body velocities $\dot{\mathbf{p}}$ in the least-squares solution in (2.4.13):

$$\begin{pmatrix} \dot{\mathbf{p}} \\ \dot{\mathbf{q}}_n \end{pmatrix} = (\mathbf{A}_n^T \mathbf{A}_n)^{-1} \mathbf{A}_n^T \mathbf{B}_s \dot{\mathbf{q}}_s \,. \qquad \text{(A6.2)}$$

We begin by forming the matrix product

$$(\mathbf{A}_n^T \mathbf{A}_n) = \begin{pmatrix} N\mathbf{I} & -\mathbf{J}_{1n} & -\mathbf{J}_{2n} & \dots & -\mathbf{J}_{Nn} \\ -\mathbf{J}_{1n}^T & \mathbf{J}_{1n}^T\mathbf{J}_{1n} & 0 & \dots & 0 \\ -\mathbf{J}_{2n}^T & 0 & \mathbf{J}_{2n}^T\mathbf{J}_{2n} & . & : \\ : & : & . & . & 0 \\ -\mathbf{J}_{Nn}^T & 0 & \dots & 0 & \mathbf{J}_{Nn}^T\mathbf{J}_{Nn} \end{pmatrix} = \begin{pmatrix} N\mathbf{I} & \mathbf{T} \\ \mathbf{T}^T & \mathbf{D} \end{pmatrix} \qquad \text{(A6.3)}$$

where N is the number of wheels and $\mathbf{I}$ is the (3×3) identity matrix. We let the block matrix $\mathbf{X}$ be the inverse of the symmetric matrix $(\mathbf{A}_n^T \mathbf{A}_n)$ in (A6.3). Since the inverse of a matrix times the matrix is the identity matrix,

$$\begin{pmatrix} \mathbf{X}_{11} & \mathbf{X}_{12} \\ \mathbf{X}_{21} & \mathbf{X}_{22} \end{pmatrix} \begin{pmatrix} N\mathbf{I} & \mathbf{T} \\ \mathbf{T}^T & \mathbf{D} \end{pmatrix} = \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{pmatrix}. \tag{A6.4}$$

We use the top block row of the matrix inverse to separate the WMR body velocity vector $\dot{\mathbf{p}}$ from the non-sensed wheel velocity vector $\dot{\mathbf{q}}_n$:

$$\dot{\mathbf{p}} = [\mathbf{X}_{11} \;\; \mathbf{X}_{12}]^{-1} \mathbf{A}_n^T \mathbf{B}_s \dot{\mathbf{q}}_s \,. \tag{A6.5}$$

From (A6.4), we obtain

$$\mathbf{X}_{11} N\mathbf{I} + \mathbf{X}_{12} \mathbf{T}^T = \mathbf{I} \tag{A6.6}$$

and

$$\mathbf{X}_{11} \mathbf{T} + \mathbf{X}_{12} \mathbf{D} = \mathbf{0} \,, \tag{A6.7}$$

from which

$$\mathbf{X}_{12} = -\mathbf{X}_{11} \mathbf{T} \mathbf{D}^{-1} \tag{A6.8}$$

and

$$\mathbf{X}_{11} = (N\mathbf{I} - \mathbf{T} \mathbf{D}^{-1} \mathbf{T}^T)^{-1} \,. \tag{A6.9}$$

The inverse of the block diagonal matrix $\mathbf{D}$ is:

$$\mathbf{D}^{-1} = \begin{pmatrix} (\mathbf{J}_{1n}^T \mathbf{J}_{1n})^{-1} & 0 & \cdots & 0 \\ 0 & (\mathbf{J}_{2n}^T \mathbf{J}_{2n})^{-1} & . & \vdots \\ 0 & . & . & 0 \\ 0 & \cdots & 0 & (\mathbf{J}_{Nn}^T \mathbf{J}_{Nn})^{-1} \end{pmatrix}. \tag{A6.10}$$

We expand the block elements in (A6.8) and (A6.9) to obtain

$$\mathbf{X}_{12} = -\mathbf{X}_{11} [-\mathbf{J}_{1n}(\mathbf{J}_{1n}^T \mathbf{J}_{1n})^{-1} -\mathbf{J}_{2n}(\mathbf{J}_{2n}^T \mathbf{J}_{2n})^{-1} \cdots -\mathbf{J}_{Nn}(\mathbf{J}_{Nn}^T \mathbf{J}_{Nn})^{-1}] \tag{A6.11}$$

where

$$X_{11} = [N\mathbf{I} - J_{1n}(J_{1n}^T J_{1n})^{-1} J_{1n}^T - J_{2n}(J_{2n}^T J_{2n})^{-1} J_{2n}^T - \ldots - J_{Nn}(J_{Nn}^T J_{Nn})^{-1} J_{Nn}^T]^{-1}$$

$$= -[\Delta(J_{1n}) + \Delta(J_{2n}) + \ldots + \Delta(J_{Nn})]^{-1} . \qquad \text{(A6.12)}$$

Finally, we substitute (A6.11) and (A6.12) into (A6.5) to obtain the least-squares solution for the WMR body velocity vector:

$$\dot{p} = -X_{11} \; [\Delta(J_{1n})J_{1s} \quad \Delta(J_{2n})J_{2s} \quad \ldots \quad \Delta(J_{Nn})J_{Ns}] \; \dot{q}_s . \qquad \text{(A6.13)}$$

In Appendix 7, we develop the adequate sensing criterion which ensures the invertibility of the matrix $(A_n^T A_n)$ in (A6.3) and thereby the applicability of the least-squares sensed forward velocity solution in (A6.13). In Chapter 4, we apply the sensed forward velocity solution in (A6.13) to design resolved motion rate and resolved acceleration WMR servo-controllers.

Appendix 7

WMR Mobility Characterization

A7.1 Overview

In this appendix, we illuminate WMR mobility, actuation and sensing characteristics. We apply the results of Appendix 4 to develop in Section A7.2 a *mobility characterization tree* which enables us to interpret properties of the composite robot equation in terms of the mobility characteristics of the WMR. The mobility characterization tree indicates whether the mobility structure is *determined*, *overdetermined* or *undetermined*, and associates specific mobility characteristics with each possibility. For example, we may apply the mobility characterization tree to determine whether a WMR allows three DOF motion, and if it does not, the tree indicates the motion constraints.

The actuated inverse velocity solution in (2.4.8) does not guarantee that the specified WMR body velocities will be attained when the actuated wheel variables are driven at the computed velocities. In Section A7.3, we investigate the possible WMR motions when the actuated wheel variables attain the velocities computed by the actuated inverse solution. We develop an *actuation characterization tree*, analogous to the mobility characterization tree, which enables us to determine the actuation structure (determined, overdetermined or undetermined) of a WMR. The actuation characterization tree is applicable for WMR design to avoid overdetermined actuation (which may cause actuator conflict) and undetermined actuation (which allows the WMR uncontrollable DOFs). From our analysis, we can determine whether the actuated wheel variables are sufficient for producing all of the motions allowed by the mobility structure.

In Section A7.4, we develop a *sensing characterization tree* which enables us to determine the character (undetermined, determined or overdetermined) of the WMR sensing. We thus can determine whether the sensed wheel variables are sufficient for discerning all of the motions allowed by the mobility structure. Finally, in Section A7.5, we summarize our development.

A7.2 Mobility Characterization

The composite robot equation in (2.4.3) has the form of the system of linear algebraic equations in Figure A4.1, in which $\mathbf{A}_0$, $\mathbf{B}_0$, $\dot{\mathbf{p}}$, and $\dot{\mathbf{q}}$ play the roles of **A**, **B**, **x** and **y**, respectively. Since the WMR body velocity vector $\dot{\mathbf{p}}$ plays the role of the dependent variable, we investigate the conditions under which the forward velocity solution may be computed. In Figure A7.1, we apply the solution tree in Figure A4.1 to the composite robot equation in (2.4.3).

By inspection of (2.4.2), we observe that the rank of the $(3N \times 3)$ matrix $\mathbf{A}_0$ is 3 and thus branch (0) always applies. Since branch (1) does not apply, the solution cannot be undetermined; and hence the WMR motion is completely specified by the motion of the wheels. From the structure of the matrices $\mathbf{A}_0$ and $\mathbf{B}_0$ in (2.4.2), we observe that the rank of the augmented matrix $[\mathbf{A}_0,\mathbf{B}_0]$ is greater than 3 when there is more than one wheel. A WMR with one wheel is determined (branch (00)), and a WMR with more than one wheel is overdetermined (branch (01)). The overdetermined nature of WMRs having more than one wheel is a consequence of the closed-chain kinematic structure of a WMR. As indicated in Figure A7.1, the composite robot equation in (2.4.3) will be consistent (and have a solution at branch (010)) or inconsistent (and have no solution at branch (011)) depending upon the wheel velocity vector $\dot{\mathbf{q}}$. Our no-translational slip assumption (in Section 2.3) ensures that the motions of the wheels and the WMR are consistent and that there is thus a solution.

We depict in Figure A7.1 the forward solution for the WMR body velocity vector $\dot{\mathbf{p}}$ from the complete wheel velocity vector $\dot{\mathbf{q}}$. Although the forward solution of the composite robot equation provides little physical insight, we gain significant understanding of WMR motion by investigating the nature of the inverse solution. For WMR control it is not necessary to compute all of the wheel velocities in the inverse solution since they are not all actuated. Because of the closed-chains, moreover, not all of the wheel variables must be actuated. In Section 2.4.2, we compute the actuated inverse velocity solution for the actuated wheel variables. In the remainder of this section, we focus on the

complete inverse solution to gain physical insight into WMR mobility characteristics.

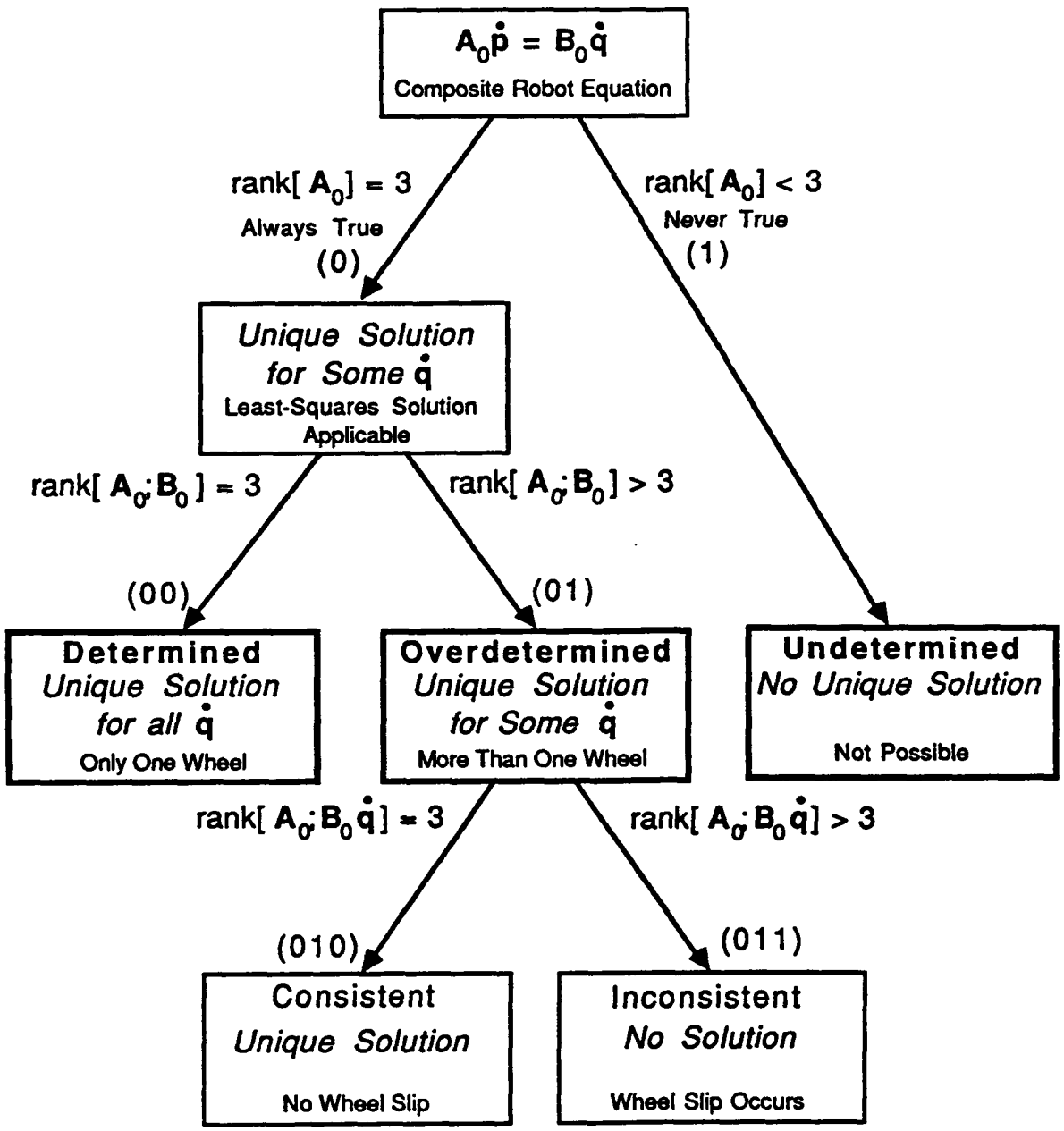


Figure A7.1: The Solution Tree for the WMR Body Velocity Vector $\dot{p}$

We investigate the inverse solution by interchanging the roles of the right and left-hand sides of the composite robot equation in (2.4.3) and applying the solution tree in Figure A4.1. Thereby, B_0, A_0, $\dot{q}$ and $\dot{p}$ in (2.4.3) play the roles of **A**, **B**, **x** and **y** in (A4.1), respectively. The solution tree for the inverse solution,

subsequently referred to as the *mobility characterization tree*, is depicted in Figure A7.2 . The branch tests indicated within curly brackets "{...}" are simplified tests which apply if there are no couplings between wheels.

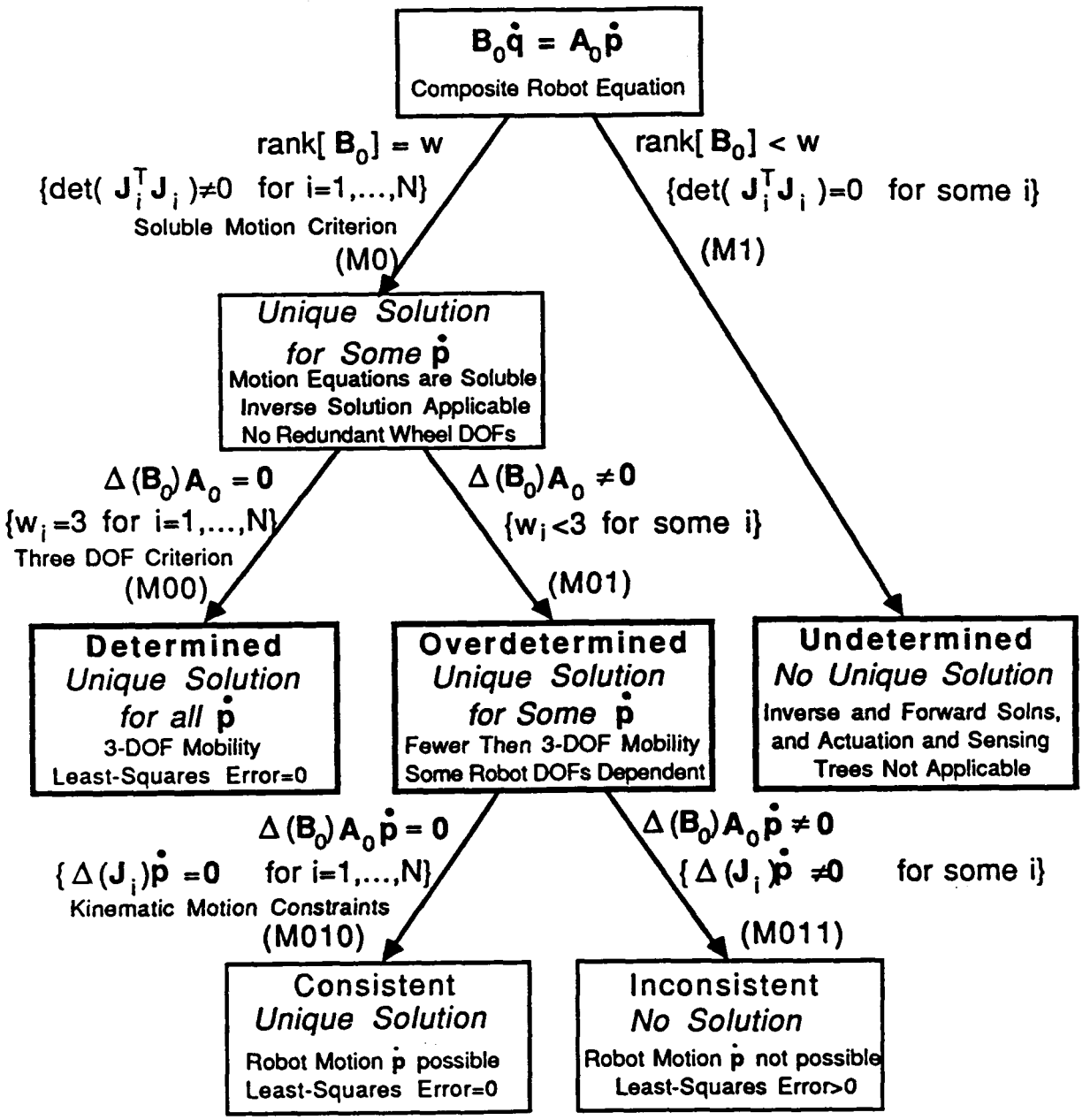


Figure A7.2: The Mobility Characterization Tree

The inverse solution can be determined, undetermined or overdetermined depending upon the kinematics (i.e., $\mathbf{B}_0$ and $\mathbf{A}_0$). The top branches test the rank of the (3N × w) matrix $\mathbf{B}_0$ against the total number of

wheel velocities w. Since the rank of $\mathbf{B}_0$ is the sum of the ranks of all of the wheel Jacobian matrices when there are no wheel couplings, we test the rank of each wheel Jacobian matrix $\mathbf{J}_i$ against the number of wheel variables w_i for all wheels i=1,...,N. The rank of the $(3 \times w_i)$ wheel Jacobian matrix $\mathbf{J}_i$ is w_i if the determinant of the matrix $[\mathbf{J}_i^T\mathbf{J}_i]$ is non-zero as indicated by the non-redundant wheel criterion in (2.3.28). We refer to branch test (M0) as the *soluble motion criterion* because it determines whether the composite robot equation can be solved.

Soluble Motion Criterion

$$\text{rank}\,[\mathbf{B}] = w \qquad \text{(A7.1)}$$

Soluble Motion Criterion With No Wheel Couplings

$$\det\,[\mathbf{J}_i^T\mathbf{J}_i] \neq 0 \quad \text{for } i = 1,...,N$$

If the determinant of the matrix $[\mathbf{J}_i^T\mathbf{J}_i]$ is zero, the associated wheel is redundant. *A WMR having redundant wheels and no wheel couplings is undetermined.* We cannot compute the inverse velocity solution for a WMR with redundant wheels. Since the inverse solution is utilized in WMR servo-control (in Chapter 4), we suggest that undetermined mobility structures (i.e., redundant wheels) be avoided.

WMRs without redundant wheels allow some WMR body motions since there is a unique solution to the system of linear algebraic equations in (2.4.3) for some $\dot{\mathbf{p}}$. Branches (M00) and (M01) test the rank of the augmented matrix $[\mathbf{B}_0;\mathbf{A}_0]$ against the rank of $\mathbf{B}_0$. From their structure in (2.4.2), the ranks of these two matrices are equal when all of the wheel Jacobian matrices are (3×3) and rank 3 (i.e., all of the wheels are non-redundant and possess three DOFs). The mobility structure of a WMR is therefore determined if the test at branch (M00) succeeds. A determined structure has a unique solution for all $\dot{\mathbf{p}}$; i.e., for any desired three dimensional WMR body velocity vector $\dot{\mathbf{p}}$ there is a wheel velocity vector $\dot{\mathbf{q}}$ which is consistent with the motion. We thus conclude: *The kinematic design of a WMR allows three DOF motion if and only if all of the wheels*

possess three DOFs. This requirement is expressed computationally in the *three DOF motion criterion* in (A7.2).

Three DOF Motion Criterion

$$\text{rank}\,[\mathbf{B}_0] = w \text{ and } \Delta(\mathbf{B}_0)\,\mathbf{A}_0 = \mathbf{0} \qquad \text{(A7.2)}$$

Three DOF Motion Criterion With No Wheel Couplings

$$\det\,[\mathbf{J}_i^T\mathbf{J}_i] \neq 0 \text{ and } w_i = 3 \text{ for } i = 1,\ldots,N$$

If branch (M0) succeeds and the WMR does not possess three DOFs, the solution is overdetermined (branch (M01)). The WMR does not allow some motions because some of the WMR body DOFs are dependent. For example, a WMR with a non-steered conventional wheel which satisfies branch (M0) must have an overdetermined mobility structure because no motions perpendicular to the wheel orientation may occur without slip. Branches (M010) and (M011) indicate the possible WMR body motions $\dot{\mathbf{p}}$ without translational wheel slip. If the least-squares error $\Delta(\mathbf{B}_0)\mathbf{A}_0\dot{\mathbf{p}}$ in (A4.3) is zero, the solution is consistent, and the motion may occur. We thus determine the kinematic constraints on the WMR body motion by equating the least-squares error in (A7.3) to zero. By examining the structure of the error in (A7.3), we find an equivalent, computationally simpler test in (A7.3) when there are no couplings between wheels.

Kinematic Motion Contraints

$$\Delta(\mathbf{B}_0)\,\mathbf{A}_0\,\dot{\mathbf{p}} = \mathbf{0} \qquad \text{(A7.3)}$$

Kinematic Motion Constraints With No Wheel Couplings

$$\Delta(\mathbf{J}_i)\,\dot{\mathbf{p}} = 0 \text{ for } i = 1,\ldots,N$$

We may thus determine the kinematic motion constraints for a WMR without redundant wheels or wheel couplings by considering each wheel independently.

The matrix $[\Delta(\mathbf{B}_0)\mathbf{A}_0]$ indicates whether the WMR possesses three DOFs at branch (M00) or fewer than three DOFs at branch (M01). When there are

fewer than three DOFs, the number of independent columns of the matrix $[\Delta(\mathbf{B}_0)\mathbf{A}_0]$ specifies the number of dependent WMR body DOFs. The number of DOFs of a WMR having no redundant wheels is:

Number of WMR DOFs

$$\textit{WMR DOFs} = 3 - \text{rank}[\Delta(\mathbf{B}_0)\mathbf{A}_0] \,. \qquad \text{(A7.4)}$$

The test at branch (M0) determines whether the *complete* inverse solution for *all* of the wheel velocities can be computed by the least-squares solution in (A4.2). In Section 2.4.2, we apply the least-squares solution to compute the *actuated* inverse velocity solution in (2.4.8) for the actuated wheel velocities. Although the actuated inverse velocity solution may exist for some WMR body velocities $\dot{\mathbf{p}}$ for which the complete inverse velocity solution does not, it is impractical to apply such an actuated inverse velocity solution because the desired WMR body velocities are constrained by the unactuated wheel velocities. We thus utilize the soluble motion criterion in (A7.1) to indicate when the actuated inverse velocity solution in Section 2.4.2 is practically applicable.

A7.3 Actuation Characterization

A WMR control engineering application of the actuated inverse velocity solution (in Section 2.4.2) is to command the actuated velocities of the wheels to their computed values (as discussed for the resolved motion rate WMR servo-controller in Chapter 4). In this section, we investigate the characteristics of the WMR body motion when the actuated wheel velocities attain the values computed by the actuated inverse velocity solution. We relate the WMR body velocity vector to the actuated wheel velocities by eliminating the unactuated wheel velocities from the composite robot equation in (2.4.2). Under the no-translational slip assumption, the unactuated wheel velocities will be consistent and *comply* to the WMR motion. We compute the unactuated wheel velocities from the WMR body velocities in the actuated inverse velocity solution in (2.4.8) by interchanging the roles of the actuated ("a" subscripts) and unactuated ("u" subscripts) variables:

$$\dot{\mathbf{q}}_u = \begin{pmatrix} [\mathbf{J}_{1u}^T\Delta(\mathbf{J}_{1a})\mathbf{J}_{1u}]^{-1}\mathbf{J}_{1u}^T\Delta(\mathbf{J}_{1a}) \\ [\mathbf{J}_{2u}^T\Delta(\mathbf{J}_{2a})\mathbf{J}_{2u}]^{-1}\mathbf{J}_{2u}^T\Delta(\mathbf{J}_{2a}) \\ \vdots \\ [\mathbf{J}_{Nu}^T\Delta(\mathbf{J}_{Na})\mathbf{J}_{Nu}]^{-1}\mathbf{J}_{Nu}^T\Delta(\mathbf{J}_{Na}) \end{pmatrix} \dot{\mathbf{p}}. \qquad \text{(A7.5)}$$

The conditions guaranteeing the computability of the unactuated and actuated inverse velocity solutions are identical and are indicated in the soluble motion criterion in (A7.1). We substitute (A7.5) into the partitioned composite robot equation in (2.4.5) to obtain:

$$\begin{pmatrix} \mathbf{I}-\mathbf{J}_{1u}[\mathbf{J}_{1u}^T\Delta(\mathbf{J}_{1a})\mathbf{J}_{1u}]^{-1}\mathbf{J}_{1u}^T\Delta(\mathbf{J}_{1a}) \\ \mathbf{I}-\mathbf{J}_{2u}[\mathbf{J}_{2u}^T\Delta(\mathbf{J}_{2a})\mathbf{J}_{2u}]^{-1}\mathbf{J}_{2u}^T\Delta(\mathbf{J}_{2a}) \\ \vdots \\ \mathbf{I}-\mathbf{J}_{Nu}[\mathbf{J}_{Nu}^T\Delta(\mathbf{J}_{Na})\mathbf{J}_{Nu}]^{-1}\mathbf{J}_{Nu}^T\Delta(\mathbf{J}_{Na}) \end{pmatrix} \dot{\mathbf{p}} = \begin{pmatrix} \mathbf{J}_{1a} & 0 & \dots & 0 \\ 0 & \mathbf{J}_{2a} & . & \vdots \\ 0 & \vdots & . & . \\ 0 & \dots & 0 & \mathbf{J}_{Na} \end{pmatrix} \dot{\mathbf{q}}_a \qquad \text{(A7.6)}$$

or

$$\mathbf{A}_0\,\dot{\mathbf{p}} = \mathbf{B}_a\,\dot{\mathbf{q}}_a. \qquad \text{(A7.7)}$$

The *robot actuation equation* in (A7.7) has the form of (A4.1) with $\mathbf{A}_a$, $\mathbf{B}_a$, $\dot{\mathbf{p}}$, and $\dot{\mathbf{q}}_a$ playing the roles of **A**, **B**, **x**, and **y**, respectively. We apply the solution tree in Figure A4.1 to (A7.7) and obtain the *actuation characterization tree* in Figure A7.3.

The actuation characterization tree, in analogy with the mobility characterization tree, indicates the properties of the actuation structure of a WMR. The branch tests are developed from the solution tree in Figure A4.1. We concentrate on the implications of the solutions.

The system of linear algebraic equations in (A7.7) representing the actuation structure of the WMR may be determined, undetermined or overdetermined. If branch (A1) succeeds, the actuation structure is undetermined and there is no unique solution for the WMR body motion $\dot{\mathbf{p}}$. Since we cannot compute the WMR motion, it is unpredictable, and some WMR

body DOFs are uncontrollable. We suggest that undetermined actuation structures be avoided.

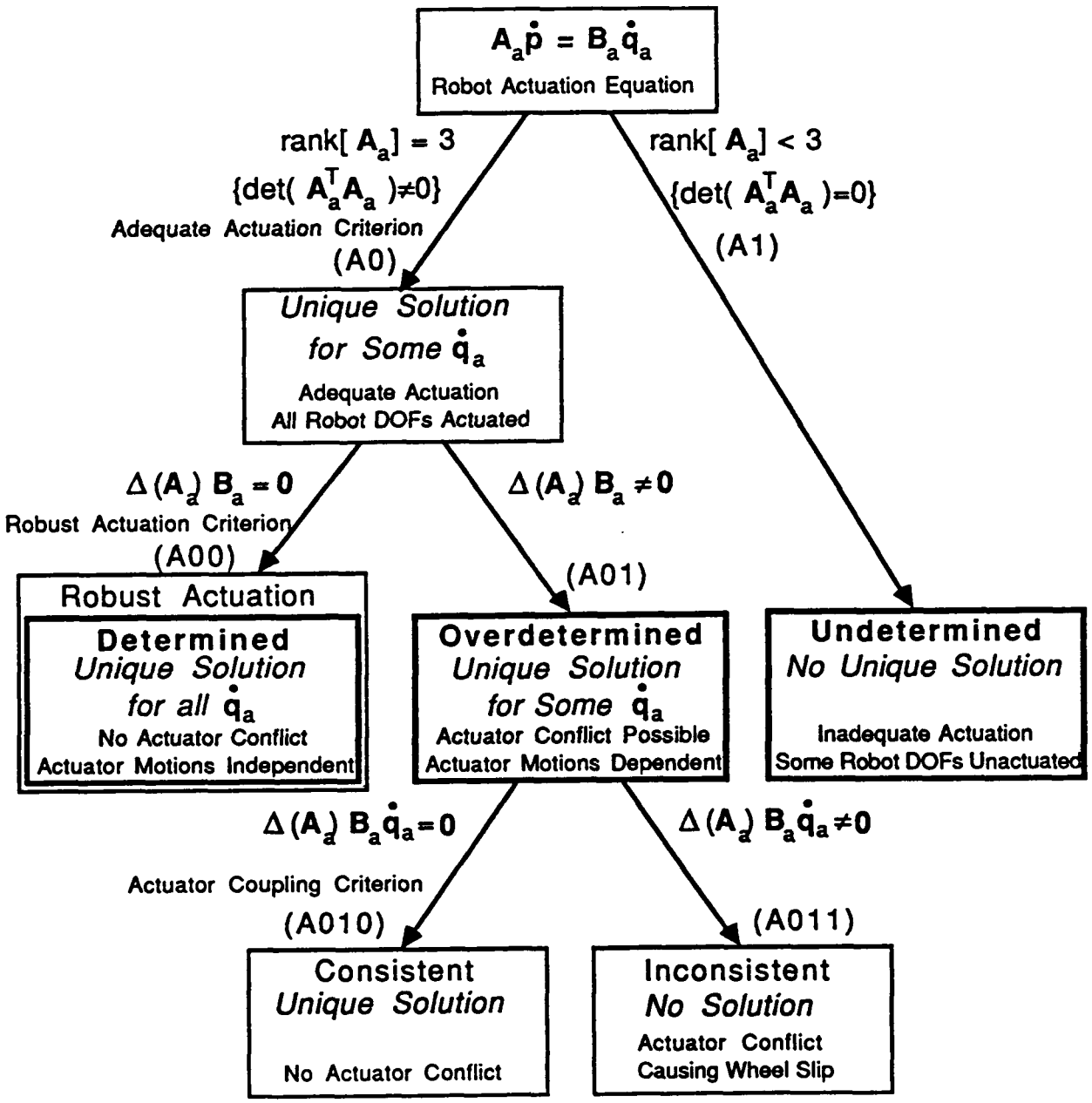


Figure A7.3: The Actuation Characterization Tree

If branch (A0) succeeds, we are assured that all WMR body DOFs are actuated. Specifically, all WMR motions allowed by the mobility structure can be produced by the actuators. Consequently, we refer to branch test (A0) as the *adequate actuation criterion*:

Adequate Actuation Criterion

$$\det(\mathbf{A}_a^T \mathbf{A}_a) \neq 0 \qquad \text{(A7.8)}$$

If the actuation structure is overdetermined (branch (A01)), some of the actuator motions are dependent. If the dependent actuator motions are consistent (at branch (A010)) WMR motion is produced, otherwise (at branch (A011)) translational wheel slip occurs. Mechanical couplings between actuated wheel variables must satisfy the actuator dependencies to allow WMR motion; we therefore refer to branch test (A010) as the *actuator coupling criterion*:

Actuator Coupling Criterion

$$\Delta(\mathbf{A}_a)\, \mathbf{B}_a \dot{\mathbf{q}}_a = \mathbf{0} \qquad \text{(A7.9)}$$

If the dependent actuator motions are not consistent (branch (A011)), translational wheel slip must occur because the least-squares error $\Delta(\mathbf{A}_a)\mathbf{B}_a\dot{\mathbf{q}}_a$ in (A4.3) is non-zero. Since a control system cannot guarantee zero actuator tracking errors, the actuated wheel velocities may deviate from the values computed by the actuated inverse velocity solution. In the presence of these tracking errors, the actuator coupling criterion is not satisfied and the system of linear algebraic equations in (A7.7) becomes inconsistent. We refer to this situation as *actuator conflict* because the forces/torques produced by the inconsistent actuator motions generate stress forces and torques within the WMR structure causing translational wheel slip instead of generating WMR motion. A determined actuation structure (when branch (A00) succeeds) is *robust* in the sense that actuator conflict cannot occur in the presense of actuator tracking errors. The actuator motions are independent and all possible actuated wheel velocity vectors map into unique WMR body velocity vectors. Branch test (A00) is thus referred to as the *robust actuation criterion*:

Robust Actuation Criterion

$$\Delta(\mathbf{A}_a)\, \mathbf{B}_a = \mathbf{0} \qquad \text{(A7.10)}$$

Because of actuator conflict, we suggest that overdetermined actuation structures be avoided. We recommend actuator arrangements leading to a

robust (determined) actuation structure. In Section A7.4, we turn our attention to relate the sensed wheel velocities to the WMR body motion.

A7.4 Sensing Characterization

The relationship between the sensed wheel variables and the WMR body motion is the *dual* of the relationship between the actuated wheel variables and the WMR body motion. Our development thus parallels the discussion in Section A7.3 on WMR actuation characteristics. We begin by rewriting the composite robot equation in (2.4.2) to relate the WMR body velocity vector to the sensed wheel velocity vector. We express the not-sensed wheel velocities in terms of the WMR body velocities by applying the actuated inverse velocity solution in (2.4.8) with the not-sensed ("n" subscripts) and sensed ("s" subscripts) wheel velocities playing the roles of the actuated ("a" subscripts) and unactuated ("u" subscripts) wheel velocities, respectively:

$$\dot{\mathbf{q}}_n = \begin{pmatrix} [\mathbf{J}_{1n}^T \Delta(\mathbf{J}_{1s}) \mathbf{J}_{1n}]^{-1} \mathbf{J}_{1n}^T \Delta(\mathbf{J}_{1s}) \\ [\mathbf{J}_{2n}^T \Delta(\mathbf{J}_{2s}) \mathbf{J}_{2n}]^{-1} \mathbf{J}_{2n}^T \Delta(\mathbf{J}_{2s}) \\ \vdots \\ [\mathbf{J}_{Nn}^T \Delta(\mathbf{J}_{Ns}) \mathbf{J}_{Nn}]^{-1} \mathbf{J}_{Nn}^T \Delta(\mathbf{J}_{Ns}) \end{pmatrix} \dot{\mathbf{p}} . \qquad \text{(A7.11)}$$

The inverse solution is applicable for any WMR satisfying the soluble motion criterion in (A7.1). We partition the sensed and not-sensed wheel velocities in the composite robot equation in (2.4.2) and substitute (A7.11) for the not-sensed wheel velocities to obtain:

$$\begin{pmatrix} \mathbf{I} - \mathbf{J}_{1n} [\mathbf{J}_{1n}^T \Delta(\mathbf{J}_{1s}) \mathbf{J}_{1n}]^{-1} \mathbf{J}_{1n}^T \Delta(\mathbf{J}_{1s}) \\ \mathbf{I} - \mathbf{J}_{2n} [\mathbf{J}_{2n}^T \Delta(\mathbf{J}_{2s}) \mathbf{J}_{2n}]^{-1} \mathbf{J}_{2n}^T \Delta(\mathbf{J}_{2s}) \\ \vdots \\ \mathbf{I} - \mathbf{J}_{Nn} [\mathbf{J}_{Nn}^T \Delta(\mathbf{J}_{Ns}) \mathbf{J}_{Nn}]^{-1} \mathbf{J}_{Nn}^T \Delta(\mathbf{J}_{Ns}) \end{pmatrix} \dot{\mathbf{p}} = \begin{pmatrix} \mathbf{J}_{1s} & 0 & \dots & 0 \\ 0 & \mathbf{J}_{2s} & \ddots & \vdots \\ 0 & \vdots & \ddots & \ddots \\ 0 & \dots & 0 & \mathbf{J}_{Ns} \end{pmatrix} \dot{\mathbf{q}}_s \qquad \text{(A7.12)}$$

or

$$\mathbf{A}_s \dot{\mathbf{p}} = \mathbf{B}_s \dot{\mathbf{q}}_s . \qquad \text{(A7.13)}$$

The *robot sensing equation* in (A7.13) has the form of (A4.1) with $\mathbf{A}_s$, $\mathbf{B}_s$, $\dot{\mathbf{p}}$, and $\dot{\mathbf{q}}_s$ playing the roles of **A**, **B**, **x**, and **y**, respectively. We apply the solution tree in Figure A4.1 to the robot sensing equation in (A7.13) to obtain the *sensing characterization tree* in Figure A7.4.

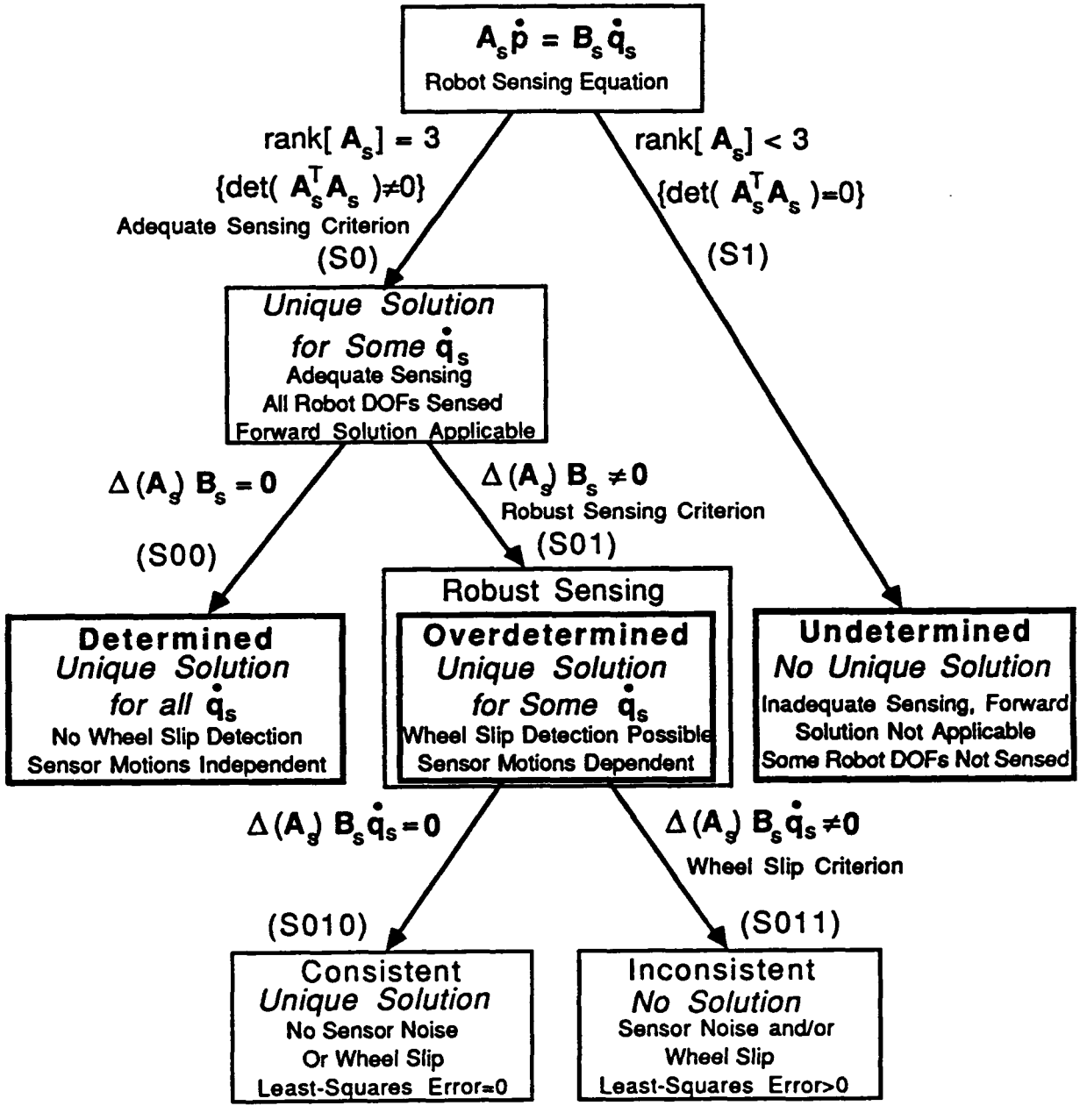


Figure A7.4: The Sensing Characterization Tree

The solution of the WMR body velocity $\dot{\mathbf{p}}$ from the sensed wheel velocities $\dot{\mathbf{q}}_s$ may be determined, undetermined or overdetermined, depending

upon the matrices $\mathbf{A_s}$ and $\mathbf{B_s}$. In parallel with WMR actuation, undetermined systems are undesirable because one or more DOFs of the WMR body motion cannot be discerned from the sensed wheel velocities. Both determined and overdetermined sensing structures allow a unique solution for consistent sensor velocities $\dot{\mathbf{q}}_s$. Branch (S0) thus provides the *adequate sensing criteria* in (A7.14) which specifies whether all WMR body motions allowed by the mobility structure are discernable through sensor measurements:

Adequate Sensing Criterion

$$\det(\mathbf{A_s}^T \mathbf{A_s}) \neq 0 \qquad \text{(A7.14)}$$

The adequate sensing criterion also specifies the conditions under which the sensed forward solution in (2.4.14) is applicable.

Determined sensing structures provide sufficient information for discerning the WMR body motion. Overdetermined sensing structures become inconsistent in the presence of sensor noise, which is analogous to the impact of actuator tracking errors on overdetermined actuation structures. Our forward solution in (2.4.14) anticipates the overdetermined nature of the sensor measurements and provides the least-squares solution. In the case of actuation, an overdetermined actuator structure causes undesirable actuator conflict. In contrast, redundant (and even inconsistent) information is desirable for the least-squares solution of the WMR body velocity from sensed wheel velocities. Redundant information in the least-squares solution reduces the effects of sensor noise on the solution of the WMR body velocity. Overdetermined sensing structures are thereby robust and branch test (S01) is referred to as the *robust sensing criterion*:

Robust Sensing Criterion

$$\Delta(\mathbf{A_s})\, \mathbf{B_s} \neq \mathbf{0} \qquad \text{(A7.15)}$$

We thus recommend that the wheels and wheel sensors be arranged so that the robust sensing criterion is satisfied. When the sensing structure is overdetermined, the least-squares error $\Delta(\mathbf{A_s})\mathbf{B_s}\dot{\mathbf{q}}_s$ in (A4.3) is zero (at branch (S010)) if there is no wheel slip or sensor noise and non-zero (at branch

(S011)) when wheel slip occurs. We therefore denote branch test (S011) as the *wheel slip criterion*:

Wheel Slip Criterion	
$\Delta(A_s)\, B_s\, \dot{q}_s \neq 0$	(A7.16)

In Appendix 10, we detect translational wheel slip by applying the fact that the system of linear algebraic equations in (A7.13) of a robust sensing structure becomes inconsistent in the presence of translational wheel slip.

A7.5 Concluding Remarks

We have extended our kinematic solution methodology in Chapter 2 to characterize the solutions of the composite robot equation and their implications for WMR mobility, actuation and sensing. We have developed the *mobility characterization tree* in Figure A7.2 to characterize the motion properties of a WMR. The implications of the mobility characterization tree are summarized by the following insights. If the *soluble motion criterion* in (A7.1) is satisfied, the actuated inverse velocity solution, actuation and sensing trees, and the WMR DOF computation in (A7.4) are applicable. The *three DOF motion criterion* in (A7.2) indicates whether the WMR kinematic structure allows three DOF motion. If the kinematic structure does not allow three DOF motion, the *kinematic motion constraints* are computed according to (A7.3). The *number of WMR DOFs* are then computed from (A7.4).

The implications of the *actuation characterization tree* in Figure A7.3 are summarized by three criteria. The *adequate actuation criterion* in (A7.8) indicates whether the number and placement of the actuators is adequate for producing all motions allowed by the mobility structure. If the adequate actuation criterion is not satisfied, some WMR body DOFs are uncontrollable. The *robust actuation criterion* in (A7.10) determines whether the actuation structure is robust; i.e., actuator conflict cannot occur in the presence of actuator tracking errors. If the actuation structure is adequate but not robust, some actuator motions are dependent. The *actuator coupling criterion* in (A7.9) computes the actuator dependencies which must be satisfied to avoid actuator

conflict and forced wheel slip. We recommend robust WMR actuation to ensure that actuator conflict cannot occur.

The *sensing characterization tree* in Figure A7.4 indicates properties of the sensing structure of a WMR. The *adequate sensing criterion* in (A7.14) indicates whether the number and placement of the wheel sensors is adequate for discerning all WMR body motions allowed by the mobility structure. The *robust sensing criterion* in (A7.15) indicates whether the sensing structure is such that the computation of the WMR body position from wheel sensor measurements is minimally sensitive to wheel slip and sensor noise. The *wheel slip criterion* in (A7.16) provides a computational algorithm for detecting wheel slip in robust sensing structures and is developed further in Appendix 10. We recommend robust WMR sensing to provide the capability of translational wheel slip detection.

The mobility, actuation, and sensing criterion developed in this appendix apply to a broad range of important WMR issues. The applicability of the actuated inverse velocity solution to WMR servo-control is determined by the *soluble motion criterion.* The *adequate actuation criterion* and *robust actuation criterion* give indications of the controllability of a WMR and the *adequate sensing criterion* and *robust sensing criterion* give indications of the observability of a WMR. In Appendix 8, we apply these actuation and sensing criteria, and the *WMR DOF computation* to kinematics-based mechanical design. The *actuator coupling criterion* is useful as a test to avoid structures which allow actuator conflict. In Appendix 10, we apply the *wheel slip criterion* in (A7.16) to detect the onset of translational wheel slip. Finally, WMR trajectory feasibilty can be assessed kinematically by applying the *kinematic motion constraints computation* in (A7.3).

Appendix 8

Kinematics-Based Mechanical Design

Just as studying the composite robot equation in Appendix 7 enables the determination of such mobility characteristics as the number of DOFs, we formulate in this appendix the design requirements for a WMR to possess desirable mobility characteristics. Desirable mobility characteristics, which are determinable from an analysis of the composite robot equation, are two or three DOFs, and the ability to actuate and sense robustly the motion. By *robust* we mean that the WMR motion is insensitive to actuator tracking errors and that the computation of the robot position from sensor measurements is insensitive to sensor noise and wheel slippage. Designing a WMR to satisfy the desired mobility, actuation and sensing characteristics before construction facilitates the subsequent control system design.

A general-purpose WMR has the ability to move along an x-y path with an orientation trajectory θ. The WMR thus is capable of controlled motion in the three dimensions x, y, and θ at all times, or equivalently possesses three DOFs. This mobility characteristic is sometimes referred to as *omnidirectionality* [Adams84]. For a WMR to operate successfully with three DOFs, it must embody the important characteristics tabulated in Table A8.1 and discussed below. First, it must *allow three DOF motion.* A WMR which possesses three DOFs satisfies the three DOF motion criterion in (A7.2). An omnidirectional WMR design must thus consist of ball, omnidirectional or non-degenerate conventional wheels to allow three DOF motion. A castered backrest used by mechanics for working underneath automobiles exhibits this characteristic.

Second, all three of the WMR body DOFs must be actuated to *produce motion in three DOFs.* The placement of wheels and actuators in the WMR design must be chosen to satisfy the adequate actuation criterion in (A7.8). We require that the actuator structure satisfy the robust actuation criterion in (A7.10) to avoid actuator conflict. The robust actuation criterion states that there be exactly three actuated wheel variables for robust three DOF motion. If there are more than three actuators, their motions must be dependent because WMR

motion occurs in three dimensions. If there are fewer than three actuators, some WMR motions are not actuated and are thus not controllable. The design should thus include only three actuators to ensure robust control.

Table A8.1: Design Criteria for an Omnidirectional WMR

Three DOF Motion:	$\det(J_i^T J_i) \neq 0$ and $w_i = 3$ for i=1,...,N
Adequate Actuation:	$\det(A_a^T A_a) \neq 0$
Robust Actuation:	$a = 3$
Adequate Sensing:	$\det(A_s^T A_s) \neq 0$
Robust Sensing:	$s > 3$

The Unimation robot (in Appendix 11) has three actuated omnidirectional wheels (Troas-whemor) and is an example of a WMR having a robust actuation structure. Uranus (in Chapter 6) has four actuated omnidirectional wheels (Tetroas-whemor) and is not robust because the actuator motions are dependent. In Appendix 11, we examine an alternate design of Uranus having a robust actuation structure. Our study of Uranus provides a technique for redesigning adequate actuation structures to be robust.

The third requirement for an omnidirectional WMR is that a control system communicates signals to the actuators so that the WMR follows a specified (x, y, θ) trajectory. An omnidirectional WMR which computes its present position from wheel shaft encoder measurements and controls the actuators to reduce the error between the desired WMR position and the actual WMR position possesses this characteristic. To compute the WMR position from wheel shaft encoder measurements, the wheel sensors must be positioned so that the *WMR motion may be discerned in three DOFs.* To discern any WMR motion, the sensing structure must satisfy the adequate sensing criterion in (A7.14). We require a robust sensing arrangement (i.e., the WMR design should include more than three wheel sensors) to allow robust computation of the robot position from wheel sensor measurements.

A WMR which does not allow three DOF motion has *singularities* in its workspace. At a singularity, the WMR cannot attain independent motion along one or more dimensions (i.e., x, y, or θ). We may determine the kinematic

motion constraints of a WMR allowing fewer than three DOFs by computing (A7.3). Once a WMR design possesses the desired mobility characteristics, we apply the actuation and sensing criteria in Appendix 7 to verify that the actuation and sensing structures are adequate or robust.

A WMR with two DOFs allows locomotion along any x-y path and thus has wide applicability for parts and materials transport. Topo [Helmers83b], Newt (in Appendix 11), and Shakey [Nilsson84] each possesses two DOFs utilizing two diametrically opposed conventional drive wheels. These *bicas-polycsun-whemors* also have zero, one, and two casters, respectively, for stability. We show in Appendix 11 that a design utilizing two diametrically opposed drive wheels is appealing because of its mechanical and modeling simplicity. Because of the practical advantages of two diametrically opposed drive wheels, we recommend the application of bicas-polycsun structures for all tasks requiring fewer than three DOFs. This guideline simplifies the design process for the majority of parts and materials transport applications.

We detail in Chapter 8 how to enhance the bicas-polycsun structure by mounting an actuated turrent having a rotational DOF in the plane of the surface-of-travel. The turret is mounted off center from the two actuated wheels. A coordinate system assigned fixed to the center of the turrent would attain x and y DOFs from the actuation of the wheels and the θ DOF from the turret actuation. Manipulators, sensors and docking instruments mounted upon the turret thereby attain three DOF in the plane. This motion of the turret is then equivalent to the motion of the body of an omnidirectional WMR. The wheel diameter of such a WMR may be made more easily larger than the wheel diameter of the rollers on omnidirectional wheels, so that the bicas-polycsun may locomote over larger surface irregularities than a WMR having omnidirectional wheels. The bicas-polycsun structure with a turret possesses the motion advantages of an omnidirectional WMR without the mechanical and modeling complexities.

Appendix 9

Dead Reckoning

In this appendix, we describe *dead reckoning*, the real-time computation of the WMR position from wheel sensor measurements. The current WMR position is utilized by closed-loop WMR control systems, performance monitoring processes and high-level WMR planning processes. The least-squares sensed forward velocity solution in (2.4.14) is the exact solution for the WMR velocities under the no translational slip assumption, if the wheel sensing structure is adequate. The adequate sensing criterion in (A7.14) is a prerequisite for implementing three dimensional dead reckoning. To determine the WMR position in real-time, the WMR body velocity is integrated over each sampling period. Since the dead reckoning computation is erroneous when translational wheel slip occurs, an alternate method (e.g., computer vision) of determining the WMR position must be applied to correct the position computation before dead reckoning is continued. In Appendix 10, we propose a method to detect the onset of translational wheel slip.

Dead reckoning begins when the WMR is at rest or has a sensed initial velocity ${}^{F}\dot{\mathbf{p}}_{B}(0)$. The initial robot position ${}^{F}\mathbf{p}_{B}(0)$ is either specified or sensed. We assume that the WMR motion is adequately modeled by piecewise constant accelerations[1] since the WMR is being actuated by constant force/torque generators in each sampling period (the same sampling period as the dead reckoning process). The WMR body velocity ${}^{\bar{B}}\dot{\mathbf{p}}_{B}$ in the sampling period from time t=(n-1)T to time t=nT is then

$$ {}^{\bar{B}}\dot{\mathbf{p}}_{B}(t) = {}^{\bar{B}}\dot{\mathbf{p}}_{B}[(n-1)T] + \frac{{}^{\bar{B}}\dot{\mathbf{p}}_{B}(nT) - {}^{\bar{B}}\dot{\mathbf{p}}_{B}[(n-1)T]}{T}(t-[(n-1)T]) \qquad \text{(A9.1)} $$

where the WMR body velocity ${}^{\bar{B}}\dot{\mathbf{p}}_{B}(nT)$ at each sampling instant is computed according to the sensed forward velocity solution in (2.4.14). We transform the

---

1 We apply this assumption to illustrate the dead reckoning algorithm. For a specific WMR, it may be necessary to utilize higher-order models of the velocity trajectory.

WMR body velocity to the floor coordinate system by applying the motion matrix in (2.3.34):

$$ {}^{F}\dot{\mathbf{p}}_B(t) = \mathbf{M}[(n-1)T]\ {}^{\bar{B}}\dot{\mathbf{p}}_B(t)\ . \qquad \text{(A9.2)} $$

We use the angular WMR position at sampling instant t=(n-1)T to compute the motion matrix $\mathbf{M}[(n-1)T]$ since the current angular WMR position at time t is unknown. We compute the WMR position at the current sampling instant t=nT by integrating the velocity over the sampling period and adding the result to the WMR position at sampling instant t=(n-1)T:

$$ {}^{F}\mathbf{p}_B(nT) = {}^{F}\mathbf{p}_B[(n-1)T] + \int_{(n-1)T}^{nT} {}^{F}\dot{\mathbf{p}}_B(t)\ dt\ . \qquad \text{(A9.3)} $$

By substituting (A9.1) and (A9.2) into the integral in (A9.3), we express the present WMR position in terms of the position at the last sampling instant and the WMR body velocity at the present and last sampling instants:

Dead Reckoning Update Algorithm

$$ {}^{F}\mathbf{p}_B(nT) = {}^{F}\mathbf{p}_B[(n-1)T] + \frac{T}{2}\mathbf{M}[(n-1)T]\left\{ {}^{\bar{B}}\dot{\mathbf{p}}_B[(n-1)T] + {}^{\bar{B}}\dot{\mathbf{p}}_B(nT) \right\} \qquad \text{(A9.4)} $$

The computational load for dead reckoning is thus the computation of the sensed forward velocity solution in (2.4.14).

Appendix 10

Wheel Slip Detection

In Section 2.4.3, we compute the WMR body velocity vector from the wheel sensor measurements (i.e., the sensed forward velocity solution), and in Appendix A7.4 we discuss the characteristics of the solution. We can discern all WMR body motions if the adequate sensing criterion in (A7.14) is satisfied. If the sensing structure is adequate but not robust, the equations-of-motion will be consistent irrespective of the presense of unmodeled (i.e., translational) wheel slip and the error in the least-squares forward velocity solution in (2.4.14) will be zero. In contrast, for a robust sensing structure (i.e., a sensing structure satisfying the robust sensing criterion), the kinematic equations-of-motion are inconsistent in the presence of translational wheel slip. The error in the least-squares forward velocity solution is then greater than zero. In this appendix,we therefore propose to detect the occurrence of translational wheel slippage for a WMR having a robust sensing structure *by computng the error in the least-squares solution.* In the improbable case that all wheels on a WMR slip simultaneously in such a manner that the equations-of-motion remain consistent, our method will fail to detect the wheel slip.

In practice, sensor noise can also cause the kinematic equations-of-motion to become inconsistent, but we expect that the least-squares error due to sensor noise will be small in comparison with the error caused by wheel slippage. Instead of testing the least-squares error against zero, we propose to compare it with an error threshold e_t set by the worst case sensor noise error. If the least-squares error in the forward velocity solution exceeds the threshold, we conclude that wheel slip has occurred. When a WMR detects that wheel slip has occurred, it should resort to absolute methods (e.g., computer vision, ultrasonic ranging sensors, and laser range finders) of determining its position before continuing the dead reckoning computations. Since currently-available locating methods are computationally slow relative to the WMR motion, the WMR should halt motion until its dead reckoning computations are updated by the absolute locating method.

Computation of the sensed forward velocity solution in (2.4.14) is the first step in determining the least-squares error. The computed WMR body velocity vector ${}^{\bar{B}}\dot{\mathbf{p}}_B$ is substituted for the actual WMR velocity vector in the robot sensing equation in (A7.13). The least-squares error vector **e** is computed by subtracting the right-hand side of (A7.13) from the left-hand side:

$$\mathbf{e} = \mathbf{A}_s \, {}^{\bar{B}}\dot{\mathbf{p}}_B - \mathbf{B}_s \, \dot{\mathbf{q}}_s \, . \qquad \text{(A10.1)}$$

We compute and compare the square of the norm of the least-squares error $[\mathbf{e}^T\mathbf{e}]$ with the scalar threshold e_t^2. If the norm of the least-squares error exceeds the threshold, we conclude that unmodeled translational wheel slip has occurred:

Detection of Wheel Slip

$$\text{If} \quad \mathbf{e}^T\mathbf{e} > e_t^2 \, , \quad \text{wheel slip has occurred.} \qquad \text{(A10.2)}$$

We note that (A10.2) is, in principle, equivalent to the wheel slip criterion in (A7.16) and has the added advantage that the sensed forward velocity solution in (2.4.14) is computed as an intermediate result. The sensed forward velocity solution may then be applied to dead reckoning and WMR servo-control.

Appendix 11

WMR Kinematic Modeling Examples

A11.1 Overview

In this appendix, we illustrate the kinematic modeling of *six* WMRs [Muir86]: the Unimation robot, Newt, an alternate design of Uranus, Neptune, Pluto, and the Stanford Cart which represent a broad range of kinematic structures. For each WMR, we provide four kinematic descriptions: a written description, a top and side view sketch, the symbolic diagram and the kinematic name. We assign the coordinate systems to create the coordinate transformation matrices. We then form the wheel Jacobian matrices by substituting elements of the coordinate transformation matrices into the symbolic wheel Jacobian matrix templates in Appendix 3. We determine the nature of the mobility, actuation and sensing structures to gain insight into the mobility characteristics of the WMR. We compute the actuated inverse velocity solution and the sensed forward velocity solution when the mobility analysis indicates that these solutions are applicable. We complete each example with remarks on the WMR kinematic structure and suitability for particular tasks.

A11.2 Unimation WMR

The Unimation WMR [Carlisle83] illustrated in Figure A11.1 utilizes three symmetrically positioned omnidirectional wheels with rollers at 90°. A motor actuates each wheel and the velocity of each wheel is sensed by shaft encoders. The rollers are neither actuated nor sensed. The coordinate system assignments and pertinent WMR dimensions are shown in the figure.

We write the coordinate transformation matrices in Tables 2.4 and 2.5 from Figure A11.1:

$$ {}^{B}T_{C_1} = \begin{pmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & l_a \\ 0 & 0 & 1 & -l_b \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad {}^{B}T_{C_2} = \begin{pmatrix} \sqrt{3}/2 & 1/2 & 0 & \sqrt{3}l_a/2 \\ -1/2 & \sqrt{3}/2 & 0 & -l_a/2 \\ 0 & 0 & 1 & -l_b \\ 0 & 0 & 0 & 1 \end{pmatrix} $$

$$ {}^{B}T_{C_3} = \begin{pmatrix} -\sqrt{3}/2 & 1/2 & 0 & -\sqrt{3}l_a/2 \\ -1/2 & -\sqrt{3}/2 & 0 & -l_a/2 \\ 0 & 0 & 1 & -l_b \\ 0 & 0 & 0 & 1 \end{pmatrix} $$

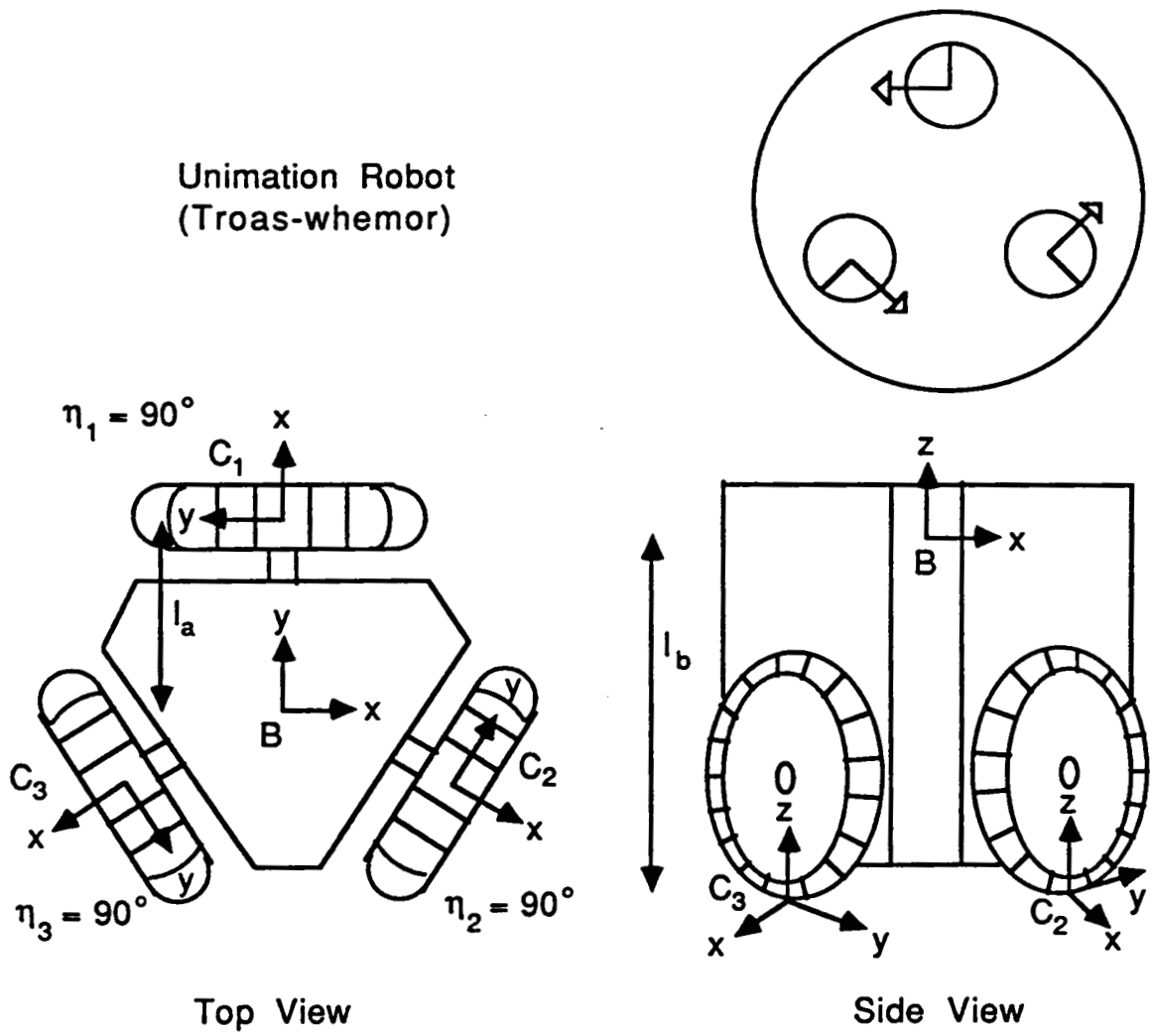


Figure A11.1: Coordinate System Assignments for the Unimation WMR

We substitute the elements of the transformation matrices, the wheel (R) and roller (r) radii, and the roller angles (η) into the symbolic Jacobian matrix for omnidirectional wheels in (A3.7) to write the kinematic wheel equations:

$$\dot{\mathbf{p}} = \begin{pmatrix} -R & 0 & l_a \\ 0 & r & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} {}^{A_1}\omega_{W_1x} \\ {}^{P_1}\omega_{R_1x} \\ \bar{C}_1\omega_{C_1z} \end{pmatrix} = \mathbf{J}_1 \dot{\mathbf{q}}_1 \qquad \text{(A11.1)}$$

$$\dot{\mathbf{p}} = \begin{pmatrix} R/2 & \sqrt{3}r/2 & -l_a/2 \\ \sqrt{3}R/2 & -r/2 & -\sqrt{3}l_a/2 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} {}^{A_2}\omega_{W_2x} \\ {}^{P_2}\omega_{R_2x} \\ \bar{C}_2\omega_{C_2z} \end{pmatrix} = \mathbf{J}_2 \dot{\mathbf{q}}_2 \qquad \text{(A11.2)}$$

$$\dot{\mathbf{p}} = \begin{pmatrix} R/2 & -\sqrt{3}r/2 & -l_a/2 \\ -\sqrt{3}R/2 & -r/2 & \sqrt{3}l_a/2 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} {}^{A_3}\omega_{W_3x} \\ {}^{P_3}\omega_{R_3x} \\ \bar{C}_3\omega_{C_3z} \end{pmatrix} = \mathbf{J}_3 \dot{\mathbf{q}}_3 \qquad \text{(A11.3)}$$

To characterize the WMR mobility, we note that the soluble motion criterion in (A7.1) is satisfied. Consequently, none of the wheels has redundant DOFs and the actuated inverse velocity solution is applicable. Since the three DOF motion criterion in (A7.2) is also satisfied, the Unimation WMR allows three DOF motion.

We compute the adequate actuation criterion $\det[\mathbf{A}_a^T\mathbf{A}_a]=27l_a/4$ as the first step in characterizing the actuation structure. Since the determinant is nonzero, all WMR body motions are controllable by the motions of the actuators. The value of $\Delta(\mathbf{A}_a)\mathbf{B}_a$ is zero which indicates that the robust actuation criterion in (A7.10) is also satisfied. The actuator motions are independent and no actuator conflict can occur. Since the adequate sensing criterion in (A7.14) is satisfied but the robust sensing criterion in (A7.15) is not, the sensing structure is adequate but not robust. Although the sensing structure allows three DOFs to be discerned by applying the sensed forward velocity solution in (2.4.14), translational wheel slip cannot be detected by the method of Appendix 10.

Since the soluble motion criterion in (A7.1) is satisfied, the actuated inverse velocity solution is computable. The actuated inverse velocity solution in (2.4.8) applies directly:

$$\begin{pmatrix} \dot{q}_{1a} \\ \dot{q}_{2a} \\ \dot{q}_{3a} \end{pmatrix} = \begin{pmatrix} [J_{1a}^T \Delta(J_{1u}) J_{1a}]^{-1} J_{1a}^T \Delta(J_{1u}) \\ [J_{2a}^T \Delta(J_{2u}) J_{2a}]^{-1} J_{2a}^T \Delta(J_{2u}) \\ [J_{3a}^T \Delta(J_{3u}) J_{3a}]^{-1} J_{3a}^T \Delta(J_{3u}) \end{pmatrix} \begin{pmatrix} {}^{\bar{B}}v_{Bx} \\ {}^{\bar{B}}v_{By} \\ {}^{\bar{B}}\omega_{Bz} \end{pmatrix},$$

from which

$$\begin{pmatrix} {}^{A_1}\omega_{W_1x} \\ {}^{A_2}\omega_{W_2x} \\ {}^{A_3}\omega_{W_3x} \end{pmatrix} = \frac{1}{R} \begin{pmatrix} -1 & 0 & l_a \\ 1/2 & \sqrt{3}/2 & l_a \\ 1/2 & -\sqrt{3}/2 & l_a \end{pmatrix} \begin{pmatrix} {}^{\bar{B}}v_{Bx} \\ {}^{\bar{B}}v_{By} \\ {}^{\bar{B}}\omega_{Bz} \end{pmatrix}. \qquad \text{(A11.4)}$$

Since the adequate sensing criterion in (A7.14) is satisfied, the sensed forward velocity solution in (2.4.14) yields:

$$\dot{p} = [\Delta(J_{1n})+\Delta(J_{2n})+\Delta(J_{3n})]^{-1} \begin{bmatrix} \Delta(J_{1n})J_{1s} & \Delta(J_{2n})J_{2s} & \Delta(J_{3n})J_{3s} \end{bmatrix} \begin{pmatrix} \dot{q}_{1s} \\ \dot{q}_{2s} \\ \dot{q}_{3s} \end{pmatrix}$$

and we obtain

$$\begin{pmatrix} {}^{\bar{B}}v_{Bx} \\ {}^{\bar{B}}v_{By} \\ {}^{\bar{B}}\omega_{Bz} \end{pmatrix} = R \begin{pmatrix} -2/3 & 1/3 & 1/3 \\ 0 & 1/\sqrt{3} & -1/\sqrt{3} \\ 1/(3l_a) & 1/(3l_a) & 1/(3l_a) \end{pmatrix} \begin{pmatrix} {}^{A_1}\omega_{W_1x} \\ {}^{A_2}\omega_{W_2x} \\ {}^{A_3}\omega_{W_3x} \end{pmatrix}. \qquad \text{(A11.5)}$$

The Unimation WMR is a general-purpose three DOF WMR. It allows three DOF motion, has adequate actuation to produce three DOF motion, and has adequate sensing to discern three DOF motion. The actuated inverse and sensed forward velocity solutions are computable in real-time for servo-control applications. The low ground clearance, which only allows locomotion on smooth, level surfaces is a disadvantage of the design. The mechanical complexity of the omnidirectional wheels increases the cost and difficulty of fabrication. It is difficult to construct perfectly circular omnidirectional wheels when the rollers are at 90° because of the discontinuities between rollers. An improved wheel design allowing circular omnidirectional wheel profiles has

been implemented for Uranus (in Chapter 6). We have noted that the sensing structure does not allow translational wheel slip detection by the method of Appendix 10. Although the wheel variables which are not-sensed are difficult to instrument, an additional instrumented caster can be incorporated in the design to provide practical robust sensing and translational wheel slip detection.

Three DOF locomotion is not necessary for parts and materials transport. A transport WMR may operate with two DOFs. The three DOF locomotion is advantageous when utilized with an onboard manipulator. The mobility of the WMR enhances and extends the workspace of the manipulator. Consequently, a manipulator having fewer than six DOFs which complement the DOFs of the mobile base has an unlimited workspace and can accomplish the tasks of a stationary manipulator having six DOFs.

A11.3 Newt

Newt [Hollis77] is a WMR having two diametrically opposed drive wheels and a free-wheeling castor, as shown in Figure A11.2. Both drive wheels are actuated and sensed, while the castor is neither actuated nor sensed.

The coordinate transformation matrices for Newt are:

$$ {}^{B}T_{C_1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -l_e \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad {}^{B}T_{C_2} = \begin{pmatrix} 1 & 0 & 0 & -l_a \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -l_e \\ 0 & 0 & 0 & 1 \end{pmatrix} $$

$$ {}^{B}T_{C_3} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -l_b \\ 0 & 0 & 1 & -(l_e - l_d) \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad {}^{H_3}\Phi_{S_3} = \begin{pmatrix} \cos^{H_3}\theta_{S_3} & -\sin^{H_3}\theta_{S_3} & 0 & 0 \\ \sin^{H_3}\theta_{S_3} & \cos^{H_3}\theta_{S_3} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} $$

$$ {}^{S_3}T_{C_3} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -l_c \\ 0 & 0 & 1 & -l_d \\ 0 & 0 & 0 & 1 \end{pmatrix}. $$

Newt
(Bicas-Unicsun-Whemor)

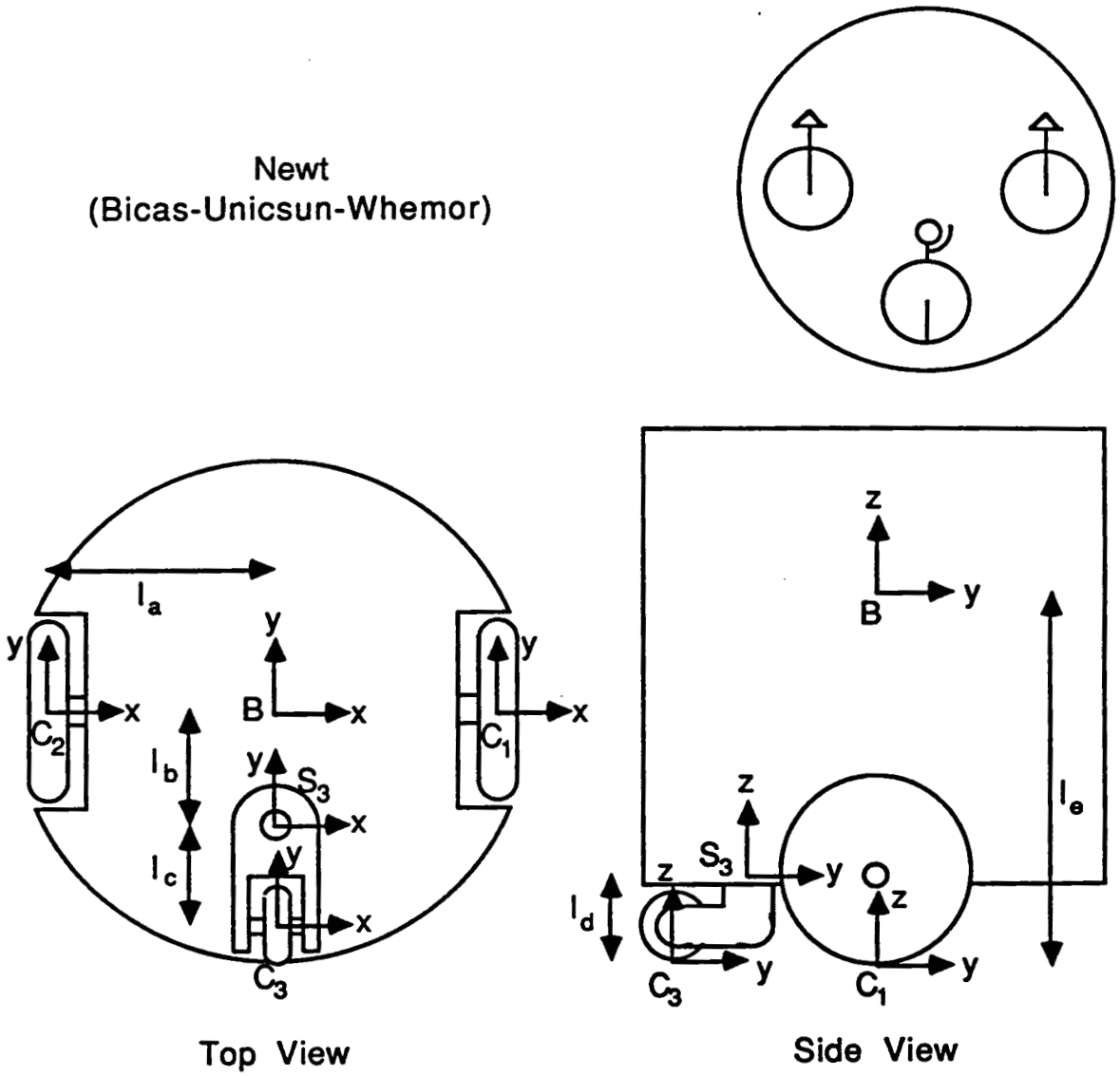


Figure A11.2: Coordinate System Assignments for Newt

The radii of wheels one and two are identical: $R_1=R_2=R$, and the radius of wheel three is $R_3=r$. By applying the Jacobian matrix for non-steered conventional wheels in (A3.2), we write the matrix equations for drive wheels one and two:

$$\dot{p} = \begin{pmatrix} 0 & 0 \\ R & -l_a \\ 0 & 1 \end{pmatrix} \begin{pmatrix} {}^{A_1}\omega_{W_1 x} \\ {}^{\bar{C}_1}\omega_{C_1 z} \end{pmatrix} = J_1 \dot{q}_1 \qquad \text{(A11.6)}$$

$$\dot{\mathbf{p}} = \begin{pmatrix} 0 & 0 \\ R & l_a \\ 0 & 1 \end{pmatrix} \begin{pmatrix} {}^{A_2}\omega_{W_{2x}} \\ \bar{C}_2\omega_{C_{2z}} \end{pmatrix} = \mathbf{J}_2 \dot{\mathbf{q}}_2 \qquad \text{(A11.7)}$$

Similarly, by applying the Jacobian matrix for a steered conventional wheel in (A3.4), we write the matrix equation for wheel three:

$$\dot{\mathbf{p}} = \begin{pmatrix} -R\sin^{H_3}\theta_{S_3} & -l_c\cos^{H_3}\theta_{S_3} - l_b & l_b \\ R\cos^{H_3}\theta_{S_3} & -l_c\sin^{H_3}\theta_{S_3} & 0 \\ 0 & 1 & -1 \end{pmatrix} \begin{pmatrix} {}^{A_3}\omega_{W_{3x}} \\ \bar{C}_3\omega_{C_{3z}} \\ {}^{H_3}\omega_{S_{3z}} \end{pmatrix} = \mathbf{J}_3 \dot{\mathbf{q}}_3 \qquad \text{(A11.8)}$$

The soluble motion criterion in (A7.1) is satisfied, indicating that the actuated inverse velocity solution is applicable. Since $w_i=2$ for wheels one and two, the three DOF motion criterion in (A7.2) is not satisfied. The WMR has fewer than three DOFs; i.e., some WMR DOFs are dependent. The matrix product $[\Delta(\mathbf{B}_0)\mathbf{A}_0]$ has rank one, and according to the expression for the number of WMR DOFs in (A7.4), Newt has two DOFs. The kinematic motion constraints for wheels one and two simplify to ${}^{\bar{B}}v_{Bx}=0$. Wheel three imposes no constraints on the WMR motion. The WMR thus allows independent motion in two DOFs: y and θ.

We determine the actuation structure by first computing the adequate actuation criterion $\det[\mathbf{A}_a^T\mathbf{A}_a]=8l_a^2$ in (A7.8). Accordingly, both WMR DOFs are actuated (i.e., all robot motions in the y and θ directions may be produced by the actuators). We find further that the robust actuation criterion $\Delta(\mathbf{A}_a)\mathbf{B}_a=\mathbf{0}$ is satisfied. All actuator motions are independent, providing robust two DOF actuation. The sensing structure is adequate but not robust because the sensed wheel variables and the actuated ones are identical. Even though the sensing structure is not robust, the sensed forward velocity solution in (2.4.14) is applicable.

Although the actuated inverse solution is applicable, only WMR body motions for which the translational velocity ${}^{\bar{B}}v_{Bx}$ is zero are possible. The actuated inverse velocity solution will thus be the exact solution if the x-component of the WMR body velocity is chosen to be zero. If the x-component of

the WMR body velocity is non-zero, the computed actuated inverse velocity solution will be erroneous. The resulting solution will be the optimal set of actuated wheel velocities which minimizes the least-squares error between the desired WMR body velocity and the resulting WMR body velocity. We apply the actuated inverse velocity solution in (2.4.8) to obtain:

$$\begin{pmatrix} \dot{q}_{1a} \\ \dot{q}_{2a} \end{pmatrix} = \begin{pmatrix} [J_{1a}^T \Delta(J_{1u}) J_{1a}]^{-1} J_{1a}^T \Delta(J_{1u}) \\ [J_{2a}^T \Delta(J_{2u}) J_{2a}]^{-1} J_{2a}^T \Delta(J_{2u}) \end{pmatrix} \dot{p}$$

and thereby

$$\begin{pmatrix} {}^{A_1}\omega_{W_1x} \\ {}^{A_2}\omega_{W_2x} \end{pmatrix} = \frac{1}{R} \begin{pmatrix} 0 & 1 & l_a \\ 0 & 1 & -l_a \end{pmatrix} \begin{pmatrix} {}^{\bar{B}}v_{Bx} \\ {}^{\bar{B}}v_{By} \\ {}^{\bar{B}}\omega_{Bz} \end{pmatrix} \qquad \text{(A11.9)}$$

Since the sensing structure is adequate, the sensed forward velocity solution in (2.4.14) is applicable:

$$\dot{p} = [\Delta(J_{1n}) + \Delta(J_{2n}) + \Delta(J_{3n})]^{-1} \begin{bmatrix} \Delta(J_{1n}) J_{1s} & \Delta(J_{2n}) J_{2s} \end{bmatrix} \begin{pmatrix} \dot{q}_{1s} \\ \dot{q}_{2s} \end{pmatrix},$$

and hence

$$\begin{pmatrix} {}^{\bar{B}}v_{Bx} \\ {}^{\bar{B}}v_{By} \\ {}^{\bar{B}}\omega_{Bz} \end{pmatrix} = \frac{R}{2l_a} \begin{pmatrix} 0 & 0 \\ l_a & l_a \\ 1 & -1 \end{pmatrix} \begin{pmatrix} {}^{A_1}\omega_{W_1x} \\ {}^{A_2}\omega_{W_2x} \end{pmatrix}. \qquad \text{(A11.10)}$$

The x-component of the WMR body velocity is zero independent of the sensor measurements. The y-component of the WMR body velocity is proportional to the sum of the wheel velocities, and the θ-component is proportional to the difference of the wheel velocities.

Newt is a general-purpose robot for tasks requiring only two-dimensional motion. Any path in a plane can be traced by a WMR possessing two DOFs. Since the vast majority of existing WMRs are applied for transporting parts,

materials, and tools from one point to another along a path, Newt has wide applicability. The simple mechanical design is advantageous over omnidirectional designs because it requires fewer parts and has reduced cost and maintenance. A robust sensing structure may be obtained by sensing the wheel and steering velocities of the castor. An important feature of this design is that the dead reckoning integration computations (in Appendix 9) for the angular position of the WMR body are not required. If no translational wheel slip occurs, the angular WMR body position can be computed at any time nT according to

$$ {}^{F}\theta_{Bz}(nT) = \frac{R}{l_a}\left[{}^{A_1}\theta_{W_1x}(nT) - {}^{A_2}\theta_{W_2x}(nT)\right] + {}^{F}\theta_{Bz}(0)\,. \qquad \text{(A11.11)} $$

The computational errors due to finite precision limits and sensor noise do not accumulate in the computation of ${}^{F}\theta_{Bz}(nT)$ in (A11.11) as they would if the dead reckoning integration in (A9.4) were required.

From our analysis, we conclude that Newt has two DOFs in the y and θ directions with the WMR body coordinate system assigned in Figure A11.2. If the WMR body coordinate system is assigned at any point along the WMR y-axis except zero, the two DOFs will be x and y. If the WMR body coordinate system is rotated 90° from its orientation in Figure A11.2, the two DOFs will be x and θ. Finally, if the WMR body coordinate system is assigned to an arbitrary position not on the x or y axes, the two DOFs cannot be specified by two of the three components x, y, and θ. We conclude that the number of DOFs of a WMR is independent of the assignment of coordinate axes, but the allowable directions of motion depend upon the placement of the WMR body coordinate system.

A11.4 Uranus Derivatives

Uranus is a convenient WMR with which to develop an understanding of the differences between inadequate, adequate and robust actuation and sensing structures, and the need for a kinematic analysis in the design of a WMR. We show in Chapter 6 that Uranus has an adequate but not a robust actuation structure which provides motion in all three DOFs, but allows actuator conflict. In Figure A11.3, we consider a slightly different WMR design.

The WMR in Figure A11.3 is identical to Uranus except the the wheels on the right and left hand sides of the WMR have been interchanged and the distances l_a and l_b are equal. The wheels are actuated and sensed as with Uranus. Upon modeling this WMR and characterizing its actuation and sensing structure, we find that it is *inadequate* (i.e., $\det[\mathbf{A}_a^T \mathbf{A}_a]=0$). The problem is that the angular rotation of the WMR is not constrained by the motions of the actuators (sensors). We observe in Figure A11.3 that the WMR can be spun about its center even if the wheel actuators are locked to one position because the rollers are free to turn.

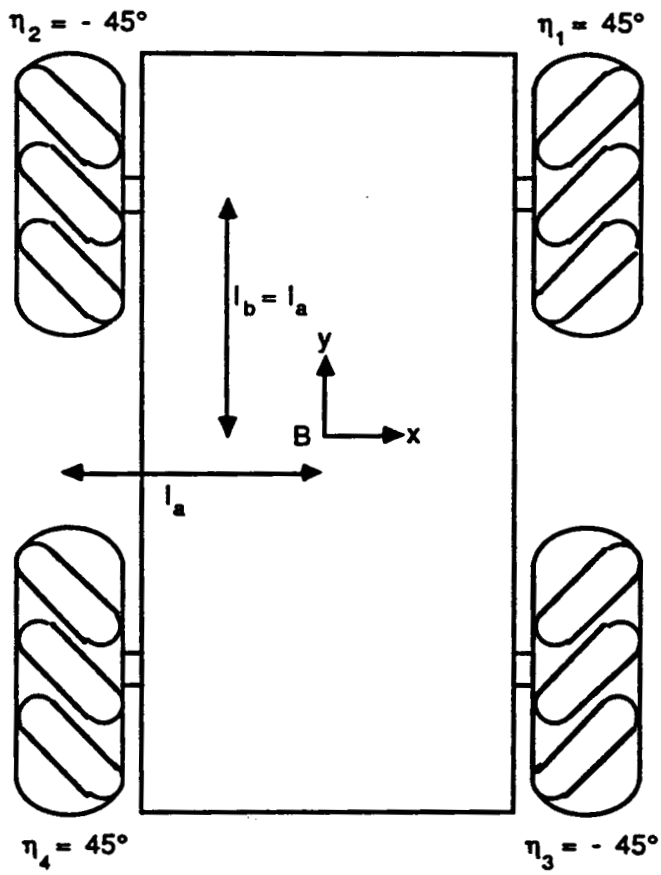


Figure A11.3: Uranus with an Inadequate Actuation Structure

We realize that the non-robust nature of Uranus' actuation structure allows actuator conflict. We now imagine how Uranus might be altered to avoid actuator conflict. Since we are interested in a practical symmetric alternative, we eliminate the possibility of simply removing one of the actuators. We must

ensure that the actuator coupling criterion in (A7.9) is satisfied. The rank one actuator coupling criterion for Uranus reduces to the scalar equation:

$$A_1\omega_{W_{1x}} + A_2\omega_{W_{2x}} - A_3\omega_{W_{3x}} - A_4\omega_{W_{4x}} = 0 \qquad \text{(A11.12)}$$

Only three of the four actuator motions are independent. Our solution in Figure A11.4 is to constrain mechanically the wheel motions with gearing between wheels to ensure that the actuator coupling criterion in (A11.12) is satisfied.

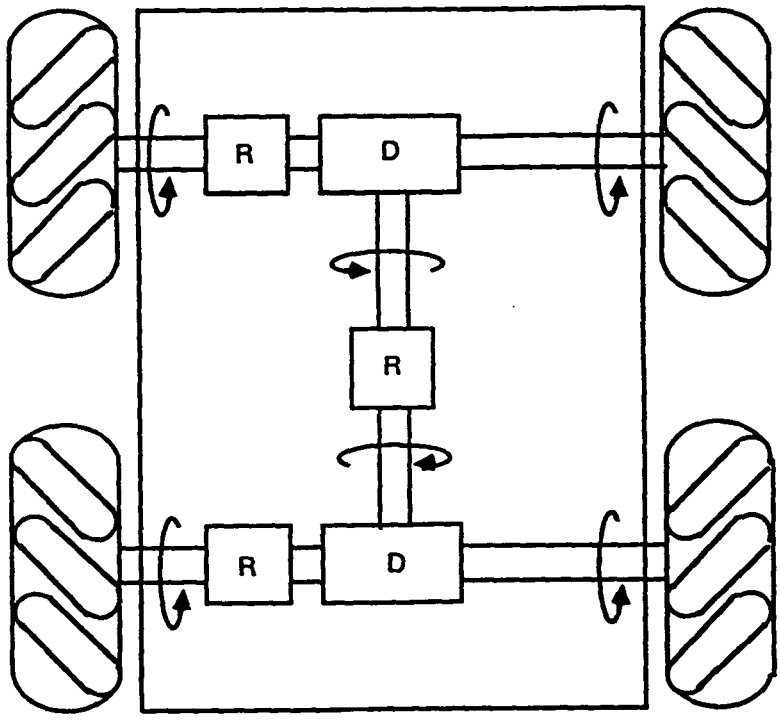


Figure A11.4: Converting Uranus into a Robust Actuation Structure

We utilize differential gearing (D) and reversing gearing (R). A differential gearbox is designed so that the output shaft rotates at a speed equal to the difference of the speeds of the two input shafts. A reversing gearbox is designed so that the output shaft rotates at a speed equal and opposite to the speed of the input shaft. In Figure A11.5, we add three symmetrically placed motors (M) for actuation. The actuation structure of Figure A11.5 is robust. We write the composite robot equation in terms of the *motor shaft rotations* (instead of the wheel axle rotations), and apply the robust actuation criterion to verify the

design. Even though the complexity of this gearing may prohibit practical implementation, the procedure may be applied to the design of any WMR.

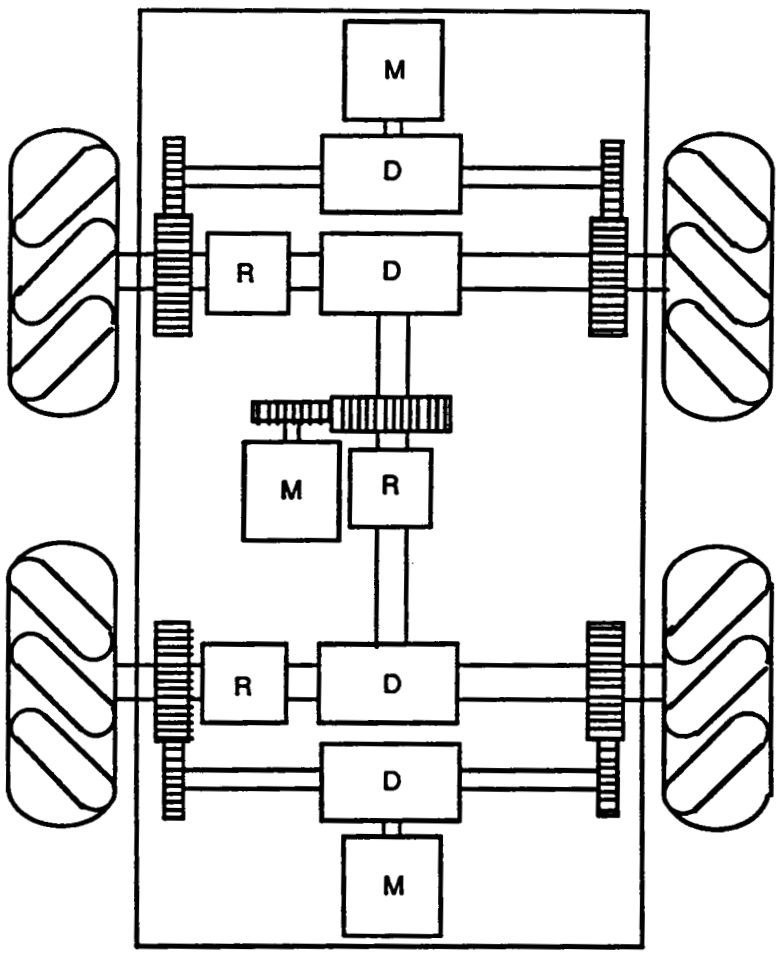


Figure A11.5: Actuating the Robust Uranus Design

A11.5 Neptune

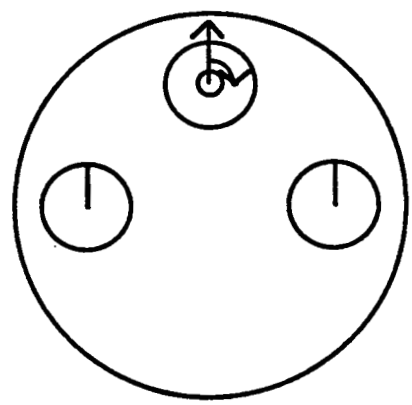
Neptune [Podnar84] has a tricycle-like kinematic structure as depicted in Figure A11.6. The front wheel is steered about its center, and both the steering and the wheel rotations are actuated. The two fixed-orientation wheels are neither actuated nor sensed.

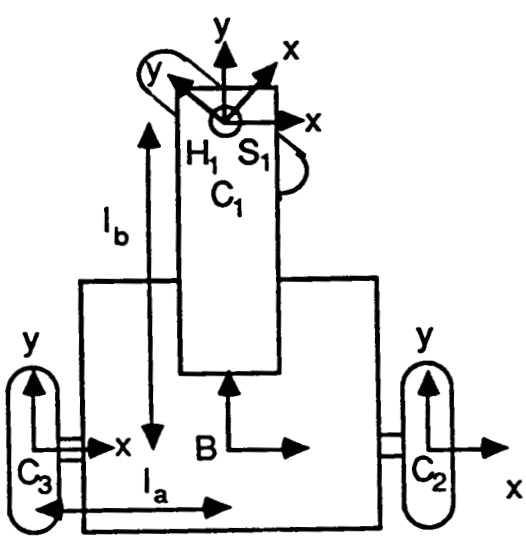
The coordinate transformation matrices are:

$$ {}^{B}T_{H_1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & l_b \\ 0 & 0 & 1 & l_d - l_c \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad {}^{H_1}\Phi_{S_1} = \begin{pmatrix} \cos^{H_1}\theta_{S_1} & -\sin^{H_1}\theta_{S_1} & 0 & 0 \\ \sin^{H_1}\theta_{S_1} & \cos^{H_1}\theta_{S_1} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} $$

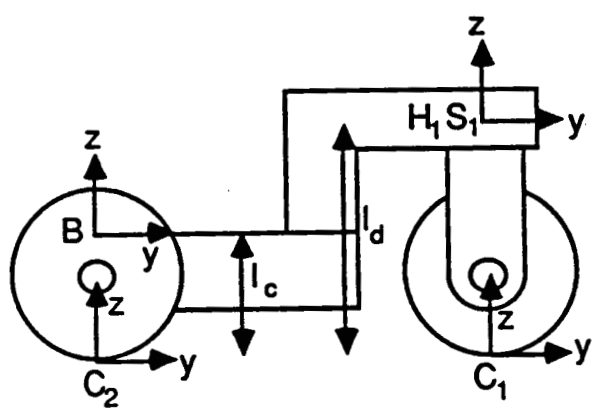
$$ {}^{S_1}T_{C_1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -l_d \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad {}^{B}T_{C_2} = \begin{pmatrix} 1 & 0 & 0 & l_a \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -l_c \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad {}^{B}T_{C_3} = \begin{pmatrix} 1 & 0 & 0 & -l_a \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -l_c \\ 0 & 0 & 0 & 1 \end{pmatrix} $$

Neptune
(Bicun-Unicsan-Whemor)





Top View



Side View

Figure A11.6: Coordinate System Assignments for Neptune

The wheel radii are $R_1=R_2=R_3=R$. We use the Jacobian matrix for a steered conventional wheel in (A3.4) to write the equation for wheel one:

$$\dot{\mathbf{p}} = \begin{pmatrix} -R\sin^{H_1}\theta_{S_1} & l_b & -l_b \\ R\cos^{H_1}\theta_{S_1} & 0 & 0 \\ 0 & 1 & -1 \end{pmatrix} \begin{pmatrix} {}^{A_1}\omega_{W_1x} \\ {}^{\bar{C}_1}\omega_{C_1z} \\ {}^{H_1}\omega_{S_1z} \end{pmatrix} = \mathbf{J}_1 \dot{\mathbf{q}}_1 \qquad \text{(A11.13)}$$

The matrix equations for wheels two and three are specified by (A3.2):

$$\dot{\mathbf{p}} = \begin{pmatrix} 0 & 0 \\ R & -l_a \\ 0 & 1 \end{pmatrix} \begin{pmatrix} {}^{A_2}\omega_{W_2x} \\ {}^{\bar{C}_2}\omega_{C_2z} \end{pmatrix} = \mathbf{J}_2 \dot{\mathbf{q}}_2 \qquad \text{(A11.14)}$$

$$\dot{\mathbf{p}} = \begin{pmatrix} 0 & 0 \\ R & l_a \\ 0 & 1 \end{pmatrix} \begin{pmatrix} {}^{A_3}\omega_{W_3x} \\ {}^{\bar{C}_3}\omega_{C_3z} \end{pmatrix} = \mathbf{J}_3 \dot{\mathbf{q}}_3 \qquad \text{(A11.15)}$$

The soluble motion criterion in (A7.1) is not satisfied because wheel one is redundant. Columns two and three of the Jacobian matrix $\mathbf{J}_1$ are linearly dependent and thus the associated wheel variables (the steering velocity ${}^{H_1}\omega_{S_1z}$ and the wheel rotational slip velocity ${}^{\bar{C}_1}\omega_{C_1z}$) are redundant. The actuated inverse velocity solution is not applicable for Neptune. We cannot determine the actuation and sensing structures because the foundations of the actuation and sensing characterization trees, the robot actuation and sensing equations in (A7.7) and (A7.13) utilize the inverse velocity solution. Furthermore, the WMR DOF computation in (A7.4) does not apply.

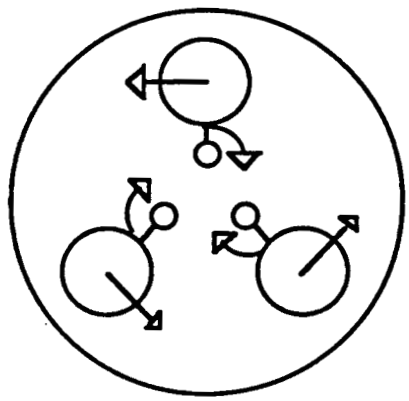
Neptune was constructed to provide a mobile platform for vision research and for that purpose the design is sufficient. From a control engineer's point-of-view, the design is undesirable because the actuated inverse and sensed forward velocity solutions cannot be computed. The redundant wheel disallows these computations. We suggest two practical design alternatives which allow the mobility and computational simplicity of Newt but require few changes to Neptune. First, wheel one can be made non-redundant by offsetting its center from the steering axis. Secondly, wheel one can be offset as in the first alternative, and the steering and drive motors can be moved from wheel one to

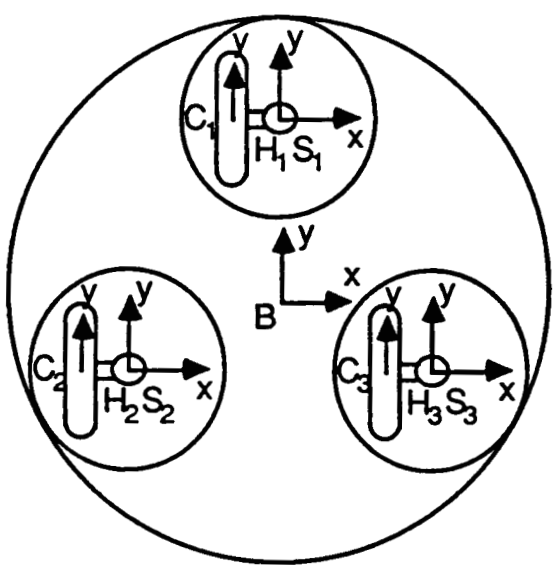
drive wheels two and three producing a structure kinematically identical to Newt.

A11.6 Pluto

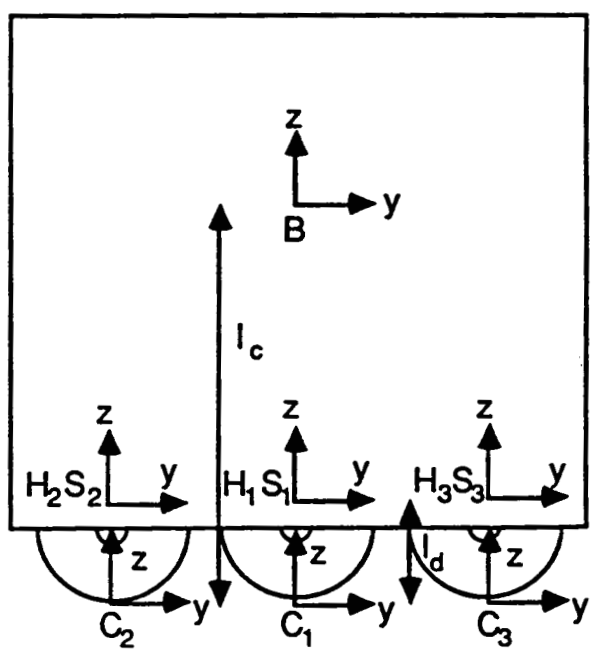
As illustrated in Figure A11.7, Pluto [Moravec83] consists of three conventional steered wheels symmetrically arranged about the center of the WMR body. The steering and drive of each wheel is actuated and sensed.

Pluto
(Tricsas-Whemor)





Top View



Side View

Figure A11.7: Coordinate System Assignments for Pluto

To simplify the coordinate transformation matrices, we have assigned all hip coordinate systems parallel to the WMR body coordinate system and all steering coordinate systems parallel to their respective contact point coordinate systems:

$$ {}^{B}T_{H_1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & l_a \\ 0 & 0 & 1 & l_d - l_c \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad {}^{H_1}\Phi_{S_1} = \begin{pmatrix} \cos^{H_1}\theta_{S_1} & -\sin^{H_1}\theta_{S_1} & 0 & 0 \\ \sin^{H_1}\theta_{S_1} & \cos^{H_1}\theta_{S_1} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} $$

$$ {}^{B}T_{H_2} = \begin{pmatrix} 1 & 0 & 0 & -\sqrt{3}l_a/2 \\ 0 & 1 & 0 & -l_a/2 \\ 0 & 0 & 1 & l_d - l_c \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad {}^{H_2}\Phi_{S_2} = \begin{pmatrix} \cos^{H_2}\theta_{S_2} & -\sin^{H_2}\theta_{S_2} & 0 & 0 \\ \sin^{H_2}\theta_{S_2} & \cos^{H_2}\theta_{S_2} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} $$

$$ {}^{B}T_{H_3} = \begin{pmatrix} 1 & 0 & 0 & \sqrt{3}l_a/2 \\ 0 & 1 & 0 & -l_a/2 \\ 0 & 0 & 1 & l_d - l_c \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad {}^{H_2}\Phi_{S_2} = \begin{pmatrix} \cos^{H_2}\theta_{S_2} & -\sin^{H_2}\theta_{S_2} & 0 & 0 \\ \sin^{H_2}\theta_{S_2} & \cos^{H_2}\theta_{S_2} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} $$

$$ {}^{S_1}T_{C_1} = {}^{S_2}T_{C_2} = {}^{S_3}T_{C_3} = \begin{pmatrix} 1 & 0 & 0 & -l_b \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -l_d \\ 0 & 0 & 0 & 1 \end{pmatrix} $$

The radii are $R_1=R_2=R_3=R$. The wheel equations are written by applying the Jacobian matrix in (A3.4) for steered conventional wheels:

$$ \dot{\mathbf{p}} = \begin{pmatrix} -R\sin^{H_1}\theta_{S_1} & l_a - l_b\sin^{H_1}\theta_{S_1} & -l_a \\ R\cos^{H_1}\theta_{S_1} & l_b\cos^{H_1}\theta_{S_1} & 0 \\ 0 & 1 & -1 \end{pmatrix} \begin{pmatrix} {}^{A_1}\omega_{W_1x} \\ {}^{\bar{C}_1}\omega_{C_1z} \\ {}^{H_1}\omega_{S_1z} \end{pmatrix} = \mathbf{J}_1\,\dot{\mathbf{q}}_1 \qquad \text{(A11.16)} $$

$$ \dot{\mathbf{p}} = \begin{pmatrix} -R\sin^{H_2}\theta_{S_2} & -l_b\sin^{H_2}\theta_{S_2} - l_a/2 & l_a/2 \\ R\cos^{H_2}\theta_{S_2} & l_b\cos^{H_2}\theta_{S_2} + \sqrt{3}l_a/2 & -\sqrt{3}l_a/2 \\ 0 & 1 & -1 \end{pmatrix} \begin{pmatrix} {}^{A_2}\omega_{W_2x} \\ {}^{\bar{C}_2}\omega_{C_2z} \\ {}^{H_2}\omega_{S_2z} \end{pmatrix} = \mathbf{J}_2\,\dot{\mathbf{q}}_2 \qquad \text{(A11.17)} $$

$$\dot{\mathbf{p}} = \begin{pmatrix} -R\sin^{H_3}\theta_{S_3} & -l_b\sin^{H_3}\theta_{S_3} - l_a/2 & l_a/2 \\ R\cos^{H_3}\theta_{S_3} & l_b\cos^{H_3}\theta_{S_3} + \sqrt{3}l_a/2 & \sqrt{3}l_a/2 \\ 0 & 1 & -1 \end{pmatrix} \begin{pmatrix} {}^{A_3}\omega_{W_{3x}} \\ {}^{\bar{C}_3}\omega_{C_{3z}} \\ {}^{H_3}\omega_{S_{3z}} \end{pmatrix} = \mathbf{J}_3 \dot{\mathbf{q}}_3 \quad \text{(A11.18)}$$

The soluble motion criterion in (A7.1) is not satisfied because the wheels are redundant. Consequently, the actuated inverse velocity solution in (2.4.8) is not applicable, the actuation and sensing structures cannot be determined and the sensed forward velocity solution in (2.4.14) cannot be computed. Likewise, the number of DOFs cannot be determined from (A7.4).

We conclude from Pluto that kinematic modeling of a WMR must be addressed in the design stage. Pluto can be redesigned to operate as an omnidirectional WMR by constructing the steering links so that the wheels are non-redundant. Since there are six actuators, the redesigned actuation structure will not be robust and will allow actuator conflict. The Denning Sentry Robot [Wilson85] replicates the kinematic structure of Pluto, with the exception that all three wheels are mechanically steered and driven in unison. The Denning Sentry Robot avoids actuator conflict by utilizing only two actuators and mechanically coupling the wheel motions, but in so doing it sacrifices omnidirectionality.

Our initial research on Pluto [Muir84a] motivated our WMR servo-control research activities. We realized the requirement to integrate the servo-controller design activities with the mechanical, electrical and computer system design. We recognized that the type and placement of wheels, actuators, and sensors should be chosen to provide the mobility characteristics required while minimizing the required servo-controller design efforts. The control processor(s), moreover, must be of sufficient throughput and must be interfaced adequately to the wheel actuators and sensors to allow sufficiently fast communication with sufficiently high precision.

A11.7 Stanford Cart

The Stanford Cart [Moravec80] has the kinematic structure of an automobile, two front wheels with coupled steering angles and two parallel non-

steered back wheels, as shown in Figure A11.8. The rotations of wheels three and four and the coupled steering for wheels one and two are actuated.

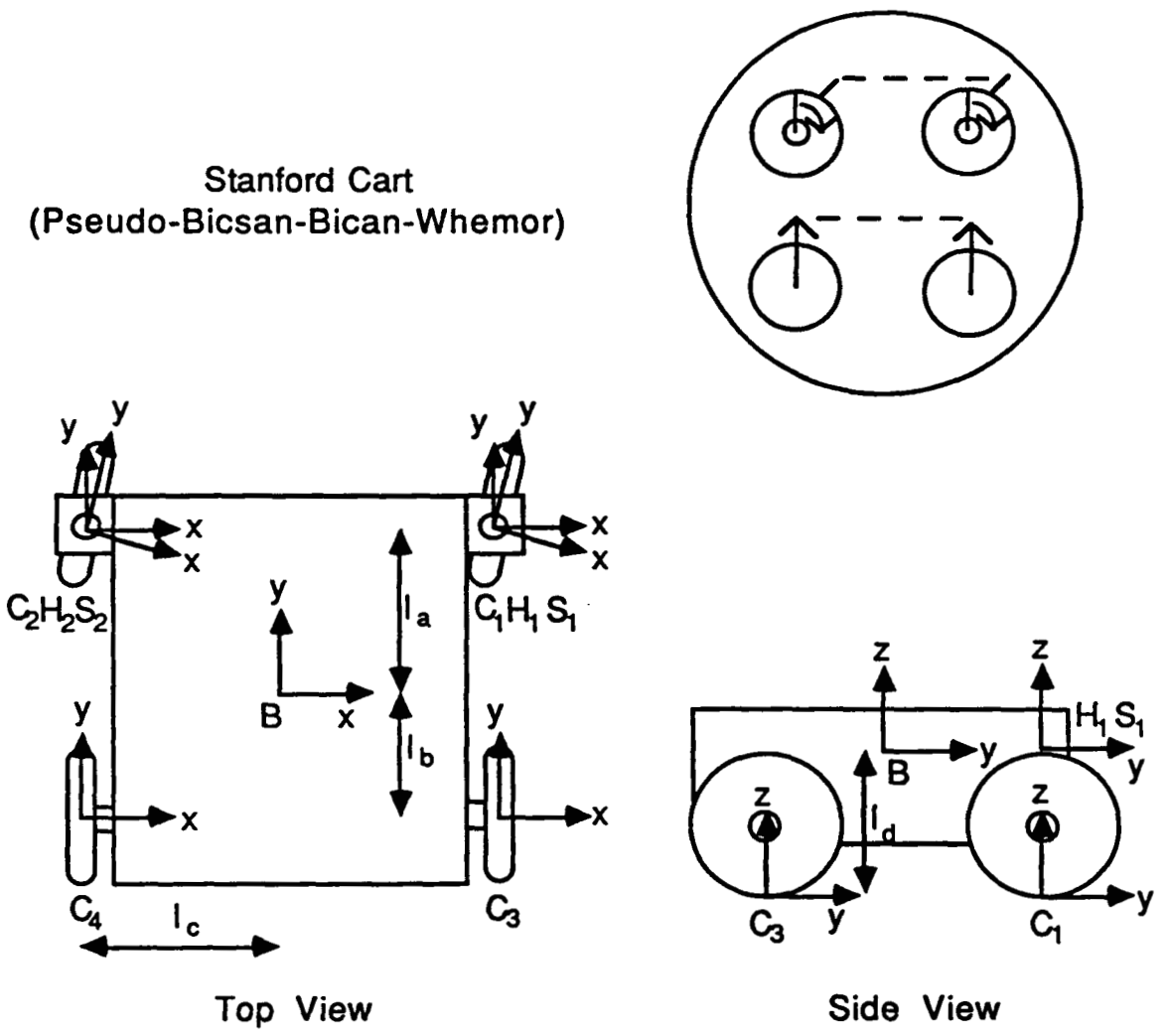


Figure A11.8: Coordinate System Assignements for the Stanford Cart

The coordinate systems assigned in Figure A11.8 lead to the following coordinate transformation matrices:

$$ {}^{B}T_{H_1} = \begin{pmatrix} 1 & 0 & 0 & l_c \\ 0 & 1 & 0 & l_a \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad {}^{H_1}\Phi_{S_1} = \begin{pmatrix} \cos^{H_1}\theta_{S_1} & -\sin^{H_1}\theta_{S_1} & 0 & 0 \\ \sin^{H_1}\theta_{S_1} & \cos^{H_1}\theta_{S_1} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} $$

$$ {}^{B}T_{H_2} = \begin{pmatrix} 1 & 0 & 0 & -l_c \\ 0 & 1 & 0 & l_a \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad {}^{H_2}\Phi_{S_2} = \begin{pmatrix} \cos^{H_2}\theta_{S_2} & -\sin^{H_2}\theta_{S_2} & 0 & 0 \\ \sin^{H_2}\theta_{S_2} & \cos^{H_2}\theta_{S_2} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} $$

$$ {}^{S_1}T_{C_1} = {}^{S_2}T_{C_2} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -l_d \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad {}^{B}T_{H_3} = \begin{pmatrix} 1 & 0 & 0 & l_c \\ 0 & 1 & 0 & -l_b \\ 0 & 0 & 1 & -l_d \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad {}^{B}T_{H_4} = \begin{pmatrix} 1 & 0 & 0 & -l_c \\ 0 & 1 & 0 & -l_b \\ 0 & 0 & 1 & -l_d \\ 0 & 0 & 0 & 1 \end{pmatrix} $$

The equations-of-motion for wheels one and two are written by applying the Jacobian matrix in (A3.4) for steered conventional wheels, and for wheels three and four by applying the Jacobian matrix in (A3.2) for non-steered conventional wheels:

$$ \dot{\mathbf{p}} = \begin{pmatrix} -R\sin^{H_1}\theta_{S_1} & l_a & -l_a \\ R\cos^{H_1}\theta_{S_1} & -l_c & l_c \\ 0 & 1 & -1 \end{pmatrix} \begin{pmatrix} {}^{A_1}\omega_{W_1x} \\ \bar{C}_1\omega_{C_1z} \\ {}^{H_1}\omega_{S_1z} \end{pmatrix} = \mathbf{J}_1 \dot{\mathbf{q}}_1 \qquad \text{(A11.19)} $$

$$ \dot{\mathbf{p}} = \begin{pmatrix} -R\sin^{H_2}\theta_{S_2} & l_a & -l_a \\ R\cos^{H_2}\theta_{S_2} & -l_c & l_c \\ 0 & 1 & -1 \end{pmatrix} \begin{pmatrix} {}^{A_2}\omega_{W_2x} \\ \bar{C}_2\omega_{C_2z} \\ {}^{H_2}\omega_{S_2z} \end{pmatrix} = \mathbf{J}_2 \dot{\mathbf{q}}_2 \qquad \text{(A11.20)} $$

$$ \dot{\mathbf{p}} = \begin{pmatrix} 0 & -l_b \\ R & -l_c \\ 0 & 1 \end{pmatrix} \begin{pmatrix} {}^{A_3}\omega_{W_3x} \\ \bar{C}_3\omega_{C_3z} \end{pmatrix} = \mathbf{J}_3 \dot{\mathbf{q}}_3 \qquad \text{(A11.21)} $$

$$ \dot{\mathbf{p}} = \begin{pmatrix} 0 & -l_b \\ R & l_c \\ 0 & 1 \end{pmatrix} \begin{pmatrix} {}^{A_4}\omega_{W_4x} \\ \bar{C}_4\omega_{C_4z} \end{pmatrix} = \mathbf{J}_4 \dot{\mathbf{q}}_4 \qquad \text{(A11.22)} $$

We assume[1] that the steering angles are equal; i.e., ${}^{H_1}\theta_{S_1}={}^{H_2}\theta_{S_2}=\theta_s$, and consequently ${}^{H_1}\omega_{S_1}={}^{H_2}\omega_{S_2}=\omega_s$. We substitute these equalities into the wheel Jacobian matrices in (A11.19) and (A11.20) to form the composite robot equation in (2.4.2):

$$\mathbf{A}_0\,\dot{\mathbf{p}} = \mathbf{B}_0\,\dot{\mathbf{q}} \qquad \text{(A11.23)}$$

where

$$\mathbf{B}_0 = \begin{vmatrix} -R\sin\theta_s & l_a & -l_a & 0 & 0 & 0 & 0 & 0 & 0 \\ R\cos\theta_s & -l_c & l_c & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -l_a & -R\sin\theta_s & l_a & 0 & 0 & 0 & 0 \\ 0 & 0 & -l_c & R\cos\theta_s & l_c & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -l_b & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & R & -l_c & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -l_b \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & R & l_c \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{vmatrix}$$

and

$$\dot{\mathbf{q}} = \left({}^{A_1}\omega_{W_1x} \quad \bar{C}_1\omega_{C_1z} \quad \omega_s \quad {}^{A_2}\omega_{W_2x} \quad \bar{C}_2\omega_{C_2z} \quad {}^{A_3}\omega_{W_3x} \quad \bar{C}_3\omega_{C_3z} \quad {}^{A_4}\omega_{W_4x} \quad \bar{C}_4\omega_{C_4z} \right)^T.$$

Because of the coupling between wheels one and two, the applicable soluble motion criterion test in (A7.1) is rank[$\mathbf{B}_0$] = w. We observe in (A11.23) that the rank of the (12×9) matrix $\mathbf{B}_0$ is eight (because column two minus column five equals column three), but there are nine wheel variables (i.e., w=9). Accordingly, the mobility structure of the Stanford Cart is not soluble and the inverse and forward velocity solutions are not applicable.

The Stanford Cart is kinematically similar to an automobile. Even though automobiles operate satisfactorily for transportation, we cannot adequately model the motion of the Stanford Cart using only kinematic characteristics. We conclude that a dynamic analysis is required to model its motion.

---

1 The Stanford Cart has an Ackerman steering linkage [McGraw82] between the two front wheels. The Ackerman linkage implements the actuator coupling criterion in (A7.9) by providing the correct wheel angles to avoid translational wheel-slip when the vehicle turns.

A11.8 Concluding Remarks

The six examples presented in this appendix illuminate the kinematic modeling methodology of Chapter 2. Furthermore, we illustrate that writing the equations-of-motion for complex kinematic structures, such as Pluto, is not practical without a systematic framework. The examples show that formulating the equations-of-motion for a WMR is a straightforward procedure which does not require insight into the operation of the WMR.

We note that the actuated inverse and sensed forward velocity solutions are applicable to WMRs which satisfy the soluble motion criterion (the Unimation robot, Newt and Uranus). The WMRs which have redundant wheels (Neptune, Rover, and the Stanford Cart) do not satisfy the soluble motion criterion and the actuated inverse and sensed forward velocity solutions are not applicable. Without these computations, the servo-control of WMRs having redundant wheels is inferior. We conclude that kinematic modeling of a WMR must be undertaken in the design stage (Appendix 8). Since kinematic modeling is critical for WMR servo-control, the design of the wheels and the positioning of the wheels, actuators and sensors must ensure that all of the modeling computations are feasible.

These six examples exhibit noteworthy features. If the wheel variables which are actuated and the wheel variables which are sensed are identical, than either the actuation or the sensing structure can be robust, but not both. For example, the actuation structure of the Unimation WMR is robust and the sensing structure is not; whereas, the sensing structure of Uranus is robust but the actuation structure is not. Since we desire both robust actuation and robust sensing, we should not limit our WMR designs by sensing only the wheel variables that are actuated[2]. When joint-space servo-control is implemented, the actuated wheel variables must be sensed to provide local feedback. For the preferred Cartesian-space servo-control (described in Chapter 4), we provide robust sensing and actuation. By sensing and actuating different wheel variables, we also reduce the mechanical complexity of the hardware. We note

---

2 If brushless motors are utilized as actuators, each actuated wheel variable must be sensed to enable electronic commutation.

further that translational wheel slip is more likely to occur with actuated wheel variables than unactuated ones because the actuated variables are force/torque sources. The effects of translational wheel slip on the computation of the WMR position from wheel sensor measurements are thereby reduced by sensing unactuated wheel variables.

The only WMRs which allow three DOF motion are the ones which consist exclusively of wheels with three DOFs (the Unimation Robot and Uranus). A WMR having non-steered conventional or redundant conventional wheels may be mechanically easier to construct but cannot allow three DOF motion. We suggest that three DOF motion can be practically utilized when the WMR has an onboard manipulator. The mobility of the WMR extends the workspace of the manipulator. When the WMR is for transportation of parts, materials or tools from place to place, only two DOFs are necessary. The mechanically simplest design to provide two DOFs is two diametrically opposed non-steered conventional wheels, as on Newt. Drive motors may be coupled directly to the wheel axles. One or two additional castors are needed for stability. This design also embodies simple and easily computed sensed forward and actuated inverse velocity solutions.

Appendix 12

Kinematic Matrix Templates

In this appendix, we tabulate the (6×6) kinematic matrices developed for the velocity transformations in Section 3.4.2 and the acceleration transformations in Section 3.4.3. We utilize the shorthand notation: $c_i = \cos({}^B\theta_{Ai})$, $s_i = \sin({}^B\theta_{Ai})$ and $d_i = {}^Bd_{Ai}$ for $i = x,y,z$.

Link Jacobian Matrix: ${}^AL_B$

$$\begin{vmatrix} c_yc_z & c_ys_z & -s_y & -c_ys_zd_z+s_yd_y & c_yc_zd_z+s_yd_x & -c_yc_zd_y+c_ys_zd_x \\ s_xs_yc_z-c_xs_z & s_xs_ys_z+c_xc_z & s_xc_y & -s_xs_ys_zd_z-c_xc_zd_z+s_xc_yd_y & s_xs_yc_zd_z-c_xs_zd_z-s_xc_yd_x & -s_xs_yc_zd_y+c_xs_zd_y+s_xs_ys_zd_x+c_xc_zd_x \\ c_xs_yc_z+s_xs_z & c_xs_ys_z-s_xc_z & c_xc_y & -c_xs_ys_zd_z+s_xc_zd_z+c_xc_yd_y & c_xs_yc_zd_z+s_xs_zd_z-c_xc_yd_x & -c_xs_yc_zd_y+c_xs_zd_y-s_xs_ys_zd_x-s_xc_zd_x \\ 0 & 0 & 0 & c_yc_z & c_ys_z & -s_y \\ 0 & 0 & 0 & s_xs_yc_z-c_xs_z & s_xs_ys_z+c_xc_z & s_xc_y \\ 0 & 0 & 0 & c_xs_yc_z+s_xs_z & c_xs_ys_z-s_xc_z & c_xc_y \end{vmatrix}$$

Joint Jacobian Matrix: ${}^AJ_B$

$$\begin{vmatrix} c_yc_z & c_ys_z & -s_y & 0 & 0 & 0 \\ s_xs_yc_z-c_xs_z & s_zs_ys_z+c_zc_z & s_xc_y & 0 & 0 & 0 \\ c_xs_yc_z+s_xs_z & c_xs_ys_z-s_xc_z & c_xc_y & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -s_y \\ 0 & 0 & 0 & 0 & c_x & s_xc_y \\ 0 & 0 & 0 & 0 & -s_x & c_xc_y \end{vmatrix}$$

Rotational Velocity Product Matrix: ${}^AR_B$

$$\begin{vmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ s_y & 0 & -s_y & -1 & 0 & 1 \\ -s_xc_y & c_x & s_xc_y & 0 & -c_x & 0 \\ -c_xc_y & -s_x & c_xc_y & 0 & s_x & 0 \end{vmatrix}$$

Squared Velocity Product Matrix: ${}^{A}S_{B}$

$$
\begin{bmatrix}
-c_y s_z d_y + s_y d_z & -c_y c_z d_x + s_y d_z & -c_y c_z d_x - c_y s_z d_y & c_y s_z d_x - c_y c_z d_y & c_y c_z d_z - s_y d_x & c_y s_z d_z - s_y d_y \\
\begin{pmatrix} s_x s_y s_z d_y \\ -c_x c_z d_y \\ -s_x c_y d_z \end{pmatrix} & \begin{pmatrix} s_x s_y c_z d_x \\ +c_x s_z d_x \\ -s_x c_y d_z \end{pmatrix} & \begin{pmatrix} s_x s_y c_z d_x \\ +c_x s_z d_x \\ -s_x s_y s_z d_y \\ -c_x c_z d_y \end{pmatrix} & \begin{pmatrix} s_x s_y c_z d_y \\ -c_x s_z d_y \\ +s_x s_y s_z d_x \\ +c_x c_z d_x \end{pmatrix} & \begin{pmatrix} s_x s_y c_z d_z \\ -c_x s_z d_z \\ +s_x c_y d_x \end{pmatrix} & \begin{pmatrix} s_x s_y s_z d_z \\ +c_x c_z d_z \\ +s_x c_y d_y \end{pmatrix} \\
\begin{pmatrix} c_x s_y s_z d_y \\ +s_x c_z d_y \\ -c_x c_y d_z \end{pmatrix} & \begin{pmatrix} c_x s_y c_z d_x \\ -s_x s_z d_x \\ -c_x c_y d_z \end{pmatrix} & \begin{pmatrix} c_x s_y c_z d_x \\ -s_x s_z d_x \\ -c_x s_y s_z d_y \\ +s_x c_z d_y \end{pmatrix} & \begin{pmatrix} c_x s_y c_z d_y \\ +s_x s_z d_y \\ +c_x s_y s_z d_x \\ -s_x c_z d_x \end{pmatrix} & \begin{pmatrix} c_x s_y c_z d_z \\ +s_x s_z d_z \\ +c_x c_y d_x \end{pmatrix} & \begin{pmatrix} c_x s_y s_z d_z \\ -s_x c_z d_z \\ +c_x c_y d_y \end{pmatrix} \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
$$

Translational Velocity Product Matrix: ${}^{A}T_{B}$

$$
\begin{bmatrix}
-s_y & -c_y s_z & s_y & c_y c_z & c_y s_z & -c_y c_z \\
s_x c_y & -s_x s_y s_z - c_x c_z & -s_x c_y & s_x s_y c_z - c_x s_z & s_x s_y s_z + c_x c_z & c_x s_z - s_x s_y c_z \\
c_x c_y & c_x s_y s_z - s_x c_z & -c_x c_y & c_x s_y c_z + s_x s_z & c_x s_y s_z - s_x c_z & -s_x s_z - c_x s_y c_z \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
$$

Appendix 13

Flexibility Modeling

A13.1 Overview

A salient feature of our dynamic modeling framework in Chapter 3 is that the three conceptually distinct components of the dynamic model: force/torques, kinematic transformations and frictional couplings, are modeled independently. The composite dynamic model of a robotic mechanism is then computed by combining the independent component models. The force/torque model is the keystone of the composite dynamic model. The kinematic transformations are then required to compute the unknown kinematic variables (i.e., velocities and accelerations within instantaneously coincident coordinate systems) from known variables (i.e., the velocities and accelerations of the main body and the joints); and the frictional coupling models are required to compute the unknown frictional parameters (i.e., the dry frictional proportionality factors) from known quantities (i.e., coefficients of friction, internal forces/torques and joint velocities).

In this appendix, we show how our dynamic modeling framework may be *extended* to model such phenomena as flexible links by computing unknown variables within our force/torque model from known quantities. Our force/torque and kinematic models in Chapter 3 assume that all links within the robotic mechanism are rigid. Consequently, the dimensions of the links and thus the position/orientation ${}^{A_1}\mathbf{p}_{A_2}$ of coordinate system A_2 fixed to link *A* relative to coordinate system A_1 which is also fixed to link *A* are constant. If link *A* is flexible, we must model the position/orientation ${}^{A_1}\mathbf{p}_{A_2}$ between coordinate systems A_1 and A_2 as functions of the forces/torques acting on link A.

We model flexible links as having the characteristic properties of a spring, a damper or a combination (e.g., series, parallel or series-parallel) of springs and dampers. We sketch the models for spring flexures in Section A13.2 and damper flexures in Section A13.3.

A13.2 Spring Flexures

An ideal mechanical spring displays the characteristic linear relationship between the applied force (torque) (f,τ) and its linear (angular) displacement $(\delta d,\delta\theta)$ *from* the nominal linear (angular) length (d^0,θ^0) in one dimension according to *Hooke's law*:

$$(f,\tau) = k_{sA}\,(\delta d,\delta\theta) \qquad \text{(A13.1)}$$

The spring coefficient k_{sA} for link *A* in (A13.1) is *Young's modulus* and is dependent upon the material and its nominal dimensions [Sears77]. We extend Hooke's law to model the deformation of a six-dimensional spring by replacing the scalar force (torque), Young's modulus and the linear (angular) displacements in (A13.1) with their six-dimensional counterparts. We model the flexibility of a spring flexure by computing the position/orientation vector ${}^{A_1}\mathbf{p}_{A_2}$ of coordinate system A_2 fixed to one end of flexible link *A* relative to coordinate system A_1 fixed to the opposite end of link *A*:

$${}^{A_1}\mathbf{p}_{A_2} = {}^{A_1}\mathbf{p}^0_{A_2} + {}^{A_1}\mathbf{K}_{sA}\left({}^{V(A)}\mathbf{P}_{A_1}\,{}^{A_1}\mathbf{f}_A - {}^{V(A)}\mathbf{P}_{A_2}\,{}^{A_2}\mathbf{f}_A\right). \qquad \text{(A13.2)}$$

In (A13.2), we compute the position six-vector ${}^{A_1}\mathbf{p}_{A_2}$ (i.e., the six-dimensional "length" of the six-dimensional "spring" link) as the sum of the nominal position six-vector ${}^{A_1}\mathbf{p}^0_{A_2}$ (i.e., the nominal six-dimensional link length) and a differential displacement obeying Hooke's law. The net force/torque on link *A* is computed by differencing the forces/torques at each end of link *A* within the common center-of-volume coordinate system V(*A*). The six-dimensional differential displacement is computed as the product of the net force/torque and the (6×6) *spring* matrix ${}^{A_1}\mathbf{K}_{sA} = \mathbf{diag}[{}^{A_1}k_{sAi}]$ containing elements which are the reciprocals of Young's modulus for each axis.

We note that a changing distribution of the mass of link A accompanies its changing dimensions. If a sufficiently accurate dynamic model is required, the changing distribution of mass must also be modeled by computing the inertial matrix ${}^{M(A)}\mathbf{I}_A$ from the spatial mass density and dimensions of link *A* as it flexes.

A13.3 Damper Flexures

An ideal mechanical damper displays the characteristic linear relationship between the applied force (torque) (f,τ) and its linear (angular) velocity (v,ω) of one end of the damper relative to the other end in one dimension according to:

$$(f,\tau) = k_{dA}\,(v,\omega) \tag{A13.3}$$

The coefficient k_{dA} in (A13.3.) is a constant damping coefficient for link *A* which is dependent upon the material and its nominal dimensions. We extend (A13.3) to model the deformation of a six-dimensional damper by replacing the scalar force (torque), damping coefficient and linear (angular) velocities in (A13.3) with their six-dimensional counterparts. We model the flexibility of a damper flexure by computing the velocity vector ${}^{A_1}\mathbf{v}_{A_2}$ of coordinate system A_2 fixed to one end of flexible link *A* relative to coordinate system A_1 fixed to the opposite end of link *A*:

$${}^{A_1}\mathbf{v}_{A_2} = {}^{A_1}\mathbf{K}_{dA}\left({}^{V(A)}\mathbf{P}_{A_1}\,{}^{A_1}\mathbf{f}_A - {}^{V(A)}\mathbf{P}_{A_2}\,{}^{A_2}\mathbf{f}_A\right). \tag{A13.4}$$

In (A13.4), we compute the velocity of deformation ${}^{A_1}\mathbf{v}_{A_2}$ of link *A* as the product of the net force/torque on link *A* within the common center-of-volume coordinate system V(*A*) and the (6×6) *damping* matrix ${}^{A_1}\mathbf{K}_{dA} = \mathbf{diag}[{}^{A_1}k_{dAi}]$ which consists of elements that are the reciprocals of the damping coefficients for each axis. To implement the damper model in (A13.4) with our force/torque models in Chapter 3, we integrate (A13.4) to compute the position six-vector ${}^{A_1}\mathbf{p}_{A_2}$. Since the model is implemented in discrete time on a digital computer, the integral of the generalized damper model in (A13.4) becomes the linear difference equation:

$${}^{A_1}\mathbf{p}_{A_2}(nT) = {}^{A_1}\mathbf{p}_{A_2}[(n-1)T] + T\,{}^{A_1}\mathbf{K}_{sA}\left({}^{V(A)}\mathbf{P}_{A_1}\,{}^{A_1}\mathbf{f}_A(nT) - {}^{V(A)}\mathbf{P}_{A_2}\,{}^{A_2}\mathbf{f}_A(nT)\right) \tag{A13.5}$$

where T is the sampling period, n is the sampling period index and ${}^{A_1}\mathbf{p}_{A_2}(nT)$ and ${}^{A_1}\mathbf{f}_A(nT)$ are the sampled values of ${}^{A_1}\mathbf{p}_{A_2}$ and ${}^{A_1}\mathbf{f}_A$ at the instant t=nT. In parallel with spring flexures (in Section A13.2), the changing distribution of mass must also be modeled by computing the inertial matrix ${}^{M(A)}\mathbf{I}_A$ from the spatial mass density and dimensions of link *A* to obtain an accurate dynamic model.

A13.4 Concluding Remarks

We develop models of flexible links in this appendix independently from the force/torque, kinematic, and frictional coupling models of Chapter 3. We describe the natural incorporation of these models into the dynamic modeling framework in Chapter 3 to illustrate its extensibility to flexible robotic mechanism characteristics. We foresee that such phenomena as temperature variations in link dimensions, joint lubrication, plastic deformation, elastic hysteresis and wear may similarly be modeled for enhanced dynamic models of robotic mechanisms.

Appendix 14

Graphical Dynamic Model Representation

In this appendix, we display graphically force/torque transformations in robotic mechanisms. The computations required for our dynamic modeling framework are predominantly matrix-matrix and matrix-vector multiplications. The multiplication of a (6×6) matrix and a six-vector (e.g., the link force/torque propagation relation in (3.3.14)) can be represented by a directed graph. In Figure A14.1, we depict the propagation of the forces/torques ${}^{A_2}\mathbf{f}_A$ in coordinate system A_2 to the forces/torques ${}^{B_1}\mathbf{f}_A$ in coordinate system B_1.

In Figure A14.1, the six scalar components of the force/torque vectors are represented by nodes on the left and right of the graph. Scalar multiplication of a force/torque component is represented by an arrow directed from the multiplicand on the left to a node on the right. The multipliers e_{ij} shown above each arrow correspond to the elements of the transposed link Jacobian matrix: ${}^{A_2}\mathbf{L}_{B_1}^T=[e_{ij}]$. Each node of the resultant force/torque vector ${}^{B_1}\mathbf{f}_A$ on the right sums the products represented by all arrows directed to it. A further understanding of the link force/torque propagation is realized in Figure A14.2 by separating the link force/torque propagation relation in (3.3.14) into six propagations, one for each element of the position vector ${}^{B_1}\mathbf{p}_{A_2}$.

An entire robotic mechanism may be represented graphically following the force/torque propagation templates in Figure A14.1 and A14.2. Velocity and acceleration transformations are represented similarly by directed graphs. The graphical representations allow the application of graph-theoretic techniques to the analysis of robotic mechanisms. Moreover, we recognize that graphical representations of robotic mechanisms provide physical insights into their operation.

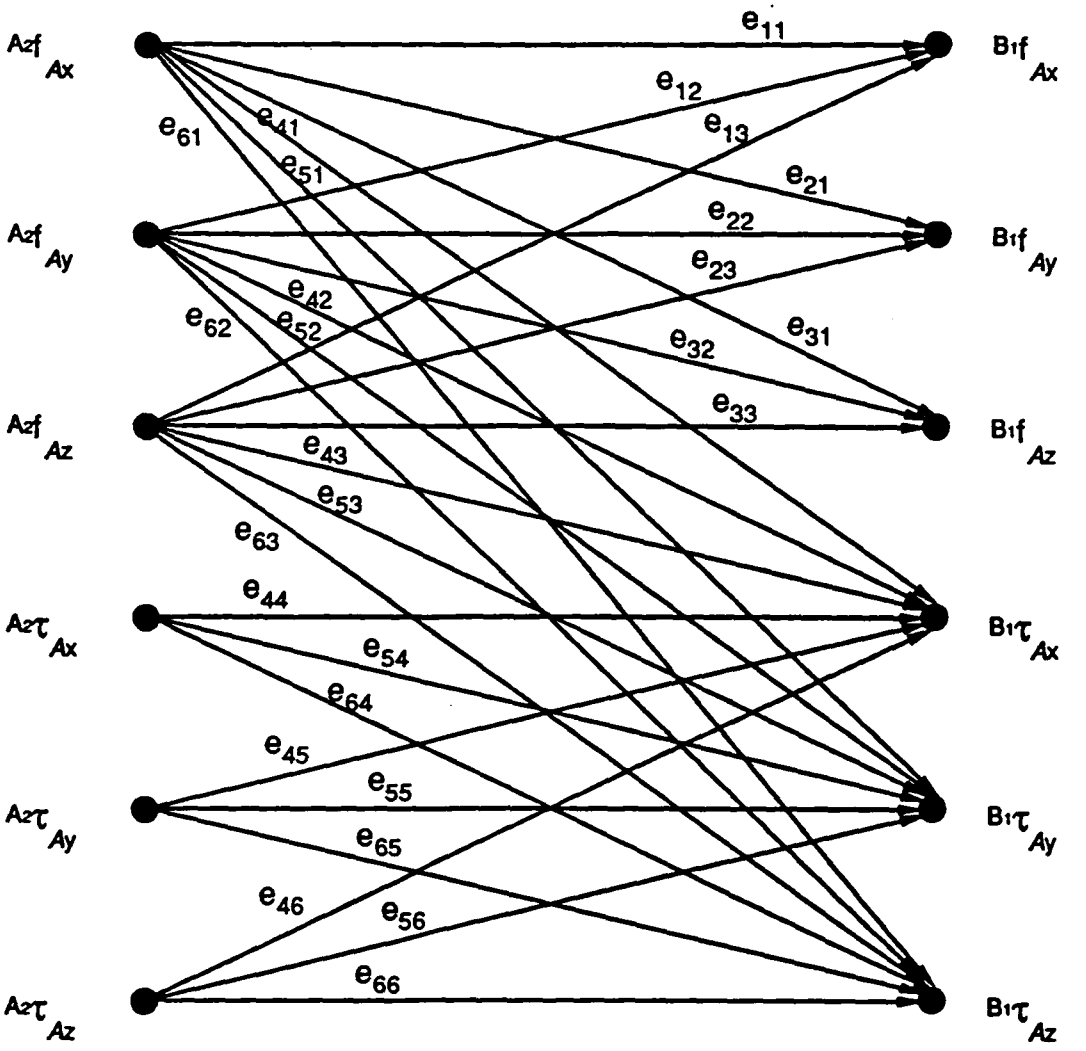


Figure A14.1: Graphical Representation of Link Force/Torque Propagation

The parallel structure of the graphical representations in Figures A14.1 and A14.2 suggest that an equivalent amount of parallelism exists in the computation of the dynamic model. The parallel computational structure in Figure A14.2 is that of a neural network [Lippman87]. Dynamic robot models, dynamics-based servo-controllers and dynamic simulation may thus be implemented on special-purpose parallel processors such as neural networks at significantly increased speeds than are currently realized. The analog implementation of such parallel structures introduces the possibility of real-time servo-control and simulation of a robotic mechanism irrespective of the complexity of the dynamic robot model.

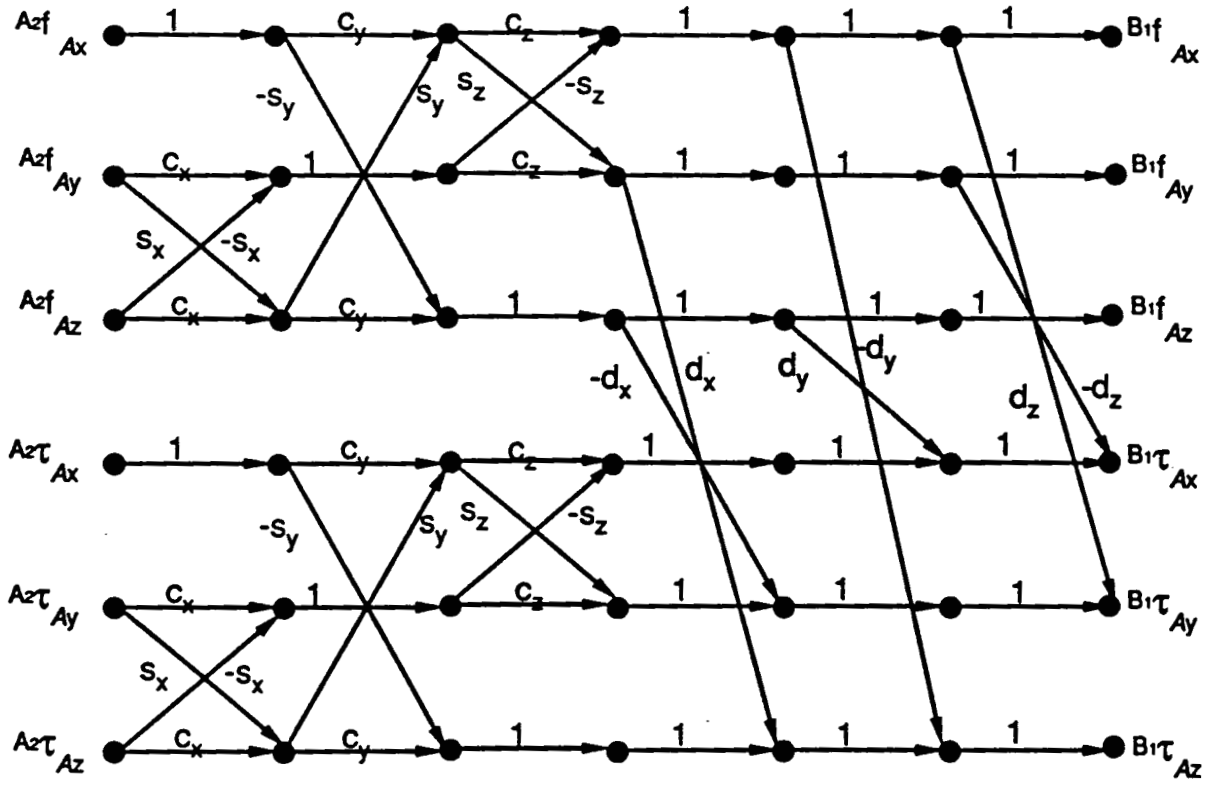


Figure A14.2: Expanded Graphical Representation of Link Force/Torque Propagation

Appendix 15

Dynamic Modeling Examples

A15.1 Overview

In this appendix [Muir88a], we apply our dynamic modeling framework (described in Chapter 3) to formulate the dynamic models of a planar double pendulum (in Section A15.2) and a biped in the frontal plane (in Section A15.3). We show agreement between our dynamic models and the dynamic models formulated by [Murray86] and [Hemami79], respectively, using Lagrangian and Newtonian based methods.

A15.2 Planar Double Pendulum

We apply our dynamics framework to the planar double pendulum (an *open-chain* mechanism) sketched in Figure A15.1.

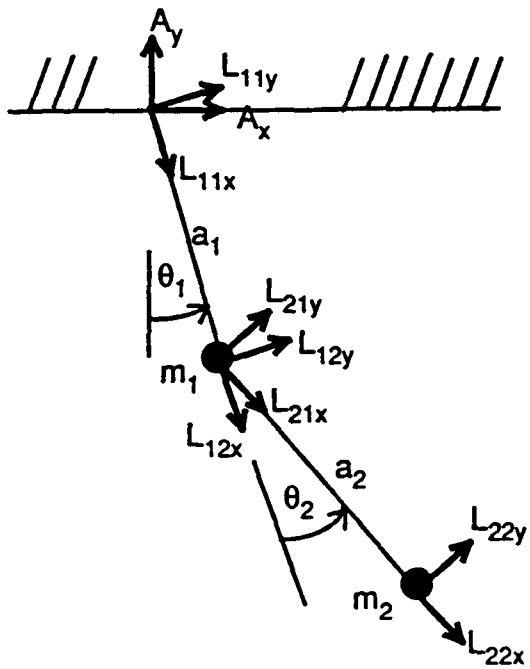


Figure A15.1: A Planar Double Pendulum

We assign coordinate systems L_{11} and L_{12} fixed to link L_1, L_{21} and L_{22} fixed to link L_2 and A fixed to a reference body E. The *four* position six-vectors of the planar double pendulum are:

$${}^{A}\mathbf{p}_{L_{11}} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & (\theta_1 - 90°) \end{pmatrix}^T, \qquad {}^{L_{11}}\mathbf{p}_{L_{12}} = \begin{pmatrix} a_1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}^T,$$

$${}^{L_{12}}\mathbf{p}_{L_{21}} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & \theta_2 \end{pmatrix}^T, \quad \text{and} \quad {}^{L_{21}}\mathbf{p}_{L_{22}} = \begin{pmatrix} a_2 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}^T;$$

the joint coupling matrices, assuming no dry friction, are

$${}^{A}\mathbf{C}_{EL_1} = {}^{L_{12}}\mathbf{C}_{L_1L_2} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix};$$

and the *nine* force/torque vectors are

$${}^{L_{12}}\mathbf{f}_{iL_1} = \begin{pmatrix} -m_1 {}^{\bar{L}_{12}}a_{L_{12x}} & -m_1 {}^{\bar{L}_{12}}a_{L_{12y}} & 0 & 0 & 0 & 0 \end{pmatrix}^T$$

$${}^{L_{22}}\mathbf{f}_{iL_2} = \begin{pmatrix} -m_1 {}^{\bar{L}_{22}}a_{L_{22x}} & -m_1 {}^{\bar{L}_{22}}a_{L_{22y}} & 0 & 0 & 0 & 0 \end{pmatrix}^T \qquad {}^{A}\mathbf{f}_{eL_1} = \begin{pmatrix} f_x & f_y & 0 & 0 & 0 & \tau_z \end{pmatrix}^T$$

$${}^{L_{12}}\mathbf{f}_{gL_1} = \begin{pmatrix} 0 & -m_1 g & 0 & 0 & 0 & 0 \end{pmatrix}^T \qquad {}^{L_{22}}\mathbf{f}_{gL_2} = \begin{pmatrix} 0 & -m_2 g & 0 & 0 & 0 & 0 \end{pmatrix}^T$$

$${}^{L_{11}}\mathbf{f}_{aL_1} = -{}^{A}\mathbf{f}_{aL_1} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & \tau_1 \end{pmatrix}^T \qquad \text{and} \qquad {}^{L_{21}}\mathbf{f}_{aL_2} = -{}^{L_{21}}\mathbf{f}_{aL_2} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & \tau_2 \end{pmatrix}^T.$$

The constants m_1 and m_2 are the point masses at the ends of links L_1 and L_2, respectively, g is the gravitational constant, and τ_1 and τ_2 are the actuator torques applied at the bases of links L_1 and L_2, respectively. Each actuator produces a torque on the link it is driving and an equal and opposite reactional torque on the link to which it is mounted. We obtain the six primary force/torque equations-of-motion by propagating all of the forces/torques to the end of the pendulum (coordinate system L_{22}), and one secondary force/torque equation-of-motion at each joint; i.e., the coordinate systems L_{12} and A. The six primary force/torque equations-of-motion lead to the three non-trivial scalar force/torque equations-of-motion:

$$-c_2 m_1 {}^{\bar{L}_{12}}a_{L_{12x}} - s_2 m_1 {}^{\bar{L}_{12}}a_{L_{12y}} - m_2 {}^{\bar{L}_{22}}a_{L_{22x}} + c_{12} m_1 g + c_{12} m_2 g + s_{12} f_x - c_{12} f_y = 0 \quad \text{(A15.1)}$$

$$s_2 m_1 {}^{\bar{L}_{12}}a_{L_{12x}} - c_2 m_1 {}^{\bar{L}_{12}}a_{L_{12y}} - m_2 {}^{\bar{L}_{22}}a_{L_{22y}} - s_{12} m_1 g - s_{12} m_2 g + c_{12} f_x + s_{12} f_y = 0 \quad \text{(A15.2)}$$

$$-a_2 s_2 m_1 {}^{\bar{L}_{12}}a_{L_{12x}} + a_2 c_2 m_1 {}^{\bar{L}_{12}}a_{L_{12y}} + a_2 s_{12} m_1 g + \tau_2 - a_2 c_{12} f_x - a_2 s_{12} f_y = 0 \quad \text{(A15.3)}$$

where $s_i = \sin(\theta_i)$, $c_i = \cos(\theta_i)$, $s_{ij} = \sin(\theta_i+\theta_j)$, and $c_{ij} = \cos(\theta_i+\theta_j)$. The two secondary force/torque equations-of-motion are:

$$\tau_1 - \tau_2 - a_1 c_1 f_x - a_1 s_1 f_y = 0 \quad \text{(A15.4)}$$

and

$$-\tau_1 + \tau_z = 0 \ . \quad \text{(A15.5)}$$

The four acceleration transformation equations required to complete (A15.1)-(A15.5) are:

$$ {}^{\bar{L}_{12}}a_{L_{12x}} = -a_1 \omega_1^2 \quad \text{(A15.6)}$$

$$ {}^{\bar{L}_{12}}a_{L_{12y}} = a_1 \alpha_1 \quad \text{(A15.7)}$$

$$ {}^{\bar{L}_{22}}a_{L_{22x}} = -a_1 c_2 \omega_1^2 + a_1 s_2 \alpha_1 - a_2(\omega_1 + \omega_2)^2 \quad \text{(A15.8)}$$

and

$$ {}^{\bar{L}_{22}}a_{L_{22y}} = a_1 s_2 \omega_1^2 + a_1 c_2 \alpha_1 + a_2(\alpha_1 + \alpha_2) \ . \quad \text{(A15.9)}$$

In (A15.6)-(A15.9), ω_1 and ω_2 are the angular velocities and α_1 and α_2 are the angular accelerations of the joints. We solve (A15.1)-(A15.9) for the two joint torques τ_1 and τ_2 and obtain the classical inverse dynamic model of the planar double pendulum [Murray86].

A15.3 Biped in the Frontal Plane

We next apply our dynamics framework to the biped in the frontal plane (a *closed-chain* mechanism) depicted in Figure A15.2 [Hemami79].

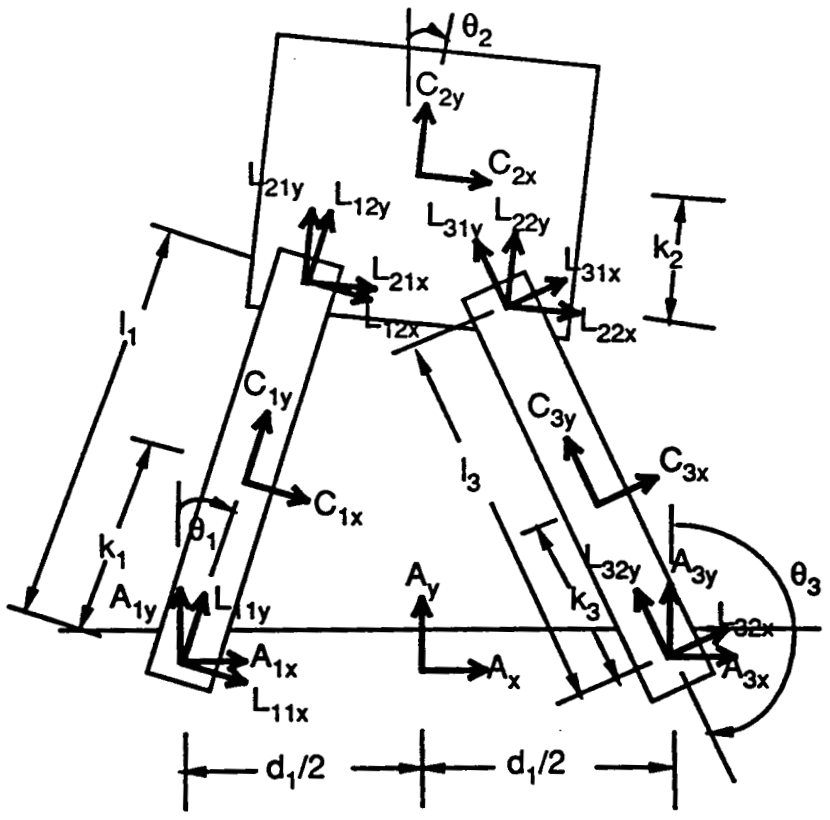


Figure A15.2: Biped in the Frontal Plane

We assign the coordinate systems L_{11}, L_{12}, and C_1 fixed to Link L_1; L_{21}, L_{22}, and C_2 fixed to main link L_2; L_{31}, L_{32}, and C_3 fixed to link L_3; and A_1, A_3, and A fixed to the reference body *E*. The *twelve* position six-vectors of the biped are:

$${}^{A}\mathbf{p}_{A_1} = \begin{pmatrix} -d_1/2 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}^T \qquad {}^{A_1}\mathbf{p}_{L_{11}} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & -\theta_1 \end{pmatrix}^T$$

$${}^{L_{11}}\mathbf{p}_{C_1} = \begin{pmatrix} 0 & k_1 & 0 & 0 & 0 & 0 \end{pmatrix}^T \qquad {}^{C_1}\mathbf{p}_{L_{12}} = \begin{pmatrix} 0 & (l_1-k_1) & 0 & 0 & 0 & 0 \end{pmatrix}^T$$

$${}^{L_{12}}\mathbf{p}_{L_{21}} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & (\theta_1-\theta_2) \end{pmatrix}^T \qquad {}^{L_{21}}\mathbf{p}_{C_2} = \begin{pmatrix} d_2 & k_2 & 0 & 0 & 0 & 0 \end{pmatrix}^T$$

$${}^{A}\mathbf{p}_{A_3} = \begin{pmatrix} d_1/2 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}^T \qquad {}^{A_3}\mathbf{p}_{L_{32}} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & (180°-\theta_3) \end{pmatrix}^T$$

$${}^{L_{32}}\mathbf{p}_{C_3} = \begin{pmatrix} 0 & k_3 & 0 & 0 & 0 & 0 \end{pmatrix}^T \qquad {}^{C_3}\mathbf{p}_{L_{31}} = \begin{pmatrix} 0 & (l_3-k_3) & 0 & 0 & 0 & 0 \end{pmatrix}^T$$

$${}^{L_{31}}\mathbf{p}_{L_{32}} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & (\theta_3-\theta_2-180°) \end{pmatrix}^T \quad \text{and} \quad {}^{L_{22}}\mathbf{p}_{C_2} = \begin{pmatrix} -d_2 & k_2 & 0 & 0 & 0 & 0 \end{pmatrix}^T.$$

The coupling matrices, assuming no dry friction, are:

$$ {}^{A_1}C_{EL_1} = {}^{L_{12}}C_{L_1L_2} = {}^{L_{31}}C_{L_2L_3} = {}^{A_3}C_{EL_3} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}. $$

And the *sixteen* force/torque vectors are:

$$ {}^{C_1}f_{iL_1} = \left(-m_1 {}^{\bar{C}_1}a_{C_1x} \;\; -m_1 {}^{\bar{C}_1}a_{C_1y} \;\; -m_1 {}^{\bar{C}_1}a_{C_1z} \;\; 0 \;\; 0 \;\; -I_1 {}^{\bar{C}_1}\alpha_{C_1z} \right)^T $$

$$ {}^{C_2}f_{iL_2} = \left(-m_2 {}^{\bar{C}_2}a_{C_2x} \;\; -m_2 {}^{\bar{C}_2}a_{C_2y} \;\; -m_2 {}^{\bar{C}_2}a_{C_2z} \;\; 0 \;\; 0 \;\; -I_2 {}^{\bar{C}_2}\alpha_{C_2z} \right)^T $$

$$ {}^{C_3}f_{iL_3} = \left(-m_3 {}^{\bar{C}_3}a_{C_3x} \;\; -m_3 {}^{\bar{C}_3}a_{C_3y} \;\; -m_3 {}^{\bar{C}_3}a_{C_3z} \;\; 0 \;\; 0 \;\; -I_3 {}^{\bar{C}_3}\alpha_{C_3z} \right)^T $$

$$ {}^{G(L_1)}f_{gL_1} = \left(0 \;\; -m_1g \;\; 0 \;\; 0 \;\; 0 \;\; 0\right)^T \qquad {}^{G(L_2)}f_{gL_2} = \left(0 \;\; -m_2g \;\; 0 \;\; 0 \;\; 0 \;\; 0\right)^T $$

$$ {}^{G(L_3)}f_{gL_3} = \left(0 \;\; -m_3g \;\; 0 \;\; 0 \;\; 0 \;\; 0 \right)^T \qquad {}^{A_1}f_{eL_1} = \left(F_1 \;\; G_1 \;\; 0 \;\; 0 \;\; 0 \;\; H_1 \right)^T $$

$$ {}^{A_3}f_{eL_3} = \left(-F_4 \;\; -G_4 \;\; 0 \;\; 0 \;\; 0 \;\; H_4 \right)^T \qquad {}^{A_1}f_{aE} = -{}^{L_{11}}f_{aL_1} = \left(0 \;0 \;0 \;0 \;0 \;u_1\right)^T $$

$$ {}^{L_{12}}f_{aL_1} = -{}^{L_{21}}f_{aL_2} = \left(0 \;0 \;0 \;0 \;0 \;u_2\right)^T \qquad {}^{L_{22}}f_{aL_2} = -{}^{L_{31}}f_{aL_3} = \left(0 \;0 \;0 \;0 \;0 \;u_3\right)^T $$

and $$ {}^{L_{32}}f_{aL_3} = -{}^{A_3}f_{aE} = \left(0 \;0 \;0 \;0 \;0 \;u_4\right)^T. $$

The principal moment of inertia of link i about the z-axis is I_i, the joint actuator torques are u_i, the environmental forces in the x and y directions are F_i and G_i, respectively, and the environmental torque about the z-axis is H_i. The coordinate systems $G(L_i)$ for i=1,2,3 which are not explicitly drawn in Figure A15.2 are gravitational coordinate systems located at the center-of-mass of link L_i and aligned with the gravitational field.

We obtain the six primary force/torque equations-of-motion by propagating the sixteen forces/torques to the center-of-mass of the biped main link (coordinate system C_2), and one secondary force/torque equation-of-motion at each of the four joints; i.e., at the four coordinate systems L_{12}, L_{31}, A_1, and A_3. We then substitute the velocity and acceleration transformations into the

force/torque equations-of-motion. Finally, we apply the two positional geometries

$$l_1 s_1 + 2d_2 c_2 + l_3 s_3 = d_1 \qquad \text{and} \qquad l_1 c_1 - 2d_2 s_2 + l_3 c_3 = 0$$

and their derivatives to obtain the Hemami and Wyman dynamic equations-of-motion for the biped in the frontal plane [Hemami79].

A15.4 Concluding Remarks

We have modeled the dynamics of two computationally simple mechanical systems: a planar double pendulum (an open-chain), and a biped in the frontal plane (a closed-chain). Our dynamic models are identical to those developed by the Lagrangian and Newton-Euler methods, respectively, in the literature. The simplicity of these examples allows insights into the application of our dynamic modeling framework introduced in Chapter 3.

References

[Adams84], R. Adams, "Patent Probe: Omnidirectional Vehicle," *Robotics Age*, Vol. 6, No. 2, February 1984, pp. 21-22.

[Alvema82], Alvema Rehab, "Wheelon - the New Movement," (advertisement), P.O. Box 17017, S-16117 Bromma, Sweden.

[Angeles82], J. Angeles, *Spatial Kinematic Chains, Analysis, Synthesis, Optimization*, Springer-Verlag, Berlin, 1982.

[Balmer82], C. Balmer Jr., "Avatar: A Home Built Robot," *Robotics Age*, Vol. 4, No. 1, January/February 1982, pp. 20-27.

[Bejczy74], A. K. Bejczy, "Robot Arm Dynamics and Control," Technical Memorandum 33-669, Jet Propulsion Laboratory, Pasadena, CA, February 1974.

[Beggs66], J. S. Beggs, *Advanced Mechanism*, The Macmillan Company, New York, 1966.

[Bekker60], M. G. Bekker, *Off-The-Road Locomotion*, The University of Michigan Press, Ann Arbor, MI, 1960.

[Bekker69], M. G. Bekker, *Introduction to Terrain-Vehicle Systems*, The University of Michigan Press, Ann Arbor, MI, 1969.

[Bell85], T. E. Bell, "Robots in the Home: Promises, Promises," *IEEE Spectrum*, Vol. 22, No. 5, May 1985, pp. 51-55.

[Beyer82], W. H. Beyer (editor), *CRC Standard Mathematical Tables*, 26th Edition, CRC Press, Boca Raton, FL, 1982, p. 358.

[Blumrich74], J. F. Blumrich, "Omnidirectional Wheel," U.S. Patent No. 3,789,947, 1974.

[Bonmartini56], G. Bonmartini, "Rolling Device for Vehicles of Every Kind," U.S. Patent No. 2,751,259, 1956.

[Bradbury80], H. M. Bradbury, "Omni-Directional Transport Device," U.S. Patent No. 4,223,753, 1980.

[Brady82], M. Brady, et al. (editors), *Robot Motion, Planning and Control*, The MIT Press, Cambridge, MA, 1982.

[Brogan85], W.L. Brogan, *Modern Control Theory* (Second Edition), Prentice-Hall, Englewoods Cliffs, NJ, 1985.

[Brooks 86], R. A. Brooks, "A Robust Layered Control System for a Mobile Robot," *IEEE Journal of Robotics and Automation*, Vol. RA-2, No. 1, March 1986, pp. 14-23.

[Brown77], E. B. Brown, et al., "Gimbaled Conveyor Balls," U.S. Patent No. 4,018,322, 1977.

[Carlisle83], B. Carlisle, "Omni-Directional Mobile Robot," *Developments in Robotics*, IFS Publishing Ltd., Kempston, Bedfordshire, England, 1983.

[Cohn58], P. M. Cohn, "Linear Equations," Library of Mathematics, Routledge and Kegan Paul, London, 1958.

[Coles75], L. S. Coles, et al., "Decision Analysis for an Experimental Robot with Unreliable Sensors," *Proceedings of the Fourth International Joint Conference on Artificial Intelligence*, Tbilisi, Georgia, USSR, 1975, pp. 749-757.

[Croix85], P. Croix, *Robots*, Crown Publishers, One Park Avenue, New York, 10016, 1985.

[Crowley85], J. L. Crowley, "Navigation for an Intelligent Mobile Robot," *IEEE Journal of Robotics and Automation*, Vol. RA-1, No. 1, March 1985, pp. 31-41.

[Daniel84], D. J. Daniel, "Analysis, Design, and Implementation of Microprocessor Control for a Mobile Platform," Master's Project Report, Department of Electrical and Computer Engineering, Carnegie-Mellon University, Pittsburgh, PA, August 1984.

[Denavit55], J. Denavit and R. S. Hartenberg, "A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices," *Journal of Applied Mechanics*, Vol. 77, No. 2, June 1955, pp. 215-221.

[Denavit65], J. Denavit, R. S. Hartenberg, R. Razi, and J. J. Uicker, Jr., "Velocity, Acceleration, and Static-Force Analysis of Spatial Linkages," *Journal of Applied Mechanics*, Vol. 87, No. 4, December 1965, pp. 903-910.

[Derry82], J. F. Derry, "Roving Robots," *Robotics Age*, Vol. 4, No. 5, September/October 1982, pp. 18-23.

[Draganoiu82], G. Draganoiu, "Computer Method for Setting Dynamical Model of an Industrial Robot with Closed Kinematic Chains," *Proceedings of the Twelfth International Symposium on Industrial Robots*, Paris, France, June 1982, pp. 371-379.

[Dubowsky79], S. Dubowsky and D. T. DesForges, "The Application of Model-Reference Adaptive Control to Robotic Manipulators," *Journal of Dynamic Systems, Measurement, and Control*, Vol. 101, No. 3, September 1979, pp.193-200.

[Elfes83], A. Elfes and S. N. Talukdar, "A Distributed Control System for a Mobile Robot," *Proceedings of the First National Congress on Industrial Automation*, Sao Paulo, Brasil, July 1983, pp. 360-366, and CMU Design Research Center Report No. DRC-18-65-83, December 1983.

[Elfes86], A. Elfes, "A Sonar-Based Mapping and Navigation System," *Proceedings of the 1986 IEEE Conference on Robotics and Automation*, San Francisco, CA, April 1986, pp. 1151-1156.

[Everett85], H. R. Everett, "A Second-Generation Autonomous Sentry Robot," *Robotics Age*, Vol. 7, No. 4, April 1985, pp. 29-32.

[Fischetti85], M. A. Fischetti, "Robots Do the Dirty Work," *IEEE Spectrum*, Vol. 22, No. 4, April 1985, pp. 65-72.

[Giralt79], G. Giralt, R. Sobek, and R. Chatila, "A Multi-Level Planning and Navigation System for a Mobile Robot; A First Approach to Hilare," Proceedings of the IJCAI, Tokyo, Japan, August, 1979, pp. 335-338.

[Grabowiecki19], J. Grabowiecki, "Vehicle-Wheel," U.S. Patent No. 1,305,535, 1919.

[Graettinger88], T. J. Graettinger and B. H. Krogh, "Evaluation and Time-Scaling of Trajectories for Wheeled Mobile Robots," The 1988 American Control Conference, Atlanta, GA, June 1988, pp. 511-516.

[Helmers83a], C. Helmers, "Ein Heldenleben, (Or, A Hero's Life, With Apologies to R. Strauss)," *Robotics Age*, Vol. 5, No. 2, March/April 1983, pp. 7-16.

[Helmers83b], C. Helmers, "The Wandering Roboteers," *Robotics Age*, Vol. 5, No. 3, July/August 1983, pp. 6-10.

[Helmers85], C. Helmers, "Photo Essay: A First Glimpse at Gemini," *Robotics Age*, Vol. 7, No. 2, February 1985, pp. 12-13.

[Hemami79], H. Hemami and B. F. Wyman, "Modeling and Control of Constrained Dynamic Systems with Application to Biped Locomotion in the Frontal Plane," *IEEE Transactions on Automatic Control*, Vol. AC-24, No. 4, August 1979, pp. 526-535.

[Holland83], J. M. Holland, *Basic Robotics Concepts*, Howard W. Sams & Co., Indianapolis, IN, 1983, pp. 107-170.

[Holland85], J. M. Holland, "Rethinking Robot Mobility," *Robotics Age*, Vol. 7, No. 1, January 1985, pp. 26-30.

[Hollerbach80], J. M. Hollerbach, "A Recursive Lagrangian Formulation of Manipulator Dynamics and a Comparative Study of Dynamics Formulation Complexity," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-10, No. 11, November 1980, pp. 730-736.

[Hollis77], R. Hollis, "Newt: A Mobile, Cognitive Robot," *Byte*, Vol. 2, No. 6, June 1977, pp. 30-45.

[Hongo85], T. Hongo, et al., "An Automatic Guidance System of a Self-Controlled Vehicle: The Command System and the Control Algorithm," IEEE Proceedings of the IECON, San Francisco, CA, November 1985, pp.18-22.

[Ichikawa83], Y. Ichikawa, N. Ozaki and K. Sadakane, "A Hybrid Locomotion Vehicle for Nuclear Power Plants," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-13, No. 6, November/December 1983, pp. 1089-1093.

[Iijima81], J. Iijima, et al., "Elementary Functions of a Self-Contained Robot 'Yamabiko 3.1'," Proceedings of the 11th Isir, Tokyo, Japan, 1981, pp. 211-218.

[Ilon75], B. E. Ilon, "Wheels for a Course Stable Selfpropelling Vehicle Movable in any Desired Direction on the Ground or Some Other Base," U.S. Patent No. 3,876,255, 1975.

[Iwamoto83], T. Iwamoto, H. Yamamoto, and K. Honma, "Transformable Crawler Mechanism with Adaptability to Terrain Variations," International Conference on Advanced Robotics, Tokyo, Japan, September, 1983, pp. 285-292.

[Johnson84], R. Johnson, "Part of the Beginning," *Robotics Age*, Vol. 6, No. 8, August 1984, pp. 35-37.

[Jones48], C. Y. Jones, "Global Wheel", U.S. Patent No. 2,448,222, 1948.

[Kanayama81], Y. Kanayama, et al., "A Mobile Robot with Sonic Sensors and its Understanding of a Simple World," Institute of Information Sciences and Electronics, University of Tsukuba, Japan, Techical Report No. ISE-TR-81-22, February 1981.

[Kane61], T. R. Kane, "Dynamics of Nonholonomic Systems," *Journal of Applied Mechanics*, Vol. 83, No. 4, December 1961, pp. 574-578.

[Kane85], T. R. Kane and D.A. Levinson, *Dynamics: Theory and Applications*, McGraw-Hill, New York, 1985.

[Khosla85], P. K. Khosla and T. Kanade, "Parameter Identification of Robot Dynamics," *Proceedings of the 24th Conference on Decision and Control*, Ft. Lauderdale, FL, December 1985, pp. 1754-1760.

[Khosla87], P.K. Khosla and T. Kanade, "Real-Time Implementation and Evaluation of Computed-Torque Scheme," Robotics Institute Technical Report No. CMU-RI-TR-87-6, Carnegie Mellon University, Pittsburgh, PA, March 1987.

[Kragelskii65], I. V. Kragelskii, *Friction and Wear*, Butterworths, Washington, DC, 1965.

[Krogh84], "A Generalized Potential Field Approach to Obstacle Avoidance Control," *Proceedings of the Robotics International Robotics Research Conference*, Bethlehem, PA, August 1984.

[Krogh86], B. H. Krogh and C. E. Thorpe, "Integrated Path Planning and Dynamic Steering Control for Autonomous Vehicles," *Proceedings of the 1986 IEEE International Conference on Robotics and Automation*, San Francisco, CA, April 1986, pp. 1657-1663.

[Labmate87], Transitions Research Corporation, "Labmate," (Preliminary Specifications), Transitions Research Corporation, 15 Durant Avenue, Bethel, CT 06801.

[Landau76], L. D. Landau and E. M. Lifshitz, *Mechanics*, Third Edition, Pergamon Press, New York, 1976.

[Lewis73], R. A. Lewis and A. K. Bejczy, "Planning Considerations for a Roving Robot with Arm," Proceedings of the Third IJCAI, Stanford CA, August 1973, pp. 308-316.

[Lippman87], R. P. Lippman, "An Introduction to Computing with Neural Nets," *IEEE ASSP Magazine*, Vol. 4, No. 2, April 1987, pp. 4-22.

[Luh80a], J. Y. S. Luh, M. W. Walker, and R. P. C. Paul, "Resolved-Acceleration Control of Mechanical Manipulators," *IEEE Transactions on Automatic Control*, Vol. AC-25, No. 3, June 1980, pp. 468-474.

[Luh80b], J. Y. S. Luh, M. W. Walker, and R. P. C. Paul, "On-Line Computational Scheme for Mechanical Manipulators," *Journal of Dynamic Systems, Measurement, and Control*, Vol. 102, No. 2, June 1980, pp. 69-76.

[Luh83], J. Y. S. Luh, "An Anatomy of Industrial Robots and Their Controls", *IEEE Transactions on Automatic Control*, Vol. AC-28, No. 2, February 1983, pp.133-153.

[Luh85], J. Y. S. Luh and Y. Zheng, "Computation of Input Generalized Forces for Robots with Closed Kinematic Chain Mechanism," *IEEE Journal of Robotics and Automation*, Vol. RA-1, No. 2, June 1985, pp. 95-103.

[Marrs85], T. Marrs, *The Personal Robot Book*, Tab Books Inc., Blue Ridge Summit, PA 17214, 1985.

[Matthies86], L. H. Matthies and S. A. Shafer, "Error Modeling in Stereo Navigation," The Department of Computer Science, Technical Report No. CMU-CS-86-140, Carnegie Mellon University, Pittsburgh, PA, 1986.

[McGraw82], *McGraw-Hill Encyclopedia of Science & Technology*, Fifth Edition, McGraw-Hill, New York, Vol. 1, 1982, p. 56.

[McInnis86], B.C. McInnis and C.F. Liu, "Kinematics and Dynamics in Robotics: A Tutorial Based Upon Classical Concepts of Vectorial Mechanics," IEEE Journal of Robotics and Automation, Vol. RA-2, No. 4, December 1986, pp. 181-187.

[Moravec80], H. P. Moravec, "Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover", PhD Thesis, Department of Computer Science, Stanford University, Stanford, CA, 1980.

[Moravec82], H. P. Moravec, Personal Communication, Carnegie-Mellon University, Pittsburgh, PA, January, 1982.

[Moravec83], H. P. Moravec, "The Stanford Cart and the CMU Rover," *Proceedings of the IEEE*, Vol. 71, No. 7, July 1983, pp. 872-884.

[Moravec84], H. P. Moravec, "Locomotion, Vision and Intelligence," *First International Symposium of Robotics Research*, Bretton Woods, NH, August 1983, and *Robotics Research - The First International Symposium*, Michael Brady and Richard Paul (editors), The MIT Press, Cambridge, MA, 1984, pp. 215-224.

[Moravec86], H. P. Moravec (editor), "Autonomous Mobile Robots Annual Report - 1985," Robotics Institute Technical Report No. CMU-RI-MRL 86-1, Carnegie-Mellon University, Pittsburgh, PA, January 1986.

[Moravec87], H. P. Moravec, "Sensor Fusion in Certainty Grids for Mobile Robots," The Robotics Institute 1987 Annual Research Review, Carnegie Mellon University, Pittsburgh, PA, 1987, pp. 33-48.

[Moravec88a], H. P. Moravec, *Mind Children*, Harvard University Press, Cambridge, MA, 1988.

[Moravec88b], H. P. Moravec, "Sensor Fusion in Certainty Grids for Mobile Robots," AI Magazine, Vol. 9, No. 2, Summer 1988, pp. 61-74.

[Muir84a], P. F. Muir, "Digital Servo Controller Design For Brushless DC Motors," Master's Project Report, Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA, April 1984.

[Muir84b], P. F. Muir and C. P. Neuman, "Pulse-Width Modulation Control of Brushless DC Motors for Robotics Applications," *Proceedings of the 27th Midwest Symposium on Circuits and Systems*, Morgantown, WV, June 1984, pp. 489-491.

[Muir85], P. F. Muir and C. P. Neuman, "Pulsewidth Modulation Control of Brushless DC Motors for Robotic Applications," *IEEE Transactions on Industrial Electronics*, Vol. IE-32, No. 3, August 1985, pp. 222-229.

[Muir86], P. F. Muir and C. P. Neuman, "Kinematic Modeling of Wheeled Mobile Robots," Robotics Institute Technical Report No. CMU-RI-TR-86-12, Carnegie Mellon University, Pittsburgh, PA, June 1986.

[Muir87a], P. F. Muir and C. P. Neuman, "Kinematic Modeling for Feedback Control of an Omnidirectional Wheeled Mobile Robot," *Proceedings of the 1987 IEEE International Conference on Robotics and Automation*, Raleigh, NC, March 1987, pp. 1772-1778.

[Muir87b], P. F. Muir and C. P. Neuman, "Kinematic Modeling of Wheeled Mobile Robots," *Journal of Robotic Systems*, Vol. 4, No. 2, April 1987, pp. 281-340.

[Muir88a], P. F. Muir and C. P. Neuman, "Dynamic Modeling of Multibody Robotic Mechanisms: Incorporating Closed-Chains, Friction, Higher-Pair Joints, and Unactuated and Unsensed Joints," *Proceedings of the 1988 IEEE International Conference on Robotics and Automation*, Philidelphia, PA, April 1988, pp. 1546-1553.

[Murray86], J. J. Murray, *Computational Robot Dynamics*, PhD Dissertation, Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA, September 1986.

[Neuman85], C. P. Neuman and J. J. Murray, "Computational Robot Dynamics: Foundations and Applications," *Journal of Robotic Systems*, Vol. 2, No. 4, Winter 1985, pp. 425-452.

[Neuman87a], C. P. Neuman and V. D. Tourassis, "Robust Discrete Nonlinear Feedback Control for Robotic Manipulators," *Journal of Robotic Systems*, Vol. 4, No. 1, February 1987, pp. 115-143.

[Neuman87b], C. P. Neuman and P. F. Muir, "From Linear Algebraic Equations to Wheeled Mobile Robot Modeling," in K. S. Narendra (editor), *Proceedings of the Fifth Yale Workshop on Applications of Adaptive Systems Theory*, Yale University, New Haven, CT, May 1987.

[Neuman87c], C. P. Neuman and J. J. Murray, "Customized Computational Robot Dynamics," *Journal of Robotic Systems*, Vol. 4, No. 4, August 1987, pp. 503-526.

[Neuman87d], C.P. Neuman and J. J. Murray, "Symbolically Efficient Formulations for Computational Robot Dynamics," *Journal of Robotic Systems*, Vol. 4, No. 6, December 1987, pp. 743-769.

[Nilsson84], N. J. Nilsson, "Shakey the Robot," Technical Note 323, Artificial Intelligence Center, Computer Science and Technology Division, SRI International, Menlo Park, CA, April 1984.

[Orin81], D. E. Orin and S. Y. Oh, "Control of Force Distribution in Robotic Mechanisms Containing Closed Kinematic Chains," *Journal of Dynamic Systems, Measurement, and Control*, Vol. 103, No. 2, June 1981, pp. 134-141.

[Paul81a], R. P. Paul, *Robot Manipulators: Mathematics, Programming and Control*, The MIT Press, Cambridge, MA, 1981.

[Paul81b], R. Paul, J. Luh, et al., "Advanced Industrial Robot Control Systems," Technical Report No. TR-EE79-35, School of Electrical Engineering, Purdue University, West Lafayette, IN, 47907, July 1979.

[Pfeifer84], M. S. Pfeifer and C. P. Neuman, "An Adaptable Simulator for Robot Arm Dynamics," *Computers in Mechanical Engineering*, Vol. 3, No. 3, November 1984, pp. 57-64.

[Pieper68], D. L. Pieper, "The Kinematics of Manipulators under Computer Control," Stanford Artificial Intelligence Report, Memo No. AI-72, Computer Science Department, Stanford University, Stanford, CA, October 1968.

[Podnar84], G. Podnar, K. Dowling and M. Blackwell, "A Functional Vehicle for Autonomous Mobile Robot Research", Robotics Institute Technical Report No. CMU-RI-TR-84-28, Carnegie-Mellon University, Pittsburgh, PA, April 1984.

[Raibert83], M. Raibert, et al., "Dynamically Stable Legged Locomotion," Robotics Institute Technical Report No. CMU-RI-TR-83-20, Carnegie-Mellon University, Pittsburgh, PA, December 1983.

[Rangan82], K.V. Rangan, "Position and Velocity Measurement by Optical Shaft Encoders," Robotics Institute Technical Report No. CMU-RI-TR-82-8, Carnegie Mellon University, Pittsburgh, PA, June 1982.

[Rao88], S. K. Rao, *The Design of Processor Arrays*, Prentice-Hall, Englewood Cliffs, NJ, 1988.

[Rogers84], M. Rogers, "Birth of the Killer Robots," *Newsweek*, June 25, 1984, p. 51.

[Schachter83], B. J. Schachter, et al., "Robot Vehicles: A Survey and Proposed Test-Bed Facility," DARPA, Proceedings: Image Understanding Workshop, 1983, pp. 163-174.

[Schmidt71], R. A. Schmidt, Jr., "A Study of the Real-Time Control of a Computer Driven Vehicle," Stanford Artificial Intelligence Project Memo AIM-149, Computer Science Department Report No. CS-231, Stanford University, Stanford, CA, August 1971.

[Sears77], F. W. Sears, M. W. Zemansky and H. D. Young, *University Physics*, Addison-Wesley, Reading, MA, 1977.

[Sheth71], P. N. Sheth and J. J. Uicker, Jr., "A Generalized Symbolic Notation for Mechanisms", *Journal of Engineering for Industry*, Series B, Vol. 93, No. 70-Mech-19, February 1971, pp. 102-112.

[Shigley69], J. E. Shigley, *Kinematic Analysis of Mechanisms*, McGraw-Hill, New York, 1969.

[Shigley80], J. E. Shigley and J. J. Uicker, Jr., *Theory of Machines and Mechanisms*, McGraw-Hill, New York, 1980.

[Smith73], M. H. Smith and L.S. Coles, "Design of a Low Cost, General Purpose Robot," Proceedings of the Third IJCAI, Stanford, CA, August, 1973, pp. 324-335.

[Strang86], G. Strang, *Introduction to Applied Mathematics*, Wellsley-Cambridge Press, Wellesley, MA 02181.

[Symbolics85], *MACSYMA Reference Manual, Version 11*, Symbolics, Inc., Cambridge, MA, November 1985.

[Thorpe83], C. E. Thorpe, "An Analysis of Interest Operators for FIDO," Robotics Institute Technical Report No. CMU-RI-TR-83-19, Carnegie Mellon University, Pittsburgh, PA, December 1983.

[Thorpe84], C. E. Thorpe, *FIDO: Vision and Navigation for a Mobile Robot*, PhD Dissertation, Computer Science Department, Carnegie Mellon University, Pittsburgh, PA, 1984.

[Thorpe85], C. E. Thorpe, L. Matthies and H. Moravec, "Experiments and Thoughts on Visual Navigation," *Proceedings of the 1985 IEEE International Conference on Robotics and Automation*, St. Louis, MO, March 1985, pp. 830-835.

[Tourassis85], V. D. Tourassis and C. P. Neuman, "Robust Nonlinear Feedback Control for Robotic Manipulators," *IEE Proceedings - D: Control Theory and Applications*, Special Issue on Robotics, Vol. 132, No. 4, July 1985, pp. 134-143.

[Wallace84], R. S. Wallace, "Three Findpath Problems," *Proceedings of the AAAI-84*, Austin, TX, August 1984, pp. 326-329.

[Wallace85], R. Wallace, et al., "First Results in Robot Road-Following," Proceedings of the IJCAI, Los Angeles CA, August, 1985, pp. 1089-1095.

[Whitaker62], L. E. Whitaker, "Stair Climbing Device," U.S. Patent No. 3,058,754, 1962.

[Whitney69], D. E. Whitney, "Resolved Motion Rate Control of Manipulators and Human Protheses," *IEEE Transactions on Man-Machine Systems*, Vol. MMS-10, No. 2, June 1969, pp.47-53.

[Wier81], J. P Wier and R. A. Garrett, "Ambulator Drive Mechanism," U.S. Patent No. 4,258,815, 1981.

[Wilson85], E. Wilson, "Denning Mobile Robotics: Robots Guard the Pen," *High Technology*, Vol. 5, No. 6, June 1985, pp. 15-16.

[Wittenburg77], J. Wittenburg, *Dynamics of Systems of Rigid Bodies*, B.G. Teuber, Stuttgart, 1977.

[Wilson85], E. Wilson, "Denning Mobile Robotics: Robots Guard the Pen," *High Technology*, Vol. 5, No. 6, June 1985, pp. 15-16.

[Wittenburg77], J. Wittenburg, *Dynamics of Systems of Rigid Bodies*, B.G. Teuber, Stuttgart, 1977.