KINEMATICS AND OPEN-LOOP CONTROL OF AN ILONATOR-BASED MOBILE PLATFORM

David J. Daniel and **Bruce H. Krogh**
Department of Electrical and Computer Engineering
Carnegie-Mellon University
Pittsburgh, Pennsylvania 15213

Mark B. Friedman
Robotics Institute
Carnegie-Mellon University
Pittsburgh, Pennsylvania 15213

ABSTRACT

This paper describes the analysis and implementation of a motor drive control system for an intelligent wheelchair being developed at the Robotics Institute at Carnegie-Mellon University. The experimental mobile platform is mounted on four so-called *Ilonators* which permit motion in any direction without conventional steering of the wheel axes. The kinematics of the platform were determined analytically and a simulation program was developed to generate reference wheel velocity trajectories corresponding to several basic platform trajectories. These reference trajectories were used to drive the platform under open-loop control. The paper includes a description of the motor-drive hardware implementation, and experimental results. The results of this work provide the foundation for the implementation of feedback control of the wheels and higher-level supervisory control.

1. Introduction

This paper concerns the analysis, design and implementation of a motor drive control system for an experimental mobile platform. The platform wheels are so-called *Ilonators* which permit motion in any direction without conventional steering of the wheel axes. The development of the motor drive control system required analysis of the wheel and platform kinematics. The paper describes the kinematic model, simulation of the wheel and platform trajectories, the system implementation and results of open-loop control experiments. This work was undertaken as part of an ongoing project to develop an advanced wheelchair control system. The issues addressed in the design of the motor drive system apply to the general problem of autonomous vehicle control.

Control of an autonomous vehicle is normally accomplished with a hierchical decomposition of the control functions. The high-level controller uses sensory information to create and update a world model for navigation, that is, for path planning and obstacle avoidance. Low-level controllers interface directly with the drive motors and sensors. The basic function of the motor drive control systems is to implement the reference trajectories generated by the high-level controller. Vehicles currently utilizing such techniques are described in [1], [2].

The motor drive control system for the Ilonator platform is described in this paper. The function of this low-level controller is to drive the platform to a desired final position and orientation. Two aspects of the motor drive control problem are addressed in the sequel. First, the motion of the wheels and platform had to be understood and mathematically modeled before the control design could be attempted. We present the analytical model of the platform kinematics which relates platform velocity trajectories to the individual Ilonator wheel velocities. Secondly, we describe the hardware implementation and results of preliminary open-loop control experiments which provide the basis for future feedback control design. Section 2 describes the Ilonator wheels and presents the kinematic equations for the experimental mobile platform. The simulation program developed to generate reference trajectories is described in Section 3 and Section 4 details the open-loop control experiments. The results are summarized in Section 5 and directions for future work are discussed.

2. Equations of Motion

In this section we consider the kinematics of the Ilonator wheels for both a single wheel and four wheels mounted together as a system. The Ilonator platform consists of a rectangular base having an Ilonator wheel mounted on each corner of the base, each being driven by a geared, permanent magnet DC motor. The axes of the motors are fixed and, due to the construction of the wheels, motion in any direction is possible without conventional 'steering' of one set of wheels. It is possible to accomplish arbitrary translation in any direction, rotation, or a combination of rotation and translation by driving the wheels at proper velocities. Encoders were mounted on the motor shafts to provide a means of gathering velocity profile data. The system controller consists of a Motorola 68000 Educational Board which is optically isolated from the motor drive circuitry. A system block diagram is shown in Figure 2-1.

The Ilonator wheels are of a unique design and have many possible applications [3], [4]. The center of each Ilonator rotates about the fixed axis. Mounted about the periphery of the wheel are twelve freely spinning rollers whose axes are mounted at 45° angles with respect to the side faces of the wheel. The center of the wheel is constructed of aluminum while the rollers are made of an elastic material to prevent slippage. The Ilonator acts as a conventional wheel when being driven forward and backward. Because the rollers can be mounted with their axes aligned in two different ways, it is possible to have "left-hand" and "right-hand" wheels, where left- or right- handedness indicates the direction in which the wheel can freely move forward at 45° with only roller rotation.

A single left-hand wheel with front facing to the right is shown in Figure 2-2. This figure is a top view of the wheel with the roller drawn being in contact with the ground. Motion of this wheel along a constant angle η and with a specified velocity uniquely determines the roller and wheel angular velocities, ω_r and ω_w (assuming no slippage). For movement forward ($\eta = 0°$) only the wheel turns, that is, the rollers do not spin. If $\eta = 45°$, the wheel does not turn while the roller in contact with the ground spins freely. To move at any other angle both the rollers and wheel spin.

By resolving the wheel velocity into two components, one directed along the line $\eta = 0°$ and one along $\eta = 45°$, moving the wheel in direction η can be viewed as a combination of motion along these components. Analyzing the motion in this way, any desired wheel velocity can be expressed as a linear combination of the unit vectors $\mathbf{u}_1$ and $\mathbf{u}_2$ shown in Figure 2-3. The relationships of ω_w and ω_r to η are shown in Figure 2-4 for both left and right-hand wheels. For left-hand wheels, the relation between local frame velocities v_x, v_y and wheel velocities are

$$\omega_w = K_1(v_x - v_y) \tag{1}$$

and

$$\omega_r = K_2 v_y \tag{2}$$

giving the ω_r and ω_w vs η curves shown in Figure 2-4(a). For right-hand wheels the relations are similar, given by,

$$\omega_w = K_1(v_x + v_y) \tag{3}$$

and

$$\omega_r = K_2 v_y \ , \tag{4}$$

as shown in Figure 2-3(b).

The kinematics of the Ilonator Platform are defined by the mapping of translational and rotational velocities of the platform, into the four wheel (motor) velocities. This mapping is required to transform a desired velocity trajectory for the platform into reference velocity trajectories for the four wheel controllers. The inverse of this mapping is also useful to compute platform trajectories from the wheel position encoders.

The platform is shown in Figure 2-4 with the axes, angles, and wheels labeled using the following notation:

ω_i	- angular velocity of the i^{th} wheel, defined as positive when the top of the wheel turns toward the front of the platform.
x-y	- inertial coordinate system (global frame).
v_x, v_y	- components of platform velocity in global frame.
x'-y'	- coordinate system fixed with respect to the platform (local frame).
$v_{x'}, v_{y'}$	- components of platform veloctity in local frame.
Θ	- orientation of the platform in degrees (0° is defined as front facing the positive x axis, with Θ increasing counter-clockwise.)
β	- direction of travel in degrees (same convention as Θ, with respect to the global frame).
v	- platform velocity vector.
D	- distance from the center of mass (the centroid) of the platform to the wheels' points of contact with the ground.
Ω	- rotational velocity of the platform (defined positive if turning counter-clockwise).
ϕ	- an angle describing the rectangular platform, formed by a line drawn form the center of the platform to one corner and the side of the platform (in degrees).
r_w	- wheel radius.

From (1) and (3) the mapping, $\mathbf{T}$, of the platform translational and rotational velocities in the local frame, $v_{x'}, v_{y'}, \Omega$, into the four wheel velicities, $\omega_1, \omega_2, \omega_3, \omega_4$, is given by the matrix [5],

$$\mathbf{T} = r_w^{-1} \begin{bmatrix} 1 & -1 & -\sqrt{2}D\cos(45-\phi) \\ 1 & 1 & \sqrt{2}D\cos(45-\phi) \\ 1 & -1 & \sqrt{2}D\cos(45-\phi) \\ 1 & 1 & -\sqrt{2}D\cos(45-\phi) \end{bmatrix} \tag{5}$$

To define an inverse mapping we note that the four wheel velocities can not be assigned arbitrarily since the platform has only three degrees of freedom. Expressing ω_3 as a linear combination of the other wheel velocities

$$\omega_3 = \omega_1 + \omega_2 - \omega_4. \tag{6}$$

the mapping $\mathbf{R}$ of the remaining wheel velocities, $\omega_1, \omega_2, \omega_4$, into the platform translational and rotational velocities, $v_{x'}, v_{y'}, \Omega$ in the local frame is given by the matrix

$$\mathbf{R} = r_w \begin{bmatrix} 0.5 & 0.5 & 0.0 \\ -0.5 & 0.0 & 0.5 \\ 0.0 & 1/\sqrt{8}D\cos(45-\phi) & -1/\sqrt{8}D\cos(45-\phi) \end{bmatrix} \tag{7}$$

The mappings (5) and (7) are linear with respect to the local frame of the platform. However, for constant wheel velocities the platform trajectory is circular in the global frame with radius

$$r = \frac{|v|}{|\Omega|} \tag{8}$$

and center

$$x_c = x_o + r\cos(\beta_o + 90° \ \mathrm{sgn}(\Omega)) \tag{9}$$

$$y_c = y_o + r\sin(\beta_o + 90° \ \mathrm{sgn}(\Omega)) \tag{10}$$

where x_o, y_o are the initial position and β_o is the initial orientation in the global frame. Thus the *natural trajectories* of the Ilonator Platform are circles.

There are two interesting degenerate cases of this circular motion: infinite radius (straight line translation) and zero radius (rotation of the platform with no translation). In the first case, the velocities of the diagonal wheel pairs are equal. In the second case the wheel velocities are all equal in magnitude and the platform rotates about its center. The following section describes how these circular trajectories were used as basic reference trajectories for open-loop control of the platform.

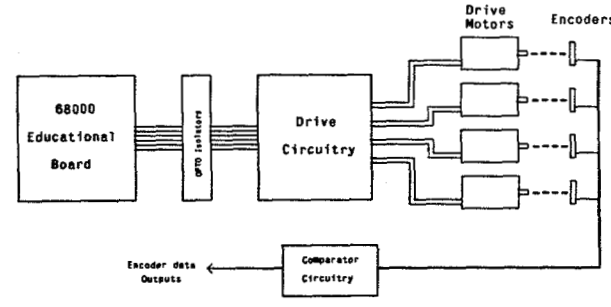

Figure 2-1: Block diagram of motor drive control system.

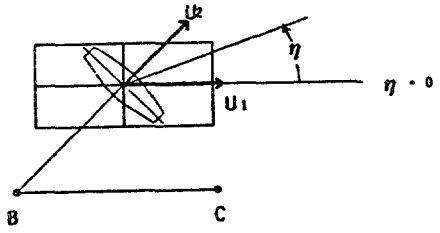

Figure 2-2: Diagram of left-hand Ilonator wheel.

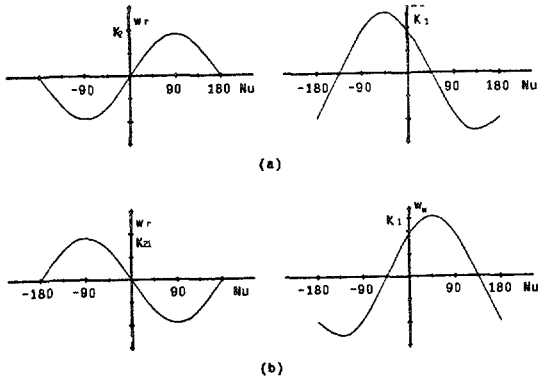

Figure 2-3: ω_w and ω_r vs η for (a) left-hand and (b) right-hand wheels.

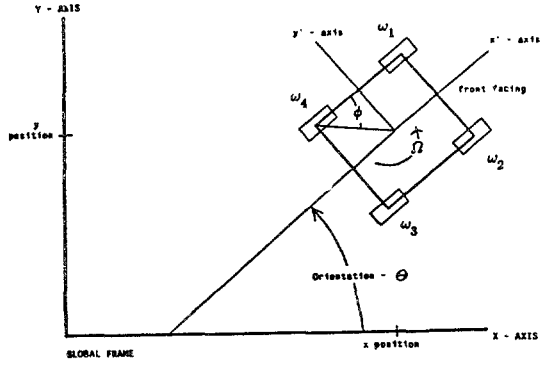

Figure 2-4: Definitions of local and global coordinate frames.

3. Reference Trajectories for Open-Loop Control

The development of the open-loop control scheme was based on the following assumptions. The operating speeds of the platform were limited to a region where dynamic considerations, such as Coriolis effects, could be neglected. Also, the velocity increments and frequency of the pulse width modulation were such that the finite time constant of the motor acceleration response was neglible, that is, we assumed perfect tracking of the velocity reference trajectory. It was assumed that outside of the dead zone the speed of the motors varied linearly with the duty cycle of the applied armature voltage [6].

The transformation (5) was used to generate the desired trajectories for the platform. Since the transformation is linear, scaled wheel velocities yield linearly scaled platform speeds with identical translation directions. Using this fact, the platform can be accelerated by increasing each of the wheel speeds proportionally. To drive the platform to a final position and orientation several possible trajectories were considered (Table 3-1).

Initially, trajectories for pure translation and pure rotation were used as building blocks for two simple trajectory types. In the first of these, Type A, the platform is rotated to its final orientation and is then translated without rotation to its desired final position. For the wheelchair application, human factors suggest that this type of trajectory may not be desirable because the patient would travel in a direction he or she cannot see. In trajectory Type B, the platform rotates to face the direction of travel, translates straight forward (with respect to the local frame) and then rotates to the desired final orientation. This corresponds to the travel of a conventional wheel chair operated by a human passenger and seems most natural for someone seated on the platform.

Other possible trajectories could be composed of circular arcs since the natural trajectories of the platform are circular. The third trajectory, Type C, moves the platform from an initial point to a specified final point by traveling in a circular arc, as shown in Figure 3-1. Given the desired radius of curvature it is possible to compute the necessary wheel velocities [5]. Translating along a circular arc may be useful for obstacle avoidance by specifying the center of the circle to be the corner of an obstacle. In this way the platform would be able to avoid the obstacle without executing a trajectory to a sub-goal near the edge of the obstacle. It may be desirable to combine this circular motion with pure translation to give a smooth trajectory around obstacles or corners in halls.

In the final trajectory, Type D, the required translation and rotation are executed simultaneously. In order to accomplish motion which is linear in translation while rotation about the center is being accomplished, time varying (sinusoidal) velocities must be applied to the wheels. This type of motion appears to be time-optimal, but it would be difficult to implement in real time. As with Type A trajectories, during Type D trajectories a person on the platform would not be facing the direction of travel.

To facilitate the implementation and testing of control algorithms for the platform, a simulation program was written on a Hewlett-Packard 9836C personal technical computer. This program simulates the kinematics of the wheels and platform and displays platform trajectories based on options selected interactively. Using the simulation program it is possible to implement and test a control algorithm before testing operation of the hardware. The program first prompts the user for an initial position and orientation of the platform. A menu then allows selection of the following options:

1. Constant Wheel Speeds: This mode allows the user to specify constant wheel velocities for three of the wheels (the program generates the fourth), the sampling interval, and the number of intervals to be executed by the platform. The motion of the platform is displayed graphically and, if specified, the trajectory of the platform is displayed upon completion of the simulation.

2. Local Motion Parameters: In this mode the user specifies the desired x- and y- velocities relative to the local reference frame of the platform and the platform's rotational speed. The sampling interval and number of steps executed are specified. The program translates the speeds given into the necessary wheel velocities which are used to generate the trajectory.

3. Computer Generated Trajectories: The user specifies the desired final state of the platform and the program generates the wheel speeds to bring the platform to rest at that point from a selected initial position and orientation. Parameters which are specified by the user are the maximum wheel speed, sampling interval, and an acceleration limit. A sub-menu for this particular item allows choice of the trajectory to be generated (Type A, B, or C, in Table 3-1).

4. File Input: Given a file containing wheel velocities, this option displays the resulting trajectory of the platform. This option was used to compare the ideal (simulated) trajectory of the platform to the actual trajectory using the wheel velocity data obtained from the encoders.

For the selected option, the simulation generates a linked list containing the complete state of the system at each sample instant. After completion of a simulation, the program allows the user to create a file of reference wheel velocity trajectories compatible with the motor control software and/or a data file for plotting. The reference trajectories are based on the simplifying assumptions discussed above. Since the motor can not attain full speed instantaneously and the change in speed is limited, the speed was slowly incremented to maximum speed. The magnitude of the velocity increments is equal to the acceleration limit so that the platform reaches constant maximum speed as quickly as possible. Since the final state of the platform is known *a priori*, the motors are brought to rest by stepping the motor velocities down at the proper time.

For each of the trajectory types in Option 3 the simulation program creates a data file of wheel velocity reference trajectories for driving the platform under open-loop control, as described in the following section. A limit ω_{max} is placed on the rotational speed of the motors which results in a direction-dependent limit on the platform speed. For pure translation of the platform the maximum velocity (neglecting friction) is given by [5]

$$V_{max}(\eta) = \min \{ \frac{\omega_{max}}{K_1 \sqrt{2} \cos(\eta + 45^0)}, \frac{\omega_{max}}{K_1 \sqrt{2} \cos(\eta - 45^0)} \} \quad (11)$$

where η is the direction of translation with respect to the local frame of the platform. A graph of this function is shown in Figure 3-2. If the maximum platform velocity is to be the same for all directions the maximum wheel velocity must be varied with η, which would impose a computational burden on a real-time control system.

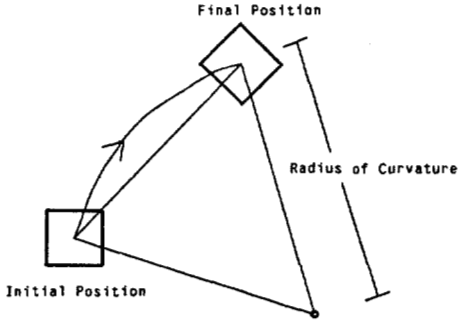

Figure 3-1: Trajectory Type C.

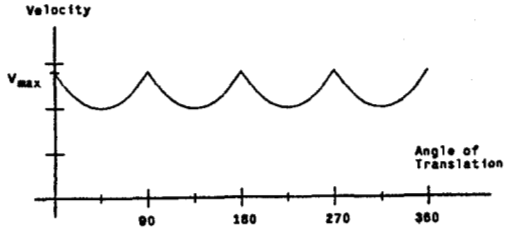

Figure 3-2: Maximum platform velocity vs. Translation angle for constant maximum wheel speeds

Type	Description
A	The platform rotates to its final orientation and then translates to its final orientation.
B	The platform rotates so that it can move forward relative to the local frame during its translation, and when it has reached its final position, it then rotates again to the final orientation.
C	The platform moves in a circular arc of specified radius from its initial position to final position.
D	The platform translates and rotates simultaneously to its final position and orientation.

Table 3-1: Description of Trajectory Types.

4. Hardware Implementation and Experimental Results

Up to this point the description of the platform's motion has neglected any real world considerations (such as friction, slippage, and non-ideal motors). It was necessary to determine experimentally the platform's operating charateristics. The goal of the experiments described in this section was to gain an understanding of the platform as it operates under open-loop control. The results of these experiments are currently being used to formulate a closed-loop control scheme which properly addresses the sources of the open-loop control errors.

Motor speed was controlled by pulse-width modulation (PWM) of power MOSFETs connected in a bridge configuration around a DC permanent-magnet geared motor. The motor drive circuit accepts as inputs a duty cycle line (the speed setpoint input) and a number of select lines allowing four possible modes of operation: (1) drive motor forward, (2) drive motor reverse, (3) coast, and (4) dynamic braking. Because the motors selected are permanent-magnet DC, braking was accomplished by shorting the armature leads together [7]. Encoders were mounted on the wheel shafts and trajectory data were collected by an Apple Computer, and associated counter circuitry, mounted on the platform. To test the operation of the encoders, speed measurements were taken for a computer generated trajectory, and repeatability was confirmed.

Programs were written for a Motorola 68000 to generate four channels of PWM waves for the four drive circuits motors. For the initial testing of the platform the motors were operated at selected constant velocities. In later experiments, the wheel velocity reference trajectories generated by the simulator were read by a 68000 control program to drive the platform along selected trajectories. The programs, written in C and 68000 assembly language, generate the desired PWM for all four motors with no hardware exteranal to the 68000 microprocessor. The program for running the motors at constant speeds allows the user to set the wheel speeds interactively, display motor speeds, raise or lower motor speeds, and reverse motor directions. A second program was used to verify the kinematic transformations and to test open-loop control for different platform trajectories using the simulator generated wheel velocity reference trajectories.

The first set of experiments was designed to characterize the effects of the frictional characteristics of the Ilonators on the platform control. It was found that the platform has a dead zone, that is, a minimum pulse width is required to initiate motion of the platform. Because of the complex frictional characteristics of the

Ilonators, this dead zone is direction dependent. For these experiments, the motors were operated under open-loop control to translate the platform in directions (η) ranging from 0 to 347.5 degrees in steps of 22.5 degrees. For each angle of translation the motors were started at the minimum pulse width required to initiate motion and then the pulse width was stepped to the next highest level.

The results of the translational experiment are shown graphically in Figure 4-1 where the average required pulse width is indicated on the vertical axis. As was expected, the graph is symmetric about 180°. The results indicate that the most power is required to drive the platform directly sideways, which reflects the fact that the bearing friction of the rollers is greatest for these directions (cf. Figure 2-3). Thus, the frictional characteristics of the Ilonators vary with the angle of travel. To properly generate trajectories to drive the platform, we must account for this dead zone to eliminate the positional error. For pure rotation the chair required a minimum PWM wave of 8% duty cycle to each wheel.

The second set of experiments determined open-loop control errors for purely rotational trajectories. The platform was placed at a fixed initial orientation and simulator-generated trajectories were used to rotate the platform. The final angular error was then measured. The results of this experiment for several angles are shown in Figure 4-2 which shows a linearly increase in the error for increasing angles. Because the percent error increases linearly, the positional error increases as the square of the distance (angle) traveled. This indicates that the actual wheel speeds were equal to a constant times the reference speeds. Since the speeds were ramped up and down using PWM and the actual speed was proportional to the reference trajectory, we conclude that the PWM does implement a speed proportional to the duty cycle of the wave. However, the constants in the kinematic transformations should be "tuned" experimentally.

A third set of experiments determined the open-loop control error for forward translation of the platform. Results are given in Table 4-1 for trajectories with and without braking applied at the end of the trajectory. The platform coasts for a significant distance without braking applied upon completion of the trajectory. Even with braking, the platform continues to move forward for approximately eight inches due to slippage of the rollers.

The nonideal characteristics of the drive motors under load also account for some of the error in both rotation and forward translation. Operating with only the front two wheels powered, the platform translated (within experimental error) only one half of the distance with all wheels powered. This performance confirmed that motor loading is a significant factor in the platform operation. Clearly, feedback control is required to obtain satisfactory performance. With closed-loop speed control breaking will still be necessary because of coasting exhibited in these experiments.

A set of experiments was performed to confirm the validity of the kinematic transformations (5),(7). The platform was pushed by hand at a number of angles from its initial position while maintaining the same orientation. During the motion, the encoders were used to collect velocity data on the Apple Computer. After being brought to rest, the final position of the platform was measured. Using the data measured for translation, the angle of translation of each sample was computed and the average value for the each trajectory was obtained. Only three of the measured wheel speeds were used for this computation, but the fourth was also measured. Since it must be a linear combination of the other three, its value was computed from the other measurements and compared to the actual value. From these computations an average deviation from the computed values was obtained. The data from these experiments are summarized in Table 4-2.

Sources of error in the measured wheel speeds are due in part to the difficulty in pushing the platform in a straight line and slippage between the rollers and the floor. Since the speed is measured by counting encoder pulses and averaging them over time, the resolution of the encoders is also a factor in the error. These experiments indicate that, within experimental error, the translation angles computed from the measured trajectories match the actual results.

Another check on the validity of the transformations was to use the simulator to execute the measured velocity profiles to obtain expected final position from the simulated trajectories. The output of the simulation program was compared to the actual final position measurements. Table 4-3 contains the measured final position and the simulation outputs, and the final positional error between the two. The average error for all trials was 3.5 inches. As expected, larger sampling times were, all other things equal, indicative of larger final positional error. Thus, a major portion of the error in this experiment can be attributed to the size of the sampling period.

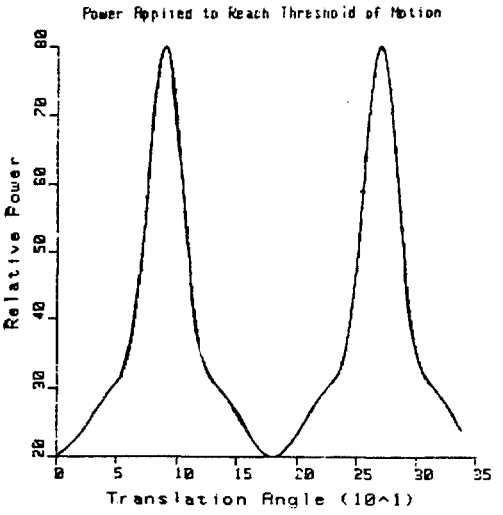

Figure 4-1: Platform dead zone as a function of translational direction.

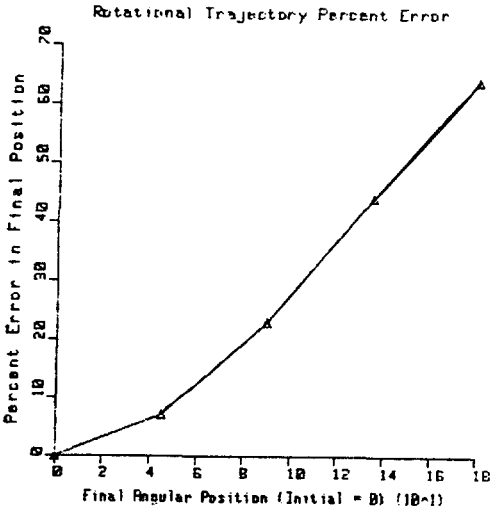

Figure 4-2: Percent final state error for pure rotation.

	Absolute Error	
Desired Distance	**No Breaking**	**Breaking**
12 in.	36 in.	15 in.
18 in.	65 in.	33 in.
24 in.	95 in.	54 in.

Table 4-1: Final position error for forward translation.

	Translation Direction				
	$\eta = 25.4^\circ$	$\eta = 158^\circ$	$\eta = 209^\circ$	$\eta = 298^\circ$	$\eta = 324^\circ$
Angle Error (in degrees)	0.8°	3.6^0	2.9°	3.6°	4.1°
Average ω_3-$\hat{\omega}_3$ (in encoder counts)	1.58	1.70	1.68	1.76	1.99

Table 4-2: Transformation Verification Experiment results.

	Translation Direction				
	$\eta = 25.4^\circ$	$\eta = 158^\circ$	$\eta = 209^\circ$	$\eta = 298^\circ$	$\eta = 324^\circ$
Measured x_f	24.75 in.	-28.25 in.	-31.25 in.	17.75 in.	32.75 in.
Measured y_f	11.75 in.	11.0 in.	-17.5 in.	-33.0 in.	-23.75 in.
Computed x_f	23.3 in.	-26.4 in.	-26.3 in.	16.8 in.	31.6 in.
Computed y_f	11.8 in.	8.2 in.	-16.3 in.	-29.5 in.	-19.9 in.
Error	1.5 in.	3.4 in.	5.1 in.	3.6 in.	4.0 in.

Table 4-3: Results of running the simulation with measured data.

5. Conclusions and Ongoing Work

This paper reports the results of analysis and preliminary controller design for the development of a motor drive system for an Ilonator-based mobile platform. The kinematic model, which has been verified experimentally, characterizes the motion of the platform and provides the basis for a dead-reckoning control system. Motor drive circuitry and pulse-width modulation software have been developed and implemented on an experimental platform. A program was written to simulate the platform kinematics and to generate reference trajectories for open-loop control experiments.

The results and implications of the control experiments are summarized as follows:

- The platform has a dead zone which depends on the direction of translation with respect to the local frame of reference.
- Experimental measurements verified the analytical kinematic model.
- Feedback control of the motors must be implemented to counter the effects of loading, friction and slippage.
- Dynamic breaking is required to obtain final position accuracy.

Current work includes the analytical derivation of a complete dynamic model of the Ilonator platform. A feedback controller for the motor drive system is being developed using the encoders on the wheels to close the loop. The platform trajectory error in the global frame will be computed in real-time from the encoder readings. Work is also underway on the higher level controller which will incorporate sonar sensory information into algorithms for path planning and obstacle avoidance. This higher level controller will provide the velocity reference trajectories to the motor drive control system.

References

1. Crowley, J.L., "Navigation for an Intelligent Mobile Platform", Tech. report 84-18, Carnegie-Mellon University Robotics Institute, 1984, Carnegie-Mellon University Robotics Institute Technical Report.
2. Moravec, H.P., "Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover", Tech. report AIM-340, Stanford University Artificial Intelligence Laboratory, September 1980.
3. Ilon, B.E., "Wheels for a Course Stable Selfpropelling Vehicle Movable in any Desired Direction on the Ground or Some Other Base". Patent number 3,876,255
4. Adams, R., "Omnidirectional Vehicle", *Robotics Age*, Vol. 6, No. 21984, pp. 21-22.
5. Daniel, David J., "Analysis, Design, and Implementation of Microprocessor Control of a Mobile Platform", Master's thesis, Carnegie-Mellon University, September 1984.
6. Muir, P.F., "Digital Servo Controller Design for Brushless DC Motor.", Master's thesis, Carnegie-Mellon University, April 1984.
7. McPherson, G., *An Introduction to Electrical Machines and Transformers*, John Wiley & Sons, New York, 1981.