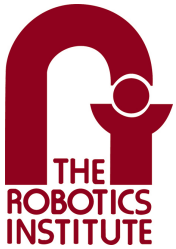
On-line Mobile Robotic Dynamic Modeling using Integrated Perturbative Dynamics

Forrest Rogers-Marcovitz

CMU-RI-TR-10-15

Submitted in partial fulfillment of the
requirements for the degree of
Master of Science in Robotics.



The Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

April 2010

Thesis Committee

Alonzo Kelly, Chair
Thomas Howard, Jet Propulsion Laboratory
Mihail Pivtoraiko

Copyright © 2010 by Forrest Rogers-Marcovitz. All rights reserved.

Abstract

Mobile robotic dynamics modeling is necessary for reliable planning and control of unmanned ground vehicles on rough terrain. Autonomous vehicle research has continuously demonstrated that a platform's precise understanding of its own mobility is a key ingredient of competent machines with high performance. I will investigate the feasibility and mechanism of enabling a platform to better predict its own mobility by learning from its own experience. The autonomy system will calibrate, in real-time, vehicle dynamics models, based on residual differences between the motion originally predicted by the platform and the motion ultimately experienced by the platform.

This thesis develops an integrated perturbative dynamics method for real-time identification of wheel-terrain interaction models for enhanced autonomous vehicle mobility. I develop perturbative dynamics model which predict vehicle slip rates. The slip rates are first learned for steady state conditions and interpolated to slip rate surfaces. An Extended Kalman Filter uses the residual pose differences for on-line identification of the perturbative parameters on a six wheel, skid steered vehicle. An order of magnitude change in relative pose prediction was observed on loose and muddy gravel.

Acknowledgments

My advisor Al Kelly helped me develop much of the math and vehicle dynamics in this paper. His encyclopedic understanding of vehicle dynamics was critical in pushing me to understand every aspect of the algorithms while never simplifying the details. My thesis committee members, Thomas Howard and Mihail Pivtoraiko, were diligent and gave constructive criticism.

The engineers and researchers at NREC have been a great supportive team. I'd like to single out the USDA positioning team - Michael George, Jean-Philippe Tardif, and Michel Laverne - for helping me on prior projects. The SACR team - Jason Ziglar, Nicholas Chan, and Rob Meyers - helped collect the vehicle data and were always willing to help, even though I had a bad tendency to break their code.

The teachers and students of the Robotics Institute challenged me and pushed me. It is a great environment to tackle the many difficult and complex problems involved with robotics as they have for the past thirty years.

Long before I started at Carnegie Mellon, my family has support my educational curiosities - be they robotics, spacecrafts, or dance. For that I am very grateful.

Contents

1 Introduction **1**

1.1 VGMI Project 1

1.2 Thesis Overview 2

2 Related Work **3**

2.1 Autonomous Vehicles 3

2.2 Dynamic Modeling in Other Robotic Applications 3

2.3 Skid-Steered Vehicle Modeling 4

2.4 Predictive Models for Planning 6

3 Perturbative Dynamics Modeling **7**

3.1 Systematic Error Modeling 7

3.2 Linearized Systematic Perturbation Dynamics 8

3.3 Non-linear Systematic Perturbation Dynamics 9

4 Steady State **10**

4.1 Path Segmentation 10

4.2 Slip Rate Optimization 10

4.3 Uncertainty Modeling 12

4.4 Experimental Setup 17

4.4.1 Experimental Design 17

4.4.2 Vehicle Platform 17

4.5 Results 18

5 Slip Rate Surfaces **20**

5.1 Least Square Regression 20

5.1.1 Least Squares Regression Results 21

5.2 Gaussian Process Regression 21

5.2.1 Gaussian Process Regression Results 22

5.3 Bayes Linear Regression 23

5.3.1 Bayes Linear Regression Results 24

6 Transient Dynamics **26**
6.1 Vehicle Simulation 26
6.2 Gradient Descent 26
6.2.1 Gradient Descent Results 29
6.3 Extended Kalman Filter 29
6.3.1 Extended Kalman Filter Results 31
6.4 Comparison 32

7 Conclusions **34**

8 Future Work **35**
8.1 Model Improvements 35
8.1.1 Pose Extended Kalman Filter 35
8.2 Data Gathering 36
8.3 Incorporation of Perception and Terrain Prediction 36
8.4 Motion Planning 37

List of Figures

2.1 Bandit & Akoya 4

3.1 Vehicle Dynamic Model 7

4.1 Path Segments 11
4.2 Optimized Slip Curves from Path Segments 11
4.3 Optimized Uncertainty 15
4.4 Generated Path Segments via Learned Uncertainty 15
4.5 Terminal spread due to initial heading errors 16
4.6 Terminal spread due to process noise 16
4.7 Experiment Commands 17
4.8 Land Tamer Vehicle 18
4.9 Optimized Angular Slip Rates for Multiple Curvatures 18
4.10 Optimized Translational Slip Rates for Multiple Curvatures 19

5.1 Slip Surfaces learned via Least Square Regression 22
5.2 Slip Surfaces learned via Gaussian Process Regression 24
5.3 Slip Surfaces learned via Bayes Linear Regression 25

6.1 Simulated Vehicle Path Compared to Ideal Path 27
6.2 Simulated Vehicle Commands and Perturbations 27
6.3 Relative Pose Residuals 29
6.4 Parameter Convergence during Gradient Descent 30
6.5 EKF Results 31
6.6 Uncertainty Matrix 32
6.7 Slip Surfaces learned via EKF 33

8.1 Available Platforms 36

List of Tables

5.1 Gaussian Process Regression Algorithm 23

6.1 Slip Parameters Extended Kalman Filter 31
6.2 Transient Relative Pose Error Comparison 32

Chapter 1

Introduction

Mobile robotic dynamic modeling is necessary for reliable planning and control of unmanned ground vehicles (UGVs) on rough terrain. Autonomous vehicle research has continuously demonstrated that a platforms precise understanding of its own mobility is a key ingredient of competent machines with high performance. Since ground vehicles are propelled over the earth by the two forces of gravity and traction, agile autonomous mobility relies fundamentally on understanding and exploiting the interactions between the terrain and tractive devices like wheels and tracks.

I will investigate the feasibility and mechanism of enabling a platform to better predict its own mobility by learning from its own experience. It will calibrate vehicle models, in real-time, based on residual differences between the motion originally predicted by the platform and the motion ultimately experienced by the platform. Although the model does not consider the direct forces, it provides an accurate model of the underlying dynamics. This thesis develops an integrated perturbative dynamics model for real-time identification of wheel-terrain interaction that enhance autonomous vehicle mobility.

1.1 VGMI Project

The work presented in this thesis is part of the larger Vehicle Ground Model Identification (VGMI) project to improve the accuracy of predictive models which are the key enabler for high performance unmanned ground vehicles. The proposed work will:

- Gather data sets specifically targeted to on-line calibration of unmanned ground vehicle motion prediction algorithms
- Investigate numerous approaches for representing the terrain properties and the vehicle dynamics
- Elaborate the merits of each approach
- Fit systematic and stochastic models to the data off-line

- Develop methods for real-time calibration of the models while a vehicle operates
- Evaluate the merits of the identification system in full-up UGV experiments

The main objective of the work is to find a way to calibrate the specialized faster-than-real time models which are ubiquitous in unmanned ground vehicles. Such models are at one extreme of the spectrum where fidelity is relatively low and speed is relatively high. Adequate fidelity is defined as the capacity to adequately predict motion for the purposes of model predictive control and sequential motion planning.

On-line calibration (identification) techniques are necessary because one of the most influential aspects of wheeled vehicle motion prediction is wheel-terrain interaction and such interactions vary continuously on time scales far shorter than typical missions. Finally, the work will pursue the objective of investigating the use of terrain perception information in order to impart the modeling system with both associative memory and perceptive prediction of the properties of terrain in view. This technique is intended to better enable systems to adapt to terrain transitions before driving over them.

The project will use simulation, analysis, controlled experimentation and evaluative field testing to accomplish these goals. Simulation will be used to generate ground truth data during algorithm development and will be used for Monte Carlo studies. Analysis will be conducted on large data sets to assess the performance and robustness of candidate approaches. Controlled experiments will be conducted on several available robotic platforms to gather data under specific conditions. Field testing will demonstrate the value of the calibrated predictive models in improving Unmanned Ground Vehicle performance.

1.2 Thesis Overview

The following chapters develop the vehicle integrated perturbative dynamics model and show results from vehicle testing. Chapter 2 of this paper discusses previous work in mobile robotics related to off road vehicle modeling and estimation of slip parameters. The vehicle perturbative dynamics model is developed in Chapter 3 which describes how the slip rates affect the vehicles motion on rough terrain. This model is used to learn steady state perturbative parameters, along with the uncertainty, of a skid-steered vehicle in Chapter 4. Next, Chapter 5 expands these the optimized perturbative parameters to general surfaces interpolated for new vehicle states; three regression techniques are applied. In Chapter 6, I develop and test on-line filters to learn the slip rate surface parameters during general driving conditions with transient dynamics. The paper will conclude with an assessment of the lessons and observations that can be drawn from this work in Chapter 7 and a discussion of further areas of investigation for the future of the VGMI project in Chapter 8.

Chapter 2

Related Work

Autonomous vehicles operate in complex environments including field robotics, exploration robotics, agricultural and mining systems, autonomous automobiles, and mobile manipulators. All of these applications have vehicle models used for motion planning and navigation, yet very few of them explicitly handle perturbations such as slip. Of the slip based modeling, none are based off residual pose error based on integrated perturbative dynamics.

2.1 Autonomous Vehicles

Autonomous vehicles in city settings have recently become a reality as show by Boss, the autonomous car that on the 2007 DARPA Urban Challenge [28]. Recent advances in robotic ground vehicles have led to the autonomous traversal of increasingly complex terrain at higher rates of speed. With these advances, the motion of a vehicle becomes progressively more difficult to predict as wheel slip slip increases due to larger momentum. Solving this problem is essential for selecting commands to avoid collisions with obstacles.

Examples of ground vehicles using vehicle models include agricultural tractors [26] and the Crusher platform [4]. Unfortunately, these models do not normally take into account realistic terrain interactions that can adapt to terrain changes. These models are characterized by the coefficient of rolling resistance, the coefficient of friction, and the shear deformation modulus, which have terrain-dependent values. Information is not well known or measurable such as coefficient of friction in braking or maximum shear strength in aggressive turns. This causes the vehicle to skid and slip over the terrain. Tire to terrain interaction must be taken into account during motion prediction to know the set of feasible motions and reachable states [15].

2.2 Dynamic Modeling in Other Robotic Applications

Dynamic modeling and system identification has been applied to a number of robotic systems that are not ground vehicles. An acceleration based parameterization has been used for learning helicopter dynamics [1]. An kinematic implicit loop method was used to calibrate robotic mechanisms

using displacement measurements [32]; at heart, this is simply a least-square fit weighted to the variances of the sources of uncertainty. Gaussian Process Regression has been combined with an Unscented Kalman Filter for control of an autonomous blimp [17] [18].

In space, on-line dynamic modeling has been developed for proximity operations. Very small spacecraft are well-suited for proximity activities and are of growing interest in the aerospace community. However, due to size and power constraints, small vehicles cannot carry traditional precision navigation systems and generally have noisy sensor and actuator options. Developed at Washington University, the Bandit inspector vehicle docks with the parent Akoya spacecraft, Figure 2.1. Bayes Linear Regression models the time-varying thruster dynamics for the inspector spacecraft [23]. The BLR has replaced the noisy accelerometer as input to the Unscented Kalman Filter, improving performance [3].

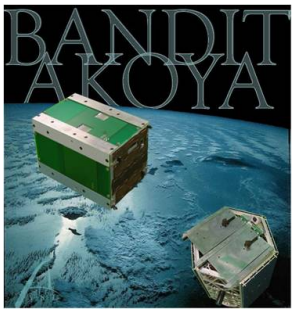


Figure 2.1: Bandit & Akoya

2.3 Skid-Steered Vehicle Modeling

The basic kinematic and dynamic equations of motion have been developed for skid-steered vehicles for flat surfaces with little or no friction or slip [19]. Using the rotational transformation, we can derive the world-frame kinematic equation from the body-frame longitudinal speed, v_x, lateral speed, v_y, and angular speed, ω.

$$\underline{\dot{q}} = \begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} \tag{2.1}$$

The left and right wheel speeds, ω_L and ω_R, control the longitudinal and angular speeds.

$$v_x = r\frac{\omega_L + \omega_R}{2}, \;\; \omega = r\frac{-\omega_L + \omega_R}{2c}, \tag{2.2}$$

while r is the effective wheel radius, and $2c$ is the spacing between wheel tracks. The lateral speed depends on the coordinate of instantaneous center of rotation (IRC), x_{IRC}. These equation describe the nonholonomic constraints of the system.

$$v_y + x_{ICR}\omega = 0 \tag{2.3}$$

In order to simplify the mathematical model of the skid-steered vehicle, the following assumptions are made in the previously mentioned paper:

- plane motion is considered only,
- achievable linear and angular velocities of the robot are relatively small,

- wheel contacts with surface at geometrical point (tire deformation is neglected),
- vertical forces acting on wheels are statically dependent on weight of the vehicle,
- viscous friction phenomenon is assumed to be negligible,

Other algorithms have been developed to learn soil parameters given wheel-terrain dynamic models [24]. The field of terra-mechanics identifies five primary material properties (soil cohesion, internal friction angle, sinkage coefficient, shear deformation modulus and the maximum shear before soil failure) which influence the forces generated by the terrain and in turn the resulting vehicle motion. Analytical models exist for steering maneuvers of planetary rovers on loose soil [12].

Related work takes into account the shear stress-shear displacement relationship on the track-ground interface assuming firm ground with minimal sinkage [33]. With track sinkage and bulldozing effect being neglected, the resultant shear stress, τ, on the track-ground interface is assumed to be an exponential function of shear displacement, j,

$$\tau = \tau_{max}(1 - e^{-j/K}) \tag{2.4}$$

where τ_{max} is the maximum shear stress between the track and the ground, and K is the shear deformation modulus.

Another method is to replace all the unknown soil parameters with slip ratios, i_L and i_R, and a slip angle, α. An Extended Kalman Filter (EKF) or a Sliding Mode Observer (SMO) can be developed to learn these parameters [35] [20] [25]. Sliding Mode Observer controllers have been proven to be robust enough to ignore the knowledge of the forces within the wheel-soil interaction, in the presence of benign sliding phenomena and ground level fluctuations [20] [21].

$$\dot{X} = v_x(\cos\theta + \sin\theta\tan\alpha) \tag{2.5}$$

$$\dot{Y} = v_x(\sin\theta - \sin\theta\tan\alpha) \tag{2.6}$$

$$\dot{\theta} = -\frac{v_x}{c} + r\frac{\omega_L(1-i_L)}{c} = \frac{v_x}{c} - r\frac{\omega_R(1-i_R)}{c} \tag{2.7}$$

Another technique models resistive wheel torques as functions of terrain properties and vehicle state. The terrain properties can be learned by placing current monitors on the wheels to calculate wheel torques instead of measuring forces [34]. This model can be extended to generate a power model of the vehicle [7].

Model-based approaches have been applied to estimate longitudinal wheel slip and detect immobilized conditions of mobile robots [31]. In this work, a explicitly differentiable traction/breaking model is used - expressed as a function of the wheel's relative velocity, rather than slip which may have singularities. The simplified model includes traction and rolling resistive forces.

$$F_{traction} = N(sign(v_{rel})C_1(1 - e^{-A_t|v_{rel}|}) + C_2 v_{rel}) \tag{2.8}$$

$$v_{rel} = r\omega - v_{fwd}$$

$$F_{roll,res} = -sign(v_{fwd})N(R_1(1 - e^{-A_{roll}|v_{fwd}|}) + R_2|v_{fwd}|) \tag{2.9}$$

where N is the normal wheel force, and the rest of the constants are specific to the vehicle-terrain interaction.

An artificial neural network was used to learn a forward predictive model of the Crusher Unmanned Ground Vehicle [4]. The model gave good results, with fast predictions, without making any assumptions about the vehicle to ground interaction. The downsides of this algorithm are the off-line training and the assumed non-varying terrain properties.

2.4 Predictive Models for Planning

Vehicle predictive models are necessary, when wheel slip is non-negligble, for higher level controls such as obstacle avoidance and regional mobility planning. Efficient inversion of the equations of motions is necessary to compute the precise controls needed to achieve a desired position and orientation while following the contours of the terrain under arbitrary wheel terrain interactions [16].

Classic motion-planning techniques sample in the space of controls to ensure feasible local motion plans. When environmental constraints severely limit the space of acceptable motions or when global motion planning expresses strong preferences, a state space sampling strategy is more effective. A model-predictive trajectory generation approach is necessary for state space sampling while satisfying the constraints of vehicle dynamic feasibility [11]. A special discretization of the state space, in a set of elementary motions via feasible motions, allows employing standard search algorithms for solving the motion planning problem [29] [30].

Chapter 3

Perturbative Dynamics Modeling

In this chapter I develop a a dynamic model with unknown parameters which are calibrated as the system runs. Errors are expressed in terms of perturbations to a dynamic model since this is the required form of the eventual prediction system. Such a system integrates the dynamics to predict motion anyway; so, its straightforward to compare predictions of where you are now to measurements of where you are now. Velocity is not always easy to measure well. RTK GPS is sometimes available and it is an excellent source of ground truth information, so state residuals (pose) are used instead of residuals in state derivatives (velocity).

3.1 Systematic Error Modeling

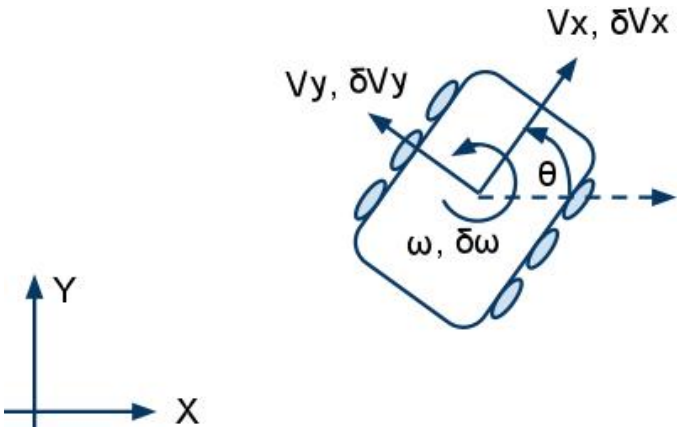


Figure 3.1: Vehicle Dynamic Model

For any vehicle moving in contact with a surface, there are three degrees of freedom of in-plane motion as long as the vehicle remains in contact with the local tangent plane, Figure 3.1. Errors in motion prediction can therefore be reduced, without loss of generality, to instantaneous values of forward slip rate, δV_x, side slip rate, δV_y, and angular slip rate, $\delta\omega$. I choose to express these in the coordinates of the body frame - where they are likely to be constant under steady state conditions.

Given these slip rates along with the vehicle's commanded forward velocity and curvature we have the following kinematic differential equation for the vehicle's 2D position and heading with respect to a ground fixed frame of reference.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} c\theta & -s\theta & 0 \\ s\theta & c\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \left\{ \begin{bmatrix} V_x \\ V_y \\ \omega \end{bmatrix} + \begin{bmatrix} \delta V_x \\ \delta V_y \\ \delta\omega \end{bmatrix} \right\} \tag{3.1}$$

$$c\theta = \cos(\theta), \quad s\theta = \sin(\theta)$$

For a symmetric skid-steered vehicle, side velocity, V_y, is zero. Using these equations, we have a function to predict the vehicle's future pose by integrating the above equations. In my implementation, Runge-Kutta 4th-order integration was used. In the planar case, curvature can be converted to angular velocity using:

$$\omega = \kappa V \tag{3.2}$$

The general form of deferential motion is:

$$\dot{\underline{x}} = \underline{f}(\underline{x}, \underline{u}, \underline{\delta u}) \tag{3.3}$$

Where:

- the state vector is $\underline{x} = [x \;\; y \;\; \theta]^\top$
- the input vector is $\underline{u} = [V \;\; \omega]^\top$
- and the error is modeled as if it was caused by an extra forcing function denoted $\underline{\delta u} = [\delta V_x \;\; \delta V_y \;\; \delta\omega]^\top$

This model is the general case. The question is how best to model the perturbations $(\delta V_x, \delta V_y, \delta\omega)$. They certainly depend on terrain composition, vehicle state and all of the applied, constraint, and inertial forces on the vehicle. Note especially that while there are two real inputs for a skid-steered vehicle, I have chosen to represent three error inputs because the state vector is 3D and it is likely that three errors can be observed as a result.

3.2 Linearized Systematic Perturbation Dynamics

The effects of the perturbations on the state can be estimated by linearizing the perturbation dynamics, (3.4). The linearized equations are necessary to calibrate random error [14].

$$\dot{\underline{\delta x}} = F(t)\underline{\delta x}(t) + G(t)\underline{\delta u}(t) \tag{3.4}$$

where:

$$F(t) = F(\underline{x}, \underline{u}, t) = \partial f / \partial x \tag{3.5}$$

$$G(t) = G(\underline{x}, \underline{u}, t) = \partial f / \partial u \tag{3.6}$$

The integrated effect of these perturbations is given by the vector convolution integral:

$$\underline{\delta x}(t) = \Phi(t, t_0)\underline{\delta x}(t_0) + \int_0^t \Phi(t, \tau)G(\tau)\underline{\delta u}(\tau)\, d\tau \tag{3.7}$$

In discrete time this is:

$$\underline{\delta x}(n) = \Phi(n, 0)\underline{\delta x}(0) + \Sigma_0^n \Phi(n, k)G(k)\underline{\delta u}(k)\, \Delta t \tag{3.8}$$

3.3 Non-linear Systematic Perturbation Dynamics

Alternatively, the original nonlinear dynamics can be integrated. This is easier for prediction when the random error is not directly calibrated.

$$\begin{bmatrix} x(t) \\ y(t) \\ \theta(t) \end{bmatrix} = x(t_0) + \int_0^t \underline{f}(\underline{x}(\tau), \underline{u}(\tau), \underline{\delta u}(\tau))\, d\tau \tag{3.9}$$

It is convenient to ignore the fact that this is an integral and represent it in the abstract integrated form:

$$\underline{x}(t) = \underline{x}(t_0) + \underline{F}[\underline{u}(t), \underline{\delta u}(t), t] \tag{3.10}$$

where the vector integral $\underline{F}$ should not be confused with the matrix system Jacobian F.

This can always be done for definite integrals since they are, in effect, functions of their limits of integration as well as the non-state variables in the integrand. Notationally, its important to remember that F depends on the entire history of its inputs, not just their present values. It is traditional to distinguish a functional from a function with square brackets as shown. Clearly, this can be rewritten in terms of the change in state over the time interval thus:

$$\Delta\underline{x}(t) = \underline{F}[\underline{u}(t), \underline{\delta u}(t), t] \tag{3.11}$$

Chapter 4

Steady State

To simplify development and testing, I initially constrained the problem to learning the perturbative dynamics model during steady state conditions. Path segments of constant commanded speed and curvature where group together to optimize the fixed slip rates and associated uncertainties. In Chapter 5, I generate the general slip rate surfaces by using all path segments. The steady state assumptions are relaxed in Chapter 6 to develop on-line dynamic modeling.

4.1 Path Segmentation

While the instantaneous slip rates can be determined by solving the state equations given predictions and measurements of velocity, the velocity measurements are either not available or not very accurate. Instead, measured path segments, of constant commanded velocity and curvature, are used to optimize the slip rates. This technique is applicable in general for any dynamic model and any dimension of error vector - not just errors that can be resolved instantaneously from the state equations. After collecting the path segments, the start of each path segment is translated and rotated to the origin. The experiment was conducted by driving a skid steer vehicle in an endless circle at steady-state commanded velocity and curvature.

Figure 4.1 show collected 3 second path segments of the vehicle moving at a constant speed of 0.5 m/s with a curvature curvature of 0.5 m^{-1}. The green lines are actual responses referenced to their local origin (initial pose) in order to assess capacity for relative motion prediction. The variance of the paths are due to terrain differences, vehicle dynamics, and control variance. The magenta dashed line is the ideal path the vehicle would take with no slip given the commanded speed and curvature. The blue dashed line is the 2nd order polynomial best fit line for the path segments - this fits well, but tells us nothing about how the vehicle slips.

4.2 Slip Rate Optimization

The measured position and orientation along path segments are used to determine the slip rates. A weighted squared residual cost function is created and minimized along the sampled path segments.

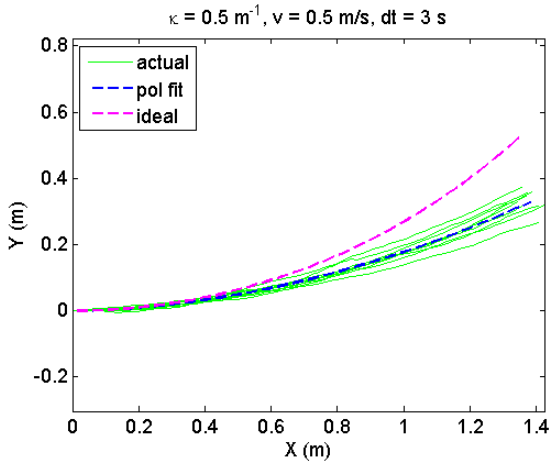
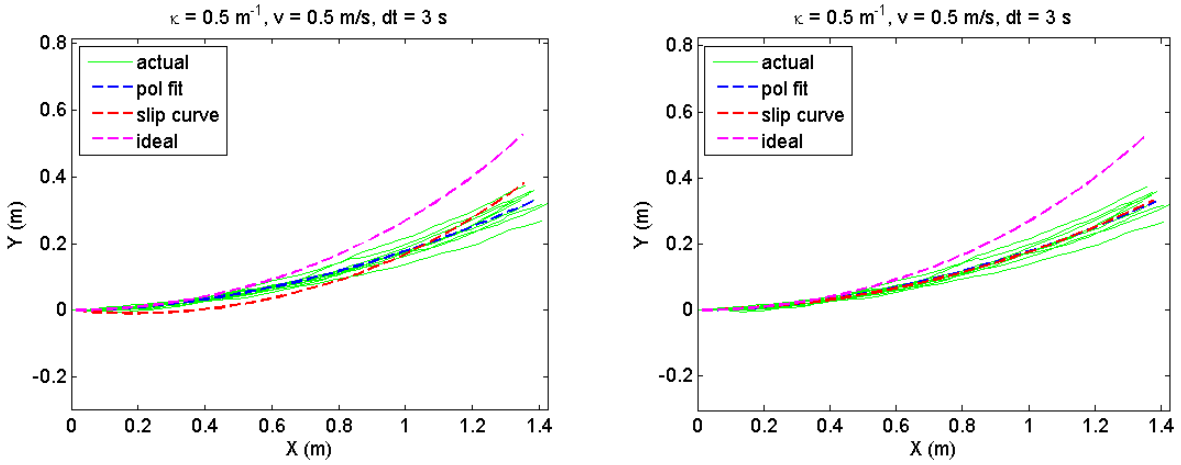


Figure 4.1: Collected Path Segments from the Land Tamer

L is a characteristic length used for relative weighting of the error between heading and position.

$$Cost = min_{\underline{\delta u}} \sum_{path\ j} \sum_{t} (x_j(t) - x(\underline{\delta u}, t))^2 + (y_j(t) - y(\underline{\delta u}, t))^2 + L * (\theta_j(t) - \theta(\underline{\delta u}, t))^2 \qquad (4.1)$$

The poses $x(\underline{\delta u}, t)$, $y(\underline{\delta u}, t)$, and $\theta(\underline{\delta u}, t)$ represent the path that would have been followed with the indicated slip parameter vector. L is a characteristic length used for weighting the error between heading and position. Essentially, there is a predicted and measured pose (x, y, θ) for every point in time for every path segment and the objective function is simply the sum of the squared distances between all such measured and predicted corresponding points.



(a) Optimized with Translational Slip Only

(b) Optimized With Translational and Angular Slip

Figure 4.2: Optimized Slip Curves from Path Segments

The slip rates were optimized using Matlab's non-linear optimization, *fminunc* or *fminsearch*,

with the cost function given in Equation (4.1). When the model accounts only for translational slip, $(\delta V_x, \delta V_y)$, the best fit solution does not fit well; see red dot-dash line in Figure 4.2(a). When angular slip, $\delta\omega$ is included, the predicted slip path is the same as the best fit polynomial, Figure 4.2(b). The deviation of green lines from the ideal red dashed line can be explained by either lateral slip or a rotation around the initial point. A polynomial can express any function according to the Taylor remainder theorem. Since the optimized slip path matches the second order polynomial, I have shown that the data can be fit correctly with a three degree of freedom slip model.

4.3 Uncertainty Modeling

The same techniques can be used to identify uncertainty. A model of uncertainty propagation is needed and it must be formulated with unknown parameters which are then solved for in an optimization framework. The propagation of error through a system dynamics differential equation is a standard component of every Kalman filter. It is known also as the linear variance equation. In discrete time form:

$$P(k+1) = \Phi(k)P(k)\Phi(k)^\top + \Gamma(k)Q(k)\Gamma(k)^\top \tag{4.2}$$

This is a recursive equation in state uncertainty P where the Q matrix is the forcing function. Let's define three constant noise sources thus:

$$Q = diag[\sigma_{uu} \;\; \sigma_{vv} \;\; \sigma_{ww}]\Delta t = diag[\underline{\sigma}]\Delta t \tag{4.3}$$

The integrated effect of this equation can be written in discrete time as:

$$P(n) = \Phi(n,0)P(0)\Phi(n,0)^\top + \Sigma_0^n \Phi(n,k)G(k)Q(k)G(k)^\top \Phi(n,k)^\top \; \Delta t \tag{4.4}$$

These quantities are the variance of the noises in the slip velocities. The input Jacobian is the Jacobian of the system dynamics with respect to the slip velocities. The Jacobian with respect to the slip velocities is:

$$\Gamma = \frac{\partial \underline{\dot{x}}}{\partial \underline{\delta u}} = \begin{bmatrix} c\theta & -s\theta & 0 \\ s\theta & c\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.5}$$

The Jacobian with respect to the states is:

$$F = \frac{\partial \underline{\dot{x}}}{\partial \underline{x}} = \begin{bmatrix} 0 & 0 & -(V+\delta V_x)s\theta - \delta V_y c\theta \\ 0 & 0 & (V+\delta V_x)c\theta - \delta V_y s\theta \\ 0 & 0 & 0 \end{bmatrix} \tag{4.6}$$

Substituting the equations of motion (3.1), this can also be written as:

$$F = \frac{\partial \dot{\underline{x}}}{\partial \underline{x}} = \begin{bmatrix} 0 & 0 & -\dot{y} \\ 0 & 0 & \dot{x} \\ 0 & 0 & 0 \end{bmatrix} \tag{4.7}$$

Hence, the transition matrix is:

$$\Phi \approx I + F\Delta t = \begin{bmatrix} 1 & 0 & -\dot{y}\Delta t \\ 0 & 1 & \dot{x}\Delta t \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\Delta y \\ 0 & 1 & \Delta x \\ 0 & 0 & 1 \end{bmatrix} \tag{4.8}$$

Once an initial state uncertainty P_0 is chosen, the covariance propagates in a manner that depends on the input velocities.

The slip velocity noises are calibrated by computing a scatter matrix of a large number of observations thus:

$$S = \frac{1}{n-1}\Sigma \begin{bmatrix} \delta x_i \delta x_i & \delta x_i \delta y_i & \delta x_i \delta \theta_i \\ \delta y_i \delta x_i & \delta y_i \delta y_i & \delta y_i \delta \theta_i \\ \delta \theta_i \delta x_i & \delta \theta_i \delta y_i & \delta \theta_i \delta \theta_i \end{bmatrix} \tag{4.9}$$

Where δx_i etc. is the deviation of x_i from either the mean of all x_is at the same point in time (precise definition) or the deviation of x_i from the best fit curve (if you want it to mean the variance from the best fit curve).

The elements of this scatter matrix are denoted:

$$S = \begin{bmatrix} s_{xx} & s_{xy} & s_{x\theta} \\ s_{yx} & s_{yy} & s_{y\theta} \\ s_{\theta x} & s_{\theta y} & s_{\theta\theta} \end{bmatrix} \tag{4.10}$$

The elements of uncertainty matrix, P are denoted:

$$P = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{x\theta} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{y\theta} \\ \sigma_{\theta x} & \sigma_{\theta y} & \sigma_{\theta\theta} \end{bmatrix} \tag{4.11}$$

Only six of these quantities are independent. Equating these to the observations gives us six

equations in 3 unknowns:

$$\begin{aligned} s_{xx}(t) &= \sigma_{xx}(\underline{\sigma}, t) \\ s_{xy}(t) &= \sigma_{xy}(\underline{\sigma}, t) \\ s_{x\theta}(t) &= \sigma_{x\theta}(\underline{\sigma}, t) \\ s_{yy}(t) &= \sigma_{yy}(\underline{\sigma}, t) \\ s_{y\theta}(t) &= \sigma_{y\theta}(\underline{\sigma}, t) \\ s_{\theta\theta}(t) &= \sigma_{\theta\theta}(\underline{\sigma}, t) \end{aligned} \tag{4.12}$$

These six equations can be solved for the three unknown variances by minimizing the objective function:

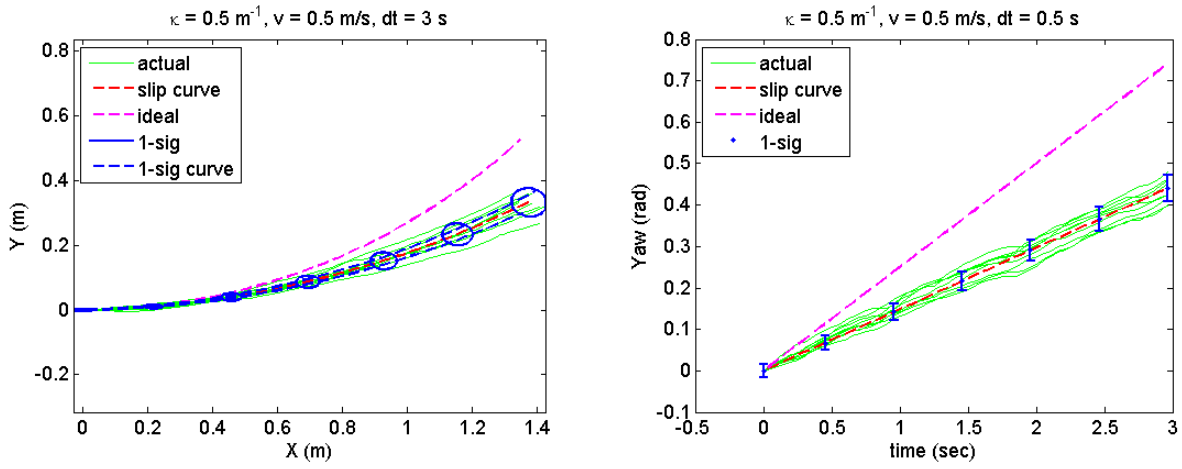
$$Cost_{uncert} = \min_{\underline{\sigma}} \sum_{t} (s_{xx}(t) - \sigma_{xx}(\underline{\sigma}, t))^2 + (s_{xy}(t) - \sigma_{xy}(\underline{\sigma}, t))^2 + \cdots \tag{4.13}$$

with respect to the parameters $\underline{\sigma} = [\sigma_{uu} \quad \sigma_{vv} \quad \sigma_{ww}]$. In addition, the initial uncertainty elements can be simultaneously solved, $P_0 = diag[\sigma_{xx,0}, \sigma_{yy,0}, \sigma_{\theta\theta,0}]$. Here $\sigma_{xx}(\underline{\sigma}, t)$ and $\sigma_{xy}(\underline{\sigma}, t)$ etc. represent the variance that is predicted from the present values of the parameters. There is a predicted and measured covariance matrix for every point in time for a set of path segments and the objective function is simply the sum of the squared distances between the covariances at all such measured and predicted corresponding points. Note that it would also be possible in principle to compute the result from a single path segment at a single point since the scatter matrix at one point provides six constraints on the three unknowns. Whether this is practical remains to be seen.

Figure 4.3 shows the results of optimizing the cost function, via fminunc, for a set of path segments of constant speed and curvature. The ellipses are 1-sigma error bounds. Notice that the first ellipse is due to P_0. The blue dashed line shows the predicted paths of adding and subtracting one standard deviation from the slip rates of the best fit slip curve, $f(\delta V_x \pm \sigma_u,\ \delta V_y \pm \sigma_v,\ \delta\omega \pm \sigma_\omega)$.

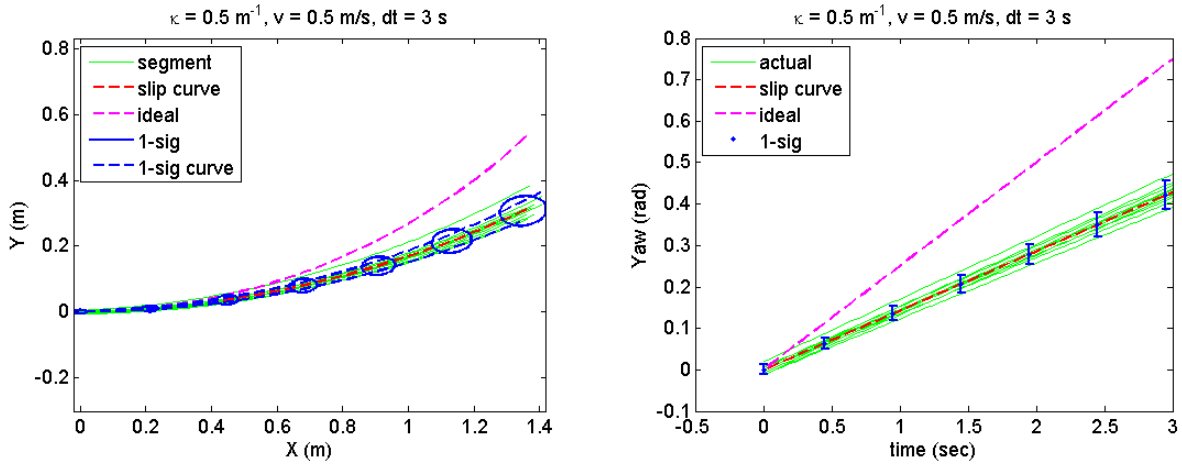
Using the learned slip rate parameters, new path segments can be generated by sampling using the learned uncertainty parameters, Figure 4.4.

The path segment generation allows us to separate the terminal spread due to initial heading errors, P_0, and spread due to the process noise, Q, Figures 4.5 and 4.6 respectively.



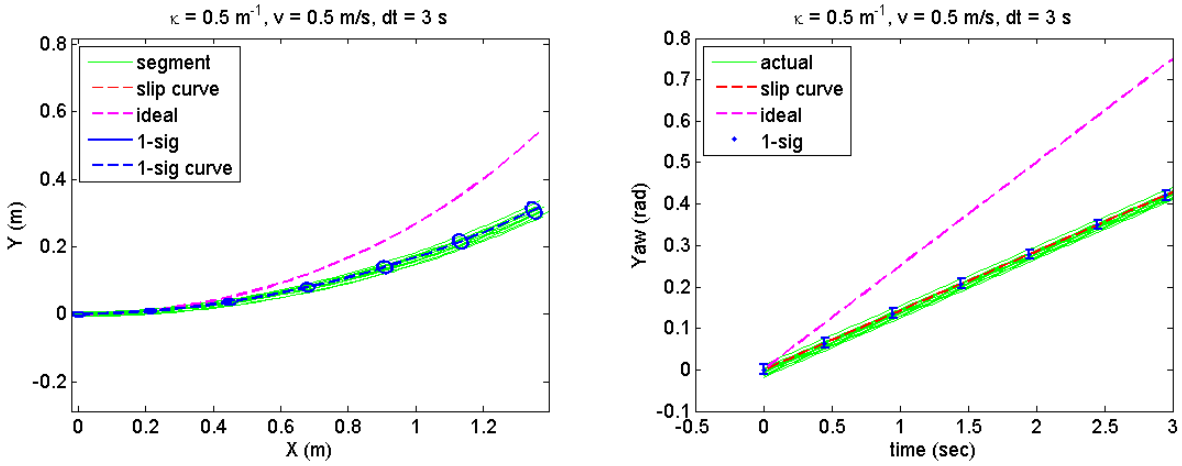
(a) Position (b) Orientation

Figure 4.3: Path Segments with Uncertainty Learned via Scatter Matrix



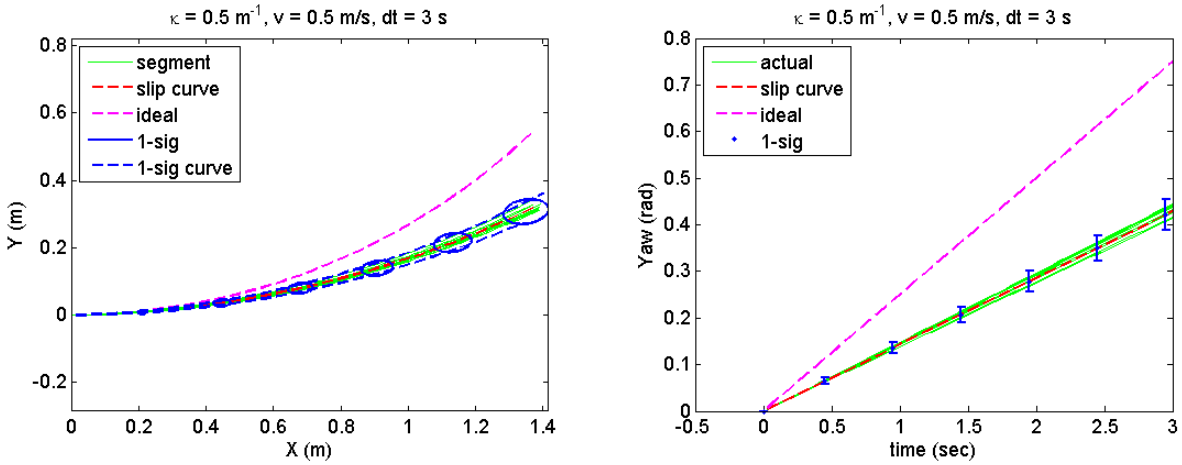
(a) Position (b) Orientation

Figure 4.4: Generated Path Segments via Learned Uncertainty



(a) Position (b) Orientation

Figure 4.5: Terminal spread due to initial heading errors



(a) Position (b) Orientation

Figure 4.6: Terminal spread due to process noise

4.4 Experimental Setup

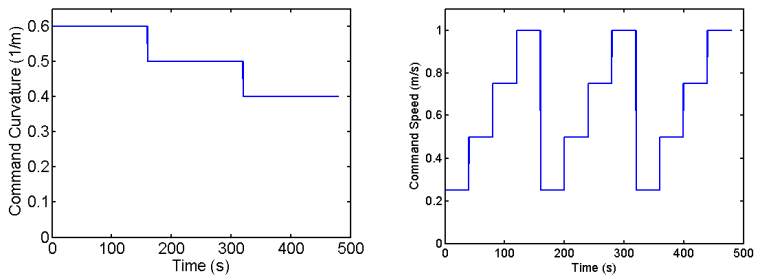
4.4.1 Experimental Design

An autonomous vehicle needs to predict relative motion from its present pose to a predicted pose a few seconds into the future. The basic procedure is to collect a data log of time-tagged commands and measured poses on an on appropriate vehicle, in appropriate terrain (mechanical, geometry), for appropriate command sequences. This log is separated into prediction segments, which can overlap. The relative poses are predicted over these path segments and compared to the ground truth. The residuals are used to identify a perturbative model.

For the first test, the experimental setup was simplified for rapid testing of system identification algorithms as follows:

- level homogeneous terrain (loose gravel)
- fixed commanded curvature and slow speed
- closed loop speed control per side

Path segments of constant speed and curvature are used to determine the slip parameters by optimization of the cost function, Figure 4.7. The intent is to validate the approach from the perspective of observability and disambiguation of the parameters of interest under benign conditions.



(a) Curvature (b) Speeds

Figure 4.7: Experiment Commands

4.4.2 Vehicle Platform

Data has been collected on the Land Tamer, a six wheeled skid-steered vehicle with a hydraulic/gear drive system, Figure 4.8. Skid-steered vehicles are often used as outdoor mobile robots due to their robust mechanical structure and high maneuverability.

For such a vehicle, execution of paths with any curvature will create wheel slip which makes both kinematic and dynamic modeling difficult. The Land Tamer vehicle is retrofitted with cameras, Lidar, and an high end Novetel SPAN Inertial Measuring Unit (IMU) unit with Real-Time

Figure 4.8: Land Tamer Vehicle

Kinematic Global Positioning System (RTK GPS). The RTK base station allows for centimeter accuracy of the vehicle's position.

4.5 Results

The slip rates were optimized for all the commanded speeds and positive curvatures, Figures 4.9 and 4.10. The scatter matrix technique provides the 1-σ uncertainty bars. The angular slip rate is linear for a constant curvature and speed. Forward and side slip rates are not linear, but are a relatively insignificant source of slip and close to constant. The 1-σ bars were learned via scatter matrix optimizations. Even though angular slip is dominant for this vehicle, most other proposed vehicle slip models do not take it into account.

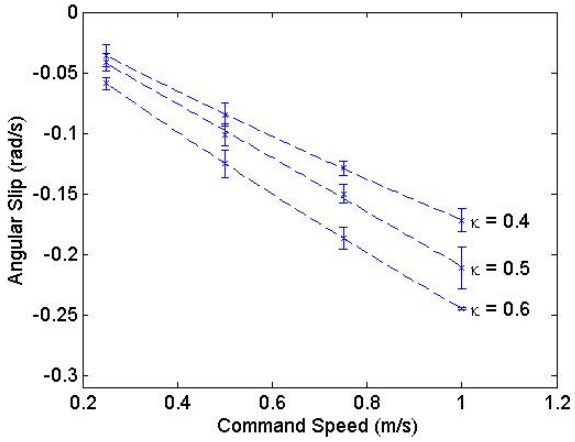
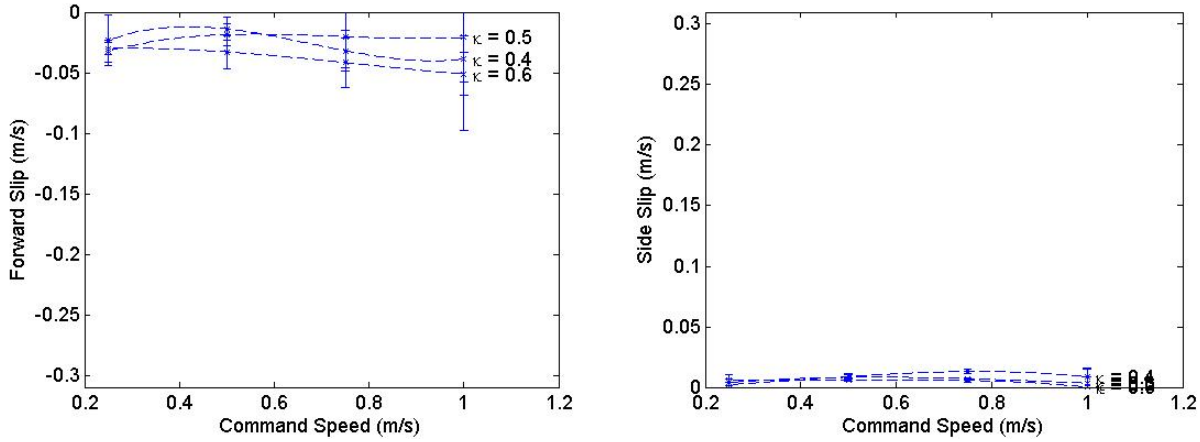


Figure 4.9: Optimized Angular Slip Rates for Multiple Curvatures

The integrated observer formulation which integrates the slip model to predict pose and then



(a) Forward (b) Side

Figure 4.10: Optimized Translational Slip Rates for Multiple Curvatures

compares this to actual RTK pose data works very well. In the past, fitting models based on poor GPS velocity observations gave poor results. Given a general formulation that uses three degrees of freedom of slip, the optimization was able to fit the data for a worst case vehicle.

There is a clear ambiguity in the problem because many errors can be explained either by accumulated slip or by errors in initial conditions. Hence observed path segments need to be long enough to disambiguate the two; I found only 2 meters to be adequate. More sophisticated formulations may have to assign uncertainty-based weights to all error sources for rapid convergence.

Chapter 5

Slip Rate Surfaces

In Chapter 4, slip rates were optimized for a specific steady state. To generalize the surface to other speeds and curvatures I represent each slip rate as a second order linear surface. The constant terms were not used to remove phantom drift when the vehicle is not moving. Side velocity was not considered as skid steered vehicles has no commanded side velocity. Additional terms may be added for other state variables.

$$\begin{aligned} \delta V_x &= \alpha_{1,x}\,\kappa + \alpha_{2,x} V + \alpha_{3,x}\,\kappa V + \alpha_{4,x}\,\kappa^2 + \alpha_{5,x} V^2 \\ \delta V_y &= \alpha_{1,y}\,\kappa + \alpha_{2,y} V + \alpha_{3,y}\,\kappa V + \alpha_{4,y}\,\kappa^2 + \alpha_{5,y} V^2 \\ \delta V_\omega &= \alpha_{1,\omega}\,\kappa + \alpha_{2,\omega} V + \alpha_{3,\omega}\,\kappa V + \alpha_{4,\omega}\,\kappa^2 + \alpha_{5,\omega} V^2 \end{aligned} \tag{5.1}$$

I present three regression techniques to learn the slip surface parameters, $\underline{\alpha}$: Least Square Regression, Gaussian Process Regression, and Bayes Linear Regression.

5.1 Least Square Regression

From the collected data, two matrices are constructed. The first matrix holds all the polynomial terms from the given commands. The second matrix includes all the slip rates learned via the optimization for each path segment.

$$\mathbf{A} = \begin{bmatrix} \kappa_1 & V_1 & \kappa_1 V_1 & \kappa_1^2 & V_1^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \kappa_N & V_N & V_N \kappa_N & \kappa_N^2 & V_N^2 \end{bmatrix} \tag{5.2}$$

$$\mathbf{b} = \begin{bmatrix} \delta V_{x,1} & \delta V_{y,1} & \delta V_{\omega,1} \\ \vdots & \vdots & \vdots \\ \delta V_{x,N} & \delta V_{y,N} & \delta V_{\omega,N} \end{bmatrix} \tag{5.3}$$

$$\mathbf{A}\underline{\alpha} = \mathbf{b} \tag{5.4}$$

We now have a overdetermined system and can solve for the polynomial coefficients via basic Least Square Regression which minimizes the sum of squared residuals.

$$\underline{\alpha} = (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \mathbf{b} \tag{5.5}$$

A similar technique is used to learn the variance of the slip rates. Once the slip surface coefficients are learned, the residual differences between the surface and measured slip rates are used to determine the coefficients of the standard deviation surface.

$$\begin{aligned} \delta\sigma_x &= s_{1,x}\,\kappa + s_{2,x} V + s_{3,x}\,\kappa V + s_{4,x}\,\kappa^2 + s_{5,x} V^2 \\ \delta\sigma_y &= s_{1,y}\,\kappa + s_{2,y} V + s_{3,y}\,\kappa V + s_{4,y}\,\kappa^2 + s_{5,y} V^2 \\ \delta\sigma_\omega &= s_{1,\omega}\,\kappa + s_{2,\omega} V + s_{3,\omega}\,\kappa V + s_{4,\omega}\,\kappa^2 + s_{5,\omega} V^2 \end{aligned} \tag{5.6}$$

$$r_i = \left\| \underline{\alpha}^\top \begin{bmatrix} \kappa_i & V_i & \kappa_i V_i & \kappa_i^2 & V_i^2 \end{bmatrix} - \underline{\delta V} \right\| \tag{5.7}$$

$$\underline{s} = (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \mathbf{r} \tag{5.8}$$

5.1.1 Least Squares Regression Results

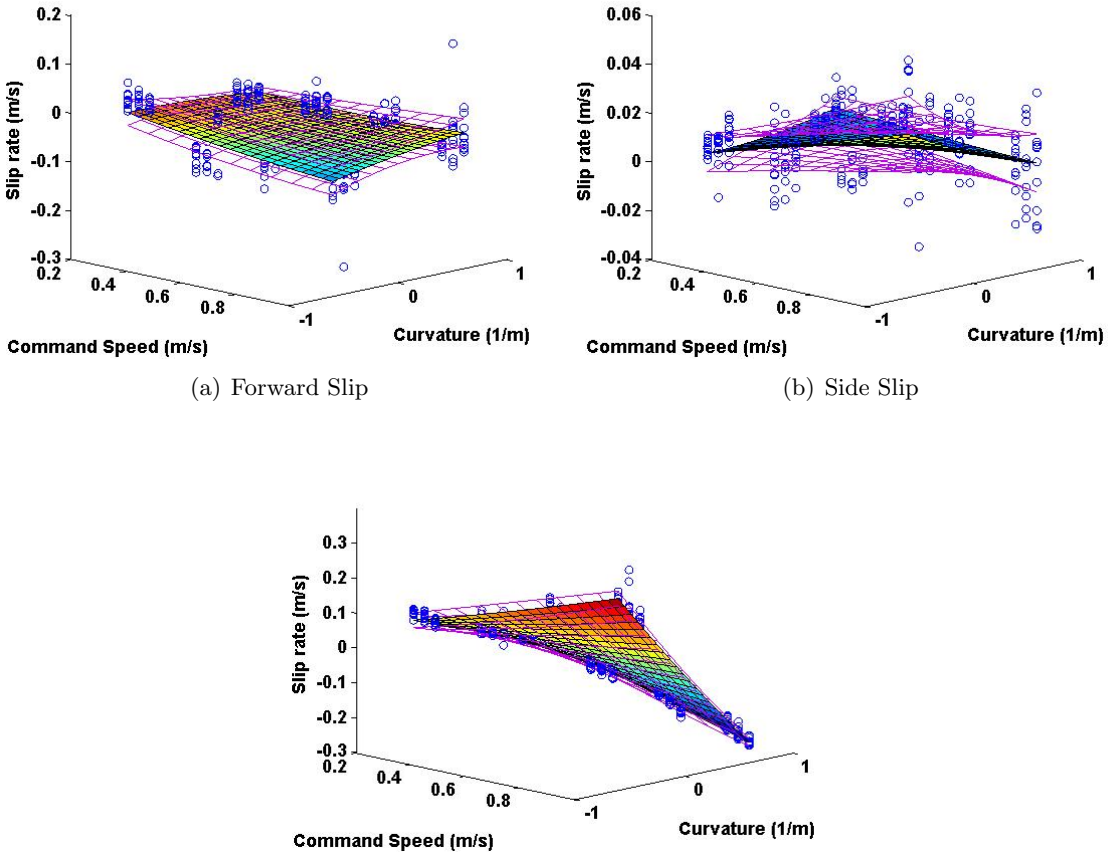
The Land Tamer vehicle was commanded to drive in a set of continuous circles at four different speeds and three positive curvatures for twelve combinations. This process was also repeated with negative curvature. The paths were separated into 3 second long path segments, of constant curvature and speed, which were optimized for the slip rates. These slip rates were used in the Least Square Regression to determine the second order coefficients of the slip surfaces and variance.

Figure 5.1 shows the three slip rate surfaces. Each circle is the learned slip rate for a given path segment used as input. The color surfaces are the mean expected values, and the transparent surfaces are the 1-σ uncertainty surfaces. Note that the z-axis scaling is not constant for the three graphs and that the angular slip rate varies the greatest.

5.2 Gaussian Process Regression

A simple linear regression cannot capture the non-linear dynamics of the ground interaction. One option would be a Bayes Linear Regression with additional non-linear basis functions, but this cannot capture all the dynamics and added biases to the included terms. To capture the non-linear dynamics, a Gaussian Process Regression (GPR) was implemented. One can think of a Gaussian process as defining a distribution over functions, and inference taking place directly in the space of functions. Each training point is stored as a kernel with a given weight. Given the training data X and Y with a new example, the outcome and variance can be predicted using the following equations.

$$\mu_{y_t} = K(x_t, \mathbf{X}) K(\mathbf{X}, \mathbf{X})^{-1} \mathbf{Y} \tag{5.9}$$



(a) Forward Slip

(b) Side Slip

(c) Angular Slip

Figure 5.1: Slip Surfaces learned via Least Square Regression

$$\Sigma_{y_t} = K(x_t, x_t) - K(x_t, \mathbf{X})K(\mathbf{X}, \mathbf{X})^{-1}K(\mathbf{X}, x_t) \tag{5.10}$$

A complete explanation of the GPR algorithm can be found in [6]. The *gpml* code library was used for testing. Table 5.1 shows the complete algorithm. Cholesky decomposition speeds up the matrix inversion in line 1. The kernel matrix, K, stores kernel covariance values between all inputs. k_* is a vector of kernel values between the test input and other input values.

5.2.1 Gaussian Process Regression Results

Using the same training examples as the Least Square Regression, slip rate surfaces were generated via Gaussian Process Regression, Figure 5.2. GPR gave similar results but made no assumptions about underline basis functions as all the training examples are stored as kernels.

Algorithm *Gaussian Process Regression*
Input: X (inputs), y (target), k (covariance function), σ_n^2 (noise level), x_* (test input)
1. $L = cholesky(K + \sigma_n^2 I)$
2. $\alpha = L^\top \ (L \ y)$
3. $\hat{f}_* = k_*^\top \alpha$
4. $v = L \ k_*$
5. $V[f_*] = k(x_*, x_*) - v^\top v$
6. $\log p(y|X) = -\frac{1}{2} y^\top \alpha - \Sigma_i \log L_{ii} - \frac{n}{2} \log 2\pi$
7. **return** $\hat{f}_*$ (mean), $V[f_*]$ (variance), $\log p(y|X)$ (log marginal likelihood)

Table 5.1: Gaussian Process Regression Algorithm

5.3 Bayes Linear Regression

Bayes Linear Regression (BLR) is a method of on-line linear regression which makes the assumption that there is a normal Gaussian matching between input data $\vec{x}$, and output data $\vec{y}$,

$$\vec{y}_t = \vec{\theta}^\top \vec{x}_t + \vec{\epsilon}_t \tag{5.11}$$

for a given set of weights, $\vec{\theta}$, and Gaussian noise, $\vec{\epsilon}_t \sim N(0, \sigma^2)$. In the the natural or canonical Gaussian parameterization, we can find the updated probability distribution of the weights given a new data point, $(\vec{x}, \vec{y})$:

$$\begin{aligned} p(\vec{y}|\vec{x}, \vec{\theta}) p(\vec{\theta}) &\propto e^{\frac{-(y-\theta^\top x)^2}{2\sigma^2}} e^{\frac{-1}{2}\theta^\top P \theta + J^\top \theta} \\ &= e^{\frac{-1}{2}\theta^\top (P - \frac{xx^\top}{\sigma^2})\theta + (J + y\frac{yx^\top}{\sigma^2})^\top \theta} \end{aligned} \tag{5.12}$$

For each training set, simple updates rules refresh the information matrix, P, and information vector, J,

$$P \leftarrow P + \frac{\vec{x}_t \vec{x}_t^\top}{\sigma^2} \tag{5.13}$$

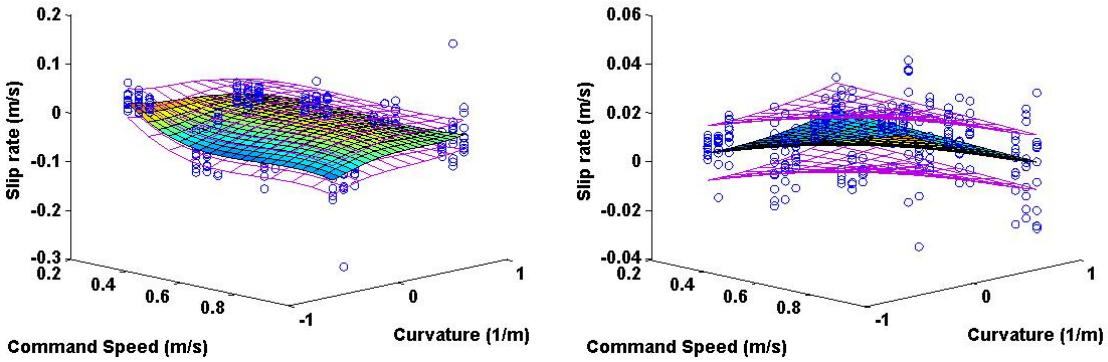
$$J \leftarrow J + \frac{\vec{y}_t \vec{x}_t^\top}{\sigma^2} \tag{5.14}$$

The same equations can be used to remove a training set from the regression via subtraction; this can create a sliding window of training examples to track time dependent changes. It is easy to transform back to the moment parameterization to find the predicted value, $\vec{y}$ as well as the weight vector, $\vec{\mu}_\theta$ and its variance Σ_θ

$$\vec{\mu}_\theta = P^{-1} J \tag{5.15}$$

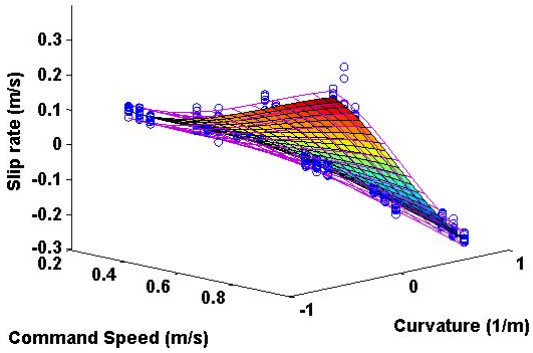
$$\Sigma_\theta = P^{-1} \tag{5.16}$$

$$\vec{y} = (P^{-1} J)^\top \vec{x} \tag{5.17}$$



(a) Forward Slip

(b) Side Slip



(c) Angular Slip

Figure 5.2: Slip Surfaces learned via Gaussian Process Regression

The output, $\vec{y}$, is linear in $\vec{x}$, so the predictive variance is just

$$\Sigma_y = \vec{x}^\top \Sigma_\theta \vec{x} \tag{5.18}$$

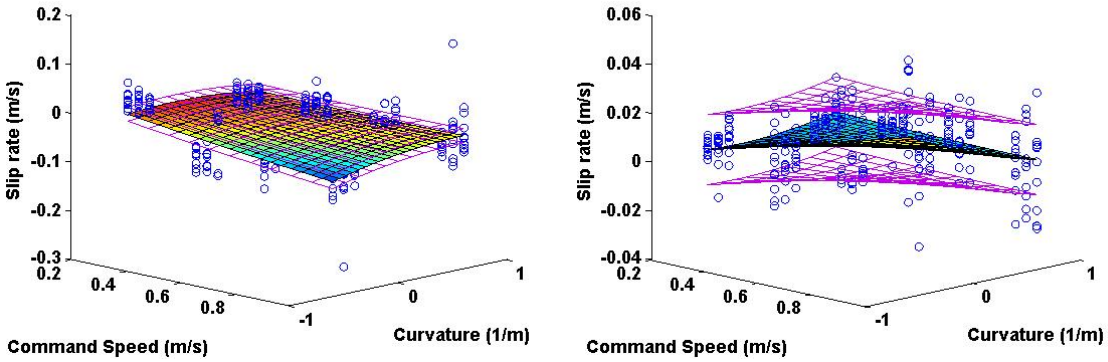
which is a scalar. This only captures uncertainty due to the parameters. The total uncertainty includes the noise variance.

$$\Sigma_y = \vec{x}^\top \Sigma_\theta \vec{x} + \sigma^2 \tag{5.19}$$

5.3.1 Bayes Linear Regression Results

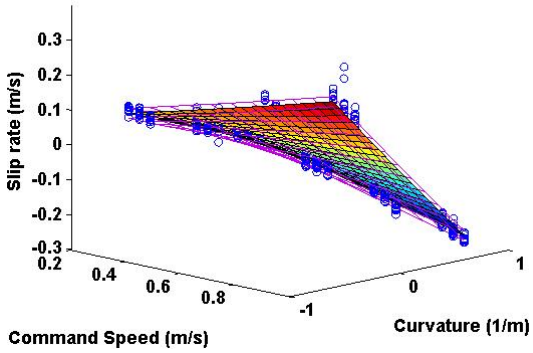
Using the same training examples as the Least Square Regression, slip rate surfaces were generated via Bayes Linear Regression, Figure 5.3. BLR gave similar results but can be trained on-line and adjust the surfaces in real time. The separation of the uncertainty surfaces can be increased by tuning the noise variance σ. The first two regression algorithms store all training examples in memory and have a time complexity of $O(N^3)$ for each new example, where N is the number of

training examples, due to matrix inversion. On the other hand BLR does not store the training examples and has a time complexity of $O(d^3)$ for each new example, where d is number of basis functions (only 5 in this case).



(a) Forward Slip

(b) Side Slip



(c) Angular Slip

Figure 5.3: Slip Surfaces learned via Bayes Linear Regression

Chapter 6

Transient Dynamics

So far all the presented techniques are off-line algorithms that assume constant speed and curvature commands during the path segments. On-line techniques that can handle transient dynamics are needed for real world vehicle driving. This chapter presents a vehicle simulation and two on-line filters for learning the parameters of the slip rate surfaces.

Platform performance is not static. It depends on damage, wear, and terrain characteristics. Terrain characteristics depend heavily on weather. It has already been clearly demonstrated that machine learning of terrain models is brittle to changes in weather and seasonal vegetation, so systems will have to adapt to real-time information experienced on the go. Self-calibration is a mandatory component of any realistic deployment of UGVs where speeds, slopes, or terrain difficulty are high.

6.1 Vehicle Simulation

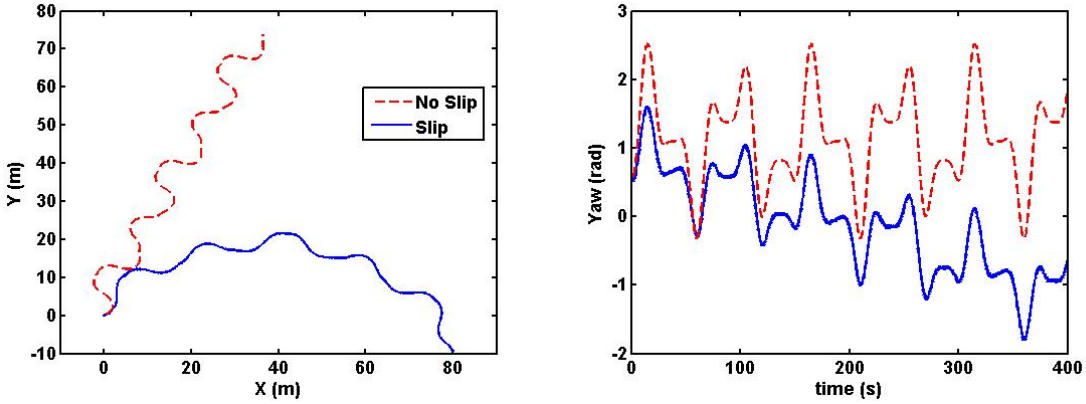
Using the perturbative dynamics model developed in Chapter 3, I developed a simple vehicle simulation to test the observability of the dynamic variables. A vehicle's position and orientation was simulated for a simple path, Figure 6.1. The command speed and curvatures along with the perturbative slip rates are show in Figure 6.2. The slip rate surfaces and variance found via the Least Square Regression where used for the perturbations. An addition small amount of noise was added to the pose states to simulate errors in the RTK GPS and IMU. The fact that the simulation gives realistic results points that the dynamic model and learned slip parameters are plausible.

6.2 Gradient Descent

In Section (3.1) I developed the general form of deferential motion given the perturbative errors, $\underline{\delta u}$.

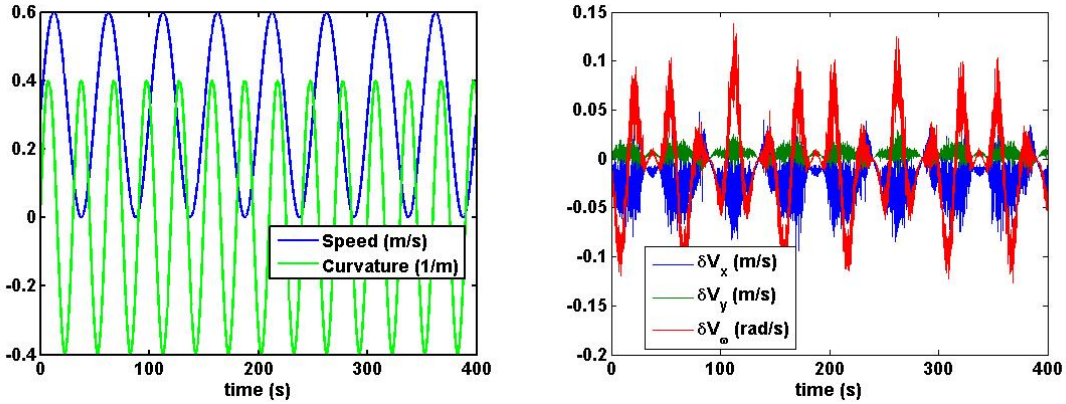
$$\underline{\dot{x}} = \underline{f}(\underline{x}, \underline{u}, \underline{\delta u}) \tag{6.1}$$

Using the slip rate surfaces for continuous functions, the general motion can be parameterized



(a) Position (b) Orientation

Figure 6.1: Simulated Vehicle Path Compared to Ideal Path



(a) Commands (b) Slip Rates

Figure 6.2: Simulated Vehicle Commands and Perturbations

with the learned slip parameters, $\underline{\delta u}(\underline{u}, \underline{p})$.

$$\dot{\underline{x}} = \underline{f}(\underline{x}, \underline{u}, \underline{p}) \tag{6.2}$$

The vehicle path is found by integrating the equations of motion.

$$\underline{X} = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} = F(\underline{u}(t), \underline{p}) = \int f(\underline{x}, \underline{u}(t), \underline{p}) dt \tag{6.3}$$

Using the current slip parameters, the difference in measured pose from predicted pose is cal-

culated.

$$\Delta \underline{X} = \underline{X}_{measured} - \underline{X}_{predicted} \tag{6.4}$$

Given a pose error between the starting pose and ending pose of a path segment, a Jacobian, J, finds the needed change in the perturbative parameters to correct the relative pose residual.

$$\Delta \underline{X} = J \Delta \underline{p} \tag{6.5}$$

$$\Delta \underline{p} = J^{-1} \Delta \underline{X} \tag{6.6}$$

$$J = \frac{\partial \underline{F}}{\partial \underline{p}} = \frac{\partial}{\partial \underline{p}} \int f(\underline{x}, \underline{u}(t), \underline{p}) dt = \int \frac{\partial}{\partial \underline{p}} f(\underline{x}, \underline{u}(t), \underline{p}) dt \tag{6.7}$$

The last step in (6.7) uses Leibniz's rule for differentiation under the integral sign. The chain rule is used for the inner derivative.

$$\frac{\partial \underline{f}}{\partial \underline{p}} = \frac{\partial \underline{f}}{\partial \underline{\delta u}} \frac{\partial \underline{\delta u}}{\partial \underline{p}} \tag{6.8}$$

The first needed derivative, describing the change in vehicle motion model relative to the change in slip rates, is simply the rotation matrix.

$$\frac{\partial \underline{f}}{\partial \underline{\delta u}} = \Gamma = \begin{bmatrix} c\theta & -s\theta & 0 \\ s\theta & c\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{6.9}$$

For this example, I assume the slip rates are the same second-order function learned for the slip rate surfaces.

$$\begin{aligned} \delta V_x &= \alpha_{1,x}\, \kappa + \alpha_{2,x} V + \alpha_{3,x}\, \kappa V + \alpha_{4,x}\, \kappa^2 + \alpha_{5,x} V^2 \\ \delta V_y &= \alpha_{1,y}\, \kappa + \alpha_{2,y} V + \alpha_{3,y}\, \kappa V + \alpha_{4,y}\, \kappa^2 + \alpha_{5,y} V^2 \\ \delta V_\omega &= \alpha_{1,\omega}\, \kappa + \alpha_{2,\omega} V + \alpha_{3,\omega}\, \kappa V + \alpha_{4,\omega}\, \kappa^2 + \alpha_{5,\omega} V^2 \end{aligned} \tag{6.10}$$

The slip rate surface parameters are grouped in a column vector. The second needed derivative describe the the change in slip rates relative to the change in surface parameters.

$$\underline{p} = \left[\alpha_{1,x}\ \ \alpha_{2,x}\ \ \alpha_{3,x}\ \ \alpha_{4,x}\ \ \alpha_{5,x}\ \ \alpha_{1,y}\ \ \alpha_{2,y}\ \ \alpha_{3,y}\ \ \alpha_{4,y}\ \ \alpha_{5,y} \alpha_{1,\omega}\ \ \alpha_{2,\omega}\ \ \alpha_{3,\omega}\ \ \alpha_{4,\omega}\ \ \alpha_{5,\omega}\right]^\top$$

$$\frac{\partial \underline{\delta u}}{\partial \underline{p}} = U = \begin{bmatrix} C & 0_{1,5} & 0_{1,5} \\ 0_{1,5} & C & 0_{1,5} \\ 0_{1,5} & 0_{1,5} & C \end{bmatrix} \tag{6.11}$$

$$C = \begin{bmatrix} \kappa, & V, & \kappa V, & \kappa^2, & V^2 \end{bmatrix}$$

We now have the desired Jacobian by integrating the products of Equations (6.9) and (6.11). The update step follows the gradient to minimize the residual pose error. The learning rate, γ,

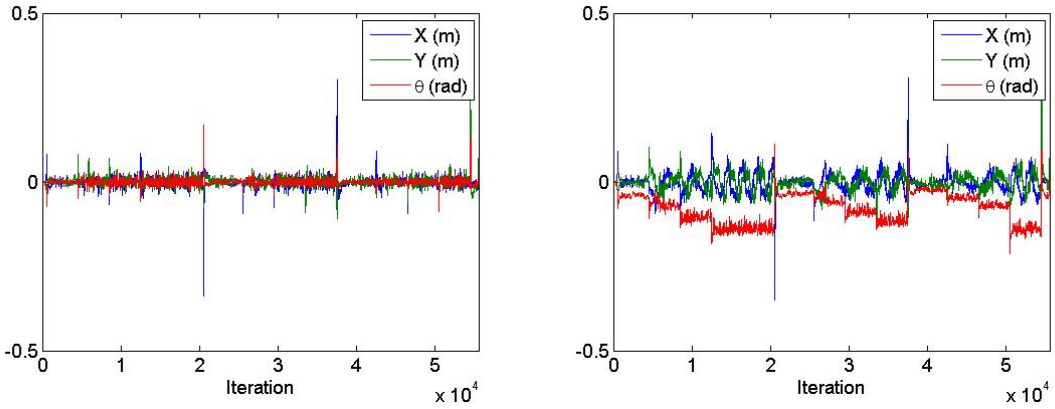
regulates how fast the surface parameters converge.

$$\underline{p}_{t+1} = \underline{p}_t + \gamma \Delta \underline{p} = \underline{p} + \gamma J^{-1} \Delta \underline{X} \tag{6.12}$$

6.2.1 Gradient Descent Results

Data was collected on the Land Tamer vehicle in the same gravel lot as before but after a heavy rain. Mud and wet gravel added to the terrain variability. Data collection occurred as the vehicle was commanded to drive in circles at various curvatures and speeds. The gradient descent algorithm used this data, in temporal order, to optimize the slip surface parameters. The data collection time was just over nine minutes.

The relative pose residuals at each iteration were minimal with a few exceptions, Figure 6.3(a). For comparison, the relative pose residuals with out predicted slip is shown in Figure 6.3(b). Overlapping five second path segments were used for the path residuals. Each iteration occurred with pose measurement updates at 20 Hz. The largest errors came during large instantaneous drops in commanded speed as the model does not currently handle command lag or transients; see Section 8.1 for future work to include vehicle plant dynamics.



(a) Pose Residuals with Gradient Decent

(b) Pose Residuals without predicted Slip

Figure 6.3: Relative Pose Residuals

The slip rate surface parameters were all initialized to zero, making no assumptions about the vehicle-ground interaction allowing the algorithm to converge on the correct answer. The surface parameters variance is show in Figure 6.4 over the algorithm iterations.

6.3 Extended Kalman Filter

Given the success of the simple Gradient Descent method, I investigate the Extended Kalman Filter to correctly incorporate uncertainty in sensor noise and the perturbative parameters. The state transition model is static as the slip parameters are assumed to remain constant while the vehicle

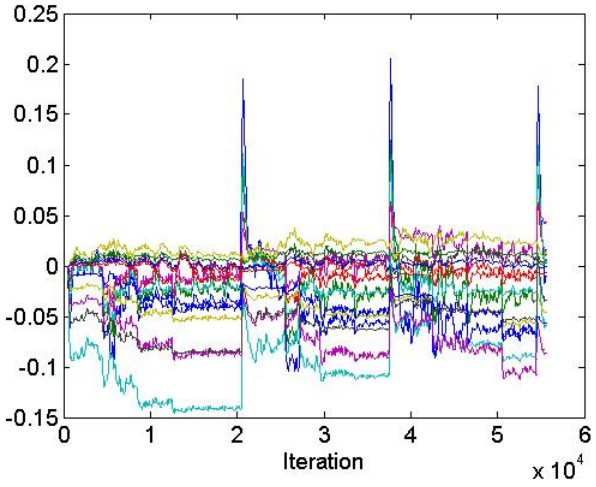


Figure 6.4: Parameter Convergence during Gradient Descent

stays on the same terrain, Eq. (6.13). The process noise, $\underline{\epsilon}_t$, describes the parameter uncertainty and allows the parameters to converge to the correct values; it is assumed to be a zero mean multivariate Gaussian noise with covariance Q_t. The process noise can be increased during terrain transitions and decreased as the terrain remains constant if there exists an external sensor to detect the change.

$$\underline{p}_t = g(\underline{p}_{t-1}) + \underline{\epsilon}_t = \underline{p}_{t-1} + \underline{\epsilon}_t \tag{6.13}$$

$$G_t = \frac{\partial g(\underline{p}_{t-1})}{\partial \underline{p}_t} = I_{5,5} \ \ (IdentityMatrix) \tag{6.14}$$

The observable state is the relative pose between the start and end of the current path segment, $\underline{X}$. The observed state is the measured relative pose; while the observation model is the predicted relative pose simulated according to the sequence of commands, Eq. (6.16). The observation noise is the expected noise of the difference between start and end poses, with covariance R_t.

$$\underline{z}_t = h(\underline{u}(t), \underline{p}_t) + \underline{\delta}_t \tag{6.15}$$

$$\underline{z}_t = \underline{X}_{measured} \tag{6.16}$$

$$h(\underline{u}(t), \underline{p}_t) = \underline{X}_{predicted} = \underline{F}(\underline{u}(t), \underline{p}) \tag{6.17}$$

The observation Jacobian is the same Jacobian used in the Gradient Descent technique, integrating the partial derivative along the path segment, (6.7).

$$H_t = \frac{\partial h(\underline{u}(t), \underline{p}_t)}{\partial \underline{p}_t} = \frac{\partial \underline{F}}{\partial \underline{p}_t} = J \tag{6.18}$$

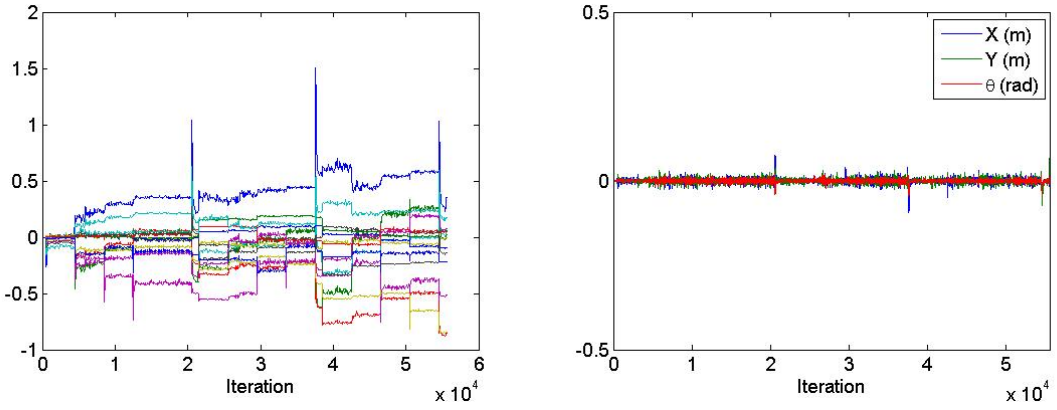
The modified EKF algorithm for the slip rate surface parameters can be seen in Table 6.1.

Algorithm *Slip Parameters Extended Kalman Filter*
Input: $\underline{p}_t, \Sigma_{t-1}, \underline{u}_t, \underline{z}_t$
1. $\bar{\underline{p}}_t = \underline{p}_{t-1}$
2. $\bar{\Sigma}_t = \Sigma_{t-1} + R_t$
3. $K_t = \bar{\Sigma}_t J_t^\top (J_t \bar{\Sigma}_t J_t^\top + Q_t)^{-1}$
4. $\underline{p}_t = \bar{\underline{p}}_t + K_t(\underline{X}_{measured} - \underline{X}_{predicted})$
5. $\Sigma_t = (I - K_t J_t)\bar{\Sigma}_t$
6. **return** $\underline{p}_t, \Sigma_t$

Table 6.1: Slip Parameters Extended Kalman Filter

6.3.1 Extended Kalman Filter Results

The EKF algorithm was run on the same collected data as the Gradient Descent technique. While the parameters converged to different results, Figure 6.5(a), the five second relative pose residuals were extremely minimal, Figure 6.5(b). The convergence to different values points to multiple local minima.



(a) Parameter Convergence

(b) Pose Residuals with EKF

Figure 6.5: EKF Results

The Kalman filter optimizes the parameters uncertainty as well. As can be seen from the final uncertainty matrix, the three slip rate surfaces are designed to be independent of one another, with some correlation between parameters of the each slip rate, Figure 6.6; darker regions correspond to higher correlation. The final learned slip rate surfaces, with uncertainty, are shown in Figure 6.7.

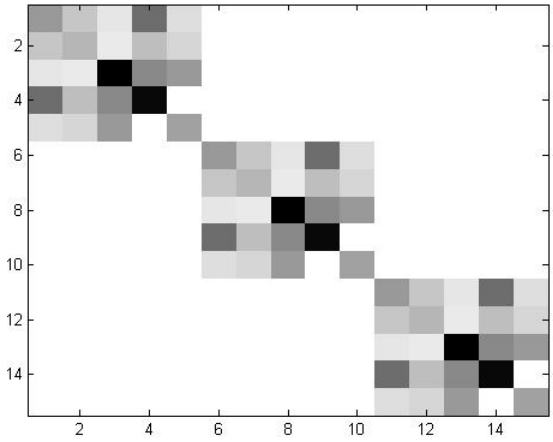


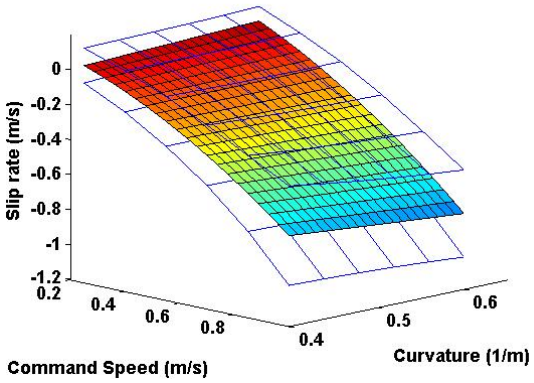
Figure 6.6: Uncertainty Matrix

6.4 Comparison

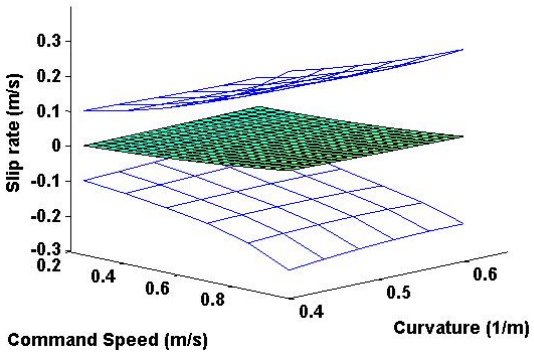
Comparison between the two on-line algorithms' five second relative pose errors, along with the standard deviations, is show in Table 6.2. While the Gradient Descent technique performed well, the EKF had close to half the relative pose error while correctly handling the parameters uncertainties. The EKF five second relative position estimation was four times better than simple prediction without slip. The relative orientation error saw an improvement by a factor of 19.

Table 6.2: Transient Relative Pose Error Comparison

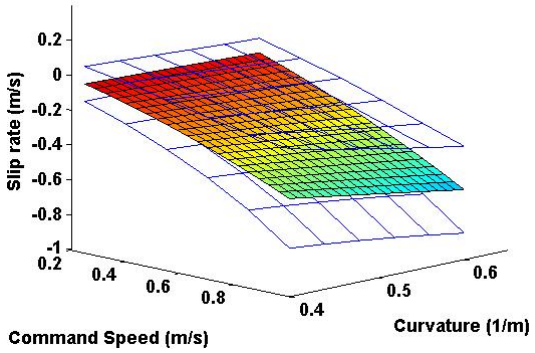
Algorithm	Position (m)	Orientation (rad)
None	$.031 \pm .027$	$.078 \pm .042$
Gradient Descent	$.014 \pm .020$	$.007 \pm .010$
EKF	$.007 \pm .007$	$.004 \pm .004$



(a) Forward Slip



(b) Side Slip



(c) Angular Slip

Figure 6.7: Slip Surfaces learned via EKF

Chapter 7

Conclusions

The work presented in thesis has developed an integrated perturbative dynamics method for real-time identification of wheel-terrain interaction models for enhanced autonomous vehicle mobility. The predicted relative poses were much better when the perturbative dynamics considered slip. The slip rates surface parameters were efficiently learned on-line via the Extended Kalman Filter using the relative pose error residuals from the predicted path instead of the traditional velocity measurements. This was accomplished via an innovative method of integrating the Jacobians across the path segment.

Many enhancements can be made to the algorithm such as including slopes and vehicle plant dynamics described in the next chapter. Perception and terrain classification will allow the system to drive on a mixture of terrains and quickly adjust to new terrains. These improvements will be incorporated into trajectory generation for vehicle prediction.

The EKF can run faster than real time and may soon be integrated into actual vehicles. This should lead to significant improvements of control and prediction of unmanned ground vehicle motion of very rough terrains.

Chapter 8

Future Work

The Vehicle-Ground Model Identification is an ongoing project with many interesting future developments. Future work includes vehicle model improvements, additional data gathering for testing, incorporation of perception and terrain prediction for motion planning.

8.1 Model Improvements

The vehicle model used in this paper used many simplifying assumptions such as flat terrain with uniform terrain properties. There are many future extension possible to the vehicle model to improve accuracy and expand the drivable terrain.

The vehicle model can be extended from three dimensions, (x, y, θ), to a full six dimensional model, $(x, y, z, \psi, \phi, \theta)$. Slip increases as slop increases and can be included in the slip rate surfaces during optimization. Other vehicles have taken slip caused by slopes into consideration while planning its paths and chassis configuration [13].

The largest relative pose errors came during sudden changes in commanded velocity. The vehicle is assumed to instantaneously react to changes in the commanded speed and curvature. In reality there are many delays caused by communication, processing, hydraulics and other physical mechanisms. A few elements of vehicle dynamic models are rate limits, joint limits, and latency.

Accounting for latency in the system is essential for generating correct trajectories when the vehicle state can change dramatically over the scale of the latency. It is hard to distinguish these delays from slip errors. A first order model of the vehicle plant dynamics should include these vehicle plant dynamics and can be learned along with the perturbative parameters.

8.1.1 Pose Extended Kalman Filter

The Slip Rate Extended Kalman Filter (EKF) can be combined with a normal vehicle pose EKF. The learned slip rates will be used to correct the vehicle's motion relative to the ground. This will allow the filter to correctly handle uncertainty in both the slip rates and sensor measurements which will become important on vehicles without high end RTK GPS or IMUs.

Figure 8.1: Available Platforms. Left to right: Autonomous LandTamer, Deere eGator, Mars Rover, DARPA LAGR platform.

An additional lumped external disturbance force can be included to represent a variety of external forces such as wind resistance or the force caused by collision with an obstacle. This will eliminate external forces being treated as wheel slip. Although a direct measure of the disturbance force is generally not available, weak constraints governing its evolution can be developed based upon insight into the physical nature of the disturbance [31]. Weak constraints are a principled method for integrating rules and constraints into the Kalman filter framework and can be viewed as virtual measurements or observations.

8.2 Data Gathering

All of the experiments conducted so far have been on the skid-steered Land Tamer ground vehicle while the vehicle model used was platform independent. Future testing will collect data on a variety of vehicle types including tracked robots, planetary rovers, and high speed Ackerman-steered vehicles. Available vehicle platforms are shown in Figure 8.1. While existing research has focused on small vehicles at low speeds or automotive-scale vehicles on roads, we intend to focus on UGVs and other large vehicles traveling at high speeds over rough terrain. The vehicle model will also be tailored specifically for use by on-line autonomy, which is highly feasible, highly relevant and largely unexplored.

Predicting motion depends on important information such as local terrain shape, soil moisture, soil compaction, and recent weather conditions. The vehicles will be driven on a wide range of terrains under different weather conditions. All of the collected data will be formated into a standardized format for future testing of new algorithms and sharing of data with other researchers who don't have the facilities or access to the large variety of vehicles at CMU.

8.3 Incorporation of Perception and Terrain Prediction

Inhomogeneous terrain is expected to stress an otherwise reasonable homogeneous terrain identification system because it will now have to track a moving target. Increased convergence rates to capture terrain class transients will come at the cost of exposure to local optima and instability.

Perception also provides memory and prediction. While a terrain transition may be an instantaneous event, perception provides a mechanism to associate each such event with all others that looked the same in the system experience. Similarly, perception can cue the system to upcoming terrain changes in order to perform model switching, or to adjust the uncertainty of parameters in the estimation system, or to predispose the parameter search to move in a certain direction.

A set of perception features can be developed, that are directly associated with proprioceptive measurements, to directly predict future model parameters. That is, the vehicle planning system will switch on the asphalt model when perception says the vehicle is about to drive over asphalt. Once on asphalt, the model will be updated continuously to reflect local weather conditions and other variations.

Slip for ground rovers has been predicted using stereo imagery by learning from previous examples of traversing similar terrain with visual odometery [2]. They cast the problem into a Mixture of Experts framework in which the input space is partitioned into subregions, corresponding to different terrain types.

In addition, terrain classification has been based on vibrations induced by wheel-terrain interaction during driving. Vibrations are measured using an accelerometer on the rover structure [5]. Work has been done on merging the results of "low-level" classifiers by fusing classification algorithms from color, texture, range and vibration features [9].

8.4 Motion Planning

As mentioned in Section 2.4, vehicle predictive models, which compensate for wheel slip, are necessary for higher level controls such as obstacle avoidance and regional mobility planning. Because deliberative planners need to evaluate hundreds of paths, it is a hard-requirement that the model be at least 1000 times faster than real-time. This means 1000 seconds of motion must be simulated in 1 second of computation. All state of the art planner models already achieve this speed so it is not at all impossible. On the other hand, very little work has been done on how to adapt such minimal high speed models to high speed motion or challenging terrain.

In addition, the EKF slip rate filter needs to be ported to run real time on a vehicle. This will allow the vehicle to use it for path planning. This phase will perform specific experiments intended to make the case, in the context of a fully operational UGV, that:

- calibrated models significantly outperform uncalibrated ones
- calibration must be performed on-line to adapt to terrain and weather
- real-time calibration is possible and effective for long periods of time

Algorithms have been developed for wheeled mobile robot trajectory generation that achieves a high degree of generality and efficiency through numerical linearization and inversion of forward vehicle models [10]. By predicting the wheel/terrain interaction in the planning stage, rather than accounting for it in the execution stage, more dynamically feasible vehicle motions can be generated.

Bibliography

[1] P. Abbeel, V. Ganapathi, and A. Ng, "Learning vehicular dynamics, with application to modeling helicopters," *Neural Information Processing Systems Foundation*, Whistler, Canada, December 2006.

[2] A. Angelova, L. Matthies, D. Helmick, G. Sibley, and P. Perona, "Learning to Predict Slip for Ground Robots," *Proceedings of the Robotics: Science and Systems Conference*, 2006.

[3] M. Alarfaj, and F. Rogers-Marcovitz, "Interaction of Pose Estimation and Online Dynamic Modeling for a Small Inspector Spacecraft," *ESA Small Satellites Systems and Services 4S Symposium*, Funchal, Portugal, May 2010.

[4] M. Bode, "Learning the Forward Predictive Model for an Off-Road Skid-Steer Vehicle," tech. report CMU-RI-TR-07-32, Robotics Institute, Carnegie Mellon University, September, 2007.

[5] C. Brooks, K. Iagnemma, and S. Dubowsky, "Vibration-based Terrain analysis for Mobile Robots," *International Conference on Robotics and Automation*, Barcelona, Spain, April 2005.

[6] C. Chih-Chung, and L. Chih-Jen, "LIBSVM: a library for support vector machines Software" available at *http://www.csie.ntu.edu.tw/ cjlin/libsvm*, 2001.

[7] O. Jr. Chuy, E. Jr. Collins, and W. Yu, "Power Modeling of a Skid Steered, Wheeled Robotic Ground Vehicle," *IEEE/RAS International Conference on Robotics and Automation*, Kobe, Japan, 2009.

[8] R. S. Gunn, *Support Vector Machines for Classification and Regression*, University of Southampton.

[9] I. Halatci, C. A. Brooks, and K. Iagnemma, "Terrain Classification and Classifier Fusion for Planetary Exploration Rovers," *IEEE Aerospace Conference*, Big Sky, Montana, March 2007.

[10] T. Howard and A. Kelly, "Optimal Rough Terrain Trajectory Generation for Wheeled Mobile Robots," *International Journal of Robotics Research*, Vol. 26, No. 2, February, 2007, pp. 141-166.

[11] T. Howard, C. Green, D. Ferguson, and A. Kelly, "State Space Sampling of Feasible Motions for High-Performance Mobile Robot Navigation in Complex Environments," *Journal of Field Robotics*, Vol. 25, No. 6-7, June, 2008, pp. 325-345.

[12] G. Ishigami, A. Miwa, K. Nagatani, and K. Yoshida, "Terramechanics-Based Model for Steering Maneuver of Planetary Exploration Rovers on Loose Soil," *Journal of Field Robotics*, Vol. 24, No. 3, March, 2007, pp. 233-250.

[13] P. M. Furlong, T. Howard, and D. Wettergreen, "Model Predictive Control for Mobile Robots with Actively Reconfigurable Chassis," *7th International Conferences on Field and Service Robotics*, July, 2009.

[14] A. Kelly, "Fast and Easy Systematic and Stochastic Odometry Calibration," *In Proceedings of International Conference on Intelligent Robots and Systems*, Sendai Japan, September 2004.

[15] A. Kelly, A. Stentz, O. Amidi, M. Bode, D. Bradley, A. Diaz-Calderon, M. Happold, H. Herman, R. Mandelbaum, T. Pilarski, P. Rander, S. Thayer, N. Vallidis, and R. Warner. "Toward Reliable Off Road Autonomous Vehicles Operating in Challenging Environments," *The International Journal of Robotics Research*, Vol 25, No 5/6, 2006.

[16] A. Kelly and T. Howard, "Terrain Aware Inversion of Predictive Models for Planetary Rovers," *Proceedings of the NASA Science and Technology Conference*, June, 2007.

[17] J. Ko, D. Klien, D. Fox, and D. Haehnel, "Gaussian Processes and Reinforcement Learning for Identification and Control of an Autonomous Blimp", *International Conference on Robotics and Automation (ICRA)*, Rome, Italy, April 2007.

[18] J. Ko, D. Klien, D. Fox, and D. Haehnel, "GP-UKF: Unscented Kalman Filters with Gaussian Process Prediction and Observation Models", *International Conference on Intelligent Robots and Systems*, San Diego, CA, October 2007.

[19] K. Kozlowski and D. Pazderski, "Modeling and control of a 4-wheel skid-steering mobile robot", *International Journal of Mathematics and Computer Science*, pp. 477-496, 2004.

[20] E. Lucet, C. Grand, D. Sall, and P. Bidaud, "Dynamic sliding mode control of a four-wheel skid-steering vehicle in presence of sliding", *Romansy*, Tokyo, Japan, July 2008.

[21] E. Lucet, C. Grang, D. Sall and P. Bidaud, "Dynamic velocity and yaw-rate control of the 6WD skid-steering mobile robot RobuROC6 using sliding mode technique", *IEEE/RSJ International Conference on Intelligent Robots and Systems*, St. Louis, Missouri, USA, October 2009.

[22] C.E. Rasmussen and C.K.I. Willimans. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[23] F. Rogers-Marcovitz, "Online Dynamic Modeling and Localization for Small-Spacecraft Proximity Operations," *AIAA/USU Conference on Small Satellites*. Logan, UT, August 2009.

[24] L. Seneviratne, Y. Zweiri, S. Hutangkabodee, Z. Song, X. Song, S. Chhaniyara, S. Al-Milli, and K. Althoefer. "The modeling and estimation of driving forces for unmanned ground vehicles in outdoor terrain," *International Journal of Modeling, Identification, and Control*, Vol. 6, No. 1, 2009

[25] X. Song, L. D. Seneviratne, K. Althoefer, and Z. Song, "A Robust Slip Estimation Method for Skid-Steered Mobile Robots," *Intl. Conf. on Control, Automation, Robotics, and Vision*, Hanoi, Vietnam, Dec. 2008

[26] A. Stentz, C. Dima, C. Wellington, H. Herman, and D. Stager, "A System for Semi-Autonomous Tractor Operations," *Autonomous Robots*, Vol. 13, No. 1, July, 2002, pp. 87-103.

[27] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT Press, Cambridge, MA, 2006.

[28] C. Urmson et al., "Autonomous Driving in Urban Environments: Boss and the Urban Challenge," *Journal of Field Robotics*, Special Issue on the 2007 DARPA Urban Challenge, Part I, Vol. 25, No. 8, June, 2008, pp. 425-466.

[29] M. Pivtoraiko, T. Howard, I. Nesnas, and A. Kelly, "Field Experiments in Rover Navigation via Model-Based Trajectory Generation and Nonholonomic Motion Planning in State Lattices," *Proceedings of the 9th International Symposium on Artificial Intelligence, Robotics, and Automation in Space*, February, 2008.

[30] M. Pivtoraiko, R. A. Knepper, and A. Kelly, "Differentially constrained mobile robot motion planning in state lattices," *Journal of Field Robotics*, Vol. 26, No. 3, March, 2009, pp. 308-333.

[31] C. Ward, and K. Iagnemma, "A Dynamic Model-Based Wheel Slip Detector for Mobile Robots on Outdoor Terrain," *IEEE Transactions on Robotics*, Vol. 24, No. 4, pp. 821-831, August, 2008.

[32] C. Wampler, J. Hollerbach, and T. Arai, "An Implicit Loop Method for Kinematic Calibration and Its Application to Closed-Chain Mechanisms," *IEEE Transactions on Robotics and Automation*, Vol. 11, No. 5, October 1995.

[33] J. Y. Wong, and C. F. Chiang, "A general theory for skid steering of tracked vehicles on firm ground," *Proceedings of the Institution of Mechanical Engineers*, Vol. 215, May 2000.

[34] W. Yu, O. Jr. Chuy, E. Jr. Collins, and P. Hollis, "Dynamic Modeling of a Skid-Steered Wheeled Vehicle with Experimental Verification," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, St. Louis, MO, USA, 2009.

[35] J. Yi, H. Wang, J. Zhang, D. Song, S. Jayasuriya, and J. Liu, "Kinematic Modeling and Analysis of Skid-Steered Mobile Robots With Applications to Low-Cost Inertial-Measurement-Unit-Based Motion Estimation," *IEEE Transactions on Robotics*, Vol. 25, No. 6, Oct. 2009.