UTILIZATION OF ICR KINEMATICS IN ESTIMATION, CONTROL, AND ENERGY-AWARE MISSION PLANNING FOR SKID-STEER VEHICLES

A Dissertation in
Mechanical Engineering
by
Jesse Lorenzo Pentzer

© 2014 Jesse Lorenzo Pentzer

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

December 2014
The dissertation of Jesse Lorenzo Pentzer was reviewed and approved* by the following:

Sean Brennan  
Associate Professor of Mechanical Engineering  
Dissertation Adviser, Co-Chair of Committee

Karl Reichard  
Assistant Professor of Acoustics  
Research Associate, Applied Research Laboratory  
Co-Chair of Committee

H. Joseph Sommer  
Professor of Mechanical Engineering

Jacob Langelaan  
Associate Professor of Aerospace Engineering

Christopher Rahn  
Professor of Mechanical Engineering

Karen Thole  
Professor of Mechanical Engineering  
Head of the Department of Mechanical and Nuclear Engineering

*Signatures are on file in the Graduate School.
Abstract

This dissertation provides details of work completed in the areas of skid-steer robot modeling, trajectory control, power modeling, and energy use prediction. The vast majority of commercially available ground mobile robots utilize skid-steer technology due to the robustness and maneuverability of the design. However, the complex track/terrain interactions developed during skid-steer maneuvers are difficult to model. Most skid-steer robots are currently teleoperated mainly due to the difficulty in implementing full dynamic models of skid-steer motion for autonomy algorithms. This thesis focuses on the development of a kinematic model of skid-steer movement that is adaptable to many terrain types and, when coupled with an appropriate model of skid-steer power usage, enables energy efficient path planning and intelligent feedback to the user of remaining vehicle endurance.
# Table of Contents

List of Figures vii

List of Tables xiv

Acknowledgments xv

Chapter 1
Introduction 1
1.1 Background ................................. 1
1.2 Modeling Skid-Steer Vehicle Motion ......................... 4
1.3 Path Following Control of Skid-Steer Vehicles ..................... 11
1.4 Modeling Power Usage in Skid-Steer Vehicles ..................... 14
1.5 Mission Energy Prediction and Path Planning ..................... 16
1.6 Conclusions ................................. 18

Chapter 2
Description of Test Platforms 21
2.1 Introduction .................................. 21
2.2 Equipment Shared Between Platforms .......................... 21
2.3 Tankbot Description ................................ 23
2.4 RMP Description .................................. 27
2.5 Conclusions .................................... 29

Chapter 3
Estimation and Modeling of Skid-Steer Robot Motion 31
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1 Introduction</td>
<td>31</td>
</tr>
<tr>
<td>3.2 Approximating the Kinematics of Skid-Steer Robots</td>
<td>33</td>
</tr>
<tr>
<td>3.3 EKF for ICR Estimation</td>
<td>37</td>
</tr>
<tr>
<td>3.3.1 Investigation of Conditions for ICR Identification</td>
<td>41</td>
</tr>
<tr>
<td>3.4 Simulation and Analysis of ICR EKF</td>
<td>44</td>
</tr>
<tr>
<td>3.4.1 Preventing Divergence Due to ICR Sign Inversion</td>
<td>47</td>
</tr>
<tr>
<td>3.4.2 Filter Convergence Response to Changing ICR Locations and Vehicle Turn Radius</td>
<td>49</td>
</tr>
<tr>
<td>3.4.3 Steady-State Analysis of ICR EKF</td>
<td>52</td>
</tr>
<tr>
<td>3.4.4 Response of Filter Due to Input Bias and Noise</td>
<td>54</td>
</tr>
<tr>
<td>3.5 Experimental Testing of ICR EKF</td>
<td>58</td>
</tr>
<tr>
<td>3.5.1 Circular Path</td>
<td>60</td>
</tr>
<tr>
<td>3.5.2 Curved Path</td>
<td>64</td>
</tr>
<tr>
<td>3.5.3 Long Distance Traversal</td>
<td>66</td>
</tr>
<tr>
<td>3.5.4 Testing ICR Methods for Robot Localization During GPS Outages</td>
<td>70</td>
</tr>
<tr>
<td>3.5.5 Testing with Conventional GPS Solution</td>
<td>76</td>
</tr>
<tr>
<td>3.5.6 Speed and Terrain Variability Testing</td>
<td>79</td>
</tr>
<tr>
<td>3.6 Alternate Methods of ICR Estimation</td>
<td>84</td>
</tr>
<tr>
<td>3.7 Conclusion</td>
<td>87</td>
</tr>
<tr>
<td>Chapter 4</td>
<td>91</td>
</tr>
<tr>
<td>Trajectory Control of Skid-Steer Vehicles Through the ICR Kinematic Mapping</td>
<td></td>
</tr>
<tr>
<td>4.1 Introduction</td>
<td>91</td>
</tr>
<tr>
<td>4.2 Unicycle Robot Trajectory Control</td>
<td>93</td>
</tr>
<tr>
<td>4.3 Simulation Implementation</td>
<td>100</td>
</tr>
<tr>
<td>4.4 Experimental Results</td>
<td>106</td>
</tr>
<tr>
<td>4.5 Conclusions</td>
<td>108</td>
</tr>
<tr>
<td>Chapter 5</td>
<td>110</td>
</tr>
<tr>
<td>Modeling the Power Usage of Skid-Steer Vehicles</td>
<td></td>
</tr>
<tr>
<td>5.1 Introduction</td>
<td>110</td>
</tr>
<tr>
<td>5.2 Modeling Skid-Steer Vehicle Power Usage</td>
<td>112</td>
</tr>
<tr>
<td>5.2.1 Surface Change Detection</td>
<td>116</td>
</tr>
<tr>
<td>5.3 Results of Power Model Implementation</td>
<td>118</td>
</tr>
<tr>
<td>5.3.1 Tracked Vehicle Power Model Results</td>
<td>118</td>
</tr>
<tr>
<td>5.3.2 Wheeled Vehicle Power Model Results</td>
<td>124</td>
</tr>
<tr>
<td>5.4 Conclusions</td>
<td>128</td>
</tr>
</tbody>
</table>
List of Figures

1.1 Comparison of measured sprocket torques, sprocket torques predicted by Wong’s general skid-steer theory, and sprocket torques predicted by Steeds’ theory of skid-steer motion [64]. 6

2.1 Skid-steer robots used during experimental testing. 22
2.2 MTi-300 AHRS from Xsens (www.xsens.com). 23
2.3 Flowchart of Tankbot manual control and drive systems. 24
2.4 Labeled picture of Tankbot drive system. 25
2.5 Dimensions of Tankbot bogie wheels relative to frame geometric center. Dimensions in meters and Tankbot is symmetrical about the X and Y axes. 26
2.6 Tankbot main computer and voltage regulation system. 27
2.7 Dimensions of RMP400 wheel centers relative to the geometric center of the vehicle frame. Dimensions are in meters, and the RMP 400 is symmetric about the X and Y axes. 28
2.8 Segway main computer. 29

3.1 Plot of measured and estimated slip with skid-steer robot from Burke. 32
3.2 Diagram of ICR location for rigid body in planar motion. 34
3.3 Diagram of ICR locations on a skid-steer robot. 34
3.4 Relation between body frame and fixed frame through heading angle $\psi$. 37
3.5 Simulated position and EKF estimate for tracked robot. 47
3.6 EKF estimated ICR positions during simulation for tracked robot. 47
3.7 Plot of $y_{ICRr}$ estimate during tracked vehicle simulations for initial conditions ranging from 2 through -2 using kinematic equations (3.9) - (3.11). 48
3.8 Plot of $y_{ICRr}$ estimate during tracked vehicle simulations for initial conditions ranging from 2 through -2 with modified kinematic equations (3.33) - (3.35). 49
3.9 Simulation run with changing ICR locations. Times of ICR changes are labeled with vertical lines at 175 seconds and 350 seconds.

3.10 Simulation run with changing ICR locations and inflation of ICR location estimate error covariance at ICR change. Times of ICR changes are labeled with vertical lines at 175 seconds and 350 seconds.

3.11 Average convergence time of ICR estimates from EKF as a function of increasing turn radius over twenty simulation runs. There is a clear increase in convergence time as the turn radius increases.

3.12 Predicted steady-state error standard deviation for ICR states at varying vehicle turn rates.

3.13 Comparison of commanded and measured left track velocity for tracked vehicle.

3.14 Error in ICR location estimate as a function of increasing input noise on simulated tracked robot.

3.15 Error in ICR location estimate as a function of increasing input bias on simulated tracked robot.

3.16 Plot of GPS position measurement, EKF position estimate, ICR kinematic odometry, and no-slip kinematic odometry for circular run with tracked robot.

3.17 Plot of ICR estimates during circular run with tracked robot. The run was 114 seconds long, but the plot has been scaled to show details of the initial estimation of ICR locations. Convergence times (10% to 90%) for each ICR location estimate are also shown on the graph.

3.18 Plot of GPS position measurement, EKF position estimate, ICR kinematic odometry, and no-slip kinematic odometry for circular run with wheeled robot.

3.19 Plot of ICR estimates during circular run with wheeled robot. The run was 145 seconds long, but the plot has been scaled to show details of the initial estimation of ICR locations. Convergence times (10% to 90%) for each ICR location estimate are also shown on the graph.

3.20 Comparison of ICR estimates when the EKF is initialized with estimated states from previous operation and when the EKF is initialized with poor guesses of ICR locations.

3.21 Plot of GPS position measurement, EKF position estimate, ICR kinematic odometry, and no-slip kinematic odometry for driving on curved path with tracked robot.

3.22 Plot of ICR estimates during driving on curved path with tracked robot.
3.23 Plot of GPS position measurement, EKF position estimate, ICR kinematic odometry, and no-slip kinematic odometry for driving on curved path with wheeled robot. .............................. 67
3.24 Plot of ICR estimates during driving on curved path with wheeled robot. ........................................ 67
3.25 Plot of GPS measured position and EKF position estimate during lawnmower search pattern with tracked robot. ................................. 68
3.26 Clustering of ICR location estimates during long distance run with tracked robot. ................................. 68
3.27 Plot of GPS measured position and EKF position estimate during lawnmower search pattern with wheeled robot. ................................. 68
3.28 Clustering of ICR location estimates during long distance run with wheeled robot. ................................. 69
3.29 Test of ICR odometry as backup system during GPS dropouts with tracked robot. ................................ 71
3.30 Histogram of open-loop ICR odometry error on tracked robot after 5 m of travel from each point traversed in lawnmower search pattern. 72
3.31 Histogram of open-loop ICR odometry error on tracked robot after 10 m of travel from each point traversed in lawnmower search pattern. 73
3.32 Histogram of open-loop ICR odometry error on tracked robot after 20 m of travel from each point traversed in lawnmower search pattern. 73
3.33 Summary of open-loop ICR odometry error on tracked robot for 1-20 m of travel from each point traversed in lawnmower search pattern. ................................. 74
3.34 Summary of open-loop ICR odometry error on wheeled robot for 1-20 m of travel from each point traversed in lawnmower search pattern. ................................. 74
3.35 Two examples of ICR and no-slip odometry performance over 15 m distances with tracked robot. ...................... 76
3.36 Two examples of ICR and no-slip odometry performance over 15 m distances with wheeled robot. ...................... 76
3.37 Plot of measured GPS position and EKF position estimate when operating without RTK GPS corrections. ................. 77
3.38 Clustering of ICR location estimates during run without RTK GPS corrections. ................................. 78
3.39 Summary of open-loop ICR odometry error on tracked robot for 1-20 m of travel from each point traversed in lawnmower search pattern. ICR locations for odometry were estimated by ICR EKF without RTK corrections to GPS solution. 79
3.40 Plot of GPS position measurement, EKF position estimate, ICR
kinematic odometry, and no-slip kinematic odometry when driving
the wheeled robot on asphalt at 0.4 m/s. ............................. 80
3.41 Clustering of ICR location estimates when driving the wheeled
robot on asphalt at 0.4 m/s. ........................................ 80
3.42 Plot of GPS position measurement, EKF position estimate, ICR
kinematic odometry, and no-slip kinematic odometry when driving
the wheeled robot on asphalt at 1.6 m/s. ............................. 81
3.43 Clustering of ICR location estimates when driving the wheeled
robot on asphalt at 1.6 m/s. ........................................ 82
3.44 Plot of GPS position measurement, EKF position estimate, ICR
kinematic odometry, and no-slip kinematic odometry when driving
the wheeled robot on asphalt at 4.0 m/s. ............................. 82
3.45 Clustering of ICR location estimates when driving the wheeled
robot on asphalt at 4.0 m/s. ........................................ 83
3.46 Comparison of difference in ICR EKF position estimate and mea-
surement from RTK GPS system. ................................. 84

4.1 Two wheel robot following planar path $C$. .................. 93
4.2 Example of point $P$ with zero lateral velocity at center of left and
right ICR locations. .............................................. 99
4.3 Augmentation of path points to create path $C'$. ............ 99
4.4 Comparison of desired path and trajectory of simulated vehicle with
forward speed 0.5 m/s. Desired path is a circle of radius 50 m
centered at the origin. .............................................. 101
4.5 Heading error (left vertical axis) and distance error (right vertical
axis) for circular path simulation run. .......................... 102
4.6 Heading error (left vertical axis) and distance error (right vertical
axis) for circular path simulation run with 0.4 second delay between
controller angular rate output and implementation. .............. 103
4.7 Desired path for simulated trajectory control overlaid on image of
testing area. .................................................... 104
4.8 Comparison of path and simulated vehicle traversal with forward
speed of 0.4 m/s. Path point locations down sampled by a factor
of five to better show vehicle trajectory. ......................... 104
4.9 Simulated distance error with forward speed of 0.4 m/s. ....... 105
4.10 Right and left track speed commands produced by controller during
simulation. ................................................... 105
4.11 Distance error and heading error during experimental run with
tracked robot. .............................................. 107
4.12 Distance error and heading error during experimental run with wheeled robot.

5.1 Diagram of right side track on skid-steer vehicle showing differential element of track contact area relative to ICR location.

5.2 Assumed locations of normal forces acting on wheeled and tracked vehicles. Frictional forces are calculated at normal force locations.

5.3 Example of EWMA surface change detection on tracked robot during run with three surface changes.

5.4 Position of vehicle during power estimation testing. Asphalt and grass surfaces were traversed, and the transition from asphalt to grass occurred 380 seconds after mission start. Test performed with tracked robot.

5.5 Power model parameters $\mu$ (a), $G$ (b), and $\beta$ (c) learned during asphalt and grass traversal. Test performed with tracked robot. Surface transition from asphalt to grass occurred 380 seconds after mission start.

5.6 (a) The measured and modeled power usage during the run. (b) The error in power estimation. Test performed with tracked robot. Surface transition from asphalt to grass occurred 380 seconds after mission start.

5.7 Path of tracked vehicle while navigating a complex grass and asphalt path.

5.8 Values of power model coefficient $\mu$ estimated during traversal of path shown in Figure 5.7.

5.9 Values of power model coefficient $G$ estimated during traversal of path shown in Figure 5.7.

5.10 Values of power model coefficient $\beta$ estimated during traversal of path shown in Figure 5.7.

5.11 Power model coefficients identified on the wheeled vehicle while traversing the grass and asphalt path shown in Figure 5.4.

5.12 Comparison of measured and modeled power usage on wheeled robot while traversing path shown in Figure 5.4.

5.13 Values of power model coefficient $\mu$ estimated during traversal of path shown in Figure 5.7.

5.14 Values of power model coefficient $G$ estimated during traversal of path shown in Figure 5.7.

5.15 Values of power model coefficient $\beta$ estimated during traversal of path shown in Figure 5.7.
6.1 Schematic of forward path energy estimation. Filters utilize information from the traversed path to form estimates of ICR location and power model coefficients. The terrain pitch is recorded with the map points for power usage estimation.

6.2 Schematic of reverse path energy estimation. Filters utilize information from the traversed path to form estimates of ICR location and power model coefficients. The path already traversed is reversed and a simulated vehicle is used to estimate energy usage. The energy usage is estimated incrementally and only the portion of the path between the current vehicle position and the previous retro-traversal energy estimate is simulated.

6.3 Comparison of the measured and estimated energy required to complete the path shown in Figure 5.4 with the tracked vehicle.

6.4 Comparison of the measured and estimated energy required to complete the path shown in Figure 5.7 in reverse with the tracked vehicle.

6.5 Comparison of the measured and estimated energy required to complete the path shown in Figure 5.4 in reverse with the wheeled vehicle.

6.6 Comparison of the measured and estimated energy required to complete the path shown in Figure 5.7 in reverse with the wheeled vehicle.

6.7 Color satellite photograph of robot testing area.

6.8 Power map image after k-means segmentation. Each of the two gray colors represent a set of power model parameters.

6.9 Power map image after k-means segmentation and filtering with 9x9 median filter. Each of the two gray colors represent a set of power model parameters.

6.10 Comparison of the measured and estimated energy required to complete the path shown in Figure 5.4 with the tracked vehicle. A power model coefficient map created during a previous traversal is used for prediction of energy usage.

6.11 Comparison of the measured and estimated energy required to complete the path shown in Figure 5.7 with the tracked vehicle. A power model coefficient map created during a previous traversal is used for prediction of energy usage.

6.12 Comparison of the measured and estimated energy required to return to the starting point during a traversal of the path shown in Figure 5.7 with the tracked vehicle.

6.13 Comparison of the measured and estimated energy required to return to the starting point during a traversal of the path shown in Figure 5.7 with the wheeled vehicle.
7.1 Example of A-star path planning algorithm operating on area with obstacles shown as red circles. ................. 150
7.2 Example of RRT path planning algorithm operating on open areas. 151
7.3 Diagram of SBMPO planning steps to compute a path from node A to the goal region. ......................... 153
7.4 Example of two node state possibilities with pseudo-grid dimensions. 155
7.5 Comparison of selecting inputs using (a) Halton points and (b) a uniform random number generator. ........... 158
7.6 Calculated trajectories to expand each node during SBMPO path planning. ...................................... 158
7.7 Path planning solution on artificial map using energy cost function with two similar surfaces. Power model parameters for each surface are labeled in the figure. ................................. 162
7.8 Path planning solution on artificial map using distance cost function with two similar surfaces. Power model parameters for each surface are labeled in the figure. ..................... 163
7.9 Path planning solution on artificial map using energy cost function with two moderately different surfaces. Power model parameters for each surface are labeled in the figure. ............ 163
7.10 Path planning solution on artificial map using distance cost function with two moderately different surfaces. Power model parameters for each surface are labeled in the figure. ....... 164
7.11 Path planning solution on artificial map using energy cost function with two significantly different surfaces. Power model parameters for each surface are labeled in the figure. .......... 164
7.12 Path planning solution on artificial map using distance cost function with two significantly different surfaces. Power model parameters for each surface are labeled in the figure. ......... 165
7.13 Path planning solution for gentle corner using energy heuristic. .... 167
7.14 Path planning solution for gentle corner using distance heuristic. .. 167
7.15 Path planning solution for sharp corner using energy heuristic. ... 168
7.16 Path planning for long straightaway using energy heuristic. Planned route operates on lower energy-use surface. .......... 169
7.17 Path planning for long straightaway using distance heuristic. Planned route traverses higher energy-use surface straight from start to goal. 170
# List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Field test parameters for tracked vehicle.</td>
<td>60</td>
</tr>
<tr>
<td>3.2</td>
<td>Field test parameters for wheeled vehicle.</td>
<td>60</td>
</tr>
<tr>
<td>3.3</td>
<td>Results of using ICR EKF algorithm at increasing speeds.</td>
<td>81</td>
</tr>
<tr>
<td>4.1</td>
<td>Trajectory control simulation parameters.</td>
<td>100</td>
</tr>
<tr>
<td>5.1</td>
<td>Identified power model parameters for repeated runs with tracked vehicle.</td>
<td>121</td>
</tr>
<tr>
<td>5.2</td>
<td>Identified power model parameters for repeated runs with wheeled vehicle.</td>
<td>125</td>
</tr>
<tr>
<td>6.1</td>
<td>Results of forward energy estimation for nine test runs with the tracked vehicle.</td>
<td>137</td>
</tr>
<tr>
<td>6.2</td>
<td>Results of forward energy estimation for eleven test runs with the wheeled vehicle.</td>
<td>139</td>
</tr>
<tr>
<td>6.3</td>
<td>Power model parameters generated for map given in Figure 6.9.</td>
<td>143</td>
</tr>
<tr>
<td>6.4</td>
<td>Results of retro-traversal energy estimation for eight test runs with the tracked vehicle.</td>
<td>146</td>
</tr>
<tr>
<td>6.5</td>
<td>Results of retro-traversal energy estimation for eight test runs with the wheeled vehicle.</td>
<td>146</td>
</tr>
<tr>
<td>7.1</td>
<td>Simulation parameters for SBMPO path planning with artificial power map.</td>
<td>161</td>
</tr>
<tr>
<td>7.2</td>
<td>Simulation parameters for SBMPO path planning with power map created from test data.</td>
<td>166</td>
</tr>
</tbody>
</table>
Acknowledgments

I would first like to thank my advisors, Dr. Sean Brennan and Dr. Karl Reichard, for their support, insight, and thoughtful prodding. Without them, none of this would have been possible.

I thank my fellow lab members, those currently here and those long gone. Research is truly a team effort, and you have helped me every step of the way.

To my family, parents Ed and Selma, and brothers Jason, Jon, Jed, and Jordan, thank you for all of your support. The laughs, love, and memories make life worth living. You are all amazing, and I owe so much of my success to you.

Finally, to my fiancée Katie, I dedicate this work to you. As this chapter of my life ends, I look forward to beginning our new adventures together.

This work was supported by the Exploratory and Foundational Research Program, administered by the Applied Research Laboratory, The Pennsylvania State University. No official endorsement should be inferred.
A capacity, and taste, for reading, gives access to whatever has already been discovered by others. It is the key, or one of the keys, to the already solved problems. And not only so. It gives a relish, and facility, for successfully pursuing the yet unsolved ones.

—Abraham Lincoln, 1859
Chapter 1

Introduction

1.1 Background

Once relegated to factory floors and research institutions due to immense size and cost, robotic technologies in the 21st century are rapidly improving. Robots are truly a multidisciplinary product, and as such, advancements in computing, actuating, sensing, and energy storage technologies are pushing robotic research forward at a fantastic pace. In particular, the area of mobile ground robots has seen an incredible increase in both research and production since the early 1990s. Improvements in battery technologies allow unmanned ground vehicles to operate for long durations, and wireless communication technologies give the vehicle operator the ability to run the system from long distances away. One commonality among the vast majority of successfully fielded ground mobile robots, and the focus of this research, is the use of a skid-steer locomotion mechanism.
Skid steering is a commonly utilized locomotion mechanism for mobile robots and large industrial and agricultural vehicles. A skid-steer system has no steering mechanism and instead changes vehicle direction by adjusting the speed of the left and right side wheels or tracks. The simplicity, robustness, and zero-radius turn capability of skid steering make it an excellent choice for all-terrain and teleoperated vehicles, but the complex track/terrain interaction developed during a skid-steer maneuver is difficult to model. Hence, when compared to steered or two-wheel robots, it is quite difficult to predict the future position of a skid-steer robot.

The number of unmanned ground vehicles utilized by civilian and military organizations has increased significantly in the past decade, the vast majority of which utilize skid-steer mobility. Technological improvements to ground robots have made them especially effective in countering the threat of improvised explosive devices (IEDs) [62], which are expected to continue to be a global threat in the foreseeable future [20, 36]. Most unmanned ground systems currently fielded are teleoperated and have limited or no autonomous functions. A recent report by the Robotic Systems Joint Project Office states one of the largest obstacles to increased autonomy is improved navigation of unmanned ground vehicles for the safety of soldiers and civilians in the operational area [45]. Improvements in sensors and algorithms, such as inertial measurement units (IMUs), the global positioning system (GPS), simultaneous localization and mapping (SLAM), and
visual odometry, to name a few, are key to improving real-time estimates of ground robot positioning and attitude. However, in the absence of environmental information, which may occur when GPS satellites are occluded or the camera view is blocked, the motion estimation of a skid-steer vehicle requires an accurate model. While dynamic models of skid-steer motion have been developed [63], extensive knowledge of vehicle and terrain properties are necessary, and the models are so complex that real-time implementation remains difficult. A major component of this thesis is the development and experimental testing of kinematic models for skid-steer motion. Terrain-related parameters within the model are learned during vehicle operation, which allows the robot to adapt to changing terrains during a mission.

In addition to motion estimation, the power usage of a skid-steer vehicle is difficult to model due to track/terrain and wheel/terrain interactions. During a single deployment, it would not be uncommon for an explosive ordnance disposal (EOD) robot to encounter multiple terrains, such as grass, asphalt, concrete, sand, etc., as well as terrain obstacles such as hills, stairs, and ditches. Currently fielded ground robots provide the user with limited feedback of remaining energy, most providing only a measurement of battery voltage, and this feedback has no knowledge of terrain conditions in which the robot is operating or may operate in the future. Two areas of important research affect skid-steer endurance estimation significantly: improving battery models to determine the remaining available energy
and developing accurate estimates of the energy required for a robot to complete its mission. The two areas are coupled in that remaining energy may in fact be a function of the expected discharge rate for the battery, and the choice of future missions (and hence the discharge rate) may strongly depend on the current estimate of remaining battery energy. This thesis focuses on the second area, modeling skid-steer energy usage, and presents a model of skid-steer power usage for locomotion with the ability to adapt to changing surfaces during operation. This power model is combined with kinematic models of vehicle motion to estimate the energy required for a vehicle to complete a path traversal.

The main contributions of this research can be divided into four highly coupled areas: the estimation of vehicle and terrain parameters necessary for motion modeling, the utilization of skid-steer motion models for trajectory control, the development of power-use models for skid-steer vehicles, and the on-line estimation of skid-steer robot energy usage. This final area can be extended in many ways, and in this work, the estimation of energy usage is utilized to perform energy-aware path planning for skid-steer vehicles. The following sections in this chapter provide a review of the literature in these areas.

1.2 Modeling Skid-Steer Vehicle Motion

Improving navigation in unmanned vehicles relies heavily on understanding the complex track/terrain interactions of skid-steer vehicles due to significant slippage
during movement. The research presented by Anousaki showed that a differential drive, two-wheel vehicle model cannot be used to accurately represent a skid-steer vehicle due to track and wheel slippage [2]. Much work has been completed in the terramechanics and vehicle dynamics fields developing complete models of skid-steer vehicle motion. Steeds first provided a detailed analysis of skid-steer motion in a series of articles published in 1950 [55–57]. Steeds’ analysis focused on the theory of shear stress between the track and ground obeying Coulomb’s law of friction. The resulting theory was that the maximum shear-stress, which is related to tractive force, occurs after a small relative movement between the track and ground. This work formed the foundation for many following studies along the same theory by Kitano and Kuma, Crosheck, and others [10,21]. In the 1950’s and 1960’s, Bekker provided many contributions to tracked vehicle mechanics in the areas of track pressure distributions and sinkage along with tractive effort from a shear stress-slippage standpoint [4]. Much of the work on tracked vehicle mechanics is well summarized by Wong in [63]. In particular, the dynamics of tracked vehicles on firm ground is treated especially well by Wong in [64]. Wong built upon Steeds’ work by using an exponential relationship between shear displacement and shear stress on the track-ground interface rather than a Coulomb friction model. As shown in Figure 1.1, reprinted from [64], the resulting prediction of movement and drive sprocket torques for steady-state turning maneuvers match experimental data much better than the models developed by Steeds [15]. Steeds’ model predicts
relatively constant torques regardless of turning radius with the inner and outer sprocket torques always being in opposite directions. Measurements show that smaller turning radii require much larger sprocket torques and that larger radii require torques in the same direction from each sprocket. These torque curves are often used to validate dynamic models of skid-steer motion in more recent works.

Thus, one might say that the field of modeling skid-steer vehicles is fairly mature, especially on firm ground where sinkage and the bulldozing effect (piling of soil along tracks during movement) can be neglected. However, a review of this research shows that it focused nearly exclusively on large military and agricultural

![Figure 1.1. Comparison of measured sprocket torques, sprocket torques predicted by Wong’s general skid-steer theory, and sprocket torques predicted by Steeds’ theory of skid-steer motion [64].](image)
type vehicles, especially from an experimental standpoint. Ground mobile robots in use today are much smaller and often operate on terrains such as asphalt and concrete that are poorly addressed by these models. With the research impetus now moving towards automation, or at the very least, computer assisted control of these vehicles, it is desirable to have a model simple enough to run on the vehicle during operation. The previously presented models are complex and require extensive knowledge of vehicle and terrain properties, making implementation on a wide variety of platforms and terrains difficult. For example, the general theory of skid-steer motion presented by Wong requires that the shear deformation modulus, coefficient of friction between track and ground, and the coefficient of external motion resistance all be known, along with complete knowledge of the vehicle design and mass distribution.

The past decade has seen increased interest in modeling focused on mobile robots. In 1999, Caracciolo, De Luca, and Iannitti presented a simplified dynamic model and corresponding trajectory controller for a wheeled skid-steer vehicle [6]. Assumptions of low vehicle speed, zero longitudinal wheel slippage, and a Coulomb friction model for wheel-ground contact forces were used in the development of the model. Simulation results show good performance, but no implementation on the target vehicle was presented so it is unclear how well the simplified model matches the dynamics of a wheeled skid-steer robot where longitudinal wheel slip may be significant, especially if the controller design was extended to tracked skid-steer
vehicles. Kozlowski and Pazderski built upon the work of Caracciolo et al. and presented a new controller design more robust to uncertainties in parameters within the dynamic model and presented extensive simulation results, but the assumption of zero longitudinal wheel slip was still utilized [22]. Additional research in skid-steer dynamics address ground contact forces from the perspective of shear displacement, as done by Wong, or using tire models common in vehicle dynamics research [7,18,61,66,68,69]. The work of W. Yu et al. is especially well done, and simulation results are confirmed through extensive experimental testing [68]. W. Yu used the same function relating shear stress and shear displacement as Wong, but developed a dynamic model for skid-steer wheeled robots instead of tracked vehicles. The experimental testing shows results very similar to Figure 1.1. Once again, extensive knowledge of terrain parameters, some involving experimental identification, are necessary to implement the model. Approaching the problem from the tire modeling perspective, Z. Yu et al. presented work modeling skid-steer motion for a wheeled robot using tire model properties such as tire longitudinal slip stiffness and cornering stiffness. While dynamic models of skid-steer motion provide the most accurate estimates of vehicle motion, implementing them on actual vehicles remains a challenge due to their complexity and the large number of surface and vehicle parameters required. Additionally, each research effort previously discussed is specific to either wheeled or tracked skid-steer vehicles. As the research proposed in this document focuses on kinematic modeling of skid steer
motion, the discussion hereafter will focus on the most closely related research. However, it must be noted that some dynamic model approaches do incorporate slip estimation to improve motion modeling. The downside when compared to the kinematic methods presented next is the reliance on known terrain properties.

The second branch of research has focused on utilizing kinematic models of vehicle movement to estimate slip in some way and therefore improve vehicle motion estimates. A purely experimental approach is proposed by Anousaki in [2] by recording the wheel speed inputs and the resulting planar motion of a skid-steer vehicle and then determining a motion model through linear regression. This model is combined with a measurement of angular velocity in a Kalman filter. Each model is specific to a robot and terrain combination with the benefit being that no prior knowledge of vehicle configuration is needed. In 2005, Martínez presented a kinematic motion model that utilizes estimates of track instantaneous centers of rotation (ICRs) for motion estimation [28]. The locations of the ICRs for a test robot are found in post processing by comparing track velocity inputs with measured vehicle motion. Implementation results show that impressive open-loop odometry estimates can be achieved once the ICR parameters are identified. Yi presented an extended Kalman filter (EKF) that integrates measurements from an inertial measurement unit (IMU) and fuses them with virtual velocity measurements created using wheel encoder measurements and estimates of slip generated from a kinematic model of motion [67]. Experimental results show position es-
timate errors below 25 cm for approximately 40 m of travel on a complex path. An EKF is also used by Dar for estimating slip [11]. The algorithm performs well on recorded data, but was not implemented on the vehicle. More recently, Burke presented a least squares method of estimating slip implemented in conjunction with a kinematic model [5]. The estimator is guaranteed to converge, but simulation results do not generally show convergence when slip values are constantly changing. No implementation is presented.

It is interesting to note vision-based algorithms have been utilized to detect slip by comparing inputs with measured motion using stereo or monocular vision odometry algorithms [1, 30, 35, 53, 54]. Historically such algorithms have been limited to mobile robots traveling very slowly, such as the Mars rover, due to the large processing load associated with optical flow algorithms. Most of the presented methods rely on features in subsequent images being repeated, so the algorithm update rate must be sufficient for this to occur when compared to vehicle velocity. More recently, Song presented work showing improvements in computation power and reductions in size have enabled optical velocity estimators to run accurately for vehicle speeds approaching 2 m/s [53].

This section has presented an overview of research focused on modeling the movement of skid-steer vehicles. Dynamic models of skid-steer motion offer excellent accuracy, but are complex and require knowledge of surface parameters which may not be available. In this thesis, an extended Kalman filter is used to estimate
the ICR locations of the tracks or wheels during operation. In this way, the parameters relating the inputs, track or wheel speeds, to vehicle motion are learned by driving the vehicle. Also, the ICR kinematic model applies to both tracked and wheeled skid-steer vehicles and requires no prior knowledge of vehicle configuration, making it very adaptable on a range of robot platforms. Finally, knowledge of ICR locations will be utilized to both control a skid-steer vehicle along a path and estimate the power usage for skid-steer motion.

1.3 Path Following Control of Skid-Steer Vehicles

The vast majority of commercially available skid-steer vehicles remain teleoperated. This is because the types of tasks ground robots most often perform, such as explosive ordinance disposal and inspection of hazardous areas, most often require human interpretation of sensor feedback and real-time decision making and intervention. There are many portions of the task; however, that could be automated, such as traversal to a target and especially retro-traversal along a predefined path to the point of retrieval. One issue preventing automation of commercial robots is the lack of an implementable motion model accurate over varying terrains, discussed in the previous section. There has been a significant amount of research in path following control for mobile robots in the past decade, but a large portion
of research focuses on control of what are termed wheeled mobile robots (WMRs) in the literature [24, 37, 50, 65]. While some refer to wheeled skid-steer robots as WMRs, the majority of published papers are referring to two-wheel robots, omni-wheel robots, and also ball wheel robots among others that do not use skid-steer locomotion nor are ever intended for realistic off-road use. The motion of these vehicles can be described with kinematic models with no dependence on terrain because the design of the locomotion system limits the amount of slip that occurs during motion. These types of vehicles exhibit impressive mobility, often able to move in any direction, but are generally unsuitable for rough terrain or obstacle traversal. Trajectory control strategies for WMRs have included sliding mode control by Yang [65], receding horizon predictive control by Seyr and Khune [24, 50], and feedback linearization by Oriolo [37]. While skid-steer vehicles exhibit motion similar to WMRs in ideal cases, when large amounts of slippage occur the motion is very different and it is unclear if control strategies will exhibit performance on skid-steer vehicles similar to that of WMRs.

There is available literature reporting control algorithms specifically designed for skid-steer vehicles. Caracciolo, De Luca, and Iannitti presented an input-output linearizing feedback control for wheeled vehicles robust to errors in vehicle and terrain parameters [6]. Simulation results show good tracking performance, but it is assumed that the rolling resistance and coefficient of friction on the terrain are known, and no experimental results are reported. Kozlowski and Pazderski
extended the work of Caracciolo et al. and presented a control using a constraint to limit lateral slippage and a kinematic oscillator to force convergence to a reference vehicle path [22]. Further work presents a controller using a similar kinematic oscillator but without the lateral slippage constraint [23]. Both works address the control of wheeled skid-steer vehicles and present simulation results but no implementation. Experimental results for a wheeled vehicle are presented for a similar controller in [38], but in this instance the vehicle kinematics are assumed to be that of a unicycle type robot with a small perturbation term to account for lateral skidding. In 2007, a tracked skid-steer vehicle controller is presented by Endo which proportionally adjusts the commanded vehicle angular rate based on the error in heading and position versus a defined path [16]. The controller is coupled with a motion model that includes slip, but key constants in the model are learned through post-analysis of experimental data. Similar control strategies are presented in other works, for example that of Lhomme-Desages [26], with the main difference being how errors in heading and velocity are converted to commanded wheel speeds through knowledge of the vehicle motion model. Receding horizon model predictive control (MPC) has also been proposed and simulated by Burke for trajectory control of skid-steer vehicles with excellent trajectory tracking results in simulation [5].

Similar to prior works in motion modeling of skid-steer vehicles, research in trajectory control has focused on either a wheeled or a skid-steer vehicle and it is
unclear how well each controller will operate if placed on a vehicle with the opposite configuration. The strategy for this thesis was to again utilize ICR kinematics for trajectory control. A skid-steer vehicle will exhibit the same input-output relationship, inputs being track/wheel speeds and outputs being vehicle velocity, as a unicycle robot with drive wheels placed at the ICR locations. A kinematic mapping between skid-steer and unicycle robot motion is implemented so that a unicycle robot control law can be implemented on a skid-steer vehicle. The control process works for both tracked and wheeled vehicles when combined with the ICR estimation algorithm.

1.4 Modeling Power Usage in Skid-Steer Vehicles

The final two research areas addressed, power and energy usage modeling in skid-steer vehicles, have received the least attention in the literature. Modeling skid-steer power usage is difficult because of the complex resistive forces developed between the tracks/wheels and the ground. The work presented by Chuy in 2009 utilizes a linear function of velocity to model power usage. To account for turning of the vehicle, the slope and offset in the linear velocity equation are exponential functions of turning radius [7]. Overall, twelve terrain dependent parameters must be found from experimental data. The resulting model shows excellent agreement.
with experimental results; however, a significant number of experiments were performed to fit the terrain parameters. These tests would need to be performed again for each robot on each type of terrain. A different approach is taken by Morales where track ICR locations are utilized to calculate slip velocities which, when combined with a friction model, produce estimates of power loss due to track terrain interactions [32]. Again, friction constants are learned through post-processing of experimental data, but the resulting power estimates match well with recorded data. The authors also suggest a control strategy for minimizing energy usage: traversal of the path with changes in heading concentrated as sharp, zero-radius turns to maximize time spent driving straight. A third approach, presented by Logan, is modeling power usage through experimentally derived models of power loss due to track/terrain interaction [27]. The resulting models are applicable to all robots of a given locomotion type and show good prediction of endurance, but power usage during turning the vehicle or driving on pitched surfaces is not addressed. Finally, vehicle longitudinal dynamics models have been utilized for predicting the power usage of unmanned ground vehicles [47]. Using the longitudinal model, the power required for skid-steer turning is neglected. This component has been shown to be significant in other work [32].

The approach in this thesis closely follows that of Morales. Instead of optimizing parameters in post-processing, power usage model parameters are identified during vehicle operation using measurement of power usage from sensors on-board
the vehicle. Three terms are accounted for in the power model: power loss due to skid-steer turning, power loss due to rolling resistance and internal resistances such as gears, and the contribution of surface pitch to power usage. Whereas ICR estimation and trajectory control required little knowledge of vehicle configuration, the positions of the tracks or wheels relative to the geometric center of the vehicle and the mass of the vehicle are necessary for the estimation of power usage.

1.5 Mission Energy Prediction and Path Planning

Once a valid skid-steer power model has been found it can be used to predict the energy usage required to traverse a given path. This can be useful in two ways. One, to plan a path through varied terrain so that minimal energy is utilized, and two, to predict the energy usage required to complete a given path or return to the starting point once the path is started.

Energy-aware path planning for skid-steer vehicles is a relatively new area of research, but because energy usage can be thought of as another metric of “cost to go”, many previously developed path planning algorithms can be adapted for energy-aware path planning. Barili proposed that vehicle velocity be regulated to conserve energy for other tasks, but a connection to path planning is not made [3]. The work presented by Mei in 2004 [29] discusses optimizing paths for searching a
predefined area, but is not focused on skid-steer vehicles and turning is not penal-
ized significantly, whereas turning a skid-steer vehicle is presented as the largest
use of power in the work of Chuy and Morales [7,32]. A theoretical method of path
planning in terrain with varying surfaces and slopes is proposed by Reif [59]. After
defining constraints on robot movement from vehicle power limitations and rollover
susceptibility, the BUSHWACK algorithm [58] is used to find the minimal energy
path. As power usage during turning is not modeled, the results are not directly
applicable to skid-steer vehicles. In a work recently published in 2011, Sharma
et al. [51] proposed the use of a Sampling Based Model Predictive Optimization
scheme [14] for energy-aware path planning. The algorithm utilizes the power pre-
diction model from Chuy [7] with A*-type optimization, and simulation results
show small increases in path distance can result in significant energy savings.

Providing feedback to a robot operator on the energy required to complete a
given task is an extremely useful, but extremely difficult, area of research. The
energy required to complete a task is a function of the robot design, the surface
on which the vehicle must operate, and the terrains found in the operational area.
Only recently has work in this area reached the general audience. The work of
Sadrpour, Jin, and Ulsoy [48] and the work presented in this thesis represent the
state of the art in this field. Sadrpour et al. approach the prediction of mission
energy requirements from a Bayesian estimation perspective. A longitudinal ve-
hicle dynamics power model, commonly used in automobile research, is used to
predict the power usage required to traverse a path. A Bayesian estimator is used to determine power model coefficients based on measurements from vehicle sensors. Future information about the terrain slope, surface type, and vehicle speed are encapsulated in probability distributions to improve mission energy prediction.

The path planning approach presented in this work is similar to that of Sharma and uses the power modeling approach developed in this thesis with a kinematically-constrained A*-type optimization. The benefit of this is that the path is guaranteed to be kinematically admissible. However, the Sampling Based Model Predictive Optimization Scheme only provides a solution as optimal as the input space discretization allows. The mission energy usage prediction algorithm varies significantly from other published work. Whereas Sadrpour et al. used very general power usage models and descriptions of the required path, in this work the skid-steer specific power usage model and a simulation of path traversal by the vehicle are used for energy prediction.

1.6 Conclusions

The preceding sections have introduced recent contributions in the areas of skid-steer motion modeling in ground robots, trajectory control of skid-steer vehicles, power usage modeling of skid-steer robots, energy-aware path planning, and mission energy usage prediction. Current motion modeling and power usage modeling techniques rely heavily on post-analysis of recorded data for parameter identifica-
tion. This prevents such algorithms from being easily adapted to multiple robotic platforms. While many published control algorithms for trajectory tracking with skid-steer vehicles exist, there is a significant lack of experimental implementation and control design from an energy conservation standpoint. Power usage modeling of skid-steer vehicle has seen vast improvements in recent years, but the couplings between power prediction, path planning, and mission energy prediction for skid-steer vehicles is on the forefront of current research. The proposed contributions in this work focus on modeling skid-steer movement with an estimator easily adapted to changing terrains and vehicles, applying trajectory control to a skid-steer vehicle using such a motion model, identifying parameters for power usage modeling during operation, and providing feedback to the operator on both estimated mission energy usage and the best trajectory for reaching an end goal. Also, the algorithms presented in this work are valid for both tracked and wheeled skid-steer vehicles.

The remaining chapters of this thesis are organized as follows. Chapter 2 provides a description of the test vehicles created for this research. Chapter 3 gives a description and field test results of a kinematic EKF that learns track ICR locations to predict movement, and Chapter 4 provides the development and results of an ICR-based trajectory controller for skid-steer vehicles. Chapter 5 presents the development and testing of a power usage model for skid-steer vehicles which is then applied in Chapter 6 for the prediction of energy usage for skid-steer robots. Chapter 7 presents a path planning algorithm utilizing an energy-
based cost function. Conclusions and suggestions for future work are presented in Chapter 8.
Chapter 2

Description of Test Platforms

2.1 Introduction

This chapter contains descriptions of the skid-steer test platforms, shown in Figure 2.1, developed for implementation and testing of this research. The first section will discuss hardware common to both platforms, and the following sections will discuss the tracked vehicle and wheeled vehicle in more detail.

2.2 Equipment Shared Between Platforms

The Hemisphere A325 Smart Antenna, mounted on top of the mast in each picture in Figure 2.1, is utilized for absolute positioning of the vehicles. The unit combines a GPS L1/L2 capable antenna with a receiver in one unit and is capable of providing real-time kinematic (RTK) solutions within 2 cm at 20 Hz. RTK solutions require a nearby base station at a surveyed location to transmit correc-
Figure 2.1. Skid-steer robots used during experimental testing.

tions. Currently, an identical A325 is used for transmitting corrections in Radio Technical Commission for Maritime Services Version 3 (RTCMv3) format. The corrections are transmitted with Freewave FGR2 radios on the robot and at the receiver. Tests with a Novatel base station GPS have proven successful, which may enable the usage of a permanent base-station with wide-area coverage in the future.

An Xsens MTi 300 system, shown in Figure 2.2 provides estimates of roll, pitch, and yaw at 100 Hz. The accuracy is specified to be 1 degree in heading and 0.3 degrees in roll and pitch, assuming sources of magnetic fields on the vehicle have been properly accounted for, and Xsens provides a calibration program for this purpose.

High-level computation and data recording are done on embedded computers
running the open source Robotic Operating System (ROS) architecture [44]. ROS is not a complete operating system, but rather a collection of open source programs and libraries released under the BSD license that facilitate hardware interfacing, distributed processing, algorithm development, and data visualization. A ROS system is constructed of individual nodes capable of advertising or subscribing to information topics published by other nodes. The ROS libraries handle all inter-process communication, making it fairly trivial to collect data from multiple sensors, such as a LiDAR, a camera, and an encoder, in a single node for processing. ROS also provides a convenient framework for recording data in binary format that allows previously recorded tests to be played back in real time, allowing for algorithm development without operating the actual system.

2.3 Tankbot Description

The tracked vehicle was designed and built at The Pennsylvania State University for prior research work and is named the Tankbot. For this work, the original chassis and motors are being used but all sensors, computing systems, and motor
controllers have been newly installed. The Tankbot drive system, diagrammed in Figure 2.3, consists of two Roboteq LDC1430 motor controllers operating two brushed DC motors, one for each track. The DC motors have a motor constant value of 0.0893 N-m/A. Optical encoders with a resolution of 0.044 degrees per count measure the speed of the final drive shaft which is connected to the tracks through a sprocket. A labeled picture of the Tankbot drive system is shown in Figure 2.3. Encoder outputs are both logged and fed into the LDC1430 for closed loop control of track speed. Reference track speeds are fed to the controller through a RS232 serial connection. The gear ratio between the motor output shaft and the track drive shaft is 9.78:1, and the radius of the track drive sprocket is 0.0635 m.

Figure 2.3. Flowchart of Tankbot manual control and drive systems.
Figure 2.4. Labeled picture of Tankbot drive system.

The Tankbot can be controlled manually using a 2.4 GHz radio controller by processing the received RC signal with an Arduino Mega microprocessor and then sending the commands to the main computer via USB. The electrical current between each motor controller and the left and right motors is measured using a LEM LTS15-NP Hall effect sensor, which is sampled by an Arduino Uno microprocessor and transmitted to the main computer via USB. The width and mass of the Tankbot are 0.52 m and 80.3 kg, respectively. The locations of the Tankbot bogie wheels relative to the frame geometric center are diagrammed in Figure 2.5.

The main computer and voltage regulation systems are shown in Figure 2.6. The computer is a JetWay JNC9B-HM67 motherboard with 8 GB of RAM and an Intel Core i5 processor running at 2.5 GHz. The hard drive is a 256 GB solid-state drive. The voltage regulation board was designed by the Applied Research Lab at
Figure 2.5. Dimensions of Tankbot bogie wheels relative to frame geometric center. Dimensions in meters and Tankbot is symmetrical about the X and Y axes.

The Pennsylvania State University. This board regulates the voltage output of a BB2590 battery to 24 V, 12 V, and 5 V to supply power to other sensors on-board the vehicle. Additionally, the voltage regulation board includes connections for the motor drive shaft optical encoders and the associated counters, shift registers, and Arduino Mega microprocessor needed to send encoder count values to the main computer.

Additional sensors available on the Tankbot include an Analog Devices ADIS 16407 10 degree of freedom inertial sensor, a Bumblebee stereo camera, and an Axis P3304 Ethernet enabled camera. While used in a limited capacity for this work, these sensors provide additional research opportunities for this vehicle. The Tankbot drive system is powered by four 12 volt lead acid batteries, and the computer and sensors are powered by a single BB2590 lithium-ion battery. Endurance
testing has not been conducted on the current Tankbot design, but field testing lasting approximately three hours has been done with battery reserves remaining at the end.

### 2.4 RMP Description

The wheeled vehicle is the RMP400, which was at one time a commercially available platform from Segway Robotics. The RMP 400 is a large and powerful robot, with a mass of 138 kg and a top speed of 29 kph. The RMP400 drive train consists of two units with each controlling a left and right wheel. Each unit contains two
lithium-ion batteries that together provide a range of 19-24 km when new. The locations of the RMP400 wheel centers relative to the geometric center of the vehicle frame are diagrammed in Figure 2.7.

An embedded computer, shown in Figure 2.8, has been mounted on the RMP400 to interface with the Segway motor controllers and added sensors. The computer is an ASUS AT3N7A-1 motherboard with 2 GB of RAM and an Intel Atom processor running at 1.6 GHz. A solid-state hard drive of 40 GB capacity is used for storage. Two BB2590 batteries are used on the RMP400, one to power the computer and the second to power the GPS receiver and the radio modem.

The motor control systems included on the RMP400 report vehicle status information at 100 Hz. The status message includes wheel speeds, motor torques, and battery voltages among other quantities. The availability of this feedback is beneficial in that additional sensors and motor controllers are not needed. However, the

![Image](image-url)

**Figure 2.7.** Dimensions of RMP400 wheel centers relative to the geometric center of the vehicle frame. Dimensions are in meters, and the RMP 400 is symmetric about the X and Y axes.
design of the RMP400 makes it very difficult to install additional sensing systems. For example, it would be useful to measure the power output of the drive batteries on the RMP400, but the design of the system makes adding the needed hardware infeasible. The RMP400 can be manually controlled using an RC controller just as described for the Tankbot.

2.5 Conclusions

The Tankbot and RMP400 provide tracked and wheeled platforms for testing and evaluation. This is useful because most published research is focused on one type of drive system. The Tankbot is by far the most open access of the two platforms. Having been custom built, all of the systems are open to inspection and available for measurement with proper sensors. The RMP, on the other hand, is a fairly closed
system. The motor drivers, motors, transmissions, and batteries are all within the sealed factory enclosure, making it extremely difficult to add new sensors within these systems. One important example is measurement of power usage. Electric current sensors have been installed in the Tankbot drive train, but it is extremely difficult to do so on the RMP400. The RMP400 does provide feedback of motor torque and wheel speed, which can be used to estimate the output power of the motors. However, accuracy values for these measurements are not provided, and it is difficult to verify them experimentally. The RMP400 is still very useful for energy usage testing because predicted energy usage can be compared with energy usage measured on the vehicle. It is more difficult, however, to compare the energy usage of the Tankbot to the RMP400.
Chapter 3

Estimation and Modeling of Skid-Steer Robot Motion

Author’s Note: The majority of the work presented in this chapter has been published by the author in a research paper in the Journal of Field Robotics [40].

3.1 Introduction

This chapter will discuss the development and testing of an extended Kalman filter (EKF) designed to estimate the locations of the instantaneous centers of rotation (ICRs) between the tracks/wheels of a skid-steer robot and the ground during operation. ICR skid-steer kinematics, well presented by Martinez [28], are unique in the way track/terrain interactions, including slippage, are captured within the locations of the track or wheel ICRs. While other algorithms have been presented that estimate track or wheel slip during operation [5, 11], the results show track slippage varying significantly over very short time periods, as small as a second in some instances. An example of slip variation from Burke is shown in Figure 3.1 [5].
These algorithms often use global position measurements of some kind, and in the absence of these measurements, such as a temporary GPS dropout, it is unclear how accurate the estimation of position would be.

![Figure 3.1](image.png)

**Figure 3.1.** Plot of measured and estimated slip with skid-steer robot from Burke [5]. Labels $s_1$ and $s_2$ correspond to the right and left wheels, respectively. Original caption: “Slip values (blue) and the corresponding estimates (red) obtained using adaptive least squares.”

ICR locations, however, have been shown to vary much less and testing presented in later sections shows good kinematic predictions of motion without global position measurements. The presented algorithm also requires no knowledge of vehicle or terrain parameters for operation; only estimates of sensor and process noise are needed to run the filter. The remaining sections in this chapter will provide the ICR kinematic equations in detail, show the development and analysis of the ICR EKF, provide simulation and testing results from the ICR EKF, and also
give an overview of alternate ICR estimation methods studied during this research.

3.2 Approximating the Kinematics of Skid-Steer Robots

For simplicity, the text in this section will refer to the tractive surfaces of a skid-steer vehicle as tracks; however, the developments presented herein are equally valid for wheeled skid-steer vehicles. It has been shown that skid-steer dynamics can be encapsulated in the track ICR locations of a mobile robot, and once known, ICR locations can be used to produce accurate movement estimates. The ICR of a body in planar motion is the point that has zero velocity for that instant in time. At that instant, all other points in the body are in pure rotation about the ICR [60]. A planar example of an ICR for a rigid body is shown in Figure 3.2. Each point, A, B, and C, are in pure rotation about the ICR P. The velocity of a point can be calculated as

\[ v_A = \omega \times r_{PA} \] (3.1)

where \( v_A \) is the velocity of point A, \( \omega \) is the angular velocity of the body, and \( r_{PA} \) is the vector from the ICR P to the point of interest A.

Figure 3.3 shows the track ICR locations relative to a right-hand body fixed coordinate system (X, Y, Z). The Y coordinate of the ICR location between the robot chassis and the ground, \( y_{ICRv} \), is

\[ y_{ICRv} = \frac{v_x}{\omega_z} \] (3.2)
where $v_x$ is the vehicle velocity along the $X$ axis and $\omega_z$ is the angular velocity of the vehicle. The $Y$ coordinates of the ICR locations between the tracks and the ground can be found by introducing the individual track velocities, yielding

$$y_{ICR_l} = -\frac{V_l^x - v_x}{\omega_z}$$  \hspace{1cm} (3.3)

$$y_{ICR_r} = -\frac{V_r^x - v_x}{\omega_z}$$  \hspace{1cm} (3.4)
where \( y_{ICR_l} \) and \( y_{ICR_r} \) are the \( Y \) coordinates of the ICR locations for the left and right hand tracks, respectively, \( V^l_x \) is the velocity of the left hand track relative to the body of the robot, and \( V^r_x \) is the velocity of the right hand track relative to the body of the robot. The \( X \) coordinate of the vehicle ICR location, \( x_{ICR_v} \), is found using

\[
x_{ICR_v} = -\frac{v_y}{\omega_z}. \tag{3.5}
\]

where \( v_y \) is the lateral velocity of the vehicle. The \( X \) location of the left track ICR can be calculated in a similar way to (3.3) using

\[
x_{ICR_l} = \frac{V^l_y - v_y}{\omega_z}. \tag{3.6}
\]

Because the spinning of the track is assumed to induce no velocity to the track in the \( Y \) direction, \( V^l_y = 0 \) and (3.6) reduces to

\[
x_{ICR_l} = -\frac{v_y}{\omega_z}. \tag{3.7}
\]

A similar analysis can be conducted for \( x_{ICR_r} \) with the result

\[
x_{ICR_r} = x_{ICR_l} = x_{ICR_v} = -\frac{v_y}{\omega_z}. \tag{3.8}
\]

Thus, the three ICR locations, those of the vehicle, left track, and right track, lie along a straight line parallel to the body-fixed \( Y \) axis.
It should be noted that the values of $y_{ICRe}$ range within $\pm \infty$ depending on vehicle motion. However, the values of $y_{ICRr}$, $y_{ICRl}$, and $x_{ICRe}$ remain bounded. As vehicle motion approaches that of driving in a straight line, the numerators and denominators of (3.3)-(3.5) are infinitesimals of the same order, resulting in finite values for the ICR locations [28].

Equations (3.3)-(3.5) can be solved to express body frame velocities in terms of input track velocities and track ICR locations. The resulting kinematic model,

\begin{align*}
v_x &= \frac{V_x^r y_{ICRl} - V_x^l y_{ICRr}}{y_{ICRI} - y_{ICRr}} \quad (3.9) \\
v_y &= \frac{(V_x^l - V_x^r)x_{ICRe}}{y_{ICRI} - y_{ICRr}} \quad (3.10) \\
w_z &= -\frac{V_x^l - V_x^r}{y_{ICRI} - y_{ICRr}} \quad (3.11)
\end{align*}

forms the basis for the EKF described in the next section. The denominator of (3.9)-(3.11) consists of $y_{ICRI} - y_{ICRr}$, so the possibility of division by zero must be addressed. Due to slippage, the ICR locations must lie outside the track centerlines [28]. Thus, to have division by zero, the tracks or wheels would need to lie along the same line, a physical impossibility for a skid-steer mobile robot. It is also interesting to note that values of $y_{ICRl}$ and $y_{ICRr}$ approach infinity when the robot is stationary but the tracks are spinning. This may occur when the vehicle operates on a low friction surface or becomes stuck in some way. Rapidly increasing values of $y_{ICRI}$ and $y_{ICRr}$ could be used to detect occurrences of pure slip such as these.
3.3 EKF for ICR Estimation

This section will discuss the formulation of a kinematic EKF for the estimation of ICR location. Starting with the kinematic equations (3.9)-(3.11), the velocities of the robot in the north, $\dot{N}$, and east, $\dot{E}$, directions are found to be

\[
\dot{N} = v_x \cos \psi - v_y \sin \psi
\]  \hspace{1cm} (3.12)
\[
\dot{E} = v_x \sin \psi + v_y \cos \psi
\]  \hspace{1cm} (3.13)

where $\psi$ is the heading angle as defined in Figure 3.4. The equations of motion for

Figure 3.4. Relation between body frame and fixed frame through heading angle $\psi$. 
where $w_N - w_x$ are additive zero-mean Gaussian process noises. Previous work in ICR kinematics has shown that the ICR locations remain within a small bounded region for robots traveling at low speeds on hard, flat terrain [28]. These criteria apply to the majority of testing in this work, so the ICR locations are modeled as constants disturbed by random noise, and results presented later confirm that ICR locations remain bounded to the same region regardless of maneuver type. Additional testing to explore the variation of ICR locations with increasing speed is presented in Section 3.5.6.
The kinematic model is discretized using the Euler method with the result

\[
\begin{bmatrix}
N_k \\
E_k \\
\psi_k \\
y_{ICRr_k} \\
y_{ICRl_k} \\
x_{ICRv_k}
\end{bmatrix}
= 
\begin{bmatrix}
N_{k-1} + \Delta t V_{N_{k-1}} + \Delta t w_N \\
E_{k-1} + \Delta t V_{E_{k-1}} + \Delta t w_E \\
\psi_{k-1} + \Delta t \omega_{2_{k-1}} + \Delta t w_\omega \\
y_{ICRr_{k-1}} + \Delta t w_r \\
y_{ICRl_{k-1}} + \Delta t w_l \\
x_{ICRv_{k-1}} + \Delta t w_x
\end{bmatrix} 
\]  

(3.15)

where \( \Delta t \) is the time step. The time step currently used on the vehicles is 0.01 seconds which matches the update rate of the heading sensor. Euler methods are sensitive to time step and model dynamics, but the filter has proven to be stable and accurate in simulation and testing with the current time step.

The EKF utilizes the discrete time model (3.15), denoted compactly as

\[
x_k^- = f(x_{k-1}^+, u_{k-1}, 0) 
\]  

(3.16)

where \( x_k^- \) is the propagated state at the current time step, \( x_{k-1}^+ \) is the propagated and updated state from the previous time step, \( u_{k-1} \) is the vector of control inputs consisting of left and right track velocity, and the third entry in \( f(x_{k-1}^+, u_{k-1}, 0) \) denotes setting the process noise to zero for propagation. Throughout this paper, a superscript ‘-’ denotes a value propagated but not updated with measurements, and a superscript ‘+’ denotes a value propagated and updated with measurements.
The state covariance, \( P \), is propagated using the standard Kalman equation \cite{52}:

\[
P_k^- = F_{k-1} P_{k-1}^+ F_{k-1}^T + L_{k-1} Q_{k-1} L_{k-1}^T
\] (3.17)

where \( F_{k-1} \) and \( L_{k-1} \) are Jacobian matrices defined as

\[
F_{k-1} = \frac{\partial f_{k-1}}{\partial x} \bigg|_{x_{k-1}^+}
\] (3.18)

\[
L_{k-1} = \frac{\partial f_{k-1}}{\partial w} \bigg|_{x_{k-1}^+} = \Delta t I_{6x6}
\] (3.19)

and \( Q \) is the process noise covariance, assumed to be uncorrelated.

To update the state estimate, measurements of vehicle north and east position, \( N_{GPS} \) and \( E_{GPS} \), and vehicle heading, \( \psi_V \), are available from sensors on the robot and collected in the vector \( y \) as

\[
y_k = \begin{bmatrix}
N_{GPS} \\
E_{GPS} \\
\psi_V
\end{bmatrix}_k
\] (3.20)

These direct measurements of states result in the measurement equations

\[
h_k = \begin{bmatrix}
N + v_N \\
E + v_E \\
\psi + v_\psi
\end{bmatrix}_k
\] (3.21)
where \( v_N - v_\psi \) represent additive measurement noise. The filter gain is calculated using the standard Kalman equation [52]

\[
K_k = P_k^{-1} H_k^T (H_k P_k^{-1} H_k^T + M_k R_k M_k^T)^{-1}
\]  
(3.22)

where \( H_k \) and \( M_k \) are Jacobian matrices defined as

\[
H_k = \frac{\partial h_k}{\partial x} \bigg|_{x_k^-} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}
\]  
(3.23)

\[
M_k = \frac{\partial h_k}{\partial v} \bigg|_{x_k^+} = I_{3 \times 6}.
\]  
(3.24)

Finally, the state and covariance estimates are updated using the standard Kalman equations [52]

\[
x_k^+ = x_k^- + K_k [y_k - h_k(x_k^-, 0)]
\]  
(3.25)

\[
P_k^+ = (I_{6 \times 6} - K_k H_k) P_k^-
\]  
(3.26)

### 3.3.1 Investigation of Conditions for ICR Identification

The structure of the kinematic model and measurement model provide insight into the ability of the presented EKF to estimate ICR locations. The state update (3.25) computes the correction to the state through the product of the Kalman
gain, $K_k$, and the measurement innovation, $y_k - h_k(x_k^-, 0)$. The measurement innovation consists only of the difference between the measured and estimated north position, east position, and heading of the vehicle, and therefore the coupling between measurement innovation and the ICR location update comes about through the Kalman gain.

It is most interesting to determine when the EKF will not update the ICR location estimates. This occurs when the bottom three rows of the Kalman gain are zero, the condition for which can be found from (3.22). The term $P_k^{-1}H_k^T$ in (3.22) evaluates to be

$$P_k^{-1}H_k^T = \begin{bmatrix}
\sigma_N^2 & \sigma_N \sigma_E & \sigma_N \sigma_\psi \\
\sigma_N \sigma_E & \sigma_E^2 & \sigma_E \sigma_\psi \\
\sigma_N \sigma_\psi & \sigma_E \sigma_\psi & \sigma_\psi^2 \\
\sigma_N \sigma_{y_{ICRv}} & \sigma_E \sigma_{y_{ICRv}} & \sigma_\psi \sigma_{y_{ICRv}} \\
\sigma_N \sigma_{y_{ICRl}} & \sigma_E \sigma_{y_{ICRl}} & \sigma_\psi \sigma_{y_{ICRl}} \\
\sigma_N \sigma_{x_{ICRv}} & \sigma_E \sigma_{x_{ICRv}} & \sigma_\psi \sigma_{x_{ICRv}} 
\end{bmatrix}$$

where $\sigma$ is the variance of the estimator error for the subscripted state at the current time step. The final three rows of the Kalman gain will become zero when the covariances between the estimate errors of robot pose and ICR locations are zero. The state error covariance matrix $P$ is initialized to be a diagonal matrix, and the process noise is assumed to be additive and uncorrelated, leaving the term
\( \mathbf{F}_{k-1} \mathbf{P}_{k-1}^{+} \mathbf{F}_{k-1}^{T} \) in (3.17) as the only contribution to the estimate error covariance between robot pose and ICR location. The linearized kinematic model matrix, \( \mathbf{F}_{k-1} \), can be split into four parts as

\[
\mathbf{F}_{k-1} = \begin{bmatrix}
\mathbf{F}_{3x3} & \mathbf{F}_{3x3} \\
0_{3x3} & \mathbf{I}_{3x3}
\end{bmatrix}
\]

(3.28)

where \( \mathbf{F}_{3x3} \) and \( \mathbf{F}_{3x3} \) are the Jacobians of the first three rows of (3.15) with respect to the estimated robot pose and ICR locations, respectively. When the state error covariance \( \mathbf{P} \) is a diagonal matrix, such as when it is initialized, the only non-zero terms in the calculation of covariance between estimate errors of robot pose and ICR locations, computed by (3.17), consist of terms in \( \mathbf{F}_{3x3} \) multiplied by ICR location estimate error variances. Therefore, estimation of ICR locations depends on the values of \( \mathbf{F}_{3x3} \) being non-zero.

Two examples of the terms in \( \mathbf{F}_{3x3} \) are

\[
\frac{\partial N_k}{\partial y_{ICRr}} = \Delta t \left( \frac{(V_r y_{ICRl} - V_l y_{ICRr}) \cos \psi}{(y_{ICRl} - y_{ICRr})^2} - \frac{V_l \cos \psi}{y_{ICRl} - y_{ICRr}} - \frac{(V_l - V_r) x_{ICRr} \sin \psi}{(y_{ICRl} - y_{ICRr})^2} \right)
\]

(3.29)

and

\[
\frac{\partial N_k}{\partial x_{ICRr}} = -\frac{\Delta t (V_l - V_r) \sin \psi}{y_{ICRl} - y_{ICRr}}
\]

(3.30)

with all other terms having a similar structure. When the track or wheel speeds are equal, \( V_r = V_l \), (3.29) and (3.30) evaluate to zero, as do all other terms in \( \mathbf{F}_{3x3} \).
The final conclusion of this analysis is that the EKF estimates of ICR locations are updated only when the vehicle is turning, or $V_r \neq V_l$. Thus, the ICR EKF conditionally updates the estimates of ICR location based on vehicle motion. This is not thought to be a large weakness of the ICR EKF because ground mobile robots do not generally spend significant periods of time driving in straight lines. Many current ground mobile robot designs feature cameras fixed to the chassis of the vehicle, forcing the vehicle to turn to view other areas of the operational environment. The filter structures also prevents adjustment of the ICR location estimates when the vehicle is stationary, which is beneficial in that previously learned ICR information is retained when the vehicle momentarily stops moving.

### 3.4 Simulation and Analysis of ICR EKF

The ICR EKF was developed and tested using MATLAB to investigate convergence properties and resilience to noise. The true state model consisted of the kinematic equations in (3.9)-(3.11). ICR locations of $y_{ICR_r} = 0.5 \text{ m}$, $y_{ICR_l} = -0.75 \text{ m}$, and $x_{ICR_v} = -0.25 \text{ m}$ were chosen based on the test robot dimensions of 0.51 m in width by 0.72 m in length. Measurements of position and heading were created by corrupting outputs of the kinematic model with zero-mean Gaussian noise of standard deviation 0.01 m for position and 1 degree for heading. These error values were chosen to simulate the performance of the sensors used in testing, described in Chapter 2.
The process noise covariance and measurement noise covariance must be defined to run the EKF. The process noise covariance $Q$ was set to be

\[ Q = \begin{bmatrix}
0.1^2 & 0 & 0 & 0 & 0 & 0 \\
0 & 0.1^2 & 0 & 0 & 0 & 0 \\
0 & 0 & \left(\frac{\pi}{180}\right)^2 & 0 & 0 & 0 \\
0 & 0 & 0 & 0.01^2 & 0 & 0 \\
0 & 0 & 0 & 0 & 0.01^2 & 0 \\
0 & 0 & 0 & 0 & 0 & 0.01^2 \\
\end{bmatrix}. \tag{3.31} \]

The first two diagonal terms of $Q$ represent process noise of 0.1 m/s standard deviation in the north and east. To determine noise level, the tracked robot was driven in a straight line by commanding a track velocity of 0.40 m/s to each side, which is a speed representative of the experimental data presented in following sections. The standard deviation of the noise in the velocity output of the RTK GPS system was calculated and used as the process noise in the north and east. The heading measurement of the AHRS system was processed for the same straight line run to determine the process noise on vehicle heading, which is the third diagonal term. The remaining three diagonal terms relate to the ICR estimates and were chosen by manually adjusting the process noise to minimize errors between simulated and measured positions of the vehicle, using off-line recordings of one data run. While more formal minimization methods could have been used, the
sensitivity of the estimation process to these parameters was low enough that errors in manual tuning clearly do not affect feasibility of the algorithm, and only affect convergence rates slightly.

The measurement noise covariance matrix $\mathbf{R}$ was set as

$$
\mathbf{R} = \begin{bmatrix}
0.01^2 & 0 & 0 \\
0 & 0.01^2 & 0 \\
0 & 0 & \left( \frac{1\pi}{180} \right)^2
\end{bmatrix}
$$

(3.32)

where the values of 0.01 m in the position measurement variances represent the accuracy of the GPS system and the value of $\left( \frac{1\pi}{180} \right)$ radians represents the accuracy of the AHRS system, both described in Chapter 2.

A comparison of the EKF position estimate and true position is given in Figure 3.5. The accurate position and heading measurements enable the filter to match the true path while quickly learning the correct ICR locations of $y_{ICRr} = 0.5$, $y_{ICRl} = -0.75$, and $x_{ICRv} = -0.25$, as shown in Figure 3.6. Some modifications to the kinematic equations were necessary to avoid divergence of the filter estimate in the presence of noise and disturbances. The next phase of simulation focused on these modifications.
3.4.1 Preventing Divergence Due to ICR Sign Inversion

It was found in initial testing that filter estimates of ICR locations diverged during a significant number of simulation runs. The simulation was repeated with initial ICR location estimates ranging from 2 to -2. The results, shown in Figure 3.7, indicate that \( y_{ICRr} \) converges only when the initial condition is greater than zero.
A similar plot for $y_{ICRl}$ shows convergence only when $y_{ICRl}$ is initialized with a negative number.

The kinematic modeling equations, given in (3.5) - (3.11), were modified to ensure filter convergence. With respect to the defined body frame, shown in Figure 3.3, the left ICR location will always be negative and the right ICR location will always be positive. To enforce this knowledge of ICR location, the kinematics were modified to be

\begin{align*}
v_x &= \frac{V_r^x y_{ICRl} - V_l^x y_{ICRr}}{-|y_{ICRl} - y_{ICRr}|} \\
v_y &= \frac{(V_l^x - V_r^x) x_{ICR}}{-|y_{ICRl} - y_{ICRr}|} \\
w_z &= -\frac{V_l^x - V_r^x}{-|y_{ICRl} - y_{ICRr}|}
\end{align*}

Figure 3.7. Plot of $y_{ICRr}$ estimate during tracked vehicle simulations for initial conditions ranging from 2 through -2 using kinematic equations (3.9) - (3.11).
such that the difference value $y_{ICRl} - y_{ICRr}$ will always have the correct negative sign. With this change, the filter converges regardless of initial condition or variation after initialization. The result is that for any initial condition the ICR location converges to the correct value as shown in Figure 3.8.

![Figure 3.8](image)

**Figure 3.8.** Plot of $y_{ICRr}$ estimate during tracked vehicle simulations for initial conditions ranging from 2 through -2 with modified kinematic equations (3.33) - (3.35).

### 3.4.2 Filter Convergence Response to Changing ICR Locations and Vehicle Turn Radius

The system model used in the ICR EKF, given in equation (3.15), assumes the ICR locations are constant values disturbed by random noise. It is important to investigate how this assumption affects the ability of the ICR EKF to estimate changing ICR locations, which may occur when a robot drives from one surface type to another. This scenario was examined in simulation with the results shown in Figure 3.9. The times at which the ICR locations changed are also labeled in Figure 3.9. The convergence time for the filter after a change in ICR location
ranged from 100 seconds to 140 seconds. In this work, the definition of rise time [25] is used to define convergence rate, e.g. the time from 10% change in ICR estimate to when the estimate is within 90% of the final value. The time of convergence in Figure 3.9 is slow, but can be improved significantly by inflating the ICR estimate state error covariances when a change in terrain is detected. This process was also simulated and is shown in Figure 3.10. The convergence times are much faster and range from 0.03 seconds to 0.17 seconds when the state error covariance is inflated to 100 m$^2$ at the time the true ICR locations change. Because both ICR location and skid-steer power usage are related to the surface of operation, a surface detection algorithm utilizing errors between measured and modeled power usage can be used to detect surface changes. This is described more fully in Chapter 5.

The change in convergence rate with vehicle turn radius was also investigated

![Figure 3.9](image)  
**Figure 3.9.** Simulation run with changing ICR locations. Times of ICR changes are labeled with vertical lines at 175 seconds and 350 seconds.
in simulation. The results, shown in Figure 3.11, show a clear increase in convergence time with larger turn radii. The convergence times given in Figure 3.11 are the averages over twenty simulation runs at the corresponding turn radius. The correlation between turn radius and convergence time is expected because of the relation between vehicle angular rate and the ability of the ICR EKF to identify ICR locations, as discussed in Section 3.3.1.
3.4.3 Steady-State Analysis of ICR EKF

The steady-state error covariance and Kalman gain matrices of a linear Kalman filter can often be calculated if the system model, process noise, and measurement noise are known [52]. Calculating these parameters for an extended Kalman filter is very difficult because the system model is linearized about the current state at each time step resulting in a changing state transition matrix. In the case of a steady-state turn, however, it is possible to analyze the ICR EKF and determine an estimate of the state error covariance.

The time-update equation for the covariance is given in Section 3.3 as

\[ P_k^- = F_{k-1} P_{k-1}^+ F_{k-1}^T + L_{k-1} Q_{k-1} L_{k-1}^T. \]  \hspace{1cm} (3.36)
Substituting the expression for $P_{k-1}^+$, given in (3.26), yields

$$P_k^- = F_{k-1}P_{k-1}^-F_{k-1}^T + F_{k-1}K_kH_kP_{k-1}^-H_k^TF_{k-1}^T + L_{k-1}Q_{k-1}L_{k-1}^T. \quad (3.37)$$

The final step is the substitution of the expression for $K_k$, given in (3.22), to yield

$$P_k^- = F_{k-1}P_{k-1}^-F_{k-1}^T + F_{k-1}P_{k-1}^-H_k^T(H_kP_{k-1}^-H_k^T + M_kR_kM_k^T)^{-1} \times H_kP_{k-1}^-F_{k-1}^T + L_{k-1}Q_{k-1}L_{k-1}^T. \quad (3.38)$$

If $P_k^-$ converges to a steady-state value, then $P_k^- = P_{k-1}^- = P_\infty$, where $P_\infty$ is the steady-state error covariance. Furthermore, under the steady-state turn assumption mentioned previously, the state transition matrix $F_k$ will be constant. This results in the equation

$$P_\infty = FP_\infty F^T + FP_\infty H(HP_\infty H^T + MRM)^{-1} \times HP_\infty F^T + LQL^T. \quad (3.39)$$

This is a form of the discrete algebraic Ricatti equation (DARE), and many software applications provide functions for finding a solution. Using the vehicle and filter parameters previously described for the ICR EKF simulation, the estimated steady-state error standard deviations for the three ICR states were calculated at a variety of angular rate values. The results are shown in Figure 3.12 and the ICR
state error standard deviation shows a significant dependence on the vehicle angular rate. As expected from the previous discussion on ICR identification, given in Section 3.3.1, it is not possible to solve the DARE when the vehicle drives perfectly straight due to the lack of covariance between the position and ICR states. The steady-state error decreases rapidly as angular rate increases, once again showing the need for vehicle heading changes when learning ICR locations.

![Graph showing predicted steady-state error standard deviation for ICR states at varying vehicle turn rates.](image)

**Figure 3.12.** Predicted steady-state error standard deviation for ICR states at varying vehicle turn rates.

### 3.4.4 Response of Filter Due to Input Bias and Noise

The preceding simulation results contain realistic sensor noise on the position and heading measurements, but noise was not introduced on the input track velocities. Two sources of left and right track velocities are available on the robots used for testing: commanded velocity to the motor controller and measured track velocity from encoders or wheel velocity from motor controllers. A comparison of the two
values for a representative run with the tracked robot is shown in Figure 3.13. There is a disadvantage to using either signal as an input to the filter. The commanded speed given to the motor controller is noise free, but a bias exists between the commanded speed and the measured speed of the track. The track speed calculated from encoder measurements is more accurate, but is corrupted by noise from differentiating encoder counts. Understanding how the ICR EKF responds to biased or noisy inputs is key for implementation on a robot.

Some insight to the effect of noise and bias on ICR estimate can be gained by analyzing the ICR kinematic equations given in (3.9)-(3.11). To simplify analysis, it is helpful to assume symmetric ICR locations such that \( y_{ICRr} = -y_{ICRl} \). Making this substitution in the ICR kinematic equations yields

\[
v_x = \frac{V_x^r + V_x^l}{2}
\]  

\[3.40\]

Figure 3.13. Comparison of commanded and measured left track velocity for tracked vehicle.
\[ v_y = \frac{(V_x^l - V_x^r)x_{ICRe}}{2y_{ICRI}} \]  
\[ w_z = -\frac{V_x^l - V_x^r}{2y_{ICRI}}. \]  

Starting with the expression for longitudinal velocity, \( v_x \), the addition of the input track speeds \( V_x^r \) and \( V_x^l \) makes longitudinal velocity fairly impervious to random noise but susceptible to biases. The magnitude of track velocity during operation is around 0.1 - 0.5 m/s, as shown in Figure 3.13. The longitudinal velocity will have the correct sign until noise values reach unreasonable magnitudes. Bias terms, however, are additive in longitudinal velocity and will bias ICR location estimates. Bias and noise have an opposite effect on lateral velocity \( v_y \) and angular velocity \( \omega_z \). While the subtraction of input terms in the numerator helps mitigate bias influence, it makes noise influence a significant factor. The magnitude of noise on the input signal must only be as large as the difference between left and right track velocities to cause a flip in the sign of lateral velocity and angular velocity. This leads to the conclusion that noise on the input signals will cause larger ICR estimate errors then biases. This was confirmed with the simulation by adding noise and then biases to the input signals, running the simulation multiple times, and then averaging ICR estimate error after convergence.

The results for adding input noise are shown in Figure 3.14 and for adding bias in Figure 3.15. As expected, adding noise to the input signal has a much larger effect than adding bias, which has a fairly negligible effect up to any reasonable
bias value. Bias values for motor controller track velocity versus encoder measured track velocity in field testing were below 0.05 m/s, corresponding to an extremely small ICR estimate error.

![Graph](image)

**Figure 3.14.** Error in ICR location estimate as a function of increasing input noise on simulated tracked robot.
Field testing was performed on a fairly level, shortly mowed grass field and asphalt area. Three types of maneuvers were performed with a human operator controlling the tracked robot and the wheeled robot on grass at similar speeds: repeated circles, driving on a curved path, and a long distance mission. In additional testing, a portion of the long distance mission was repeated without RTK corrections to investigate the accuracy of the filter with a more commonly used GPS solution, and the two robots were operated at varying speeds on asphalt and grass to investigate differences in filter performance caused by speed and terrain.

For the first two tests presented, to analyze the performance of the ICR estimation, the position of the vehicle was calculated open-loop using only the kinematic equations and the final ICR estimates from the EKF. The final estimate was chosen
because, for these tests, the goal was to confirm that off-line processing of kinematic data gives good agreement to predicted motion, similar to the work done by [28]. For all other open-loop odometry presented, the ICR estimates used were the most recent estimates from the ICR EKF for that time step. For all open-loop odometry, no information from the GPS or AHRS was used. This ICR odometry estimate is compared with standard two-wheel no-slip odometry, defined as

\[
V_{NS} = \frac{V^r_x + V^l_x}{2},
\]

\[
\omega_{NS} = \frac{V^l_x - V^r_x}{b},
\]

where \(V_{NS}\) is the forward velocity of the vehicle, \(\omega_{NS}\) is the angular velocity of the vehicle, and \(b\) is the distance between the tracks or wheels.

The first three tests, presented in Sections 3.5.1-3.5.3, involved the vehicle traveling over circular or curved paths. The distance traversed by the vehicle in these tests is summarized along with the mean \(\mu\) and standard deviation \(\sigma\) of the vehicle speed and estimated ICR locations in Table 3.1 for the tracked vehicle and Table 3.2 for the wheeled vehicle. Section 3.5.4 presents the results of utilizing the ICR EKF for odometry predictions when GPS is unavailable, and Section 3.5.5 presents performance results of the ICR EKF when a conventional GPS solution is used for position measurements. Section 3.5.6 presents experimental results for the performance of the EKF with increasing vehicle speeds.
Table 3.1. Field test parameters for tracked vehicle.

<table>
<thead>
<tr>
<th>Test Name</th>
<th>Traversal (m)</th>
<th>Speed (m/s)</th>
<th>y_{ICR_r} (m)</th>
<th>y_{ICR_l} (m)</th>
<th>x_{ICR_v} (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>µ  σ</td>
<td>µ  σ</td>
<td>µ  σ</td>
<td>µ  σ</td>
</tr>
<tr>
<td>Circular Path</td>
<td>42.7</td>
<td>0.37 0.08</td>
<td>0.18 0.04</td>
<td>-0.45 0.02</td>
<td>-0.11 0.03</td>
</tr>
<tr>
<td>Curved Path</td>
<td>40.5</td>
<td>0.39 0.12</td>
<td>0.27 0.02</td>
<td>-0.38 0.02</td>
<td>0.04 0.08</td>
</tr>
<tr>
<td>Long Distance</td>
<td>1089.7</td>
<td>0.45 0.15</td>
<td>0.41 0.14</td>
<td>-0.41 0.05</td>
<td>0.14 0.08</td>
</tr>
</tbody>
</table>

Table 3.2. Field test parameters for wheeled vehicle.

<table>
<thead>
<tr>
<th>Test Name</th>
<th>Traversal (m)</th>
<th>Speed (m/s)</th>
<th>y_{ICR_r} (m)</th>
<th>y_{ICR_l} (m)</th>
<th>x_{ICR_v} (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>µ  σ</td>
<td>µ  σ</td>
<td>µ  σ</td>
<td>µ  σ</td>
</tr>
<tr>
<td>Circular Path</td>
<td>63.6</td>
<td>0.47 0.05</td>
<td>0.63 0.02</td>
<td>-0.76 0.03</td>
<td>-0.02 0.06</td>
</tr>
<tr>
<td>Curved Path</td>
<td>30.7</td>
<td>0.40 0.11</td>
<td>0.57 0.04</td>
<td>-0.81 0.05</td>
<td>-0.08 0.06</td>
</tr>
<tr>
<td>Long Distance</td>
<td>983.9</td>
<td>0.44 0.14</td>
<td>0.63 0.02</td>
<td>-0.76 0.03</td>
<td>0.06 0.06</td>
</tr>
</tbody>
</table>

3.5.1 Circular Path

The first test consisted of driving in repeated circles. A plot of the results with the tracked robot is shown in Figure 3.16. The dark gray line on the plot shows the GPS solution of vehicle position. The EKF estimate, shown as a solid black line, lies close enough to the GPS solution that it is difficult to see errors in the EKF estimate. The estimated ICR locations throughout the tracked robot run are shown in Figure 3.17.
Figure 3.16. Plot of GPS position measurement, EKF position estimate, ICR kinematic odometry, and no-slip kinematic odometry for circular run with tracked robot.

Figure 3.16 shows that the EKF follows the GPS solution closely which is expected due to the accuracy of the RTK GPS system and the availability of heading measurements from the AHRS. The mean difference between the GPS and EKF position estimates is 0.7 cm. The ICR open-loop odometry solution forms a perfect circle because of the constant motor command inputs used to drive in a circular path, but the radius of the circle is very accurate in comparison to actual robot movement. The no-slip odometry estimate predicts a circle radius smaller than the actual due to unmodeled slippage that occurs with a skid steer vehicle. The ICR estimates, shown in Figure 3.17, converge to stable values after 7.2 to 12.8 seconds of driving.

The wheeled robot position results, shown in Figure 3.18, are very similar to the tracked vehicle performance. The mean error between GPS and EKF position
estimates is 0.5 cm. The ICR open-loop odometry is again much closer to the actual path than the no-slip odometry estimate. The ICR estimates for the run are given in Figure 3.19 and converge after 2.9 to 7.53 seconds of driving. The ICR locations are significantly different than those of the tracked vehicle, which is expected because of the different drive type and heavier weight of the wheeled robot.

The ICR EKF was initialized with ICR location estimates of $y_{ICRr} = 1.0$ m, $y_{ICRI} = -1.0$ m, $x_{ICRV} = 1.0$ m for both the tracked and wheeled vehicle circular runs. The large amount of variation in the ICR estimates at the beginning of the circular runs, from 0 to 7 seconds in Figure 3.17 and from 0 to 2 seconds in Figure 3.19, appear to be products of the filter learning process. Specifically,
the variations do not appear when the filter is allowed to retain previous ICR information between runs. This is shown in Figure 3.20 where the estimates of ICR location from Figure 3.17 are shown in solid black and estimates from a post-processed ICR EKF are shown in gray dashed lines. The post-processed ICR EKF was initialized with ICR estimates from the end of the previous run, and the estimates do not exhibit the large amount of variation seen when the filter is initialized with poor guesses of ICR location.

Figure 3.18. Plot of GPS position measurement, EKF position estimate, ICR kinematic odometry, and no-slip kinematic odometry for circular run with wheeled robot.
3.5.2 Curved Path

The second test consisted of navigating along a curved grassy path turning left and right, a more realistic driving scenario than the first presented test. The
plot of position information for the tracked robot is shown in Figure 3.21 and the ICR estimates throughout the run are shown in Figure 3.22. The position

![Figure 3.21](image1.png)

**Figure 3.21.** Plot of GPS position measurement, EKF position estimate, ICR kinematic odometry, and no-slip kinematic odometry for driving on curved path with tracked robot.

![Figure 3.22](image2.png)

**Figure 3.22.** Plot of ICR estimates during driving on curved path with tracked robot.

data is similar to the first test in that the EKF matches the GPS closely, the ICR odometry provides a good estimate of movement, and the no-slip odometry is
poor in comparison. Mean and maximum differences between the ICR odometry and ICR EKF position estimates over 40.5 m of travel are 0.58 m and 1.3 m, respectively. The ICR estimates throughout the run are different than those during the circular runs and the estimate of $x_{ICRv}$ varies significantly, possibly due to the more dynamic nature of the run over that of the circular movement.

The position results of the wheeled robot traveling on a curved path are shown in Figure 3.23 and also show similar performance to the circular run. The ICR odometry provides a good estimate of vehicle motion, and the mean and maximum errors over 30.7 m of travel are 0.78 m and 1.49 m. Figure 3.24 shows the ICR estimates for the wheeled robot on a curved path. The estimates of $y_{ICRr}$ and $y_{ICRl}$ are similar to the estimates for the circular path test. As with the tracked vehicle, the estimate of $x_{ICRv}$ varies significantly, again suggesting that the longitudinal location of the ICRs depends more on the dynamics than the lateral position.

### 3.5.3 Long Distance Traversal

The third segment of testing consisted of driving each robot in a lawnmower search pattern for approximately 1000 m of travel as shown for the tracked robot in Figure 3.25 and the wheeled robot in Figure 3.27. The purpose of this test was to investigate filter performance over a longer time of operation. After convergence, the ICR location estimates remain fairly stable throughout the run for each vehicle. This is illustrated in Figure 3.26 for the tracked robot and Figure 3.28 for the
Figure 3.23. Plot of GPS position measurement, EKF position estimate, ICR kinematic odometry, and no-slip kinematic odometry for driving on curved path with wheeled robot.

Figure 3.24. Plot of ICR estimates during driving on curved path with wheeled robot. The mean error between the ICR EKF and the GPS position solution was 1.8 cm for the tracked robot and 1.6 cm for the wheeled robot. As expected, the $y_{ICR}$ values for the long distance
run lie outside the robot track/wheel locations, which in local coordinates are \( \pm 0.26 \) m for the tracked robot and \( \pm 0.31 \) m for the wheeled robot. More slippage is expected to occur with the wheeled robot as the distance between the wheels and ICR locations is larger than the analogous distance on the tracked robot. The
motors and batteries are mounted forward of the longitudinal center of the tracked robot, which shifts the center of gravity forward and accounts for the positive estimated $x_{ICRv}$ location. The wheeled robot has a very even weight distribution due to it consisting of two identical systems operating together. This is reflected
in the results by the estimated $x_{ICR}$, staying near 0 m.

### 3.5.4 Testing ICR Methods for Robot Localization During GPS Outages

Because ICR position estimates are quite accurate, they can be used to localize ground robots in situations where the GPS, INS, or odometry systems may have outages or errors. Poor GPS solutions and outages are caused by a number of factors including multi-path signals and sky occlusions. Here a situation is tested where the ICR EKF is run alongside a traditional GPS based position estimate: when GPS (or similar position information) is available, this algorithm updates ICR location estimates. When GPS is unavailable, the ICR kinematic equations are used to propagate movement with the most recently learned ICR locations.

The results of this system are shown for the tracked robot in Figure 3.29 where three GPS dropouts of twenty second duration are forced to occur. The ICR kinematic equations are used to predict the movement of the robot throughout the GPS dropout and are accurate to within 0.4 m in the first and third dropout and within 0.65 m in the second dropout. The accuracy of the ICR odometry is heavily influenced by the movement of the robot. Higher numbers of turns and also tighter turns will result in more divergence in the position estimate.

The performance of the open-loop ICR odometry during GPS dropouts was further investigated using the lawnmower search pattern data presented in Fig-
Figure 3.29. Test of ICR odometry as backup system during GPS dropouts with tracked robot.

Using each recorded state as a starting location, the position of the robot was estimated forward in time for a set distance. The ICR locations estimated by the on-board filter at the starting point were used for the odometry. The resulting position was compared with the EKF estimate that utilized GPS and heading measurements for estimation. This analysis was done for traversal distances of 1-20 m for both the tracked and wheeled robots.

Examples of the results for individual traversal distances are shown in Figures 3.30-3.32 for 5 m, 10 m, and 20 m GPS dropouts on the tracked vehicle. The result of the 5 m dropout analysis indicates that the ICR odometry is within 1.6 m for 95% of the trials and that the expected error value based on a weighted average of the histogram distribution is 0.89 m. For the 10 m GPS dropout, the odometry estimate is within 3.8 m for 95% of the trials and has an expected error value of
1.6 m. The odometry estimate is within 10 m for 95% of the trials and has an expected error value of 3.5 m with a 20 m GPS dropout.

The results for all GPS dropout lengths are summarized for the tracked robot in Figure 3.33 and the wheeled robot in Figure 3.34. The performance of the ICR and no-slip odometry is similar up to around 10 m for the tracked robot, but at longer dropout lengths the ICR outperforms the no-slip odometry. With the wheeled vehicle, the ICR odometry estimate is significantly better than the no-slip odometry for nearly all tested values of dropout length. In fact, for all dropout lengths greater than 6 m, both of the recorded ICR error values lie below the expected value of error for the no-slip odometry, indicating that the ICR algorithm performs much better than the no-slip on the wheeled robot. The improvement of the ICR odometry over the no-slip between the tracked and wheeled vehicle is

![Figure 3.30. Histogram of open-loop ICR odometry error on tracked robot after 5 m of travel from each point traversed in lawnmower search pattern.](image)

- Expected Value of Error 0.89 m
- 95% of Occurrences Below 1.6 m
most likely due to increased slip with the wheeled robot. While the tracked and wheeled robots are similar in width, the estimated $y_{ICR_t}$ and $y_{ICR_w}$ locations for the wheeled robot are much larger than those for the tracked robot, indicating increased slip with the wheeled vehicle.
The no-slip odometry trajectories in Figures 3.21 and 3.23 indicate that larger errors would be expected for no-slip odometry than appear in Figures 3.33 and 3.34. However, the shorter runs in Figures 3.21 and 3.23 consist of many more turns at higher frequencies than the lawnmower pattern used in the GPS dropout analysis.
During cornering maneuvers, the ICR odometry is much better than no-slip, as shown in Figure 3.35, but during straight sections the two odometry methods are very similar, as shown in Figure 3.36. For shorter GPS dropout lengths, the lawnmower pattern appears mainly as a series of short straight sections with few curves, allowing the no-slip to perform comparably to the ICR with the tracked vehicle. For longer dropouts the corners become more apparent, and the ICR begins to outperform the no-slip odometry. For the wheeled robot, the increased amount of slippage causes the no-slip odometry to become very inaccurate with only small turns, which accounts for the large improvement seen in Figure 3.34 with the ICR over the no-slip odometry. In summary, a skid-steer vehicle traveling in straight lines would not benefit significantly from ICR odometry, but improvements to motion estimation will be seen when using ICR odometry with a robot operating on any path requiring it to turn.
3.5.5 Testing with Conventional GPS Solution

The previous work utilized an RTK GPS system for position measurement. This is a specialized system that requires local corrections that may not be available in all situations. Three laps of the long distance course presented in Section 3.5.3 were
repeated using the tracked vehicle with the local GPS correction station turned off. Thus, the GPS system utilized only the less-accurate satellite based corrections from the Wide Area Augmentation System (WAAS). The GPS manufacturer provides an estimate of 0.60 m RMS horizontal position accuracy when using WAAS, although the actual accuracy could be worse depending on the number of satellites in view, satellite geometries, multipath errors, and ionospheric activity. The resulting position plot of GPS measured and estimated position is shown in Figure 3.37, and the estimated ICR locations throughout the run are summarized in Figure 3.38. The mean velocity of the robot was 0.7 m/s with standard deviation 0.15 m/s. The ICR location estimates have means and standard deviations of $\bar{y}_{ICR_e} = 0.41$ m, $\sigma_{ICR_e} = 0.08$ m; $\bar{y}_{ICR_l} = -0.57$ m, $\sigma_{ICR_l} = 0.04$ m; and $\bar{x}_{ICR_e} = 0.3$ m, $\sigma_{ICR_e} = 0.06$ m. These values of ICR locations are slightly different

![Figure 3.37. Plot of measured GPS position and EKF position estimate when operating without RTK GPS corrections.](image-url)
than those found in the long distance traversal section. This is most likely due to changing terrain properties: the long distance traversal tests of Section 3.5.3 were performed in late spring when the grass on the surface was three to four inches tall and very lush, while the testing without RTK corrections was performed in early fall when grass was 1-2 inches tall and very dry.

The GPS outage analysis of Section 3.5.4 was repeated on the traversal data without RTK corrections. The results, shown in Figure 3.39, indicate that with a standard GPS the estimated ICR locations still provide improved odometry over a no-slip estimate. The improvement is greater than that of the long distance traversal test shown in Figure 3.33, but more detailed comparisons may be difficult due to differences in terrain conditions and average robot speed. Overall, the ICR EKF provides a significant improvement over no-slip odometry without a high-
accuracy GPS system.

![Graph showing open-loop ICR odometry error](https://via.placeholder.com/150)

**Figure 3.39.** Summary of open-loop ICR odometry error on tracked robot for 1-20 m of travel from each point traversed in lawnmower search pattern. ICR locations for odometry were estimated by ICR EKF without RTK corrections to GPS solution.

### 3.5.6 Speed and Terrain Variability Testing

To investigate the performance of the ICR EKF at higher speeds and a different terrain, the wheeled robot was driven along curved paths on asphalt at speeds of 0.4 m/s, 1.6 m/s, and 4.0 m/s. The mean and standard deviation of the estimated ICR locations for each speed are summarized in Table 3.3. The results of driving the wheeled robot at 0.4 m/s, shown in Figure 3.40 and Figure 3.41, show performance of the ICR EKF similar to previous testing on grass. The open-loop ICR odometry, which is calculated without measurements of position or heading, is quite close to the actual trajectory of the robot.

The results for operating the wheeled robot at a speed of 1.6 m/s are shown in
Figure 3.40. Plot of GPS position measurement, EKF position estimate, ICR kinematic odometry, and no-slip kinematic odometry when driving the wheeled robot on asphalt at 0.4 m/s.

Figure 3.41. Clustering of ICR location estimates when driving the wheeled robot on asphalt at 0.4 m/s.

Figure 3.42 and Figure 3.43. The open-loop ICR odometry matches the shape of the actual vehicle trajectory very well for the first 30 m of travel, but subsequently drifts away from the actual path. However, the shape of the trajectory remains
Table 3.3. Results of using ICR EKF algorithm at increasing speeds.

<table>
<thead>
<tr>
<th>Test Speed (m/s)</th>
<th>$y_{ICRv}$ (m)</th>
<th>$y_{ICRL}$ (m)</th>
<th>$x_{ICRv}$ (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\mu$</td>
<td>$\sigma$</td>
<td>$\mu$</td>
</tr>
<tr>
<td>0.4</td>
<td>0.65</td>
<td>0.01</td>
<td>-0.64</td>
</tr>
<tr>
<td>1.6</td>
<td>0.49</td>
<td>0.02</td>
<td>-0.56</td>
</tr>
<tr>
<td>4.0</td>
<td>0.49</td>
<td>0.01</td>
<td>-0.61</td>
</tr>
</tbody>
</table>

Figure 3.42. Plot of GPS position measurement, EKF position estimate, ICR kinematic odometry, and no-slip kinematic odometry when driving the wheeled robot on asphalt at 1.6 m/s.

The results of the final high speed run, conducted at 4.0 m/s, are shown in Figure 3.44 and Figure 3.45. The open-loop ICR odometry again matches the vehicle trajectory initially but then drifts away from the true path. As before, the shape of the ICR odometry is very similar to that of the recorded GPS.

Comparing Figures 3.40, 3.42, and 3.44, the variance in the estimate of $x_{ICRv}$ increases as the vehicle speed increases. However, the estimates of $y_{ICRv}$ and $y_{ICRL}$
Figure 3.43. Clustering of ICR location estimates when driving the wheeled robot on asphalt at 1.6 m/s.

Figure 3.44. Plot of GPS position measurement, EKF position estimate, ICR kinematic odometry, and no-slip kinematic odometry when driving the wheeled robot on asphalt at 4.0 m/s.

remain tightly clustered around similar values for all three runs. Increased variance in $x_{ICRv}$ occurs at larger lateral velocities caused by higher speeds. In these situations, larger track/tire forces are required due to dynamics not modeled in
the ICR EKF. The slip angle of a moving vehicle, which relates actual direction of travel to the direction it is pointing for skid-steer vehicles, is for small slip values proportional to the track/tire force [63]. A tighter turn at high speed will have a larger slip angle, and therefore larger lateral velocity, which is related to $x_{ICRv}$ as shown in (3.10). At higher speeds, the assumption that the $x_{ICRv}$ location is constant in the ICR EKF model breaks down. A comparison of the absolute difference in the ICR EKF North-East position estimate and the measurement of vehicle position provided by the GPS is shown in Figure 3.46. At low speeds, the difference is within the accuracy of the RTK GPS, and, as expected due to the difficulty in estimating ICR locations, the error increases at higher speeds. It was quite difficult to manually operate the wheeled robot such that it followed a curved path at 4.0 m/s. The turning ability of the vehicle is limited at such speeds, so

Figure 3.45. Clustering of ICR location estimates when driving the wheeled robot on asphalt at 4.0 m/s.
Figure 3.46. Comparison of difference in ICR EKF position estimate and measurement from RTK GPS system.

avoiding obstacles in the test area becomes a significant issue. The tracked vehicle has a maximum speed of approximately 1 m/s in its current configuration, so no additional high speed testing was performed with it.

Future work in speed variability testing will focus in the area of utilizing the ICR kinematic model for trajectory control of skid-steer vehicles. This will enable repeated traversal of set paths at varying speeds, which is currently difficult to do at higher speeds. Direct comparison of ICR EKF performance on the same path type will then be possible.

3.6 Alternate Methods of ICR Estimation

While the EKF is the chosen method of ICR estimation for this work and has received the most attention and field testing, two other possible methods of ICR
identification were also explored and are described here. The first method is presented by Martinez [28] and utilizes a parameter called the steering efficiency index, \( \chi \), of a vehicle. This is defined as

\[
\chi = \frac{b}{y_{ICRr} - y_{ICRl}} \tag{3.45}
\]

where \( b \) is the width between the tracks or wheels. If the ICR locations lie along the track or wheel centerlines, than the steering efficiency will be \( \chi = 1 \). This would mean the skid-steer vehicle drives exactly like a two wheel or unicycle type robot and no slippage occurs. If one assumes symmetric ICR locations such that \( y_{ICRr} = -y_{ICRl} \) and \( x_{ICRr} = 0 \), another method of calculating the steering efficiency is [39]

\[
\chi = \frac{b\phi}{\int V_r dt - \int V_l dt} \tag{3.46}
\]

where \( \phi \) is the total rotated angle after equal and opposite speed inputs \( V_r \) and \( V_l \) have been applied. This experiment is easily carried out with a skid-steer vehicle, and once done, the steering efficiency \( \chi \) can be calculated from (3.46). Then, the unknown ICR location can be calculated from (3.45). This algorithm was tested using recorded data from rotating the tracked robot three times with a resulting ICR \( Y \) coordinate of 0.29 m. This value agrees well with the EKF results presented in Table 3.1. The main disadvantage of this method of ICR identification is the assumption of symmetric ICR locations. Field testing with the ICR EKF has
shown that ICR locations are rarely symmetrical on actual robots.

The second method of ICR identification comes from the structure of the kinematic equations themselves. The equation for angular rate, (3.11), can be rearranged as

\[ y_{ICRl} - y_{ICRr} = \frac{V_x^l - V_x^r}{\omega_z}. \tag{3.47} \]

The input speeds, \( V_x^l \) and \( V_x^r \), and the vehicle angular rate, \( \omega_z \) can be measured on the vehicle allowing the ICR difference to be calculated. The kinematic equations for longitudinal and lateral velocities now form linear relationships with the unknown ICR locations as

\[ v_x(y_{ICRl} - y_{ICRr}) = V_x^r y_{ICRl} - V_x^l y_{ICRr} \tag{3.48} \]

\[ v_y(y_{ICRl} - y_{ICRr}) = (V_x^l - V_x^r) x_{ICRr}. \tag{3.49} \]

The measurements required are the longitudinal \( v_x \) and lateral \( v_y \) velocities. Unfortunately, it is extremely difficult to accurately measure these parameters on a ground robot. The GPS does provide a measurement of vehicle velocity magnitude, but it is not nearly as accurate as the measurement of position. So that the most accurate sensors on board the vehicle could be utilized, the ICR EKF was implemented.
3.7 Conclusion

This chapter presents an EKF algorithm capable of learning the ICR locations of a skid-steer ground robot during operation. An ICR based kinematic model of skid-steer movement using track velocity inputs is used to model the system, and measurements of position and heading are used to update the state and enable learning of ICR locations. The algorithm was first developed in simulation where it was found that modification of the ICR kinematics to ensure correct sign is necessary for filter convergence in the presence of noise. Further investigation showed that the filter is more sensitive to noise than bias on the track velocity inputs.

The main contribution of this chapter is an algorithm which learns ICR locations relating input track or wheel speed to skid-steer vehicle motion without prior knowledge of terrain type or vehicle configuration. ICR locations remain relatively constant for skid-steer vehicle operation at low speeds on hard, flat terrain regardless of vehicle motion. This allows estimated ICR locations to be used for odometry or to map desired vehicle motion, such as forward speed and angular rate, to required track or wheel speeds. ICR locations have also been successfully used to model power usage in skid-steer wheeled and tracked vehicles [32] and will be integral in the power model presented in Chapter 5. Estimation of ICR positions during operation enables identification and prediction of skid-steer power usage without prior knowledge of terrain properties.
The algorithm has undergone substantial field testing on the Tankbot and RMP400. Field test results show that ICR locations estimated by the ICR EKF algorithm provide good predictions of robot movement and that assuming no-slip when performing odometry with skid-steer vehicles gives comparatively poor position estimates. ICR estimates generally converge within 5-15 seconds of operating the vehicle depending on the type of movement the robot performs. Overall, the performance of the ICR EKF was quite good considering no prior knowledge of robot design or terrain parameters were needed. Over a 40.5 m meter traversal, ICR odometry using learned ICR locations predicted positions with a mean error of 0.58 m with the tracked robot when using the most recent ICR estimates at each time step. The ICR EKF algorithm was tested with both robots on separate 1000 m runs and provided estimates of position within 2 cm of GPS on average. This accuracy is expected and not a major contribution from the filter because the ICR EKF receives GPS measurements at 20 Hz. Rather, the real-time estimates of ICR location and their use in accurate odometry and real-time estimation of wheel slip in the case of GPS dropouts are the main contributions of the algorithm. ICR estimates during the long distance test cluster around values of \( y_{ICRr} = 0.38 \, \text{m}, \)
\( y_{ICRl} = -0.42 \, \text{m}, \) and \( x_{ICRr} = 0.16 \, \text{m} \) for the tracked robot and \( y_{ICRr} = 0.64 \, \text{m}, \)
\( y_{ICRl} = -0.76 \, \text{m}, \) and \( x_{ICRr} = -0.04 \, \text{m} \) for the wheeled robot. The vehicles are similar in width, so the larger \( y_{ICR} \) values for the wheeled robot indicate larger amounts of slip during operation than with the tracked robot. The longitudinal
location, $x_{ICR_e}$, seems to vary more than the lateral locations for both robots and remained constant during testing only for the circular run where the dynamics of the vehicle are fairly constant. Further testing of the wheeled robot at speeds of 1.6 m/s and 4.0 m/s showed increasing variance in the estimate of $x_{ICR_e}$ for increasing vehicle speeds. The increased variance is thought to be caused by increased lateral slippage due to unmodeled dynamics at higher speeds.

There are many applications for continued work in this area. An obvious one demonstrated in this chapter is that of a backup to GPS positioning. Loss of GPS signal is a common occurrence in ground robotics, and the ICR EKF is well suited to learn parameters in the background and step in as needed to estimate movement in the absence of GPS. Dropouts of the GPS position solution for various lengths of vehicle traversal were simulated using recorded data. Results indicated that ICR odometry on the tracked robot is accurate to within 1.6 m for a 5 m GPS dropout, 3.8 m for 10 m GPS dropout, and 10 m for 20 m GPS dropout for 95% of trials. This analysis was extended to the wheeled robot and GPS dropout lengths from 1 to 20 m. The results showed that vehicles with larger amounts of slippage will obtain more benefit from ICR odometry than no-slip odometry. For example, the wheeled robot in this work slipped significantly more than the tracked robot and therefore had larger amounts of improvement with ICR odometry. The ICR estimates produced open-loop prediction errors on the wheeled robot that are roughly 18% of the distance traveled, versus 45% for the no-slip model. On
the tracked robot, ICR odometry produced open-loop prediction errors of roughly 17% of the distance traveled, versus 22% for the no-slip model. For both robots there is a significant improvement in the accuracy of kinematic prediction with ICR odometry compared to using the same measurements with a no-slip model. Experimental results show these improvements will be most pronounced on paths with large numbers of curves.

Previous work in odometry for skid-steer vehicles have shown estimation accuracies of 2% of distance traveled or better [46, 67], but such algorithms relied on empirically determined constants related to terrain and/or vehicle configuration and high cost sensors not typically found on ground mobile robots. The algorithm presented in this work provides less accurate odometry predictions, but is easily adapted to tracked or wheeled skid-steer vehicles without extensive vehicle modeling or calibration maneuvers.

The ICR estimates produced by the algorithm presented in this chapter will be used for both trajectory control and power usage estimation for skid-steer vehicles.
4.1 Introduction

Skid-steer vehicles are difficult to control due to the significant slippage that occurs whenever a skid-steer vehicle turns. The amount of slippage is dependent on vehicle configuration and the surface of operation. To model skid-steer movement using dynamic models, surface properties must be characterized through standard measurement techniques or identification algorithms. This can be time consuming when multiple vehicles and surface types must be modeled. In Chapter 3, ICR kinematics are shown to produce accurate predictions of skid-steer motion using
parameters identified during vehicle operation. This chapter will show that the locations of the ICRs provide a mapping between skid-steer and two-wheel robot movement. Because two-wheel robot control has been studied so extensively, it is literally a textbook control problem [8, 12, 17]. Thus, by mapping skid-steer movement to that of a two-wheel vehicle, previous work in unicycle robot control can be leveraged.

Unicycle, or Hilare type, robots utilize two independently driven wheels and most often a free-wheel caster for stability. These types of vehicles experience practically no slip during operation [17]. The absence of slip allows the motion of these vehicles to be accurately modeled through kinematic equations involving wheel speeds and drive wheel spacing [17]. As one might expect, the control of a unicycle type robot is much simpler than that of a skid-steer vehicle due to the ease and accuracy of modeling unicycle robot motion.

The fusion of ICR kinematics and unicycle robot control presented in this chapter is beneficial in that no prior knowledge of surface properties is required because the ICR locations describing vehicle movement are estimated during operation. The following sections will describe an example two-wheel robot trajectory control algorithm, show how the ICR kinematic mapping allows this control algorithm to operate on skid-steer robots, and finally give simulation and experimental results of controller implementation.
4.2 Unicycle Robot Trajectory Control

A unicycle robot operating in a plane, shown in Figure 4.1, can be modeled using

\[
\begin{bmatrix}
\dot{X} \\
\dot{Y} \\
\dot{\psi}
\end{bmatrix} =
\begin{bmatrix}
U \cos \psi \\
U \sin \psi \\
\omega
\end{bmatrix}
\]  

(4.1)

where \( \dot{X} \) is the vehicle speed in the \( X \) direction, \( \dot{Y} \) is the vehicle speed in the \( Y \) direction, \( \dot{\psi} \) is the rate of change in the heading of the vehicle, \( U \) is the vehicle forward velocity, and \( \omega \) is the angular rate of the vehicle. If no slippage occurs between the wheels and the ground, the forward and angular velocities of the vehicle are

\[
U = \frac{V_r + V_l}{2}
\]  

(4.2)

\[
\begin{array}{c}
\text{Figure 4.1. Two wheel robot following planar path } C. \\
\end{array}
\]
\[ \omega = \frac{V_l - V_r}{b} \]  \hspace{1cm} (4.3)

where \( V_r \) and \( V_l \) are the right and left wheel velocities, respectively, and \( b \) is the distance between the wheels.

Figure 4.1 shows a unicycle robot and a desired path of traversal, \( C \). The center of the vehicle, defined to be midway between the wheels on the axle, is labeled as point \( P \), and the orthogonal projection from the robot position to the path is labeled as point \( P' \). The task of path following involves determining a controller such that the distance to the path, \( d \), and the heading error, \( \tilde{\psi} = \psi - \psi_s \), converge to zero. The design of such a controller is aided by expressing the kinematics in the Serret-Frenet frame \((X_s, Y_s)\) [43]. Taking the derivative of the heading error with respect to time yields

\[ \dot{\tilde{\psi}} = \dot{\psi} - \psi_s = \psi - \dot{s} \mathcal{K}(s) \]  \hspace{1cm} (4.4)

where \( \dot{s} \) is the time rate of change of the curvilinear distance from the start to point \( P' \) and \( \mathcal{K}(s) \) is the curvature of the path at point \( P' \). The relationships between velocities in the \((X, Y)\) frame and in the \((X_s, Y_s)\) frame are given by [43]

\[ \dot{d} = \begin{bmatrix} -\sin \psi_s & \cos \psi_s \end{bmatrix} \begin{bmatrix} \dot{X} \\ \dot{Y} \end{bmatrix} \]  \hspace{1cm} (4.5)
and

\[ \dot{s} = \frac{1}{1 - \mathcal{K}(s)d} \begin{bmatrix} \cos \psi_s & \sin \psi_s \end{bmatrix} ^T \begin{bmatrix} \dot{X} \\ \dot{Y} \end{bmatrix}. \]  \tag{4.6}

Expanding each equation, substituting in the relations for \( \dot{X} \) and \( \dot{Y} \) in (4.1), and utilizing trigonometric angle difference identities yields the following kinematic equations:

\[
\dot{s} = U \frac{\cos \tilde{\psi}}{1 - \mathcal{K}(s)d} \]  \tag{4.7}
\[
\dot{d} = U \sin \tilde{\psi} \]  \tag{4.8}
\[
\dot{\tilde{\psi}} = \omega - U \cos \tilde{\psi} \frac{\mathcal{K}(s)}{1 - \mathcal{K}(s)d}. \]  \tag{4.9}

These kinematic equations are commonly used in the development of trajectory control strategies for unicycle robots.

De Wit et al. present the following nonlinear controller for path following with unicycle robots [12]. Introducing a change of control variable

\[
q = \omega - \frac{v \cos \tilde{\psi} \mathcal{K}(s)}{1 - \mathcal{K}(s)d}, \]  \tag{4.10}

the kinematic equations (4.7)-(4.9) are rewritten as

\[
\dot{s} = U \frac{\cos \tilde{\psi}}{1 - \mathcal{K}(s)d} \]  \tag{4.11}
\[
\dot{d} = U \sin \tilde{\psi} \]  \tag{4.12}
\[ \dot{\psi} = q. \] (4.13)

Asymptotic stability can be obtained by setting

\[ U = u(t) \]

\[ q = -k_1 Ud \frac{\sin \psi_e}{\psi_e} - k_2 |U| \psi_e. \] (4.14)

with proof given in [12]. The function \( u(t) \) must be defined such that

\[ \lim_{t \to \infty} u(t) \neq 0 \] (4.15)

and \( k_1 \) and \( k_2 \) are greater than zero. In this work, the forward velocity of the vehicle is assumed to be constant. The control law (4.14) guarantees convergence only when the distance to the path can be computed uniquely, such as in the case of a straight line [34]. For an arbitrary curve, the normal projection may not be unique for all positions, so global convergence is not assured. To ensure a unique projection for all positions, the robot is initialized near the start of the path and only a portion of the path near the last station position is searched for the orthogonal projection. This allows for self-intersecting paths.

Equations (4.10) and (4.14) provide a value of vehicle angular rate to converge the vehicle to the path. The wheel speeds necessary to produce the desired \( \omega \) with a unicycle robot are easily calculated from (4.2) and (4.3). The difficulty in
controlling skid-steer vehicles arises from the fact that the mapping from track or wheel speeds to vehicle motion is much more complex than that of a unicycle robot due to the slippage necessary for skid-steer movement. The ICR kinematics presented in Chapter 3 provide the wheel or track speeds necessary to produce the desired vehicle angular rate and speed. The control law (4.14) cannot be directly applied to a skid-steer robot due to the lateral velocity expressed in (3.10). The lateral velocity is nonzero when the left and right track or wheel speeds are not equal and when the longitudinal ICR coordinate \( x_{iCR} \) is nonzero.

The key contribution of this research is the use of ICR kinematics to develop a mapping from skid-steer motion to unicycle robot motion. A unicycle robot is assumed to have zero lateral velocity when the no-slip assumption is applied. Therefore, it is necessary to determine a point on the robot with zero lateral velocity. The velocity of the origin of the body fixed frame \((X, Y)\) is known from the ICR kinematics. The velocity \( V_p \) of a point \( P \) chosen in this frame is found through

\[
V_p = v_x \hat{e}_x + v_y \hat{e}_y + \omega \hat{e}_z \times r_p
\]  

(4.16)

where \( r_p \) is the vector from the origin of \((X, Y)\) to \( P \). The lateral velocity of point \( P \), \( v_{p,y} \), is therefore

\[
v_{p,y} = v_y + \omega r_{p,x}.
\]  

(4.17)
Substituting the ICR kinematic equations (3.10) and (3.11) for $v_y$ and $\omega$ yields

$$v_{p,y} = \frac{(V^l_x - V^r_x)x_{ICRv}}{y_{ICRl} - y_{ICRr}} + r_{p,x} \left[ \frac{-V^l_x - V^r_x}{y_{ICRl} - y_{ICRr}} \right]. \tag{4.18}$$

It is apparent that any point with an $X$ coordinate equal to $x_{ICRv}$ will have zero lateral velocity regardless of vehicle movement. This is identical to the drive-wheel axis of a unicycle robot. Furthermore, the angular velocity of the robot can be written as

$$\omega = \frac{V^l_x - V^r_x}{y_{ICRl} - y_{ICRr}} = \frac{V^l_x - V^r_x}{b} \tag{4.19}$$

where $b$ is the distance between ICR locations. Equation (4.19) is identical to (4.3), thus, the movement of a skid-steer vehicle with wheel velocities $V_r$ and $V_l$ is identical to the movement of a unicycle robot with wheels located at the right and left ICR locations and turning at speeds $V_r$ and $V_l$. This transformation exists because the ICR locations represent the point where the relative track and ground velocities are zero. Point $P$ is chosen to be at the center of the equivalent unicycle robot axle, given by the vector $\mathbf{q}$ as

$$\mathbf{q} = \begin{bmatrix} q_x \\ q_y \end{bmatrix} = \begin{bmatrix} x_{ICRv} \\ \frac{y_{ICRr} - y_{ICRl}}{2} \end{bmatrix}. \tag{4.20}$$

and shown in Figure 4.2. Because the center of the axle for a unicycle robot analogous to a skid-steer robot is located at $P$, an augmented path $C'$ must be
Figure 4.2. Example of point $P$ with zero lateral velocity at center of left and right ICR locations.

created such that the point $P$ tracks the augmented path and the center of the skid-steer robot tracks the original path. The augmented path is created by translating the original path by the vector $q$ as shown in Figure 4.3. In this work, the path is a collection of recorded GPS positions, so each point is translated by $q$ within the path frame $(X_s, Y_s)$ at that point.

The most critical issue with implementing this trajectory control algorithm on a vehicle is ensuring that the robot can achieve the desired angular rate in a reasonable time. With a true unicycle robot there is no wheel slip. Because of this, unicycle robots are able to change angular rate extremely quickly. Skid-steer

Figure 4.3. Augmentation of path points to create path $C'$. 
vehicles, however, experience large frictional sliding forces during turns making rapid adjustment of vehicle angular rate difficult to achieve. The majority of field testing for this research was conducted near 0.5 m/s forward speed. At this speed, the controller has operated reliably on both test vehicles.

4.3 Simulation Implementation

The control law (4.14) was implemented in simulation on a skid-steer robot with characteristics representative of that of the wheeled experimental test vehicle, provided in Table 4.1. The control gains $k_1$ and $k_2$ were selected based on [12] such that

$$k_1 = a^2$$  \hspace{1cm} (4.21)

$$k_2 = 2\zeta a$$  \hspace{1cm} (4.22)

<table>
<thead>
<tr>
<th>Table 4.1. Trajectory control simulation parameters.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
</tr>
<tr>
<td>Time Step</td>
</tr>
<tr>
<td>$u(t)$</td>
</tr>
<tr>
<td>$y_{ICRr}$</td>
</tr>
<tr>
<td>$y_{ICLl}$</td>
</tr>
<tr>
<td>$x_{ICRv}$</td>
</tr>
<tr>
<td>$k_1$</td>
</tr>
<tr>
<td>$k_2$</td>
</tr>
</tbody>
</table>
where \( a \) is a transient ‘rise distance’ and \( \xi \) is the damping coefficient, chosen to be the critical value \( 1/\sqrt{2} \) for this work. A value of \( a = 2 \) was used for both simulation and field testing.

A circular path of radius 50 m with center at \((N, E) = (0, 0)\) was generated for initial simulation testing. The results of simulating the controller along the desired trajectory are shown in Figures 4.4 and 4.5. The vehicle position was initialized at \((N, E) = (50, 20)\) with heading \( \psi = 200 \) degrees. The distance and heading errors converge to zero quickly, as shown in Figure 4.5, because the controller has perfect knowledge of the vehicle parameters and control authority with zero delay.

As mentioned previously, one major weakness of this control scheme is the assumption that the vehicle is able to attain the desired angular velocity instantaneously. To test this, the actual angular velocity of the vehicle was delayed from
the commanded value in the simulation. Increasing the delay at 0.1 second increments, it was found that at 0.5 seconds of delay and above, the controller fails to guide the vehicle along the path. The path and heading error with 0.4 seconds of delay are shown in Figure 4.6. The delay in angular rate command causes significant oscillations of the vehicle, reflected by the oscillations seen in the heading error values. Additional simulations found that the allowable delay increased with slower forward speeds. When operating on the terrains tested for this thesis, both vehicles were able to attain the required angular rate well within the required time. For vehicles operating on extremely soft terrains, such as sand, a different control strategy would most likely be required.

For operation on the vehicle, an allowable path consistent with the terrain and obstacles in the area must be created. This was done by manually driving a robot along a path in the testing area and recording the output of the ICR EKF. This
Figure 4.6. Heading error (left vertical axis) and distance error (right vertical axis) for circular path simulation run with 0.4 second delay between controller angular rate output and implementation.

output was post-processed to create a map consisting of north position, east position, vertical position, path heading, path station, and path pitch. The ICR EKF output was utilized because both the GPS measurements, compass measurements, and speed inputs are fused together producing a smoother estimate of traversal than the individual measurements. One of the field testing paths was also used for simulating the trajectory control algorithm and is shown in Figure 4.7. The position trace is overlaid on a satellite image of the testing area.

A comparison of the experimental path and simulated vehicle trajectory is shown in Figure 4.8. The distance error and heading error relative to the path are shown in Figure 4.9. The mean absolute value of the distance error was 0.057 m, and the mean absolute value of the heading error was 2.8 degrees. The commanded right and left wheel speeds, shown in Figure 4.10, are within the capabilities of
both the test platforms. Distance and heading errors calculated when simulating the experimental path are higher than when simulating the circular path. This increase is due to variations in the experimental data used to form the path.

The simulation shows that the control law (4.14) is capable of controlling a
Figure 4.9. Simulated distance error with forward speed of 0.4 m/s.

Figure 4.10. Right and left track speed commands produced by controller during simulation.

skid-steer vehicle through the kinematic ICR mapping, and the required wheel speeds shown in Figure 4.10 are well within the capabilities of the test vehicle. However, the stability of (4.14) is only ensured if the vehicle can be controlled to match the forward speed $U$ and angular rate $\omega$ produced by the controller. The ICR kinematic equations provide the wheel speeds needed to match $U$ and $\omega$, but
at higher speeds the dynamics of the vehicle and the terrain will affect how quickly a set point $U$ and $\omega$ will be reached. Also, it is possible that the kinematic mapping may cause control saturation for some ICR locations. Thus, field testing of the controller on the robot is required to assess the feasibility of this control mapping.

4.4 Experimental Results

The control algorithm was implemented on the wheeled skid-steer vehicle and the tracked skid-steer vehicle with a desired constant forward speed of 0.4 m/s and controller gain values of $k_1 = 2.0 \text{ m}^{-1}$ and $k_2 = 2.83 \text{ m}^{-1}$, values identical to the simulation parameters. The path trajectory was the same as that used in simulation, and the resulting trajectory error values for the tracked vehicle are given in Figure 4.11. As shown in Figure 4.7, the vehicle started on a grass surface and transitioned to asphalt at approximately $t = 250$ seconds into the test. Over four repeated runs of the course by the tracked vehicle, the mean absolute distance error was 0.056 m and the mean absolute heading error was 4.3 degrees. During the initial grass portion of the test run, shown up to time $t = 250$ seconds in Figure 4.11, the distance and heading error exhibit larger periodic variations than during the asphalt portion of the test. While operating on grass, the tracked vehicle must overcome much larger forces during turning than when operating on asphalt. The inability of the tracked vehicle to achieve the commanded angular rate results in an oscillation around the path while operating on grass.
The wheeled robot is much larger than the tracked vehicle and is less affected by the change from grass to asphalt terrain as shown in Figure 4.12. There is no significant difference in the distance or heading error between the grass portion, $t = 0$ s to $t = 430$ s, and the asphalt portion, $t = 430$ s to $t = 974$ s. Combining data from four traversals of the course with the wheeled vehicle, the mean absolute distance error was 0.033 m and the mean absolute heading error was 2.7 degrees.
Figure 4.12. Distance error and heading error during experimental run with wheeled robot.

4.5 Conclusions

This chapter presented a method of applying unicycle robot control methods to skid-steer vehicles through the ICR kinematic mapping. The ICR kinematic equations provide a method of calculating the required track or wheel speeds to produce desired vehicle motion if the ICR locations are known. The ICR locations are learned through the EKF presented in Chapter 3, and this learning process enables the controller presented in this work to be implemented without prior knowledge of vehicle size or configuration.

The controller successfully navigated a skid-steer vehicle along a specified path in both simulation and experimental testing. One possible source of error when using this control scheme is the inability of the vehicle to attain the commanded angular rate. This behavior occurred with the tracked vehicle which is not powerful
enough to turn on grass surfaces quickly enough. Overall, the controller performed very well, controlling the tracked vehicle within 6 cm of the desired path and the wheeled vehicle within 4 cm of the desired path. This is impressive performance for implementing an off-the-shelf controller on a skid-steer vehicle with no prior knowledge of vehicle configuration or operational surface type. While not a direct focus of this thesis, the development of the trajectory controller was necessary so that testing runs can be reliably repeated and to simulate vehicle movement for energy estimation, discussed in subsequent chapters.
Chapter 5

Modeling the Power Usage of Skid-Steer Vehicles

Author’s Note: A major portion of the work presented in this chapter has been published by the author in a research paper in the proceedings of the 2014 American Control Conference [41].

5.1 Introduction

Current ground robot designs provide little to no feedback on the remaining endurance of the vehicle. At best, the operator may see a simplified display of the voltage level of the battery. However, the voltage level of the battery does not provide an accurate estimate of the remaining endurance of the vehicle. This endurance is tightly coupled with the tasks the robot must complete, the surfaces and terrains upon which the vehicle must operate, and the energy storage technology used by the vehicle. For example, an operator that expends half of the energy in the batteries driving the robot down a hill will not have the energy required to return the robot to the starting position at the top of the hill. An obvious improve-
ment to current designs is feedback on the energy required to complete a traversal or return to the mission starting point. Accurate models of vehicle motion and power usage are required to calculate such feedback. The focus of the research presented in this chapter is the development of an accurate and adaptable power model of skid-steer movement relying on limited prior knowledge of vehicle design and operational surface type.

The approach presented in this chapter uses the kinematic model and ICR estimation algorithm presented in Chapter 3 with an ICR based power model similar to that presented in [32]. In this power model, track or wheel ICR locations are utilized to calculate slip velocities which, when combined with a friction model, produce estimates of power loss due to track terrain interactions. Preliminary results of adapting this algorithm for on-line parameter estimation are presented in [40] where the effects of terrain pitch were neglected. While ICR kinematics are valid only for low speed operation on relatively hard terrains, the combination of ICR and power model parameter identification still yield a flexible power usage estimation system requiring no prior knowledge of surface type.

The research presented in this chapter provides a novel method of estimating skid-steer robot power usage and mission energy requirements using little prior knowledge of the surface types to be traversed and including the effects of skid-steer turning and terrain gradients. Estimation techniques are employed to learn parameters describing vehicle motion and power usage during operation, which
makes the algorithms easily adaptable to a variety of robotic platforms.

The remainder of this chapter will discuss the development of a power model for skid-steer vehicle locomotion, the results of implementation and field testing of the algorithm on the vehicles, and end with concluding remarks on this topic.

5.2 Modeling Skid-Steer Vehicle Power Usage

Previous work on skid-steer power modeling has utilized ICR kinematics and terrain-related model parameters optimized in post-processing to accurately model power usage [32, 33, 41]. This section will present a version of this power model adapted to estimate terrain-related parameters during operation.

Three types of power usage are included in the power model. The first is power loss due to dynamic friction between the tracks or wheels and the ground. This sliding friction is significant because skid-steer vehicles rely on large amounts of slippage for turning. The power loss due to dynamic friction, $P_S$, is modeled as

$$P_{Sl,r} = \int A_{l,r} \vec{F}(a) \cdot \vec{v}(a) dA$$

where $P_{Sl}$ is the power loss from the left tractive surfaces, $P_{Sr}$ is the power loss from the right tractive surfaces, $A_l$ and $A_r$ are the left and right areas of the tractive surfaces, respectively, $\vec{F}(a)$ is the friction force resisting slipping at a point $a$ on the tractive surface, $\vec{v}(a)$ is the relative velocity between the tractive surface and
the ground at point $a$, and $dA$ is the differential of the surface integral. A coulomb friction model is used to calculate the force at a point of interest as

$$\vec{F}_a = -\mu p(a) \frac{\vec{v}_a}{\|\vec{v}_a\|}$$  \hspace{1cm} (5.2)

where $p(a)$ is the normal force between the tractive surface and the ground and $\mu$ is the coefficient of friction.

The relative velocity between the tractive surface and the ground, $\vec{v}(x)$, is difficult to determine, but knowledge of the ICR locations provides a method of calculating it. A diagram showing the right side track of a skid-steer vehicle is shown in Figure 5.1. A vector drawn from the geometric center of the vehicle to a differential area $dA$ is labeled as $\vec{a}$. The vector from the geometric center to the ICR location is known through estimation and is labeled as $\vec{C'}$. The track is in pure rotation relative to the ICR location, so the relative velocity between the track and ground can be calculated as

![Diagram showing right side track on skid-steer vehicle](image)

**Figure 5.1.** Diagram of right side track on skid-steer vehicle showing differential element of track contact area relative to ICR location.
\[ \vec{v}(a) = \vec{\omega}_z \times (\vec{a} - \vec{C}_{r,l}) \]  (5.3)

where \( \vec{\omega}_z \) is the angular velocity of the vehicle. Substituting (5.2) and (5.3) into (5.1) yields

\[
P_{S_{l,r}}^{l,r} = \int_{A_{l,r}} \mu p(a) \| \vec{v}_a \| dA
\]

\[
= \mu |\omega_z| \int_{A_{l,r}} p(a) \| \vec{a} - \vec{C}_{r,l} \| dA
\]  (5.5)

where \( \mu \) is assumed to be constant over the tractive surface.

Contact forces between the tractive surface and the ground are assumed to act at the center of the wheels for wheeled vehicles and at the track support wheels for tracked vehicles. The diagram in Figure 5.2 shows the assumed locations of the normal forces \( p \) on each type of vehicle. The normal force is assumed to be evenly distributed among the contact points so that \( p = W/N \) where \( W \) is the weight of the vehicle and \( N \) is the number of contact points. This reduces the integral in (5.5) to a summation

\[
P_S = \mu |\omega| p \sum_{n=1}^{N} \| \vec{a} - \vec{C}_{r,l} \|
\]  (5.6)

where the appropriate ICR vector, \( \vec{C}_r \) or \( \vec{C}_l \), is chosen based on the location of the point of interest \( n \). This equation provides an estimate of the power lost due to frictional forces during skid-steer turn maneuvers given the ICR locations and loading characteristics of the vehicle.
Figure 5.2. Assumed locations of normal forces acting on wheeled and tracked vehicles. Frictional forces are calculated at normal force locations.

The point load assumption simplifies the calculation of the skid-steer turn power loss calculation. Other methods of modeling this term would include an integral over the entirety of the contact patch. The point load model was chosen because it is simple and produces accurate power usage estimates, as will be shown in subsequent results. In addition, because the vehicles operate at low speeds during testing, the effects of vehicle dynamics on the normal forces at each contact point are neglected.

The next power loss term accounts for internal resistances in the drive train such as forces required to bend the track at link connection points and the rolling resistance of tires. The model is linear with respect to the absolute track or wheel speeds and is defined as

\[ P_I = G (|V_r| + |V_l|) \]  \hspace{1cm} (5.7)

where \( G \) is a resistance coefficient to be identified.

The final power usage term accounts for increased or decreased weight effects
due to vehicle pitch angle and is modeled as

\[ P_P = \beta mgV_x \sin \theta \]  

(5.8)

where \( \beta \) is a gain constant to be identified, \( m \) is the mass of the vehicle, \( g \) is the acceleration due to gravity, \( V_x \) is the forward velocity of the vehicle, and \( \theta \) is the pitch angle. The total power loss for the skid-steer vehicle is therefore

\[ P = \mu |\omega|p \sum_{n=1}^{N} \| \ddot{a} - \bar{C}_{r,l} \| + G \left( |V_r| + |V_l| \right) + \beta mgV_x \sin \theta. \]

(5.9)

Equation (5.9) is linear in the unknown parameters \( \mu \), \( G \), and \( \beta \). A recursive least squares (RLS) estimator is used to estimate the unknown parameters during vehicle operation. The first and third terms related to \( \mu \) and \( \beta \), respectively, require specific conditions for parameter identification. For \( \mu \) to be identified, the vehicle must undergo significant angular accelerations, and for \( \beta \) to be identified, the vehicle must travel along a surface with non-zero pitch. When a vehicle drives straight or on a flat surface, estimated values for \( \mu \) and \( \beta \) may not converge until the vehicle motion or surface pitch changes.

5.2.1 Surface Change Detection

A mobile ground robot is likely to encounter a variety of operating surfaces during a mission. It is likely that these surfaces will require different amounts of power for
the same skid-steer movement. Therefore, it is necessary to detect when surface changes occur and learn new model parameters. Properties that may be considered for surface detection include visual cues such as color and textural cues such as roughness. However, there may be no correlation between these properties and the power usage of the vehicle. A surface change detection algorithm working solely with power usage data is presented here that ensures surface changes resulting in changing power usage properties are detected.

The power estimate error $P_E$, defined as

\[ P_E = P_m - P \]  

(5.10)

where $P_m$ is the measured power usage and $P$ is the estimated power usage, is used to detect changes in surface type. A series of sample points are first produced by averaging the $P_E$ signal over a constant-length time window. This averaged signal is then used with an exponentially weighted moving average (EWMA) control chart to detect deviations from the expected zero mean of the $P_E$ signal [31]. An example of the EWMA control signal is shown in Figure 5.3 for data collected on the tracked vehicle during a run with three surface transitions. Large spikes are obvious in the power model error data at each surface change and are detected by the EWMA system.
5.3 Results of Power Model Implementation

This section will present the results of implementing the skid-steer power model of Section 5.2 on the test vehicles. The vehicles traversed curved paths on asphalt and grass surfaces using ICR-based trajectory control. The results from the tracked robot are presented first, and the results of the wheeled robot follow.

5.3.1 Tracked Vehicle Power Model Results

The GPS position of the tracked vehicle is shown in Figure 5.4 with the asphalt area as a white background and the grass area as gray. The vehicle started on asphalt and transitioned to the grass surface 380 seconds after the traversal began. During the asphalt portion of the run, the vehicle pitch was between -2.5 degrees and -6.2 degrees, and during the grass portion the vehicle pitch was between 4

\[\text{Figure 5.3. Example of EWMA surface change detection on tracked robot during run with three surface changes.}\]
degrees and 6 degrees.

The skid-steer power model parameters identified by the RLS algorithm during the run are shown in Figure 5.5. At time $t=390$ seconds after mission start, the EWMA control chart identifies a change in surface and the RLS algorithm is reinitialized to learn new power model coefficients. During the period of operation on asphalt, the coefficients were $\mu = 0.5$, $G = 89$, and $\beta = 1.6$. During operation on grass, the coefficients were $\mu = 1.3$, $G = 130$, and $\beta = 1.7$. Large changes in the values of $\mu$ and $G$ are expected in a change from asphalt to grass. The asphalt presents a relatively smooth surface to the tracked vehicle and creates almost no resistance to the tracked vehicle moving forward. The values of $\mu$ and $G$ increased significantly while traversing the grass portion. A $\mu$ value larger than one may appear strange, but the softer grass surface presents significant turning resistance.

![Figure 5.4](image_url)

**Figure 5.4.** Position of vehicle during power estimation testing. Asphalt and grass surfaces were traversed, and the transition from asphalt to grass occurred 380 seconds after mission start. Test performed with tracked robot.
to a skid-steer vehicle. The vehicle may sink into the surface, and the height of
the grass provides a non-negligible side-loading on the track. The softer surface
of the grass also provides increased rolling resistance resulting in a larger value of
$G$. The values of $\beta$ remain relatively constant for the asphalt and grass portions
of the traversal. This is expected because $\beta$ is simply a scaling factor on the pitch
contribution to power usage. The identified power model parameters for four runs
on the path are shown in Table 5.1. The parameters are consistent for each surface
type throughout all four of the test runs.

The measured and estimated power usages are compared in Figure 5.6. The
increase in power usage when transitioning from the asphalt to grass surfaces
can be seen at the time $t = 380$ seconds mark. The increase in power usage from
asphalt to grass is due to the surface change and the reversal in pitch. The increase

\begin{figure}
\centering
\includegraphics[width=\textwidth]{figure5_5}
\caption{Power model parameters $\mu$ (a), $G$ (b), and $\beta$ (c) learned during asphalt and grass traversal. Test performed with tracked robot. Surface transition from asphalt to grass occurred 380 seconds after mission start.}
\end{figure}
Table 5.1. Identified power model parameters for repeated runs with tracked vehicle.

<table>
<thead>
<tr>
<th>Run Number</th>
<th>Grass Surface</th>
<th></th>
<th>Asphalt Surface</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\mu$</td>
<td>$G$</td>
<td>$\beta$</td>
<td>$\mu$</td>
</tr>
<tr>
<td>1</td>
<td>1.3</td>
<td>130</td>
<td>1.7</td>
<td>0.5</td>
</tr>
<tr>
<td>2</td>
<td>1.2</td>
<td>138</td>
<td>1.7</td>
<td>0.5</td>
</tr>
<tr>
<td>3</td>
<td>1.3</td>
<td>138</td>
<td>1.7</td>
<td>0.4</td>
</tr>
<tr>
<td>4</td>
<td>1.5</td>
<td>133</td>
<td>1.8</td>
<td>0.5</td>
</tr>
</tbody>
</table>

In noise on the grass surface is due to the roughness of the surface affecting the movement of the tracked vehicle. Overall, the assumed constant-coefficient power model accurately tracks the measured power usage of a skid-steer vehicle.

The tracked vehicle was also tested on the more complex path shown in Figure 5.7 which included three transitions between grass and asphalt surfaces. Using the estimated vehicle pose from the ICR EKF, it is possible to map the identified power model parameters onto an image of the operational area. The estimated

![Figure 5.6](image)

**Figure 5.6.** (a) The measured and modeled power usage during the run. (b) The error in power estimation. Test performed with tracked robot. Surface transition from asphalt to grass occurred 380 seconds after mission start.
values of $\mu$, $G$, and $\beta$ are shown in Figures 5.8-5.10. The estimated values of $\mu$ and $G$ show a clear difference between the two surfaces just as in the previously presented results. There is slight overlap between surface types because the surface change detector utilizes an exponentially weighted moving average which results in some delay before a surface change is detected. It is interesting to note that initially the estimate of $\beta$ is poor. This is because the trajectory starts on the

**Figure 5.7.** Path of tracked vehicle while navigating a complex grass and asphalt path.

**Figure 5.8.** Values of power model coefficient $\mu$ estimated during traversal of path shown in Figure 5.7.
Figure 5.9. Values of power model coefficient $G$ estimated during traversal of path shown in Figure 5.7.

Figure 5.10. Values of power model coefficient $\beta$ estimated during traversal of path shown in Figure 5.7.

asphalt road which has a terrain pitch near zero at that location. After the vehicle transitions to the grass surface, the estimate of $\beta$ converges due to the increased terrain pitch variation within the grassy area.
5.3.2 Wheeled Vehicle Power Model Results

The wheeled vehicle was tested on the same paths previously presented for the tracked robot. The estimated power model parameters while traversing the path in Figure 5.4 at a forward speed of 0.3 m/s are shown in Figure 5.11. The asphalt to grass transition occurred at time $t = 558$ seconds but there is no surface change detected from a power usage perspective. Compared to the tracked vehicle, the power usage of the wheeled vehicle is similar for operation on asphalt or grass surfaces. The measured and modeled power usage, shown in Figure 5.12, match extremely well using the same power model parameters for both grass and asphalt. The identified parameters over four runs along the course are summarized in Table 5.2 and remain very similar for all runs.

The wheeled vehicle also traversed the more complex path shown in Figure 5.7. The identified parameters were again overlaid on a satellite image of the test area.

![Figure 5.11](image)

**Figure 5.11.** Power model coefficients identified on the wheeled vehicle while traversing the grass and asphalt path shown in Figure 5.4.
Figure 5.12. Comparison of measured and modeled power usage on wheeled robot while traversing path shown in Figure 5.4.

Table 5.2. Identified power model parameters for repeated runs with wheeled vehicle.

<table>
<thead>
<tr>
<th>Run Number</th>
<th>$\mu$</th>
<th>$G$</th>
<th>$\beta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.1</td>
<td>38</td>
<td>1.2</td>
</tr>
<tr>
<td>2</td>
<td>1.0</td>
<td>46</td>
<td>1.2</td>
</tr>
<tr>
<td>3</td>
<td>1.1</td>
<td>45</td>
<td>1.1</td>
</tr>
<tr>
<td>4</td>
<td>1.1</td>
<td>48</td>
<td>1.2</td>
</tr>
</tbody>
</table>

and are shown in Figures 5.13-5.15. In contrast to the tracked vehicle results, the power model parameters remain relatively constant while traversing both grass and asphalt surfaces. Limitations to the learning process are again visible in the data such as $\mu$ converging only after significant heading changes have occurred and $\beta$ converging only after the vehicle leaves the relatively flat asphalt road.
Figure 5.13. Values of power model coefficient $\mu$ estimated during traversal of path shown in Figure 5.7.

Figure 5.14. Values of power model coefficient $G$ estimated during traversal of path shown in Figure 5.7.
Figure 5.15. Values of power model coefficient $\beta$ estimated during traversal of path shown in Figure 5.7.
5.4 Conclusions

This chapter has presented the development and testing of a power modeling framework for skid-steer vehicles. Using the ICR locations estimated with the algorithm described in Chapter 3, it is possible to estimate the power usage of a skid-steer vehicle using a model that is linear in three unknown parameters. The well known recursive least squares algorithm is applied to identify the unknown model parameters. Finally, an exponentially weighted moving average control chart is utilized to detect surface changes that cause a significant difference in power usage for skid-steer movement.

The algorithm has been tested with a tracked and a wheeled skid-steer vehicle. An interesting result is that for the grass and asphalt surfaces used in testing, the power usage of the tracked vehicle is strongly coupled with the surface type on which the vehicle operates. The testing results from operation of the wheeled robot on grass and asphalt has shown that power usage is relatively constant for both surfaces. From the perspective of path planning or mission energy use prediction, this result is very useful because the wheeled robot may plan paths mainly from a distance perspective with negligible additional energy costs on grass and asphalt. The tracked vehicle, on the other hand, must consider the surface type to be traversed and the energy required to complete the path if energy usage is a mission concern. The following chapters will utilize this skid-steer power model for the prediction of robot energy usage along planned paths. This is useful for
both informing the operator of remaining vehicle endurance during a mission and for planning movement trajectories from an energy-usage perspective.
Energy Use Prediction in Skid-Steer Robots

6.1 Introduction

This chapter presents the integration of skid-steer motion modeling, trajectory control, and power usage modeling into a cohesive system for skid-steer energy use prediction. The vast majority of skid-steer robots available still require a human operator. This is because human interpretation of the environment is often required to complete ground mobile robot tasks. Examples of this include explosive device disposal and hazardous area inspection. Because these robots are predominantly powered by batteries, providing feedback to the operator concerning the energy required to complete planned tasks and of the energy remaining in the batteries would be a significant advancement of this technology.

Two energy prediction scenarios will be developed in detail. The first is the prediction of the amount of energy required to traverse a given path. The estimate of the energy required to complete a path can be compared to remaining stored
energy and the operator can be informed if the path requires more energy than is available. This will be referred to as forward energy use estimation. The second scenario is that of retro-traversal energy use estimation. Often the robot will enter a hazardous area during a mission and must eventually return to the operator for retrieval. By estimating the energy required for the retro-traversal path to the operator, it is possible to inform the operator when the robot will no longer have enough energy to return to the starting point if the mission is continued. Investigating these scenarios will be the focus of this chapter.

6.2 Prediction of Mission Energy Requirements

Equation (5.9) provides an estimate of the instantaneous power required for a skid-steer vehicle maneuver, but of more interest is the energy required to complete a series of skid-steer maneuvers that can be grouped together and considered a mission. The approach in this work will be to simulate a vehicle traversing the path of interest using a trajectory control algorithm. The movement of the vehicle is predicted using the kinematic models of motion and the ICR location estimates presented in Chapter 3. The trajectory control algorithm provides wheel speed commands necessary for driving a vehicle along a path. These wheel speeds, along with the estimated ICR locations, are inputs to the power use estimation algorithm presented in Chapter 5. Integration of the calculated power usage provides estimates of energy use for the path segment of interest. Knowledge of the traversal
path, either through a path-planning algorithm or by user choice, is necessary with this approach. The paths were created by manually driving the vehicle through the testing area. During path creation, the position, heading, and pitch of the vehicle were recorded and saved to a file that is read by the trajectory control algorithm during autonomous runs.

A schematic of the forward energy prediction method is given in Figure 6.1. The ICR EKF and power model coefficient estimators run continuously throughout the run. At frequent points during a run, the estimates of ICR locations and power model coefficients are utilized to estimate the energy required to traverse the path. When a surface change is detected, the vehicle energy usage is simulated from the previous surface change (or the mission start if no surface change has occurred) to the point of the change in surface. This energy is saved and that segment of the path is no longer simulated. Using the model parameters estimated for the new surface, the energy usage to finish the run is estimated and added to the energy required to reach the transition point. With this algorithm, the proper coefficients for each surface are utilized for energy estimation, and the length of the path traversal simulated is reduced for each surface change, improving computation time.

To create estimates of energy usage for retro-traversal, or returning to the starting point along the path already traversed, a simulated robot traverses the path back to the start and energy usage is estimated. A schematic of this process
Energy required to reach surface change is stored. This section no longer simulated. Path remaining after surface change. Energy required to reach the end of the path is calculated and added to the energy required to reach the surface change.

Figure 6.1. Schematic of forward path energy estimation. Filters utilize information from the traversed path to form estimates of ICR location and power model coefficients. The terrain pitch is recorded with the map points for power usage estimation.

is shown in Figure 6.2. To account for surface changes, the retro-traversal energy estimation is done incrementally. The simulation of energy usage is only performed from the current position to the start of the previous estimate of retro-traversal energy. The most recent estimates of model parameters are used for only the current segment of the path. This incremental method improves computation time because only the most recent segment of the traversal is simulated to create each estimate of retro-traversal energy usage. To obtain a measurement of the retro-traversal energy, the robot is commanded to traverse the entire path in reverse during a subsequent run. This second data set is used to determine measurements of retro-traversal energy based on the position of the robot along the path at any given time during the initial forward run.
Figure 6.2. Schematic of reverse path energy estimation. Filters utilize information from the traversed path to form estimates of ICR location and power model coefficients. The path already traversed is reversed and a simulated vehicle is used to estimate energy usage. The energy usage is estimated incrementally and only the portion of the path between the current vehicle position and the previous retro-traversal energy estimate is simulated.

6.3 Results of Forward Energy Prediction

This section will present experimental data for forward energy usage estimation. When performing forward energy prediction, the estimate could be improved by numerically integrating measured power usage to determine the energy usage for the route already traversed and only the remaining path would be estimated. However, the results for estimated forward energy usage presented in this section use no measurements of power usage. This was done to better ascertain the effectiveness of the forward energy estimation algorithm. A comparison of the estimated forward energy use and the measured energy required to complete the path is shown in Figure 5.4 with the tracked vehicle is shown in Figure 6.3. During this run, the
Figure 6.3. Comparison of the measured and estimated energy required to complete the path shown in Figure 5.4 with the tracked vehicle.

vehicle started on the grass portion of the run and finished on the asphalt portion. Initially, the estimator identifies the power model parameters for grass terrain and uses these parameters to estimate the power usage for the entire run. Because the latter portion of the run is on asphalt, the total energy required for the run is initially overestimated. When the vehicle transitions to the asphalt portion of the run, the surface change is detected and new power model coefficients for the asphalt terrain are learned. This is shown in the graph at time $t = 270$ seconds when the estimated energy requirement becomes very close to the measured energy usage. Without some source of prior knowledge about the surface types to be traversed by the robot, it is difficult to improve the estimate of required energy at the beginning of the run.

The result of forward energy estimation on the more complex path shown in
Figure 5.7 with the tracked vehicle is presented in Figure 6.4. As with the previous test traversal, the error in estimated energy usage exhibits large changes after each detected surface change. The error is less severe in this instance because the assumed power model parameters are correct for a portion of the path to be traversed.

The error in forward energy estimate at the end of a run was chosen as a metric of algorithm performance because at that point the entire traversal path is simulated with the best estimates of power model coefficients on each surface. A summary of the performance of the algorithm on the tracked vehicle over nine test runs is presented in Table 6.1. The final estimate of forward energy is quite good with eight of the nine runs being within 5% of the measured value.

Results of estimating forward mission energy on the wheeled robot for the same
Table 6.1. Results of forward energy estimation for nine test runs with the tracked vehicle.

<table>
<thead>
<tr>
<th>Run Number</th>
<th>Final Error in Forward Energy Estimate (J)</th>
<th>Measured Forward Energy for Run (J)</th>
<th>Percent Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2030</td>
<td>63840</td>
<td>3.1</td>
</tr>
<tr>
<td>2</td>
<td>2763</td>
<td>62590</td>
<td>4.4</td>
</tr>
<tr>
<td>3</td>
<td>111</td>
<td>71510</td>
<td>0.1</td>
</tr>
<tr>
<td>4</td>
<td>963</td>
<td>68488</td>
<td>1.4</td>
</tr>
<tr>
<td>5</td>
<td>4176</td>
<td>57313</td>
<td>7.2</td>
</tr>
<tr>
<td>6</td>
<td>2750</td>
<td>63415</td>
<td>4.3</td>
</tr>
<tr>
<td>7</td>
<td>1178</td>
<td>60356</td>
<td>1.9</td>
</tr>
<tr>
<td>8</td>
<td>2539</td>
<td>65520</td>
<td>3.8</td>
</tr>
<tr>
<td>9</td>
<td>1610</td>
<td>62511</td>
<td>2.5</td>
</tr>
</tbody>
</table>

paths are shown in Figures 6.5 and 6.6. The large jumps at surface transitions that were seen in the tracked vehicle data are not visible in the wheeled robot data. This is a result of the constant power model coefficient behavior observed with the wheeled vehicle for both surfaces, which was previously discussed in Chapter 5.

A summary of the performance of the algorithm over eleven test runs with the wheeled vehicle is presented in Table 6.2. For all eleven runs, the final estimate of forward energy is within 5.5% of the measured value.
Figure 6.5. Comparison of the measured and estimated energy required to complete the path shown in Figure 5.4 in reverse with the wheeled vehicle.

Figure 6.6. Comparison of the measured and estimated energy required to complete the path shown in Figure 5.7 in reverse with the wheeled vehicle.
Table 6.2. Results of forward energy estimation for eleven test runs with the wheeled vehicle.

<table>
<thead>
<tr>
<th>Run Number</th>
<th>Final Error in Forward Energy Estimate (J)</th>
<th>Measured Forward Energy for Run (J)</th>
<th>Percent Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1592</td>
<td>51621</td>
<td>3.0</td>
</tr>
<tr>
<td>2</td>
<td>2582</td>
<td>51788</td>
<td>4.9</td>
</tr>
<tr>
<td>3</td>
<td>1105</td>
<td>58946</td>
<td>1.8</td>
</tr>
<tr>
<td>4</td>
<td>2568</td>
<td>57366</td>
<td>4.4</td>
</tr>
<tr>
<td>5</td>
<td>487</td>
<td>53011</td>
<td>0.9</td>
</tr>
<tr>
<td>6</td>
<td>1872</td>
<td>55077</td>
<td>3.3</td>
</tr>
<tr>
<td>7</td>
<td>2784</td>
<td>50465</td>
<td>5.5</td>
</tr>
<tr>
<td>8</td>
<td>1327</td>
<td>56089</td>
<td>2.3</td>
</tr>
<tr>
<td>9</td>
<td>1501</td>
<td>52293</td>
<td>2.8</td>
</tr>
<tr>
<td>10</td>
<td>796</td>
<td>53632</td>
<td>1.4</td>
</tr>
<tr>
<td>11</td>
<td>1099</td>
<td>50541</td>
<td>2.1</td>
</tr>
</tbody>
</table>
6.4 Improving Forward Energy Prediction with Maps

The estimated power model parameters for the tracked vehicle show a strong correlation with surface type, as shown in Figures 5.8 and 5.9. If a vehicle has previously traversed an area, an obvious extension of the parameter identification is to create maps of expected power model parameters for the operation area. A color photograph of one of the testing areas is shown in Figure 6.7. There is an obvious color difference between the two surfaces identified during tracked vehicle testing: asphalt appears as light gray and grass as green and tan. One approach to creating the map would be to apply color thresholding and/or image segmentation algorithms to determine homogeneous regions within the image. Power model parameters could then be associated with the regions in which they were identified. This approach is difficult because, as shown in the grass area, surfaces may

Figure 6.7. Color satellite photograph of robot testing area.
not appear homogeneous in color but do act homogeneously from a power usage standpoint. The approach in this work is to apply clustering algorithms for map creation.

The three estimated power model parameters and the red, green, and blue image intensity values at each position during a traversal create a data set in six-dimensions. By applying the k-means clustering algorithm [49], it is possible to determine how groups of similar power model parameters map to pixel colors within the image. One drawback of the k-means algorithm is that the number of clusters must be known before the algorithm is applied. A satellite image must be available for this approach, so this isn’t thought to be a major disadvantage because the operator should be able to visibly determine the number of significant surface regions in the image.

The first step in creating the map is identifying the color intensity values for each position along the trajectory traveled by the vehicle. These intensity values are then matched with the corresponding estimates of power model coefficients. In this way, only the pixels in the image corresponding to the actual trajectory of the robot are used in the clustering algorithm. Next, the k-means algorithm is applied to the data to determine the power model coefficient values and the color intensity values that best represent clusters within the set. At this point, only the pixels traversed by the robot have been clustered. The remaining pixels are placed in the closest cluster as determined by the euclidean distance in the red-green-blue color
space. An example power map produced by this algorithm is shown in Figure 6.8. This map was created using the tracked vehicle data presented in Figure 5.8-5.10.

The grass and asphalt surfaces are well defined in the map but there is significant noise within each region. The result can be improved by filtering with a 9x9 mask median filter as shown in Figure 6.9. A median filter is ideal in this situation because as long as a mask with an odd number of values is chosen, the resulting filter output will always be one of the input values with no averaging. This retains the two-valued gray scale image representing the two surfaces. The estimated power model coefficients for each surface type are given in Table 6.3. It is interesting to note that the estimated value of $\beta$ is different for grass and asphalt when it is expected to be about the same. The large standard deviation on the asphalt $\beta$ estimate shows that due to the lack of significant pitch changes on the asphalt surface, it is difficult to estimate $\beta$ while operating on asphalt.

Using the power coefficient map as a lookup table, it is possible to improve the

![Figure 6.8. Power map image after k-means segmentation. Each of the two gray colors represent a set of power model parameters.](image)
forward energy estimates previously presented for the tracked vehicle. Figure 6.10 shows the forward energy prediction results for a traversal of the path shown in Figure 4.7. Compared to the results shown in Figure 6.3, usage of the map improves forward energy estimation significantly during the beginning portion of the run. Forward energy estimation with a map was also done for the complex path shown in Figure 5.7. The energy estimate, shown in Figure 6.11, does not exhibit the large jumps at each surface transition and the energy usage estimate is within 2% after 100 seconds.
Figure 6.10. Comparison of the measured and estimated energy required to complete the path shown in Figure 5.4 with the tracked vehicle. A power model coefficient map created during a previous traversal is used for prediction of energy usage.

Figure 6.11. Comparison of the measured and estimated energy required to complete the path shown in Figure 5.7 with the tracked vehicle. A power model coefficient map created during a previous traversal is used for prediction of energy usage.

6.5 Retro-Traversal Energy Use Estimation

As discussed in Section 6.2, the estimation of retro-traversal energy is simpler than forward energy prediction because the terrain to be estimated has already
been traversed and the parameters identified. A representative comparison of the measured and estimated retro-traversal energy usage for the tracked robot operating on the path shown in Figure 5.7 is given in Figure 6.12. The retro-traversal energy increases with time because the vehicle has traversed farther along the path and must therefore travel farther to return to the starting point. Similarly, a retro-traversal energy comparison for the wheeled robot operating on the same path is shown in Figure 6.13. A summary of retro-traversal energy results is presented in Table 6.4 for the tracked vehicle and in Table 6.5 for the wheeled vehicle.

![Figure 6.12. Comparison of the measured and estimated energy required to return to the starting point during a traversal of the path shown in Figure 5.7 with the tracked vehicle.](image-url)
Table 6.4. Results of retro-traversal energy estimation for eight test runs with the tracked vehicle.

<table>
<thead>
<tr>
<th>Run Number</th>
<th>Final Error in Retro Energy Estimate (J)</th>
<th>Measured Retro Energy for Run (J)</th>
<th>Percent Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3514</td>
<td>71511</td>
<td>4.9</td>
</tr>
<tr>
<td>2</td>
<td>3406</td>
<td>68488</td>
<td>5.0</td>
</tr>
<tr>
<td>3</td>
<td>491</td>
<td>68488</td>
<td>0.7</td>
</tr>
<tr>
<td>4</td>
<td>276</td>
<td>63838</td>
<td>0.4</td>
</tr>
<tr>
<td>5</td>
<td>2590</td>
<td>62590</td>
<td>4.1</td>
</tr>
<tr>
<td>6</td>
<td>4321</td>
<td>57292</td>
<td>7.5</td>
</tr>
<tr>
<td>7</td>
<td>3186</td>
<td>60356</td>
<td>5.3</td>
</tr>
<tr>
<td>8</td>
<td>2728</td>
<td>62510</td>
<td>4.4</td>
</tr>
</tbody>
</table>

Table 6.5. Results of retro-traversal energy estimation for eight test runs with the wheeled vehicle.

<table>
<thead>
<tr>
<th>Run Number</th>
<th>Final Error in Retro Energy Estimate (J)</th>
<th>Measured Retro Energy for Run (J)</th>
<th>Percent Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2948</td>
<td>47963</td>
<td>6.1</td>
</tr>
<tr>
<td>2</td>
<td>2809</td>
<td>53414</td>
<td>5.3</td>
</tr>
<tr>
<td>3</td>
<td>2142</td>
<td>51789</td>
<td>4.1</td>
</tr>
<tr>
<td>4</td>
<td>1276</td>
<td>48318</td>
<td>2.6</td>
</tr>
<tr>
<td>5</td>
<td>2660</td>
<td>55306</td>
<td>4.8</td>
</tr>
<tr>
<td>6</td>
<td>514</td>
<td>52114</td>
<td>1.0</td>
</tr>
<tr>
<td>7</td>
<td>1545</td>
<td>53626</td>
<td>2.9</td>
</tr>
<tr>
<td>8</td>
<td>1881</td>
<td>50532</td>
<td>3.7</td>
</tr>
</tbody>
</table>
6.6 Conclusions

This chapter has presented the results of estimating the energy requirements for skid-steer missions. The two scenarios investigated were predicting the energy required to traverse a given path and predicting the energy required to return to a starting point from the current position along a path. This is accomplished by integrating ICR estimation, trajectory control, and skid-steer power modeling into an energy estimation algorithm. Considering that the vehicle has no prior knowledge of the surfaces to be traversed, the resulting energy estimates match the measured energy very well.

When estimating the energy required to complete a given path, final energy estimates were within 5% of the measured value for eight separate test runs with the tracked vehicle and within 5.5% for of the measured value for eleven separate
test runs with the wheeled vehicle. When estimating the energy required to return to the starting point during a path traversal, both estimates were within 7.5% of the measured value for eight separate test traversals with each vehicle.

A path traversing multiple surface types causes large inaccuracies in forward energy estimation because power model parameters for future path segments are unknown. It was shown that an accurate map of power model parameters can be created from satellite imagery and previous traversals within the operational area. Use of the map during forward energy traversal greatly improves energy estimates at the beginning of a mission.
Chapter 7

Energy-Aware Path Planning

7.1 Introduction

The previous chapter showed that significant improvement in energy use estimation can be achieved by utilizing maps of power model parameters for a given operational area. The desired path was assumed to be known at the beginning of a mission. This chapter will show that power-related maps are also useful tools for energy efficient path planning of skid-steer vehicles.

After over half a century of advancement, path planning for ground mobile robots remains an important area of research. Following the work of Dijkstra in 1959 [13], well known methods for planning paths such as A-star, D-star, potential field, and many others have been developed [9]. Most traditional path planning algorithms operate within a grid placed over the operational area. An example path created using the A-star algorithm on an area with obstacles is shown in Figure 7.1.
Grid planning algorithms are extremely efficient and useful when planning paths where the cost of moving from one grid cell to another is well defined. The simplest example of a cost function would be the distance between cell centers with obstacle cells having infinite cost. However, energy cost functions for skid-steer vehicles are difficult to implement in this form. A skid-steer robot that spins in place to make a turn through a cell will use a different amount of energy than a smooth curve traversal through a cell, even if the robot ends up at the same end location from the same starting location.

Another approach to path planning is the Rapidly-exploring Random Tree (RRT) algorithm [9]. A diagram of the RRT algorithm is given in Figure 7.2. The first step is selecting a random state, labeled as $X_{rand}$ and shown in red in the diagram. The selection of the random state is often constrained. For example, in the 2-D path planning problem, the random state of vehicle location would need to

![Figure 7.1. Example of A-star path planning algorithm operating on area with obstacles shown as red circles.](image-url)
exist within the bounds of the operational area. The next step is to determine the state within the tree that is closest to the random state using a nearest neighbor algorithm. This state is labeled $X_{\text{near}}$ and shown in gray in the diagram. Finally, an input is determined that will move the system, subject to the system model, from $X_{\text{near}}$ to as close to $X_{\text{rand}}$ as possible. The state resulting from this input, labeled as $X_{\text{new}}$ and shown in green, is then added to the tree along with the edge from $X_{\text{near}}$ to $X_{\text{new}}$. Path planning is accomplished by repeatedly selecting a random state and expanding the tree. One of the benefits of RRT is the use of a system model when expanding the tree. This ensures that the final solution path is feasible for the system. However, it is difficult to apply an energy-based heuristic to RRT because the tree is expanded by the random selection of possible states and not driven by a measure of cost.

An approach similar to RRT, and the approach used in this research, is that

Figure 7.2. Example of RRT path planning algorithm operating on open areas.
of Sampling Based Model Predictive Optimization (SBMPO) [14]. In contrast to RRT, which selects a random state and then determines the inputs that move the system towards that state, SBMPO computes the change in system states due to a set of pre-determined inputs and then uses a cost function to determine which state should next be analyzed. The remainder of this chapter will discuss the details of the SBMPO algorithm and provide simulation results showing the reductions in energy usage that can be obtained through path planning.

7.2 Description of SBMPO

The SBMPO path planning algorithm combines a system model, which is used to predict a change in the system states from a given input, with a goal-directed optimization scheme. In this work, the well known A-star algorithm is used to direct the planning sequence towards a goal [9]. This section will provide a description of the SBMPO algorithm used in this research. A more detailed description of SBMPO can be found in the original published work of Dunlap [14].

A diagram of the SBMPO path planning process for a vehicle moving in two dimensions is given in Figure 7.3. The algorithm is initialized with a start position, a goal position, and a set of input values. The start position is labeled as node A in the diagram. Node A is added to a list called OPEN, which contains all nodes that have been found but not yet ‘expanded’. To expand a node, the movement of the vehicle created by each of the possible input values is simulated using the
Figure 7.3. Diagram of SBMPO planning steps to compute a path from node A to the goal region.

node position as the initial state. When node A is expanded, three new nodes are created and are labeled B, C, and D in the diagram. Node A is now removed from the OPEN list because it has been expanded, nodes B, C, and D are added to the OPEN list, and nodes B, C, and D are labeled as children of node A.

The next step is to choose a new node from the OPEN list to expand. This is done by computing a cost for each of the nodes on the OPEN list. In this example, the cost is the sum of the distance traveled to reach the node and the distance remaining to the goal. For example, the cost for node C would be

\[ C_C = d_{A,C} + D_{C,GOAL} \]  

(7.1)

where \( C_C \) is the cost of node C, \( d_{A,C} \) is the distance traveled from node A to node C, and \( D_{C,GOAL} \) is the euclidean distance from node C to the goal position. In the example of Figure 7.3, node C has the lowest cost and is the next node to be
expanded. The result is the new nodes E, F, and G being added to the open list as children of node C.

When the node costs are now computed, node G has the lowest cost and is expanded to produce nodes H, I, and J. At this point, node H is located within the goal and the path planning is complete. By traversing the stored parent-child relationships, the path of A-C-G-H is found from the start to the goal.

The similarities between SBMPO and A-star are now apparent. Both algorithms maintain a list of nodes to be expanded, called the OPEN list, and at each iteration expand the node with the lowest cost. When the goal is reached, the path is determined by iterating backwards through the stored parent/child node relationships. The main difference is that rather than expanding a node to the surrounding grid spaces as is done with classical A-star, the SBMPO algorithm expands by simulating many different possible trajectories originating at the node to be expanded. The simulation of vehicle movement is beneficial because the power model developed in Chapter 5 can be applied to determine the energy usage for each trajectory. This energy usage can be used as the cost value in place of a distance calculation.

In the algorithm description up to this point, there is nothing preventing two nodes from having the same state. Previous work in SBMPO algorithms has shown that it is necessary to implement some type of ‘pseudo-grid’ mechanism [14] to prevent the algorithm from continually expanding nodes within the same area,
especially when obstacle avoidance is necessary. In this research, a pseudo-grid is implemented to prevent two nodes with similar positions and headings from being created. A diagram of two nodes located at \((E_1, N_1, \psi_1)\) and \((E_2, N_2, \psi_2)\) is shown in Figure 7.4 where \(E\) is the east coordinate, \(N\) is the north coordinate, and \(\psi\) is the heading. The state differences used for implementing the pseudo-grid are the euclidean distance between the vehicle positions, \(\delta_d\), and the difference in headings for the two states, \(\psi_d\). The distance \(\delta_d\) is calculated using

\[
\delta_d = \sqrt{(E_1 - E_2)^2 + (N_1 - N_2)^2}
\]  

(7.2)

**Figure 7.4.** Example of two node state possibilities with pseudo-grid dimensions.
and the heading difference $\psi_d$ is calculated using

$$
\psi_d = |\psi_1 - \psi_2|.
$$

(7.3)

One complexity that arises from planning vehicle motion is that two states with similar positions but different headings must be preserved. When implementing the pseudo-grid, the distance metric $\delta_d$ is first calculated. If this value is below a set threshold, $\Delta_d$, the heading metric $\psi_d$ is then calculated. The node is created only if the heading metric is larger than a set threshold $\Delta_\psi$. The implementation of the pseudo-grid enables the algorithm to plan around obstacles and help to reduce the number of nodes in the queue. However, along with the choice of input space sampling, the restriction created by the pseudo-grid does remove any optimality properties of the planned path [14].

This section has presented a general overview of the SBMPO path planning routine. For application of the algorithm, the system model, selected system input values, cost function, and pseudo-grid thresholds must be defined. The following section will discuss application details for applying SBMPO to skid-steer robots using an energy-based cost function.
7.3 Application of SBMPO to Skid-Steer Path Planning

This section will discuss the details of applying the SBMPO path planning algorithm to a skid-steer vehicle. The ICR kinematic model presented in Chapter 3 is used to simulate vehicle movement. The selection of system input values and the application of the energy cost function are also key aspects of SBMPO implementation.

Following the work of others in this field [51], the input space is sampled using a Halton set to determine the possible inputs for path planning [14]. Halton point sequences are deterministic and are an example of a quasi-random number sequence. The algorithm for computing a Halton sequence is given in [19]. The input track speeds were allowed to range from 0 m/s to 1 m/s. The left and right track speed combinations selected using two Halton point sets of length forty (one for the left track, and one for the right track) are shown in Figure 7.5a. For comparison, forty combinations of left and right track speeds selected using a uniform random number generator are given in Figure 7.5b. The Halton points give much better coverage over the possible combinations of inputs than the random number generator.

Using the input combinations shown in Figure 7.5(a), a collection of possible vehicle movements can be predetermined for use in path planning. These
movements are given in Figure 7.6 and were determined by integrating the ICR kinematic equations for two seconds using each of the input track speed combinations shown in Figure 7.5(a). Computation time is saved by pre-computing the trajectories for a vehicle with 0 degrees heading and then rotating the trajectories into the vehicle frame during node expansion.

To calculate the cost of traveling along one of the trajectories given in Figure 7.6, the power usage model of Chapter 5 was implemented. Equation 5.9, which describes the power loss due to traction for a skid-steer vehicle, is repeated
here to aid in discussion:

\[
P = \mu |\omega| p \sum_{n=1}^{N} \|\vec{a} - \vec{C}_{r,l}\| + G (|V_r| + |V_l|) + \beta mgV_x \sin \theta.
\] (7.4)

Assuming constant input track speeds, ICR locations, and pitch for the duration of the trajectories shown in Figure 7.6, the equation can be written as

\[
P = \mu J_{mu} + G J_G + \beta J_\beta
\] (7.5)

where \( J_{mu} \), \( J_G \), and \( J_\beta \) are different constants for each of the possible system inputs. When a node is expanded, the values of \( mu \), \( G \), and \( \beta \) are determined using the node position and the map shown in Figure 6.9. The energy required to traverse the \( i^{th} \) trajectory is

\[
E_i = \Delta_t (\mu J_{mu,i} + G J_{G,i} + \beta J_{\beta,i})
\] (7.6)

where \( E_i \) is the required energy for the \( i^{th} \) trajectory and \( \Delta_t \) is the time duration of two seconds used to create each trajectory. Thus, when a node is expanded, the energy required to traverse each of the trajectories in the expansion can be calculated by determining the proper power model parameters from a pre-computed map.

The second cost to be computed is that between the final point in each expanded trajectory and the goal. This is generally done with the distance between the points in traditional grid-based A-star, but in this work the energy required to drive from
the node location to the goal is estimated. First, the vehicle is rotated until the vehicle heading points towards the goal point. Energy usage for this portion was estimated using the minimum value of $\mu$ in the map. Next, the time required to drive to the goal was calculated assuming a constant speed of 0.5 m/s. Knowing the time required to drive to the goal, the energy usage can be estimated using (7.5). The minimum value of $G$ within the map was used for this calculation. Minimum values of $\mu$ and $G$ were chosen because one of the restrictions of A-star optimization is that this cost heuristic between an expanded node and the goal must never be overestimated.

7.4 Path Planning Results

The SBMPO algorithm is computationally intensive and solutions for very short paths, less than 10 m, have been published for skid-steer vehicles. The paths planned in this work are slightly longer, usually 20-30 m, and provide some interesting insights on the energy improvements available with intelligent path planning. Two types of simulation results will be presented in this section. First, a series of artificial power maps were created to show the effects of changing surface types on the planned path. Second, possible real-world planning scenarios were simulated using a power map created from field tests with the tracked test vehicle.
7.4.1 Path Planning with Artificial Power Map

To investigate the behavior of the SBMPO algorithm, a path was planned repeatedly for the same start and end points while the power model coefficients within an artificial map were varied. The ICR values and pseudo-grid constraints for these simulations are given in Table 7.2. The ICR locations chosen are representative of values estimated on the tracked vehicle, and all parameters needed for power simulation, such as vehicle mass and size, were set to those given for the tracked vehicle in Chapter 2. The artificial map consists of two surfaces and models a road making a ninety degree turn.

In the first simulation, the two surfaces in the map have very similar power model parameters. Figure 7.7 provides the path planning solution using the energy cost function and Figure 7.8. The series of green markers represents the optimal path while the blue markers show the nodes created when planning the path. When the two surfaces are very similar, the energy cost function produces a planned path very similar to that of the distance cost function. There are slight differences in the two paths such that the energy-based path spends less time on the surface requiring

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$y_{ICRr}$</td>
<td>0.50 m</td>
</tr>
<tr>
<td>$y_{ICRl}$</td>
<td>-0.50 m</td>
</tr>
<tr>
<td>$x_{ICRv}$</td>
<td>0.10 m</td>
</tr>
<tr>
<td>$\Delta_d$</td>
<td>3.0 m</td>
</tr>
<tr>
<td>$\Delta_\psi$</td>
<td>50.0 degrees</td>
</tr>
</tbody>
</table>
more power. The result is that the energy-based path saves 1.6 kJ compared to the minimum distance path.

For the second simulation, the power model parameters within the dark regions of the map were increased. The resulting path with the energy cost function is shown in Figure 7.9 and the resulting path with the distance cost function is shown in Figure 7.10. The energy-based path shows a large increase in the number of nodes created while planning the path. The resulting path shows a significant bias towards traveling on the low-power surface, and 2.8 kJ of energy are saved by using the energy cost function.

In the third simulation, the power model parameters within the dark regions of the map were increased again, resulting in the energy-based path completely avoiding dark regions as shown in Figure 7.11. The energy-based path saves 4.8 kJ compared to the minimum distance path, shown in Figure 7.12.

**Figure 7.7.** Path planning solution on artificial map using energy cost function with two similar surfaces. Power model parameters for each surface are labeled in the figure.
Figure 7.8. Path planning solution on artificial map using distance cost function with two similar surfaces. Power model parameters for each surface are labeled in the figure.

Figure 7.9. Path planning solution on artificial map using energy cost function with two moderately different surfaces. Power model parameters for each surface are labeled in the figure.

The evolution of the energy-based path in Figures 7.7, 7.9, and 7.11 demonstrate the utility of the SBMPO path planning algorithm with an energy cost function. With sufficient knowledge of the surfaces on which the vehicle will operate, significant energy usage can be obtained through intelligent path planning.
Figure 7.10. Path planning solution on artificial map using distance cost function with two moderately different surfaces. Power model parameters for each surface are labeled in the figure.

Figure 7.11. Path planning solution on artificial map using energy cost function with two significantly different surfaces. Power model parameters for each surface are labeled in the figure.
Figure 7.12. Path planning solution on artificial map using distance cost function with two significantly different surfaces. Power model parameters for each surface are labeled in the figure.
7.4.2 Path Planning with Tracked Vehicle Power Map

The SBMPO algorithm has also been applied to the power model map created from field test results and shown in Figure 6.9. The ICR locations and other parameters used for the path planning simulations presented in this section are given in Table 7.2. As in the previous simulations, the ICR locations and other vehicle parameters were modeled after the tracked test vehicle. The light and dark area power model parameters refer to the values of $\mu$ and $G$ mapped in Figure 6.9.

A path planning solution for navigating a gradual corner is given in Figure 7.13. The algorithm determines that operating on the surface requiring less energy is the best path. The path planning solution using a distance heuristic, given in Figure 7.14, is a straight line as expected. The energy-based path planning algorithm saves 1.2 kJ of energy while increasing the path length by 1.55 m. However,

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$y_{ICRr}$</td>
<td>0.50 m</td>
</tr>
<tr>
<td>$y_{ICRl}$</td>
<td>-0.50 m</td>
</tr>
<tr>
<td>$x_{ICRv}$</td>
<td>0.10 m</td>
</tr>
<tr>
<td>$\Delta_d$</td>
<td>0.5 m</td>
</tr>
<tr>
<td>$\Delta_\psi$</td>
<td>5.0 degrees</td>
</tr>
<tr>
<td>Dark Area $\mu$</td>
<td>1.92</td>
</tr>
<tr>
<td>Dark Area $G$</td>
<td>134 N</td>
</tr>
<tr>
<td>Light Area $\mu$</td>
<td>0.59</td>
</tr>
<tr>
<td>Light Area $G$</td>
<td>94 N</td>
</tr>
</tbody>
</table>
Another interesting example of energy-based path planning is given in Figures 7.15. In this case, the start and goal points are on opposite sides of a sharp
corner. As shown in Figure 7.15, the planner investigates the possibility of planning a route around the corner, staying on the same terrain for the duration of the run. Because the duration of time spent on the higher energy use surface is so short, the straight line path is the most energy efficient option. This is similar to the simulated path on the artificial map shown in Figure 7.9.

A final example of energy-based path planning is shown in Figure 7.16. In this scenario, the vehicle starts and ends within the dark gray areas, which require more power to move, but the path is planned within the light gray area. Compared to the distance heuristic, shown in Figure 7.17, the energy-based path requires 2.5 kJ less energy. This is 20% less energy for an increase in distance of 1.3 m. The tracked robot was driven along the routes shown in Figures 7.16 and 7.17 to measure energy usage. Energy usage following the low energy path was measured to be 10,141 J, and energy usage following the minimum distance path was measured

![Figure 7.15. Path planning solution for sharp corner using energy heuristic. Blue circles represent nodes in the queue. The path solution is shown in green.](image)
to be 16,583 J. This is a reduction by 6.4 kJ, or 38%. The measured energy usage for the low energy path, 10.1 kJ, compares well with the simulated energy of 9.8 kJ. The measured energy on the minimum distance path, 16.5 kJ, is larger than the simulated energy for this path, 12.4 kJ. This is most likely due to changes in the grass surface during the period of time between map creation and energy measurement and also pitch variations in the grass surface that are not accounted for in the simulation.

Figure 7.16. Path planning for long straightaway using energy heuristic. Planned route operates on lower energy-use surface.
Figure 7.17. Path planning for long straightaway using distance heuristic. Planned route traverses higher energy-use surface straight from start to goal.

7.5 Conclusions

The focus of this chapter was to demonstrate the feasibility of using power model parameter maps to improve the performance of energy-based path planning algorithms. By integrating the skid-steer power model described in Chapter 5 with a Sample Based Model Predictive Optimization (SBMPO) path planning algorithm, energy-efficient path planning was demonstrated for a variety of scenarios using both an artificial power map and a power map created from experimental data. The results show a decrease in energy usage for the new path planning algorithm, but the paths cannot be considered optimal due to the sample-based approach and the implementation of the pseudo-grid. There remains opportunities for significant work in this field.

The path planning algorithm presented in this chapter did not consider the pitch of the terrain when determining the most energy efficient path. Pitch could
be easily included if available, but because the surface pitch can change significantly with relatively small deviations in position, accurate pitch maps are not available and are difficult to create. Through repeated testing in the operational area, it may be possible to create pitch maps for use in path planning.

The current implementation of the path planning algorithm takes significant time to compute a path. While A-star is one possible solution for selecting the optimal path, various other algorithms such as D-star and D-star Lite are available in the literature as well. Because the focus of this portion of research was to show the feasibility of maps for energy efficient path planning, other optimization schemes were not explored.
Conclusions and Future Work

8.1 Summary

This dissertation has presented a framework for modeling and control of skid-steer ground robot motion and energy usage. A tracked skid-steer robot and a wheeled skid-steer robot were modified and instrumented to perform experimentation related to this research. Through the use of an open-source software libraries and tools, along with a significant number of custom-written device drivers, advanced sensors and algorithms were implemented on the vehicles for field testing. In initial work, an extended Kalman filter (EKF) designed to estimate the instantaneous centers of rotation (ICRs) between skid-steer tractive surfaces and the ground was developed. Using inputs of wheel or track speed and measurements of vehicle position and heading, field test results showed that ICR locations remain relatively constant when skid-steer vehicles are operated at low speeds. After traversing ap-
proximately 1 km with both the tracked and wheeled vehicles, the largest standard deviation in ICR estimate was 0.10 m and occurred on the tracked vehicle. The mean ICR estimate corresponding to this standard deviation was 0.38 m, showing that at low speeds the ICR locations remain bounded in small regions. Also, once ICR locations are identified, the ICR kinematic equations provide accurate open-loop estimates of vehicle motion. Testing with the tracked vehicle showed that open-loop ICR odometry predicted vehicle position with a mean error of 1.4% of the distance traveled.

Next, ICR locations are utilized to create a kinematic mapping between the motion of a skid-steer robot and that of a two wheeled, or unicycle, robot. Unicycle robot trajectory control has been an active area of research for many years, and the mapping allows unicycle robot control strategies to be applied to skid-steer vehicles. A nonlinear trajectory control law was selected from the literature and adapted for skid-steer robot control in both simulation and practice. Through extensive field testing, the control law produced impressive performance on skid-steer vehicles, especially in light of the fact that the algorithm requires no prior knowledge of the vehicle size, vehicle configuration, or surface type on which the vehicle operates. When controlling the tracked vehicle along a curved path, the mean absolute distance error was 0.056 m and the mean absolute heading error was 4.3 degrees. The wheeled robot produced results showing a mean absolute distance error of 0.033 m and a mean absolute heading error of 2.7 degrees for the
same paths. There are limitations to unicycle control implementation. Because the dynamics of the vehicle are neglected, the system works best at low speeds where the skid-steer vehicle is able to achieve the commanded angular rate within a reasonable amount of time.

Presented next is an algorithm for skid-steer vehicle power usage modeling. The power model accounts for power losses due to dynamic friction from slippage, internal and rolling resistances, and the effects of pitched terrains. The presented model is linear in three unknown parameters, and it is shown that the well known recursive least squares (RLS) algorithm is well suited for parameter identification. Repeated field testing of both the tracked and wheeled vehicles shows that the assumed constant-coefficient power model accurately estimates skid-steer power usage regardless of maneuver type. Also, it is shown that the tracked vehicle power usage is much more sensitive to a change in the terrain types tested than the wheeled vehicle. While the wheeled vehicle will certainly exhibit a change in power usage behavior for terrains not tested in this work, it is interesting to note that wheeled vehicle configurations may provide power-use benefits in certain terrains.

By integrating ICR identification, trajectory control, and power usage modeling, it is next shown that the energy required to complete a predefined path can be accurately estimated during traversal. Over a total of nine runs with the tracked vehicle and eleven runs with the wheeled vehicle, the energy predicted to complete
the path is estimated within 7.2% of the measured value. Retro-traversal energy, the energy required to return to the starting point along the previously traversed path, is estimated within 7.5% of the measured value over eight runs with both test vehicles. Inaccuracies due to unknown surface changes in the future are shown to be mitigated by the use of a power model coefficient map created during previous runs. In this work, the map is created using a k-means clustering algorithm working on the identified power model parameters and the corresponding satellite image color properties for the location of operation.

Finally, a recently developed path planning algorithm suitable for skid-steer mobility was adapted to use the power model developed in this research as a cost function. Power model parameter maps created during experimental testing of the tracked vehicle were used for path planning. While computationally intensive, the path planning algorithm shows promise for reducing energy usage when compared to a traditional distance-based heuristic. Simulations using power model parameter maps created from experimental data show energy reductions of 20% with an increase in path length of only 1.3 m. However, the reduction in energy usage is highly dependent on the path planning situation.

In summary, while spanning the topics of motion modeling, trajectory control, power use modeling, energy use prediction, and path planning, the algorithms developed in this research fit together as a cohesive unit for skid-steer robot mission analysis.
8.2 Future Work

This research offers many opportunities for future work. Several possible topics are outlined as follows:

- Many robots used within buildings are unicycle robots. While the ICR EKF was designed with skid-steer vehicles in mind, implementing the ICR EKF on unicycle robots may provide useful feedback for fault detection. If ICR estimates stray from the vehicle wheel locations, the vehicle is no longer moving with no wheel slip or one of the sensors on the vehicle must be faulty.

- Many additional opportunities exist for the use of ICR kinematics in the control of skid-steer vehicles. While the application of unicycle control strategies is a useful technique, the application of ICR kinematics to other areas, such as model predictive control, could prove fruitful.

- The power model presented in this work identifies power losses found after the motor output. Using previous work in robot power train modeling, the overall power usage of the vehicle could be estimated for more accurate mission energy prediction.

- Battery state of charge estimation is a rapidly advancing field of research. Work in combining the skid-steer energy use models developed in this research with accurate state of charge estimators would provide valuable feed-
back to robot operators. Also, battery health degradation is a function of the rate of battery discharge and depth of battery discharge. Proper planning could adjust battery discharge rates so that battery health is preserved.

- The ICR EKF developed in this research utilizes measurements of position and heading for ICR identification. Many additional sensors are commonly available on robotic platforms, including cameras, LIDARS, and angular rate sensors. Including these measurements may improve ICR EKF performance.

- Investigate different optimization solvers for path planning using sample-based approach. While the A-star algorithm is effective, other schemes may prove more computationally efficient.

- Improving the implementation of the pseudo-grid necessary for sample-based path planning could significantly increase computational efficiency. Pseudo-grid adjustments also have a significant impact on the optimality of the final path.

8.3 Conclusions

This research has developed a framework for the motion estimation, control, power use modeling, and energy use prediction for tracked and wheeled skid-steer robots. The research has spanned many different areas in robotics, but each component of the work builds upon the others to create a cohesive algorithm for energy use
prediction and energy-aware path planning for skid-steer mobile ground robots.

A common theme throughout this work is the identification of model parameters
during the operation of the vehicle. This is in contrast to most available works
on this topic, which generally require extensive prior knowledge of the terrain for
implementation.
Bibliography


[43] Plaskonka, J. The path following control of a unicycle based on the chained form of a kinematic model derived with respect to the serret-frenet frame.


Vita

Jesse Lorenzo Pentzer

Jesse Pentzer was born in August, 1985 in Cottonwood, Idaho. He received his Bachelor’s and Master’s degrees from the University of Idaho in 2008 and 2010, respectively.

Jesse received the University Graduate Fellowship and arrived at The Pennsylvania State University in August of 2010 to begin research work in pursuit of a Ph.D. Jesse immediately began working in the lab of Dr. Sean Brennan on the modeling and design of robot power trains and was co-advised by Dr. Karl Reichard. Ground robot modeling swiftly turned to implementation using robot platforms available at the Applied Research Lab. Funded under the Applied Research Lab Walker Graduate Research Assistantship, Jesse eventually focused his Ph.D. work on modeling the movement and power usage of skid-steer ground robots. He has also been involved with the evaluation of the Standard Test Methods for Response Robots developed by the National Institute for Standards and Technology, and he also assisted with the setup and operation of the DARPA Robotics Challenge Trials in December, 2013.