OMNI-DIRECTIONAL LOCOMOTION FOR MOBILE ROBOTS

A Thesis Presented to

The Faculty of the

Fritz J. and Dolores H. Russ
College of Engineering and Technology

Ohio University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

by

Brian Edward Carter

June, 2001

OHIO UNIVERSITY
LIBRARY
# Table of Figures

Figure 1: The Mecanum Wheel .................................................. 2  
Figure 2: Ball Wheel Design (West et al, 1997) .................................. 3  
Figure 3: Model of the Dual-Wheel Caster Omni-directional Robot (Han et al, 2000) .... 4  
Figure 4: The Phase II RoboCup Robot ........................................ 10  
Figure 5: The Phase III RoboCup Robot ....................................... 11  
Figure 6: The Transwheel .......................................................... 12  
Figure 7: The Phase IIIB RoboCup Robot ...................................... 13  
Figure 8: The Phase IV RoboCup Robot ....................................... 14  
Figure 9: Kinematic Diagram of Phase IV RoboCup Robot .................. 15  
Figure 10: Block Diagram of Simplified Dynamic Model .................... 26  
Figure 11: Step Response of Simplified Dynamic Model .................... 27  
Figure 12: Ramp Input Response of Simplified Dynamic Model ............ 28  
Figure 13: 3rd Order Input Response of Simplified Dynamic Model ........ 30  
Figure 14: Wheel Rotations for 3rd Order Input for the Simplified Dynamic Model .... 31  
Figure 15: Diagram of Complex Dynamic Model ............................ 32  
Figure 16: Ramp Input Response of Complex Dynamic Model .............. 33  
Figure 17: 3rd Order Input Response of Complex Dynamic Model ........... 34  
Figure 18: Trajectory of Complex Dynamic Model with a Circular Input ...... 35  
Figure 19: Orientation of Complex Dynamic Model with a Circular Input .... 36  
Figure 20: Wheel Velocities of the Complex Dynamic Model with a Circular Path Input ................................................................. 37
Figure 21: Omni-Directional Mobile Robot Model

Figure 22: Omni-Directional Wheel Detail

Figure 23: Friction Coefficients vs. Sliding Velocity

Figure 24: Experimental Set-up for the Omni-Directional Mobile Robot

Figure 25: The Dynamic Model with Slip and Roller Gaps

Figure 26: Simulated and Experimental Results for the Omni-Directional Mobile Robot
   Without Rollers

Figure 27: Simulated and Experimental Results for the Omni-Directional Mobile Robot
   With Rollers

Figure 28: Dimensions of the RoboCup Playing Field

Figure 29: Top Level Diagram of Simple Dynamic Model

Figure 30: Schematic of the Dynamics Block of the Simplified Dynamic Model

Figure 31: Block Diagram of the Complex Dynamic Simulink Model

Figure 32: Exploded View of "Dynamics" Block of Slip-Simulation
Symbols Used

\( R \) Wheel radius (m)

\( L_1 \) Radial distance to front wheels from the robot's center of gravity (m)

\( L_2 \) Radial distance to rear wheel from the robot's center of gravity (m)

\( M \) Mass of robot (kg)

\( g \) Gravity Constant (9.81 m/s\(^2\))

\( k \) Driving torque gain factor

\( I_w \) Moment of inertia of the robot's wheels (kg\( \cdot \)m\(^2\))

\( I_v \) Moment of inertia of the robot (kg\( \cdot \)m\(^2\))

\( c \) Viscous friction factor of the wheels (kg\( \cdot \)m\(^2\)/s)

\( x_m, y_m \) Cartesian position of robot with respect to the mobile robot coordinate frame (m)

\( X_w, Y_w \) Cartesian position of robot with respect to the world frame (m)

\( \phi \) Orientation of the robot with respect to the world frame (rad)

\( \theta_i \) Angular position of wheel \( i \) (rad)

\( \mu_{\max} \) Maximum friction coefficient parallel to wheel rotation

\( \mu_{\max} \) Maximum friction coefficient orthogonal to wheel rotation

\( \mu_{\text{space}_{\max}} \) Maximum friction coefficient when wheel space between rollers is in contact with the base surface.

\( \delta \) Angular offset of two front wheels with respect to the mobile robot frame (for our final design, \( \delta = 15^\circ \))
1 Introduction

This project began when a cross-disciplinary collection of engineering students and faculty (Electrical Engineering, Computer Science, and Mechanical Engineering) decided to take part in the 2001 RoboCup competition in Seattle, Washington on August 2\textsuperscript{nd}. This thesis will cover the basic history of this competition, the design process (with respect to Mechanical Engineering), the kinematics, inertial dynamics, and slip dynamics of the final design. In addition, experimental results shall be presented, and compared to the simulated performance of the mobile robot.

1.1 Mobile Robot Background

The field of mobile robotics is vast, covering nearly 40 years of research and development. Basically, a mobile robot is a device capable of autonomous movement throughout its environment. The movement does not need to be complex (follow a light, sound, or stripe, for example), but the robot must control its own movement. A remote control toy car is not a mobile robot, but that same toy car would be a mobile robot if a computer were controlling its motion. See (Schofield, 1998) for an overview of practical mobile robots, and (Nehmzow et al, 1998) for an overview of research in this area. In addition, see (De Wit, 1998) and (Borenstein et al, 1997) for overviews of mobile robots in general.

1.2 Omni-Directional Locomotion Background

Omni-directional locomotion is the ability to move in any direction while facing any orientation. This is a desirable ability for a mobile robot, making it more agile and flexible. One of the driving factors in the development of omni-directional locomotion is
the wheelchair industry. A conservative estimate indicates over 2 million people with severe needs in the European Community alone could benefit from an omni-directional wheelchair (Borgolte et al, 1998). Another use for omni-directional locomotion is the personal (commercial) robot. Forays are already being made in the use of robots to mow lawns and sweep floors. Omni-directional capabilities would be of great advantage to these types of machines.

There are many different designs being built and tested to develop omni-directional locomotion, from many-legged walking robots to slithering snake-like robots. There are even floor scrubbing robots without wheels or legs: the rotating brushes themselves are the propulsion for omni-directional movement (Shin et al, 2000). As this thesis covers the development of a wheeled omni-directional mobile robot, this background will discuss some of the most common wheeled designs.

Many of the wheeled omni-directional designs make use of the Mecanum wheel, shown below in Figure 1 (Borgolte, 1998). This wheel allows a standard four-wheel set-up (similar to an automobile) to obtain omni-directional locomotion through various rotational combinations. As can be seen in the Figure, the wheels are capable of powered locomotion along their primary axis like any other wheel, while the angled "rollers" along the outer axis of the wheel are allowed to freely rotate.

Figure 1: The Mecanum Wheel
Another omni-directional locomotion design is the Ball Wheel Design (West et al, 1997), shown below in Figure 2. This consists of powered rotation along one axis due to the motor, while the ball is free to rotate along any other axis due to the chassis mounted rollers. However, this design is complex and difficult to manufacture.

![Figure 2: Ball Wheel Design (West et al, 1997)](image)

A third design for wheeled omni-directional locomotion is the Active Dual-Wheel Caster mechanism (Han et al, 2000). This was created to overcome the liabilities inherent in the Mecanum wheel, namely vibrations caused by the rotation of the wheel (due to the roller contact) and the relatively small load-bearing properties. A model of the dual caster omni-directional robot is shown in Figure 3. As can be seen in the figure, there are two dual-wheel caster mechanisms in line on the robot. Each of the four wheels can be independently driven, while two joints (located at points $O_a$ and $O_b$) are free to rotate passively. The omni-directional locomotion is created by a combination of rotations in the four wheels of the assembly. This design has the advantage of not needing specialized wheels.
Although these three designs are the most commonly researched, other methods of omni-directional locomotion are being developed. See (Jung et al., 2000), (Mori et al., 1999), and (Witus, 2000) for alternative designs.

1.3 RoboCup Background

The Robot World Cup Initiative, or RoboCup for short, is the international project for the promotion of Artificial Intelligence (AI), robotics, and related fields (RoboCup 2001 Official Website). Soccer was the vehicle chosen for this promotion, due to the international interest in the game, as well as the fact that the game of soccer has clearly defined rules and aims. The ultimate goal of RoboCup is "By 2050, develop a team of fully autonomous humanoid robots that can win against the human world
champion team in soccer." Various technologies are integrated towards that goal, including design principles of autonomous agents, strategy acquisition, real-time reasoning, and of course robotics.

The RoboCup competition is now several years old. The idea of robots playing soccer was first put forth by Dr. Alan Mackworth, from the University of British Columbia, in his paper "On Seeing Robots". Independently, the Workshop on Grand Challenges in the field of Artificial Intelligence was organized by a group of Japanese researchers, proposing Grand Challenge topics. This workshop led to serious discussion of using the game of soccer as a platform to promote AI and robotics. After this group performed several investigations (including a technological feasibility study and a social impact assessment), it was deemed that the project was both feasible and desirable.

In 1993, a group of researchers desired to launch a robotic soccer league, tentatively title the J-League. However, they received worldwide interest, and thus renamed the league the Robot World Cup Initiative. In September 1993, the first public announcement of RoboCup was made, and rules were created for the competition. Debate over these rules, and the technological issues involved, were discussed at many meetings and conferences. At the International Joint Conference on Artificial Intelligence held in Montreal, Canada, August, 1995, the decision was made to have the first RoboCup Competition in conjunction with that Conference in 1997. In addition, a Pre-RoboCup would be held in 1996, at the International Conference on Intelligent Robotics and Systems in Osaka, Japan from November 4th through the 8th.

The Pre-RoboCup competition consisted of 8 teams competing in the Simulator League, and a demonstration of real robots competing in the Middle Size League.
Although a relatively small event, it was the first competition using soccer to promote research and education.

The first RoboCup Competition, held in 1997 was quite a success, with over 40 competitors (real and simulator leagues), and over 5000 spectators. Since then, the competitions have only gotten bigger. The latest, held August 28th to September 3, 2000 in Melbourne, Australia, hosted 110 teams from over 19 countries (RoboCup 2000 Official Website). The event itself attracted over 50,000 fans, while approximately 3 million fans closely followed their nations teams through television and the Internet.

In addition to the games of soccer, the RoboCup competitions feature other demonstrations and competitions. There is RoboCup Jr., which is aimed at younger students. They can, using simpler robots, compete in three competitions: 2 on 2 games of soccer, rescue, and dance. The 2 on 2 soccer game is self explanatory, while the rescue competition requires the student's robot to "rescue" a victim from a simulated disaster. The dance competition allows one or more robots to perform to music, being rated on costume creativity and movement.

1.3.1 RoboCup Leagues

There are five leagues in the RoboCup Competition: Simulator League, Small-Size League (F-180), Middle-Size League (F-2000), Sony Four Legged Robot League, and the Humanoid League. The Simulator League consists of teams playing a simulated game of soccer on a computer. Each team consists of 11 programs, each controlling a single "player" (Asada et al, 1999). There is no real world hardware involved in this league, only programming savvy.
The Small-Size League (F-180) consists of small, real-world robots competing against each other in a game of soccer. Each team consists of at most five robots, one of which plays goalie. This is the only league Ohio University is planning to compete in at the RoboCup 2001 competition in Seattle. The rules for this league will be covered in detail in the next section (1.3.2 RoboCup Small-Size Rules).

The Middle-Size League (F-2000) consists of bigger, real-world robots competing against each other in a game of soccer. It is basically the Small-Size League on a grander scale. The robots, ball, and field are all bigger.

The Sony Four Legged Robot League consist of teams of Sony's four legged, autonomous robot "dogs" competing in a game of soccer. These robots are provided by Sony, and are all identical. Only the internal working (i.e., programming) of each robot is determined by the programmers who prepare them (Sony Legged Robot League Official Website).

The final league, the Humanoid League, is not yet up and running. It will one day consist of teams of human-like robots playing games of soccer. It is the ultimate goal of the RoboCup Competition, and will be difficult to achieve. An upright, humanoid robot is one of the biggest challenges in the field of robotics, but a biped humanoid league is planned for RoboCup 2002.

1.3.2 RoboCup Small-Size League Rules

Although the RoboCup competition follows the rules of human soccer closely, due to the format involved, some rules have been modified. The rules for the 2001 RoboCup Competition for the Small-Size League in their entirety are shown in Appendix
A. These can be found on the Official RoboCup 2001 Rules Draft Page (RoboCup 2001 Rules Draft Website).

1.4 Thesis Objective

The primary purpose of this thesis was the documentation of the work done to develop the player robots for the Ohio University RoboCup 2001 team. Another design team is working on the development of specialized goalie robots. This documentation concentrated on the mechanical design, development of the kinematic equations for the robot, and the computation of the dynamic equations of motion. In addition, simulations of kinematic and dynamic control will be presented, as well as experimental results. The secondary purpose of this thesis was the documentation of the background and rules for the RoboCup competition itself. As this work is the first on the subject of RoboCup from Ohio University, it will hopefully aid any future ventures in this direction.

1.5 Thesis Organization

The first subject discussed in this thesis is the design of the RoboCup robots. This will cover each design iteration in detail, discussing problems and advances in the design process. Next the kinematics for the final robot design will be analyzed, giving the inverse kinematic equations. The dynamics will then be derived step by step. Simulation results will be presented in the next two sections, describing in detail both the simple dynamic model, and a more complex RoboCup model that follows the proposed control architecture for the RoboCup competition. A third simulation will be presented, accounting for wheel slippage, and this will be compared with experimental results. Finally, conclusions will be drawn from the work.
2 Design

There were several factors that influenced the design process for the RoboCup 2001 robot. It was necessary for the robot to be agile, but heavy enough not to be pushed around by the other teams on the field. While purposefully ramming another team's robot is against the rules, some incidental contact would of course occur. The robot should also be able to "kick" the ball relatively hard, with a degree of accuracy. While a detailed list of the rules can be found in Appendix A, four rules in particular complicated the Mechanical Engineering design process.

- Kicker - Specialized kicking mechanisms are acceptable.
- Area - The robot can be any shape, but the total floor area occupied by a robot must be less than 180 cm$^2$. This rule was later rescinded for the 2001 competition, but only after several design iterations were completed.
- 18 cm rule - The robot must be able to fit into an 18 cm diameter cylinder.
- Height - If a global vision system is used, the height must be less than 15 cm. If a local (robot-mounted) vision system is used, the height must be less than 22.5 cm. All the players will be using a global vision system.

2.1 Phase II Design

The Mechanical Engineering design process began with what was known as the Phase II RoboCup Robot (see Figure 4). Designed by John Hall, a former Ohio
University graduate student, this was a simple two-wheeled robot with skids on the front and a roller-ball on the back. The kicking mechanism consisted of a solenoid propelling an aluminum block forward, "kicking" the ball. However, in practice, this design had several faults. The four supports gave the entire robot a flimsy feel, and the kicking mechanism never worked properly. A solenoid capable of "kicking" the ball with the desired force would have been too large to mount on the robot.

The final flaw in the Phase II design became apparent from a videotape of the RoboCup 2000 competition. Many of the robots on this videotape possessed two greatly desired capabilities: omni-directional motion and a "dribble-bar". Omni-directional motion is the ability to move in any direction while facing any orientation. The "dribble-bar" was a spinning rod that would impart backspin upon the ball, allowing the robot to move while maintaining its control of the ball.

![Figure 4: The Phase II RoboCup Robot](image)

Several different methods of omni-directional locomotion were considered, but most were too bulky, or the wheels were too expensive. However, one relatively simple solution fit our size constraints, as well as our budget. Three wheels, which were able to
have free rotation in one direction, and powered rotation in an orthogonal direction, should enable omni-directional movement (Watanabe et al, 1998). This theory is expanded upon in the kinematics section.

2.2 Phase III Design

The next design iteration undertaken when the author joined the RoboCup 2001 team was dubbed the Phase III, and followed the theory of (Watanabe et al, 1998) exactly, making the dynamics of this Phase the simplest of any alternative considered due to the inherent symmetry of the design. The wheels were arranged in a symmetrical manner, 120° apart. As can be seen from Figure 5, this Phase only made it as far as the basic design, as there was no way to mount a kicking mechanism of suitable size while maintaining observance of the RoboCup rules. However, it was used as a test bed to verify that the selected motors would be able to move the vehicle at a sufficient velocity for the competition. Special note should be taken of the wheels, shown in Figure 6.

Figure 5: The Phase III RoboCup Robot
These wheels, produced by the Kornylak Corporation (Kornylak Corporation Website) and dubbed the Transwheel, were designed for the material processing industry. Not only are they sturdy, they're also economical. As can be seen from the figure, the wheels can be used for powered rotation along the primary diameter, just as any other wheel. However, the smaller rollers along the outside of this diameter allow free rotation along an orthogonal direction to the powered rotation.

Due to the size limitations from the RoboCup rules, the Phase III was not a feasible design. To make room for a kicking apparatus, the symmetry of the Phase III would have to be lost, complicating the kinematics and dynamics calculations. However, this change in geometry did optimize the speed in the forward direction, with a corresponding loss of overall omni-directional agility.

2.3 Phase IIIB Design

The next design was dubbed the Phase IIIB, and displays the kicker/dribbler design created by Mechanical Engineering undergraduate students working on this project. As can be seen from Figure 7, the bar at the front of the robot is the dribble bar.
The bar that rotates about the dribbler is the kicker bar. Unlike the Phase III, the Phase IIIB's front wheels are at a 105° angle from the back wheel, rather than the symmetrical 120°. However, this allowed a full 5" of kicker area, allowing easier contact with the ball. This was the sturdiest design, and was promising.

![Figure 7: The Phase IIIB RoboCup Robot](image)

Unfortunately, there were some design issues that the prototype of the Phase IIIB brought up. The first was the height restriction. With this design, we had approximately 3.5" of available space for the electrical components necessary to run the robot. Since we wanted to maximize the amount of payload, several design suggestions were implemented in the next design iteration.

### 2.4 Phase IV Design

The final design, dubbed the Phase IV (see Figure 8), was the final product of the previous designs. Although it is very similar to the Phase IIIB, it has recessed wheels (i.e., notches cut into the base plate allowing the chassis to sit lower on the wheels, much like the wheel wells of a car), as well as a thinner top plate and motor mounts. This
increased the vertical space within the robot by just over half an inch, allowing more room for the required circuits and processors.

Figure 8: The Phase IV RoboCup Robot
3 Kinematics

The inverse kinematic model for the Phase IV was relatively simple to derive. As can be seen from Figure 9, the moving frame \([x_m, y_m]^T\) is located at the center of gravity of the robot. The two front wheels (1 and 2) are offset from \(y_m\) by 15°. Please note that the arrows projecting from each of the three wheels indicates the velocity vector generated by that wheel when rotating in the positive direction.

![Figure 9: Kinematic Diagram of Phase IV RoboCup Robot](image)

The inverse kinematic equations with respect to the mobile robot frame \([x_m, y_m]\) are as follows:

\[
R\dot{\theta}_1 = -\sin(\delta)x_m + \cos(\delta)y_m + L_4\dot{\phi} \tag{1}
\]
Please note that the constant $L_1$ refers to the distance from the center of gravity of the robot to the center of the two front wheels (1 and 2) along a radial path, and the constant $L_2$ refers to the distance from the center of gravity of the robot to the center of the rear wheel (3) along a radial path. The constant $\delta$ refers to the wheel orientation with respect to the mobile robot frame. For the Ohio University design, this value is $15^\circ$. $R$ is the radius of the wheels, $\dot{\theta}_i$ (for $i=1,2,3$) is the rotational velocity of each wheel, $x_m$ and $y_m$ are the Cartesian position of the robot in reference to the world frame in terms of the mobile frame, and $\phi$ is the orientation of the robot with respect to the world frame. Equations (1)-(3) allow us to form the Inverse Jacobian Matrix for the mobile robot system, as follows:

\[
R \dot{\theta}_2 = - \sin(\delta) \dot{x}_m - \cos(\delta) \dot{y}_m + L_1 \dot{\phi} 
\]

\[
R \dot{\theta}_3 = \dot{x}_m + L_2 \dot{\phi} 
\]

The rotation matrix used to change from the mobile robot frame $[x_m \ y_m]^T$ to the fixed world frame $[x_w \ y_w]^T$ is shown below.
The variable $\phi$ is the orientation of the robot with respect to the world frame.

Therefore, the Jacobian matrix with respect to the world frame would be expressed as:

$$w[R_m]^{-1} = \begin{bmatrix} w[R_m] & 0 \\ 0 & w[R_m]^{-1} \end{bmatrix} m[J]^{-1}$$
4 Inertial Dynamics

A dynamic model for the robot was difficult to derive. Although there was an existing derivation for a similar model (Watanabe et al, 1998), the lack of symmetry in the Ohio University wheel design caused complications. The derivation began with Newton's Second Law:

\[ M\ddot{X}_w = F_x \]  \hspace{1cm} (8)

\[ M\ddot{Y}_w = F_y \]  \hspace{1cm} (9)

\[ I_v\ddot{\phi} = M_I \]  \hspace{1cm} (10)

where \( M \) is the mass of the mobile robot, \( X_w \) and \( Y_w \) refers to the position with respect to the world frame, \( I_v \) is the moment of inertia of the mobile robot, \( \phi \) is the orientation of the robot with respect to the world frame, \( F_x \) and \( F_y \) are the Cartesian forces acting upon the robot with respect to the world frame, and \( M_I \) is the moment acting upon the center of gravity of the mobile robot.

Please note that the Cartesian dynamics equations are given with respect to the world frame, and need to be transformed to the moving frame (see Figure 9). This is accomplished by use of the Rotation Matrix shown in equation (6). Converting equations (8) and (9) into a more standard form reveals:

\[ M\ddot{S}_w = F_w \]  \hspace{1cm} (11)

where:

\[ M = \begin{bmatrix} M & 0 \\ 0 & M \end{bmatrix} \]  \hspace{1cm} (12)
Converting equation (11) into moving coordinate system terms (while remaining in reference to the inertial frame) requires the following transformation.

\[
M \left( \begin{array}{c} wR_m^T \dot{w} \end{array} \right) \dot{m} + \ddot{m} = f_m
\]  

(15)

where:

\[
s_m = \left( \begin{array}{c} x_m \\ y_m \end{array} \right)
\]  

(16)

\[
f_m = \left( \begin{array}{c} f_x \\ f_y \end{array} \right)
\]  

(17)

Solving equation (15) allows the dynamic properties of the mobile robot to be described as follows:

\[
M \left( \begin{array}{c} \ddot{x}_m - \dot{y}_m \dot{\phi} \end{array} \right) = f_x
\]  

(18)

\[
M \left( \begin{array}{c} \ddot{y}_m + \dot{x}_m \dot{\phi} \end{array} \right) = f_y
\]  

(19)

\[
I_{\phi} \ddot{\phi} = M_I
\]  

(20)
Please note that \( f_x \) and \( f_y \) refer to the Cartesian forces acting upon the mobile robot with respect to the world frame in terms of the mobile robot frame, and \( x_m \) and \( y_m \) refer to the Cartesian position with respect to the world frame in terms of the mobile robot frame.

The driving system dynamic model for each wheel is assumed to be given by (Saito et al, 1990):

\[
I_w \ddot{\theta}_i + c \dot{\theta}_i = ku_i - RD_i \tag{21}
\]

for \( i = 1, 2, 3 \). \( I_w \) is the moment of inertia of the wheel assemblies, \( \theta_i \) is the angular position of each wheel, \( c \) is the viscous friction factor of the wheel assembly, \( k \) is the driving gain factor (assumed to be equal to 1), \( u_i \) is the driving input torque, \( R \) is the radius of the wheels, and \( D_i \) is the driving force due to each wheel assembly.

\( f_x, f_y, \) and \( M_I \) from equations (18) - (20) are given by:

\[
f_x = -\sin(\delta)D_1 - \sin(\delta)D_2 + D_3 \tag{22}
\]

\[
f_y = \cos(\delta)D_1 - \cos(\delta)D_2 \tag{23}
\]

\[
M_I = (D_1 + D_2)L_1 + D_3L_2 \tag{24}
\]

As presented in Section 3 (equation (5)), the following matrices give the inverse kinematic relationships with respect to the mobile robot frame:

\[
\begin{bmatrix}
\dot{\theta}_1 \\
\dot{\theta}_2 \\
\dot{\theta}_3 \\
\end{bmatrix} = \frac{1}{R} \begin{bmatrix}
-\sin(\delta) & \cos(\delta) & L_1 \\
\sin(\delta) & \cos(\delta) & L_1 \\
1 & 0 & L_2 \\
\end{bmatrix} \begin{bmatrix}
\dot{x}_m \\
\dot{y}_m \\
\phi \\
\end{bmatrix} \tag{25}
\]
Given the previous equations, it is now possible to represent the dynamic properties of the system in the standard form of:

\[ [P][\ddot{x}_m] + \{N(X_m, \dot{X}_m)\} = [A][U] \]  

(26)

From equation (26), the equations of motion can be represented as follows.

\[ \{\ddot{x}_m\} = [P]^{-1}([A][U] - \{N(X_m, \dot{X}_m)\}) \]  

(27)

To do this, first solve equations (18)-(20) for the respective accelerations.

\[ \ddot{x}_m = \dot{y}_m \dot{\phi} + \frac{f_x}{M} \]  

(28)

\[ \ddot{y}_m = -\dot{x}_m \dot{\phi} + \frac{f_y}{M} \]  

(29)

\[ \ddot{\phi} = \frac{M_1}{I_v} \]  

(30)

Next, substitute equations (22)-(24) into equations (28)-(30).

\[ \ddot{x}_m = \dot{y}_m \dot{\phi} + \frac{1}{M}(-\sin(\delta)D_1 - \sin(\delta)D_2 + D_3) \]  

(31)

\[ \ddot{y}_m = -\dot{x}_m \dot{\phi} + \frac{1}{M}(\cos(\delta)D_1 - \cos(\delta)D_2) \]  

(32)

\[ \ddot{\phi} = \frac{1}{I_v}((D_1 + D_2)L_1 + D_3L_2) \]  

(33)

Solve equation (21) for \( D_i \) and substitute into equations (31)-(33).
Substitute equation (25) and its derivative into equations (34)-(36) to give the following equations of motion in the format of equation (26):

\[
\ddot{x}_m = \dot{y}_m \dot{\phi} + \frac{1}{MR} \left( -\sin(\delta) \left( ku_1 - \frac{I_w R \dot{\theta}_1}{R} - \frac{c R \dot{\theta}_1}{R} \right) - \sin(\delta) \left( ku_2 - \frac{I_w R \dot{\theta}_2}{R} - \frac{c R \dot{\theta}_2}{R} \right) \right)
\]

\[
\ddot{y}_m = -\ddot{x}_m \dot{\phi} + \frac{\cos(\delta)}{MR} \left( ku_1 - \frac{I_w R \dot{\theta}_1}{R} - \frac{c R \dot{\theta}_1}{R} - ku_2 + \frac{I_w R \dot{\theta}_2}{R} + \frac{c R \dot{\theta}_2}{R} \right)
\]

\[
\ddot{\phi} = \frac{M_1}{I_v R} \left[ \left( ku_1 - \frac{I_w R \dot{\theta}_1}{R} - \frac{c R \dot{\theta}_1}{R} + ku_2 - \frac{I_w R \dot{\theta}_2}{R} - \frac{c R \dot{\theta}_2}{R} \right) L_1 \right]
\]

\[
[R] = \begin{bmatrix}
MR^2 + 2I_w \sin^2(\delta) + I_w & 0 & -2I_w \sin(\delta) L_1 + I_w L_2 \\
MR^2 & 0 & MR^2 \\
0 & MR^2 + 2I_w \cos^2(\delta) & 0 \\
-2I_w L_1 \sin(\delta) + L_2 I_w & 0 & I_v R^2 + I_w L_2^2 + 2I_w L_4^2 \\
I_v R^2 & 0 & I_v R^2
\end{bmatrix}
\]

\[
\{N_1\} = \begin{bmatrix}
\frac{(2c - c \cos(2\delta)) \dot{x}_m + (c L_2 - 2c \sin(\delta))L_1 \dot{\phi}}{MR^2} - \dot{y}_m \dot{\phi} \\
\frac{2c \cos^2(\delta) \dot{y}_m + \dot{x}_m \dot{\phi}}{MR^2} \\
\frac{(-2L_1 c \sin(\delta) + L_2 c) \dot{x}_m + \left(\frac{2c L_1^2 - c L_2^2}{I_v R^2}\right) \dot{\phi}}{I_v R^2}
\end{bmatrix}
\]
\[
[A_4] = \begin{bmatrix}
-\sin(\delta)k & -\sin(\delta)k & k \\
\frac{MR}{\cos(\delta)k} & \frac{MR}{\cos(\delta)k} & \frac{MR}{0} \\
\frac{L_1k}{I_yR} & \frac{L_1k}{I_yR} & \frac{L_2k}{I_yR}
\end{bmatrix}
\]

These equations can also be represented in a state space equation format as follows:

\[
\dot{x}_m = \left(\frac{2L_1I_w \sin(\delta) - L_2I_w}{MR^2 + 2L_1I_w \sin^2(\delta) + I_w}\right)\dot{\phi} + \left(\frac{-c - 2c \sin^2(\delta)}{MR^2 + 2L_1I_w \sin^2(\delta) + I_w}\right)\ddot{\theta}_m + \\
\left(\frac{-cL_2 + 2cL_1 \sin(\delta)}{MR^2 + 2L_1I_w \sin^2(\delta) + I_w}\right)\dot{\phi} + \left(\frac{MR^2}{MR^2 + 2L_1I_w \sin^2(\delta) + I_w}\right)\dot{\theta}_m\dot{\phi} + \\
\left(\frac{-kR \sin(\delta)}{MR^2 + 2L_1I_w \sin^2(\delta) + I_w}\right)u_1 + \left(\frac{-kR \sin(\delta)}{MR^2 + 2L_1I_w \sin^2(\delta) + I_w}\right)u_2 + \\
\left(\frac{kR}{MR^2 + 2L_1I_w \sin^2(\delta) + I_w}\right)u_3
\]

\[
\dot{\theta}_m = \left(\frac{-2c \cos^2(\delta)}{MR^2 + 2L_1I_w \cos^2(\delta)}\right)\ddot{\theta}_m + \left(\frac{-MR^2}{MR^2 + 2L_1I_w \cos^2(\delta)}\right)\ddot{\phi}_m + \\
\left(\frac{kR \cos(\delta)}{MR^2 + 2L_1I_w \cos^2(\delta)}\right)u_1 + \left(\frac{-kR \cos(\delta)}{MR^2 + 2L_1I_w \cos^2(\delta)}\right)u_2
\]

\[
\ddot{\phi} = \left(\frac{2L_1I_w \sin(\delta) - L_2I_w}{I_yR^2 + I_wL_2^2 + 2I_wL_1^2}\right)\ddot{x}_m + \left(\frac{2L_1c \sin(\delta) - L_2c}{I_yR^2 + I_wL_2^2 + 2I_wL_1^2}\right)\ddot{\theta}_m + \\
\left(\frac{-2cL_2^2 - cL_2^2}{I_yR^2 + I_wL_2^2 + 2I_wL_1^2}\right)\ddot{\phi} + \left(\frac{L_1kR}{I_yR^2 + I_wL_2^2 + 2I_wL_1^2}\right)(u_1 + u_2) + \\
\left(\frac{L_2kR}{I_yR^2 + I_wL_2^2 + 2I_wL_1^2}\right)u_3
\]
As can be seen in equations (40) and (42), there are accelerations present in the equations other than those desired. This can be solved by substituting equations (40) and (42) into each other, giving the final form of the equations in state space format. Note that the acceleration in the y direction (equation (41)) remains unchanged.

\[
\dot{x}_m = \frac{1}{\beta}\begin{pmatrix}
-2c_l I_w L_1^2 - c_l R^2 - 4 I_w L_1 L_2 c \sin(\delta) - 2c_l I_w L_2^2 \\
2c_l R^2 \sin^2(\delta) - 2c_l I_w L_2^2 \\
-c L_2 I_v R^2 + 2c L_4 I_v R^2 \\
2MR^2 I_w L_1^2 + MR^4 I_v + MR^2 I_w L_2^2 \\
- I_w L_2 L_4 kR - kR^3 I_v \sin(\delta) - kRI_w L_2^2 \sin(\delta) \\
- I_w L_2 L_4 kR - kR^3 I_v \sin(\delta) - kRI_w L_2^2 \sin(\delta) \\
2RkI_w L_1^2 + 2I_w L_1 L_2 kR \sin(\delta) + R^3 kI_v
\end{pmatrix}\dot{\phi} +
\dot{\phi} = \frac{1}{\beta}\begin{pmatrix}
-L_2 cMR^2 + 2L_4 cMR^2 \sin(\delta) \\
-c L_2 MR^2 - 2c L_4 MR^2 - 2c I_w L_1^2 - 2c I_w L_2^2 \sin^2(\delta) \\
-4I_w L_1 L_2 c \sin(\delta) \\
-L_2 I_w MR^2 + 2L_4 I_w MR^2 \sin(\delta) \\
L_4 kR^3 M + L_4 kRI_w + L_2 kRI_w \sin(\delta) \\
L_4 kR^3 M + L_4 kRI_w + L_2 kRI_w \sin(\delta) \\
2L_4 I_w kR \sin(\delta) + L_2 kR^3 M + 2L_2 I_w kR \sin^2(\delta)
\end{pmatrix}\mu_1 +
\mu_2 +
\mu_3
\]

where:

\[
\beta = I_w L_v R^2 + 2I_w L_1^2 + 2MR^2 I_w L_1^2 +
MR^2 I_w L_2^2 + MR^4 I_v + 2I_w L_v R^2 \sin^2(\delta) +
2I_w L_2^2 \sin^2(\delta) + 4I_w L_4 L_2 \sin(\delta)
\]
The equations of motion can also be represented in the standard format of equation (26) as follows.

\[
[P_2] = \begin{bmatrix}
\beta & 0 & 0 \\
0 & MR^2 + 2I_w \cos^2(\delta) & 0 \\
0 & 0 & \beta
\end{bmatrix}
\] (46)

\[
\{N_2\} = \begin{bmatrix}
2cI_wL_1^2 + cI_vR^2 + 4I_wL_1L_2c\sin(\delta) + 2cI_vR^2 \sin^2(\delta) + 2cI_wL_2^2 \sin^2(\delta) \dot{k}_m + \\
(cL_2I_vR^2 - 2cL_1I_vR^2 \sin(\delta))\dot{\phi} + \left(-2MR^2I_wI_1^2 + MR^4I_v + MR^2I_wL_2^2\right)\dot{y}_m \phi \\
\left(2c \cos^2(\delta)\right)\dot{k}_m + \left(MR^2\right)\dot{k}_m \phi \\
\left(L_2cMR^2 - 2L_1cMR^2 \sin(\delta)\right)\dot{k}_m + \left(L_2I_wMR^2 - 2L_1I_wMR^2 \sin(\delta)\right)\dot{y}_m \phi + \\
\left(cL_2^2MR^2 + 2cI_1^2MR^2 + 2cI_wL_1^2 + 2cI_wL_2^2 \sin^2(\delta) + 4I_wL_1L_2c\sin(\delta)\right)\phi
\end{bmatrix}
\] (47)

\[
[A_2] = \begin{bmatrix}
\psi & \psi & \kappa \\
k \cos(\delta) & -k \cos(\delta) & 0 \\
\mu & \mu & \sigma
\end{bmatrix}
\] (48)

where:

\[
\psi = -I_wL_2L_1kR - kR^3I_v \sin(\delta) - kRI_wL_2^2 \sin(\delta)
\] (49)

\[
\kappa = 2RkI_wL_1^2 + 2I_wL_1L_2kR \sin(\delta) + R^3kI_v
\] (50)

\[
\mu = L_1kR^3M + L_1kRI_w + L_2I_wkR \sin(\delta)
\] (51)

\[
\sigma = 2L_1I_wkR \sin(\delta) + L_2kR^3M + 2L_2kRI_w \sin^2(\delta)
\] (52)
5 Simple Simulation

The dynamic system for the Phase IV RoboCup robot was created in Simulink, and deals only with the mechanical properties of the system. In other words, it does not concern itself with the motor and circuit dynamics, dealing with only the masses and inertias of the mechanical system.

5.1 The Dynamic Model

A simplified version of the dynamic model is shown below in Figure 10, the actual Simulink diagram used is given in Appendix B. As can be seen from the figure, a desired Cartesian position and orientation (pose) with respect to the world frame is given to the system, which transforms it into the required wheel rotations in the Inverse Kinematics box. This is then summed in the summing junction with the actual wheel rotations, producing the error. Three independent Proportional-Integral-Derivative (PID) wheel controllers are used to control the commanded wheel rotations.

![Figure 10: Block Diagram of Simplified Dynamic Model](image-url)
The constant parameters used in the equations of motion (equations (41), (43)-(44)) are as follows:

\[ R = 0.0254 \text{ m} \]
\[ L_1 = 0.0742 \text{ m} \]
\[ L_2 = 0.0787 \text{ m} \]
\[ M = 2.5 \text{ kg} \]
\[ k = 1 \]
\[ I_w = 0.00014 \text{ kgm}^2 \]
\[ c = 5.983 \times 10^{-6} \text{ kgm}^2/\text{s} \]
\[ I_v = 0.003692 \text{ kgm}^2 \]
\[ \delta = 15^\circ \]

5.2 Step Input Results

The first simulation displays the response of the system to a simple step input for the \(X\) and \(Y\) position, while maintaining a constant orientation \((\phi = 90^\circ)\). The simulated results can be seen below in Figure 11.

![Figure 11: Step Response of Simplified Dynamic Model](image)
As can be seen from Figure 11, the overshoot in the X direction is 14.36%. The Y direction overshoot is 10.97%. The settling time for both is 0.08 s. The orientation fluctuates slightly when the step takes affect, but quickly returns to the desired value of 1.57 rad (90°).

5.3 Ramp and Hold Input Results

For the next simulation, a ramp input was given as the desired $X$ and $Y$ positions over time, while maintaining a constant orientation ($\phi = 90^\circ$ or 1.57 rad). This allowed for the calibration of the PID controllers used for the wheel independent control by trial and error using the simulation. The results can be seen below in Figure 12.

![Figure 12: Ramp Input Response of Simplified Dynamic Model](image-url)
As can be seen in the first two plots of Figure 12, the simulated results follow the desired results quite closely, with only a 0.8% overshoot in the X direction, and 0.9% overshoot in the Y direction. There was very little steady state error for these two values (less than 0.1%). The simulated values for \( \phi \) fluctuated, although the magnitude of these fluctuations was quite small, as can be seen from the last plot in Figure 10. The greatest error over the time span for \( \phi \) was approximately 0.035 rad (2.01°).

5.4 3rd Order Path Planning Results

The third input into the simulation was a third order path for \( X \), \( Y \), and \( \phi \). Using third order path planning, a variable is changed over time according to a 3rd order polynomial, as follows (Craig, 1989):

\[
z = A_0 + A_1 t + A_2 t^2 + A_3 t^3
\]

where \( z \) is a sample variable, \( A_i \) for \( i=0,1,2,3 \) are constants, and \( t \) is time. Given the start point \( (z_0) \), the end point \( (z_f) \), the initial and final velocities \( (dz/dt) \), and the final time \( (t_f) \) allows the constants to be solved. This gives a more realistic path for the robot to follow.

For this case, the desired velocity at the start and end of the path for all directions, as well as initial and final rotational velocity, was zero. Over the five second time period, the \( X \) position will change from 2 m to 0.5 m, the \( Y \) position will change from 2 m to 2.5 m, and the orientation \( \phi \) will change from 0° to -90°. Using these parameters, the three equations of desired motion are as follows.

\[
X_w(t) = 2 - 0.18t^2 + 0.024t^3
\]

\[
Y_w(t) = 2 + 0.06t^2 - 0.008t^3
\]
\[ \phi(t) = -0.1885 t^2 + 0.0251 t^3 \]  

(56)

The simulated movement of the RoboCup robot to the desired 3rd order inputs is shown below in Figure 13.

![Figure 13: 3rd Order Input Response of Simplified Dynamic Model](image)

As can be seen from the figure, the simulated path follows the desired path exactly, more closely than the ramp input. This is due to the smoothness of the desired path, without any sudden stops, starts, or jumps. The wheel rotation in radians is shown below in Figure 14.
As can be seen from the simulated data, the RoboCup robot follows the 3rd order path exactly. For the real-world hardware, these 3rd order inputs are recommended. This methodology can be extended to 5th order polynomials if zero acceleration is desired at the beginning and end of the path (Craig, 1989).
6 Complex Simulation

The complex dynamic model was also created in Simulink, and again deals only with the physical properties of the system. This model more closely mimics the proposed control algorithm for controlling the robot in the RoboCup competition, as developed by the Ohio University RoboCup controls team.

6.1 The Dynamic Model

A schematic of the complex dynamic model is shown below in Figure 15, while the actual Simulink diagram is given in Appendix C. As can be seen in the figure, a desired $\dot{X}_w$, $\dot{Y}_w$ velocity vector, as well as an orientation ($\phi$) is inputted into the system. Please note that only an orientation is inputted, not a rotational velocity ($\dot{\phi}$). The robot is controlled by two control loops, a wheel independent controller (the smaller loop) and a camera loop (the larger loop). Three independent Proportional-Integral-Derivative controllers (Motor Minds) are used for the smaller loop to control the commanded wheel positions, while camera feedback from the global vision system is used for the larger loop to control the Cartesian pose.

Figure 15: Diagram of Complex Dynamic Model
As can be seen from the figure, in addition to the two feedback loops, there is also a feed forward loop, where the theoretically necessary wheel rotations to achieve the desired Cartesian motion are passed forward, bypassing the Cartesian error calculation resulting from the global vision system loop. Please note that in the first Integrator box from the left, only $\dot{X}_w$ and $\dot{Y}_w$ are integrated, the $\phi$ is passed through unchanged.

The constant parameters used for this model are the same as those used for the simple dynamic model (see section 5.1).

6.2 Ramp and Hold Input Results

The first input given to the complex dynamic system is a ramp input in the $X$ and $Y$ directions, while maintaining a constant orientation ($\phi=90^\circ$, 1.57 rad). The results are shown below in Figure 16.

![Figure 16: Ramp Input Response of Complex Dynamic Model](image)
As can be seen in the Figure, the X and Y positions closely mimic the desired values. Again, there is some small fluctuation in the orientation.

6.3 3rd Order Path Planning Results

The second input given to the complex dynamic model was another third order polynomial path. Please see section 5.4 for a discussion of this input. Over the 5 second travel interval, the X position varied from 0 m to -1.5 m, the Y position varied from 0 m to 0.5 m, and the $\phi$ orientation varied from 0° to -90° (-1.57 rad). The results from this input are shown below in Figure 17.

![Figure 17: 3rd Order Input Response of Complex Dynamic Model](image)

As can be seen in Figure 17, the simulated motion closely mimics that of the desired motion.
6.4 Circle Path Results

The final input given to the complex dynamic model was a circular path with a 1.6 m diameter. This was created by giving the simulation a sine velocity vector in the X direction, and a cosine velocity vector in the Y direction. While moving on this circular trajectory, the desired orientation would remain a constant 0°. The desired equations for Cartesian position and velocity are shown below.

\[
\begin{align*}
X_w(t) &= -0.8 \cos(1.25t) \\
Y_w(t) &= 0.8 \sin(1.25t) \\
\phi(t) &= 0
\end{align*}
\]  

(57) \hspace{1cm} (58) \hspace{1cm} (59)

The results are shown below in Figure 18.

![Figure 18: Trajectory of Complex Dynamic Model with a Circular Input](image)
As can be seen from Figure 18, the simulated trajectory follows the desired trajectory quite closely. A plot of the orientation of the robot versus time is shown below in Figure 19.

![Figure 19: Orientation of Complex Dynamic Model with a Circular Input](image)

As can be seen from Figure 19, the orientation also fluctuates in a sinusoidal manner, with a maximum error of 0.0016 rad (0.0917°). Shown below in Figure 20 are plots of the Cartesian and radial velocities versus time. As can be seen in the figure, the simulated $X$ and $Y$ velocities follow the desired values precisely, while there is some fluctuation in the radial velocity. The maximum radial velocity error is 0.0024 rad/s.
Figure 20: Wheel Velocities of the Complex Dynamic Model with a Circular Path Input
7 Slip Simulation and Experimental Results

From the initial experimental data obtained for the omni-directional robot, it was found that the two preceding methods of simulation were not accurate enough. Through experimental observation, it was found that the inertial disturbances of the robot were minor in comparison to the slipping error from the wheels. Therefore, a new simulation was developed with the assistance of Dr. Paolo Gallina and Ph.D. candidate Giulio Rosati, both from the University of Padova, Italy. This takes into account not only wheel slippage, but also disturbances caused by the spaces between the wheels outer rollers.

7.1 Theoretical Derivation

The omni-directional robot model is shown below in Figure 21. The modeling of the $i^{th}$ wheel is shown below in Figure 22.
As seen in Figure 21, $\mathbf{\hat{r}}_i$ is the unit vector of the position of the $i^{th}$ with respect to the center of the mobile robot, while $\mathbf{s}_i$ is the unit vector in the opposite direction of the velocity vector generated by the $i^{th}$ wheel when rotated in the positive direction. From Figure 22, $P_i$ (for $i = 1,2,3$) is the contact point between the $i^{th}$ wheel and the ground. The velocity vector $v_i$ is denoted as:

$$v_i = v_G + \dot{\phi} \times r_i + v_{ir}$$

(60)

where $v_G$ is the vehicle center of mass translational velocity with respect to the global frame, $\dot{\phi}$ is the vehicle rotational velocity with respect to the global frame, $r_i$ is the position vector of the wheel center with respect to the moving frame expressed in the global frame, and $v_{ir}$ is the peripheral wheel speed with respect to the local frame, expressed in the inertial frame. Please note that when $v_i$ is equal to zero, there is no slipping.
\( \mathbf{v}_{ir} \) can be expressed as a function of the wheel angular velocity vector \( \hat{\mathbf{\theta}}_i \) and the wheel radius vector \( \mathbf{R}_i \) as shown below.

\[
\mathbf{v}_{ir} = \hat{\mathbf{\theta}}_i \times \mathbf{R}_i
\]  

(61)

Wheel/ground friction must be considered in both the direction of wheel rotation (defined as \( \hat{s}_i \), see Figure 21) and the transverse direction (defined as \( \hat{r}_i \), a unit vector parallel to the position vector of the wheel center with respect to the moving frame). Both defined vectors are expressed with respect to the global frame.

For friction in the direction of wheel rotation, the sliding velocity of the \( i^{th} \) wheel is obtained by the dot product of the total point \( P_i \) velocity into the \( \hat{s}_i \) unit direction as shown.

\[
\mathbf{v}_i \cdot \hat{s}_i = \mathbf{v}_G \cdot \hat{s}_i + (\hat{\phi} \times \hat{r}_i) \cdot \hat{s}_i + R_i \hat{\theta}_i
\]  

(62)

Please note that \( R_i \hat{\theta}_i \) is the scalar product of equation (61) in the unit wheel direction.

To convert wheel positions and rotations from the local frame to the global (world) frame, the rotation matrix from equation (6) was used. Therefore, equation (62) becomes:

\[
\mathbf{v}_i \cdot \hat{s}_i = \mathbf{v}_G \cdot w \mathbf{R}_m \hat{s}_i + (\hat{\phi} \times w R_m \hat{r}_i) \cdot w R_m m \hat{s}_i + R_i \hat{\theta}_i
\]  

(63)

Based on the assumption that the weight of the mobile robot is equally displaced on all three wheels, the friction force is given as:

\[
\mathbf{F}_i = -\mu (\mathbf{v}_i \cdot \hat{s}_i) \frac{Mg}{3} \hat{s}_i
\]  

(64)
where $\mu$ is a function representing the friction coefficient versus the wheel direction sliding velocity. The dynamic equations of motion are:

$$Mx = \sum_{i=1}^{3} \hat{x}_w \cdot F_i$$

(65)

$$My = \sum_{i=1}^{3} \hat{y}_w \cdot F_i$$

(66)

$$I\ddot{\phi} = \sum_{i=1}^{3} \hat{z}_w \cdot (r_i \times F_i)$$

(67)

where $\hat{x}_w$, $\hat{y}_w$, $\hat{z}_w$ are the unit vectors of the inertial frame. The dynamic equations of the form $\ddot{X} = f(X, \dot{X})$ are shown below.

$$\begin{bmatrix} \dot{x}_w \\ \dot{y}_w \\ \dot{\phi} \end{bmatrix} = f(x_w, y_w, \phi, \dot{x}_w, \dot{y}_w, \dot{\phi})$$

(68)

The vector expressions for the right hand side of equation (67) are shown below.

$$\begin{bmatrix} \dot{z}_w \\ \dot{\gamma}_w \\ \dot{\gamma}_w \end{bmatrix} = \frac{1}{M} \sum_{i=1}^{3} \hat{x}_w \cdot \left[ a\mu \left( v_i \cdot wR_m^m \hat{s}_i + (\dot{\phi} \times wR_m^m \hat{r}_i) \right) wR_m^m \hat{s}_i + R_i \dot{\theta}_i \right] wR_m^m \hat{s}_i$$ 

(69)

where:

$$a = \frac{Mg}{3}$$

(70)
As previously stated, friction must also be accounted for in the transverse ($\hat{r}_i$) direction. To include this, equation (64) is modified as follows.

$$\bar{F}_i = -\frac{Mg}{3} (\mu(v_i \cdot \hat{s}_i)\hat{s}_i + \mu_T(v_i \cdot \hat{r}_i)\hat{r}_i)$$

The simplified equations for $\mu(.)$ and $\mu_T(.)$ are shown below. These artificial functions were used to represent the friction coefficients stably in the simulation, avoiding the computational problems that may occur when using a discontinuous function at zero sliding velocity.

$$\mu(v) = \mu_{\text{max}} \frac{2}{\pi} \arctan(kv)$$

$$\mu_T(v) = \mu_{T\text{max}} \frac{2}{\pi} \arctan(kv)$$

A graphical representation of equations (73) and (74) is shown below in Figure 23.
Figure 23: Friction Coefficients vs. Sliding Velocity

Please note that the positive friction coefficient is defined to correspond with positive sliding velocity. The opposite sign behavior (Coulomb friction act opposite the sliding direction) is taken into account in equations (64) and (72).

From early simulation results and experimental data, it was discovered that for an accurate simulation, the spaces between the rollers must be accounted for. This was done by simply assuming that 92% of the wheel circumference consists of rollers, with the remaining 8% consisting of the spaces between the rollers. The simulation then calculated the wheel position, and when the wheel was positioned so that the spaces were in contact with the ground, $\mu_{r_{\text{max}}}^T$ and $\mu_{\text{max}}$ were replaced by $\mu_{\text{space}_{\text{max}}}$. 
With the sliding coefficients now calculated, a simulation is fairly simple to create.

7.2 Experimental Set-up

The first step taken to create the experimental results was the calculation of experimental values for $\mu_{\text{max}}$, $\mu_{T_{\text{max}}}$, and $\mu_{\text{space}_{\text{max}}}$ for both the flat and carpeted surfaces for use in the simulation. This was done by placing a mass supported by the wheels on pieces of wood covered in each of the two surfaces. The piece of wood was then raised until the mass began to slip. Once slippage occurred, the angle was recorded. Knowing the angle of the board and the mass of the object, the three values for friction $(\mu_{\text{max}}, \mu_{T_{\text{max}}}, \mu_{\text{space}_{\text{max}}})$ were calculated for both the flat surface and the carpeted surface.

To control the robot during the experimental trials, WinCon 3.1 was used. This enabled us to use a Quanser Multi-Q 3 board to control the angular velocities of the motors through a feedback loop. WinCon 3.1 allowed the use of a Simulink environment to control the motor velocities. The experimental set-up is shown below in Figure 24. Please note that the wires atop the robot must be manually held, otherwise they cause discrepancies in the data. Attaching a light pencil to the center of mass of the robot created the experimental data for the flat surface. This was not feasible for the carpeted data, so only the end points and final orientations were recorded.
7.3 Simulation and Experimental Results

The Simulink simulation is shown below in Figure 25. As can be seen from the figure, the desired velocities and orientation \( (\dot{x}, \dot{y}, \dot{\phi}, \phi) \) are inputted into the inverse kinematics, giving the three desired wheel velocities \( (\dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3) \). These are then used to solve the dynamic equations (equations (68) and (69)). The expanded model, including Matlab code, is located in Appendix D.

![Figure 25: The Dynamic Model with Slip and Roller Gaps](image-url)
The input given to the simulation, as well as the actual model, was to travel 0.4 m in the positive $x$ direction in 3.25 seconds. The first simulation did not account for the spaces between the wheels. The results of this simulation are shown below in Figure 26.

As can be seen in the figure, the experimental results do not match the expected simulated data. It was deduced that this was because of the spaces between the rollers. When these points were in contact with the surface, the transverse friction would be much higher than when the rollers were in contact. We compensated our simulation for this, and the results are shown below in Figure 27.
Figure 27: Simulated and Experimental Results for the Omni-Directional Mobile Robot with Rollers

As can be seen in the figure, the simulation matched the experimental data closely after compensating for the spaces between the rollers. As shown, the results of the mobile robot are quite poor on the carpeted surface, with orientation error at the end of the path of 28° (compared to a simulated orientation error of 29°). The orientation error at the end of the path while on the flat surface is much smaller, ranging from 9.2° to 14.6°. The simulated orientation error while on the flat surface was calculated as 11.2°.
8 Conclusions

This Thesis describes the development of an omni-directional mobile robot using an asymmetrical universal wheel configuration. After a brief history of mobile robots, omni-directional robots, and the RoboCup competition, the design process was discussed in detail. Next, the inverse kinematic equations, and the dynamic equations of motion were derived, and simple, complex, and slip simulations were created. The simple dynamic model consisted of a standard motor speed feedback control. The complex dynamic model closely mimicked the tentative control algorithm that would be used in the RoboCup competition. The slip simulation disregarded the majority of the inertial disturbances, concentrating instead on the slip dynamics between the wheels and the surface. The most accurate of the models (slip simulation) compared favorably with the experimental data. Once the spaces between the rollers were compensated for, the results of the simulations closely followed the experimental results.

Future plans include the improvement of the simulation, and experimentally validating the simulation for a wider variety of motions. The most pressing work involved, however, is the improvement of the mobile robot's performance on the carpeted surface. Through simulation and experimental results, it has been shown that the robot behavior is quite poor on the carpet. Several methods of improving the motion are currently under discussion, such as filing away as much of the material between the rollers as possible and increasing the friction coefficient of the wheel in the normal direction, making it greater than the transverse friction coefficient. Also, an advanced controller to compensate for wheel slippage could be the answer. In addition to this, the
RoboCup competition itself is rapidly approaching, and practical preparations for the event must be completed.
References


RoboCup 2001 Rules Draft Website


Sony Legged Robot League Website


Appendix A - Rules of the RoboCup Small-Size League

1. Playing Field

   a. Surface

       The floor material is green felt mat or carpet. The specific color and texture is not specified and may vary from competition to competition.

       The surface underneath the carpet will be level and hard. Examples include: cement, linoleum, hardwood flooring, plywood, ping-pong tables and particle board; carpeted or cushioned surfaces are not allowed. Every effort shall be made to ensure that the surface is flat, however, it is up to individual teams to design their robots to cope with slight curvatures of the surface.

   b. Markings

       Markings on the field are 1 cm wide white lines. Markings include: a white center circle, center line, and defense zone line. Penalty kick position and free kick positions are also painted on the mat as 1 cm diameter dots.

   c. Dimensions
Dimensions of the playing surface are 152.5 cm by 274 cm. Positions and dimensions of the markings and goals are given in the figure below (see Figure 28). The height of the surface above the floor is not specified.

- field x: 274 cm, field y: 152.5 cm
- center circle: 45 cm
- goal x: 18 cm, goal y: 50 cm
- defense zone x: 22.5 cm, defense zone y: 100 cm
- penalty spot: 45 cm from the goal line
- free kick spots: 15 cm from the x-wall, 37.5 cm from the y-wall

Figure 28: Dimensions of the RoboCup Playing Field

d. Walls

Walls around the field are made from pieces of 5 cm x 5 cm wooden bars diagonally cut in half, i.e., they are 5 cm high at the outside and 5 cm wide, forming a 45° slope which faces toward the playing field.
There will be 10 cm high vertical walls behind each defense zone and 15 cm high vertical walls behind each goal area. The walls behind the goals will be painted one of two appropriate colors (yellow or blue).

e. Secondary Walls

A 10 cm tall secondary white wall will be placed around the field at a distance of 15 cm from the boundary of the field.

f. Goals

The width of each goal is 50 cm. The goal is 18 cm deep. The wall continues behind the goal but increases to a height of 15 cm. There is no safety net over the goal, nor is there a horizontal goal bar. The wall behind one goal will be painted yellow, the other will be painted blue. The floor of the goal consists of the same carpet as the rest of the playing field. It should be noted that a robot may use the area behind the goal.

g. Defense Zone

A defense zone is created around each of the goals. It extends from the front of the goal to 22.5 cm into the field. The zone is 100 cm wide. Entry into the defense zone is regulated by rules outlined below in the section regarding fouls.

2. Robots
a. Shape

Any autonomous robot can participate in the F-180 league provided it complies to these criteria.

b. Kicking Devices

Kicking devices are permitted.

c. 18 cm Rule

The robot must fit within an 18 cm diameter cylinder.

d. Height

If the team is using a global vision system robot height is restricted to 15 cm or less. Otherwise the robot height must be 22.5 cm or less. Height restrictions do not apply to radio antennas and visual markers.

e. Robot Colors and Markers

Before a game, each of the two teams has a color assigned, namely yellow or blue. The yellow team attacks the yellow goal and the blue team attacks the blue goal. Each team must be able to use either yellow or blue markers.

Circular markers of the assigned color must be mounted on the top of the robots. The center of the marker must be located in the visual center of a
robot when viewed from above. The markers must have a diameter of 4 cm.

Robots may use black and white coloring without restriction. The use of markers of other colors is not encouraged, but allowed provided that these colors do not conflict with other colors. The colors of additional markers should be as different as possible from the reserved colors, specifically ball orange, field green, marker yellow and marker blue. The suitability of any additional marker is finally decided by the rules committee upon inspection of the robots before a tournament. Such markers may be situated on any surface of the robot.

f. Locomotion

Robot wheels (or other surfaces that contact the playing surface) must be made of material that does not harm the playing surface. Metal spikes and Velcro are specifically prohibited.

g. Inspection

The robots will be examined by the referee before the game to ensure that they meet these constraints

As one test for area compliance, the robots must fit into an 18 cm diameter cylinder. Whilst being inspected each robot must be at its
maximum size; anything that protrudes from the robot must be extended.
Except as allowed under "conflict resolution" below.

Robots will also be inspected for conformance with the official coloring rules. If, in the opinion of the inspecting official a robot's markings are not close enough to the official colors, the team will be asked to replace their markers with material provided by competition officials.

h. Team

A team shall consist of no less than 1 and no more than 5 robots.

3. Ball

An orange golf ball provided by the RoboCup organization shall be used.

4. Pre-Game Setup

Organizers will make every effort to provide the teams access to the competition area at least two hours before the start of the competition. They will also strive to allow at least one hour of setup time before each game. Participants should be aware, however, that conditions may arise where this much time cannot be provided.

5. Length of the Game
The games consist of the first half, break, and the second half; each is 10 minutes. All time for stoppages will be added to the end of the half they occur in.

Each team will be allowed some setup time at the start of the game. Before the beginning of the second half,

1. Teams must switch sides.
2. Teams must switch team color markers unless both teams agree not to.
3. If both teams use the same set of wireless frequencies, they must switch frequencies unless both teams agree not to.

6. Timeouts/Delays of Game

Each team will be allocated three timeouts at the beginning of the game. A total of 15 minutes is allowed for all timeouts (e.g. a team may take two one-minute timeouts followed by one 13 minute timeout). In case a team is not ready to start at the scheduled time, they may use their timeouts to delay the game up to 15 minutes.

During a game, timeouts will only be granted during a break in play.

7. Substitutions and Removal of Damaged Robots

In general, substitutions are only allowed for damaged robots during a break in play. However, if in the opinion of the referee, a damaged robot is likely to cause
serious harm to humans, other robots or itself the referee will stop the game immediately and have the damaged robot removed. In this case, the game will be restarted with a free kick for the opposing team (the team that did not have the damaged robot). If there is no immediate danger however, the referee may allow the game to continue.

To replace a robot by substitute the following conditions must be observed:

a) A substitution can only be made during a stoppage of play.

b) The referee is informed before the proposed substitution is made.

c) The substitute is placed on the field after the robot being replaced has been removed.

d) The substitute is placed on the field in the position from which the damaged robot was removed.

8. Wireless Communication

In order to avoid interference, a team should be able to select from two carrier frequencies before the match. The type of wireless communication shall follow legal regulations of the country where the competition is being held. Compliance with local laws is the responsibility of the competing teams, not the RoboCup organization.

The use of a global vision system or an external distributed vision system is permitted, but not required, to identify and track the position of robots and balls. This is achieved by using one or more cameras.

Cameras positioned above the field will be mounted on a beam suspended from the ceiling. The beam will be positioned 3 meters above the field. If both teams agree, and the hosting facilities allow it, another height may be used. Cameras may not protrude more than 15 cm below the bottom of the beam. The placement of cameras is performed on a game by game basis, and the teams choose camera positions by tossing a coin to find which team places a camera first. The use of a global vision system shall be advertised at the time of registration, and the detailed arrangements shall be discussed with the RoboCup organizing committee.

The local organizer will inform all participants of the camera attachments required to use the beam provided.

10. Lighting

A description of the lighting will be provided by the local organizer. The intent is to provide 700-1000 LUX uniform light. But this cannot be guaranteed.

11. Goal Keepers
Each team may designate one robot as a goal keeper. The goal keeper can hold and manipulate a ball for up to 15 seconds within its defense zone. After releasing the ball the keeper must not recapture the ball until the ball touches an opponent or a member of its own team outside the defense zone. If the ball is released by the keeper and it reaches the halfway line without touching any other robot, the opponent is given an indirect free kick positioned anywhere along the halfway line (borrowed from Futsal rule).

Any of the robots may change roles with the goal keeper (and thus be permitted to manipulate the ball) provided the referee is informed before the change and that the change is made during a stoppage of play.

12. Movement of Robots by Humans

In general, movement of robots by humans is not allowed. However, at kick-offs and restarts one member of the team is allowed on the pitch to place robots. Gross movement of robots is not allowed, except: before kick-offs, to place the designated kicker for a free kick, or to ensure robots are in locations required for penalty and free kicks.

Humans are not allowed to free stuck robots except during a stoppage of play, and then they should move the robots only far enough to free them.
13. Stopping and Restarting the Game

a. Play stoppage

Play is stopped in the following situations:

- When a goal is scored.
- When the ball is kicked out of play.
- Fouls resulting in a free kick.
- Fouls resulting in a penalty kick.
- Fouls resulting in a yellow card.
- At the end of a half.

b. Robot Halting

Once play has been stopped robots should cease movement until play is restarted by the referee. The referee may check or adjust the placements of the players prior to restart.

c. Kick-Off

All robots shall be located on their side of the field. The ball will be positioned at the center of the field and all robots on the team not kicking off must be outside the center circle. The ball has to go forwards at a kick-off or the kick-off will be restarted.

d. Penalty Kick
Only a goal keeper shall be in the defense zone, and the ball shall be located 45 cm from the goal along the lengthwise centerline of the field. All other robots shall be located at least 30 cm behind the ball.

Robots cannot move until the referee signals the resumption of play (by whistle, etc.).

e. Free Kick

Free kicks are taken after a foul or a stoppage in play.

If the free kick is taken after a foul the ball is placed at the point where the foul was committed (or as close as possible to that point). If the free kick is taken after a stoppage in play, the ball remains in place. In either case, the ball should be placed at least 15 cm from any wall.

If the ball is within 15 cm of a defensive zone, it should be moved to the closest free kick marker.

The nearest robot of the kicking team to the ball is to be designated as the kicker. It may be moved into position by a designated human for the kicking team. If the nearest robot is the goalie robot, then the kicking team has the option of using the next nearest robot.
Except for the kicking team, all robots must be 15 cm from the ball.

In case of a free kick for the offense, the defending team is not allowed to move its robots.

None of the robots may move until play is resumed by the referee.

f. Signals Sent to Robots

For the start or restart of the game the referee will call verbally, or by whistle, and the operator of the team can send signals to robots. The signal can be entered through a keyboard attached to a server being used on the side lines. No other information may be sent after the starting whistle.

The keyboard operator may not send any information during play.

Strategy revision during half time and timeouts is permitted.

14. Yellow Cards

a. Stoppage of Play and Free Kick
Play is stopped when a yellow card is assigned. A free kick is provided to the opposing team at the location where the foul occurred, or at the nearest free kick marker as deemed appropriate by the referee.

b. Assignment of Yellow Cards and Player Removal

Yellow cards are assigned to the team and not the player.

Each time a team receives two yellow cards, it must remove one player from the field (the number of players on the field is reduced by one after every two yellow cards).

Once a robot has been removed from the field, it may only be used to replace a robot. For example, should a team start the game with only four robots and receive two yellow cards, it must remove a robot and continue to play with three robots. The removed robot may only be used as a replacement for a robot. If the removed robot is entered into the game again, a different robot must be taken out of the game (for example, to replace a broken down robot in the remainder of the game).

c. Pushing

Pushing by a robot is defined as contact with an opponent robot with a movement vector through the opponent robot. Pushing another player,
even when in pursuit of the ball, is a "yellow card" offense also resulting in a free kick for the opposing team. The pushed robot is the kicker. Incidental contact between robots (except the goalie) is allowed.

Pushing "through the ball" to another player is not a foul, but the referee should take immediate action to reposition the ball to a neutral position.

d. Contact with the Goalie

Contact with the goalie is a foul if the point of contact is within the defense zone. A free kick is awarded to the defense.

e. Non-Moving Robots

If the referee determines that a robot is not moving for a period of 20 seconds or longer, he will remove it from play and give the team a yellow card.

Participants may repair the robot and ask that it be put back in play if they desire. A second failure of the same robot to move for 20 seconds will result in a red card and permanent removal of the robot from the game. Goal tenders and robots further than 20 cm from the ball will not be penalized.

f. Court or Ball Damage
Modification or damage to the court and the ball is forbidden. Should this occur, the game is suspended and the appropriate team is assigned a yellow card. Restoration is done immediately before the game resumes.

g. Damage to Other Robots

During play, if a player utilizes a device or an action which continuously exerts, or is likely to exert, serious damage to another robot, it will be presented a yellow card and ordered to go outside the court and correct the problem. This includes excessively hard kicking of the ball that may damage other robots.

Once the correction is made, the robot can resume play after approval by the referee. In case the problem is repeated, the referee presents a red card to the responsible player telling it to leave the game (in this special case the referee may direct that a specific robot be removed).

15. Fouls Resulting in a Free Kick

a. Shot on Goal too High

If the ball crosses the goal line 15 cm above the field, the goal is disallowed and a free kick is awarded to the defending team.

b. Ball Kicked out of Field
If the ball is kicked out of play, a free kick is assigned to the opposing team at a location near where the ball left the field. Determination of responsibility for kicking the ball out is made by the referee.

c. Too Many Attacking Robots in Defense Zone

Only one attacker may enter this area. Brief passing and accidental entry of other robots is permitted, but intentional entry and stay is prohibited.

d. Stuck Ball

If the ball gets stuck (as determined by the referee) in a corner or along a wall adjacent to a goal, play is stopped and a free kick is awarded to the offense.

Otherwise, when a ball is stuck, the referee will reposition the ball away from the wall in a location that does not provide an advantage to either team (play is not stopped). The referee will use a black/gray stick or some other device to reduce the chance of interference with vision systems.

16. Fouls Resulting in a Penalty Kick

a. Too Many Defending Robots in Defense Zone
When more than one robot of the defending side enters the defense zone and substantially affects the game, play will be stopped, and a penalty kick will be declared.

A robot substantially affects the game if:

- The robot contacts the ball.
- The robot prevents another robot from movement toward the ball.
- The robot is in a position that blocks a portion of the goal

b. Ball Holding

A player cannot 'hold' a ball unless it is a goal keeper in its defense zone. Holding a ball means taking a full control of the ball by removing all its degrees of freedom; typically, fixing a ball to the body or surrounding a ball using the body to prevent access by others. 80% of the ball should be outside the convex hull of the robot. It is up to the referee to judge whether a robot is holding the ball. In general another robot should be able to remove the ball from another player. If a robot is deemed to be holding the ball then a free kick will be declared. If this happens in the defense zone by the defense team, a penalty kick will be declared.

17. Offside

The offside rule is not adopted.
18. Fair Play

Aside from the above items, no regulations are placed against possible body contacts, charging, dangerous plays, obstructions, etc. However, it is expected that the aim of all teams is to play a fair and clean game of football.

19. Conflict Resolution

Resolution of dispute and interpretation of ambiguity of rules shall be made by three officials, who will act as referees, designated prior to the match. The referees shall not have any conflict of interest to teams in the match. The referees may consult with the tournament officials for resolving conflicts. Ambiguities shall be resolved by referring to FIFA official regulations, where appropriate.

20. Rule Changes at the Competition

Rule changes at the competition may be proposed by any team captain. Acceptance of a proposed change requires a 2/3 majority of team captains. Approved changes do not become a permanent part of the F-180 rules, but will be considered by the rules committee.

21. Future Competitions

a. It is the rules committee's intent to ensure that robots that meet current size rules will be permitted in all future competitions.
b. It is the rules committee's intent to change the playing field (perhaps as early as 2001) in the following manner: the current angles wall at the playing field boundary will be eliminated and replaced with a simple white line.

c. It is the rules committee's intent to permit global vision in all future competitions. Onboard sensing is encouraged and may be supported through separate sub-competitions, but global vision will continue to be allowed.
Appendix B - The Simple Dynamic Model

Shown above in Figure 29 is the top-level diagram of the Simple Dynamic Model for the Ohio University RoboCup Robot. As can be seen in the figure, the first block is the path generator function. In the figure it is the third-order path generator, but this can be replaced with whatever input is desired. The second large block (disregarding the small scope and workspace blocks) is the actual dynamic system. The system lying under that block is shown below in Figure 30.
As can be seen in the figure, the desired Cartesian pose is inputted into the inverse kinematics block, transforming the desired pose into desired wheel rotations. These values are then summed in the three summing junctions with the simulated value, giving the error signals. These error signals are then used as the input for the DEE (Differential Equation Editor), which solves the dynamic equations of motion (see section 4 Inertial Dynamics).
Appendix C - The Complex Dynamic Model

Figure 31: Block Diagram of the Complex Dynamic Simulink Model
As can be seen in Figure 31, the Complex Dynamic Model is, in fact, very complex. Discussion of the model will begin at the bottom, and work its way to the top. The beginning of the model is the path generator. This generates the desired $X$ and $Y$ Cartesian velocity, as well as the orientation $\phi$. Please note that only a orientation is inputted, not an angular velocity.

These signals are now branched off for the feed forward loop, bypassing the Cartesian error calculation due to the global vision system. The other branch is passed to the global vision system feedback loop, creating a Cartesian error signal. This error signal is then transformed to wheel rotation error, and summed with the desired wheel rotation from the feed forward loop.

This error signal is then summed with the smaller, wheel independent feedback loops, creating the final error signal. This is then used as the input for the DEE (Differential Equation Editor), to solve for the simulated position and orientation.
Appendix D - Slip Simulation Model

The top-level diagram of the Slip Simulation was shown and explained in Section 7.2. The Dynamics block from that top-level diagram is shown below in Figure 32.

As can be seen in the figure, the desired wheel rotations (as well as wheel positions) and the computed Cartesian velocities are fed into the Matlab function "dynamic". This was created to solve first for the coefficients of slip in both the wheel direction and the transverse direction, and then the equations of motion for the mobile robot. The Matlab Code for this function is shown below.

```matlab
function output = dynamic(input)

%dynamic model to solve slipping simulation
%inputs:
phi, xdot, ydot, phidot, theta1dot, theta2dot, theta3dot, theta1, theta2, theta3
%outputs: xddot, yddot, phiddot, slip1, slip2, slip3
```

global s1 s2 s3 r1 r2 r3 mu mu_lateral mu_space m I R sect con_rollers g

phi=input(1);
xdot=input(2);
ydot=input(3);
phidot=input(4);
theta1dot=input(5);
theta2dot=input(6);
theta3dot=input(7);
theta1=input(8);
theta2=input(9);
theta3=input(10);

% Rotation Matrix to change local frame to global frame
Rot = [cos(phi)  -sin(phi); ...
sin(phi)     cos(phi)];

% Slipping Friction in each wheel
v1 = [xdot ydot]'+ [0 -phidot;phidot0]*Rot*r1+theta1dot*R*Rot*s1;
v2 = [xdot ydot]'+ [0 -phidot;phidot0]*Rot*r2+theta2dot*R*Rot*s2;
v3 = [xdot ydot]'+ [0 -phidot;phidot0]*Rot*r3+theta3dot*R*Rot*s3;

% Friction only along wheel axis
slip1 = v1'*Rot*s1;
slip2 = v2'*Rot*s2;
slip3 = v3'*Rot*s3;

% Friction perpendicular to wheel direction
s1_p = [-s1(2) s1(1)]';
s2_p = [-s2(2) s2(1)]';
s3_p = [-s3(2) s3(1)]';
slip1_p = v1'*Rot*s1_p;
slip2_p = v2'*Rot*s2_p;
slip3_p = v3'*Rot*s3_p;
%% Friction as a function of wheel angle %%

tronc1 = abs((theta1/sect)-fix(theta1/sect));
if tronc1<con_rollers,
    mu_1 = mu;
    mu_11 = mu_lateral;
else
    mu_1 = mu_space;
    mu_11 = mu_space;
end

tronc2 = abs((theta2/sect)-fix(theta2/sect));
if tronc2<con_rollers,
    mu_2 = mu;
    mu_12 = mu_lateral;
else
    mu_2 = mu_space;
    mu_12 = mu_space;
end

tronc3 = abs((theta3/sect)-fix(theta3/sect));
if tronc3<con_rollers,
    mu_3 = mu;
    mu_13 = mu_lateral;
else
    mu_3 = mu_space;
    mu_13 = mu_space;
end

%% Friction Magnitude %%

F1 = -(m*g/3)*(2/pi)*mu_1*atan(1000*slip1)*Rot*s1;
F2 = -(m*g/3)*(2/pi)*mu_2*atan(1000*slip2)*Rot*s2;
F3 = -(m*g/3)*(2/pi)*mu_3*atan(1000*slip3)*Rot*s3;

F1_p = -(m*g/3)*(2/pi)*mu_11*atan(1000*slip1_p)*Rot*s1_p;
F2_p = -(m*g/3)*(2/pi)*mu_12*atan(1000*slip2_p)*Rot*s2_p;
F3_p = -(m*g/3)*(2/pi)*mu_13*atan(1000*slip3_p)*Rot*s3_p;

F_x = F1(1)+F2(1)+F3(1)+F1_p(1)+F2_p(1)+F3_p(1);
F_y = F1(2)+F2(2)+F3(2)+F1_p(2)+F2_p(2)+F3_p(2);

torque = (F1(2)-F1(1))*Rot*r1+(F2(2)-F2(1))*Rot*r2+(F3(2)-F3(1))*Rot*r3+...
\[ [F_{1_p}(2) - F_{1_p}(1)] \cdot \text{Rot} \cdot r_1 + [F_{2_p}(2) - F_{2_p}(1)] \cdot \text{Rot} \cdot r_2 + [F_{3_p}(2) - F_{3_p}(1)] \cdot \text{Rot} \cdot r_3; \]

\[
\begin{align*}
x_{ddot} &= F_x/m; \\
y_{ddot} &= F_y/m; \\
\phi_{ddot} &= \text{torque}/I; \\
\end{align*}
\]

\[
\begin{align*}
\text{output}(1) &= x_{ddot}; \\
\text{output}(2) &= y_{ddot}; \\
\text{output}(3) &= \phi_{ddot}; \\
\text{output}(4) &= \text{slip}_1; \\
\text{output}(5) &= \text{slip}_2; \\
\text{output}(6) &= \text{slip}_3;
\end{align*}
\]
Abstract

In this Thesis, a brief overview of mobile robots and omni-directional robots is presented, as well as a detailed history of the RoboCup competition. The player robot design process undertaken by the Ohio University Mechanical Engineering department (led by the author) is discussed in detail, and the inverse kinematic equations and dynamic equations of motion are derived. These dynamic equations were then used to create two Simulink simulations, the simple and complex dynamic models. A third simulation was created to compensate for the slipping disturbances in the wheel motivated by initial experimental work. The most accurate of the simulations (the third, dubbed the Slip Simulation) was then compared with the experimental data.