Adaptive Motion and Behavior for Four-Legged Soccer Robots

Sonia Chernova
School of Computer Science
Carnegie Mellon University

Advisor: Professor Manuela Veloso

May 15, 2003

Contents

1 Introduction **1**
1.1 Robot Soccer Environment . . . 1
1.2 Robot Platform . . . 2
1.3 Overview of CMPack on board Modules . . . 2

2 Motion **3**
2.1 Overview of the Motion System . . . 3
2.1.1 Parameterized Motion . . . 3
2.1.2 Non-Parameterized Motion . . . 4
2.2 Motion Analysis . . . 4
2.3 Results . . . 6
2.3.1 Body Angle . . . 6
2.3.2 Body Height . . . 7
2.3.3 Period . . . 7
2.4 Summary . . . 8

3 Modeling the Effects of Kicking Motions **10**
3.1 Ball Vision Analysis . . . 10
3.1.1 Trajectory Angle . . . 12
3.1.2 Distance . . . 13
3.2 Results . . . 14
3.2.1 Angle Analysis . . . 14
3.2.2 Distance Analysis . . . 18
3.3 Summary . . . 19

4 Selection of Kicks **21**
4.1 The Kick Selection Algorithm . . . 21
4.2 Results . . . 21
4.2.1 Experiment 1 . . . 22
4.2.2 Experiment 2 . . . 23
4.3 Summary . . . 24

5 Conclusion **26**

Abstract

The Sony AIBO four-legged robots bridge the gap between traditional wheeled robots and biped robots. Four legged robots offer a great challenge from a motion point of view. With twenty degrees of freedom, the motion system of the robot provides the user with over fifty parameters to control the motion. Motion development becomes more challenging with the introduction of a specific task. In particular, in robot soccer, we need fast, stable walking motions, robust kicking motions and a way to select which behavior to use.

In this thesis we study motion and behaviors using four-legged robots. The thesis has three main contributions:

- The analysis of the sensitivity of the walking motion with respect to the different parameters. We examine the impact of the body height, body angle and cycle period parameters on the velocity of the walking motion. Through our empirical results we are able to identify the existence of clear bounds for superior motion performance.

- The design of a set of robust kicks and a thorough study of their effects. We analyze the distance and angle traveled by the ball when actuated by the kick. This data is then used to build a model that represents each kick in terms of its effects on the world.

- The incorporation of the kick model into the selection of which behavior is to be performed in a variety of world situations. Our results show that using this model the robot achieves its goals more effectively than a robot using a strategy that does not take into account the predicted effects of its actions.

This research was initiated within the CMPack'02 robot soccer team, which became the world champion of the RoboCup 2002 competition. All experimental results in this thesis compare the performance of our algorithms to the performance of the CMPack'02 team. The kick model and selection algorithm have now been incorporated as part of the CMPack'03 team, demonstrating that these techniques indeed have significant impact on research on multi-robot teams in challenging environments, such as robot soccer.

Chapter 1

Introduction

Legged robots are still a novelty, as most of the robots used in research are wheeled. The AIBO robots, as remarkably developed by Sony, are very robust quadruped robots which are now commercially available. These robots have been used to examine a variety of research topics, and in particular robot soccer. The four-legged AIBO robots offer a challenge in terms of motion and its use in the complex task of robot soccer. This thesis first examines the motion module, and then proceeds to model the effects of kicking motions on the ball. We then introduce an approach to incorporating the effects model into the behavior module.

1.1 Robot Soccer Environment

The robot game environment is a playing field that is 430 cm in length and 280 cm in width with a green carpet surface. Two goals are located on opposite sides of the field, each one 60 cm wide and 30 cm tall. Six uniquely colored landmarks are positioned around the perimeter of the walls to help the robots localize on the field. The robots used a hard orange ball to play. Each team has four players, three attackers and one goaltender, with each match lasting twenty minutes. Robots are equipped with wireless wavelan cards and are able to communicate with their teammates. Figure 1.1 shows a sketch of the playing field.

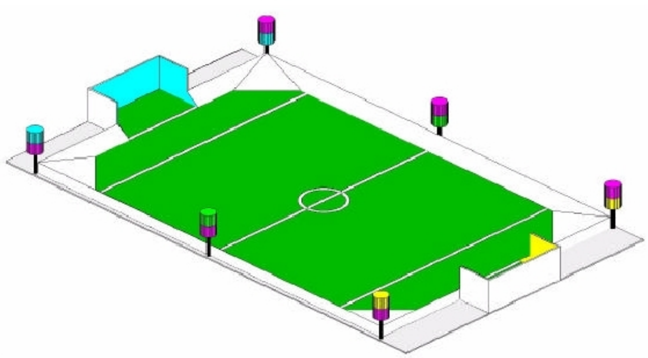

Figure 1.1: The playing field.

1.2 Robot Platform

The robots used in this research are the quadruped Sony AIBO ERS-210 Entertainment Robots which are designed to look like a small dog. The body of the robot contains a 400 MHz processor which runs Sony's proprietary real-time operating system Aperios. The neck and legs of the robot each have three degrees of freedom with five additional degrees of freedom in the tail, ears and mouth. The camera mounted in the head is the robot's primary sensor, with additional touch sensors in the feet. The camera has approximately $55°$ field of view and, together with the $90°$ pan of the head, it is able to scan the entire area in front of the robot. The robot is designed to be fully autonomous. It is able to perceive the world with its sensors, perform onboard computations to select actions, and execute these actions.

1.3 Overview of CMPack on board Modules

The soccer robots achieve their autonomy though a system of five modules - vision, localization, world model, behaviors and motion. The vision module is responsible for interpreting data from the camera using color thresholds. Four object types, each having a predefined color, exist in the robot's environment - the ball, six uniquely colored landmarks, goals and robots. The vision module reports the estimated distance and bearing to each identified object. The localization module uses information about the relative distance and angle of the landmarks to estimate the robot's location on the field. When no markers are visible the location of the robot is estimated based on the robot's last known position, the motions executed and the sensor readings. The world model module maintains information about the ball and other robots, and also provides rough location estimates for these objects when they are outside the camera's field of view.

The description of the environment provided by the localization and world model modules is used by the behaviors to select the next action to execute. The behavior module is represented as a state machine to enable sequencing of behaviors. Actions are specified at a high level, such as walking with a certain velocity or performing a specific kick motion. The motion module interprets the command and performs the low-level calculations for required joint angles and PID values of the joints needed to carry out the command.

Chapter 2

Motion

The motion module allows the robot to walk in any direction and execute prerecorded motions such as kicks and get up motions. Given a request from the behavior module for a high level action to be performed, such as walking in a specified direction or executing a certain type of kick, the motion module determines the required angles and PID values for each of the joints to carry out the command. The module is designed to provide stable and fast locomotion, and to allow smooth, unrestrictive transitions between different types of locomotion and other motions, such as kicks. Motions are implemented in two ways, a parameterized technique for implementing continuous walking motions and a non-parameterized technique for specifying additional non-dynamic motions such as kicks.

2.1 Overview of the Motion System

2.1.1 Parameterized Motion

The walking system is implemented as a generic walk engine that can encode a variety of gaits. The walk parameter structure consists of 51 parameters. The movement of each leg is determined by 11 parameters: neutral kinematic position (3D point), lifting and set down velocities (3D vectors), and a lift time and set down time during the walk cycle. 7 global parameters define the behavior of the body: body height and angle during the walk, hop and sway amplitudes, walk period, and lift height limits for the front and rear leg pairs. Using these parameters, the walk engine calculates the path of the body, the timing of the leg transitions, and airpath of the legs that satisfy the kinematic constraints and fulfill the behavior's requests for direction and speed of motion.

Before new parameter sets are executed on the robot they are tested for kinematic consistency through motion simulation. Parameter testing can indicate invalid parameter combinations that will lead to unbalanced motion. By specifying velocities the walk is tested to determine if there are requests made by the parameters on the motion system that are beyond the physical limits of the motors. Such situations occur when the specified parameters request that the robot assume a position that is not reachable with any joint angle combination. Kinematic error values are reported each time this occurs. In such cases, positions as close as possible to the desired configuration are used. Large numbers of kinematic errors signal that the planned motion will not be the one that is executed and that the parameter set should be modified.

2.1.2 Non-Parameterized Motion

Non-dynamic motions, such as kicks, are defined through motion script files. Generally lasting only a few seconds, these motions are designed to be executed the same way every time. Each motion script describes the transitions of the body frame by frame by specifying a series of body, leg and head positions and a time period for interpolating between one position and the next. The motion module provides a large suite of different kicks to be used by the behaviors. Figures 2.1, 2.2, 2.4, 2.3 show several kicks that will be discussed in Chapters 3 and 4.

2.2 Motion Analysis

The development of a walk that optimizes speed, stability and smoothness is a long, challenging process of selection and testing. The motion parameters form a complex interdependent network in which changes to a single parameter can lead to unpredicted changes in the motion. For example, changing the body height parameter can limit the range of leg positions and body angles, a total of 13 parameters.

The current walk development strategy consists of slowly adjusting existing parameters to achieve desired motion effects. Special walking motions have been developed for turning, forward and sideways walking, and dribbling the ball. Due to the competitive nature of the Robot Soccer environment, speed has been highly emphasized in walk development, and the motion module has been designed in a way that allows it to switch to the motion parameters that are best designed for its current task.

Our goal is to develop a procedure using which the robot will be able to adjust its gate autonomously and improve its walking motion on new surfaces. Given several parameter sets the robot should be able to select one with the best performance. Through a limited number of trials the robot would then modify this parameter set in order to develop a new walking motion that is better suited for the current surface. Similar to the way humans are able to shift their center of mass and control the size and pattern of their step depending on the ground surface, we expect that the robot can learn to adjust the period, position and velocity of its leg motion to achieve greater stability and speed.

A large number of parameter combinations is possible with such a large number of motion parameters. In order to avoid excessive wear on the robot our goal is to find a way to limit the number of trials to be executed to approximately 50. In order to develop such an algorithm, a metric measuring the performance of each motion parameter set must first be established. Parameters must also be analyzed in order to categorize and predict the changes caused by parameter modifications.

We establish the following metric for assigning performance score values to each parameter set. The score for each trial is based on the distance traveled by the robot in a set amount of time, with deductions for perpendicular and angular displacement. In order to simplify the domain, we consider only straight forward motion with an emphasis on speed. Each parameter set is evaluated in three independent trials, with the final performance score equaling the average over these trials. The parameter trial process is executed autonomously by the robot, requiring no human supervision except battery changes every 30 minutes. The final evaluation score for each parameter set is acquired by averaging the scores of three separate trials. During each trial the robot walks for a period of 7 seconds and then uses the colored landmarks around the field to localize its position

Figure 2.1: The dive kick. By falling onto the ball with its chest, the robot uses the weight of its body to propel the ball forward.

Figure 2.2: The forward arm kick. The robot uses its arms to grab the ball and position it in the middle of its chest. The mouth is opened to keep the ball from settling too close to the chest. The arms are then used to drive the ball forward.

Figure 2.3: Normal head kick. The robot uses the side of its head to kick the ball to either the left or right side, nearly perpendicular to the robot.

Figure 2.4: Hard head kick. Similar to the normal head kick, this motion also kicks the ball to the side. By swinging the entire body in addition to the head, the robot achieves greater force for a stronger kick.

and determine the distance it has traveled.

With over fifty parameters involved in controlling walking motions, it is important to know how each parameter affects motion and how sensitive the motion is to changes in that parameter. We would like to examine the effects of single parameter variation on motion performance. Using the metric described above, we can now select a handful of parameters that we believe strongly effect the motion and examine the sensitivity of the walk to changes in these parameters.

2.3 Results

Three parameters were initially chosen for motion sensitivity analysis: body height, body angle and the period of the motion cycle. In each experiment all other parameter values remained fixed while the value of a single parameter was tested over a range of values. The results of the three experiments are shown in Figures 2.5, 2.6, and 2.7. Below each performance graph, a plot of kinematic error values is shown.

2.3.1 Body Angle

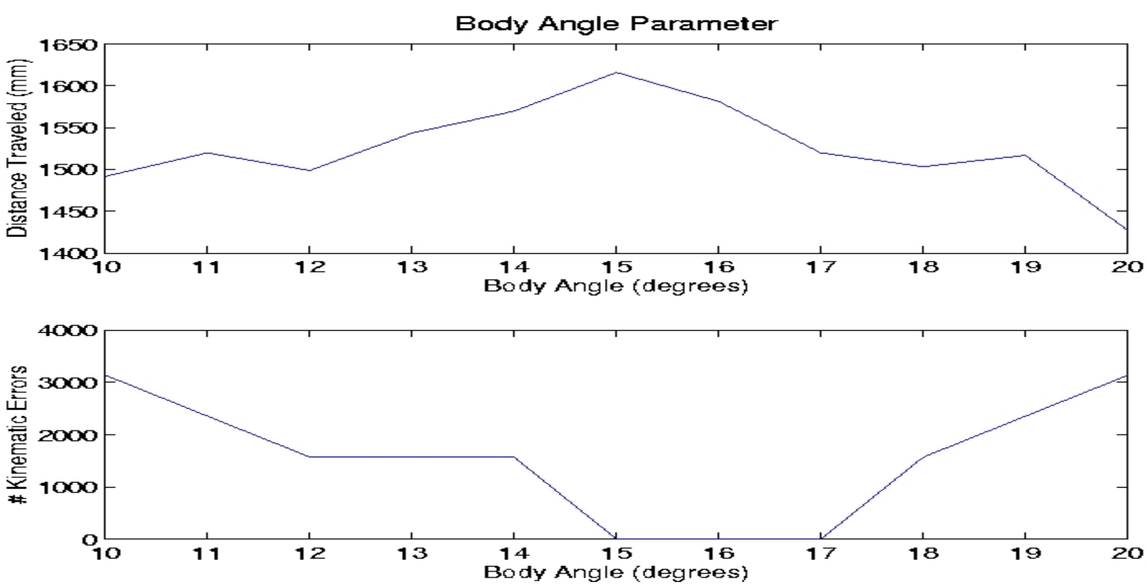

Figure 2.5: Body angle parameter analysis.

The body angle parameter defines the desired angle of the robot's torso relative to the ground. The tested parameter values range from $10°$ to $20°$ at $1°$ intervals. All other parameters remained fixed throughout the trial. The top graph displays the average distance traveled by the robot using each parameter value, while the bottom graph displays the number of kinematic errors reported by the motion simulation for each parameter set.

The bottom graph shows that this particular combination of the other 50 parameters leaves only a very small range of body angle values for which the number of kinematic errors is 0. The number of errors quickly rises on both sides of this range. The performance peak roughly corresponds to body angle values that result in few or no errors. As the number of errors grows the walk becomes increasingly unbalanced. Velocity ranges from 202mm/sec to 231mm/sec with highest performance when the body angle value is $15°$.

2.3.2 Body Height

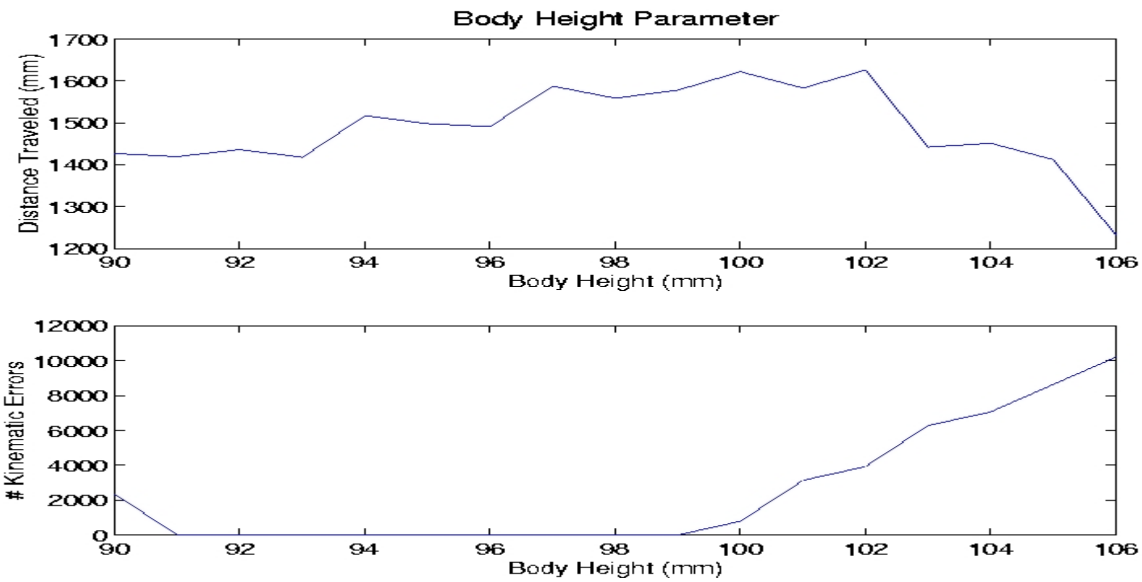

Figure 2.6: Body height parameter analysis.

The body height parameter defines the height of the robot torso above the ground measured directly below the neck base of the robot. Parameter values were tested over a 17 mm range starting at 90 mm with 1 mm intervals. All other parameters remained fixed throughout the trial. The top graph displays the average distance traveled by the robot using each parameter value, while the bottom graph displays the number of kinematic errors reported by the motion simulation for each parameter set.

The error graph shows a much larger range of body height values that are free from kinematic errors. Once kinematic errors appear, the number rises very quickly to over 10,000 errors per motion cycle. Motion performance is high for all values with no kinematic errors, and continuing high performance while the number of errors is low. As the errors increase, motion performance drops. This most likely results from the fact that the body height value is not achievable from the fixed leg position and body angle values. Overall, velocity ranges from 178 mm/sec to 234 mm/sec with highest performance at a body height of 102 mm.

2.3.3 Period

The motion cycle period is measured in milliseconds and controls the length of one motion cycle in which all four legs complete one stepping motion. Longer cycle periods result in wider strides

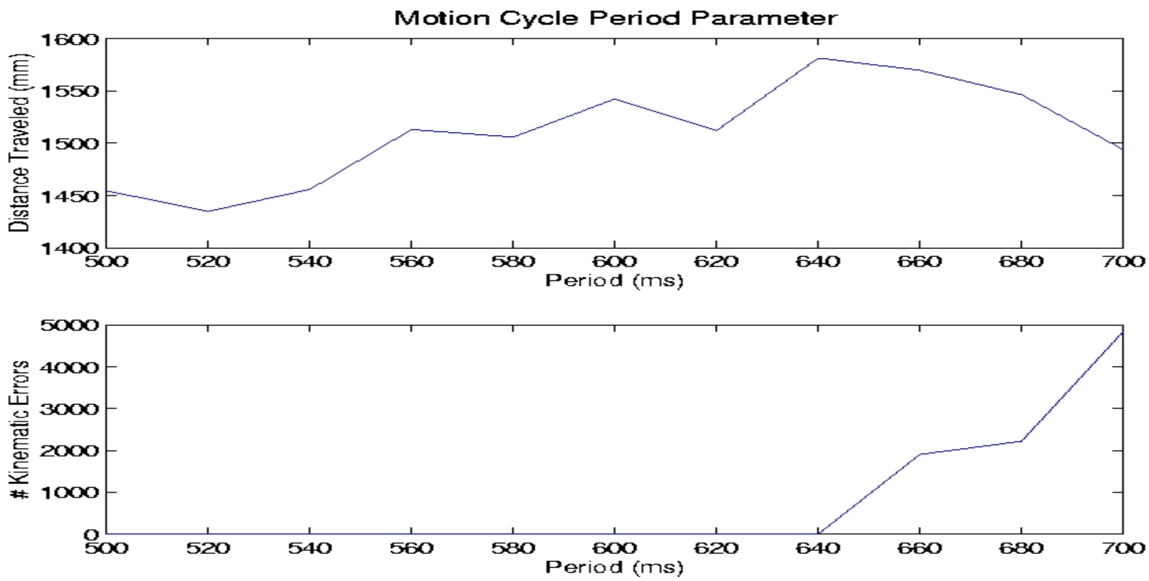

Figure 2.7: Period parameter analysis..

while shorter cycle periods result in frequent short steps. Parameter values were tested over a 200ms range at 20ms intervals. All other parameters remained fixed throughout the trial. The top graph displays the average distance traveled by the robot using each parameter value, while the bottom graph displays the number of kinematic errors reported by the motion simulation for each parameter set.

A large range of period values has no kinematic errors and is expected to result in balanced motion. Results show that performance steadily increases as the period value grows, reaching its peak at the last value for which there are no errors. Performance declines with increasing numbers of kinematic errors. Velocity ranges from 205 mm/sec to 225 mm/sec with highest performance when the period is 640 ms.

2.4 Summary

Our original goal was to develop an algorithm that would enable the robot to modify the current set of motion parameters in order to optimize its walking motion on a new surface. After specifying a metric for measuring motion performance, we examined how small changes to a single parameter affect motion. With 51 interdependent parameters controlling motion, we found that changing even a single parameter can have adverse effects on motion performance. With so many interdependent parameters, it is impossible to predict changes in one parameter upon the motion of the robot. The interdependent nature of the parameter structure prevents us from limiting the parameter search space to 50 sets; hundreds of parameter sets must be evaluated to select an optimal walking motion. Testing hundreds of parameter combinations for any new surface is impractical due to time limitations and possible wear on the robot.

We then turned our focus to studying the effects that certain motions have on the robot's environment and how modeling the effects of these motions can be used to improve action selection algorithms in the behavior module. After extensively testing our set of kick motions, we proceeded to accurately model their effects on the ball. We then incorporate this effects model into the behaviors and show how it can be used to improve the performance of the kick selection algorithm.

Chapter 3

Modeling the Effects of Kicking Motions

The behavior system takes input from the vision, localization and world model modules and determines the appropriate action to be executed by the motions. The system must be able to deal with an arbitrary number of game situations; this includes specifying behaviors in situations where the robot is controlling the ball, the ball is in an unknown location, or the robot is playing a supporting role while its teammate attempts a play. This research focuses on behavior selection in situations when the robot has gained control of the ball and must decide on an action that is best suited for achieving its goal of scoring on the opposing team. Information from the vision, localization and world model systems provides the robot with a description of its environment and the location of its target. Before the robot is able to select an appropriate action, it must first know the effects that each action has on the ball. The first step in implementing a selection algorithm is to model the effects of the different actions available to the robot. The attributes that most interested us are the angle of the ball's trajectory after the kick and the average distance the ball will travel.

3.1 Ball Vision Analysis

All information about the robot's environment is gathered though the on-board camera mounted in the head of the robot. Before analyzing the effects of the kicks, it is important to test the reliability of this data source. Our main interest is to determine the accuracy of the angle and distance measurements to the ball that are reported by the vision system.

A simple experiment was designed to test the accuracy of the ball vision. The robot was positioned on the middle of an empty field with a single ball placed a set distance away from the robot. Once the robot focused on the stationary ball, the reported position of the ball relative to the robot was recorded for a period of approximately 200 frames, or 8 seconds. This was repeated using five different ball locations at varying distances and angles (see Figure 3.1). The same procedure was then repeated with the robot pacing in place (Figure 3.2) and turning in place (Figure 3.3).

Figure 3.1 shows that while standing, the vision's angle estimate is much more accurate than the distance, causing the estimates to fall within a narrow, long strip instead of a circular mass. Variation in both angle and distance measurements grows proportionally with the distance to the ball. The striped pattern, seen clearly in the more distant points, is a product of the distance estimation algorithm and the low resolution of the robot's camera. The vision system uses the size of

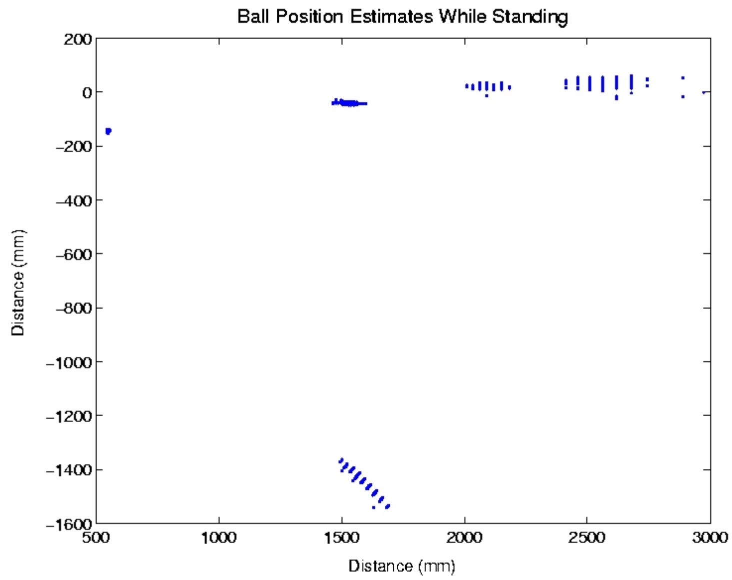

Figure 3.1: Ball position estimates reported by the vision system while standing

the ball to estimate the distance to the object. At two meters away the ball takes up approximately seven pixels in the camera. As the number of orange pixels detected by the camera varies slightly from frame to frame, it causes the distance estimate to vary as well. Each stripe in the pattern represents a one-pixel difference in the size of the ball in the image.

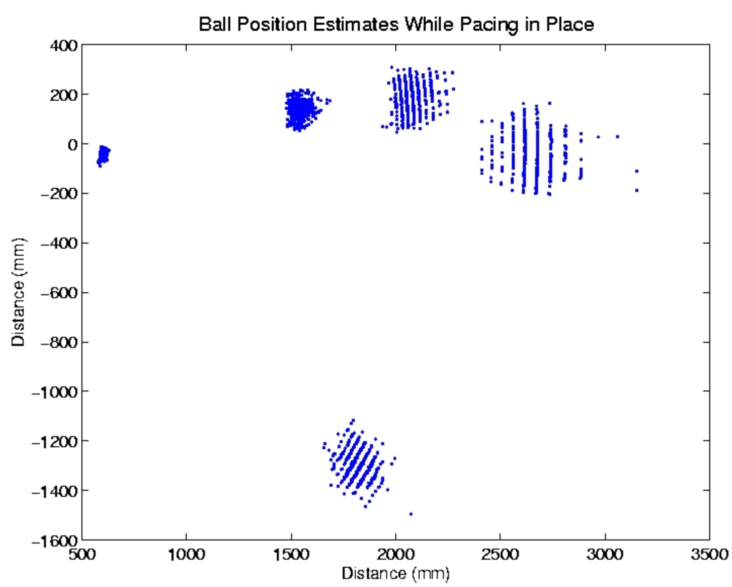

Figure 3.2: Ball position estimates reported by the vision system while pacing in place

Figure 3.2 shows the effects that small movements of the camera have on the ball position estimates. The distribution of points becomes more circular, showing greater variation in angle values while variation in distance grows only slightly. The striped pattern seen in the previous figure also remains.

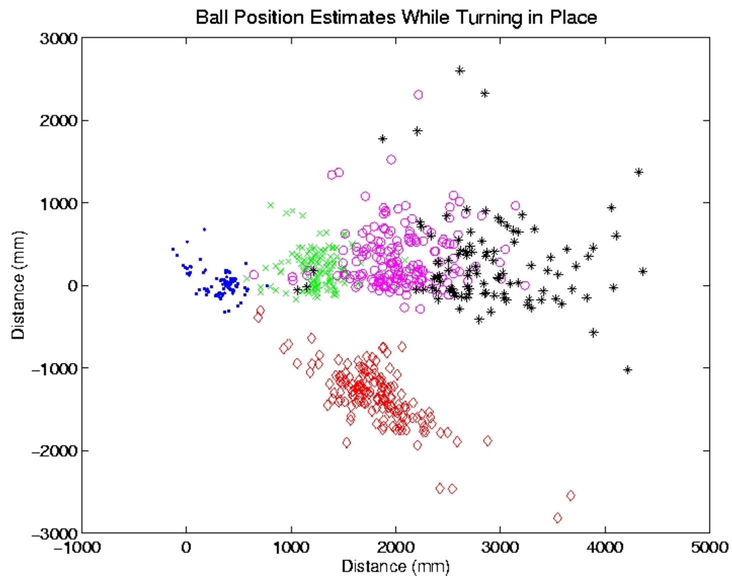

Figure 3.3: Ball position estimates reported by the vision system while turning in place in a counter-clockwise direction.

Figure 3.3 shows what happens to the estimate of the ball position as the robot spins counter-clockwise in place. The estimate of distance and angle become wild and unreliable. Even estimates for nearby points are spread over wide areas, with estimates for the farthest point ranging over a two meter wide area. Much of the problem can be attributed to the fact that as the robot spins at a speed of 2.5 radians/sec, the ball appears blurred or elongated in the images captured by the camera. The unusual shape of the ball, as well as frames where the ball is only partially visible, account for the wide distribution that we see.

The results of this experiment show that the most reliable data is reported when the robot is standing still. Gathering data on the location of the ball or the robot should be done while stationary and close to the target if possible.

3.1.1 Trajectory Angle

The angle of the ball's trajectory after a kick is an important characteristic of all kicking motions. No external cameras or sensors are used in this experiment in order to make it applicable in any environment. The robot's vision module provides information about the ball's location for every vision frame where the ball is in the visual range of the camera. The ball position history records the position of the ball relative to the robot's frame of reference over the past 25 frames, or approximately 1 second. The ball history reports the ball position as unknown for frames where the ball is outside the camera field of view. Ball location estimates from the world model are not used in order to minimize error. We developed the following algorithm for approximating the angle of the ball's trajectory:

Algorithm 3.1.1: TRACKANGLE()

comment: Input: Estimated ball distance and angle values from the vision module.

comment: Output: Angle of the ball's trajectory.

$timeOfKick \leftarrow 0$
while 1

$$
\textbf{do} \begin{cases} \text{TRACKBALLWITHHEAD}() \\ \textbf{if } BallWithinKickingRange = \textbf{ true} \\ \quad \textbf{then} \begin{cases} \text{KICK}() \\ timeOfKick \leftarrow currentTime \end{cases} \\ \textbf{if } currentTime - timeOfKick > t_{delay} \\ \quad \textbf{then} \begin{cases} angle \leftarrow \text{CALCANGLEUSINGBALLHISTORY}() \\ \textbf{output } (angle) \end{cases} \end{cases}
$$

The robot is able to perform all analysis autonomously, and requires human assistance only in placing the ball in front of the robot for the next kick. The robot does not approach the ball on its own in order to assure the consistency of the trials.

The value of the variable t_{delay} is a key factor in the success of this procedure. Since the vision ball history is continuously keeping track of the ball's location, it is important to analyze the data in the history at the right time - before the ball is so far away that the values become unreliable, but after enough time has passed to record at least a second of the ball's trajectory. The standard value of t_{delay} is $t_{kick} + 1.0sec$, where t_{kick} is the duration time of the kick. Slightly larger t_{delay} values are used for kicks with a lot of head movement that may cause a delay in tracking the ball after the completion of the kick.

A linear regression algorithm is applied to the points in the history buffer to calculate the angle of the ball's trajectory. The slope of the line is used to approximate the angle of the trajectory. In order to assure accuracy, we require that a minimum of 20 out of the last 25 vision frames contain information about the ball. Trials with fewer data frames are often a sign of an error. This can occur if the t_{delay} value is too small, if the kick fails or if the robot for some reason fails to locate and track the ball. Eliminating these trials ensures that inaccurate data is not introduced into the analysis and makes this learning process more efficient, requiring little human intervention.

3.1.2 Distance

The second attribute important in understanding the effects of different kick motions is the distance the ball travels or the strength of the kick. The robot is unable to track the entire trajectory of the ball because the ball travels out of the robot's visual range for many of the powerful kicks. Instead, the robot calculates the final position where the ball stops relative to the original position of the robot before the kick. The following algorithm was developed for accurately calculating the displacement of the ball after a kick:

Algorithm 3.1.2: TRACKDISTANCE()

comment: Input: Estimates of ball and robot locations provided by the vision module.

comment: Output: A vector representing the ball's displacement relative to the location of the kick.

while 1

$$
\textbf{do} \begin{cases} \text{APPROACHBALL}() \\ \text{KICKBALL}() \\ \text{STANDANDLOCALIZE}() \\ initialBallPosition \leftarrow currentRobotPosition \\ \text{FINDBALL}() \\ \text{APPROACHBALL}() \\ \textbf{if } ballDistance < 50cm \\ \quad \textbf{then} \begin{cases} \text{STANDANDLOCALIZE}() \\ finalBallPosition \leftarrow currentBallPosition \\ ballDispVector \leftarrow finalBallPosition - initialBallPosition \\ \textbf{output } (ballDispVector) \end{cases} \end{cases}
$$

The robot is able to perform this analysis completely autonomously with no human assistance. Each trial takes approximately 1-2 minutes. It is important to note that the robot records the ball's location while remaining at a distance of 50 cm in order to avoid accidentally touching the ball and possibly changing its location.

In addition to estimating the average distance the ball travels, this procedure can also be used to determine the expected success rate of the kick. Failed kicks can be detected easily because the ball remains very close to the robot and a simple distance threshold can be used to distinguish between successful and unsuccessful trials. Detecting trials where failure occurred allows us to establish a reliability measure for each kick, as well as exclude these results from the final distance analysis.

3.2 Results

3.2.1 Angle Analysis

We selected two kicks, a forward arm kick and a side head kick, for the analysis of the trajectory angle. Generally considered reliable and widely used motions, together these kicks cover a wide range of angles. Figures 3.4-3.6 display angle analysis data from the ball history for the forward kick. Figures 3.7-3.9 display similar data for the head kick. Data following all successful kicks shows very few outliers and fits the linear approximation model well. Data for unsuccessful kicks is easily distinguished and excluded from analysis.

From observation of this experiment it was noted that the variation in the angle of the kick resulted from the position of the ball relative the head of the robot before the kick. If the ball is located very close to the head, the angle of trajectory is likely to be greater relative to the robot's frame of reference, with the ball traveling perpendicular to the way the robot is facing. Trials where the ball was located slightly further resulted in the ball being hit by the edge of the head, causing the ball to have more forward velocity and a smaller angle.

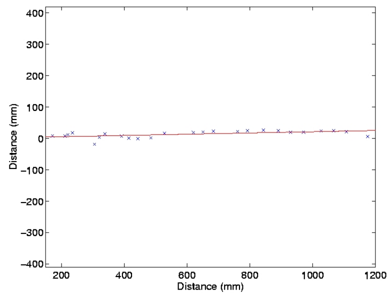

Figure 3.4: Forward kick angle analysis. Ball history after a successful forward arm kick. Each point represents the relative position of the ball in a vision frame as reported by the vision system. A regression line was fitted to the points to estimate the angle of the ball's trajectory.

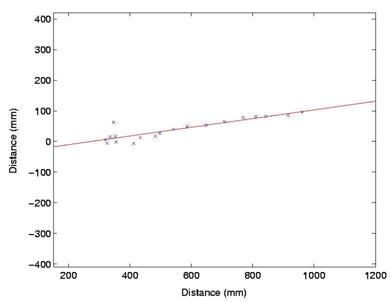

Figure 3.5: Forward kick angle analysis. Ball history after another successful forward arm kick.

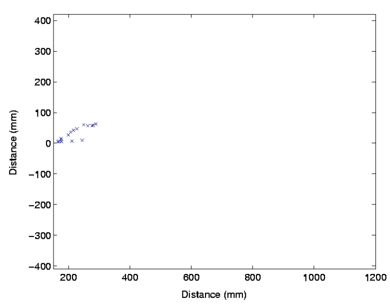

Figure 3.6: Forward kick angle analysis. Ball history after an unsuccessful forward arm kick. Vision reports that the ball remains close to the robot, although slightly displaced from some impact with the robot. Since only 15 frames reported the position of the ball, no analysis was performed on this trial.

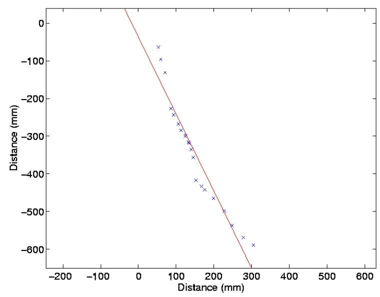

Figure 3.7: Head kick angle analysis. Ball history after a successful left head kick. Each point represents the relative position of the ball in a vision frame as reported by the vision system. A regression line was fitted to the points to estimate the angle of the ball's trajectory.

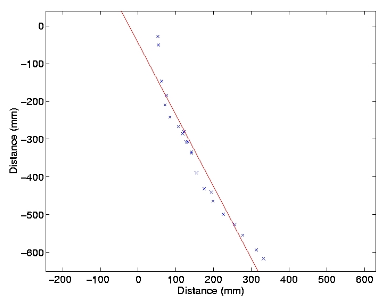

Figure 3.8: Head kick angle analysis. Ball history after another successful left head kick.

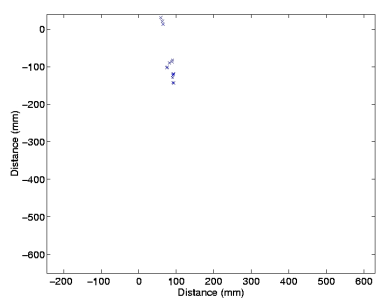

Figure 3.9: Head kick angle analysis. Ball history after an unsuccessful left head kick. The ball remains close to the robot. Not enough data points are present and this trial is also excluded from analysis.

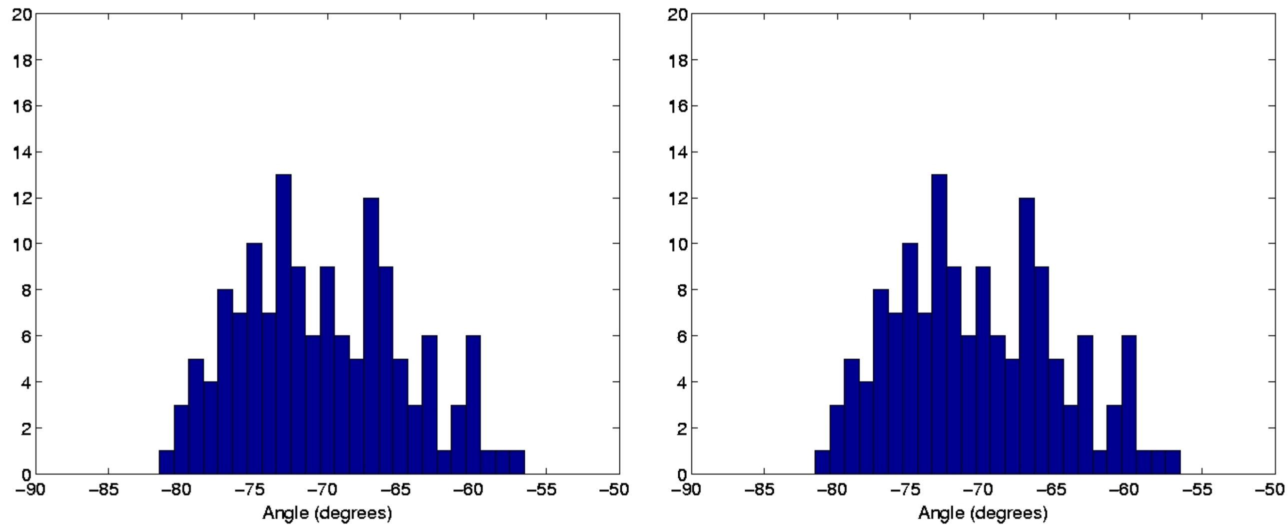

Figure 3.10: Head kick angle distribution.

Figure 3.10 displays the final angle analysis data after 280 total trials for the left and right head kick. The data is represented as a histogram showing the frequency of each angle value. The means of the distributions are $72.6°$ and $-70.4°$ for the left and right kick respectively, with standard deviations of $4.5°$ and $5.6°$.

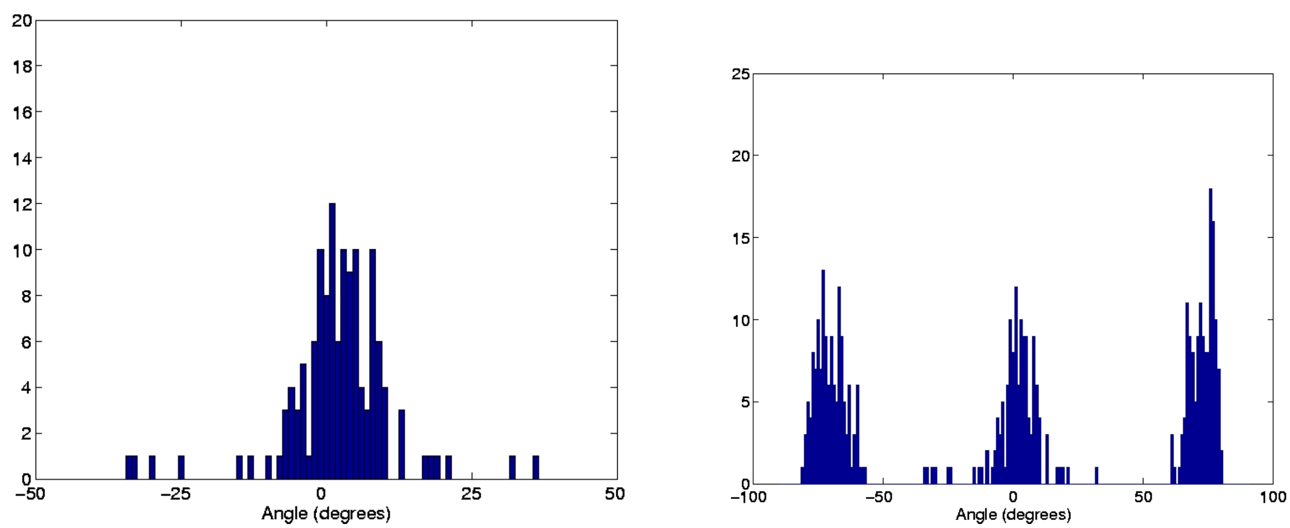

Figure 3.11: Forward kick angle distribution and the distribution of angle values for all three kicks.

The left image of Figure 3.11 displays the final angle analysis of the forward kick. The mean angle was found to be $2.1°$, with a standard deviation of $9.1°$ after 130 trials. It is interesting to note that although the forward kick motion is completely symmetrical relative to the left and right sides of the body, the kick lightly favors the left (positive) direction. This can be attributed to the uneven distribution of the weight in the body of the robot.

The large number of outliers and large standard deviation are the first sign of inconsistency in the forward kick. As we will see later, the success rate of this kick is significantly smaller than the head kick, resulting in many outliers. Similar to the head kick, the angle of the ball's trajectory also varies relative to the initial proximity of the ball.

The right image of Figure 3.11 displays the angle distributions of all three kicks together. A wide range of angles is covered even with just three kicks.

3.2.2 Distance Analysis

In addition to the forward and head kicks used in the trajectory angle analysis, another head kick, the hard head kick, was also analyzed. While similar to the normal head kick, the hard head kick uses the weight of the robot's body to propel the ball greater distances by using a swinging motion. Analyzing both head kicks provides a good comparison between outwardly similar kicks.

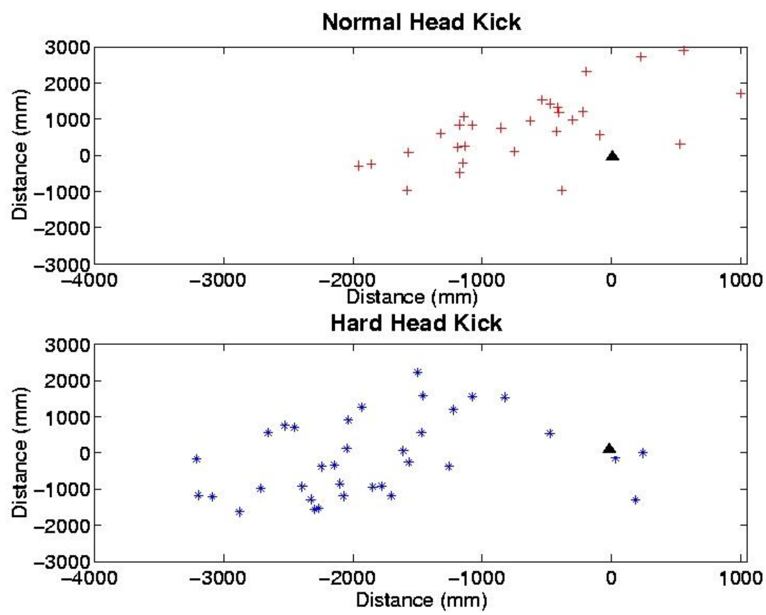

Figure 3.12: Head kick distance comparison.

Figure 3.12 shows the result of the distance analysis of the two left head kicks. The triangle in the figure represents the original position of the robot, standing at (0,0) facing up, and each point represents a final resting position of the ball after a kick. The difference in the kick strengths is clearly apparent. The hard head kick propels the ball much farther, with some distances nearing 3.5 meters with an average distance of 3.07 meters. The normal head kick has a range of at most 2 meters with an average of 1.64 meters.

This graph also displays data for trials where the kick possibly failed or behaved unexpectedly. A kick is considered to have failed if proper contact was not made and the ball moved only a few centimeters if at all. An example of such a kick can be seen in the bottom graph where the ball's final location coincides with the initial robot position. Other outliers, points located to the right or behind the robot, can be attributed to two factors. The first is a kick where significant contact was made but possibly with the wrong part of the body or at an incorrect angle. This could cause the ball to roll in a random direction. The second factor is that the field is not a completely flat surface and the ball is not perfectly round. These imperfections often cause the ball to curve and roll in a circular pattern, so that a kick to the left may result in the ball curving around and stopping behind the robot. Since the robot is unable to track the entire trajectory of the ball, it is unable to distinguish a trial where the ball was hit correctly but curved around, from one where the kick did not function correctly. The robot is only able to detect when the ball has not moved. As can be seen from the graph, the success rate for both head kicks is very high, with approximately 1% of kicks failing.

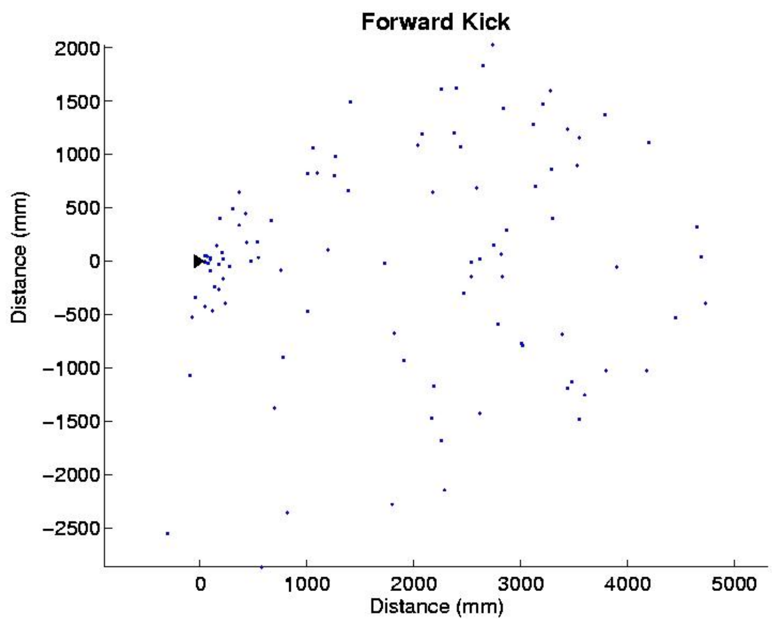

Figure 3.13: Fwd kick dist.

Figure 3.13 displays distance analysis results for the forward kick. The small triangle represents the robot's initial position at (0,0) facing right. The points are spread over a large area with wide variations in both distance and angle. A number of points can be seen clustered within a 0.5 meter radius of the initial position, signaling a large number of failed kicks. The success rate of the kick has been estimated to be approximately 85%. The average distance traveled by the ball is 2.2 meters.

3.3 Summary

We selected two specific attributes to model the effects of the kicking motions, the angle of the ball's trajectory and the distance reached by the ball after the kick. Analysis of both kick attributes was performed using only the robot platform with no additional sensors and minimal human intervention. We used the acquired data to build a model that represents each kick in terms of its effects on the ball. A summary of the model data for the kicks described in this chapter is shown in Table

Kick	Angle (degrees)	Distance (m)
Forward	2.1	2.2
Normal Head Left	72.6	1.64
Normal Head Left	-70.4	1.64
Hard Head Left	72.6	3.07
Hard Head Right	-70.4	3.07

Table 3.1: Kick model description of the five kicks discussed in this chapter.

3.1. Note that due to the similarity of the motions, the same angle value was used for the hard head kick as for the normal head kick in this model. No experimental results are available for the hard head kick and the actual angle values may differ slightly.

Chapter 4

Selection of Kicks

Once we have modeled the kick motions, we developed an algorithm for selecting an appropriate kick to use. Previous implementations of kick selection algorithms have not relied on prediction or modeling of actions. Decisions were usually hard coded based on the programmer's intuition about different effects of the kicks. Although a human can roughly estimate the average distance that the ball is expected to travel after seeing a series of repeated trials, this data is not very accurate. Estimating the angle of the trajectory can be especially difficult. Here we present a new kick selection algorithm that takes advantage of the acquired data and selects actions in a more robust manner.

4.1 The Kick Selection Algorithm

The data gathered in the modeling stage can be organized into a motion library where each kicking motion is classified by its effects on the ball, mainly the mean angle and distance values. The library provides an easy reference and allows the user to easily add or remove motions.

The new kick selection algorithm was designed to take advantage of this new organization in the following manner. The localization system provides information about the robot's position on the field and the relative location of the goal, which allows the robot to calculate the desired trajectory of the ball. The motion library is then referenced to select the most appropriate kick whose effects match the desired trajectory. If no kick motion provides a close match, then a series of behaviors can be sequenced together to achieve the desired effect. For example the robot may chose to turn or dribble the ball in order to achieve a better scoring position.

4.2 Results

Two types of experiments were used to test the performance of the developed algorithm. The first experiment tests the robot's ability to score a goal on an empty field without any other robots present. Testing single robot performance is very important because it allows us to fine tune individual performance and analyze the robot's strategy without interference from other robots. Additionally, since the robot's vision of other robots is often inaccurate, this is one of the best metrics of performance. The second experiment tests the ability of the robot to score a goal when playing against a single opponent. Testing the algorithm in a one-on-one play situation allows us to test performance in an adversarial environment.

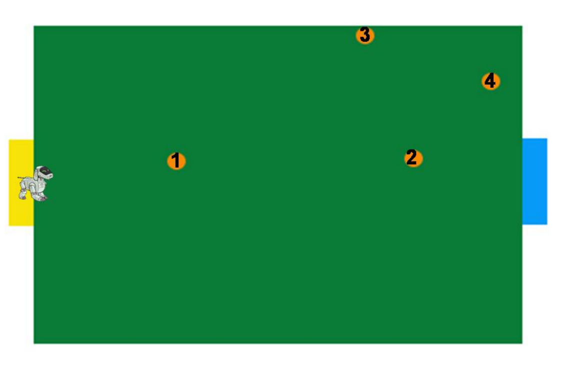

Figure 4.1: Experiment 1: ball positions on field.

4.2.1 Experiment 1

During this experiment the robot is placed on the goal line of the defending goal. The ball is placed at one of the four marked points of the field, see Figure 4.1. The location of the ball is unknown to the robot ahead of time and the ball may or may not be within visible range. The robot's performance is evaluated by recording the time it takes to score. The four points in Figure 4.1 were selected to test various cases.

- Point 1 tests the robot's ability to score from a very large distance. The ball is located close to the initial position and in many cases the robot will require more than one kick to score.
- Point 2 tests the accuracy of the kicks. The point is located at a short distance and a direct angle to the goal. The robot is expected to score directly.
- Point 3 test the robot's ability to score from the side when the ball is near a wall. The robot may or may not see the ball at this distance and often requires more than one kick to score.
- Point 4 tests situations when the ball is located in the corner at a wide angle to the goal. The robot does not see the ball at this distance. Scoring with one kick is very unlikely from this position.

Table 4.1 summarizes the results of the experiment. A total of 104 trials were tested, with 13 trials for each ball position. Results show that the modeling and prediction algorithm performed better for every point, with an overall improvement of 13 seconds. The statistical significance of the results was confirmed using the Wilcoxon Signed Rank Test.

Point	CMPack'02	Modeling
Point1	56.7	39.8
Point2	42.5	27.2
Point3	76.5	60.0
Point4	55.0	52.0
Total	57.8	44.8

Table 4.1: Experiment 1, performance comparison of the CMPack'02 and the kick modeling selection algorithms. Values represent mean time to score in seconds, averaged over 13 trials per point.

4.2.2 Experiment 2

During this experiment two robots from opposing teams compete with each other to score on the opponent's goal. The robot from the red team is running the new modeling and prediction algorithm and is defending the yellow goal. The robot from the blue team is running the CMPack'02 kick selection algorithm and is defending the blue goal. At the beginning of every trial the robots are placed on specified points on the field, with the ball located in an open area equidistant from their positions. To begin the trial, the robots are simultaneously unpaused, and play continues until a goal is scored. Performance is ranked by the number of goals scored by each robot.

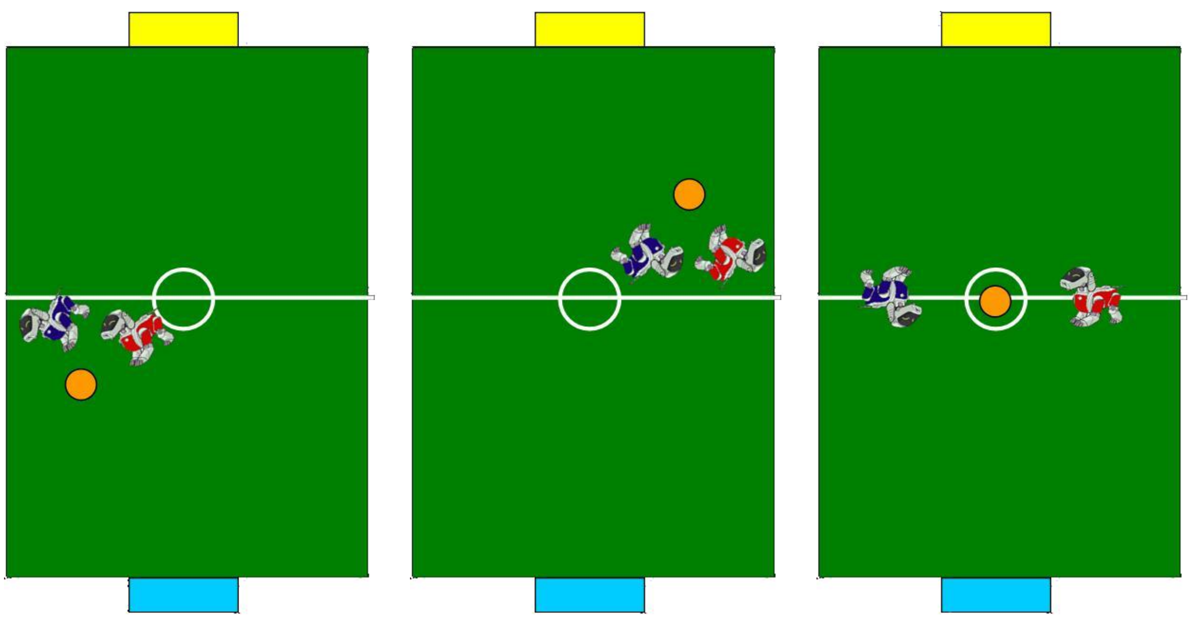

Figure 4.2: Experiment 2, robot starting configurations 1, 2 and 3.

Three different starting configurations are shown in Figure 4.2.

Configuration 1: This configuration presents a clear advantage to the red robot, which is shooting on the blue goal. The angle to the goal allows the robot to immediately shoot using one of the head kicks. If playing alone on an empty field, the robot would have a very high chance of scoring.

The purpose of this trial is to test whether the red robot is able to take advantage of this opportunity in an adversarial environment.

Configuration 2: In this configuration the roles of the two robots are reversed and the blue robot now has the advantage. The red robot must prevent the blue robot from scoring and clear the ball to the other half of the field before it has a good chance of scoring.

Configuration 3: This configuration places the robots in equal positions at the center of the field where neither robot has an initial advantage. The first robot to gain control of the ball would gain a slight advantage and the first chance to move the ball in its own direction.

Configuration	Blue (CMPack'02)	Red (Modeling)
Configuration 1	3	7
Configuration 2	4	6
Configuration 3	5	5

Table 4.2: Experiment 2, performance comparison of the CMPack'02 and the kick modeling selection algorithms in one-on-one competition. Values represent number of goals scored by each robot.

Each configuration was tested ten times. Table 4.2 summarizes the results of this experiment. Although the red robot shows better performance in the first two configurations, neither robot dominates in the final test. The red robot shows ability to take advantage of its position in Configuration 1 and scores seven of the ten goals. The blue robot fails to take advantage of a similar situation in Configuration 2 and remarkably the red robot scores six of the ten goals.

The tie in the case of Configuration 3 shows that in this experiment we do not see the marked performance improvement with the new algorithm as we did in Experiment 1. Several factors contribute to the difficulty of this experiment and the challenges of analyzing the results. One key factor is that because no vision algorithm in CMPack'02 is currently available for recognizing other robots on the field, both robots behave as if they are playing alone without an opponent. This oversimplification of the environment does not allow for strategy modification in the presence of other robots. Both robots are likely to kick the ball directly at an opponent, or walk into each other in an attempt to get to the ball. The second factor is that as the robots push each other in the fight for the ball, their localization estimates become extremely inaccurate, often causing them to kick in the wrong direction. For these reasons the results of this experiment are not as conclusive as the outcome of Experiment 1. Further tests must be performed once the vision system is able to recognize other robots on the field.

4.3 Summary

The kick selection algorithm used by the CMPack'02 behavior module used sets of hard coded rules in order to select which kick should be executed by the motions. This approach not only made it difficult to incorporate additional kick motions, but also limited performance due to its reliance solely on human intuition.

Using the experimentally acquired kick models, we introduce a new kick selection algorithm that relies on modeling and prediction to select more effectively which kick to use. Experiments with one robot on an open field show empirically that robot performance strongly improves with the new algorithm. Results of the second experiment, in which robots compete one-on-one, are less conclusive but also suggest improved performance over old algorithm.

Chapter 5

Conclusion

Quadruped motion remains a difficult research topic in robotics. Combined with the challenging domain of robot soccer, and its emphasis on the speed of motion and the accuracy of kicks, motion development becomes an even greater challenge. This thesis has three contributions to robotics research. We presented an empirical analysis of the sensitivity of the motion to parameter changes. We then introduced a set of kicks carefully designed in agreement with motion parameter analysis, and study their effects upon the robot's environment. We presented a method for using the acquired data to model kicking motions in terms of their effects on the environment. Finally, we described an approach for incorporating the effects model into the kick selection algorithm in the behavior module.

Every step of this research is followed with empirical results, comparing the performance of our algorithms to those of the Carnegie Mellon CMPack'02 robot soccer team. CMPack'02 became world champions in the Robot Soccer Legged League in the RoboCup 2002 competition. The kick effects model and the selection algorithm presented in this thesis have been incorporated into the CMPack'03 team code. In the recent RoboCup American Open competition the CMPack'03 team took first place, beating the Georgia Institute of Technology Yellow Jackets who were running an improved version of the CMPack'02 code. This demonstrates that the techniques introduced in this thesis have a significant impact on robotics research on the adversarial, multi-robot domains.