To appear in *The Fifth Workshop on Humanoid Soccer Robots* (HSR-10),
Nashville, TN, December 2010.

# Controlled Kicking under Uncertainty

Samuel Barrett, Katie Genter, Todd Hester, Michael Quinlan, Peter Stone

*Department of Computer Science, University of Texas at Austin*
*1616 Guadalupe St, Suite 2.408, Austin TX 78701, USA*
{sbarrett,katie,todd,mquinlan,pstone}@cs.utexas.edu

***Abstract*— In RoboCup, robots must make quick decisions under uncertainty. To this end, this paper introduces a new approach to enable humanoid soccer robots to execute kicks quickly and ensure that they move the ball down field. This paper presents a kick engine capable of kicking at a variety of distances and angles and then describes a novel kick decision method for selecting from among a large set of possible kicks. This method prunes and orders the kicks according to a metric and then chooses the first possible kick that ensures that our field position is improved. For the RoboCup 2010 challenge events, we took a more cautious approach, taking additional time to line up the ball and being more conservative in our kick selection. These methods proved successful at RoboCup 2010, as our UT Austin Villa team came in third in the soccer competition and second in the challenges.**

## I. INTRODUCTION

RoboCup, or the Robot Soccer World Cup, is an international research initiative designed to advance the fields of robotics and artificial intelligence, using the game of soccer as a challenge domain. The long-term goal of RoboCup is, by the year 2050, to build a team of 11 humanoid robot soccer players that can beat the best human soccer team on a real soccer field [1].

RoboCup is organized into several leagues, including both simulation leagues and leagues that compete with physical robots. The Standard Platform League (SPL)[1] uses teams of identical humanoid robots, making it essentially a software competition. Each team fields 3 Aldebaran Nao robots, shown in Figure 1. The game is played on a 6 by 4 meter field with two color-coded goals and line markings, shown in Figure 3.

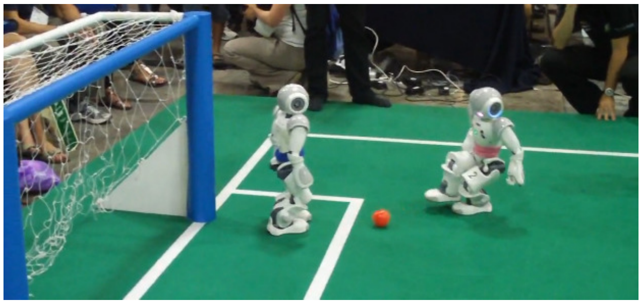


Fig. 1. An Aldebaran Nao robot kicking a goal on the RoboCup SPL field.

All of the decisions that the robot makes during the course of the game are made under a great deal of uncertainty. The robot is uncertain where it is on the field, both because of the limited landmarks available for localization, and because of the noisiness inherent in bipedal walking. In addition, there is uncertainty in the outcome of kicks as the exact heading and distance that the ball travels can vary widely. The robot also has uncertainty in the locations of its teammates and opponents.

The robot must deal with this uncertainty intelligently, as making a poor decision can have severe consequences. For example, when the ball is kicked out of bounds over the endline, the rules stipulate that it is moved back to midfield or a meter behind the robot, whichever is worse for the kicking robot's team.

In response to such strong incentives to control the ball, this paper introduces the UT Austin Villa kick strategy used in RoboCup 2010, which placed a high premium on acting quickly and being certain that each kick improves our field position. We created a kick decision engine that enables the robot to keep the ball away from the sidelines and to continue moving it towards the opponent's goal, all while avoiding opponents. To do this, we check if the result of the kick will place the ball in a dynamically calculated infundibuliform that forces the robot to funnel the ball towards the goal. To facilitate this strategy, we developed a kick engine capable of kicking the ball at variable distances and headings, and we use the target kick region to select from among the possible kicks the robot can make using its kick engine.

In addition to making sure all of our kicks move the ball consistently towards the opponent's goal, our second aim was to be *quick* at the ball. The final steps of approaching and lining up the ball for a kick can be quite slow, and opponents can get in the way of the kick if these phases take too long. For this reason, our robots take the first kick allowed by the strategy rather than spending more time to line up for a stronger, more accurate kick.

For the technical challenges, we used a more cautious approach since errors were more costly and speed near the ball was less important for these tasks. As such, we present an alternative approach for utilizing the control provided by our kick engine, which was evaluated in the challenges.

In this paper, we describe our kick engine in Section II, the kick decision strategy used during games in Section III, and the kick decision strategy used for the technical challenges in Section IV.

## II. KICK ENGINE

Rather than having a large set of static kicks for each situation the robot might encounter, we instead created a

---

[1] http://www.tzi.de/spl/

small set of parameterized kicks. Our kick engine selects the appropriate parameters and then executes the sequence of actions for a given kick. Our kick engine is designed to provide a wide range of angles and distances, as well as handle variance in the ball's starting position. This reduces the need to align carefully to the ball, which allows the robot to move the ball faster with less chance of an opponent blocking the kick. The engine selects the parameters based on the position of the ball and the desired kick target as described in the following sections. Specifically, the engine can handle desired kick angles from 10° to the inside of the kicking leg out to 30° to the outside, and distances ranging from 0.5 meters to 3.5 meters.

### A. Basic Control

The robot's kick engine used the state machine described in Table I. In our tests, separating moving the swing foot back and placing it down helped stability significantly. The Check Ball state stores the current position of the ball, calculated directly from the robot's camera image to eliminate the dependence of the kick on the filtering of the ball position. Furthermore, the head and body pose were fixed for the Check Ball state, so the robot did not need to filter out the noise from its own movement. Note that if the kick type does not specify a leg, the leg is chosen in the Check Ball state. For all of this work, we control the robot's joints using inverse kinematics.

| **State** | **Description** |
|---|---|
| Check Ball | Assume a standard position and check the current position of the ball |
| Shift Weight | Shift the robot's weight to the stance foot |
| Align | Lift the swing leg and position it for the kick |
| Kick | Move the swing leg forwards to perform the kick |
| Move Foot Back | Move the swing foot back to be in line with the stance leg |
| Place Foot Down | Place the swing foot back on the ground |
| Shift Weight Back | Shift the robot's weight back to balanced on both feet |

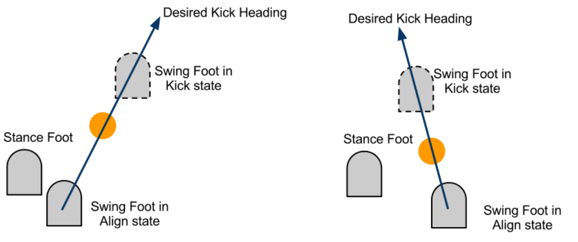
TABLE I

KICKING ENGINE'S STATE MACHINE

### B. Variable Ball Position and Angle Control

To handle variable ball positions and kicking at a range of angles, the robot must control the starting and ending positions of its foot for the kick. The goal is for the foot to move along a straight line through the ball at the desired kick angle, with the ball centered on the foot.

Most of the states defined above are not affected by the ball position and kicking angle; only the Align and Kick states must be altered based on these factors. The Align state must move the foot to the back point of the desired line segment, specifically 8 cm behind the center of the ball at the appropriate angle. The Kick state ends with the foot 4 cm in front of the original center of the ball. We use trigonometry to find these points from the angles, but bound both the side and forward distances of the foot to avoid singular values in the inverse kinematics and to prevent clipping the stance foot. Two examples of the foot placements for the kick are shown in Figure 2.



(a) Kicking to the outside. (b) Kicking to the inside.

Fig. 2. Foot placements during the Align and Kick states.

Note that in the Check Ball state, the robot saves the position of the ball in the image as a coordinate in pixel values from the camera image. For determining the positions of the foot, we directly use the ball's coordinates within the camera image. Unfortunately, not all of the robots are calibrated the same; as such, the same commanded angles of the head may result in different positions on different robots. Therefore, for each robot, we calculated the offset of the head and used this value to correct the estimate of the position of the ball.

### C. Distance Control

Kicking distance was controlled by changing the interpolation time of the Kick state, i.e. the time taken to swing the leg forwards during the kick. We determined the relationship between this time and the distance empirically, collecting distances for varying interpolation times and averaging over three kicks. Then, we fit a linear model to the logarithm of the distance, arriving at the model:

$$t = -0.215 \log(d) + 1.824$$

Also, we discovered that using times of less than 0.05 seconds had no effect on the kick due to the maximum speed of the joint and times longer than 0.5 seconds did not reliably result in the robot moving the ball at all. It is important to note that the function relating distances and interpolation times is dependent on the field surface, so it was necessary to recalibrate it for the RoboCup competition.

## III. KICK SELECTION STRATEGY

Having defined the method for implementing a quick kick with controllable kick and angle, we are then presented with the challenge of selecting from among all the possible ways in which to kick the ball at any given time. In this section, we describe our approach to this problem. One of our main objectives was to minimize the time we spent at and around the ball, while still maximizing the outcome of each kick. To achieve this we construct a parametrized system that allows us to define a valid *kick region* that ensures that any kick chosen will result in the ball being moved closer to the opponent's goal and away from the sidelines. The robot has a set of possible kicks that can be made using the kick engine, and determines which ones are *valid* by determining if they will likely result in the ball being in the kick region. The potential kicks are ordered based on distance, and if there are multiple valid kicks, the kick that goes the farthest is selected. When no valid kicks exist, the robot continues to approach the ball, which consists of walking up to and circling it, until a viable kick is found. By following this approach the robot spends the minimum time approaching the ball, while still guaranteeing that the chosen kick will move the ball closer to the goal.

### *A. Defining the Kick Region*

The kick region defines an area on the field which we deem as *valid* to kick into. This region ensures that the robot always kicks the ball towards the opponent's goal, but away from opponents and the sidelines. An example valid kick region for a particular ball position is represented by the shaded area in Figure 3. The principle is that any kick, factoring in the robot's possible orientation error, that will result in the ball staying inside the kick region is a *valid* kick to choose from. On the other hand, any kick that will not place the ball in this region is an *invalid* kick that should not be selected.

The kick region is described by 6 parameters, presented in Table II and visualized in Figure 3. The general concept is that the region allows most kick options (to keep the ball moving) but funnels the ball towards the goal, therefore keeping the ball out of the corners of the field. We can define the steepness of the funnel stem, the width of the funnel, and the opening angle of the funnel's mouth. The kick strategy can be changed at any stage during the match. For example, we can have different strategies depending on the game situation or even depending on the current field setup.

| **Parameter (unit)** | **Description** |
|---|---|
| Edge Buffer (mm) | Creates a non-valid region of this width on each sideline. |
| Post Angle (deg) | Shapes the valid region such that it narrows towards the target goal with the given angle. |
| Opening Angle (deg) | Defines the valid region as angled outwards from the robot, i.e. always kick forwards at a minimum of the given angle. |
| Inside Post Buffer (mm) | Minimum distance inside each goal post that the robot should aim. |
| Shooting Radius (mm) | Defines a semi-circle in which the robot should strongly attempt to score, i.e the valid kick regions becomes the area inside the target goal. |
| Own Goal Radius (mm) | Defines a semi-circle around your own goal in which no valid kicks should exists, i.e. do not kick across the face of your own goal. |

TABLE II

PARAMETERS THAT DEFINE THE VALID KICK REGION

### *B. Avoiding Opponents*

In addition to making sure that all kicks move the ball towards the opponent's goal, the kick region is also used to ensure that the ball is not kicked towards opponents. When an opponent robot is detected in the robot's camera image, its location and the time it was seen are saved to memory. The location of this opponent is remembered for 6 seconds, after which point we assume there is a high chance that the opponent has moved. All locations behind the opponent robot, and locations up to 20 cm in front of the opponent robot, are considered to be invalid. This prevents the robot from selecting a kick that would attempt to kick at or through an opponent.

Along with using the opponents to invalidate parts of the kick region, we also use them to calculate the best heading towards the goal. This heading gives us a possible direction to aim our kick to maximize the chance of scoring by shooting directly between the keeper and one of the goal posts. We calculate the bearing to each goal post and the bearing to any opponent between the two posts. We then calculate the size of the 'gap' between the opponent and each post in degrees. The larger of these two is considered the 'large gap' and the smaller is the 'small gap'. Along with the kicks at discretized distances and headings, two kick choices are added that kick at maximum distance towards the center of each of the two gaps. If no opponent is detected between the goal posts, the large gap kick is directed towards the center of the goal, and the small gap kick is directed to be just inside the near post.

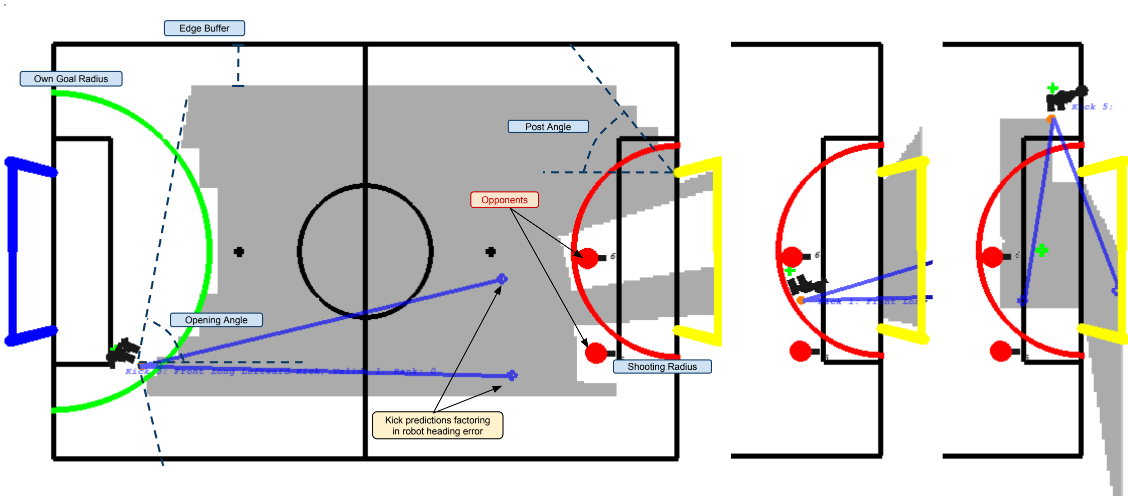
### *C. Choosing a Kick*

Algorithm 1 shows how the robot decides when and where to kick using the kick region. The robot has a set of possible kicks it can make, using a small set of the kick ranges and angles possible with our kick engine. Table III-C shows the kick choices that are given to the robot. The first two choices use kick headings calculated to aim towards the gaps between the goal keeper and the goal posts. The remaining kicks use discretized headings of $-30°$, $0°$, and $30°$ with discretized distances of 0.85, 1.75, and 3.5 meters. The kicks are sorted based on distances and headings, such that longer kicks are examined first by the kick selection algorithm.

The robot goes through each of these kicks in order, and checks if they are valid by checking if the final ball location, including likely heading error, is in the valid region. For each kick, the robot adds both a positive and negative heading error to the kick based on the robot's localization uncertainty and the variance in the kick itself. These two locations indicate the maximum and minimum heading that we expect the ball might move. The robot calls *check_kick_region* for each of these locations, and it determines if this possible ball location is in the valid region. If both possible ball locations are valid, then the kick is a valid kick, otherwise, it is invalid.

The robot calls *approach_ball_arc_to_goal*, which walks towards and circles the ball to face the opponent's goal. It returns false after a timeout for circling the ball too long. Upon finding a valid kick, the algorithm stops circling the ball and executes that kick. If none of the kicks are valid, then the robot continues its approach until it has a valid kick or the approach times out. It is often the case that a short kick may be valid while a long kick would put the ball out of bounds, but had the robot circled the ball more it could have reached an angle such that a longer kick would have been better. This biases our robot to take faster, shorter kicks as opposed to longer ones that require more alignment.

Figure 4 shows all the possible kicks from one field location. The first two choices aim for the gap between the keeper and the goal posts. The robot is close enough to the goal and has



(a) Kick Region Parameters (b) Shooting Radius (c) Corner Behavior

Fig. 3. (a) The parameters that define the kick region and an example of avoiding opponents. The highlighted area indicates the valid kick region. (b) Example of the kick region when the ball is inside the shooting radius (the semi-circle). In this case, the robot will strongly prefer to score a goal; this is achieved by setting only the area inside the goal as valid kick region. (c) When the ball is in the corners, we modify the kick region to include an additional area in front of the goal.

**Algorithm 1** Kick Strategy

```
$b \leftarrow ball$
$r \leftarrow robot$
while approach_ball_arc_to_goal($A$) do
    if $b_{dist} > A_{MAX\_DISTANCE\_FROM\_BALL}$ then
        continue
    end if
    $best\_kick \leftarrow -1$
    for $k \in K$ do
        $left \leftarrow$ check_kick_region($R$, $k_{dist}, k_\theta, r_{\theta_\sigma}$)
        $right \leftarrow$ check_kick_region($R$, $k_{dist}, k_\theta, -r_{\theta_\sigma}$)
        if $(left + right) = 2$ then
            $best\_kick \leftarrow k$
            break
        end if
    end for
    if $best\_kick > -1$ then
        execute $K_{best\_kick}$
    end if
end while
execute $K_{SHORT\_KICK}$
```

low enough localization uncertainty that it believes that both the left and right possibilities for these kicks will result in scoring. Therefore, it will determine that both of these kicks are valid and return Kick 0, towards the largest gap between the keeper and goal post. The validity of the other possible kicks is shown. Some of the kicks are invalid due to going out of bounds (Kicks 3 and 4), going towards the opponent (Kicks 2 and 5), or not meeting the opening angle from the robot (i.e. not going forward enough (Kicks 6 and 9)). In addition to the two scoring kicks, two of the shorter kicks (Kicks 8 and 10) are valid, but the longer scoring kicks would be preferred.

| **Kick** | **Distance (m)** | **Heading (deg)** |
|---|---|---|
| Large Gap Kick | 3.5 | Towards Large Goal Gap |
| Small Gap Kick | 3 | Towards Small Goal Gap |
| Long Straight Kick | 3.5 | 0 |
| Long Leftward Kick | 3.5 | 30 |
| Long Rightward Kick | 3.5 | -30 |
| Medium Straight Kick | 1.75 | 0 |
| Medium Leftward Kick | 1.75 | 30 |
| Medium Rightward Kick | 1.75 | -30 |
| Short Straight Kick | 0.85 | 0 |
| Short Leftward Kick | 0.85 | 30 |
| Short Rightward Kick | 0.85 | -30 |

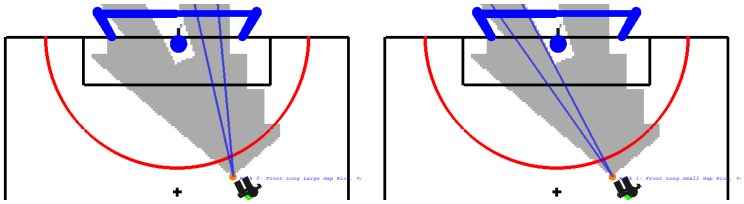
TABLE III
POSSIBLE KICKS

## IV. TECHNICAL CHALLENGES

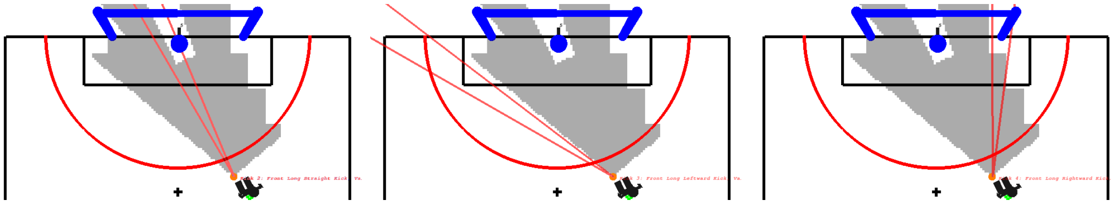
In this section, we describe our approach to kick selection for the passing and dribbling challenges. The passing challenge exhibited our ability to execute kicks over various distances, while the dribbling challenge allowed us to experiment with a different process of making kick decisions than what was used in games. In the passing challenge, the distance and accuracy of each kick were of the utmost importance. Our ability to accurately execute kicks over varying distances is the main reason we were able to win the passing challenge. In the dribbling challenge, we were most concerned with ensuring that the ball did not leave the field or contact a defender while we advanced the ball up the field. To achieve this, we based our kick decisions on the location of defenders in a relative occupancy grid. Using this approach, we were able to successfully complete the challenge in second place.

### A. Passing Challenge

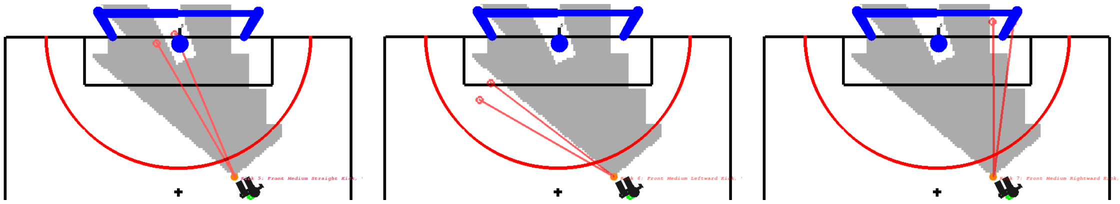
In the passing challenge, three successive passes must be made back and forth between two robots across the center of



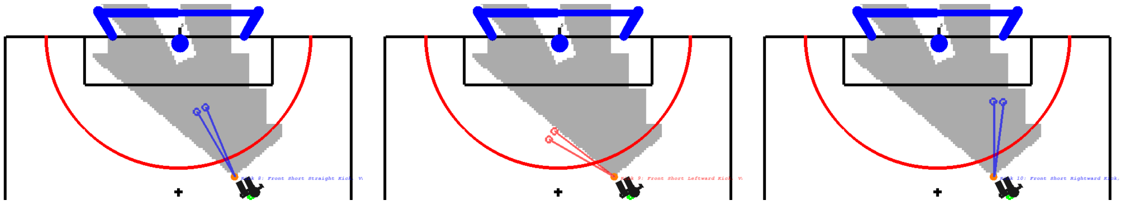
(a) Kick 0: Long Kick towards Large Gap. (b) Kick 1: Long Kick towards Small Gap.



(c) Kick 2: Long Straight Kick. (d) Kick 3: Long Leftward Kick. (e) Kick 4: Long Rightward Kick.



(f) Kick 5: Medium Straight Kick. (g) Kick 6: Medium Leftward Kick. (h) Kick 7: Medium Rightward Kick.



(i) Kick 8: Short Straight Kick. (j) Kick 9: Short Leftward Kick. (k) Kick 10: Short Rightward Kick.

Fig. 4. Example of kick choices from a location on the field. Kicks 0 and 1 are targeted at the gaps around the keeper. Kicks 2 and 5 are invalid because they would go straight at the keeper. Kicks 3, 4, 6, 7, and 9 go outside the valid region. Kicks 8 and 10 are valid, but have a lower ranking than valid kicks 0 and 1. Kicks are sorted by length, and the first valid kick is selected. So in this case, Kick 0 would be selected and the remaining kicks would never be evaluated.

the field. The trial ends in failure if the ball or one of the robots leaves the field, if the ball stops inside the dead zone in the middle of the field or if one of the robots enters the dead zone. The trial ends in success once the receiving robot touches the ball after the third pass.

*1) Using Variable Length Kicks:* The passing challenge served as an excellent application of our team's variable length kicks described in Section II. In the passing challenge, our approach was to have a player wait in the middle of each side of the field between the penalty cross and goal box. When the ball is on a player's side of the field, the player approaches the ball and aligns to kick straight to the penalty cross on the other side of the field. We calculate the desired kick distance as the distance from the ball to the penalty cross on the other side of the field, and give this distance and a 0° heading to the kick engine. It is important to note that unlike in games, we align exactly to our target and pass a 0° heading to the kick engine in this challenge. Although this takes more time, the added accuracy and consistency is worth the additional time in the challenges.

The use of variable length kicks allowed our team to kick the ball to approximately the same place on the field, regardless of where we kicked the ball from. This allowed us to recover robustly from a poor pass, as we could kick the ball softly if the previous pass barely crossed over the dead zone or with more power if the previous pass ended up near the goal box. Variable length kicks also allowed us to kick the ball any distance within our kicking range, as opposed to only being able to choose from a limited number of predefined kick distances and often having to select a shorter or longer kick than desired.

*2) Results:* Twenty teams attempted the passing challenge, but only three teams completed three passes in three minutes or less. We claimed first place with the fastest completion time of 1 minute and 26 seconds, beating the second place team by 17 seconds and the third place team by 62 seconds.

### B. Dribbling Challenge

In the dribbling challenge, three stationary opponents were placed on the field such that direct kicks to the goal would be unsuccessful. The dribbling robot starts near its goal, and must dribble the ball to the other side of the field and score in three minutes or less. However, the robot must be cautious because the challenge is restarted with the remaining time when the robot kicks the ball out of bounds or into a defender, or if it collides with a defender.

Although the dribbling challenge is designed to encourage development of skills that are useful in games (such as flexible ball manipulation, obstacle detection, and avoidance skills) our process of kick selection for the dribbling challenge is inherently different than the process used in games. For example, in games it is important to minimize the time we spend at the ball. However, in the dribbling challenge, much like in the passing challenge, it is better to take as much time as needed to ensure that our kicker is well aligned to the ball and the correct kick is chosen. As such, kick decision behavior for the dribbling challenge can afford to be slower, and needs to be more conservative in the kicks that it chooses. Due to these differing needs, we opted to base our kick decisions for the dribbling challenge on the location of defenders in an occupancy grid instead of using the kick decision strategy described in Section III.

*1) Using a Relative Occupancy Grid:* We implemented a relative occupancy grid to help with kick selection in the dribbling challenge. Our occupancy grid is relative in the sense that it keeps track of where opponents are in relation to our dribbling robot. We update the relative occupancy grid every cycle based on the opponent robots detected via vision.

In the dribbling challenge, we choose between a long straight kick, a short straight kick, short 30° kicks, 80° side kicks, and dribbling. Our kick decisions are based on our position on the field and the proximity of the defenders. We use the relative occupancy grid to determine which kick trajectories hit a defender, and the distance to these defenders.

We always choose the most effective kick that is safe, where preference is given to kicking in the direction that brings us closer to the middle of the field when everything else is equal. We consider a kick "safe" if its path does not hit any defenders, including a buffer in both the direction and distance of the kick. Effective kicks are defined as those that advance the ball down the field; as such, we want to kick straight if it is safe and avoid side kicking or dribbling unless no straight kicks or 30° kicks are safe.

*2) Results:* Fourteen teams attempted the dribbling challenge, but only five teams successfully completed the challenge in three minutes or less. Requiring one early restart after kicking the ball into a defender, we finished in second place. Our final time in the challenge was 2 minutes and 23 seconds, just 8 seconds slower then the winner.

## V. RELATED WORK

B-Human, a robot soccer team from the University of Bremen, won the 2010 RoboCup competition. For their kicks, they considered the trajectory of the leg through space and time [2], [3]. Overall, their kick engine was more elaborate than ours, and allowed for the development interesting kicks, such as a backwards kick. However, their system was not as configurable with respect to the position of the ball or the desired kick angle. Instead, they developed several kicks and selected the most appropriate one.

Another approach to dynamic kicking was investigated by Yuan Xu and Heinrich Mellman [4]. Their approach allows for variable starting positions of the ball as well as kicking at different angles. While their approach is promising, it required significantly more computation than ours and still had problems with fine control of the ball.

In addition to our work on dynamic kicking, there is also work related to our kick decision-making. Specifically, B-Human [3] addresses the problem in a different way. While our decisions on when and where to kick were based on taking the first available kick that was acceptable, B-Human planned out the amount of time each kick would take, both in terms of the time to approach and kick the ball. They used this information in conjunction with the stability of the kick and how strong it is to select a kick.

## VI. CONCLUSIONS

This paper introduces the UT Austin Villa 2010 RoboCup team's humanoid robot soccer behavior that approaches and kicks the ball quickly, while consistently moving it down field. This work was broken into two parts: a new kick engine and a new algorithm for selecting kicks. We developed a new kick engine that enables the robot to kick at a variable distance and heading. We paired this with a novel kick decision engine that ensures that any kick the robot takes will move the ball downfield while keeping it in-bounds and away from opponents. This kick decision engine takes into account the agent's uncertainty in its location and the kick's outcome, while limiting the amount of time taken to approach the ball. We also created an alternative kick decision approach used for the RoboCup challenges.

The combination of these two developments resulted in a behavior that moved the ball downfield quickly. It resulted in great success at RoboCup, as our team placed third in the competition and second in the technical challenges.

## ACKNOWLEDGEMENTS

This work has taken place in the Learning Agents Research Group (LARG) at the Artificial Intelligence Laboratory, The University of Texas at Austin. LARG research is supported in part by grants from the National Science Foundation (CNS-0615104 and IIS-0917122), ONR (N00014-09-1-0658), DARPA (FA8650-08-C-7812), and the Federal Highway Administration (DTFH61-07-H-00030). Thanks to the entire UT Austin Villa robot soccer team. Samuel Barrett was supported by an NDSEG fellowship.

## REFERENCES

[1] H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, and E. Osawa, "RoboCup: The robot world cup initiative," in *Proceedings of The First International Conference on Autonomous Agents*. ACM Press, 1997.

[2] J. Müller, T. Laue, and T. Röfer, "Kicking a ball - Modeling complex dynamic motions for humanoid robots," in *RoboCup 2010: Robot Soccer World Cup XIV*, ser. Lecture Notes in Artificial Intelligence. Springer.

[3] T. Röfer, T. Laue, J. Müller, A. Burchardt, E. Damrose, A. Fabisch, F. Feldpausch, K. Gillmann, C. Graf, T. J. de Haas, A. Härtl, D. Honsel, P. Kastner, T. Kastner, B. Markowsky, M. Mester, J. Peter, O. J. L. Riemann, M. Ring, W. Sauerland, A. Schreck, I. Sieverdingbeck, F. Wenk, and J.-H. Worch, "B-Human team report and code release 2010," 2010.

[4] Y. Xu and H. Mellmann, "Adaptive motion control: Dynamic kick for a humanoid robot," in *KI 2010: Advances in Artificial Intelligence*, ser. Lecture Notes in Computer Science, R. Dillmann, J. Beyerer, U. Hanebeck, and T. Schultz, Eds. Springer Berlin / Heidelberg, 2010, vol. 6359, pp. 392–399, 10.1007/978-3-642-16111-7_45.