THE UNIVERSITY OF NEW SOUTH WALES
THE SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

Ball Modelling and its Applications in Robot Goalie Behaviours

By: Belinda Teh
tehbelinda@cse.unsw.edu.au

A Special Project B submitted for the degree of
COMPUTER SCIENCE

August 24, 2011
Supervisor: Bernhard Hengst

Abstract

During the previous summer of Taste of Research, rUNSWift's position-based ball model had been improved through the addition of ball velocity. However, its methods were still rather primitive and did not accurately model other world variables. Similarly, only basic goalie behaviours had been written to test the new velocity features, while a complete goalie needed to be created for the final competition. Since tracking the ball and being able to defend one's goals are such vital aspects in any soccer game, this project aims to further extend this work. This report presents the improvements made to the Unscented Kalman Filter used for the ball model and the eventual components of rUNSWift's goalie behaviour for 2011. Overall, the project was successful in that the ball position and velocity were both accurate enough to track a ball and determine if it were moving or stationary. The goalie was sufficiently versatile as it was able to dive to save a goal, as well as move forward to kick balls away from the goals. Unfortunately, it was not always 100% reliable, especially with localising within the goals remaining an issue. Needless to say, this project has paved the way for a vast potential of future work and been a significant improvement over the ball model and goalie skill of rUNSWift 2010.

Contents

1 Introduction **4**

1.1 Tracking The Ball 4

1.2 Goalie Behaviour 5

1.3 Report Outline 5

2 Related Work **6**

2.1 Ball Modelling 6

2.1.1 Aibos of the Past 6

2.1.2 BHuman 2010 6

2.1.3 rUNSWift 2010 7

2.1.4 Filters 7

2.1.5 Modelling Uncertainties and Adjustments 7

2.2 Goalie 8

3 Modifications to the Ball Filter **10**

3.1 Unscented Kalman Filter 10

3.2 Current Issues 11

3.2.1 Cartesian Co-ordinates 11

3.2.2 Dual Modal 13

3.2.3 Odometry 14

3.2.4 Head Pitch and Yaw 14

3.2.5 Team Ball 15

4 Goalie Components **16**
4.1 Behaviour . . . 16
4.1.1 Find and Track Ball . . . 17
4.1.2 Patrol Goal . . . 17
4.1.3 Localise . . . 19
4.1.4 Defend Goal . . . 19
4.1.5 Attack Ball . . . 21
4.1.6 Reposition . . . 22
4.2 Motion . . . 22
4.2.1 Blocking The Centre . . . 22
4.2.2 Blocking The Sides . . . 25
4.2.3 Recovery . . . 25
4.2.4 Tracking From Afar . . . 26
4.2.5 Reducing Joint Stress . . . 27
4.3 Penalty Behaviour . . . 27

5 Results **29**
5.1 Ball Tracking . . . 29
5.2 Goalie Motion . . . 30
5.3 Behaviour at Istanbul 2011 . . . 30

6 Discussion **33**
6.1 Ball Tracking . . . 33
6.2 Motion . . . 34
6.3 Overall Behaviour . . . 34

7 Future Work **36**
7.1 Filter Improvements . . . 36
7.1.1 Multi-Modal Distributed Ball Filter . . . 36
7.1.2 Modelling Uncertainty . . . 36

7.2 Goalie Motions . . . 37
7.2.1 Improvements . . . 37
7.2.2 Future Actions . . . 37
7.3 Integrating Velocity . . . 38
7.4 Field Line Localisation . . . 38
7.5 Incorporating Robot Detection . . . 38

8 Conclusion **39**

Bibliography **40**

Chapter 1

Introduction

1.1 Tracking The Ball

In any game of soccer, strategies tend to be based around the ball, so knowing its position is very important. The same applies to the Robocup Standard Platform League where teams of four robots must co-ordinate themselves to play games of soccer. A robot will not be able to get to a ball and kick it into the goals if it does not know where the ball is in the first place. Now, to be able to efficiently defend the goals, a goalie must be able to track not just the position of a stationary ball, but the velocity of a moving ball. If it has this additional information, it can estimate where the ball's next position will be and what direction the ball is travelling in, thus knowing which area of the goal it should protect.

Effective ball modelling can also be helpful in many other situations. If a robot had just lost sight of the ball, whether it be due to obstruction by another robot or a missed frame, the ball velocity could be used to estimate its new position. This could also be applied to the team as a whole, so that all robots could work together to estimate the location of the ball. As such, it was highly motivating that the ball model be improved to be as accurate as possible.

Since the only available input data is the current relative position of the ball provided from the vision module, the filter used to integrate this information over time must also model the other variables in the robot's world. Due to noise and inaccuracies in the sensors as well as unpredictable movements during game-time, achieving an accurate representation can be quite challenging. We propose to use a dual-mode Unscented Kalman Filter for tracking both a stationary and a moving ball with added adjustments to account for uncertainties in the robot's state, such as its odometry and localisation. This approach is able to handle typical game-time noise and model the ball in its different modes accurately enough for a robot goalie to react appropriately. Although this report addresses the issues of ball tracking within the Robocup Standard Platform League, similar techniques can be applied to object tracking in general for other usages.

1.2 Goalie Behaviour

As the last line of defense and the only robot of its team allowed in the goal box, it is imperative for the goalie to be able to effectively block goals. Previous goalies simply stood in the centre of the goals, however with a more advanced ball model comes the potential for more advanced goalie behaviour. With the improvements in the ball model and addition of velocity, the goalie can calculate which side it should position itself in to block as much of the goal as possible, or even dive if need be. Improvements also need to be made to the different motion stances of the goalie so it can effectively block goals with little damage to the hardware.

The goalie can also be used to track the ball and aid its team members by sending out this information, as since the goalie tends to move the least, it is less susceptible to noise and inaccuracies in its localisation. Once the ball is close to the goal box, the goalie can also be made to clear the area since it is usually already in the prime location and facing the right direction. The aim is to use a state machine to incorporate all the different aspects of the goalie's behaviour, with different tasks prioritised appropriately.

1.3 Report Outline

Chapter 2 provides some insight into the foundations behind ball tracking and past goalie behaviours. Chapter 3 continues on to the methods used to improve the ball filter while Chapter 4 presents the new components in the goalie. Chapter 5 and 6 present the results and a discussion of the overall project, Chapter 7 offers suggestions for future possibilities, with Chapter 8 finally concluding the report.

Chapter 2

Related Work

2.1 Ball Modelling

2.1.1 Aibos of the Past

In the past Robocup 4-Legged League, tracking the ball was a similar problem to what it is now. rUNSWift's previous teams had used various versions of the Kalman Filter and by 2004, a multi-modal Extended Kalman Filter was implemented to track the state of the robots with a separate filter for the ball. By 2006 these had been combined into one extensive filter which took advantage of the correlation between robot pose, ball position and ball velocity.[10] These methods proved to be quite successful as the rUNSWift team placed top three in the international competition for several years.

Nubots from the University of Newcastle similarly used an Extended Kalman Filter with both ball and robot information combined into one world state in 2005.[7] They also proved this method quite successful by coming second place, and in fact, this method became a basis for many other teams including rUNSWift themselves.[10]

However, the differences between the hardware of the Sony Aibos in the 4-Legged League and that of the Aldebaran Naos in the current Standard Platform League as well as the differences in field structure mean that the problems faced in filtering robot and ball information are no longer quite the same. Regardless, the approach to filtering ball information through the use of a Kalman Filter can still be similar.

2.1.2 BHuman 2010

As the current reigning champions of Robocup's Standard Platform League, it is no surprise that B-Human have quite a sophisticated ball filtering system in place. In 2010, they moved from using a Particle Filter to estimate the velocity and position of the ball to using twelve Kalman Filters instead. Of note, these twelve included filters modelled specifically for stationary and moving balls. Additional calculations were made for kicked or obstructed balls by taking into account the motion of whatever the ball had been clipped against.[9]

2.1.3 rUNSWift 2010

During 2010, rUNSWift only used a basic Kalman filter to track the ball. This was done for a robot-relative position which was typically the most accurate, an egocentric absolute ball position which incorporated the robot's position, and finally a shared team ball position which was based off every robot's egocentric ball filter.[8] The problems with this approach was that the basic Kalman Filter did not take into account any non-linearities and did not track the ball's velocity. Although this method did work and rUNSWift succeeded in coming second in the competition at Singapore, it did not take advantage of the potential benefits of having additional ball information. During the past summer of 2010-2011, a Taste of Research project was undertaken to improve rUNSWift's ball modelling, though there was still much work to be done.[11]

2.1.4 Filters

As can be seen from past results, a more sophisticated Kalman Filter approach that tracked both position and velocity was definitely worth pursuing. Although using a Particle Filter to approximate the mean and track multiple modes through the weighting of a large number of particles had been common in the past[6, 9], its complexity was far too high for the Naos. As such, Extended Kalman Filters had been accepted as the new norm for tackling ball tracking, as can be seen through its use by the top teams in recent years. They expanded upon the basic Kalman Filter by using the derivative of the predict and update functions to find linear approximations.[3] However, an Unscented Kalman Filter was chosen in the hope that its use of sigma points a standard deviation around the mean and covariance would return an even truer estimate. As found by Wan and Van der Merwe,[13] this deterministic sampling can achieve an accuracy up to the third order for any nonlinearity, as opposed to the Extended Kalman Filter's first order accuracy. Furthermore, their expansion on the Unscented Kalman Filter with the use of square root forms added numerical stability, guaranteed positive semi-definite covariance matrices, and reduced its computational complexity.[12] Another reason to use the Unscented Kalman Filter was that it was also being trialled for a new robot localisation filter[1], and the ideal was to use a common base since they were similar tracking problems.

2.1.5 Modelling Uncertainties and Adjustments

However, though the approach was theoretically sound, there were still a lot of inaccuracies with the filter as many of the inputs in to the filter were not properly modelled. The Darmstadt Dribblers of Robocup's Humanoid League for example, which also share the common problem of ball tracking, offer a method of rectifying the issues of modelling updates from the camera in particular. There are typically two approaches for estimating the distance to the ball: intersecting the camera-to-ball vector with the ground plane and measuring based on size. Each method produces different error characteristics, and by combining the two, Darmstadt were able to use two Gaussian estimates to optimally fuse the means and variances.[4] As the champions of their league in 2009 and 2010, with 74 goals scored for and only 2 against at the Singapore competition, their ball tracking is definitely worth taking note of. The need to improve the level of rUNSWift's approach in a similar fashion thus led to the additions made in this project.

2.2 Goalie

During previous years, the typical goalie behaviour would simply be to stand in the centre of the goals to defend it. Some would attempt to patrol the goals, although this would often be problematic due to the difficulties of localising from within the goal box. Some would not even bother, but instead just treat their goalie as another supporter player.

In 2010, BHuman reshaped the scene with their versatile diving goalie. By manually specifying each of the joint angles and changing the stiffness of each at different times, they were able to move their goalie in the least damaging way possible. When the dive reached its falling point, the goalie's joints could be made limp such that there would be no impact on the motors when contacting the ground. They also had a slightly wider centre stance to block the goal by spreading the robot's legs in case of close encounters, though this could sometimes be prone to ball holding.

Behaviourally, if the ball was calculated to roll straight towards the goalie or not even intersect with the goal, then their goalie would remain in its prepared crouching position. It should be noted that this kind of accuracy was possible due to the sophistication of their ball filtering. However, they did have some slight errors as shown by their timing delay in switching from their prepared crouching position to standing as normal. This was done in order to avoid a constant cycle between the two. With their stationary ball filters, their goalie was able to detect if the ball was close and had stopped moving, then move to kick it away. Additionally, the goalie was used to aid the team in finding the ball and tracking its information for all to use. BHuman's goalie was ultimately extremely successful, and by aiding in finding the ball and saving several goals, proved itself an asset to the team. [9]

Meanwhile, rUNSWift 2010's goalie was still rather basic. Its patrol state aimed to stand between the centre of the goal and the ball, but was plagued with localisation problems. It was not fully incorporated into team play, but simply knew whether or not it should go for the ball if another team mate was within one metre of it. It did have a crouching position, but this was aimed at reducing stress on the joints. Overall, this meant that the goalie was not particularly effective at blocking goals, and would sometimes clash with fellow team members when going for a ball near the goals.[8]

As part of a Taste of Research project over the summer from 2010-2011, attempts were made to address the lacking components of the goalie. Implementing the ability to manually specify stiffness on a per joint basis rather than a per action basis, as well as a ball filter that tracked velocity, allowed for the creation of rUNSWift's own goalie dive.[11] However, customised motions for the goalie are not limited to diving only. A centre squat for blocking the goals when the ball is rolling towards the goalie needs to be created, in particular, one that does not succumb to ball holding. If its range can be made wide enough such that the need to dive is reduced, this will greatly benefit the team as diving can damage the robots and reset their localisation.

With the issue of team play, the goalie needs to be incorporated into role switching and, like the supporter robots, be treated with the potential to become another attacking robot. Ideally, it should use similar code to that of the striker robot to attack the ball. We also aim to tackle the situations when the ball is not near the goalie's own goal box, but on the far side of the field. Typically the goalie is only made to react once the ball gets closer to its half, rendering it useless if the ball is down the other end of the field. However, if the goalie could track the ball across the whole field whilst remaining localised, its ball information would be invaluable to its team members. To assist with detecting the ball on the other side of the field, we propose a new goalie motion which involves the robot standing tall and straight to increase its height, thus increasing its visibility.

Chapter 3

Modifications to the Ball Filter

3.1 Unscented Kalman Filter

The base Unscented Kalman Filter this project expanded on was first formed during the summer of 2010-2011 as part of a Taste of Research project.[11] It was chosen over the other Kalman Filter options for two main reasons. Firstly, the same base filter was to be used in filtering the robot's location, and secondly, the Unscented Kalman Filter's method of estimating sigma points of 1 standard deviation around the mean and covariance would ideally provide for truer weighted estimates.[1, 13]

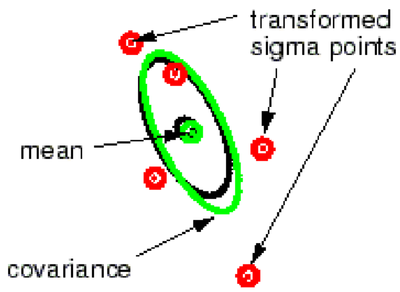

Figure 3.1: Transformed sigma points around the mean and covariance of an Unscented Kalman Filter

At that stage, the filter state tracked four variables: the ball's robot-relative distance, the ball's robot-relative heading, the change in the ball's robot-relative distance per millisecond, and the change in the ball's robot-relative heading per millisecond. Key calculations in this filter included the Cholesky decomposition to calculate the square root of the covariance matrix, which was handled through the use of libeigen. Also of note was the multiplication of the Jacobian Motion Matrix to generate values for velocity[10] during the time update phase of the filter.

$$Position: \ d_{new} = d_{old} + d't$$

$$Velocity: \ d'_{new} = 0.9d'_{old}$$

$$State = \begin{pmatrix} distance & distance' & heading & heading' \end{pmatrix}$$

$$Motion\,Matrix = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0.9 & 0 \\ 0 & 0 & 0 & 0.9 \end{pmatrix}$$

The entries in $(2, 0)$ and $(3, 1)$ correspond to the distance' and heading' respectively, and are formed by taking the derivative of the functions for the position and velocity updates with respect to time t. Entries $(2, 3)$ and $(3, 3)$ signify the change in velocity due to friction, and were simply set to a constant of 0.9. The values in the diagonal matrix can also be viewed as the derivative of the functions for the position and velocity updates with respect to themselves.

3.2 Current Issues

Despite the theoretical ideals of Unscented Kalman Filter, it still suffered from many problems, especially due to unknown variables of the robot's changing world and inaccuracies in the Nao's hardware. Some of the issues experienced and the methods taken to address them are explained below.

3.2.1 Cartesian Co-ordinates

It was discovered that the Unscented Kalman Filter could not handle the tracking of angles in its state which resulted in rather inaccurate estimates, particularly when exposed to noise. This was due to the nature of robot-relative heading wrapping around from -180 to 180 degrees. Attempts to normalise the angles throughout the filter were unsuccessful, especially since only the robot-relative heading, as opposed to the change in robot-relative heading, needed to be capped.[1] There was also a possibility of instabilities with the libeigen calculations involving heading, as they were extremely small values due to sampling at frequent intervals (as dependent on the frame rate) and being stored in radians.

Fortunately, this problem in the ball filter could be solved by filtering the robot-relative Cartesian x and y co-ordinates as opposed to the Polar distance and heading co-ordinates. Most of the key calculations remained the same, however the polar data received in the observation update needed to be converted appropriately before being fed into the filter.

$$x = distance \times \cos(heading)$$

$$y = distance \times \sin(heading)$$

It should be noted that the multiplication of cos and sin are switched around for x and y due to the system used for the robot's relative co-ordinates.

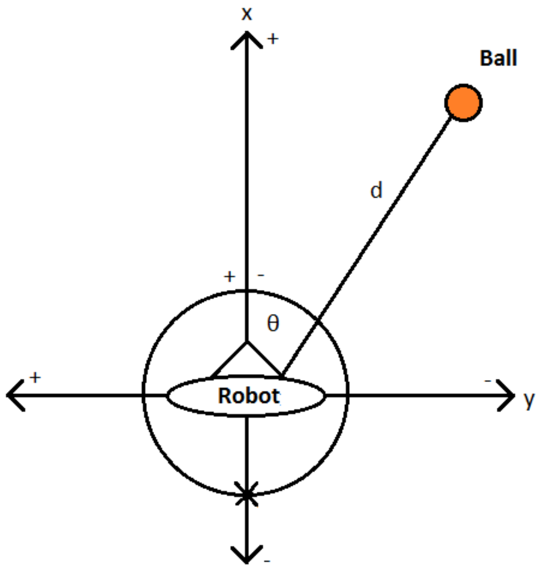

Figure 3.2: Robot relative co-ordinate system

Since the observation data still matched the format of the state in that they were both Cartesian, the prediction update remained quite similar.

$$New\,State = \begin{pmatrix} x & y & x' & y' \end{pmatrix}$$

$$Position: \; x_{new} = x_{old} + x't$$

$$Velocity: \; x'_{new} = 0.9x'_{old}$$

3.2.2 Dual Modal

Previously the Unscented Kalman Filter consisted of only a single mode and did not differentiate between a moving and stationary ball. Due to noise and inaccuracies in the filter, the ball's velocity would never be zero even if the ball was stationary. Typically, velocity values for both x and y would fluctuate between ± 30 millimetres per second. This caused serious problems with robot behaviour as its movement would spasm to match the fluctuation and none of the stationary ball based behaviours would be run.

A second filter was added to model a stationary ball by assuming its velocity was always 0, in other words, it simply tracked the ball's position. To decide which mode was the correct one, a weighting w would be calculated based on velocity and its variance. The logic was as follows:

- High *velocity* / Low *variance* = Large w
  - We are fairly certain that the ball is moving.
- Low *velocity* / Low *variance* = Medium w
  - We are fairly certain that the ball is stationary.
- Low *velocity* / High *variance* = Small w
  - We are not sure where the ball is but we can assume it is stationary, especially since if no one knows where it is, the chances are it is stationary and simply has not been found yet.
- High *velocity* / High *variance* = Medium w
  - We are not sure where the ball is, but it is safer to assume it is stationary for reasons mentioned above. It is also possible that the high velocity was due to a false positive or the ball being obstructed and stopped by another robot.

This showed that large weightings should swing favour towards the moving ball model, otherwise the ball should be assumed to be stationary.

Keeping in mind that variance is stored as a 4 by 4 matrix of squared values for the whole state, the relevant subset for velocity is just the 2 by 2 matrix beginning at (2, 2). Velocity is the 2 by 1 vector $x'$and y' also formed from taking the subset of the 4 by 4 state vector at (2, 0). As such, the scalar weight w was calculated as follows:

$$w = velocity^T \times variance^{-1} \times velocity$$

$$= \begin{pmatrix} x' \\ y' \end{pmatrix} \times \begin{pmatrix} var_{2,2} & var_{3,2} \\ var_{2,3} & var_{3,3} \end{pmatrix}^{-1} \times \begin{pmatrix} x' & y' \end{pmatrix}$$

With much testing, it was found that on average, a weighting greater than 3.5×10^{-3} meant the ball was moving.

3.2.3 Odometry

The original Unscented Kalman Filter did not take into account the robot's odometry which caused inaccuracies whenever the robot moved around the ball. For example, suppose a striker robot saw a stationary ball and started moving towards it. If the striker were to look away, say to localise, its robot-relative distance to the ball would never be updated due to the ball having zero velocity. However, the distance should actually be decreasing, and when the robot turns its head back to view the ball, it would lose it instead.

Since rUNSWift represents robot odometry using the values of forward movement, left movement, and turn angle, incorporating it into the ball filter simply involved the addition of a few more calculations during the time update. At each time step, the robot's previous odometry would be subtracted from its current odometry for the change in odometry δo. The ball state s would then be transformed as follows:

$$s.x = s.x - \delta o.forward$$

$$s.y = s.y - \delta o.left$$

$$s = s \times \begin{pmatrix} \cos(\delta o.turn) & -\sin(\delta o.turn) \\ \sin(\delta o.turn) & \cos(\delta o.turn) \end{pmatrix}$$

The variance in odometry would also be added to the ball variance to account for any error in the measure of the robot's movement.

3.2.4 Head Pitch and Yaw

Whilst searching for the ball, the robot would at times perform a rather quick head scan. This would add a lot of blur and noise to the readings, especially in the first few frames of the ball being found. Mode-switching was the most affected as the robot would often think the ball was moving when it was actually stationary.

At every time step, the current joint values of head yaw and head pitch would be subtracted from the previous values. This change in head yaw and head pitch would then be added during the filter's time update phase. These adjustments were done in order to account for any fast head movements when finding the ball.

3.2.5 Team Ball

All the robots broadcast certain data to their team mates, including their perception of the ball. These would then be combined into one team ball, however previous calculations did not properly take variance into account. To fix this biased perception, the team ball was recalculated as follows, where n represents the number of robots, x_i represents the current robot's perception of the ball, and σ_i^2 represents its variance:

$$new\,weighted\,mean = \frac{\sum_{i=1}^{n}(x_i/\sigma_i^2)}{\sum_{i=1}^{n}(1/\sigma_i^2)}$$

$$new\,weighted\,variance = \frac{1}{\sum_{i=1}^{n}(1/\sigma_i^2)}$$

The new calculated team ball was then added to Off-Nao to aid in debugging as well as visualising its location for behaviours.

Chapter 4

Goalie Components

4.1 Behaviour

The tasks a goalie should perform were broken down into several different states based on the ball's position in the field. Figure 4.1 provides a rough overview of how the goalie should react based on ball location in the field, while the following sections explain what all the states involve in further detail.

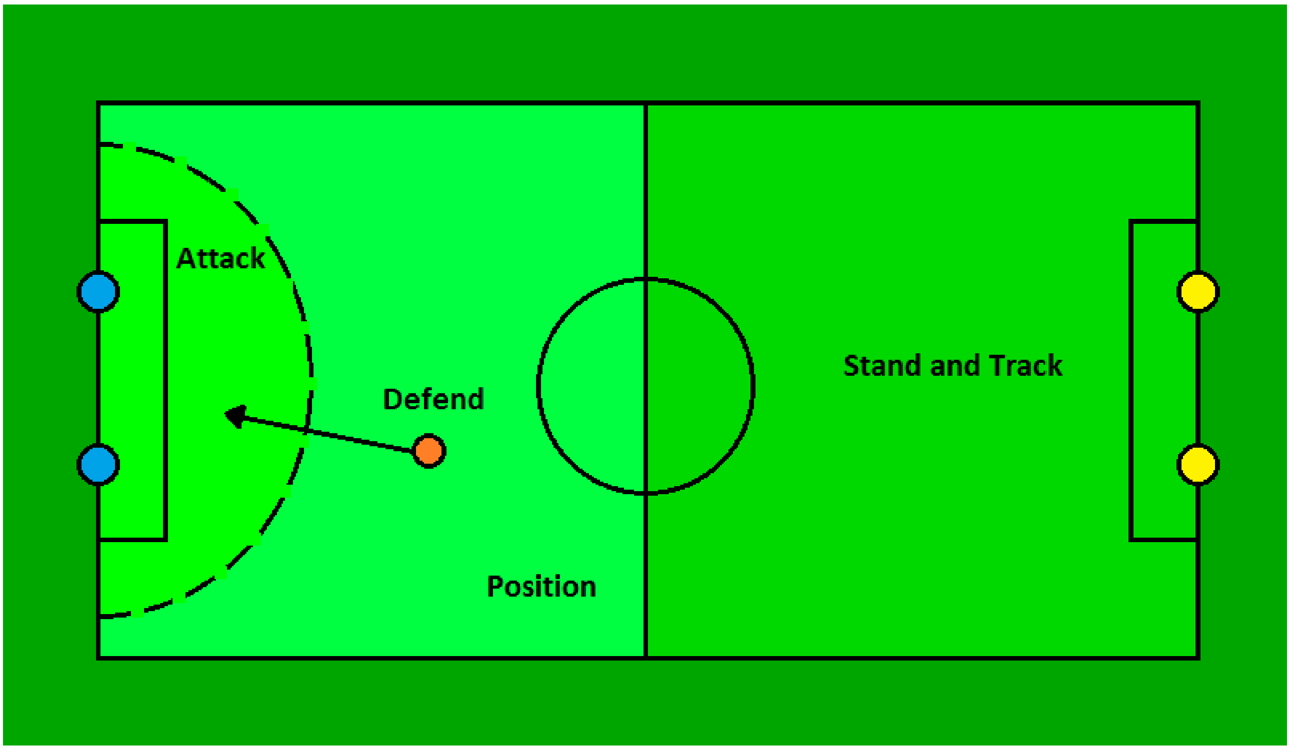

Figure 4.1: Overview of goalie behaviour for different sections of the field

4.1.1 Find and Track Ball

The most basic task the goalie needed to perform was finding the ball, as most of its other tasks are based upon the ball's location. Since finding the ball was a common problem amongst all four robots on the team, the goalie was set to reuse the same base skill. Finding the ball involved performing a series of head scans across the robot's field of view with the bottom then top camera, then rotating if the ball was not found. However, due to the goalie's location typically being in the goal box, rotating towards the back of the field was actually rather counter-productive.

Instead of using the same skill for finding the ball, a sub-classing skill was written to replace the rotations with oscillations. At first, the goalie simply oscillated back and forth at an angle of 60 degrees. Unfortunately, this basic method caused problems if the robot started off-centre. The skill was then updated to make the robot oscillate back and forth depending on its own absolute heading. If the robot was facing a negative heading, then it would rotate left to adjust itself accordingly. Similarly, if the robot was facing a positive heading, it would then rotate right. Although this method was rather dependent on the robot's localisation, it generally fared better than the basic blind oscillation.

If the ball had been found on the far side of the field, then the goalie would simply stand and track the ball. As this was the goalie's default state, it then had the option to transition into any of the remaining states depending on its and the ball's location.

4.1.2 Patrol Goal

As soon as the ball entered the robot's own side of the field, the positioning 'Patrol' state would be activated. This was designed to keep the robot between the ball and the goal at all times, but also far enough forward from the goal line to cover as much goal area as possible. At first, the point of intersection between the two lines was used as the target position for the robot. It was found that the ideal space forward from the goal was 40 centimetres, so the robot's target x position would be -2600 (since the goal is set at -3000). The target y position was then calculated as follows:

$$POI_y = \frac{(goal_y - ball_y)}{(goal_x - ball_x} \times (-2600 - ball_x) + ball_y$$

However, due to noise in the robot's localisation and ball filter, its target position would constantly fluctuate. This meant that the robot would never quite reach its target but instead get stuck walking back and forth. The solution to this was to classify the target positions into three main positions within three main regions based on the ball's global heading, with a 5 degree margin for hysteresis:

- Left Region
  - If global ball heading is > 38 degrees, target $x = -2670$, $y = 520$
- Centre Region

– If global ball heading is within ±33 degrees, target $x = -2600$, $y = 0$

- Right Region

  – If global ball heading is < -38 degrees, target $x = -2670$, $y = -520$

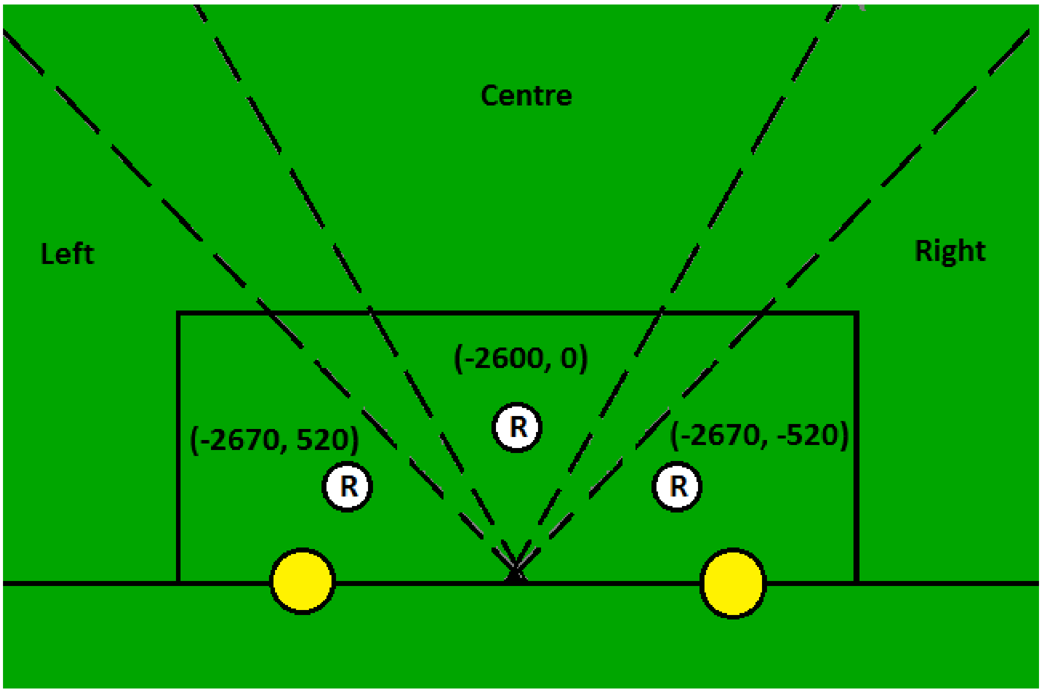

Figure 4.2: Classified regions and the goalie's target position in each

Now, the robot could not simply be set to walk forward and left by the difference in its position's and target's x and y. Instead, a straight path needed to be calculated for the robot to follow, else it would wander into the goal posts. This movement could be expressed in terms of forward x and left y by rotating the difference in x and y by the robot's heading.

$$move_x = diff_x \times \cos(-\theta_{robot}) - diff_y \times \sin(-\theta_{robot})$$

$$move_y = diff_x \times \sin(-\theta_{robot}) + diff_y \times \cos(-\theta_{robot})$$

Once the robot walked to within 10 centimetres of its target position, its current region would be updated to the new region. The robot would only move again to readjust itself if the ball had rolled into a different region.

The goalie would also have a target heading set depending on whether or not it was facing the ball. In this situation, robot-relative ball heading would be used instead. Since the goalie had a reasonable view range and constantly turning to face the ball took up too much movement time, it would only readjust itself if the ball moved past its range of 45 degrees.

4.1.3 Localise

Deciding when would be the optimal time to localise was no easy task, and eventually it was decided that a localisation scan would only be triggered under the following circumstances:

- Goalie is not localised[1] AND it has seen the ball for awhile AND is sure that the ball is not moving, OR
- Goalie is not localised AND has not seen the ball for awhile AND 10 seconds had passed since the last scan, OR
- Goalie is not localised AND it has not yet reached its designated patrol region AND the ball is not moving, OR
- Goalie is not localised AND it has reached its designated patrol region AND it has seen the ball for awhile AND is sure that the ball is not moving

Another problem with goalie localisation was that it was rather impossible to localise in the standard fashion of looking at two goal posts. Since the goalie usually stood in between the two posts on its side of the field, it could not see them both in one frame. Although it could see the far goal posts in a single frame, the lengthy distance caused many inaccuracies which usually made its localisation worse. Thus, a special localisation scan was written for the goalie which took advantage of rUNSWift's new vision of field features.[2]

This scan was aimed at finding the corners of the goal box in front of the goalie and a goal post to its side as quickly as possible. If the goalie happened to be facing slightly off-centre and did not find a goal post at its current side, it would then switch scan directions to check the opposite side. Now if the goalie was standing in one of the side patrol regions, this information would be passed through so the localise scan would know which direction to check for the closest goal post. Additionally, if the robot were facing off-centre, the scan would also aim to find T-junctions next to the goal posts, and if possible, the far corner representing the boundary of the field. To ensure as many accurate frames of these features as possible, the scan would be slowed down if it detected any corners, T-junctions, or goal posts of the robot's own colour.

4.1.4 Defend Goal

The defending state would only come into play once the ball rolled within 2 metres of the goalie and if the goalie were in position and facing the ball. The goalie would then drop down into its crouching goalie sit stance and constantly watch the ball in preparation for it coming towards the goals.

[1] A helper function available to all robot behaviours, it simply calculated how well localised the robot was based on its variance in position and heading.

At first, the absolute co-ordinates of the ball were used to calculate if it was worth diving to block the goal. This was because the goal posts would always be available in absolute co-ordinates and diving would only be needed if the ball was actually projected to roll between the two posts.

1. Calculate the absolute position of the ball using the robot's own location and the robot-relative ball position
$$x_{abs} = x_{robot} + \cos(\theta_{robot}) \times x_{rr} - \sin(\theta_{robot}) \times y_{rr}$$
$$y_{abs} = y_{robot} + \sin(\theta_{robot}) \times x_{rr} + \cos(\theta_{robot}) \times y_{rr}$$
2. Use the robot's robot-relative ball velocity to project the ball's next location after time t
$$x_{new} = x_{abs} + \cos(\theta_{robot}) \times x'_{rr} \times t - \sin(\theta_{robot}) \times y'_{rr} \times t$$
$$y_{new} = y_{abs} + \sin(\theta_{robot}) \times x'_{rr} \times t + \cos(\theta_{robot}) \times y'_{rr} \times t$$
3. Sanity check that the ball is actually coming towards the goals
$$x_{new} < x_{abs}$$
4. Assuming the goals can be represented with the equation $x = -3000$, calculate the point of intersection along the y-axis
$$A = y_{new} - y_{abs}$$
$$B = x_{abs} - x_{new}$$
$$C = A \times x_{abs} + B \times y_{abs}$$
$$POI_y = \frac{(C + 3000A)}{B}$$
5. If this point lies within the goal posts, assuming they are at $y \pm 1000$ to account for error margins, then the goalie should block the ball
$$abs(POI_y) < 1000$$

However, by introducing the goalie's own position as a factor, the reliability of the overall projection was reduced due to inaccuracies in localisation. This was particularly an issue since there was no opportune time for the goalie to relocalise, as it had to keep watch on the ball when it was so close. As such, the choice was instead made on robot-relative ball information alone, with the robot only diving if the ball were to roll within its range.

1. Extrapolate the next robot-relative location of the ball at time t
$$x_{new} = x_{rr} + x'_{rr} \times t$$

$$y_{new} = y_{rr} + y'_{rr} \times t$$

2. Sanity check that the ball is actually coming towards the goals

$$x_{new} < x_{rr}$$

3. Assuming that the line the robot is on is equivalent to the equation $x = 0$, calculate the point of intersection along the robot's y-axis

$$m = \frac{(x_{new} - x_{rr})}{(y_{new} - y_{rr})}$$

$$b = x_{rr} - m \times y_{rr}$$

$$POI_y = \frac{-b}{m}$$

4. Assuming the dive stretches the robot out (including its arms) to a length of 800 millimetres, with a little extra included to account for error margins, if the point lies within this range then the goalie should block the ball

$$abs(POI_y) < 800$$

5. Assuming the range of the goalie's squat for blocking its centre is 100 millimetres on either side, choose the appropriate action accordingly

$$POI_y > 100 = Dive\,left$$

$$POI_y < -100 = Dive\,right$$

$$abs(POI_y) < 100 = Centre\,squat$$

To ensure that the goalie would not waste its time diving for a false positive, the same action had to be calculated for at least 7 frames before the goalie would actually perform it. Since the rules stated the goalie could not be in such extended positions for longer than 5 seconds at a time, it was also important to stop the goalie from reacting too early. As such, it would only start diving once the ball was within 1.4 metres, or squatting in the centre once the ball was within 1 metre. The minimum distance for the centre block was much less since it was a much faster transition than diving.

4.1.5 Attack Ball

The attacking state made use of the stationary ball filter in that if the ball wasn't moving, then there was no need to block the goal. Instead, if the ball was close, the goalie could show some aggression and kick it away. Using the goalie to attack the ball had an added benefit in that the goalie would typically already be behind the ball and facing the right direction for a good line up. It was also rather necessary since if the ball stopped within the goal box, only the goalie could clear it as none of the other team members were permitted to enter the goal box. As such, we adopted a rather aggressive strategy in that the robot would attack if:

- The ball was within a robot-relative distance of 800 millimetres, OR
- The ball was in an absolute position that satisfied $x < -2000$ millimetres AND $abs(y) < 1500$ millimetres as a buffer zone around the goal box

The code for attacking the ball was simply reused from the striker behaviour, but with two slight modifications. Firstly, its localisation thresholds for kicking the ball were lowered, as aiming perfectly into the opposite goals would be a lower priority compared to clearing the ball from its own goal. Secondly, a higher preference was given to forward kicks as this line up would conveniently block the opponent robots from scoring, whereas lining up for a side kick would leave the area wide open.

Now, a more aggressive goalie meant more obvious clashes with fellow team members when going for the ball. Fortunately this aided in debugging the role switching problem even further. Instead of having its own attack state as part of the goalie skill, the attacking conditions were pulled out into a higher level team skill.[5] This team skill then acted as a role switcher to run the striker skill instead of the goalie skill. The goalie-specific localisation and line up modifications were subsequently implemented by passing a special goalie flag through to the striker.

4.1.6 Reposition

If the goalie ever left its goal box, say after attacking the ball or getting penalised, it would need to find its way back to the centre of the goals. Since this problem was identical to positioning in the ready state at the start of the game, the skill was simply reused for the goalie's repositioning.

4.2 Motion

4.2.1 Blocking The Centre

Making the goalie dive whenever the ball was coming towards it was quite a risky venture for several reasons:

- Every dive and fall would damage the robot's hardware
- Localisation would be reset each time the robot fell, and relocalising from within the goal box would be no easy task

- Diving was a relatively long transition, especially with the time taken to perform the get up routines

Thus, a new centre squat was developed to block the ball if it was rolling towards the robot's centre region. This meant that the robot would only dive left or right when absolutely necessary.

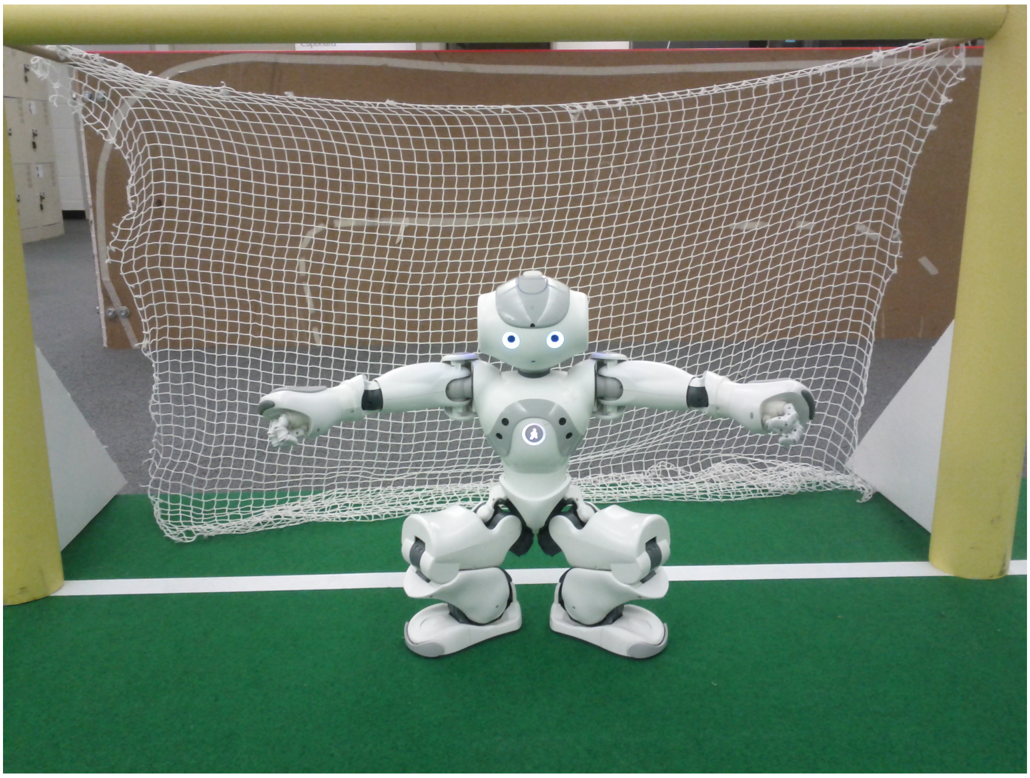

Figure 4.3: Squat for blocking the goalie's centre region

However, like many centre blocks seen in past goalies from other teams, this particular squat was susceptible to the ball holding penalty when returning to the crouching goalie sit position. Since the goalie would simply return its feet to the centre, the ball would get stuck between both legs. This was particularly likely to occur if the ball happened to stop right near the middle of the robot's ankles.

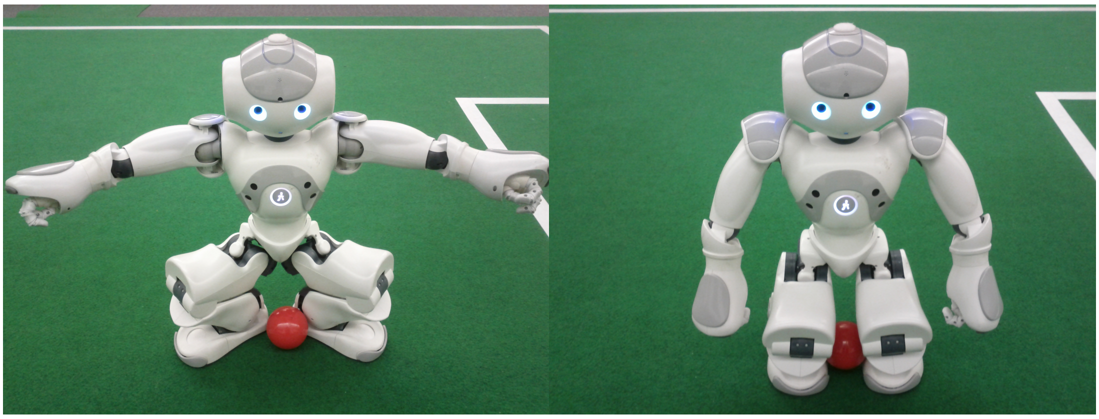

Figure 4.4: Ball holding as the goalie returns to its default sitting position

To prevent the goalie from getting penalised for the occasional ball holding, a new pos file[2] was added. It was set to run only when transitioning from the centre squat into the goalie sit. By first shifting one foot forward to move the ball away from the middle of its legs and then only returning its feet to the centre, the goalie should not hold the ball for longer than 5 seconds. Instead, it could sometimes even project the ball outwards from its stance if it managed to close its feet onto the back of the ball.

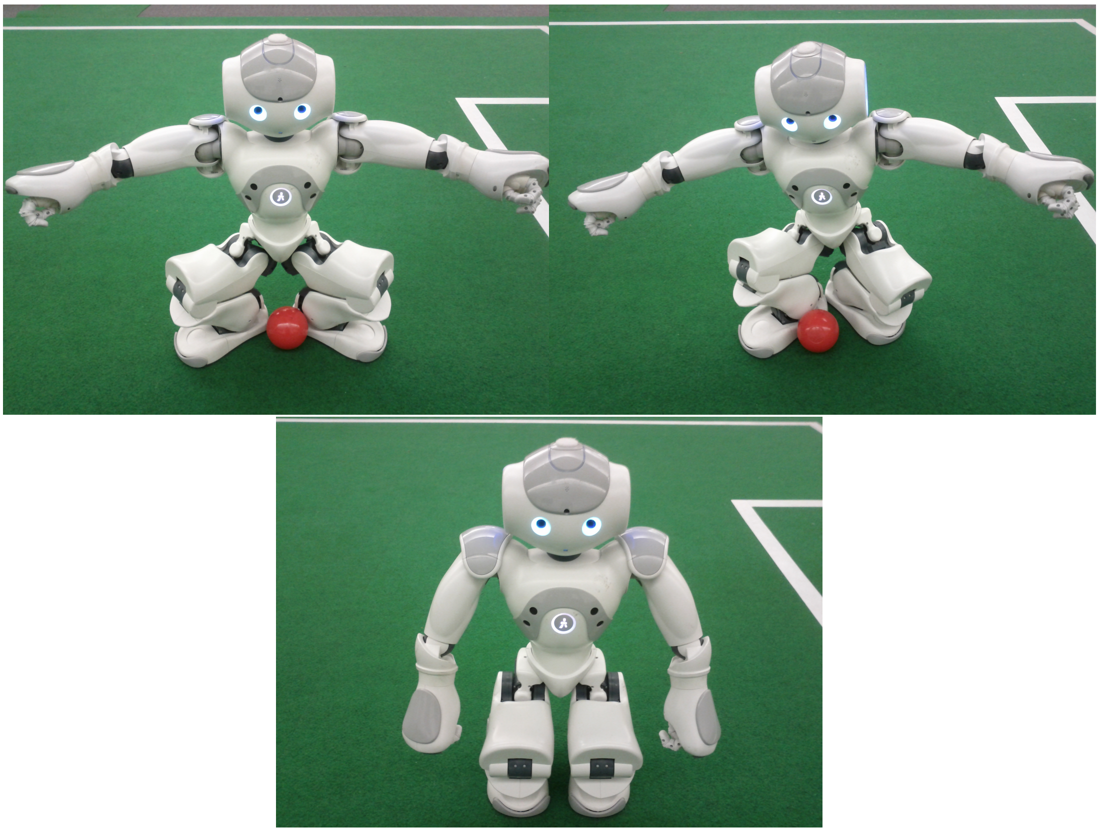

Figure 4.5: New centre squat transition to avoid ball holding

[2]A file format for specifying joint angles and joint stiffness to make the robot perform certain actions.

4.2.2 Blocking The Sides

Although diving to block the ball was risky, it was at times necessary, and thus it needed to be improved. As mentioned in the previous section, there were three main issues with the dive. The damaging of hardware on contact with the field had already been minimised as much as possible with the lack of stiffness in the joints, while the localisation issue was not really addressable within the scope or time frame of this project. As such, the only point to improve on was the dive's transition time.

First, the time of the falling section of the dive was shrunk down to two-thirds of the original time. This allowed the goalie to reach the ground faster, which meant it could spend more time watching the ball before making a diving decision. The angle of the arm used to break its fall then had to be adjusted slightly outwards to account for the faster speed.

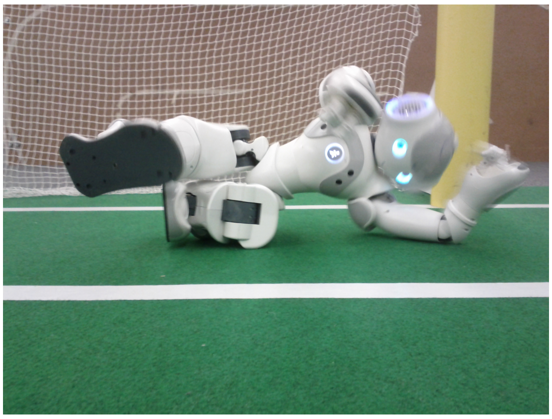

Figure 4.6: Initial landing of the goalie dive

4.2.3 Recovery

The other side to improving the overall time of the dive was improving the time of the get up routines. Since the dive would land the goalie on its side, it needed to quickly turn its body flat else the get up routine would not be triggered. It was found that the front get up position was easier to reach by straightening the goalie's arms and kicking its top leg over the other.

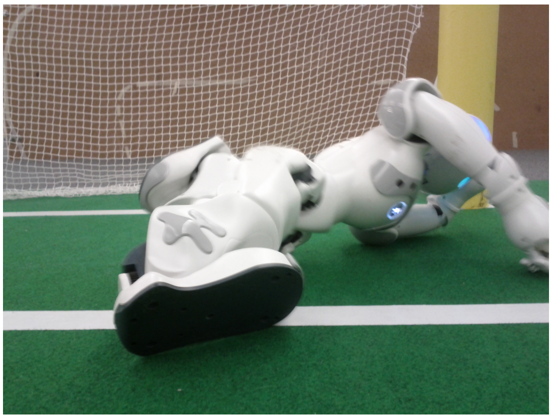

Figure 4.7: Goalie dive rolling forward to get up

Then there was the matter of speeding up the actual get up routine so the goalie could get back to defending the goal as soon as possible and clearing the ball it had just blocked. Although the main points of the get up routines could not be sped up much due to delicate balancing, prepping movements such as straightening the arms and ending movements such as straightening the body to stand up had their times reduced to almost half of the original. The front get up routine also needed to be stabilised further, particularly when bringing its feet together and standing up. For better control over the whole process, the set of joint angles which specified this movement was split into three separate sets. They were then likened to the stabler way the goalie entered and exited its centre squat since the stances were rather similar.

4.2.4 Tracking From Afar

If the ball was down the far end of the field, the goalie would simply track the ball and hopefully help transfer team ball data to its team members. However, due to the lengthy distance, its readings would not always be very accurate. The ball would also often be obstructed by other robots. Thus to improve the goalie's vision of the far end of the field, a tall standing stance was developed with the robot's knees straightened as much as possible.

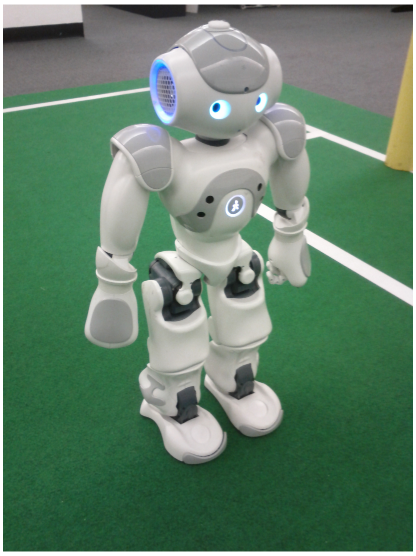

Figure 4.8: Tall goalie stand for tracking far balls

4.2.5 Reducing Joint Stress

One of the main goals of the crouching goalie sit stance was to reduce stress on the knee joints by letting them bend to their maximum potential and rest on the ankles, as opposed to the normal stand. However, after long periods of testing, it was found that this stance would still cause overheating problems. This was most likely because it was a higher strain on the motors to keep the knees bent at an even lower level. Fortunately, the new stiffness feature in the pos files made it possible to remove all stiffness from the knee motors once the goalie completed the transition into the sitting stance.

4.3 Penalty Behaviour

During the latter stage of the Robocup SPL competition, if games ended with tied scores, then both teams would be forced to a tie-breaker with a penalty shoot-out. This would involve one goalie robot from the defending team being placed in the middle of the blue goals facing the centre of the field, and one striker robot from the attacking team being placed at the centre of the field facing the blue goals. The ball would then be placed on the penalty spot between them and a time limit of 1 minute given for a goal to be scored. The same rules applied to the penalty kick, which could occur if a robot caused any damage to its surroundings. However, the goal would be taken against the opponent goals, not necessarily the blue ones.

It must also be noted that special rules applied during the penalty kick. Namely, the goalie would not be allowed to touch the ball if it were completely outside the goal box, else the goal would be awarded to the attacking team. On the other hand, the striker would not be allowed to touch the ball if it were completely inside the goal box, else the penalty shot would be deemed unsuccessful.

The goalie would not be penalised for inactivity provided its stiffness was on, however usual penalties such as ball holding and pushing would still apply.

All these rules are particularly important as the strategy devised for the penalty goalie took advantage of each. Since the goalie would be placed within the goals from the start, localising accurately would be rather difficult. It would instead be safer to remain stationary in its placed location. Considering this was right in the centre of the goals, it was actually quite an optimal position anyway.

From this point, a diving behaviour similar to that of the normal goalie would be run to block the ball. It too would dive depending on which direction the ball was travelling in, but with a slight twist. After diving, the get up routine could sometimes knock the ball either back out of the goal box, or potentially even score an own goal. If the ball rolled out of the goal box, the attacking robot would have another opportunity to kick the ball again while the goalie was out of position. The same situation applied to the centre squat, as it too could sometimes shoot the ball back outwards. As such, it was strategically decided that after performing its blocking action, the goalie should not move any further. Even if the goalie was then penalised for not getting up within 5 seconds or staying in a wider stance for longer than 5 seconds, the blocked ball should still remain in the goal box, thus rendering the attacking robot useless. Similarly if the goalie was penalised for ball holding, which could occur in the centre squat or if the dive fell on top of the ball, the ball would still be inside the goal box and unreachable to the attacking robot.

Finally, a special modification for the penalty goalie had to be made to the common ball tracking skill used during normal games. If the ball was close, then it would be tracked using the robot's bottom camera, whereas if the ball was far, then it would be tracked using the top camera. This meant that the ball could sometimes be lost during the switching of cameras, especially if it were travelling fast. Now since the ball would be placed on the close penalty spot and would typically travel towards the goals rather than to the far end of the field, there was no need for the top camera. A new sub-class was thus made that disabled camera switching and searched and tracked the ball using only the bottom camera. The searching scan also had to be adjusted with a higher pitch to allow the bottom camera to see as far as the penalty spot.

Chapter 5

Results

5.1 Ball Tracking

Overall, the ball filter was generally successful in that our robots did not have a problem following the ball during games. It was noticed once during the competition that the robot would walk slightly past the ball while lining up for a kick. However, this could also have been a result of the way in which the striker behaviour used the filter information to calculate the line up point. Since competition data was not the most accurate way to test the ball filter alone, other arrangements had to be made.

To test the filter out more specifically, a special experiment was set up involving two robots, say R_1 and R_2. They would be placed such that their field of views overlapped and set to find and track the ball. R_2's vision would then be turned off so it could only find and track a ball using the shared filter information. A ball was placed between both robots with R_1 allowed to find the ball as normal, and then rolled around within R_2's field of view. It was observed that R_2 moved its head to follow the ball despite not being able to see the ball itself.

Now, it must be noted that the two robots in the above experiment were both well localised, with R_2's position and heading being hardcoded due to its lack of vision. This had the effect of reducing the error introduced by robot localisation, however in realistic games, this would not be the case. As such, a second experiment was set up without any particular localisation tweaks. This time the two robots were placed on the field such that they were facing two different directions. A ball was placed such that it was in both robot's blind spots, and then the robots were allowed to run their find and track ball skills. It was observed that once one robot had found the ball, the other would also turn to face it. However, as time went on, the robots began to oscillate away from the ball more and more.

The dual filters for stationary and moving ball modes were tested through showing that the goalie could decide which motion it should take to block a ball, whether it be left, right or centre, by rolling a ball towards it. It could also decide to attack the ball if it were stationary. However, there were some fluctuations in the modes whenever the goalie was changing its stance height. More specific results on the blocks are presented in section 5.2, with the performance of the goalie's overall behaviour of defending and attacking being described in section 5.3.

5.2 Goalie Motion

To test the new centre squat, a ball was placed at intervals of 3 centimetres along the range of the stance. The transition back to the crouching goalie sit was then performed 3 times for each position. In terms of not ball holding, the new centre squat was a great success. Of the 33 total tests for the 11 positions, not a single ball was held stuck between the goalie's legs. In fact, due to the way the goalie's feet closed onto the ball, 3 of the more central positions and 2 of the left positions actually caused the ball to shoot forwards out of the goalie's stance.

In terms of the centre squat's range, it was measured to block a total of 34 centimetres. Considering the width of the goals is 140 centimetres, the stance covered 24.2% of the whole goal line, which could be increased depending on how far forward the goalie squatted. This was quite an improvement over simply sitting or standing in the goals which only covered 20 centimetres, in other words, 14.2% of the goals. Unfortunately, the goalie centre squat did not get tested during the competition as most of the balls that came towards the goals were not actually directed at the goalie.

The penalty goalie's main purpose was to dive, as such, it was a convenient way to test how successful the goalie dive was. Its range was measured to reach up to 63 centimetres. Now, since the penalty goalie was never actually used during competition and the dive itself only occurred once during the games, an experiment to gather data was run versus the penalty striker. 40 penalty shots were made by the striker, which aimed to kick the ball between the goalie and the goal posts.[5] The results are summarised in table 5.1 where total kicks represents the number of times the striker kicked in that particular direction, total dives represents the number of times the goalie dived in response, total squats represents the number of times the goalie performed the centre squat in response, and goals saved numbers how many of these were actually successful.

Kick Direction	Total Kicks	Total Dives	Total Squats	Goals Saved
Left	18	18	0	18
Centre	2	1	1	2
Right	20	20	0	19

Table 5.1: Penalty goalie's responses to a variety of ball directions and their success

5.3 Behaviour at Istanbul 2011

Table 5.2 counts the number of times the goalie was correctly positioned when the ball was on its own side of the field. The goalie was considered 'out of position' if it was at least facing the ball, but too far away from the ideal defending location. 'Incorrect heading' would be if the goalie was in the prime location to defend the goal, but was not actually facing the ball.

Game	Correct	Out of Position	Incorrect Heading	Incorrect Position & Heading
L3M	1	1	0	0
WrightEagleUnleashed	9	3	0	0
Cerberus	0	0	0	0
Team Nanyang	7	0	0	1
NTU Robot PAL	5	1	0	0
HTWK	5	1	1	1

Table 5.2: Goalie positioning during its Patrol state

Although the ball rolled to the goalie's own side of the field for it to be positioning 36 times in total, not all of them actually came close to the goals. Table 5.3 describes the goalie's reactions to the 12 balls that actually came close to it, as well as the eventual outcomes. 'Ball Motion' refers to the ball's action when close to the goalie, for example, 'Left' signifies the ball was rolling towards the goalie's left, 'Stationary' indicates the ball stopped nearby the goalie instead, while 'Left→Stationary' means the ball went from rolling left of the goalie to being stopped nearby. 'Goalie Reaction' describes how the goalie moved in response, and a 'Scored?' of 'Yes' means the opposing team successfully scored a goal against the goalie.

Opposing Team	Ball Motion	Goalie Reaction	Scored?
L3M	Stationary	Attacked ball	No
WrightEagleUnleashed	Centre	None, attacked after rebounding	No
WrightEagleUnleashed	Stationary	Attacked ball	No
Team Nanyang	Centre	None, attacked after rebounding	No
NTU Robot PAL	Right	Fluctuated between sitting & standing	Yes
NTU Robot PAL	Stationary	Attacked ball	No
NTU Robot PAL	Right	Dived, cleared the ball on get up	No
NTU Robot PAL	Left→Stationary	None, attacked after stopping	No
HTWK	Stationary	Attacked ball	No
HTWK	Stationary	Attacked ball	No
HTWK	Right	Sat, did not adjust as ball was dribbled right	Yes
HTWK	Stationary→Right	Attacked ball	Yes

Table 5.3: Goalie's reactions to nearby balls during the competition and the resulting effects

Of the 2 balls moving towards the goalie's centre, it would have been safer for the goalie to perform the centre squat, however the ball rebounded off it both times. For 8 out of the 9 times the goalie attacked the ball, it had done so appropriately, with 2 of the attacking kicks landing the ball in the opponent's goal posts. However for the last case, the ball was dribbled right after it had started attacking. When the goalie went to readjust its line up, it prepared for a side kick, and as a result it did not obstruct the incoming kick from the opponent robot. Of the one dive the goalie performed, it successfully blocked the ball. However, there was one occasion where the goalie should have dived, but instead could not decide what to do. Another occasion was where the goalie did not dive due to the ball being out of its range, though it should have first readjusted its position and then dived.

In total, 6 goals were scored against rUNSWift during the whole competition. The 3 goals not addressed above occurred during the game versus HTWK. For one of the goals, the goalie was not actually in position within the goal box as it was stuck sonar avoiding the goal posts. As a result, it did not even see the ball. As for the remaining two goals, the goalie was not actually present on the field at the time.

Chapter 6

Discussion

6.1 Ball Tracking

As shown by the ball tracking experiments, the filter itself was a success in that a robot could track a ball purely based off ball filter information. However, introducing localisation and time delays when calculating an absolute ball position proved rather unreliable and was not suitable for real game conditions. Tracking a team ball proved especially difficult as while the robot focused on the ball, the robot could not localise. If the robot then scanned elsewhere to localise, it would not be able to see the ball. As such, the variance introduced from both uncertainties were much too high. This could also have been a result of not modelling the ball and its variance accurately enough. In fact, it was sometimes found that the team ball calculations provided negative variances. It was not fully discovered what had caused this, but it was mainly attributed to instabilities in the Unscented Kalman Filter. In particular, the Cholesky Decomposition used to calculate the square root of a matrix was an experimental algorithm of the Eigen library and was thus found to be unreliable at times.

Robots that walked just past the ball could have done so due to the way the filtered ball information was used. Velocity data was applied in goalie behaviours, whereas striker behaviours used the ball's position data alone. Thus if the ball was moving, it did not take into account the ball's upcoming position, meaning the ball would often roll past instead.

In terms of a dual modal filter, the weighting method was generally successful as the robot could tell if the ball was moving or stationary when standing or walking around as normal. However, sudden changes such as the switching of cameras and the changing of stance height would cause fluctuations between the two modes. This was mainly associated with the fact that the robot's kinematics was only calibrated at the normal standing height, which did not bode well for a goalie with multiple stances of varying heights.

6.2 Motion

The centre squat stance was extremely successful in that it was not at all susceptible to ball holding. It also covered 10% more of the goal area than simply standing still. Additionally, shooting out the ball while returning to the goalie sit stance had the benefit of clearing the ball from the goal area all in the one same motion. However, the centre squat would definitely benefit from an even wider range as it would further reduce the need to dive.

The reason for diving despite a centre-aimed ball during the penalty goalie test was due to the behaviour itself rather than the motion. The range of the centre squat had been set to 20 centimetres, as opposed to 34 centimetres, to allow for inaccuracies in the ball filter. The area covered by the dive overlapped the end sections anyway, as can be seen by the successful block in the penalty goalie test.

As for the dive itself, it too worked quite well in that it covered 63 centimetres, in other words, 45% of each side of the goals. It subsequently proved itself capable of saving a vast majority of penalty shots. However, this last 5% is what allowed the penalty striker to score a goal on its kick into the rightmost edge of the goals. Though it was not an issue during the actual competition, a more optimal arrangement of the joints could have been found to protect the extra 7 centimetres and ensure a complete blockade. It was also noted that the goalie dived at a slight angle and thus would not always cover the full 63 centimetres depending on which way it was facing. Unfortunately, there was no systematic robot model to work with, and there was not enough time for extensive experimentation with just joint values.

6.3 Overall Behaviour

Goalie positioning performed reasonably well with 27 perfect defending stances out of a total of 36 situations, with a majority being in the side regions. 8 were mostly due to localisation problems from being in the goals, in particular, the one incorrect position and heading during the game with Team Nanyang was a result of manual positioning right between the goals. The remaining one situation where the goalie was both out of position and facing the incorrect direction was unfortunately due to a behavioural bug where it would not readjust itself into a prime defending location once the ball got too close.

Overall, this demonstrated that the goalie's localise state was quite effective at remaining localised within the goals, especially in the side regions. Presumably, this was due to the global update of a T-intersection and a goal post in a single frame. However, relocalising after going to kick the ball, being penalised or diving remained a problem. Since the localisation filter could only switch modes based on global updates of either two goal posts or a goal post and a T-intersection, starting within the goals where only one post updates and corner updates were available was disastrous. There was also a slight trade-off when the goalie had to track the ball as opposed to looking for landmarks to localise off.

The opposite trade off was also applicable to the goalie, in that it would at times be localising instead of tracking the ball during critical moments. Errors in heading when positioning within the goals and losing track of close balls can be attributed to this as it was difficult to decide on the opportune time to localise. The main problem with the current method of localising was that the goalie assumed it was 'safe' to localise if the ball was stationary, as it was not going to move anywhere, and would rescan based off time intervals. However, a stationary ball could also imply that an opponent robot would have had the time to reach it and line up for a shot, and simple blind conditions such as localising every few seconds did not help. More sophisticated methods, such as detecting if there were any opponent robots nearby, would have vastly improved the goalie's performance in such crucial moments.

Losing track of close balls can also be associated with the way the ball tracking behaviours would switch cameras depending on the distance of the ball. Changing heights through the goalie's different stances caused the ball to fluctuate between seeming close and then seeming further than it actually was. The two combined not only caused much flickering between cameras, but also fluctuations between the two ball modes of stationary and moving. This problem was epitomised in the game versus NTU where they managed to score a goal because the goalie was too busy being indecisive in its stance. Camera switching did not take the ball's velocity into account and was too slow, and as mentioned in section 6.1, the lack of kinematics calibration for the goalie's other stances was also an issue. A time delay was introduced to make the goalie wait a few seconds and stabilise before changing its stance again, however this was only done in later stages of the competition. However, in some instances the whole system worked well, as shown by the successful dive moments later.

Deciding to attack the ball when it was stationary was actually quite successful as it cleared the goal 8 out of 9 times, almost scoring a goal twice. The only hiccup was during the HTWK game where the ball was dribbled past the goalie's right. Although it managed to catch up to the ball, it lined up for a side kick and left the front open for HTWK to score. Preference to the forward kick was obviously not set high enough. Instead, it would have been beneficial if side kicks were disabled while the goalie was between the goals, but then given preference if the goalie had to clear the ball from the side of the goals. However, this would also depend on how well localised the goalie was.

Unfortunately, our robots were plagued with hardware issues such as foot sensor, sonar sensor, and chest board errors, as well as networking issues such as dropping off the wireless. During the game with HTWK, the goalie suffered from a networking issue and thus thought it was the only robot left on its team. This caused it to turn into a striker robot as part of the team skill strategy, leave its goal box, and clash with the team members that were still on the network. These kinds of problems resulted in HTWK scoring two goals while the goalie was not even present on the field.

Overall the new ball model and resulting behaviours worked, but not reliably, and could have benefited from more thorough testing. The level of sophistication was still not at that of reigning champions like BHuman, though they did fulfill the original aim of improving rUNSWift's ball model and goalie behaviours.

Chapter 7

Future Work

7.1 Filter Improvements

7.1.1 Multi-Modal Distributed Ball Filter

There is a lot of potential for future development on the ball filter. Namely, making it a multi-modal distributed filter should provide more accurate readings, enabling ball tracking to improved. Modelling the ball in certain situations, such as bouncing off a robot like BHuman do[9], would be even more beneficial. This would help filter velocity in particular as rebounding balls can cause vast changes in velocity that are not accounted for in the current ball filter. Accurately measuring the friction of the field would further increase the accuracy of the ball velocity. Implementing an Extended Kalman Filter as opposed to the Unscented Kalman Filter should also be considered as a future option, at least until the instabilities in the Unscented Kalman Filter can be ironed out.

7.1.2 Modelling Uncertainty

A vast majority of the variances in observations were simply estimates and thus not accurate models of the true state. An approach to improving the representation of observation variance could be to rotate the covariance matrix by the pitch and yaw angle as opposed to just adding the difference. Another improvement would be to adopt the Darmstadt Dribblers' approach of using both ball detection methods of size and distance to estimate the observation variance of the ball.[4] Finally, the filter would also benefit from the development of a more accurate representation of the robot's odometry, for example, making use of the accelerometers and visual odometry.

7.2 Goalie Motions

7.2.1 Improvements

If the range of the centre squat could be made wider, less diving would be needed, and as a result, less damage would be more caused to the hardware. It would also provide for a more stable localisation as it would not need to be reset. Team Nanyang's goalie performed an example of a wide centre stance, as shown in Figure 7.1. Luckily for rUNSWift, the Nanyang goalie was penalised for ball holding several times. However, if it could be combined with the non-ball holding transition developed in this project and sped up, it would be quite formidable.

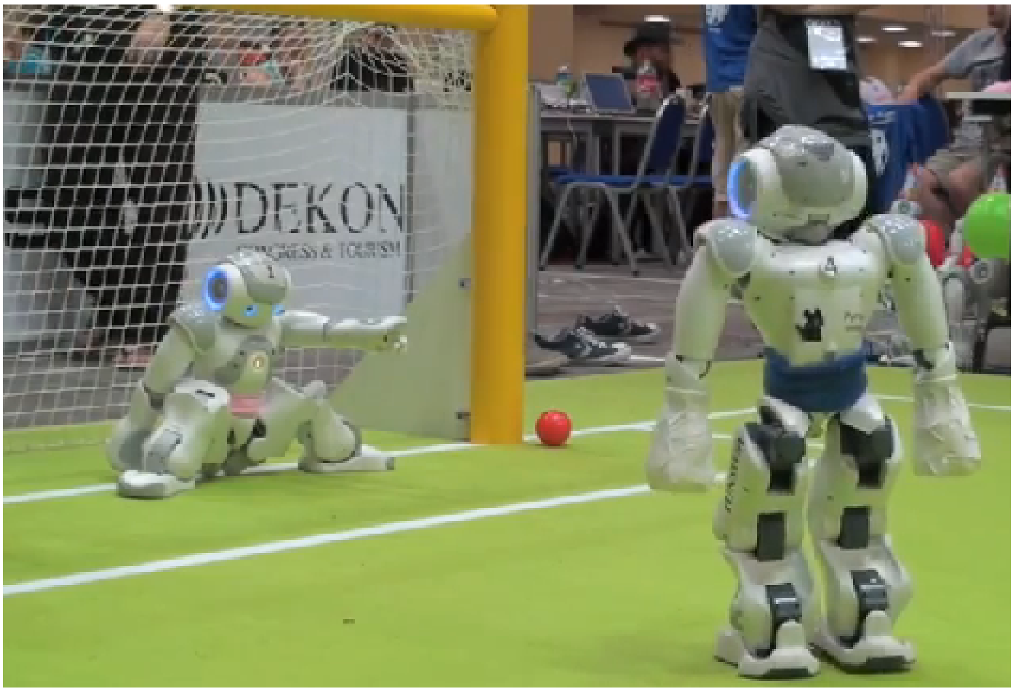

Figure 7.1: Team Nanyang's goalie kicking the ball away while performing its centre squat

Many teams also had a further reaching dive compared to rUNSWift's goalie, covering that missing 5% and more. If a proper robot model could be created such that an optimal dive could be machine learned, this would reduce much of the time and error produced in manually specifying the joint angles.

7.2.2 Future Actions

There is no reason to simply stop at just the development of a dive motion or a squat motion, for the goalie is the only robot that can touch the ball with its arms. Much potential is available for goalies to use their arms more creatively to block and move balls, for example, the throw-in developed by rUNSWift in 2010 could be integrated here. New high kicks are also another avenue to explore in order to kick the ball into the air, and potentially over other robots. Noxious Kouretes demonstrated such an example during the Open Challenge in 2011, though the whole process took a lengthy amount of time and was not always reliable. If this could be improved, it would help the goalie to easily clear the goal of any nearby balls. Although, this would also require the development of a motion to block incoming high balls.

7.3 Integrating Velocity

Although the ball's velocity was made available, only the goalie actually made use of it. It would be ideal if it was incorporated into other behaviours such as tracking a ball, finding a ball, camera switching, and walking towards a ball. In fact, this could be expanded into kicking moving balls, much like Noxious Kouretes during the Open Challenge in 2011.

7.4 Field Line Localisation

Since localising from within the goals was such a problem, creating a field line sensor model would be extremely beneficial to the goalie. Although field lines were detected in vision, there was no probability distribution as to where the robot could be, and the data was not used in the localisation filter.

7.5 Incorporating Robot Detection

A major issue for the goalie was deciding when to localise as opposed to staying focused on the ball. The improvement and integration of robot detection into the goalie behaviour would be quite useful in this situation. Instead of assuming it was safe to localise while the ball was stationary, the goalie could first check if there were any opponent robots near it.

Chapter 8

Conclusion

The aim of this project was to create a more sophisticated ball model for rUNSWift in the Robocup Standard Platform League domain. Along with the development of new goalie motions, this ball model would subsequently be used to improve rUNSWift's goalie behaviours. The results have shown that this was ultimately possible through the use of an Unscented Kalman Filter to track the ball more effectively.

Adding ball velocity to the model was particularly beneficial as it allowed robots to reliably calculate which direction the ball was travelling in and estimate its next location. By expanding on this with the two ball modes of stationary and moving, the goalie could decide which action was more appropriate, whether it should dive to defend a moving ball, or walk forward to kick a stationary ball away.

The goalie itself ended being a major improvement over rUNSWift's 2010 goalie. The new positioning method allowed it to be in prime defending location many a time. With the addition of more goalie motions, such as the dive, the goalie saved a goal during the competition to keep the score at a draw as opposed to a loss. Its ability to switch to attacking a stationary ball also cleared the ball away from some close calls and nearly even scored a goal itself.

However, there is still much room for improvement as described in the future work section, particularly in increasing the accuracy of the ball model and improving goalie localisation. Although the new ball model and goalie behaviours were not always reliable, they still performed reasonbly well overall and remained beneficial to rUNSWift's performance.

Bibliography

[1] David Gregory Claridge. Multi-Hypothesis Localisation for the Nao Humanoid Robot in RoboCup SPL. Honours thesis, The University of New South Wales, 2011.

[2] Sean Harris. Efficient Feature Detection Using RANSAC. Honours thesis, The University of New South Wales, 2011.

[3] S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte. A new approach for filtering nonlinear systems. In *Proceedings of the American Control Conference*, volume 3, pages 1628 – 1632, 1995.

[4] S. Kohlbrecher and O. von Stryk. Modeling Observation Uncertainty for Soccer Playing Humanoid Robots. In *Proc. 5th Workshop on Humanoid Soccer Robots at the 2010 IEEE-RAS Int. Conf. on Humanoid Robots*, Nashville, Dec. 6 - Dec. 8 2010.

[5] Jimmy Kurniawan. Multi-Modal Machine-Learned Robot Detection for RoboCup SPL. Honours thesis, The University of New South Wales, 2011.

[6] Nicholas Metropolis and S. Ulam. The Monte Carlo Method. *Journal of the American Statistical Association*, 44(247):pp. 335–341, September 1949.

[7] Michael J. Quinlan, Naomi Henderson, Robin Fisher, Steven P. Nicklin, Stephen R. Young, Phavanna Douangboupha, Richard H. Middleton, Kenny Hong, Timothy G. Moore, Stephan K. Chalup, and Robert King. The 2005 NUbots Team Report. Available online: `http://robots.newcastle.edu.au/publications/NUbotFinalReport2005.pdf`, 2006.

[8] Adrian Ratter, Bernhard Hengst, Brad Hall, Brock White, Benjamin Vance, Claude Sammut, David Claridge, Hung Nguyen, Jayen Ashar, Maurice Pagnucco, Stuart Robinson, and Yanjin Zhu. rUNSWift Team Report 2010 Robocup Standard Platform League. Available online: `http://www.cse.unsw.edu.au/~robocup/2010site/reports/report2010.pdf`, 2010.

[9] Thomas Röfer, Tim Laue, Judith Müller, Armin Burchardt, Erik Damrose, Alexander Fabisch, Fynn Feldpausch, Katharina Gillmann, Colin Graf, Thijs Jeffry de Haas, Alexander Härtl, Daniel Honsel, Philipp Kastner, Tobias Kastner, Benjamin Markowsky, Michael Mester, Jonas Peter, Ole Jan Lars Riemann, Martin Ring, Wiebke Sauerland, André Schreck, Ingo Sieverdingbeck, Felix Wenk, and Jan-Hendrik Worch. B-Human Team Report and Code Release

2010. Available online: `http://www.b-human.de/download/publications/2010/bhuman10_coderelease.pdf`, 2010.

[10] Oleg Sushkov. *Robot Localisation Using a Distributed Multi-Modal Kalman Filter, and Friends.* Honours thesis, The University of New South Wales, 2006.

[11] Belinda Teh. Ball Tracking and Velocity. Taste of research report, University of New South Wales, 2011.

[12] R. van der Merwe and E. Wan. The Square-Root Unscented Kalman Filter for State and Parameter Estimation. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Salt Lake City, Utah, May 2001. IEEE.

[13] E. Wan and R. van der Merwe. The Unscented Kalman Filter for Nonlinear Estimation. In *Proceedings of Symposium 2000 on Adaptive Systems for Signal Processing, Communication and Control (AS-SPCC)*, Lake Louise, Alberta, Canada, October 2000. IEEE.

List of Figures

3.1 Transformed sigma points around the mean and covariance of an Unscented Kalman Filter . . . 10

3.2 Robot relative co-ordinate system . . . 12

4.1 Overview of goalie behaviour for different sections of the field . . . 16

4.2 Classified regions and the goalie's target position in each . . . 18

4.3 Squat for blocking the goalie's centre region . . . 23

4.4 Ball holding as the goalie returns to its default sitting position . . . 24

4.5 New centre squat transition to avoid ball holding . . . 24

4.6 Initial landing of the goalie dive . . . 25

4.7 Goalie dive rolling forward to get up . . . 26

4.8 Tall goalie stand for tracking far balls . . . 27

7.1 Team Nanyang's goalie kicking the ball away while performing its centre squat . . . 37

List of Tables

5.1 Penalty goalie's responses to a variety of ball directions and their success 30

5.2 Goalie positioning during its Patrol state 31

5.3 Goalie's reactions to nearby balls during the competition and the resulting effects . . 31