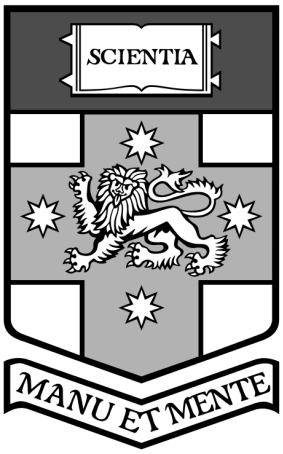
THE UNIVERSITY OF NEW SOUTH WALES
THE SCHOOL OF COMPUTER SCIENCE AND ENGINEERING



Humanoid Robot Localisation for the RoboCup Standard Platform League

rUNSWift

Author: Youssef Hunter (yhunter@cse.unsw.edu.au)
Supervisors: A/Prof. Maurice Pagnucco, Bernhard Hengst
Assessor: Prof. Claude Sammut

Thesis report submitted as a requirement for the degree
Bachelor of Engineering (Software Engineering)

August 24, 2012

Abstract

This thesis details the various methods and techniques applied to the localisation system of team rUNSWift for the RoboCup Standard Platform League. Extended Kalman Filters were used extensively with an emphasis on accurate transition and noise models. For robot pose localisation in particular, there was a focus on providing a stable and accurate main estimate by selectively applying observed feature matched estimates to a dual-modal system. This and the other techniques used proved successful during the RoboCup 2012 held in Mexico City.

Acknowledgements

I'd like to extend my personal gratitude to everyone who supported me throughout this incredible endeavour. Thank-you to the School of CSE and my team, supervisors, friends and family.

Contents

Chapter 1

Introduction

1.1 Motivation

Autonomous robotics is a relatively new and rapidly developing field of research, the aim of which is to develop robots capable of performing with little to no human interaction or guidance. This autonomy is particularly desirable in situations where a human presence would be considered uncomfortable or dangerous, as well as where constant remote human control isn't guaranteed – such as space exploration or investigating disaster affected areas.

Ideally, an autonomous robot would have the ability to plan intelligently into the future and react to their surrounding environment with some rational set of actions. However, in order to do this effectively, a robot must first gain information about the environment it is situated in. For example, whilst playing a game of soccer, to successfully kick a ball into the goal, a robot must first know the position of itself in relation to the position of the ball and the goals, as well as any obstacles that may be in the way. Localisation systems are used to interpret sensory observation data and create an estimate of the world state, in this way allowing the robot to localise itself by becoming aware of its surroundings as well as its own position within the environment.

This thesis is concerned with the continued development of localisation systems

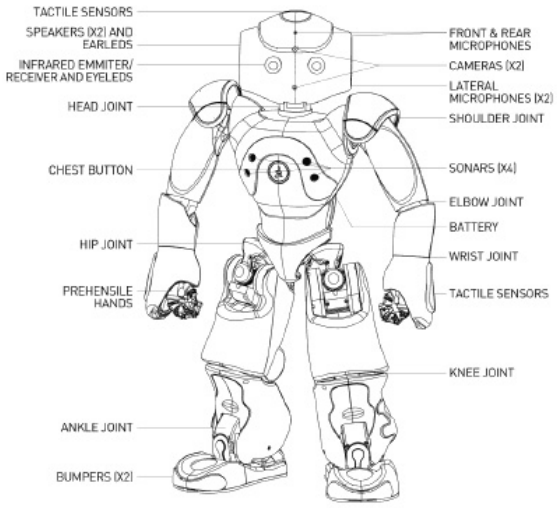


Figure 1.1: Aldebaran Nao Humanoid

within the domain of the International RoboCup Soccer World Cup, and in particular the University of New South Wales' team rUNSWift – participants of the Standard Platform League (SPL). The RoboCup SPL is a competitive soccer league with teams consisting of identical autonomous humanoid robots: the Aldebaran Nao (see Figure 1.1). The game of soccer provides a complex and interesting challenge for the robots as well as an exciting spectacle, while the competitive nature of the tournament encourages innovation and progress with each passing year. Furthermore, the standardised hardware platform of the league allows the developing teams to focus on the complex software systems required. As such, the RoboCup SPL provides an ideal domain and standardised performance measure for the development of localisation systems. The RoboCup Federation also approved a UNSW project to provide a game management, robotic commentary and refereeing system. This provided a simple initial starting point, requiring a less complicated localisation system that focuses on tracking the position and velocity of the ball as seen by four stationary sideline referee robots [1].

Localisation systems within the RoboCup SPL benefit from having accurate knowledge of the dimensions and layout of the field that the competition is played on. However measures must be taken to account for noisy partial observations from the sensors, false positives, as well as inaccurate motion controls. Historically, two main types of recursive Bayesian filters have been used by teams participating in the league to overcome these

obstacles and provide a robust and accurate localisation system [2–4]. Monte-Carlo particle filters use particles and statistical resampling to represent and approximate world states, allowing for observations which do not linearly map to a world state, as well as the tracking of multiple state hypotheses [5]. However, accurate prediction necessitates that a large number of particles must be used, meaning that this is significantly more computationally expensive than the other popular approach – Kalman filters.

In previous competitions, rUNSWift has predominantly employed Kalman filters because of their relative efficiency [2]. Basic Kalman filters use a normal Gaussian distribution to represent or approximate a belief (probability distribution) of the world state [6, 7]. Whilst efficient, this method quickly falls apart in situations where a single Gaussian distribution is not sufficient – for example the non-linear observations as mentioned previously. Throughout the years rUNSWift members have experimented with various improvements to, and adaptations of, the Kalman filter in an effort to overcome its limitations and improve the effectiveness of the team's localisation system [8–10].

In 2006, a combined robot Kalman filter used a complex multi-modal state space to encapsulate information of multiple robots as well as the ball into one filter [8]. Whilst quite successful, this is no longer feasible as the complexity of the approach scales exponentially – an increasing concern as the number of robots in competition matches increases over the years. In 2010, a simple hybrid of the particle and Kalman filters was pursued, switching between the two where appropriate: using the particle approach to refine the hypotheses down to one most probable estimate, then moving to the less demanding Kalman filter [2]. Unfortunately this approach was prone to compounding errors if the chosen hypothesis was inappropriate. To avoid the necessary fluctuation between the two approaches when observations clashed with the chosen hypothesis, a multi-modal Kalman filter was developed in 2011 which could handle multiple hypotheses [9]. However, the majority of 2011 development time was focused on the development of an Unscented Kalman Filter (UKF) that was abandoned due to complications and unpromising results, and hence the multi-modal Kalman Filter remained in relative infancy and had ample room for improvement.

1.2 Aims

The aim of this thesis is to continue the development of an effective localisation system that adequately fulfils a variety of necessary criteria;

- The system must provide both accurate and reliable information on the robot's own position as well as the position and velocity of the game ball. This knowledge is essential for effective soccer performance and strategy, such as the ability to accurately kick the ball towards the opponent's goal, respond to ball movement in order to successfully defend against incoming shots, and make informed decisions regarding team role assignment and positioning.

- The system must be robust against inevitably noisy sensory data (such as inaccurate odometry readings from the motion controls) and inaccurate position estimates of visual features, especially over large distances (this includes the ball, landmarks, and field features). Also of particular importance are the occasional falsely classified visual features, the presence of which can easily disrupt filters and systems that are not designed to cope with false positive information.

- The "speed" of the system is significant in more ways than one. It is imperative that the robot runs in real time using only the available on-board hardware – therefore, the computational complexity and processing requirements of the implemented system are strictly limited and enforced.

- The localisation itself must also be as fast as possible, with the goal being to maximise the reactivity and efficiency of the robot's play by minimising the amount of time spent actively localising – pausing and scanning or looking away from the ball with the specific purpose of gathering data to help pinpoint a location. The importance of the speed in which the localisation system reacts to sudden changes in ball position and velocity is highlighted by the often miniscule timeframe that the goalie has to decide to dive to successfully defend a strong kick from an opponent.

- The system must be aware of which direction the robot is facing without the aid of the obvious goal colour landmarks, and risks scoring a potentially devastating own

goal if the heading of the robot becomes incorrect. This is a new development for this year, where the RoboCup SPL rules were amended by altering the colours of the goal posts to be uniform and thereby creating an entirely symmetrical playing field. Previously, the distinct colours made distinguishing between home and away goals fairly trivial. This amendment comes as part of the gradual improvement to the league and has introduced a challenging new level of complexity to robot localisation.

- Finally, the system would be ideal if it remained as easy to understand, implement, use, develop and improve upon as possible. As the rUNSWift team changes each year with a new group of students and researchers, this will help the transition between years as well as ensuring the continued utilisation and advancement of the system.

1.3 Outline

The following Background chapter of this thesis will explain in further detail some of the challenges and requirements of a localisation system (Section 2.1), before evaluating in further detail some of the various aforementioned approaches to localisation, along with their benefits and limitations. The chapter will conclude with Section 2.7, which explains the rationale behind the approach that this thesis has undertaken.

Chapter 3 begins the detailed description and analysis of the methods of this thesis, examining the system used to track the ball's position and velocity.

Chapter 4 continues with the methods that the system uses to track the robot's own position and heading. Both Chapters 3 and 4 will provide performance results and an analysis of the effectiveness of the techniques used.

Chapter 5 includes suggestions for future improvements and developments to the localisation system.

Chapter 6 will provide a concluding summary for this thesis.

Chapter 2

Background

2.1 Localisation, Noise and Bayesian Filters

The challenge for a localisation system within the RoboCup SPL is to maintain an internal representation of the environment around it, known as its belief or estimate of the world state. This includes tracking a number of dynamically moving objects: its own pose (incorporating its position and the direction it is facing), the game ball's position and velocity, and ideally the pose of other robots (both friendly and opposing). These objects are represented as a collection of real number variables. This system must be accurate, robust against any false observations, and as efficient as possible, needing to run in real-time using the Nao's limited on-board processing power.

As the world around it is not directly observable in its entirety, the system must use sensor data (partial observations of the world) to estimate the world state over time. The sensor data mostly comprises noisy visual observations of certain field features that have been identified in the robot's camera frames. This includes detected goal posts, field lines (corners, edges, the centre circle), balls and other robots. In addition to the sensor data, knowledge of the control actions sent to the motors provides rough odometry information regarding the motion of the robot.

The abundance of noise in the environment and observations poses the greatest

obstacle to an ideal localisation system. Noise in this situation can be categorised into two basic types: process noise and observation noise [7, 11]. Process noise (also known as control noise) refers to the uncertainty associated with actions that the robots take. Consider a robot at some point A, instructed to walk forward for five seconds. In an ideal situation, we would predict that the robot is now at some point B a certain distance away – however a number of factors such as possible slippage, bumping into obstacles, or a slight imperfection or curvature in the walk, means that we instead say that it is probable the robot is located somewhere in the area around point B. Put simply, the existence of process noise obscures our predictions regarding robot actions such as movement.

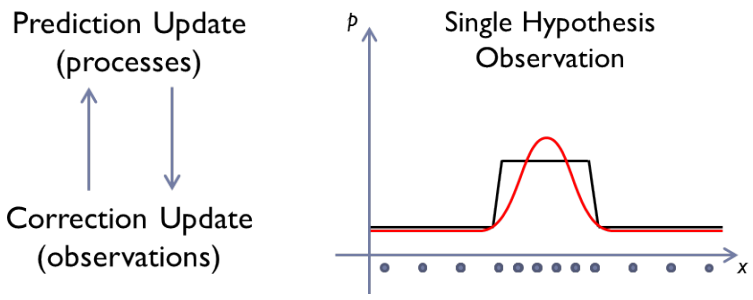
Observation noise refers to the uncertainty associated with the information received from the sensory input. In its simplest form, we can see observation noise in the slight differences between repeated measurements of the same variable – ideally our methods would be perfect and multiple observations would be equal, but realistically the best estimate might be the average or median of them all. In the domain of RoboCup localisation, observation noise is a major factor because the computer vision systems that are developed to extract useful observations from camera data are not perfect (as inaccuracies are to be expected) [2]. Observation noise must be considered when observations are used in order to update beliefs about the world state.

These process and observation noises are typically statistical in nature (or can be roughly modelled as such) and ultimately create a level of uncertainty in a robot's belief of the world state. As such the state estimate becomes a probability distribution to which we can apply stochastic methods. Two types of Recursive Bayesian Filters have emerged as the popular choices for stochastic state estimation within the domain of RoboCup localisation systems. Both types involve two basic recurring steps: prediction updates and correction updates (see Figure 2.1a) [11].

Prediction updates for Recursive Bayesian Filters are also known as process updates, and use the processes/controls/instructions given to the robot. These are incorporated with the previous belief state to infer an estimate of the current world state. Using the example scenario mentioned previously, a prediction update would cause a robot whose control data claims it has moved forward a certain amount to move the belief of

its current position forward. In most cases, the process noise associated with the control input is taken into account by increasing the uncertainty of the belief, as it has yet to be validated by any observation data.

Correction updates then apply the sensor input observations of the world to provide an improved estimate – possibly either reinforcing or weakening the current belief depending on its comparison to the current estimate. For example, if the prediction update shifted the robot's position belief forwards by ten centimetres, but the visual observations of certain landmark data suggests the robot has only moved forward five centimetres, then the correction update would shift the belief backwards to more closely fit the observations. The observation noise associated with the observations affects the amount in which the belief is shifted because of the observation, as well as the resulting uncertainty of the new belief. The prediction and correction updates are repeated at each time step, using the continuous stream of process control data and observations. Over time the robot's belief about the current world state is established and refined.



(a) Update Cycle (b) Probability Representations

Figure 2.1: Recursive Bayesian Filters

The two popular Bayesian Filters in use – particle and Kalman Filters – mainly differ in their representations of the state probability distribution. Particle filters, also known as Monte Carlo Localization (pioneered by Fox et. al [5]) use a large number of particles, each representing a different state hypothesis, with the largest concentration of particles indicating the most likely world state. With each cycle of the filter, a process update is applied to each particle, estimating and adjusting the particles to the new state. The observation update then compares the particles to the observations it has received, rewarding those that match more closely with a higher weight. The particles are then

resampled using Monte Carlo methods, with the higher weighted particles more likely to appear in the resultant set. The particle filter approach has been quite popular in the past [3,4], but the large number of particles necessary for an accurate estimation requires a great deal of processing, which further increases as the state becomes more complicated with more robots to track.

2.2 Kalman Filter Algorithm

The Kalman Filter, first proposed in 1960 [6], is less computationally expensive than its particle counterpart, and has been the approach favoured by the UNSW team. This approach represents the world state as a normal Gaussian distribution, with a mean vector (the state estimate, $\hat{x}$) of the state variables, as well as an error covariance matrix (the estimated uncertainty of the state estimate, P). The assumption is made that the state can be approximated as such a distribution, as well as that the types of noise described earlier are also Gaussian. Welch and Bishop [7] provide an introductory explanation to the Kalman Filter, which will be summarised here.

The underlying model of the filter is represented by the following two equations and their associated definitions. Equation 2.1 depicts the evolution of the current true state of the system (x_k) from the previous state (x_{k-1}), while Equation 2.2 represents observations (z_k) of the current true state.

$$x_k = Fx_{k-1} + Bu + w \tag{2.1}$$

$$z_k = Hx_k + v \tag{2.2}$$

- F: The state transition model: a matrix relating the previous vector state to the current state.
- u: The control vector: representing the control inputs of the robot to the filter.
- B: The control-input model; a matrix relating the control vector to the world

state: that is, transforming the control input to the same form as the world state's representation.

- w: The process noise, as described in Section 2.1.
- H: The observation model: a matrix mapping the true state into the observation space, used to compute the predicted measurement of the predicted state.
- v: The observation noise, as described in Section 2.1.

2.2.1 Prediction Update

$$\hat{x}_k = F\hat{x}_{k-1} + Bu \tag{2.3}$$

$$P_k = FP_{k-1}F^T + Q \tag{2.4}$$

- $\hat{x}_k$; The state estimate: the mean vector of the Gaussian distribution representing the world at state k, that is, the best estimate of the variables tracked by the localisation system at some time k.
- P_k; The error covariance matrix: a measure of the estimated uncertainty of the state estimate at time k.
- Q; The covariance matrix of the process noise w, a measure of the uncertainty of the control input.

The above equations (2.3 & 2.4) and their relevant notation describe the prediction (process) update of the Kalman Filter, based on the prediction model (Equation 2.1). Equation 2.3 predicts the new state estimate vector $\hat{x}_k$ by combining the previous estimate $\hat{x}_{k-1}$ and its transition matrix F, as well as the control input u and its transition matrix B. Equation 2.4 similarly updates the error covariance matrix to reflect the new predicted state estimate: the transformation F is applied to the previous error P_{k-1} to coincide with the transformation of the state estimate. The covariance Q of the control noise is also added.

The result of the update is the new "predicted" state estimate, with a mean vector that has progressed since the last state and taken into account any control actions the robot has carried out. For Kalman Filters, the uncertainty of the state estimate increases due to this prediction, with the size of the increase correlated with the uncertainty of the control input.

2.2.2 Correction Update

The correction update of the Kalman Filter, based on the observation model described earlier with Equation 2.2, uses the following equations to refine the state estimate.

$$\tilde{y} = z - H\hat{x}_k \tag{2.5}$$

The innovation $\tilde{y}$ is a measure of the difference between the predicted state estimate and the actual observed state of the world. The predicted state estimate $\hat{x}_k$ (from the prediction update equation 2.3) is mapped into the observation state space using the observation model H, which is then subtracted from the current observation z.

$$K = P_k H^T (H P_k H^T + R)^{-1} \tag{2.6}$$

This equation (2.6) provides the optimal Kalman Gain, denoted by K, which regulates the effect that the observation will have on the state of the filter. The result is optimal in the sense that this value for K will minimise the final estimated error covariance of the state; in other words, it provides the most accurate state estimate assuming the assumptions of the Kalman Filter are held. While the precise derivation of this formula lies beyond the scope of this thesis (see Welch and Bishop [7] for more information), the major premise is the combination of the error covariance (P) of the predicted state as well as the error covariance (R) of the observation noise (v). The observation model H is again used a transformation into the observation state space.

$$\hat{x}_k^+ = \hat{x}_k + K\tilde{y} \tag{2.7}$$

$$P_k^+ = (I - KH)P_k \tag{2.8}$$

Equation 2.7 refines the state estimate using the innovation $\tilde{y}$ (equation 2.5) combined with the Kalman Gain K (equation 2.6). The Kalman Gain incorporates the uncertainties of the current state estimate and the observation – a correction update with an initially highly uncertain predicted state and an accurate observation would heavily weight the innovation and bring the resulting state estimate close to the observation. However, if the correction update was based off an uncertain observation and the predicted state's uncertainty was relatively low, then the innovation would not have a substantial effect on the filter's state. Equation 2.8 shows how the Kalman Gain is also used to adjust the resulting state's error covariance appropriately.

In short, at the conclusion of a prediction update followed by a correction update, the current filter's state has been extrapolated from the previous state and its control data, and then adjusted using current observations.

2.2.3 Assumptions and Limitations

While computationally less expensive than the particle filter, the Kalman Filter's updates rely on a number of assumptions that should be further explored. Consider an observation which results in only one hypothesis, such as detecting a door in front of a robot. The black line in Figure 2.1b (page 8) reflects the true probability density function of the robot's location, while the red line depicts a normal Gaussian approximation used by the Kalman Filter, with the candidate particles of the particle filter shown for comparison. Both filters represent this situation adequately; however, the Kalman Filter does so more efficiently by taking advantage of a situation that has adhered to its strict assumptions.

However, merely expanding on the above example leads to a situation disastrous for the simple Kalman Filter's assumptions. If the robot is standing in a hallway with two doors, an observation of a door leads to two hypotheses. Figure 2.2 shows how a

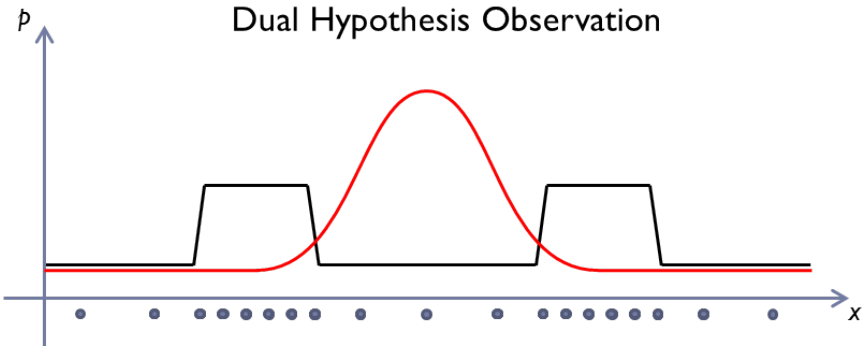


Figure 2.2: Normal Gaussian Probability Representation

single normal Gaussian fails to approximate this new probability density function, while the particle filter approach to representation remains accurate.

The standard Kalman Filter also assumes that the state transition and observation models (F, B, and H in the model and update equations) are linear functions, which is often not the case. Furthermore, the Kalman Filter assumes that no false positive observations occur, and reacts quite poorly in their presence. To overcome the aforementioned limitations while maintaining an efficient localisation system, rUNSWift teams over the years have experimented with numerous variations and extensions of the standard Kalman Filter.

2.3 Extended Kalman Filter

The Extended Kalman Filter (EKF) relaxes the assumptions of linearity of the standard Kalman Filter. In contrast with the underlying model of the Kalman Filter (Equations 2.1 and 2.2), the EKF allows for non-linear state transition, control-input and observation models, and hence can be applied to a much larger variety of systems. Using a similar notation to the standard Kalman Filter, the equations that define the underlying model of the EKF are as such;

$$x_k = f(x_{k-1}, u) + w \tag{2.9}$$

$$z_k = h(x_k) + v \tag{2.10}$$

The differentiable functions f and h replace the linear matrix operators A and H of the Kalman Filter, with f being used to predict the current state from the previous, and h used to transform the state into observation space (and calculate the predicted measurement of the predicted state). Crucially, these non-linear functions are not applicable to the covariance matrix of the state estimate P, implying that variations on the original prediction and correction update equations are required.

Hence, the EKF uses Jacobian matrices of the non-linear functions to adjust the state estimate's covariance matrix. These Jacobian matrices represent tangential linear approximations to the non-linear functions. They are computed by taking the multi-dimensional partial derivatives of the non-linear functions with respect to the state variables.

The prediction and correction updates detailed in Section 2.2 (Equations 2.3 to 2.8) retain the same form, however the matrices F and H are replaced with their appropriate Jacobian approximations:

$$F = \frac{\delta f}{\delta x} \qquad H = \frac{\delta h}{\delta x} \tag{2.11}$$

Welch and Bishop provide a more extensive description and derivation of the Extended Kalman Filter equations in their Introduction to the Kalman Filter [7].

2.4 2006 – Multi-Modal Extended Kalman Filter

Quinlan and Middleton [12] provide a detailed description of Multi-modal Kalman Filters and their application to the RoboCup SPL. These filters are an extension of the basic Kalman Filter, used to handle multiple hypotheses as described in the example above. Rather than using one Gaussian to attempt to approximate the situation, multiple Gaussians are used and a weighted sum of them is calculated to represent the full state. Each Gaussian represents a single mode, and the Gaussians are weighted according to the probability that the corresponding mode represents the whole system. Consider again

the example above, of a robot observing a door resulting in two hypotheses. Figure 2.3 depicts two Gaussians representing the two hypotheses, with the sum of the Gaussians representing the probability distribution of the whole state.

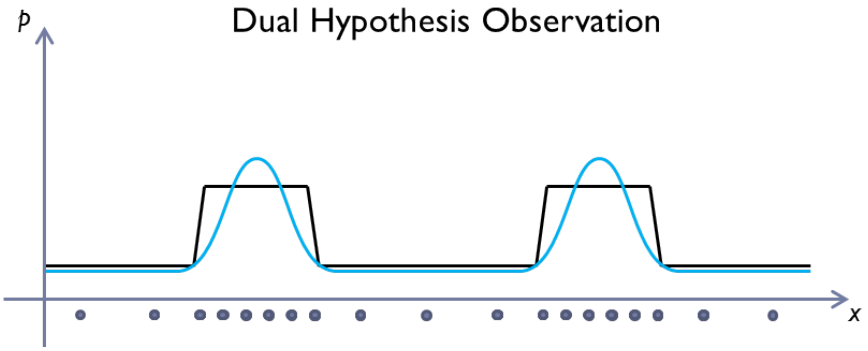


Figure 2.3: Multi-modal Gaussian Probability Representation

In 2006, Oleg Sushkov [8] combined the Multi-modal approach with the Extended Kalman Filter (introduced in Section 2.3) to successfully track the pose of a robot, its 3 teammates, as well as the location and velocity of the ball in one filter. This 16-dimensional state space allowed for each robot to easily incorporate its teammates' observations into its own filter, which greatly increased the accuracy of their ball tracking. Furthermore, as the ball itself is encapsulated by the filter, it acted as a moving beacon to other robots, which helped increase the accuracy of their robot pose tracking. Unfortunately however, grouping together the robots requires a multiplicative number of modes to adequately represent hypotheses, and hence this complex approach scales poorly with the introduction of more robots (either with larger teams or attempts to also track opponent robots).

2.5 2011 – Unscented Kalman Filter

Julier and Uhlmann [13] proposed the Unscented Kalman Filter: a theoretically better approximation technique than the Extended Kalman Filter discussed previously. The principle idea is to use the unscented transform (a deterministic sampling technique) to select a small number of points around the mean of the observation estimate. These points, known as sigma points (distributed one standard deviation around the mean), are

then propagated through the non-linear prediction and update functions (that the UKF is modelling). The result is a more accurate estimation of the true mean and covariance. Another advantage of this approach is that the Jacobian Matrices of the EKF are not required, meaning that the potentially complex problem of calculating them is avoided.

2011 saw an attempt from David Claridge [9] to develop a generic Unscented Kalman Filter that could be applied to a variety of observations, and be used to track either robot poses or the ball. This approach ran into persistent technical difficulties, and was showing unpromising results when applied to this domain. The filter seemed adequate for use as a ball filter however, and a separate ball filtering system was developed by Belinda Teh [10], using two UKFs – one assuming the ball is still, the other that the ball is in motion. As for robot localisation however, the UKF was abandoned.

2.6 2011 – Multi-Modal Linear Kalman Filter

With the international RoboCup fast approaching and the UKF development abandoned, Claridge [9] pursued a much simpler method. A Multi-modal Linear (standard) Kalman Filter was developed, with a number of slight adjustments to make it suitable for RoboCup robot localisation. To overcome the non-linearity of certain observations (such as observing only one field edge, or a single goal post), a number of 'hand-crafted' geometric linear transformations were developed – avoiding the more complex linearisation methods. An ad-hoc heuristic was also developed for determining which of the multiple modes should be updated with each observation.

This filter performed well in the 2011 RoboCup, but the limited development time left plenty of room for improvement. One of the largest drawbacks for the multi-modal system was the robots would occasionally became "lost", requiring "localisation scans" that involved pausing and moving the head to collect more visual data to provide the filter's modes with enough information to be certain of the robot's position. This would often result in the loss of many precious seconds, especially important when the speed in which the robot can locate, move toward and then kick a ball into the opponent's goal

is such a critical factor.

2.7 A Different Approach

A variety of considerations were taken into account before deciding on a final approach to pursue for this thesis. Efficiency is often a key factor in the decision making process, and 2011 saw an increase in team size from three to four robots with suggestions of further increments to five robots in the near future. The quality of higher level strategy and planning in the competition has also steadily improved, making it increasingly more worthwhile to track the opposing team. Hence efficiency and scalability to further complexity is an increasingly important consideration as the competition becomes more complex and the number of objects that need to be tracked increases.

The computational expense of particle filters has been mentioned previously, and with the increasing emphasis on efficiency it was decided to continue exploring Kalman filters. Furthermore, despite their unpopularity in the past, there has been a growing trend amongst other teams in the competition towards incorporating the more efficient Kalman filter, lending merit to Kalman based decisions [12, 14].

2.7.1 Ball Tracking

The RoboCup Federation approved the UNSW project to develop a game management system, which calls for four stationary robots positioned around the field to collaborate their observations and collectively referee a match [1]. This undertaking increased the emphasis on the accuracy of ball observations and also the distribution and integration of this information between team members. This is because a more accurate estimate of the true position is possible by combining the ball observations of all the robots. This combined "team ball" information is also useful in identifying and correcting robots which have a mistaken belief of their position, as the ball is a unique landmark and a robot who disagrees with the rest of the team on its position is likely to be misguided.

With those motivators in mind the ball tracking system was re-implemented. The approach is similar to the 2011 approach developed for rUNSWift by Teh [10], but with a number of key differences:

- The error covariance matrices of the ball observations more closely resemble the true uncertainty, replacing the previous arbitrary estimates with a function of the polar coordinates of the observation. Providing more accurate uncertainty estimates of observations greatly improves the accuracy of a Kalman Filter's resultant state error covariance matrix – which is of particular importance when the robots are collaborating and sharing their state estimates. The B-Human team's approach also makes use of covariance matrices based on the ball observations [15].

- As the UKF used in 2011 was complex and plagued with technical issues, and also didn't seem to be a significant improvement over an EKF, the new system utilises the simpler Extended Kalman Filter variant.

- The dual-modal approach of 2011 is maintained – one filter assuming the ball is stationary, the other that it is moving. However choosing between the two filters is based on a comparison of the innovations of the two filters rather than ad-hoc error covariance checks, to maintain a quick and reliable reaction to sudden changes in the ball's velocity (such as when it is kicked or stopped). This is similar to the B-Human approach, which utilises a complex system of 12 filters – half assuming the ball is stationary, and the other half that it is moving [15].

Further, Teh's 2011 [10] approach of the calculation of the team ball was refined. This contrasts to Sushkov's 2006 approach [8] of a complex state space that includes every robot's position and the position of the ball, which we established requires a multiplicative number of modes and scales poorly (see Section 2.4). Instead, each robot filters their individual observations of the ball, and then simply distributes this filtered estimate of the ball's position to the rest of the team. Each robot then uses these along with its own estimate to calculate the team's combined estimate.

The final combination of the collected estimates has been refined in this thesis'

approach. Rather than naively incorporating every estimate into the combined team estimate, only the best subset of the estimates is combined. This allows for one or more of the robots to be lost without completely throwing off the team's estimate. Furthermore, it allows us to identify if a robot is lost if it disagrees with a majority of the team.

2.7.2 Pose Tracking

So far, much of the discussion of Kalman Filters has centred on linearisation techniques (Unscented versus Extended Kalman Filters), as well as multi-modal mechanisms for dealing with observations that provide multiple hypotheses. This is reasonable, as the rUNSWift localisation systems to date have used a variety of observed field features from each camera image to generate a series of observations at each state – some of which are non-linear or provide multiple hypotheses.

This year, the advancement of the standard platform to the Nao v4 has facilitated a slightly different approach to localisation. Most importantly, the Nao v4 allowed the use of both of the Nao's cameras simultaneously, as well as a faster processor. The faster processor also meant more processing power could be used to extract features from a higher resolution of the top camera (where the majority of field features are observed). The result of the upgrade essentially meant that the number and quality of observed field features dramatically increased.

The approach developed last year by Claridge [9] (see Section 2.6) used each observed feature independently, aside for a small number of exceptions such as a pair of goal posts. The new approach uses an Iterative Closest Point (ICP) feature matching algorithm developed with Peter Anderson [16] to combine all the available feature information, with the current estimate of the robot's position as a starting point, to produce a single powerful observation of the robot's position. The details of this module are contained within a separate report. This innovative single observation greatly simplifies the requirements of the filter:

- The observation provided by the ICP method is directly comparable to the state of

the filter; a position of the robot and a covariance error matrix. Hence it is possible to filter this information directly, without the need for non-linear extensions of the Kalman Filter's correction update, or any manual linearisation methods. The simplification of the observation model has also allowed time for a more accurate process model to be pursued, using the EKF's prediction update.

- The ICP method uses the robot's currently estimated position as a starting point and only provides one combined observation/hypothesis for each set of visual features. As this single hypothesis is easily represented by a single Gaussian, there is little need for complex multi-modal extensions of the Kalman Filter, or ad-hoc heuristics to determine which modes should receive which updates, or when to create, merge, limit, remove or switch modes.

The single ICP observation almost allows the system to track the robot's position using a single EKF. However a single filter will still be susceptible to highly inaccurate or false positive visual feature data – which in some cases might cause the ICP matcher to provide a false hypothesis as an observation to the filter. The system must also overcome the "kidnapped robot" problem; being able to localise without prior information or a false prior belief of its position. In this domain this situation can occur in a variety of ways, such as when a robot recovers from a penalty (it must decide which sideline it has been placed on), or when it has been moved (whether accidentally or to prevent interfering with other robots).

The approach used in this thesis is another dual-modal system, utilising two relatively simple EKFs. One filter is the main mode, tracking the current best estimate of the robot's position. The other filter is the alternate mode, which is only occasionally active and represents a single alternative position for the robot. The alternative mode captures outlying observation hypotheses from the ICP that are deemed too far away from the current main mode to be reasonable – implying that the observation is flawed, a significant jump has occurred or an error has developed in the main mode. The alternate mode is only swapped into the main mode when it has proven to be an obviously superior estimate.

Using the single ICP observation and simple dual-modal filter reflects a much greater emphasis on providing a stable estimate of the robot's position: once localised, the robot relies on its main mode and the now-improved stream of field features to stay localised. This avoids having to interrupt higher level play strategy with requests to gather more useful data to decide between competing modes.

The alternate mode is intended for use as a backup error correction mechanism, allowing for larger and more sudden "jumps" when the system is sure of the alternate hypothesis. Of particular interest are jumps that swap the position of the robot to the opposite side of the symmetrical environment. The team ball information, as well as natural landmarks that are detected in camera frames, are also incorporated in the ICP matching algorithm to indicate which side of the field a hypothesis belongs to. This helps to ensure that "flips" to an erroneously opposite robot heading can be recovered from, hopefully before any disastrous own goals.

Chapter 3

Ball Tracking

This chapter will examine and analyse the system used to track the game ball's position and velocity. The approach used to distribute this information between each robot and determine a collective "team ball" belief is also addressed.

3.1 Method

As outlined in Section 2.7.1, the ball tracking system was re-implemented with a greater emphasis on a more accurate model of observation uncertainty – with the intent to improve the accuracy and reliability of each robot's ball state estimate. The accuracy of each robot's individual ball estimate, as the well as the accuracy of the uncertainty of that estimate, becomes increasingly important as the robots distribute and combine their beliefs. This collaborative estimate of the ball's position using the combination of each individual robot's beliefs is the next step in accurately tracking the game ball, which in turn can be applied to improving the localisation of the robot's themselves. Section 4.1.4 provides more information regarding the use of the team ball to help localise in a symmetrical environment – particularly relevant this year with the advent of uniform goal colours.

3.1.1 Extended Kalman Filter Approach

Chapter 2 provides an introduction to Kalman Filters as an efficient method of filtering the noisy observations that are characteristic of our robotic sensors. Also mentioned are the Extended (EKF) and Unscented (UKF) variants of the Kalman Filter, which provide the additional functionality of filtering non-linear systems and observations by using approximations – with the EKF employing derived tangential functions, and the UKF utilising distributed sigma points. The EKF was chosen for the re-implementation of the ball tracking system, allowing for sufficient linearisation capabilities whilst avoiding the added complexities and technical difficulties of the UKF experimented with in 2011 [9].

Dual-Modal

The dual-modal approach used in 2011 [10] was re-implemented using two new EKFs. The approach involves two similar filters, each with a different assumption of the ball's movement. The first filter simply assumes the ball is stationary and tracks only its position. The ball is stationary for a significant proportion of each game; however when it does move the stationary filter is slow to react to the sudden change in position. The second filter assumes the ball can be in motion, tracking its position as well as the velocity of its movement. This allows it to react more quickly to sudden changes in the ball's position and provide valuable information on where the ball is headed, although when the ball is stationary the moving filter reacts poorly to noisy observations and becomes unstable. Hence both the stationary and the moving filter are used in the system to overcome the limitations of each used separately.

State Representation

Both filters track the ball's state in the Cartesian form of rUNSWift's standard robot-relative coordinate system. Shown in Figure 3.1, this system treats the robot's base as the origin, with the positive x and y axes extended forward and to the left of the robot respectively. Other possibilities included the polar robot-relative system, or absolute

Cartesian coordinates (with origin at the centre of the field).

The decision to use robot-relative coordinates stemmed from the importance of the ball's position to individual robot behaviours, as well as the robot-relative nature of the ball observations. This allows the ball tracking system to operate independently of the robot's localisation. The final resultant robot-relative estimate of the ball's position is then easily combined with the robot's position to provide an absolute ball position (see Section 3.1.6).

Cartesian coordinates were chosen over polar coordinates as it was found that this more easily facilitated the tracking and display of the ball's state, as well as the covariance of the state estimate. In particular, converting the ball's observations and the associated noise to Cartesian form allowed for simpler and more uniform filters that did not require internal conversions.

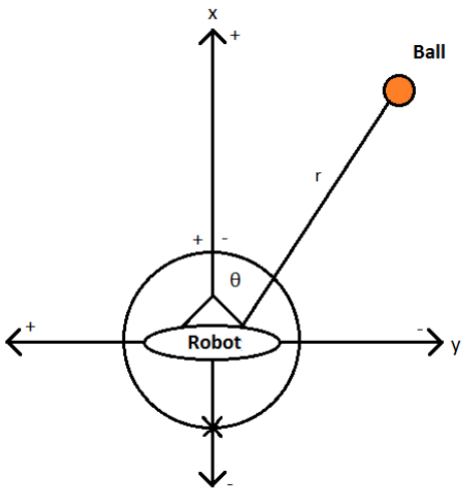


Figure 3.1: rUNSWift Robot-Relative Coordinate Systems [10]

Process Odometry and Noise

As the state of each filter is represented as robot-relative coordinates, they are affected by the robot's own movement – a robot walking towards a stationary ball would clearly observe the ball closer and closer. If the robot's movement is not taken into account, this can easily be misconstrued as noise or even the ball having a velocity of its own.

Consequently, each filter uses odometry information provided by the robot's walk engine [17] and new visual odometry system [16] as control inputs to appropriately shift the coordinate system and predict the ball's new relative position as the robot moves. The odometry information is in the form of a vector (u) of three variables representing an estimate of the robot's counter-clockwise rotation in radians ($\Delta\theta$), as well as an estimate of its movement forward (ΔF) and to the left (ΔL) in millimetres. By convention the robot's rotation is applied first. Sections 3.1.3 and 3.1.4 explain the individual filters in more detail, including an explanation of how odometry is used as part of the process update's state transition.

$$u = \begin{bmatrix} \Delta\theta \\ \Delta F \\ \Delta L \end{bmatrix} \tag{3.1}$$

The typical assumption is made that the process noise associated with the odometry updates of the ball filters is additive. This implies that an estimate of the covariance of the process noise is directly added to the state covariance with each prediction update. This covariance was determined (via experiment) for each filter to arrive at estimates that provided stable yet sufficiently reactive state estimates.

An attempt was made to incorporate the noise of the odometry update based on each variable's impact on the state; however this proved to be complex and problematic, and was abandoned as a linear approximation of such a relationship was unlikely to provide any substantial improvement on the filters' performance.

3.1.2 Observations and Elliptical Covariance

Ball observations are provided to the system from the vision module as polar robot-relative coordinates. This system is also shown in Figure 3.1, with a straight line distance (r) as well as its heading (θ); the angle from the robot to the ball (with positive angles to the left and negative angles to the right).

These ball observations are understandably susceptible to noise, and one of the

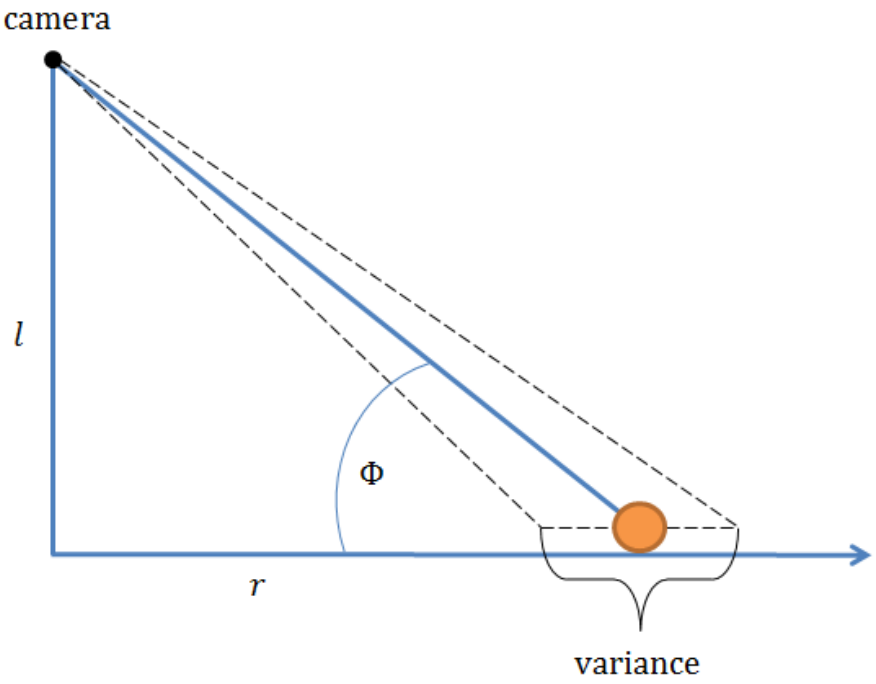


Figure 3.2: Ball Distance Observation Variance [18]

aims of this thesis is to more accurately model the covariance of the noise of these observations. Over the years it has been empirically noted that estimates of the robot's angle to the ball are relatively far more accurate than the estimate of the distance to the ball. This is to be expected as the calculation of the distances to objects is more heavily influenced by errors in the robot's kinematic chain and ground plane projection [2]. Errors in the kinematic chain are also clearly more detrimental to estimates of objects that are further away.

These properties of the observation noise are successfully reflected by calculating the variances of the r and θ observations of the ball. Hengst [18] describes an approach using the assumption that the standard deviation (σ) of the angle of the ray projecting from the camera (ϕ, shown in Figure 3.2) to the ball is a constant (σ_ϕ). Equations 3.2 to 3.4 examine the trigonometric relationship between ϕ, r, and l (the rough height of the robot's camera above the ball), isolating the change in r with respect to ϕ. Hengst's resulting equation (3.4) implies σ_r is a function of r^2, l, and σ_ϕ. Squaring σ_r easily provides us with the variance of the distance observation (used later in equation 3.8), which predictably grows rapidly as r increases.

$$\tan(\phi) = l/r \tag{3.2}$$

$$\frac{\delta r}{\delta \phi} = -\frac{l}{\sin^2(\phi)} \tag{3.3}$$

$$\delta r = -l\delta\phi - \delta\phi\frac{r^2}{l} \tag{3.4}$$

Similar to ϕ, the standard deviation of the θ observation is estimated and assumed to be a constant (σ_θ). However instead of passing the r and θ observations and their variances to the filters and having them use complex observation Jacobian matrices to translate these variances into appropriate Cartesian values, the observations and their variances are first converted into the Cartesian plane and then passed into the filters. The transformation from the polar observation coordinates to their Cartesian form is fairly simple:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} r\cos(\theta) \\ r\sin(\theta) \end{bmatrix} \tag{3.5}$$

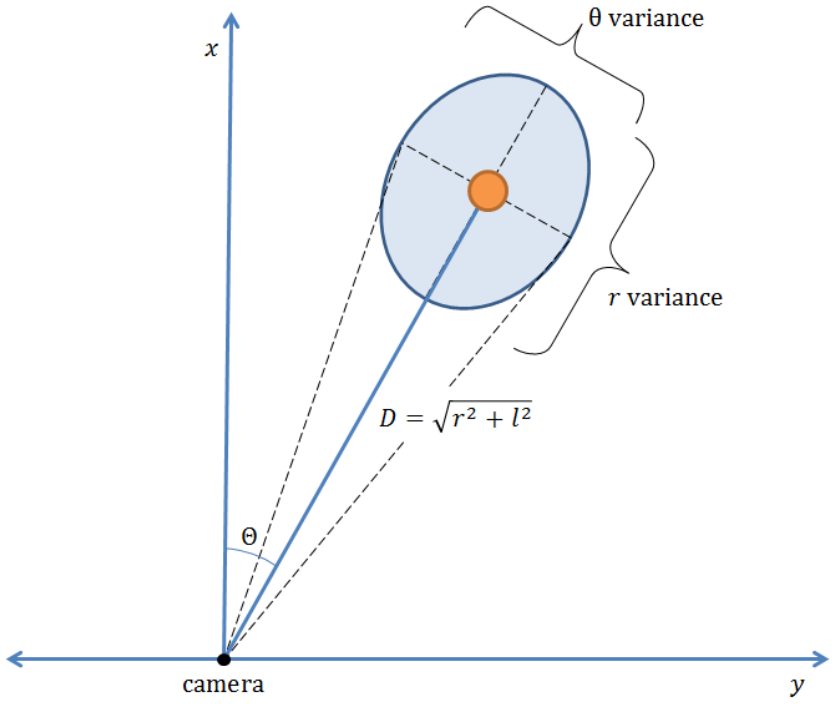


Figure 3.3: Ball Heading Observation Variance

The critical step is to create the new covariance matrix to accompany the Cartesian observations. Figure 3.3 shows a top-down view of a robot observing a ball, with the r and θ observation variances shown as the major and minor axes of a Cartesian covariance ellipse. The easiest method to create this ellipse is by creating a simple covariance matrix using the major and minor axes, and then rotating it to the desired orientation using the ball's observed θ. The variance of r has been derived above, and this can be used as the magnitude of the major axis directly. To derive the length of the minor axis however, the constant σ_θ is applied to the following equation:

$$\text{Length of minor axis} = (D\tan(\sigma_\theta))^2, \text{ where } D = \sqrt{r^2 + l^2} \tag{3.6}$$

The rotation of the covariance ellipse by θ is achieved by applying a rotation matrix (R) using the following equation for the transformation of covariance matrices described by Soler & Chin [19]:

$$\Sigma_{\bar{x}} = R\Sigma_x R^T, \text{ where } R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cot\theta \end{bmatrix} \tag{3.7}$$

The final resulting covariance matrix of an observation in Cartesian form is thus as follows:

$$\begin{bmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{yx} & \sigma_y^2 \end{bmatrix} = R \begin{bmatrix} (l\sigma_\phi + (\sigma_\phi r^2)/l)^2 & 0 \\ 0 & (\sqrt{r^2 + l^2}\tan(\sigma_\theta))^2 \end{bmatrix} R^T \tag{3.8}$$

By correctly modelling the covariance of the noise of the ball observations, the filters can more accurately track the state of the ball. The effect of the accurate covariance is to treat each observation according to the properties that were outlined earlier; weighting observations of the angle to the ball (θ) heavily over the distance to the ball (r), as well as increasing the uncertainty of observations further away.

The resulting covariance estimates of the filter states also reflect this model of uncertainty – Section 3.2.1 showcases a variety of resulting state estimates and their covariance ellipses, and how these estimates can be combined from multiple robots to provide more accurate measurements of the ball's true position.

3.1.3 Stationary Filter

As the name suggests, the stationary filter assumes the ball is stationary and tracks only the position of the ball. Hence the state estimate vector ($\hat{x}$) and its corresponding uncertainty (P) are as follows:

$$\hat{x} = \begin{bmatrix} x \\ y \end{bmatrix} \qquad P = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{yx} & \sigma_y^2 \end{bmatrix} \tag{3.9}$$

Prediction

With the odometry vector u (see equation 3.1) as input, the non-linear prediction update (state transition) equations of the stationary filter are shown below. They offset the robot's movement by rotating the ball's coordinates by the opposite of the robot's turn, and then shifting the position of the ball by the robot's forward and left movement.

$$\begin{bmatrix} x_{\text{new}} \\ y_{\text{new}} \end{bmatrix} = \begin{bmatrix} \cos(-\Delta\theta) & -\sin(-\Delta\theta) \\ \sin(-\Delta\theta) & \cos(-\Delta\theta) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} \Delta F \\ \Delta L \end{bmatrix} \tag{3.10}$$

$$= \begin{bmatrix} x\cos(-\Delta\theta) - y\sin(-\Delta\theta) - \Delta F \\ x\sin(-\Delta\theta) + y\cos(-\Delta\theta) - \Delta L \end{bmatrix} \tag{3.11}$$

The state transition Jacobian (Equation 3.12) is obtained by partially differentiating the state transition Equation 3.11, and is used to appropriately adjust the covariance of the state estimate in accordance with the effect of the odometry on the state. When applied to the covariance estimate it rotates the covariance ellipse by the same angle the state estimate is adjusted. To facilitate computation, the odometry rotation matrix used in Equation 3.10 is reused as the Jacobian matrix.

$$F = \begin{bmatrix} \frac{\delta x}{\delta x} & \frac{\delta x}{\delta y} \\ \frac{\delta y}{\delta x} & \frac{\delta y}{\delta y} \end{bmatrix} = \begin{bmatrix} \cos(\Delta\theta) & \sin(\Delta\theta) \\ -\sin(\Delta\theta) & \cos(\Delta\theta) \end{bmatrix} \equiv \begin{bmatrix} \cos(-\Delta\theta) & -\sin(-\Delta\theta) \\ \sin(-\Delta\theta) & \cos(-\Delta\theta) \end{bmatrix} \tag{3.12}$$

Correction

As explained in Section 3.1.2, the ball observations are transformed and provided to the filter in robot-relative Cartesian coordinates with their associated covariance matrices. The intended effect of this is a very simple linear correction update, as the state vector and observations are of the same form. The standard linear Kalman Filter observation correction equations can be used, with an identity observation model.

3.1.4 Moving Filter

The moving filter allows for ball movement and has the capability of tracking the ball's position as well as its velocity. The filter is thus slightly more complex than its stationary counterpart. Its state vector and covariance matrix are defined as:

$$\hat{x} = \begin{bmatrix} x \\ y \\ x' \\ y' \end{bmatrix} \qquad P = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} & \sigma_{xx'} & \sigma_{xy'} \\ \sigma_{yx} & \sigma_y^2 & \sigma_{yx'} & \sigma_{yy'} \\ \sigma_{x'x} & \sigma_{x'y} & \sigma_{x'}^2 & \sigma_{x'y'} \\ \sigma_{y'x'} & \sigma_{y'y} & \sigma_{y'x'} & \sigma_{y'}^2 \end{bmatrix} \tag{3.13}$$

Prediction

Similar to the prediction update of the stationary filter, the moving filter's state update equations must account for the movement of the robot provided by the odometry vector u. However the moving filter first accounts for the movement of the ball by predicting the ball's new position with its tracked velocity and the time elapsed since the previous update (T). The position and velocity of the ball are then rotated and translated using the odometry values. The velocity of the ball is also scaled down by a constant λ to account for the effect of friction on the ball.

$$\begin{bmatrix} x_{\text{new}} \\ y_{\text{new}} \\ x'_{\text{new}} \\ y'_{\text{new}} \end{bmatrix} = \begin{bmatrix} (x + Tx')\cos(-\Delta\theta) - (y + Ty')\sin(-\Delta\theta) - \Delta F \\ (x + Tx')\sin(-\Delta\theta) + (y + Ty')\cos(-\Delta\theta) - \Delta L \\ \lambda(x'\cos(-\Delta\theta) - y'\sin(-\Delta\theta)) \\ \lambda(x'\sin(-\Delta\theta) + y'\cos(-\Delta\theta)) \end{bmatrix} \tag{3.14}$$

Again, the state transition Jacobian matrix below is derived by partial differentiation of the prediction Equation 3.14 with respect to the state variables, and is used to adjust the covariance of the state estimate.

$$F = \begin{bmatrix} \frac{\delta x}{\delta x} & \frac{\delta x}{\delta y} & \frac{\delta x}{\delta x'} & \frac{\delta x}{\delta y'} \\ \frac{\delta y}{\delta x} & \frac{\delta y}{\delta y} & \frac{\delta y}{\delta x'} & \frac{\delta y}{\delta y'} \\ \frac{\delta x'}{\delta x} & \frac{\delta x'}{\delta y} & \frac{\delta x'}{\delta x'} & \frac{\delta x'}{\delta y'} \\ \frac{\delta y'}{\delta x} & \frac{\delta y'}{\delta y} & \frac{\delta y'}{\delta x'} & \frac{\delta y'}{\delta y'} \end{bmatrix} = \begin{bmatrix} \cos(\Delta\theta) & \sin(\Delta\theta) & T\cos(\Delta\theta) & T\sin(\Delta\theta) \\ -\sin(\Delta\theta) & \cos(\Delta\theta) & -T\sin(\Delta\theta) & T\cos(\Delta\theta) \\ 0 & 0 & \lambda\cos(\Delta\theta) & \lambda\sin(\Delta\theta) \\ 0 & 0 & -\lambda\sin(\Delta\theta) & \lambda\cos(\Delta\theta) \end{bmatrix} \tag{3.15}$$

Note the recurrence of the rotation matrix as both the position and velocity are robot-relative and hence were adjusted by the robot's turn. The presence of T in the upper right corner of the matrix represents the relationship between the ball's position and velocity, while the λ in the bottom right shows friction affecting only the velocity.

Correction

As with the stationary filter, the correction update for the moving filter is greatly simplified due to the transformation of the observation vector into Cartesian coordinates – comparable with the state vector. A linear Kalman Filter correction update suffices for the moving filter, with a simple observation model to extract only the x and y coordinates from the state mean vector:

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{3.16}$$

3.1.5 Filter Utilisation

Selection

With two filters tracking the position of the ball, a decision must be made regarding which filter's state estimate to select as the current best estimate. It would be logical to use the stationary filter while the ball is stationary and the moving filter while the ball is in motion, however in practice this is hard to distinguish; as the moving filter is configured to react quickly to ball motion and can be quite unstable at times.

Teh [10] experimented with an approach that combined the velocity of the moving filter with its covariance, and decided the ball was in motion if it was beyond a tuned threshold. However, the approach of this thesis examines and compares the innovation of each filter; the precise measurement of the difference between where the filter predicts the ball will be, and where the ball is actually observed to be. With a small level of hysteresis, slightly biased towards the more stable stationary filter, the filter with a smaller innovation is chosen as the dominant filter. The moving filter also cannot be dominant unless its estimated ball velocity is above a certain threshold. This simpler measurement proved to be quite reliable in indicating which filter was correct, and requires less tuning and monitoring than the previous approach which was relatively ad-hoc.

As the movement of the robot greatly affects the level of noise of the ball observations, the hysteresis for filter selection is adjusted when the robot is in motion as opposed to standing or squatting. If the robot is in motion, then the moving ball filter must have a smaller innovation over more consecutive frames before it is decided as the dominant filter. This results in a moving robot being less susceptible to false positive moving ball observations, whilst preserving the reactivity of a robot observing from a stationary position (such as the Goalie or a sideline referee).

Interaction

To further improve the accuracy of the ball tracking, the information from the dominant filter is used to update the state of the other filter. It was observed that if the ball was

moving but then came to an unexpectedly quick stop (a common occurrence when the field is heavily populated) then both filters would be inaccurate for a noticeable amount of time. The moving filter would overestimate the ball's position and have to adjust its velocity as well as shift its position. Meanwhile the stationary filter would take time to adjust to the new position of the ball.

To overcome this issue, the following filter interaction was implemented:

- If the ball is in motion (i.e. the moving filter is dominant), then at each tick the stationary filter's estimate of the ball's position is replaced with that of the moving filter. As the ball is in motion it is almost certain that the stationary filter is lagging behind, and hence this update reduces the time taken for the stationary filter to 'catch up' without much risk of sacrificing accuracy.
- If the ball is stationary (i.e. the stationary filter is dominant), then at only one point shortly after the ball was in motion, the moving filter is reset with zero velocity and the stationary estimate of the ball's position. This has the effect of clearing any lingering velocity estimate that the moving filter might have after the ball stops suddenly, which is a potential cause of inaccuracy.

Reset

Due to the dynamic nature of the game, if a robot loses sight of the ball (i.e. no observations are received by the filters) for much more than a couple of seconds, then it is unlikely that the old estimate of the ball's position will provide a good indicator of the ball's current position. To slightly improve the reactivity and convergence of the filters, but more importantly to remove misleading stale estimates of the ball's position, the filters are "reset" by having them jump to the next observation that is received. This is equivalent to vastly increasing the covariance of the state estimate, but avoids any potential initial misleading velocity estimates. For this reason the moving filter is also reset earlier than the stationary filter.

Absolute Estimate

For certain capabilities it is necessary for the robot-relative estimate of the ball's position to be combined with the estimate of the robot's state, to provide an absolute estimate of the ball's position relative to the centre of the field. Most importantly, this absolute estimate facilitates communication and collaboration between the robots, as their ball estimates become directly comparable.

The transformation from the robot-relative position of the ball to absolute coordinates is achieved by a simple rotation by the heading of the robot (θ_R), followed by a translation of the robot's position (x_R and y_R):

$$\begin{bmatrix} x_{\text{Abs}} \\ y_{\text{Abs}} \end{bmatrix} = \begin{bmatrix} \cos(\theta_R) & -\sin(\theta_R) \\ \sin(\theta_R) & \cos(\theta_R) \end{bmatrix} \begin{bmatrix} x_{RR} \\ y_{RR} \end{bmatrix} + \begin{bmatrix} x_R \\ y_R \end{bmatrix} \tag{3.17}$$

For the covariance of the new absolute estimate, the covariance of the ball's robot-relative position is combined with the covariance of the robot's position. Again, the first step is to rotate the robot relative covariance by θ_R so the two covariance estimates are using the same frame of reference, before simply adding the two. This approach ignores the effect of the robot's heading variance; however it suffices for our current needs, and a linear approximation of the effect of heading uncertainty was not pursued.

"Out by [Blue/Red]!"

During games, referees are quick to pick up balls and return them to the playing area if they are kicked out of bounds. If a robot loses sight of the ball, it is beneficial for the robots to identify if the ball was out of the playing area, and if so to quickly "forget" the old estimate and begin searching for the ball again, rather than chasing a ball that is no longer on the field.

The identification of balls that are out of play is achieved by examining the ball estimate's absolute position and covariance, and deducing whether the entire covariance ellipse is outside the playing field – which would imply with some certainty that the ball

was out of bounds or at least heading out of bounds before vision of it is lost. If this is deemed true, the timer until a "reset" of the filters is accelerated by a considerable margin.

An early attempt at out detection was experimented with, that checked that the absolute position of the ball estimate was beyond a threshold past the boundaries of the field, and that it was quite close to the robot (implying a good estimate of the ball's position). However, in practice balls are most often kicked out and the moving filter predicts the ball's position as increasingly far away from the robot. By first checking that the absolute position estimate is past the boundaries of the field, and then also using the covariance of the estimated ball position, the check becomes more versatile and remains valid for use by robots further away from the ball estimate.

To test whether the covariance ellipse is outside the bounds of the field, the intersection of the field line in question and the line between the robot and the ball estimate is first found. This is the point on the field most likely to be within the confines of the ellipse, except for rare boundary cases where the lines are roughly parallel – where the test is abandoned as it is unlikely we will be sure the ball is out. This point is then tested using the Mahalanobis distance [20] of the point to the ball position estimate and its covariance, to determine if it is sufficiently probable that the ball is out. The Mahalanobis distance is explained elsewhere in Section 4.1.3, as it is used more extensively and to greater effect.

3.1.6 Distributed Team Ball

The individual absolute ball position estimates of each robot on the team can be distributed to other members of the team, and combined to great advantage by exploiting their differing vantage points. The collective "team ball" estimate improves the efficiency of find ball behaviours by providing a starting point for robots who are just entering the game or who have lost the ball [21]. Section 4.1.4 examines how the team ball is also useful in identifying and correcting robots that have become incorrectly localised in the symmetric playing environment.

Claridge [9] describes the approach used for the transmission of information between the team members, including identifying and disregarding old or misleading data from 'incapacitated' robots (whether they have dropped off the network, been picked up, or fallen over). Teh [10] describes the previous method of each robot calculating the weighted average of the estimates it receives, which has been refined in order to provide more accurate and useful information.

As the team ball is no longer used exclusively for finding the ball once a robot has lost sight of it, the combination now also includes the robots own current estimate of the ball's position, and not just those received from others. The increased importance of the team ball has placed new emphasis on using only the timeliest of observations. This is due to the fact that the ball can move quite large distances in a short amount of time; if it is moved by a referee or kicked particularly hard from one side of the field to the other, then one robot might report an estimate based on its old (yet unexpired) position, while another robot might report a conflicting estimate based on what it can see now. In the worst case this might falsely suggest that one of the robots has "flipped" (become mislocalised and begun attacking in the opposite direction, due to the symmetrical environment). Hence the new system ignores position estimates that are not based on very recent estimates, or were received by the network too long ago.

Estimate Selection

The previous approach to combining the estimates into the team ball was characterised by using the weighted average of all the estimates. However when two estimates are far apart and in obvious disagreement, the average provides a result that disagrees with both and is almost certainly inaccurate. When such a case occurs it is more appropriate to select the more certain of the two estimates. Furthermore, with multiple estimates and possible combinations available, the team ball can be created by combining only the agreeing estimates by disregarding the outliers.

Algorithm 1 summarises the new method used to select a subset of the available valid estimates for combination into the team ball. The estimate used to create each

subset (the first to be added to it, *current*), is the main or centre estimate of that subset, and the other estimates in that set all "agree" with the main estimate. Estimates are defined as being agreeable if they are within a small absolute distance threshold of each other. An estimate is also deemed as agreeing if the main estimate is determined to be within its covariance ellipse (the Mahalanobis distance [20] is again utilised, but is explained in further detail within Section 4.1.3). The estimates must also be checked that they are not too far apart, to prevent far away observations with particularly large covariance ellipses from being included.

Algorithm 1 Selection of subset of ball estimates for combination into team ball

```
generate list of valid ball estimates (valid)
for all estimates (current) in valid :
        create candidate set for team ball (set)
        add current to set
        for all other estimates (other) :
                if other and current agree then
                        add other to set
                end if
        end for
        if set better than bestSet then
                bestSet ← set
        end if
end for
teamSet ← bestSet
```

A set of estimates is determined as being "better" than another if the set is larger than the other; that is, more robots would be agreeing on the position of the ball and contributing to the team ball. For sets of the same size, such as when two robots observe the ball but whose estimates are not close enough to be combined, then the set with a more certain main estimate is selected.

Combination and Status

Once the subset of estimates has been chosen, they are combined together using the update equations of a linear Kalman Filter to take into account each estimate's uncertainty. The main estimate of the subset is used to initialise the filter, and the other estimates are applied as observations; essentially calculating the weighted average of the estimates,

analogous to the combination approach used previously [10]. This method can easily be modified to maintain an estimate of the team ball over time by not initialising the filter with each new subset. While this might provide a smoother estimate, it was avoided as a simple combination of the most recent – already filtered – estimates was desired, that would react quickly to changes in the ball's position.

After comparing its estimate with the received estimates from its teammates, deciding upon the most sensible subset, and calculating the combined team ball, the final consideration is the status of the robot's estimate in relation to the team ball. This status is one of three possibilities:

- **Neutral**: It had no valid estimate of the ball's absolute position, either because it was incapacitated itself (and so was unsure of its own position), or had not recently observed the ball.
- **Agree**: Its valid estimate of the ball's position was used as part of the calculation of the team ball.
- **Disagree**: Its valid estimate of the ball's position was not included in the subset used for the calculation of the team ball.

This status, along with the number of contributors to the team ball (the size of the subset used) can provide a strong indicator of the status of the localisation of the robot. A robot that disagrees with the majority of its team suggests that something may have gone wrong. Section 4.1.4 describes how this information is incorporated into the localisation of the robot itself.

3.2 Results

3.2.1 Elliptical Covariance and Team Ball

To demonstrate the observation noise model and the resulting covariance ellipses of the ball position estimates (Section 3.1.2), as well the combination of these estimates into the team ball (Section 3.1.6), the following experiment was undertaken;

- Nine ball locations were selected to represent possible ball locations on the field, and to provide a variety of ball observation angles and distances for the robots. They are labelled 1 through 9 and are shown in Figure 3.4 in orange.
- Four stationary robot positions were selected to provide sufficiently different observations of each ball position, and good coverage over the field. They are similar to the positions used for robot referees in the game management system [1], and are shown in blue in Figure 3.4, denoted positions A, B, C and D.
- Three configurations of the robot positions were selected, to observe the effect of an increasingly large number and variety of ball observations on the team ball; A, AB, ABCD.
- For each configuration, the ball was placed in turn in each of its positions, and for each position a single frame of output was captured once all the robots were observing the ball – including a visual representation of each robot's ball estimate and covariance, as well as the team ball's estimate and covariance. A subset of the collected frames is shown in Figure 3.6.
- The exact coordinates of each team ball estimate were also recorded with each frame, and compared with the known position of the ball at that position for a precise measurement of the error of the team ball estimate.
- The experiment was repeated three times for each configuration. The minimum, maximum and average of the error measurements for each position and configuration are shown in Figure 3.5 (See Appendix A for precise numerical data).

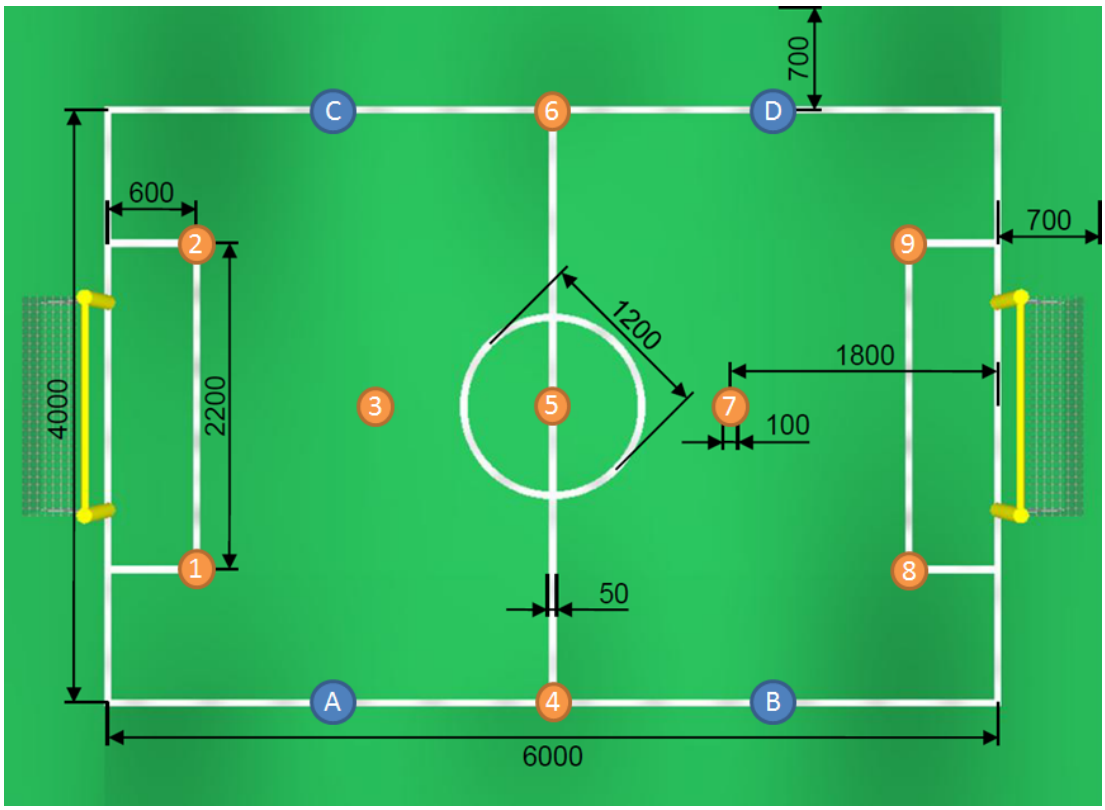


Figure 3.4: Map of SPL field showing experiment's robot and ball positions

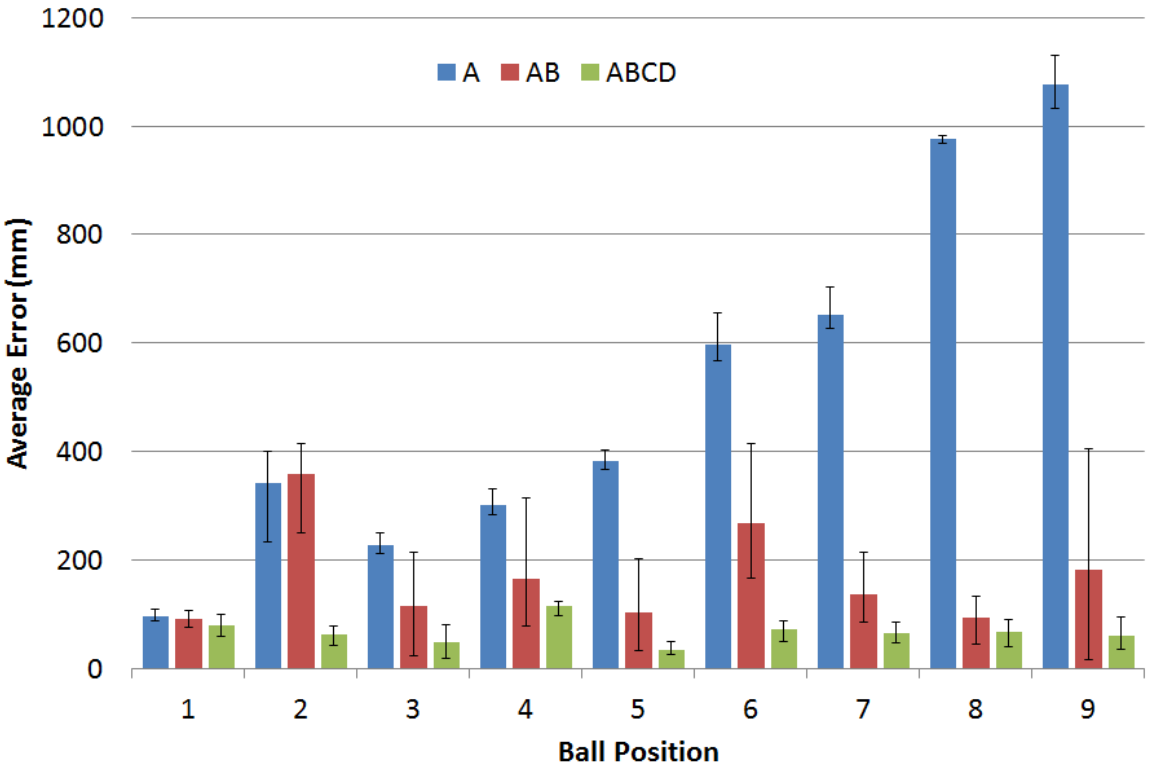
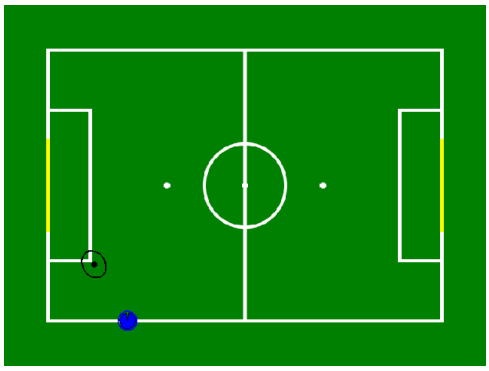
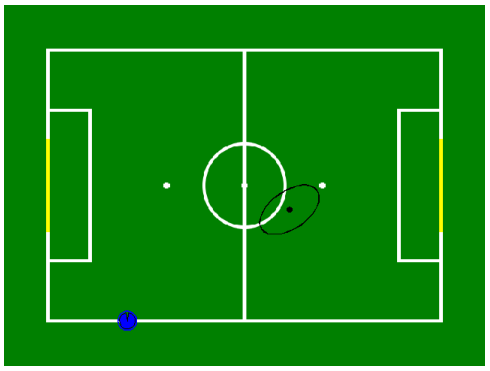


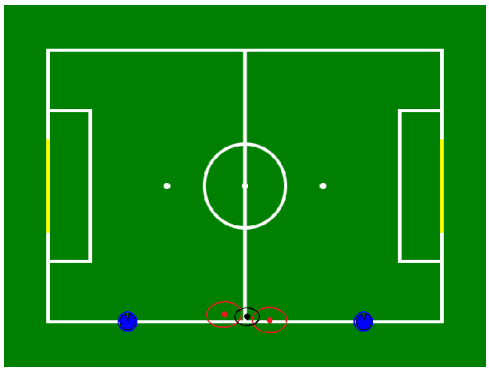
Figure 3.5: Average team ball error at variety of positions and robot configurations



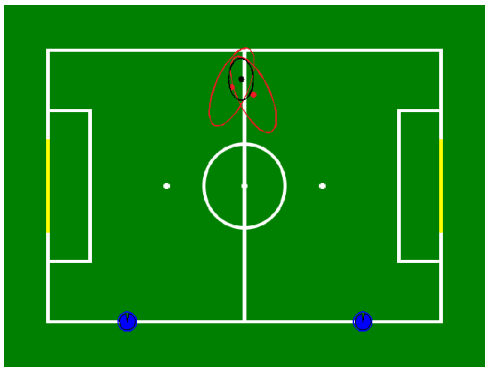
(a) Configuration A, Position 1



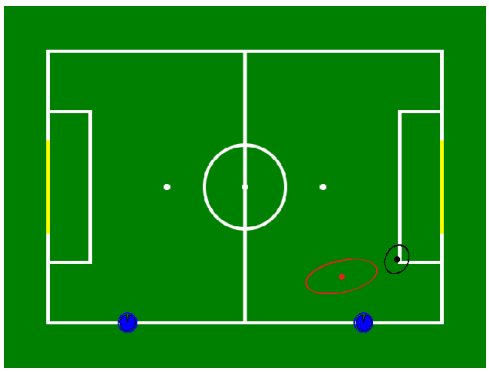
(b) Configuration A, Position 7



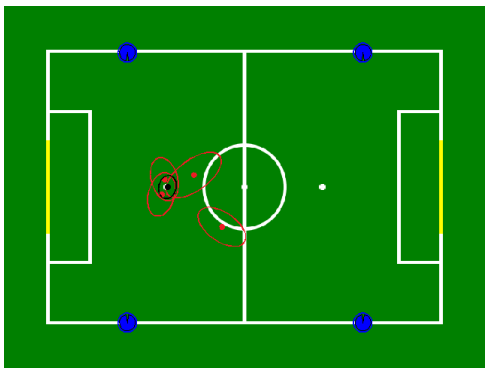
(c) Configuration AB, Position 4



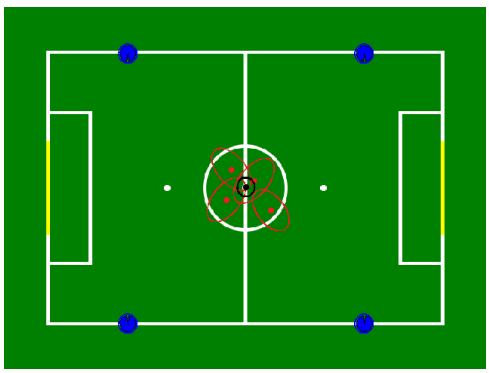
(d) Configuration AB, Position 6



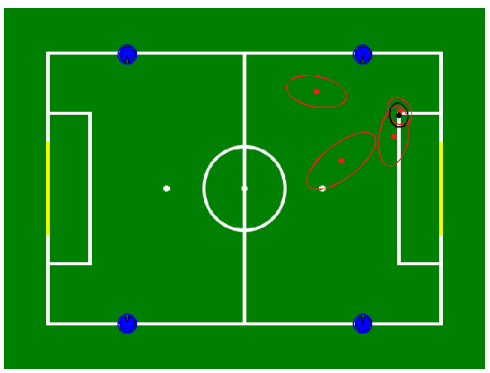
(e) Configuration AB, Position 8



(f) Configuration ABCD, Position 3



(g) Configuration ABCD, Position 5



(h) Configuration ABCD, Position 9

Figure 3.6: Experiment's different robot configurations (blue), individual ball estimates and covariance ellipses (red), and combined team ball estimates with covariance (black)

3.2.2 Kicked Ball Velocity

To illustrate the behaviour of the ball tracking system and the capability of its dual-modal filter system to track both stationary and moving balls, its reaction to changes in ball position and velocity under certain common and important scenarios were tested as examples.

For each test, the position and velocity of the ball estimate at each tick were recorded, and then plotted to provide comprehensive diagrams showing the progression of the estimate. The position of the ball estimate at each tick is represented by a small blue dot, while a red line extending from a blue circle represents the velocity vector of that particular ball estimate. Thus the direction of the red line away from its associated blue dot shows the ball's estimated trajectory, while the magnitude of the red line shows its speed.

These diagrams can then be compared with the actual ball movement to provide an indication on the accuracy and correctness of the estimate. This process closely mimics the method of implementation and relative observation used to develop the system, as precise measurements of a moving ball's position and velocity to be used as a comparison are difficult to obtain reliably and efficiently.

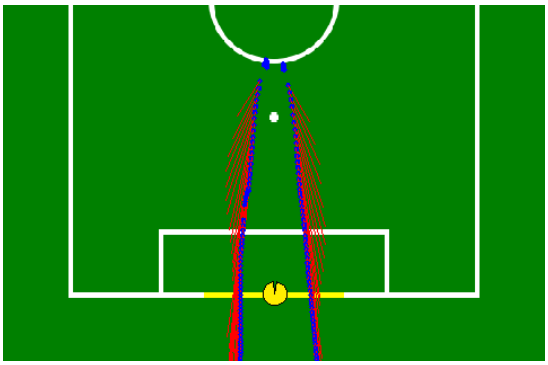
One of the most important uses of the tracked position and velocity of the ball are within the Goalie's behaviours: dictating if and when to dive, and in what direction. Hence the first test was to place a ball some distance away (on the centre circle) from a robot stationary at the goal line, and then kick it in various directions towards the robot:

- Figure 3.7a shows examples of the output from the kicks that went past the robot to the left and right, scoring goals.
- Figure 3.7b shows kicks that missed the goal by a small amount on the left and right, going out.
- Figure 3.7c shows a kick that was aimed towards the robot, resulting in a successful block and the ball rebounding slightly.

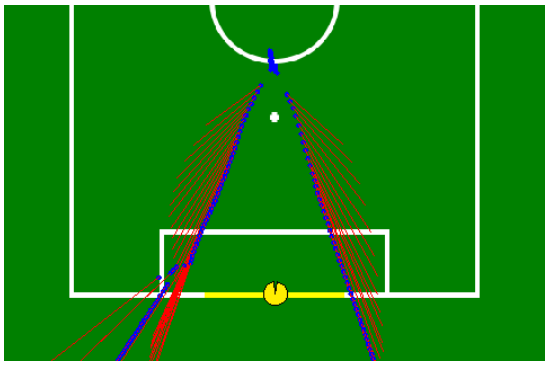
- As the abundance of data may be hard to interpret, Figure 3.7d shows single sample data of the estimated position and velocity, once the velocity had converged, for each of the above kicks.

To more clearly illustrate the effect of speed on the ball estimate's velocity, balls were kicked to the left and right of a stationary robot at different speeds. Figure 3.8a shows the balls that moved quite fast, while Figure 3.8b shows the balls that were kicked much more softly, moving slowly and eventually coming to a stop before the robot.

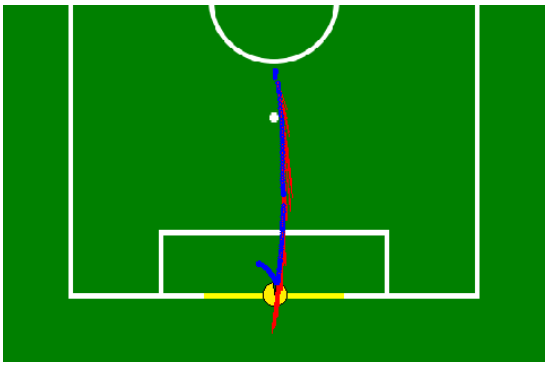
Also of significance is the ability of the ball tracking system to be robust against the noise associated with a moving robot – avoiding overestimation of the noise as a ball velocity. Figure 3.9 shows two examples of a very common and important scenario in RoboCup. The ball estimate is depicted as a robot approaches a stationary ball (including shifting the estimate of its own position), adjusts the position of its feet to line-up the ball, then kicks the ball and tracks its progression into the goal.



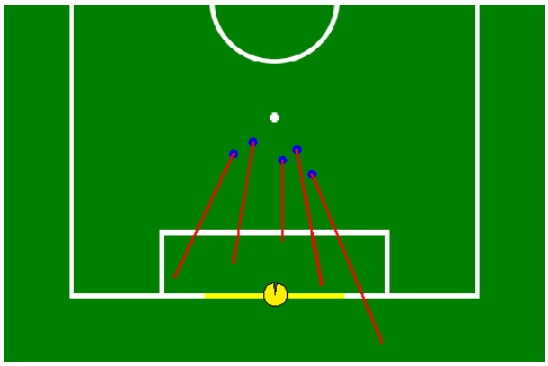
(a) Goals



(b) Near Misses

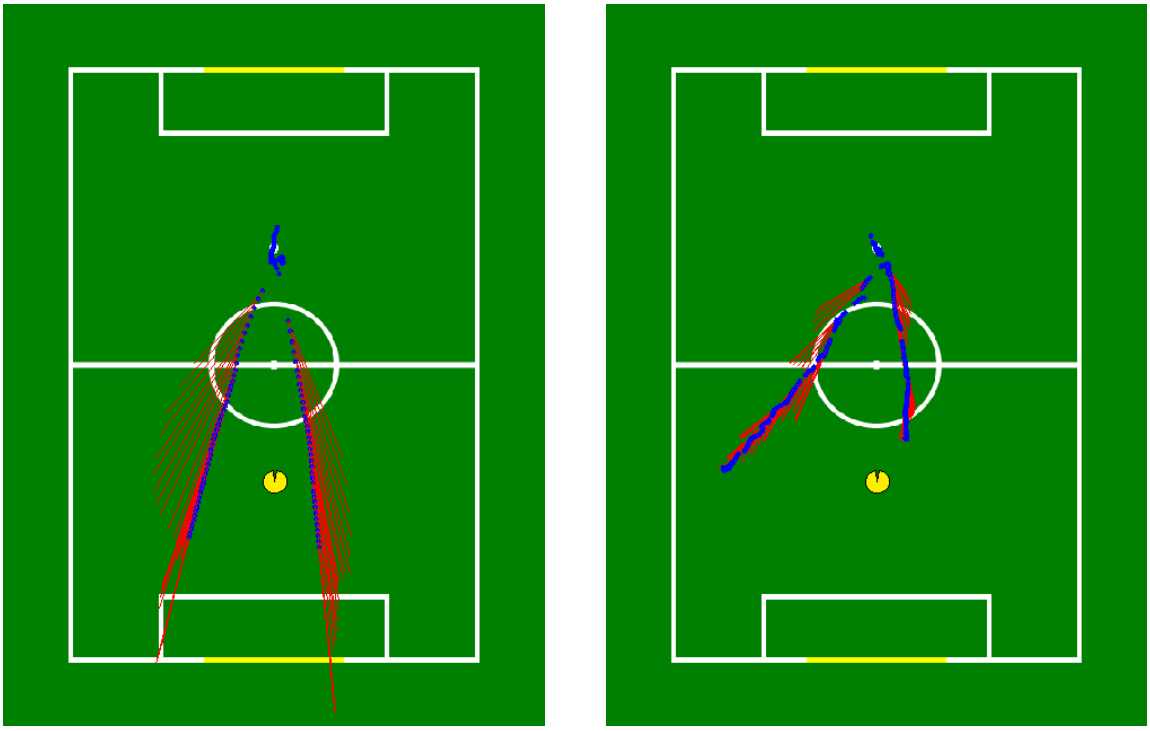


(c) Block



(d) Samples

Figure 3.7: Example ball position (blue) and velocity (red) estimates of stationary robot tracking five goal attempts



(a) Fast Kicks (b) Slow Kicks

Figure 3.8: Example ball position (blue) and velocity (red) estimates of stationary robot tracking balls moving with different speeds

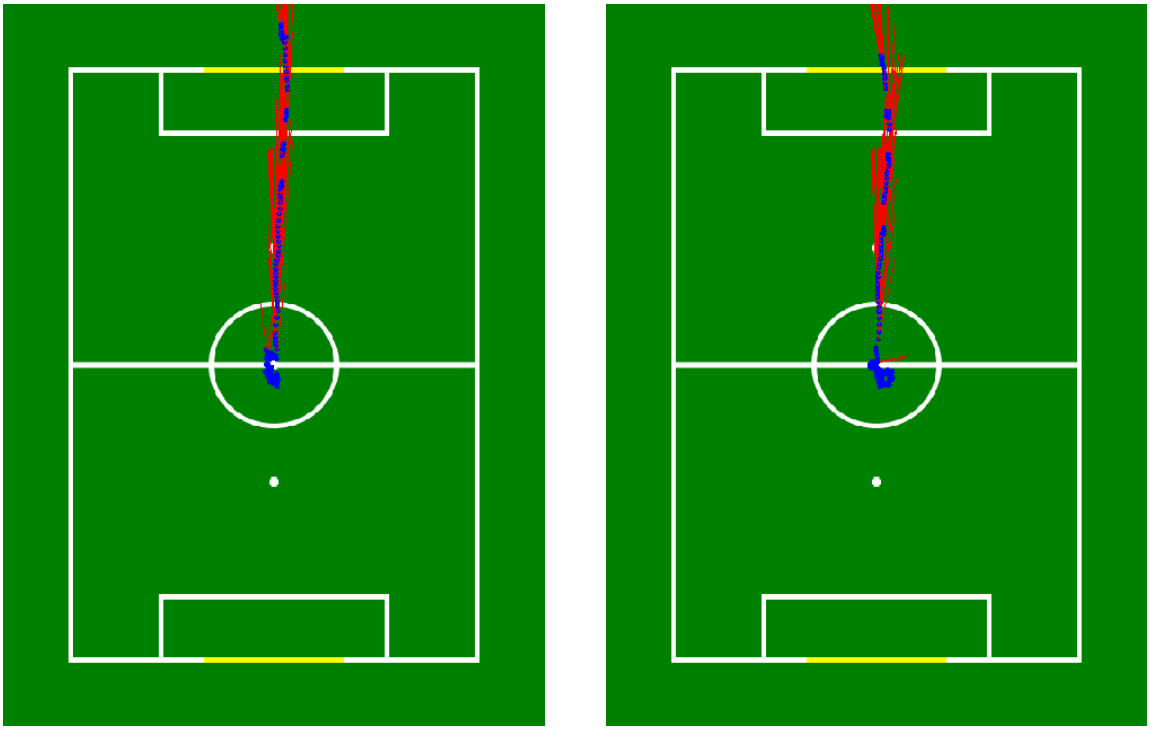


Figure 3.9: Example ball position (blue) and velocity (red) estimates of attacking robot approaching ball and successfully scoring a goal

3.3 Discussion

The introduction of the model of elliptical observation covariance (Section 3.1.2), combined with the more detailed process and observation models of the two new Extended Kalman filters (Sections 3.1.3 and 3.1.4), has improved the accuracy of the resulting covariance matrix (estimate of uncertainty) of the localisation system's filtered ball estimate. This is evidenced by the visualisations of ball estimates from the individual robots of the experiment outlined in Section 3.2.1.

Figure 3.6 showcases a number of different ball estimates, resulting from balls placed relative to robots at a variety of distances and angles. Comparing Figures 3.6a and 3.6b, it is clear that the uncertainty of the ball estimate (size of the covariance ellipse) is larger for the ball that is further away. The covariance ellipses are also at different angles, oriented in such a way that the longer axis points towards the robot. These properties of the covariance representations more accurately reflect the true nature of the uncertainty of the robots' ball observations than the previous system [9, 10]. They accurately portray the findings that visual detection of angles to the ball is more reliable than estimations of the distances to the detected balls (also apparent in Figure 3.6).

The Kalman Filter method is heavily influenced by the accuracy of the estimated covariance of the process and observation noises [6, 7, 11]. It follows that improving the estimate of the noise associated with the robot's observation inputs used would improve the effectiveness of the filters that are used. Consistent with the logic of the Kalman Filter, it was found that the new ball tracking system provided a more stable and consistently accurate estimate of the ball's position.

In particular, slight discrepancies in the observed distance to the ball were of little consequence to the estimate, as these were weighted less. Furthermore, false positive ball observations that are occasionally seen in the background and calculated as being relatively far away, have less of a detrimental impact on the ball estimate – especially in the presence of consistent closer ball observations. Most notable however, is the improved responsiveness of the filter to changes in the (more reliable) observed heading to the ball, as the new models allow for them to be weighted more heavily than the noisy distance

observation. These improvements were observed throughout the development and tuning of the new localisation system, by assessing the behaviour of the ball estimate using OffNao; a rUNSWift visualisation tool that enables a comparison of the state estimate and the observed classified camera images over time.

The output of the OffNao tool over time was replicated to a certain extent in the Figures of Section 3.2.2. Figure 3.7, shows the dual-modal filter accurately estimating the stationary ball's position, before appropriately switching to the moving filter once the ball's velocity has become apparent; with the slight gap in the strings of blue dots marking these occurrences. Also shown is the correct and accurate estimate of the ball's velocity for each of the five trials, further evidenced by Figure 3.8. As well as the stability of the estimate of the stationary ball as the robot walks towards it (Figure 3.9). These results reflect well on the dual-modal system and the approach used for the selection and switching between the filters, described in Section 3.1.5.

Unfortunately the images themselves do not adequately convey the passage of time and exact progress of the filter with respect to the actual position of the ball, without OffNao's accompanying camera stream. However the reaction speed of the Goalie [17] and minimal time required for the velocity to converge on an accurate estimate, provides testament to the reactivity of the ball tracking system. In Figure 3.7c the position of the ball estimate is seen following the progress of the ball into the feet, and then back out again with the rebounding of the actual ball – without the velocity filter following through and overestimating the ball's position. This provides evidence for the interaction between the two filters (also described in Section 3.1.5) improving the reactivity of the filter's modes.

Returning to the effect of the elliptical covariance of the resulting ball estimates, it also facilitated a more accurate calculation of the "team ball" estimate; a combination of a robot's estimate and the distributed estimates it receives from its teammates, described in Section 3.1.6. The elliptical covariance estimates are associated with a greater weighting towards the observations of angles to the ball, which allows for essentially a more accurate triangulation of the team ball. This triangulation is apparent in Figure 3.6d, where the combination of two robots' ball estimates has resulted in a team ball more accurate

than either individual estimate. The resultant team ball is also more accurate than a combination of the two estimates using the previous system's circular covariance matrices.

However Figure 3.6e illustrates the dangers of combining robot estimates of the ball, and the usefulness of the estimate selection approach introduced by this thesis in Section 3.1.6. In this case, combining the estimates would provide a more accurate estimate than robot A's single estimate, but less accurate than B's. Instead, the selection algorithm has decided not to combine the observations as they are significantly different, and selected the more certain observation as the team ball. Figure 3.6h shows a further example of particularly inaccurate ball observations being excluded from the calculation, and how the resulting team ball (a combination of the two selected observations) remains far more accurate than a combination of all four observations.

Figure 3.5 summarises the experiment detailed in Section 3.2.1 and showcased in Figure 3.6, providing evidence supporting the accuracy of the combined team ball using only a subset of the team's estimates. For the single robot case the team ball is simply equal to the robot's own estimate, and as established earlier, the error in the estimate grows rapidly with increasing distance from the robot. The graph shows that as the number of robots observing the field increases to two and then four, the coverage of the field and hence diversity of available ball estimates grows – drastically reducing the error of the combined team ball down to considerably accurate levels. The estimate selection allowed for beneficial combinations of the observations, but minimised the effect of poor observations on the already accurate positions.

The benefit of selectively combining ball estimates does not just apply to minimising the impact of poor observations. To combine the ball estimates of each robot, the robot must first be aware of its own position and heading. While in the experiment the robots were stationary and their poses hard-coded, but in reality during matches this is not the case. The selection of estimates to be used for combination into the team ball thus also helps to minimise the detrimental effect of mislocalised robots.

Furthermore, the selective use of the ball estimates for the team ball lead to an effective method of determining a robot's heading in the symmetrical environment

introduced this year. The advancements of the pose tracking system has led to a relatively stable and accurate estimate of the robot's pose (see Chapter 4). When the heading of the robot occasionally does become wrong, it is common for the position estimate to remain accurate, but merely "flipped" to the opposite side. Hence, a fairly reliable method of determining if the robot is flipped emerged. If such a mislocalised robot disagrees with its teammates on the location of the ball, the team ball estimate remains accurate as a result of the selective approach. If this team ball corresponds to the opposite of the robot's own ball estimate, it suggests that the robot is flipped.

The robustness of this suggestion is improved by the measures used to ensure only the most timely of ball estimates are used for calculation of the team ball (Section 3.1.6). As such, this mechanism would only be triggered falsely in the very rare circumstance of two robots seeing the ball in one position, while a third also observes a second ball in roughly the same position but on the opposite side of the field. Unfortunately however, to avoid a single flipped robot flipping the rest of the team, this method is only triggered when a robot disagrees with two or more teammates. This implies that the mechanism can only be used when at least three robots are on the field, localised (with one flipped), and with a direct line of sight to the ball.

Chapter 4

Pose Tracking

This chapter will examine and analyse the system used to track the robot's own position and heading, including the methods used to resolve the robot's heading in the now symmetric playing field.

4.1 Method

The new approach used this year was briefly introduced in Section 2.7.2, including a comparison with some of the methods used in the past. The localisation system has been almost entirely re-implemented, with a focus on utilising the improved detection of visual features to its fullest potential to provide a more accurate and stable estimate of the robot's position.

Section 4.1.1 describes the module used to combine all the observed features into a single estimate of the robot's position and heading. This simplification of the observations to be filtered has facilitated the use of a much simpler system of two Extended Kalman Filters, detailed in Sections 4.1.2 and 4.1.3. The methods applied to address the challenges of the symmetrical environment are explained in Section 4.1.4.

4.1.1 Single Observation Using Iterative Closest Point

Over the years the rUNSWift vision system has developed to provide data on observed field edges, goal posts, the centre circle, individual field lines as well as valuable field line constructs such as corners, parallel lines, and T-intersections [22]. Previous approaches [2, 8, 9] to robot localisation have been characterised by the use of these field features as separate observations, occasionally combining two features into a more precise estimate – such as using a pair of goal posts or a goal post and a nearby T-intersection.

With the increase in the types, quantity, availability, and accuracy of these individual features, it has become increasingly beneficial to escalate the concept of combining features. By using our knowledge of the field layout we are able to deduce an estimate of the robots position and heading that incorporates the entirety of the field feature data available from the current camera frames.

A modified Iterative Closest Point matching algorithm was developed with teammate Peter Anderson to achieve this precise goal, detailed in a separate report [16]. The algorithm incorporates all the observed features, as well as the current estimate of the robot's pose to provide a single estimate of the most likely position and heading of the robot. Figure 4.1 shows the algorithm successfully combining observations of a variety of field features into a single accurate estimate of the robot's position. While complex in its own right, this matching algorithm and the single estimate it provides has considerable positive impact on the requirements of the filter system required for pose localisation.

Firstly, the creation of the single estimate isolates a pose filter from the variety of possible field feature observations, replacing the need for any of the complex geometric "manual linearisation methods" used previously by Claridge [9]. This abstraction also replaces the need for any non-linear extensions to the Kalman Filter's correction update – as the combined estimate is directly comparable to the tracked state of the filter (the robot's position and heading).

Furthermore, the combined estimate drastically reduces the need for a multi-modal approach, as the single estimate is inherently unimodal. This is in stark comparison with

using observations individually; some of which, such as the single observation of a corner, may provide multiple pose hypotheses.

The matching algorithm is also capable of ignoring false positive or inaccurate feature data, especially when other features are strongly correlated. In the case of strongly incoherent or inconclusive data (such as when a camera frame is subject to a large amount of noise), the algorithm can identify such noise and choose not to provide any potentially confusing estimate to the filter. Estimates that are provided to the filter by the matcher are also associated with the number of points that were used to generate the estimate. Information regarding the types of features incorporated is also available, which provide valuable information on the certainty/reliability of the estimate. The use of this certainty information, and the estimate itself, is described in more detail in the following sections.

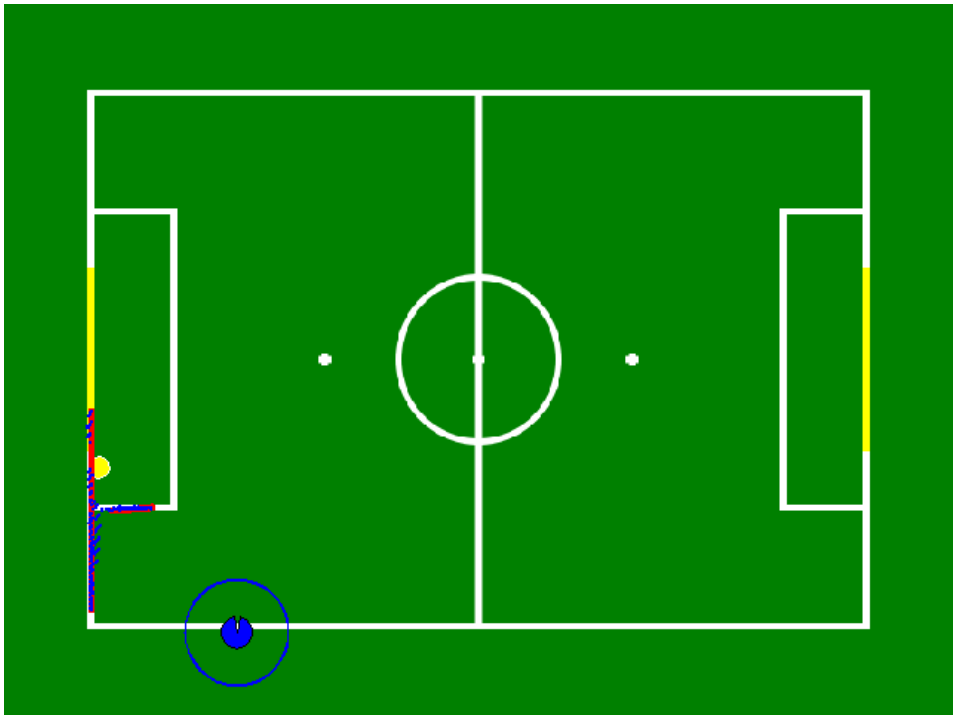


Figure 4.1: Successful robot pose estimation (blue) using observations of two field line fragments (red), a goal post (yellow), and a field edge (not visible).

4.1.2 Extended Kalman Filter Approach

A system of two Extended Kalman Filters (introduced in Section 2.3) are used to filter the estimates provided by the feature matching algorithm and track the robot's pose over time. The two filters are identical in their implementation, only differing in their purpose, maintenance and use within the system – further detailed in Section 4.1.3. This section examines the details the of the filter mechanics used in the system.

The state vector of the filter used represents the robot's absolute Cartesian coordinates and heading (θ), using the rUNSWift absolute field coordinates established in 2010 [2]. As usual the covariance matrix represents the degree of uncertainty associated with the state estimate.

$$\hat{x} = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \qquad P = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} & \sigma_{x\theta} \\ \sigma_{yx} & \sigma_y^2 & \sigma_{y\theta} \\ \sigma_{\theta x} & \sigma_{\theta y} & \sigma_\theta^2 \end{bmatrix} \tag{4.1}$$

Prediction

Similar to the ball filters described in Chapter 3, odometry data is used as input to the prediction update of the robot pose filter (see Equation 3.1). The forward, left and turn variables are directly applied to the estimate of the robot's pose using the state transition equations shown below. Again, the convention dictates that the robot's rotation be applied before the forward and left components.

$$\begin{bmatrix} f(x,u) \\ f(y,u) \\ f(\theta,u) \end{bmatrix} = \begin{bmatrix} x_{\text{new}} \\ y_{\text{new}} \\ \theta_{\text{new}} \end{bmatrix} = \begin{bmatrix} x + \Delta F \cos(\theta + \Delta\theta) - \Delta L \sin(\theta + \Delta\theta) \\ y + \Delta F \sin(\theta + \Delta\theta) + \Delta L \cos(\theta + \Delta\theta) \\ \theta + \Delta\theta \end{bmatrix} \tag{4.2}$$

The below state transition Jacobian matrix was derived in accordance with the EKF approach, used to approximate the relationship and covariance between the state estimate variables, as well as the effect of the odometry on the covariance matrix.

$$F = \begin{bmatrix} \frac{\delta x}{\delta x} & \frac{\delta x}{\delta y} & \frac{\delta x}{\delta \theta} \\ \frac{\delta y}{\delta x} & \frac{\delta y}{\delta y} & \frac{\delta y}{\delta \theta} \\ \frac{\delta \theta}{\delta x} & \frac{\delta \theta}{\delta y} & \frac{\delta \theta}{\delta \theta} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -|\Delta F| \sin(\theta + \Delta\theta) - |\Delta L| \cos(\theta + \Delta\theta) \\ 0 & 1 & |\Delta F| \cos(\theta + \Delta\theta) - |\Delta L| \sin(\theta + \Delta\theta) \\ 0 & 0 & 1 \end{bmatrix} \tag{4.3}$$

The effect of this state transition Jacobian are subtle improvements to the adjustment of the covariance matrix over time, most apparent when visual observations are scarce. These adjustments include increasingly scaling the variance of the robot's position as the variance of the robot's heading increases – if the robot is unsure of which direction it is moving, it becomes less and less certain of its position on the field.

The absolute values surrounding the forward and turn odometry variables in the state transition Jacobian above (Equation 4.3) were added after experimentation found that without them, the Jacobian incorrectly reduced the uncertainty of the robot's position as it moved backwards. Intuitively, the absolute values are used to isolate the magnitude of the robot's movement and rectify this issue.

Experiments were also undertaken to examine the effect of formalising the relationship between the covariance of the odometry vector, and that of the state estimate, to provide a more accurate model of the filter's process noise. A process noise Jacobian was derived by taking the partial derivatives of the noisy state transition equations (slight generalisations of those found in Equation 4.2, which included the noise of the odometry variables) with respect to each of the noise variables.

However in practice when this Jacobian is evaluated and applied to the covariance matrix of the odometry noise, only an incredibly marginal effect is observed and effectively the same covariance matrix is added as process noise to the state estimate's covariance. This is due to the nature of the estimated odometry noise; the estimated variances of the forward and left variables are symmetric and are not scaled or adjusted with magnitude, while the estimated variance of the turn variable is quite small when calculated per-cycle. As such, evaluation of the noise Jacobian was deemed unnecessary and the code has been left unused – to be re-evaluated for use if improvements to the odometry noise model are

made.

For now, the common assumption of additive process noise has been made; and the system uses a simple approach of estimating the odometry/process noise and directly adding it to the state estimate covariance. This simple additive approach was partly refined by varying the amount of process noise that is added; a small amount when the robot is stationary (still accounting for the possibility of external forces sliding the robot forward), and a larger addition if the robot is in motion. The result is a simple and effective process noise model that adequately and reasonably increases the uncertainty of the state estimate over time, further enhanced by the addition of the state transition Jacobian.

Correction

The combined "observations" provided by the feature matching algorithm deliberately mimic the representation of the state estimate – the position and heading of the robot in absolute terms (Equation 4.1). This allows for a simple linear Kalman Filter correction update, using the identity matrix as the observation model. This state representation is deliberately mimicked by the estimates provided by the feature matching algorithm to streamline the filtering of the observations.

A simple model of observation noise accompanies the estimates provided by the feature matcher. The variance of each variable in the estimate is estimated as either small or large, depending on the number and type of visual features that are used within the matcher to provide the estimate. An estimate derived from only a single point feature (such as a singular goal post, or the centre circle) is used with a large heading, x and y variance. Estimates derived from two or more points, such as observations of line features or multiple goal posts are assumed to have a small heading variance. The x and y variance are reactive to the types of line features; matches to vertical or horizontal lines reducing the x or y variance respectively. Furthermore, matches to multiple field lines, or field lines and single points, reduce the estimate of both the x and y variance, as these combinations allow for the robot's position to be estimated more precisely.

4.1.3 Dual-Modal: Providing an Alternative

The Extended Kalman Filter implementation detailed above in Section 4.1.2 successfully tracks the robot's pose by filtering the estimates provided by the observed field feature matcher. However a single instance of the filter does not sufficiently overcome all of the difficulties posed by the RoboCup SPL. The single unimodal filter is still susceptible to error caused by false positive information, which may occasional slip through the feature matching algorithm. Nor is the single filter satisfactorily capable of solving the kidnapped robot problem; localising itself with no prior knowledge, or a possibly misleading prior estimate – a common occurrence due to referee interference during SPL matches.

To simply yet effectively overcome these challenges, a dual-modal approach using two separate instances of the filter has been implemented. The approach focuses on the provision of a relatively stable estimate of the robot's pose by selectively applying observations to the filters with careful logical management. The first and most important filter is aptly referred to as the "main" mode, and tracks the current best estimate of the robot's state. The main mode assumes its current estimate is approximately correct, and hence disregards observation estimates that are too far away from its current estimate to be reasonable (such as those caused by false or highly inaccurate data).

The second filter, dubbed the "alternate" mode, captures the observation estimates that are discarded from the main mode. In doing so, it tracks a possible robot pose alternative to the estimate provided by the main mode. This alternate pose allows for the "kidnaps" of the robot that involve large sudden shifts in the robot's position, as well as for recovery of the robot's true pose if errors have compounded in the main mode. A 'jump" occurs only if a consistent stream of reliable and coherent information strongly suggests the alternate mode is correct. The jump occurs by replacing the main mode with the alternate's estimate.

The following sub-sections describe more concrete details of the dual-modal system, including: the measures used to determine if an observation is "too far" from a state estimate to be applied, the algorithms used for the management of the two filters, as well as the initialisation and reset process used for the system.

Mahalanobis Distance

One key aspect of the implemented dual-modal approach is the screening of observations that are too far from the current estimate to be feasible. The identification of observation estimates that are close enough to be considered valid uses two methods. The first simply checks if the absolute difference between the two estimates' positions and heading lie beneath small defined thresholds. This method ensures that estimates within some small area around the current estimate are valid by default.

The second method uses a metric known as the Mahalanobis distance, introduced by its namesake Prasanta Mahalanobis [20]. Equation 4.4 below is used to calculate the Mahalanobis distance between a vector x and a comparable vector μ with a covariance matrix S:

$$D_M(x) = \sqrt{(x-\mu)^T S^{-1}(x-\mu)} \tag{4.4}$$

The resulting Mahalanobis distance between the two vectors is based on μ's covariance matrix. A Mahalanobis distance less than 1 implies the estimate x is within the boundaries of μ's covariance matrix, while greater than 1 implies the opposite. Essentially a μ vector with a higher uncertainty will be "closer" to points around it than a μ with a smaller uncertainty when using this metric.

For an observed state estimate provided by the feature matcher to be considered valid, the Mahalanobis distance between the current state estimate and the observed estimate must be below a defined threshold. Note that this incorporates the covariance of the observed estimate, not that of the current state estimate. Both methods were investigated, and minimum and maximum values of the two were also considered. However, the covariance of the filter's current estimate varied greatly depending on the availability of observation data. This meant that the area that was considered valid would range from unreasonably small to far too large. As such, the observed state estimate was used as μ for the calculations, with its controlled covariance remaining practical for the purposes of identifying far away observations.

The Mahalanobis distance described here is also utilised within the ball tracking

system; for determining if our tracked ball estimate suggests it is outside the play area (Section 3.1.5), and for examining distances between the team's absolute ball observations (Section 3.1.6).

Filter Management

Algorithm 2 provides a summary of the process used to improve stability of the current state estimate, by selectively applying each observed estimate of the robot's pose to either the main or alternate filter/mode. The algorithm also briefly outlines the conditions required for the resetting and replacement of the alternate mode; being cleared when the main mode is verified by a very reliable observation (generated with a large number of observed features), replacing the main mode when it has proven itself, and being replaced by a new observation if it shows little promise.

Algorithm 2 Observation application, filter replacement and reset

```
{observation estimate received from feature matcher}
if observation close to main then
        apply observation to main
        if observation very reliable then
                reset alternate
        end if
else {observation too far from main}
        if observation heading is certain then
                {special consideration – see Algorithm 3, Section 4.1.4}
        else if alternate empty then
                set alternate to observation
        else if observation close to alternate then
                apply observation to alternate
                strengthen alternate
                if alternate very strong then
                        replace main with alternate ("jump")
                end if
        else {observation also too far from alternate}
                weaken alternate
                if alternate very weak then
                        replace alternate with observation
                end if
        end if
end if
```

One additional restriction on the alternate filter, not shown in Algorithm 2, is its timeout. If the alternative mode is not reinforced with an observation for some amount of time, then it is reset. This timeout minimises the effect of rare but consistent false estimates, as it prevents the alternate mode from building up slowly over time and eventually replacing the main mode.

Also of importance to the dual-modal system is the idea of the robot being "unsure" of its position. A robot becomes unsure of its position when it detects it has been picked up by a referee, or when it has fallen over. The effect of being unsure prevents observation estimates that were derived from a single post or line segment from being used to update the state estimate. This is because these observations are ambiguous, especially when the heading provided to the feature matcher may have been wrong. For example, if a robot is pushed during a getup routine, resulting in it facing a horizontal line rather than a vertical one, then if the field feature matcher is only provided one line; it would reasonably yet mistakenly match this observed line to the vertical line it was previously facing! Hence, the unsure status of the robot remains in effect, blocking ambiguous observation updates until a very reliable estimate confirms its location.

Filter Initialisation

Upon initialisation of the filters, when a robot begins playing (whether at the start of the game, or recovering from a penalty), it has no accurate prior estimate of its current position with which to begin its pose tracking using the aforementioned filter management and selective observation application. At this initial point the robot is considered "lost", and first establishes its current position before resuming play.

While lost, the two filters of the system serve a different purpose to the usual main/alternate relationship. Each filter is initialised with one possible starting position of the robot with a relatively large variance. For usual normal entry to play, these positions are on each sideline of the home half, facing inwards, seen in Figure 4.2. However when a robot is "manually placed" the initialisation positions are near the left and right corners of the goal box, facing forwards.

Initialised with their separate starting positions, each filter is run independently. As the robot scans its head, each filter requests an observation estimate from the field feature matcher using the observed features and their own current estimate – the result being two observation estimates using the same visual features but from the two different initial beliefs. The match rules also dictate that a robot entering play must enter from its own half, helping to restrict observation estimates to the friendly half.

The two supplied observation estimates are instrumental in quickly localising the robot and allowing it to enter the game. Quite reliably, when the robot observes a large amount of visual features, the observation estimates are very similar – and it is assumed that when both initial different positions seem to "agree", that a position is the current estimate. It is accepted as such and the filters transition out of the lost state.

A more complex decision making process was also implemented, however the agreement method just described is very effective and reliable, often localising the robot before the more complex method makes its decision. Despite this, it has been left as a part of the system as a backup mechanism. The decision making process assesses the two filter's progress over time, weighting towards one or the other if: one filter's variance is significantly smaller than the other; one filter's update is too far from it but fits with the other filter's estimate; or the distance between each filter and its observation is significantly smaller for one of the filters.

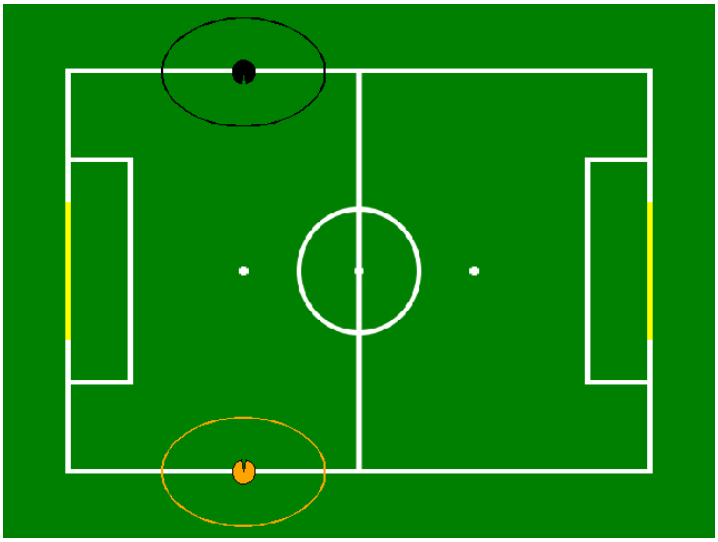


Figure 4.2: Initialised filter positions, expecting but not requiring sideline entry

4.1.4 Coping With A Symmetrical Environment

As outlined in Section 1.2, one of the primary aims of the innovations of this year's localisation system was to overcome the difficulties arising from the introduction of uniform goal colours creating a symmetrical playing field. The general approach implemented towards achieving this goal involved incorporating methods of determining the robot's absolute heading with certainty into the formation of the robot pose estimate observations.

The observed field feature matcher, briefly described in Section 4.1.1 (with further detail provided by Anderson [16]), uses the field geometry and a modified Iterative Closest Point algorithm to narrow the possibilities of the robot's current pose. Once the hypotheses have been generated based on the observed features, the matcher uses the robot's current position estimate to identify which hypothesis is most likely – supplying it as the combined feature estimate of the robot pose to the filter system.

Often when field feature data is in abundance or conclusive, the matcher will result in two hypotheses; the correct position, and its symmetric equivalent on the opposing side with a mistaken heading. However, before examining the current estimate to deduce which is correct, two approaches are used to resolve the ambiguity in the robot's heading;

- **Team Ball**: The combined team estimate of the ball's position, detailed in Section 3.1.6, is used as an indicator to the heading of the robot when at least two other teammates agree on a position of the ball that clashes with the robot's own observation of the ball (a team ball status of Disagree). Two possible absolute positions of the ball are calculated using the robot's filtered robot-relative estimate and the two symmetric estimates of the robot's current pose. If one absolute position of the ball fits well with the team ball estimate, and is significantly closer to the team ball than the other absolute ball estimate, then this is used as evidence that the associated robot pose has the correct heading.
- **Natural Landmarks**: Introduced by Anderson [16], observed visual features ("natural landmarks") of the environment surrounding the playing area are com-

pared with stored feature data associated of the two ends of the field – observed at the commencement of each half. If the observed features strongly correlate with one end, this implies that the robot's camera is pointed towards that end and from this the robot's heading can be extrapolated and the appropriate pose hypothesis chosen.

When one of the above methods is used to validate the robot's heading, then the estimate provided by the feature matcher is given special consideration in order for the robot to avoid discarding this vital information, as well as to react to it as quickly as possible to minimise the amount of time the robot spends "flipped".

The implementation of the special consideration given to these observations is outlined in the following Algorithm 3, with Algorithm 2 of Section 4.1.3 illustrating its place within the broader application of observations. Simplified, the algorithm describes how if such an observation estimate agrees with the position (but not necessarily the heading) of the mode of one of the system's filters, then this mode's heading is corrected and its estimate becomes the new current estimate.

Algorithm 3 Special consideration for observations with a verified heading

```
{observation estimate received from feature matcher, with verified heading}
{observation not applied directly to main mode}
if observation heading different to current heading, or unsure of current heading then
    calculate flipped main mode {opposite of current main}
    calculate flipped alternate mode {opposite of current alternate}
    trustAlternate ← observation very reliable, or alternate already strong
    if observation close to flipped main then
        replace main with flipped main
    else if observation close to alternate, and trustAlternate then
        replace main with alternate
    else if observation close to flipped alternate, and trustAlternate then
        replace main with flipped alternate
    else if observation very reliable then
        replace main with observation
    end if
end if
```

4.1.5 Applied Localisation: Dynamic Ready Skill

One of the most obvious and telling examples of the localisation system's pose tracking is the team's behaviour during the "Ready" state of the game. In this state, each team of robots has 45 seconds to strategically position themselves in their half of the field, in anticipation of the commencement of the next stage of play. The rules define legal positions for the attacking and defending teams, and failure to reach legal positions results in the significant penalty of "manual placement". Hence, the "Ready Skill" of a team is an important exercise in the speed, reliability and accuracy of the localisation system.

To highlight the improvements to the pose tracking system, a dynamic approach to the Ready Skill was implemented. In such an approach, the kick-off positions are strategically assigned to each robot using the entire team's estimated positions – a stark contrast to statically assigning each robot's kick-off positions for the duration of the match. However this does not include the Goalie position, as the rules dictate that player number one is always assigned to that role.

Two main aims of the dynamic approach emerged, to provide the most significant improvement over the static assignment. The first is to always prioritise the "striker" position; the position closest to the ball and obviously the most important. With static assignment, if the robot assigned to that position is incapacitated, it will go unfilled even if other robots are available. The second aim of the strategic assignment is to attempt to minimise the amount of walking required from the robots – minimising the time required and also maximising the stability and chance of success.

The challenges faced by the dynamic approach included: possibly quite large shifts in the estimates of each robot's position; the entrance and exit of teammates at any time from crashes or penalties; and the presence of obstacles or other errors that cause some robots to move slower than the others.

With these factors in mind, an algorithm for the assignment of positions was developed, to be run on each robot individually at every time step (constantly re-evaluating

the assignments), using the most recent information available regarding the position of each team member. Algorithm 4 provides a summary of the approach used, and Figure 4.3 provides an example of its application.

Algorithm 4 Ready skill dynamic position assignment

```
assign closest player to striker (position 2)
supporter1 ← closest available player to position 3
supporter2 ← closest available player to position 4
if supporter1 = supporter2 then
        {same player closest to both positions}
        if supporter1 closer to position 3 then
                assign supporter1 to position 3
                assign closest available player to position 4
        else
                assign supporter1 to position 4
                assign closest available player to position 3
        end if
else
        assign supporter1 to position 3
        assign supporter2 to position 4
end if
```

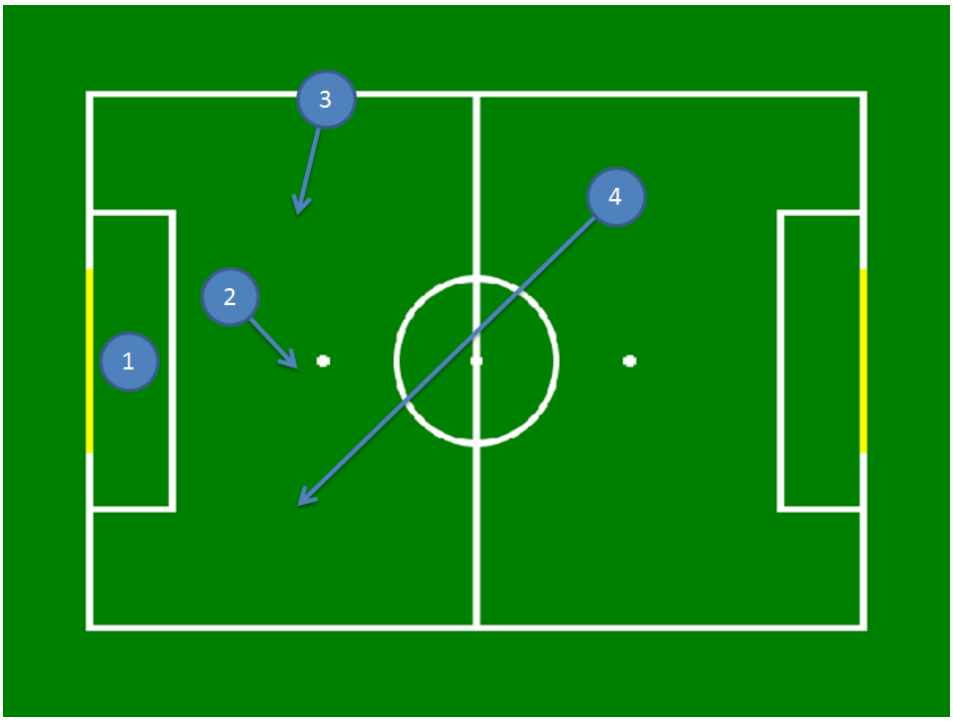


Figure 4.3: Ready Skill Dynamic Position Assignment Example

4.2 Results

4.2.1 Ambiguous Entry, Kidnapping and Line-Up Accuracy

To demonstrate the robot's pose tracking system, a variety of experiments were undertaken. The purpose of these experiments is to provide examples of a robot;

- Localising without a prior estimate – such as when it enters the field at the start of play or returns from the penalty state. Usually entry is assumed to be from the sideline, but the robot is also capable of initially determining its position from within the field.
- Recovering its localisation from false prior information – such as after being "kidnapped" and placed elsewhere. This kidnapping is akin to being displaced by falling or pushing or being moved around by referees.
- Accurately tracking its own pose as it moves towards a destination – whether it is lining up for a Ready skill, or following a ball and preparing to kick it.

The experiments fall into two categories. Table 4.2 displays the results of the experiments involving the robot entering from the sidelines, and then proceeding towards a number of predefined positions on the field. The destination positions chosen are a representative sample of the positions used during the Ready skill line-ups.

Table 4.3 summarises the experiments where the robot is started in the centre of the field (in two different directions), and then is almost immediately kidnapped and taken to a position some distance off (also with a variety of headings). In each of the kidnap experiments the robot is then required to proceed to the Striker destination. A precise description of the entry and kidnapping drop-off positions used in the experiments are provided in Table 4.1.

For each experiment, the robot stops once it believes it has reached a point close to its destination. The results shown in the tables are defined as the differences between the actual measurement of the robot's final position, and its own estimate of its position

once it had stopped. Precise data on the actual final heading of the robot was not recorded, however it was observed to be within very reasonable margins of error for each experiment. The position results for the sideline entry experiment are also shown as a graph in Figure 4.4, to aid in comparing the final position error with the different destination points selected for the robot's line-ups.

The actual final position and estimated final pose of the robot used for the calculation of the results are also shown in the visualisations of the data in Figures 4.5 and 4.6. Also visible in these visualisations are the actual entry positions of the robot and the initial estimate of its position in each experiment. The precise differences between the actual and estimated starting pose are provided as addendums in Appendix B, as these are only the very first estimates of the robot's pose – once it has decided on a main mode, but not necessarily before it has converged. It suffices to note here that on average the starting estimates are within 15cm and 5° of the actual starting position.

Furthermore, the figures represent the robot's estimated position over time as a large number of small dots (forming the purple line), in order to observe the pose estimate as it reacts to the robot's progression. Figure 4.6 also shows where the estimated pose has switched (or "jumped") to the alternative mode, as described in Section 4.1.3, in reaction to the kidnapping of the robot. Figure 4.6b shows a clear single example of such a switch, while Figure 4.6c shows an example of the main mode instead converging to the new position of the robot.

The different entry positions of the robot, and the kidnapping's changes of the robot's position and heading, were deemed sufficient to demonstrate the pose tracking system's ability to react to a variety of scenarios and changes in observation estimates – provided from the field feature matching algorithm (see Section 4.1.1). Scenarios where the robot's heading is "flipped", such as kidnapping it to the opposite position on the field, were not assessed in these experiments, as the measures against such flips rely on the team ball and detected natural landmarks. These are more suited to assessment individually or in proper game conditions. See Section 3.2.1 for team ball results, Anderson's report on natural landmarks [16], and Sections 4.2.2 to 4.2.4 for data on the system's performance during RoboCup 2012.

Table 4.1: Experiment Position Descriptions

Name	X (mm)	Y (mm)	Heading (°)	Description
Right Sideline	-1200	-2000	90	Upper right sideline entry
Left Sideline	-1800	2000	-90	Lower left sideline entry
Penalty Forwards	-1200	0	0	Penalty spot facing forwards
Penalty Backwards	-1200	0	180	Penalty spot facing backwards
Drop Forwards	-1200	-1100	0	Side position facing forwards
Drop Left	-1200	-1100	90	Side position facing left
Drop Backwards	-1200	-1100	180	Side position facing backwards
Drop Right	-1200	-1100	-90	Side position facing right

Table 4.2: Sideline Entry and Line-Up Experiments

Experiment	Entry Position	Destination	Final Position Error (mm)
1	Right Sideline	Midfield Centre	161.29
2	Right Sideline	Striker	64.68
3	Right Sideline	Midfield Right	100.19
4	Right Sideline	Supporter	88.62
5	Right Sideline	Goalie	243.97
6	Left Sideline	Midfield Centre	187.90
7	Left Sideline	Striker	45.56
8	Left Sideline	Midfield Right	140.82
9	Left Sideline	Supporter	94.40
10	Left Sideline	Goalie	208.71
		Average	133.61

Table 4.3: Field Entry and Kidnapping Experiments

Experiment	Entry Position	Kidnap Drop Position	Final Position Error (mm)
11	Penalty Forwards	Drop Forwards	22.74
12	Penalty Forwards	Drop Left	54.72
13	Penalty Forwards	Drop Backwards	85.61
14	Penalty Forwards	Drop Right	169.91
15	Penalty Backwards	Drop Forwards	106.59
16	Penalty Backwards	Drop Left	71.59
17	Penalty Backwards	Drop Backwards	67.93
18	Penalty Backwards	Drop Right	26.69
		Average	75.72

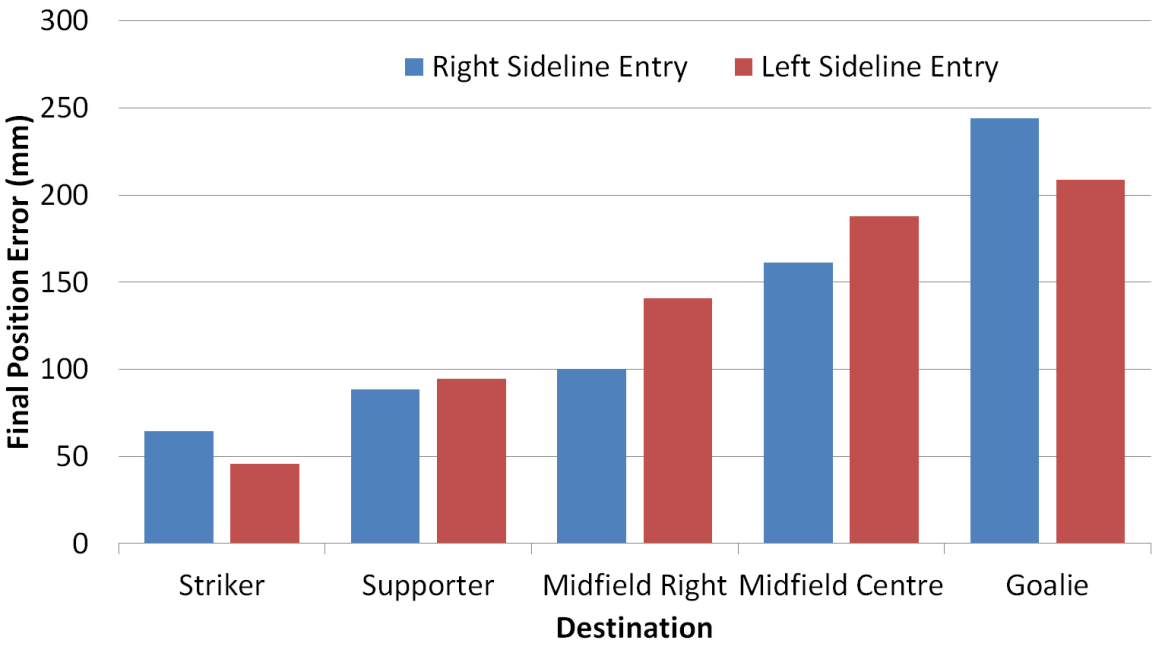
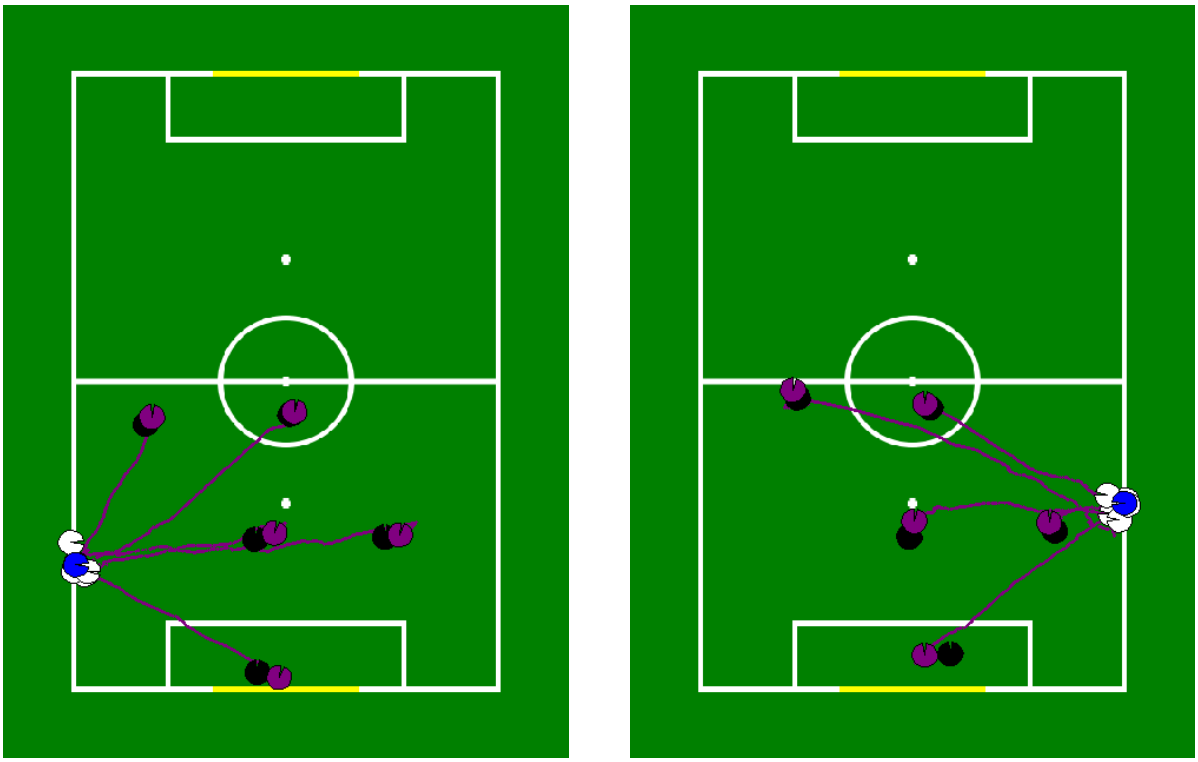


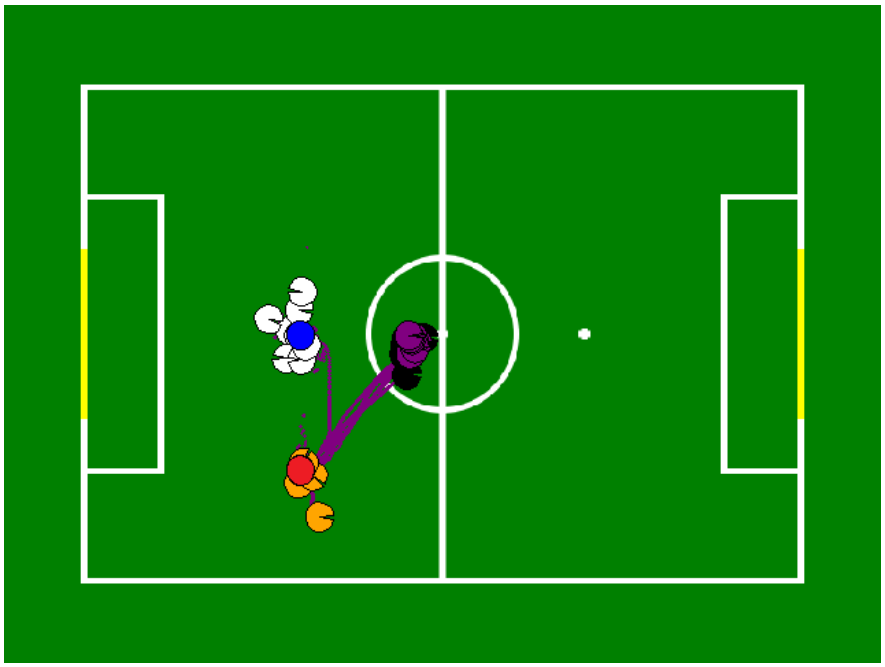
Figure 4.4: Sideline Entry and Line-Up Experiment Results



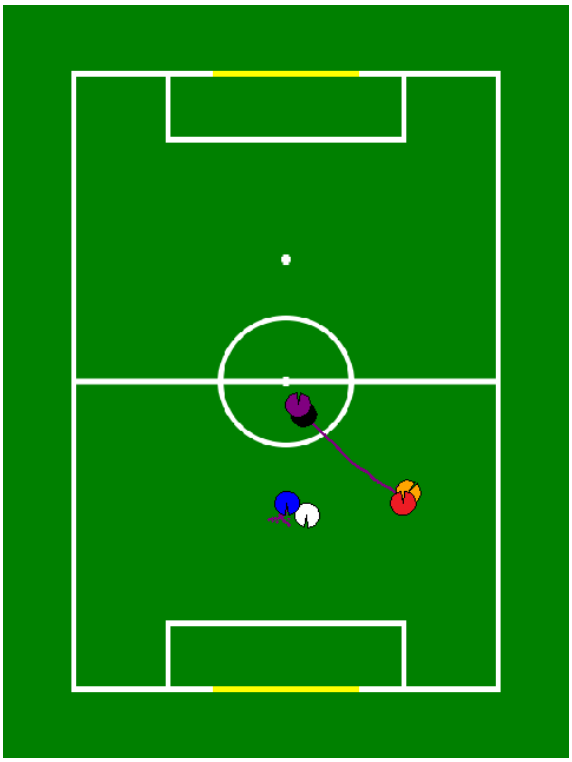
(a) Left Sideline Entry Experiments

(b) Right Sideline Entry Experiments

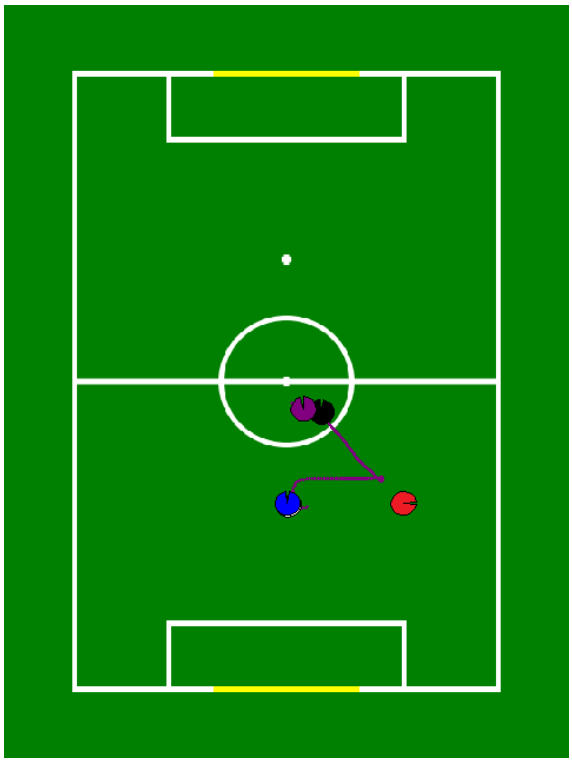
Figure 4.5: Sideline Entry and Line-Up Experiment Visualisations
Actual entry point (blue) vs. initial estimated belief (white).
Actual final position (black) vs. estimated final pose (purple).



(a) Combination of all Field Entry and Kidnapping Experiments.
Each experiment demonstrated successful recovery after kidnapping.



(b) Mode Switching (Experiment 15)



(c) Mode Convergence (Experiment 14)

Figure 4.6: Field Entry and Kidnapping Experiment Visualisations
Actual entry point (blue) vs. initial estimated belief (white).
Actual final position (black) vs. estimated final pose (purple).
Actual kidnapped drop point (red) vs. mode switch estimate (orange).

4.2.2 RoboCup 2012 General Performance

rUNSWift participated in eight official matches of RoboCup 2012, providing the ultimate stage for the team's work, as well as ample opportunity to observe and assess the effectiveness of the described changes to the localisation system. Video footage of seven of the matches were further analysed in detail. Unfortunately the footage of one match did not provide sufficient coverage of the match to be included in these results.

The streamlined dual-modal pose tracking system had no issue running within the imposed time constraints, and was able to process all the incoming observation estimates. Of particular significance this year was the absence of the need for the robot to intermittently pause with the objective of providing more definitive visual observations. This resulted in vast improvements in regards to the time taken to move towards the ball and shoot, as valuable seconds are no longer spent looking up to scan for the goal in order to decide on the angle with which to aim. The absence of the need for such an "active localisation" was one of the critical factors in the success of the team; placing 3rd overall and ranking as the top scoring team in the competition. A variety of factors are attributed to this significant advancement, a further discussion of which follows in Section 4.3.

In addition to maintaining the estimate of the robot's pose over time, the pose tracking system also establishes an initial estimate remarkably quickly and reliably. This was important at the beginning of each half's initial Ready skills, but also throughout matches as robots return to play from being penalised or otherwise removed. Quickly establishing a position allowed robots to enter play or begin their Ready skill as soon as possible, already with a reliable pose estimate. When re-entering the game this clearly aids the team by maximising game time for each robot, but for the Ready skill this was particularly important.

A minimal time required for determining an initial pose estimate allowed for the robots to be placed in any configuration, without worrying about the possibility of mistakenly placing robots in the wrong pre-defined positions – a fairly common occurrence under the stress of competition. Combined with the dynamic approach to robot allocation

developed 4.1.5, this helped maximise the robustness of the Ready Skill (further examined in Section 4.2.4. The quick initial estimate also facilitated the robots moving to their Ready positions quickly, thus allowing enough time for a full scan of the surrounding environment's natural landmarks.

Robot entries were observed 45 times in the video footage of the matches. Of these, 37 ($\sim$ 82%) had determined their initial estimate and begun moving into play almost instantaneously – within roughly one second of becoming active. Within roughly two seconds, 44 ($\sim$ 98%) of the observed robots had begun their entrance, whilst the one remaining entry took roughly three seconds. The vast majority of the time was spent waiting for the head to pan and provide feature rich camera frames of the goal area or centre circle. One further robot entry was observed in the footage; however this robot had not been turned to face the field by the referee in time. Despite this error, the robot completed its initial scan, initiated a wider scan by rotating its body, decided on an estimate of its pose from a glimpse of the home goal area, and begun moving into play within six seconds.

The team also performed particularly well in regards to the introduction of the uniform goal posts and symmetrical environment. Out of the entirety of the seven games analysed, only a total of thirteen discernible flips were observed, and the team successfully avoided scoring any own goals. The localisation and in particular its use of detected natural landmarks [16] to detect and correct flipped robots, was also showcased in the Technical Challenge of the competition. This demonstration was awarded 2$^{\text{nd}}$ place.

4.2.3 RoboCup 2012 Striker Accuracy

To provide an indication of the accuracy of the robot's pose estimate throughout the matches, the kicks observed in the match footage were assessed. These included any normal kick carried out by a robot, or quick forward "jabs" made. These kicks are only ever made by robots believing they are aiming for the opponent's goal. As such, determining whether the aim of the kick was "on target" provides a clear indication of whether the robot was localised at that time with an accurate estimate of its pose.

The aim of the kick is defined as essentially the direction the kicking foot attempts to kick the ball. This is equivalent to the direction the ball would travel assuming it is struck correctly and strongly, and is not intercepted by any obstacles. For example, if a robot is perfectly lined up to kick the ball into the goal, but is interrupted by an adversary pushing it over, this can still be interpreted as the robot having been localised. Using the aim of the kick for data analysis more data and removes the uncertainty associated with the actual kick and resulting velocity of the ball.

Table 4.4 shows a breakdown of all the observed kicks, sorted into one of the following categories;

- **Correct**: The kick was aimed directly into the opponent's goal. This is evidence that the robot's pose estimate was accurate.
- **Close**: The kick was aimed very close to the goal, either aimed towards hitting the goal post or going out over the baseline. This implies the robot's pose estimate was incorrect by a small margin.
- **Miss**: The kick was not aimed towards the goal, but instead towards the sideline. This implies the robot's pose estimate was incorrect by some considerable margin.
- **Flipped**: The kick was aimed directly towards the home goal! This implies that the robot's pose estimate was "flipped".

Table 4.4: Kick accuracy results from official matches

Match (versus)	Correct	Close	Miss	Flipped
RoboCanes	17	1	0	0
Dutch Nao Team	19	0	1	0
B-Human	15	0	1	0
TJArk	15	2	1	3
Austrian Kangaroos	18	0	1	1
Austin Villa	26	1	1	0
HTWK	24	4	1	1
Total	134	8	6	5

4.2.4 RoboCup 2012 Ready Skill Performance

For further indication of the accuracy of the pose tracking system, the results of each Ready Skill observed in the official matches was recorded and is summarised in Table 4.5. As explained in Section 4.1.5, the Ready skill requires each robot to line-up at its assigned position within 45 seconds, hence relying heavily on an accurate estimation of the robot's pose. Figure 4.7 shows an example of an observed complete Ready Skill.

The outcome of each robot observed attempting the Ready Skill was sorted into one of the following categories;

- **Correct**: The robot successfully completed the Ready Skill, finding it's correct position and remaining there. This implies the robot's pose estimate was accurate.
- **Error**: The robot was on track to successfully completing the Ready Skill, observed either in its correct position or very clearly progressing towards it. However an error occurred that prevented it from fully completing the Ready Skill, which was not related to the localisation of the robot. Appendix C provides a more detailed breakdown and description of these errors.
- **Incorrect**: The robot was not in the correct position by the end of the Ready Skill.
- **Flipped**: The robot ended up in the correct position, but on the opposing half of the field!

Table 4.5: Ready skill performance results from official matches

	Localised		**Mislocalised**	
Game	Correct	Error	Incorrect	Flipped
RoboCanes	19	0	1	2
Dutch Nao Team	39	0	0	1
B-Human	19	4	0	0
TJArk	17	3	2	1
Austrian Kangaroos	19	5	0	2
Austin Villa	51	2	0	1
HTWK	49	2	0	2
Total	213	16	3	9

Figure 4.7: Team rUNSWift successfully completing a defensive Ready Skill

4.3 Discussion

This year, the rUNSWift team has enhanced a variety of the robots' perceptions, whose effect on the localisation systems should not be underestimated. An improved walk engine provided more accurate walk odometry data [17], while visual odometry data was also introduced to help detect and account for the robot slipping on the carpet, being caught on obstacles, as well as identifying changes in the heading of the robot as it was being lifted through the air [16]. This enhanced odometry data then directly influenced the accuracy of predictions of the robot's pose estimate.

The upgraded features of the Nao v4 robots were also utilised. The faster processor and availability of both cameras at once, coupled with versatile feature detection methods [22], provided a far greater stream of observed field feature data. This feature data was then combined with the new matching algorithm developed to provide accurate observation estimates of the robot's pose (see Section 4.1.1, or [16] for further detail and assessment).

The experiment results and analysis of RoboCup matches in Section 4.2 provides strong evidence of improvements to many key aspects of the robot's localisation system. These improvements are attributed to a combination of the advancement of the prediction and observation inputs described above, as well as the pose tracking system that utilises them – the focus of this thesis, detailed earlier in Section 4.1.

The pose tracking system used Extended Kalman Filters with a more accurate process model that encapsulates the relationship between the odometry prediction inputs and the estimate of the robot's pose and its uncertainty. This has facilitated the incorporation of the enhanced odometry data, thus a more accurate predicted estimate of the robot's pose is available over time. This estimate can also be provided to the observed feature matching algorithm, enabling it to then provide more accurate observation estimates – as the more accurate initial estimate and importantly its associated uncertainty allows it to select more appropriate estimated positions. The proven result of this is the capability of the robot to maintain its pose estimate without the need for "active localisation" during play, with only a very minor (often almost instantaneous) initial entry scan, as explained in Section 4.2.2.

The stability of the new pose tracking system and its inputs is also visible in the results of the experiments of Section 4.2.1 and Figure 4.5 in particular. In each experiment, the robot proceeded directly to its destination, and the estimates of the robots' pose closely resemble this, without any significant observable deviations. The stability of the estimates throughout the experiments also shows the effectiveness of the approach used to remain robust against noisy and false observations, by selectively applying and discarding observed estimates (see Section 4.1.3).

The experiments are also testament to the accuracy of the system, with the results of the error of its final position estimate being quite small compared to the size of the field (see Tables 4.2 and 4.3). Figure 4.4 also shows the expected correlation between the accuracy of the robot's estimate, and the quality and availability of field features. The Goalie destination resulted in the least accurate line-up in the experiments; as the position within the goals facing outwards has the least amount of available feature data – usually only the goal box line in front of it.

Additionally, Figure 4.5 shows how the Goalie's observed single line feature has only provided an accurate estimate of the Goalie's position away from the line. The majority of its final error is clearly visible as being associated with the ambiguity of the robot's lateral translation along the line. Observations of the field sidelines to the left and right of the Goalie normally reduced this ambiguity and refine the robot's position estimate during matches; however the camera calibrations used in the experiments were not tuned to the standard used for competitions and such observations at the limits of the vision system's capabilities were not available. This highlights the importance of field features and how with even a rough initial estimate of the robot's pose, combining observations of merely two distinct field lines can result in very precise localisation.

The kidnapping experiments shown in Table 4.3 and Figure 4.6 provide further evidence of the accuracy of the pose tracking system, even with sudden shifts in the robots position. In each test the robot was successful in quickly and accurately re-establishing the estimate of its pose and proceeding to its destination with very little discernable impact on the measurement of its final error. These experiments help validate the systems dual-modal approach (Section 4.1.3). Figure 4.6b shows an example of the system reacting by switching to the alternate mode when a clear substantial shift in the robot's pose has been detected.

However these mode switches are relatively rare; they are not seen in any of the sideline experiments (Figure 4.5), and only four are visible in the visualisation of all eight kidnapping experiments (Figure 4.3). Figure 4.6c shows an example of the main mode instead converging to the robot's new pose. In this case the filter system would have increased the uncertainty of the robot's state estimate as the robot was carried – sufficiently increasing the range of observations not discarded to the alternate mode (again, see Section 4.1.3 for more detail on the precise methods used). It is also quite possible that visual odometry [16] indicated a change of heading as the robot was moved, which also would have resulted in a shift and expansion in the range of valid observations. The range of valid observations, now encompassing the observations from the robot's new position, allows the main mode to quickly converge to the new position rather than relying on a mode switch.

The selective application of valid observations hence results in the main state estimate remaining robust to outlying noisy or false positive observations. However, the main mode is still capable of slightly larger shifts if its uncertainty grows sufficiently – such as recovering from a slight discrepancy in the pose after a small tussle or period of fewer observations. This allowed the alternate mode to focus on only larger adjustments to the state estimate, but only when they have proven to be quite certain (Section 4.1.3). Such an approach addresses one of the most problematic issues of the 2011 system, identified by Claridge [9] as the mode switching algorithms not taking the ongoing validity of its modes into account, and only making instantaneous decisions when particular field features were observed.

The effectiveness of the new method of using mode switching to manage the problem of kidnapped or mislocalised robots is highlighted again by the stability of the robot's pose estimate, visible in the experiment visualisations; Figures 4.5 and 4.6. There is a distinct lack of unnecessary and erroneous "jumping" of the robot's pose estimate in the experiments, and the significant reduction in the occurrence of these was also noted in the development and effectiveness of the team's various higher level behaviours. Of particular importance are the improvements in the Striker's line-up speed [23], and simplification (removal) of the Goalie's complex localising behaviour [10, 17].

The results from the analysis of the RoboCup 2012 Ready Skills (Section 4.2.4) and Striker kicks (Section 4.2.3), are also indicators of the stability of the filter system. The lack of active localisation meant that the robots were often observed moving with speed and efficiency, directly for their assigned line-up positions, right from the commencement of Ready skills. The Striker also spent much less time oscillating around the ball, as it was able to decide on a heading to the goal much faster.

However the results of the Striker and Ready Skill analysis also reflect the general reliability and accuracy of the new pose tracking system. A definitive 134 kicks were aimed correctly towards the opponent's goal, while 213 separate instances of correctly lined-up robots were observed. These figures comprise 87% and 88% of their respective total observations, and both these large correct majorities provide significant evidence that overall the robots were reliably able to maintain an accurate estimate of their pose.

The official match observations also indicated a fairly low occurrence of robots that were "flipped", as in they had estimated their heading to be roughly opposite to their true heading and were mistakenly lined-up on the wrong half or shooting towards their own goal. The analysis of striker accuracy and Ready Skill performance reported flipped robots as only comprising between three and four percent of total observations, and these flipped occurrences were often short lived and appeared to have little effect on the outcome of games or of the impressive performance of the rUNSWift team overall.

The low occurrence of flipped robots reflects well on the ability of the pose tracking system to maintain a reliable and accurate estimate, as the initial cause of a flipped robot is a false estimate of a robot's ambiguous heading. Indeed many of the flips occurred after the robot fell over multiple times close to the centre of the field, where accurate re-establishment of the robot's heading is the most susceptible to the problem.

The minimal impact of the flipped robots suggests that the team ball (Section 3.1.6) and natural landmarks [16] measures used to detect and correct such occurrences were reasonably effective. These measures may also have intervened and corrected flipped robots before the mistake had become apparent.

Chapter 5

Future Work

5.1 Spherical Coordinate Ball Filter

This year, the introduction of a more accurate model of ball observation covariance had a pronounced effect on the accuracy and reliability of the resulting state estimate. This approach used assumptions of the errors of the observed camera angles and mathematical models to transform these into estimates of observation covariance ellipses in a convenient Cartesian coordinate system.

Tasse [24] has explored this concept further, comparing the accuracy of filters with different measurement coordinate system choices. He finds that by simply filtering the ball observations as expressed in the original spherical coordinates of the camera (the closest coordinate system to the actual perception process the robots use), significant improvements can be made to the accuracy of the filtered estimate. This approach is worth investigating and experimenting with further.

5.2 Observation Variance Measurements

While improvements to the observation and process models have been made, it still remains for precise measurements of various processes and observations along with their

associated errors and biases to be made. These measurements can be then used to calculate more accurate process and observation updates (and their covariance estimates) across the entire localisation system.

In particular, an analysis of ball observations would be beneficial, as their distances appear to be consistently under-estimated, which may possibly be accounted for. Precise measurement of the robot's walk, not just in a straight line, but as it turns and moves laterally, would also help to improve the prediction updates of both the ball tracking and pose tracking filter systems.

5.3 Distributed World Model Filter

While the accuracy of the distributed team ball was improved, and was able to influence the correction of flipped robots, this was a relatively rare and situational use case. A more complete approach to utilising the combined estimate of the team ball could involve a larger more complex filter, tracking all of the team's pose estimates and the position of the team ball, reminiscent of Sushkov's 2006 approach [8] (also briefly described in Section 2.4). This would facilitate the ball observations making finer adjustments to robot pose estimates over time, as well as helping to avoid the flipped robot problem in addition to reversing it.

Logical further development of the world mode would be to include the increasingly accurate visual observations of other robots, which would have a more direct effect on adjusting other teammates' pose estimates. Friendly and opposing robot observations could be incorporated and combined between teammates, with the eventual goal of tracking the ball as well as every robot on the field. Such a world model would provide opportunities for far more advanced strategic behaviour.

The feature matching algorithm and selective application methods currently applied to the team ball and robot state estimates may also still be applicable to such a world model, to reduce the complexity and required number of modes. The associated processes dictating the procedure for dealing with outlying observations would need to

be more complex to ensure correct data is not thrown away. Care must also be taken to not allow errors to compound in such a world model, as it is possible to imagine that false information could easily begin to propagate and recycle through the system once it has been established.

5.4 Robust Dynamic Ready Skill

While the dynamic position allocation introduced this year had some success, an unforseen problem arose on occasion, when an excess of network traffic at the competition hindered all communication between the robots. As each robot was biased towards ensuring the Striker position was filled, clashes easily occurred when the team's communication was down.

Simple methods of detecting network failures and offsetting the allocated positions appropriately were proposed, however where only partially explored. If the dynamic position allocation is to remain incorporated into the team's behaviour, appropriate measures against network failure should be included.

Chapter 6

Conclusion

The use of the Kalman Filter algorithm and its variants has become popular for use within localisation systems of the RoboCup Standard Platform League. This approach provides an efficient method of filtering noisy process and observation inputs. However it relies on a variety of assumptions and constraints on the system to be estimated. This thesis has described a number of the techniques used by the new rUNSWift localisation system to overcome these limitations in the pursuit of efficient methods of providing accurate and robust estimates of the game ball's state and the individual robots' poses.

More accurate approximations of the state transition models were used with Extended Kalman Filters to precisely adjust the state estimate covariance with respect to robot motion. An improved model of the elliptical covariance of received ball observations was incorporated along with a comprehensive switching and interaction system of two filters to result in a more reactive and accurate estimate of the ball's position, velocity, and the their associated uncertainty. The enhanced estimate of this uncertainty then allowed for increased precision when combining the ball state estimates of each individual robot into a calculation of the collective team estimate of the ball's position.

The upgraded Nao v4 Humanoid facilitated the greater availability of observed visual features. These were matched together with field geometry and the current robot pose estimate to provide a single powerful observed estimate of the robot's pose, contrasting with various single field features providing multi-modal hypotheses as observations.

This observation model allowed for the avoidance of complex multi-modal approximations, in favour of a much simpler system of two Extended Kalman Filters.

The simplicity of the unimodal observation model allowed for a focus on improving the stability of the state estimate, rather than the processes required for the maintenance of a large number of generated modes. A selection process relying heavily on the Mahalanobis distance between the state and observed estimates is used to shelter the main mode from particularly noisy observations – such as those caused by false positive visual classification. This process, combined with the matched feature observations and improvements to the robot's odometry data resulted in a significantly more stable and accurate estimate of the robot's pose. A similar selection process was used to choose the best subset of available ball estimates that agreed with each other, to combine into a team ball estimate not hindered by inaccurate observation data or pose estimates.

However, observations not selected by these processes were not simply discarded. The pose tracking system's alternate mode collected and assessed the validity of the outlying data over time, making adjustments to the main mode only when clearly appropriate. This enabled the system to recover from compounded error or large shocks to the robot's pose. Observations screened from contribution to the team ball were also still used, serving as identifiers of possibly mislocalised robots. The team ball analysis, main and alternate modes of the robot's pose, and detected natural landmark data together form the basis of the mechanism used to recover mislocalised robots from being flipped in the now symmetrical environment.

As a result of the above methods and techniques used, the localisation system has experienced a dramatic improvement. The robots have successfully demonstrated an ability to maintain an estimate of their position and heading with a high degree of reliability and accuracy. They establish their position quickly, both without prior knowledge and following sudden disturbances. The system no longer requires the use of active localisation behaviours, and remains robust against noisy or false positive data. Furthermore, the system's stable estimate displays an aversion to flipping in the symmetric environment, yet it is also capable of detecting and recovering from such mistakes. Overall, the new localisation system helped rUNSWift achieve 3$^{\text{rd}}$ place at RoboCup 2012!

Bibliography

[1] B. Hengst, D. Claridge, B. Hall, and M. Pagnucco, "RoboCup Local Game Management System," RoboCup Federation Project Proposal, 2011.

[2] A. Ratter, B. Hengst, B. Hall, B. White, B. Vance, C. Sammut, D. Claridge, H. Nguyen, J. Ashar, M. Pagnucco, S. Robinson, and Y. Zhu, "rUNSWift Team Report," 2010.

[3] T. Röfer, T. Laue, J. Müller, O. Bösche, A. Burchardt, E. Damrose, K. Gillmann, C. Graf, T. J. de Haas, A. Härtl, A. Rieskamp, A. Schreck, I. Sieverdingbeck, and J.-H. Worch, "B-Human Team Report and Code Release 2009," 2009.

[4] A. Burchardt, T. Laue, and T. Röfer, "Optimizing particle filter parameters for self-localization," in *RoboCup 2010: Robot Soccer World Cup XIV* (J. R. del Solar, E. Chown, and P. G. Ploeger, eds.), vol. 6556 of *Lecture Notes in Artificial Intelligence*, pp. 145–156, Springer Berlin / Heidelberg, 2011.

[5] D. Fox, W. Burgard, F. Dellaert, and S. Thrun, "Monte Carlo Localization: Efficient Position Estimation for Mobile Robots," in *Proceedings of the sixteenth national conference on Artificial intelligence*, American Association for Artificial Intelligence, 1999.

[6] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering Transactions*, vol. 82, 1960.

[7] G. Welch and G. Bishop, "An Introduction to the Kalman Filter," *SIGGRAPH*, 2001.

[8] O. Sushkov, "Robot Localisation Using a Distributed Multi-Modal Kalman Filter, and Friends," Honours Thesis, The University of New South Wales, 2006.

[9] D. Claridge, "Multi-Hypothesis Localisation for the Nao Humanoid Robot in RoboCup SPL," Honours Thesis, The University of New South Wales, 2011.

[10] B. Teh, "Ball Modelling and its Applications in Robot Goalie Behaviours," Special Project Report, The University of New South Wales, 2011.

[11] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, Massachusetts: MIT Press, September 2005.

[12] M. Quinlan and R. Middleton, "Multiple Model Kalman Filters: A Localization Technique for RoboCup Soccer," in *RoboCup 2009: Robot Soccer World Cup XIII*, Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 2010.

[13] S. Julier and J. K. Uhlmann, "A New Extension of the Kalman Filter to Nonlinear Systems," in *The Proceedings of AeroSense: The 11th International Symposium on Aerospace/Defense Sensing, Simulation and Controls*, 1997.

[14] T. Laue, T. J. de Haas, A. Burchardt, C. Graf, T. Röfer, A. Härtl, and A. Rieskamp, "Efficient and Reliable Sensor Models for Humanoid Soccer Robot Self-Localization," in *Proceedings of the 4th Workshop on Humanoid Soccer Robots*, pp. 22 – 29, IEEE-RAS International Conference On Humanoid Robots, 2009.

[15] T. Röfer, T. Laue, J. Müller, A. Fabisch, F. Feldpausch, K. Gillmann, C. Graf, T. J. de Haas, A. Härtl, A. Humann, D. Honsel, P. Kastner, T. Kastner, C. Könemann, B. Markowsky, O. J. L. Riemann, and F. Wenk, "B-Human Team Report and Code Release 2011," 2011.

[16] P. Anderson, "New Methods for Improving Perception in RoboCup SPL," Honours Thesis, The University of New South Wales, 2012.

[17] R. Liu, "Bipedal walk and goalie behaviour in RoboCup SPL," Honours Thesis, The University of New South Wales, 2012.

[18] B. Hengst, "Zero Velocity Ball Extended Kalman Filter," 2011.

[19] T. Soler and M. Chin, "On transformation of covariance matrices between local Cartesian coordinate systems and commutative diagrams," in *Technical Papers. 45th Annual Meeting ACSM*, pp. 393–406, 1985.

[20] P. C. Mahalanobis, "On the generalised distance in statistics," in *Proceedings of the National Institute of Sciences of India*, vol. 2, pp. 49–55, 1936.

[21] R. Roy, "Low Level Behaviours For Soccer-Playing Robots," Special Project Report, The University of New South Wales, 2012.

[22] S. Harris, "Efficient Feature Detection Using RANSAC," Honours Thesis, The University of New South Wales, 2011.

[23] B. Teh, "Dynamic Omnidirectional Kicks on Humanoid Robots," Honours Thesis, The University of New South Wales, 2012.

[24] S. Tasse, M. Hofmann, and O. Urbann, "On sensor model design choices for humanoid robot localization," in *RoboCup 2012: Robot Soccer World Cup XVI* (X. Chen, P. Stone, L. E. Sucar, and T. V. der Zan, eds.), Lecture Notes in Computer Science, p. to appear, Springer Berlin / Heidelberg, 2013.

Appendix A

Team Ball Results

See Section 3.2.1 for a detailed explanation of the experimental source of these results, including robot configuration layouts and ball positions.

Table A.1: Team Ball estimate error (mm), robot configuration A

	Trial 1	Trial 2	Trial 3	Average
Position 1	107.67	86.53	95.17	96.46
Position 2	231.91	398.79	393.16	341.29
Position 3	210.40	250.13	221.44	227.32
Position 4	331.76	282.90	290.01	301.55
Position 5	381.04	401.57	366.52	383.04
Position 6	567.16	567.88	656.46	597.17
Position 7	627.15	702.16	628.55	652.62
Position 8	981.69	969.09	981.06	977.28
Position 9	1031.57	1130.89	1067.31	1076.59
Average	496.70	532.21	522.19	517.04

Table A.2: Team Ball estimate error (mm), robot configuration AB

	Trial 1	Trial 2	Trial 3	Average
Position 1	105.75	91.50	75.33	90.86
Position 2	414.52	412.06	249.94	358.84
Position 3	214.25	107.84	22.10	114.73
Position 4	314.58	77.93	100.10	164.20
Position 5	201.79	73.63	32.59	102.67
Position 6	413.69	220.84	167.15	267.23
Position 7	212.69	110.18	85.03	135.97
Position 8	99.39	44.84	132.00	92.08
Position 9	404.77	15.78	121.48	180.67
Average	264.60	128.29	109.53	167.47

Table A.3: Team Ball estimate error (mm), robot configuration ABCD

	Trial 1	Trial 2	Trial 3	Average
Position 1	77.86	57.79	98.49	78.05
Position 2	42.46	70.41	76.82	63.23
Position 3	18.15	79.13	45.05	47.44
Position 4	96.85	122.42	122.25	113.84
Position 5	30.50	24.15	49.20	34.62
Position 6	74.46	49.55	88.19	70.73
Position 7	60.67	84.77	47.68	64.38
Position 8	89.75	40.16	71.43	67.12
Position 9	94.14	35.81	50.58	60.18
Average	64.98	62.69	72.19	66.62

Appendix B

Ambiguous Entry Results

Table B.1: Amiguous Entry Experiments, Initial Pose Estimate Error
See Section 4.2.1 for a detailed explanation of these experiments.

Experiment	Position Error (mm)	Heading Error (°)
1	166.64	8.07
2	38.46	1.43
3	170.46	1.88
4	179.80	8.31
5	37.62	7.11
6	131.01	4.33
7	8.25	0.75
8	222.03	1.32
9	65.62	1.74
10	132.69	6.39
11	94.81	2.14
12	204.85	6.27
13	296.06	16.23
14	16.80	4.30
15	223.13	7.29
16	356.79	3.11
17	196.78	1.23
18	97.68	1.61
Average	146.64	4.64

Appendix C

Ready Skill Performance

Correct: As defined in Section 4.2.4, the robot successfully completed the Ready Skill.

Close: The kickoff robot was penalised for being slightly too far forward. On one occasion an opponent had pushed the robot forward just before the time ran out. On the other occasion a robot was not properly calibrated for its old body.

Clash: Network difficulties caused two robots to be assigned to the same position.

Timed Out: Too many obstacles or falls caused the time to run out before the robot made it into position.

Penalised: The robot was penalised for pushing as it approached its position.

Table C.1: "Localised" robots during official match Ready states

Game	Correct	Close	Clash	Timed Out	Penalised	Total
RoboCanes	19	0	0	0	0	19
Dutch Nao Team	39	0	0	0	0	39
B-Human	19	0	4	0	0	23
TJArk	17	0	2	0	1	20
Austrian Kangaroos	19	0	3	1	1	24
Austin Villa	51	2	0	0	0	51
HTWK	49	0	0	1	1	51
Total	213	2	9	2	3	229