Physical Model Based Multi-objects Tracking and Prediction in RoboSoccer

Kwun Han and Manuela Veloso
Computer Science Department, Carnegie Mellon University
5000 Forbes Ave., Pittsburgh PA 15213
{kwunh,mmv}+@cs.cmu.edu

Abstract

RoboSoccer is a multi-agent framework in which multiple robots collaborate in an adversarial environment. RoboSoccer can be built with robots and fields of different sizes. In the smallest version of the game, robots cannot incorporate on-board full autonomous capabilities. In particular, vision processing is off-board, centralized, and connected to the individual clients that control the robots. The vision system needs to overview the complete field and to compute in real time positioning information for the moving ball and players. This paper describes our on-going work on developing a multi-object tracking and prediction in this challenging setup. This paper presents our preliminary work applying an extended Kalman filter to follow the trajectory of multiple moving colored objects. We present empirical results that show the effectiveness of the method both in position tracking and prediction. We conclude with a discussion of the approach, results and future work.

Introduction

Robotic Soccer has recently been introduced as a testbed for multiagent AI research, where teamplayers must interact and coordinate with each other to compete against a team of opponents (Kitano *et al.* 1997). Both simulator-based and real robot competitions have been set. We have pursued research on both tracks, developing layered learning approaches within the simulator (Stone & Veloso 1997) and constructing real robots (Veloso, Stone, & Achim 1997).

In this paper, we focus on our work building a reliable system of real robots for robotic soccer. The nature of the game requires good players to have fast perception, fast planning, fast action, accurate control and cooperation. This brings together techniques from different fields such as vision, learning, planning, robotics control and multiagent systems.

Modules which deal with each of the these issues have been or are being designed and implemented to achieve our overall goal (Veloso, Stone, & Achim 1997). This paper concentrates on the perception aspects of our research. Concretely we design a vision algorithm that uses an extended Kalman filter to track and predict the position of the moving robots and ball. The algorithm robustly tracks objects after radical changes of direction (for example due to collisions against a wall or other players), and after intersection of trajectories.

RoboSoccer

RoboSoccer can be built with robots and fields of different sizes. In the smallest version of the game, up to 5 robots in each team of size not larger than 15cm3 compete. Due to the small size of these robots, it is very difficult to perform perception and computation on-board the robot, thus they are usually performed off-board. An overhead camera with a global view of the field is usually used. The output of the camera is fed into the off-board computer(s) and then the vision input is processed. The off-board computer(s) communicate with the robots wireless and send strategic navigational control commands (Veloso, Stone, & Achim 1997). Special color markings are allowed on the robot to differentiate between team members and opponents. The soccer ball used in these competition is an orange-colored golf ball.

The simplest and robust vision operation to pin point the location of robot in the field is color segmentation. This operation has the advantage that it is simple and fast to compute. We are interested in near frame-rate acquisition frequencies.

In order to achieve extremely fast perception, dedicated hardwares are often employed to speed up processing (Sohota *et al.* 1995). We use a Cognachrome Vision System developed at *Newton Labs* which can perform color segmentation and blob orientation finding at a peak rate of 30 frame/sec. The Cognachrome system can track objects of three different types of color. It returns a list of blobs it found for each color. In the setup of the game, similarly to other tasks, such as radar sensing, we have a list of homogeneous moving objects and we need to distinguish between them to apply the correct control. Reliably tracking such system is a challenging problem to address. Furthermore, the readings from the Cognachrome system is rather noisy. (See Figure 1(a) for an example of the returned track of tracking a rolling ball.)

Robot soccer players often need to intercept balls during the game. To do that, we need to perform prediction on the ball position. Predicting the future location of a ball is essential in a speedy and accurate interception. Due to the noisy location readings, it is difficult to extract stable velocities measurements since the extraction process is very sensitive to noise.

In this paper, we present how we solved the latter task, namely tracking the ball and predicting its future location robustly.

Detection of the ball's location is done with the Cognachrome vision system. The Cognachrome vision system can perform very fast objection detection by color segmentation. However, there are two drawbacks with this system. Firstly, the location of the ball is often very noisy. Secondly, the system often misses the ball and produces a no-detection output. In order to locate the ball and perform prediction robustly, we must devise a mechanism to compute the best estimate of the ball's location in the presence of noise and missing data. The technique we are going to use is the Extended Kalman Filter.

Extended Kalman Filter

The Kalman filter provides a solution to the least square method and at the same time is computationally efficient. Its recursive nature is well suited for real-time estimation of system states. Furthermore, it is robust against system noise and it is able to produce good estimations even when the precise nature of the modeled system is unknown. The basic Kalman Filter can be applied only to linear systems. The *Extended* Kalman Filter (EKF) pushes the power of the Kalman Filter further to non-linear systems by linearizing the system equations centered at the current best estimate. The system we are estimating is non-linear, thus we use the EKF.

Here we give a brief description of the extended Kalman Filter. It is by no means the most detailed treatment of the subject, and more extensive references for Kalman-Bucy Filters can be found in (Kalman 1960; Kalman & Bucy 1961).

The Extended Kalman filter is a recursive estimator for a possibly non-linear system. The goal of the filter is to estimate the state of a system. The state is usually denoted as a n-dimensional vector x. A set of equations are used to describe the behavior of the system, described through the equation:

$$x_{k+1} = f(x_k, u_k, w_k),$$

where $f(\cdot)$ is a non-linear function which represents the behavior of the non-linear system, u_k is the external input to the system and w_k is a zero mean, Gaussian random variable with covariance matrix Q_k. w_k captures the noise in the system and any possible discrepancies between the physical system and the model. The subscript k denotes the value of the variable at time step k.

The system being modeled is being observed (measured). The observations can also be non-linear:

$$z_k = h(x_k, v_k),$$

where z_k is the vector of observations and $h(\cdot)$ is the non-linear measurement function, and v_k is another zero-mean Gaussian random variable with covariance matrix R_k. It captures any noise in the measurement process.

The EKF involves a two-step iterative process, namely *update* and *propagate*. The current best estimate of the system's state $\hat{x}$ and its error covariance P_k is computed on each iteration. During update, the current observation is used to refine the current estimate and recompute the covariance. On propagate, the state and covariance of the system at the next time step is calculated using the system's equations. The process then repeats, the update step refines the estimate using the observation, and so on. The update and propagate equations are as follows:

Propagate Equations

$$\begin{aligned} \hat{x}_{k+1}^- &= f(\hat{x}_k, u_k, 0) \\ P_{k+1}^- &= A_k P_k A_k^T + W_k Q_k W_k^T \end{aligned}$$

Update Equations

$$\begin{aligned} K_k &= P_k^- H_k^T (H_k P_k^- H_k^T + V_k R_k V_k^T)^{-1} \\ \hat{x}_k &= \hat{x}_k^- + K_k(z_k - h(\hat{x}_k^-, 0)) \\ P_k &= (I - K_k H_k) P_k^- \end{aligned}$$

Matrices A, W, H and V are Jacobian matrices obtained during the linearization step of the EKF. A and W are the Jacobian of $f(\cdot)$ with respect to x and w, respectively. H and V are the Jacobian of $h(\cdot)$ with respect to x and v, respectively. These matrices are evaluated using the current estimate $\hat{x}_k$.

The Kalman Gain matrix K_k deserves special explanation. The Kalman Gain at the current time step governs how much the filter *trusts* the current observation values. The value $(z_k - h(\hat{x}_k^-, 0))$ is called the innovation. It is the difference between the actual obserervation and the calculated (predicted) observation. A large Kalman Gain means that we trust the observation and therefore adjust the current state estimates according to that new reading. A small Kalman Gain means that we do not trust the observation and rely more on the system equation to update the state estimate.

Prediction

Prediction in a Kalman Filter is performed by repeated applications of the propagate equations to the current state estimate. To obtain the prediction of the system's state in n time-steps ahead, we have to apply the propagate equations n times.

Ball Equations

We modelled our ball in the Kalman filter framework in the following way. We capture the ball's state into 5 variables: the ball's x and y location, the ball's velocities in the x and y direction and a friction parameter (λ_k) for the surface.

These variables are related via the following set of non-linear difference equations:

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \dot{x}_{k+1} \\ \dot{y}_{k+1} \\ \lambda_{k+1} \end{bmatrix} = \begin{bmatrix} x_k + \dot{x}_k \cdot \Delta t \\ y_k + \dot{y}_k \cdot \Delta t \\ \dot{x}_k \cdot \lambda_k \\ \dot{y}_k \cdot \lambda_k \\ \lambda_k \end{bmatrix}$$

The above equation models the ball with simple Newtonian dynamics. λ_k is a friction term which discounts the velocity at each time step. Δt is the time-step size.

The prediction equations are:

$$\begin{aligned} x_{k+n} &= x_k + \dot{x}_k \cdot \Delta t \cdot \alpha_{kn} \\ y_{k+n} &= y_k + \dot{y}_k \cdot \Delta t \cdot \alpha_{kn} \\ \alpha_{kn} &= \begin{cases} 1 & \text{if } \lambda_k = 1 \\ (1-(\lambda_k)^n)/(1-\lambda_k) & \text{otherwise} \end{cases} \end{aligned}$$

The prediction equations are derived by solving the recursive equation obtained by substituting the value of x_{k+i} where i decreases from n to 1. We are only interested in the predicted spatial location of the ball thus we do not explicitly calculate the predicted velocity.

Non-linearities

The RoboSoccer field is surrounded by a 3" high wall. The purpose of the wall is to prevent the ball from falling off the playing field. Naturally, the ball bounces off the field and the dynamics of the ball changes. We attempted to incorporate such non-linearity into our system by modifying the ball's dynamics. The size and shape of the field is known and thus the locations of the wall. We can predict ahead of time the ball-wall contact time and schedule to flip the sign of one of the x or y velocity parameter, depending on which wall the ball is heading towards. This assumes the walls run parallel to the axis.

The prediction value also needs to be adjusted to take into account the existence of the walls. All prediction values which lie outside of the field are adjustable by performing a reflection of the predicted coordinate with respect to the line which forms the wall.

Multi-Object Tracking

Tracking multiple objects simultaneously is trivial if the objects were non-homogeneous, that is, if there is a method in which one can reliably distinguish between them. For example, a red ball and a blue ball. However, problems arise when the objects to be tracked are homogeneous, or are similar enough such that distinguishing between them is not possible. Examples of such scenarios are radar readings, and human/object tracking in a cluttered scene.

This data association problem described above is investigated in detail in (Bar-Shalom 1978; Yao 1992; Bar-Shalom & Fortmann 1988). Our approach is more simplistic. In a less clustered environment such as a robot soccer field, such simplifications do not pose problems. It should be noted however, that the performance of our approach will probably deteriorate very quickly as the object density increases. We will continue this investigation and will test the limitations of our approach.

Our design relies on a distance metric between perceived and projected objects. The perceived object is the reading we obtained from our vision system and the projected object is the value obtained from a one-step prediction using our object model with Kalman Filter. We seek to pair up the location of objects returned by the vision system and the locations predicted by the Kalman filter.

Since the Kalman Filter gives an estimated covariance matrix representing the uncertainties of the objects' locations, we can take advantage of the availability and use the Mahalanobis distance as our distance metric.

The vision system does not guarantee a detection even if the ball is present, thus, we have a potentially non-trivial matching problem. We simplify this by seeking a matching which minimizes the following least square criteria:

$$\sum_i (d(z_{ik}, \hat{x}^-_{ik}, P_k))^2,$$

where $\hat{x}^-_{ik}$ and z_{ik} are the i^{th} matching pair. And the function $d(x, y, P)$ is the Mahalanobis distance metric given by:

$$d(x, \mu, P) = \sqrt{(x-\mu)^T P^{-1}(x-\mu)},$$

where P is the covariance matrix.

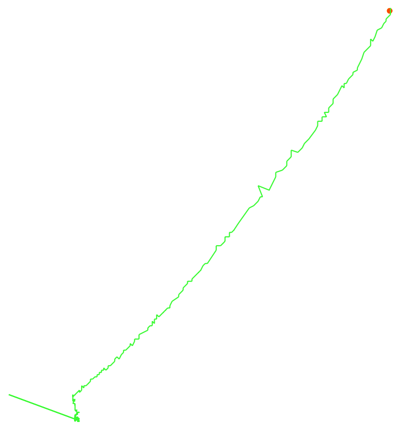
Unmatched predictions are handled by not performing the update step of the Kalman filter.

Results

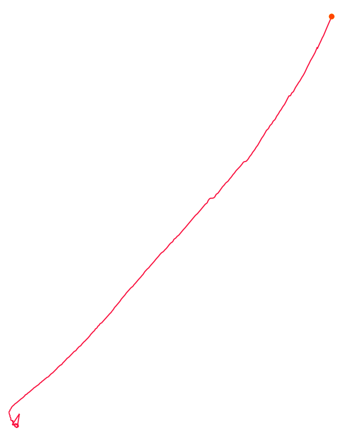
Figure 1(a) shows a typical track that is returned by the vision system and Figure 1(b) shows the track estimated by our algorithm. The estimated track is much smoother than the perceived track.

Figure 2 shows the average prediction error over one of our test track. It plots the error vs. prediction time. The further we try to predict into the future, the more unreliable the prediction. The prediction keeps reliable for a large acceptable window of future.

Figure 3(a) shows the variation of the friction term over time running on a plain track. The friction parameter fluctuates slightly during the run and at around time-step 235, it touches the wall and decelerates quickly. By time-step 300, it came to a complete



(a) Typical ball track raw data obtain from the vision system



(b) Ball track estimation by Kalman Filter

Figure 1: Comparison of raw data input and Kalman estimation output

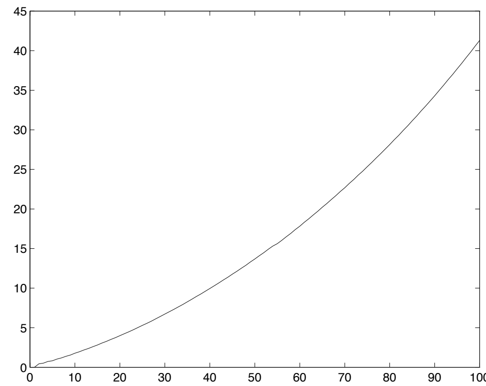


Figure 2: Graph of average Euclidean distance error vs prediction time

stop. Figure 3(b) shows the variation of the friction term when a ball rolls perpendicularly towards a wall. The impact occured at around time-step 60. The fast deceleration (indicated by the drop in the friction parameter) is due to the back spin of the ball caused by the impact. As the spin dies off, the friction term regains its value back to normal at around time-step 200.

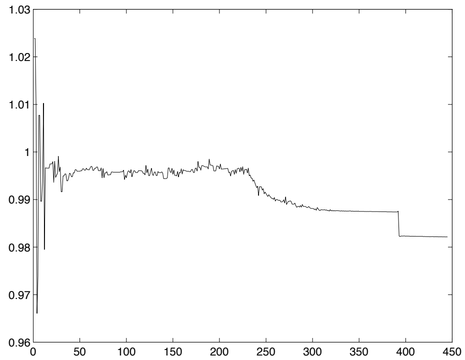
Figure 4 shows results of tracking two balls. The balls started at the top of the diagram and were pushed towards one another. At the crossing point, the balls nearly collided. However, even with such close encounter, the algorithm managed to distinguish between the two balls and continues its tracking successfully.

Conclusion

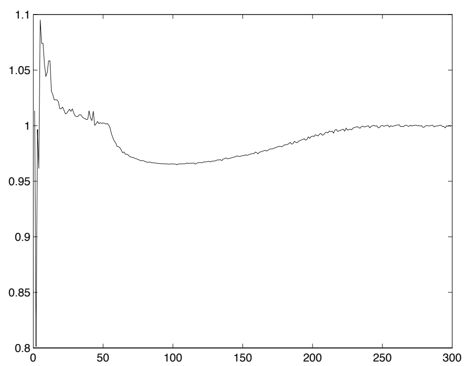
This paper describes our current approach to track multiple moving objects in a robot soccer environment. We assume a non-linear physical model of the system and utilize the Extended Kalman filter to estimate the system parameters. Non-linearities in the environment are also taken into account and resolved. The Kalman Filter framework also allows prediction of the system's future state. Empirical analysis shows that the state estimation and state prediction are very accurate.

Discussion and Future work

Even though our system works very well most of the time, we have encountered a few possible failure modes. Firstly, in the multi-ball case, balls' collision often confuses the system. In such cases, the system will fail to find a good matching. Second, the Kalman filter's parameters (the error covariance matrix) are initialized to some predefined values that were found to be the best from previous runs. However, the parameters of a fresh, new run often deviates from the one of previous runs. This gap in the initial parameters setting often causes the system to be unstable for the first 1/2 sec. The system regains stability after the Kalman filter adjusts its own parameters.



(a) graph of friction term vs time on a plain track



(b) graph of friction term vs time on a bounced track

Figure 3: Variation of friction term over time in two different scenarios

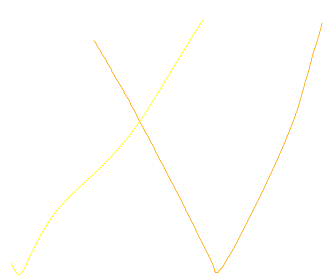


Figure 4: Multi-track detection shows data points matched correctly

In the near future, we plan to experiment with the more robust data association filters. We hope that will solve our problem with the ball mismatch described in the previous paragraph. We are also very eager to modify our algorithm to work with robot soccer player where manuveuring (system input) will be added to our Kalman Filter equations.

References

Bar-Shalom, Y., and Fortmann, T. E. 1988. *Tracking and Data Association*, volume 179 of *Mathematics in Science and Engineering*. Academic Press.

Bar-Shalom, Y. 1978. Tracking methods in a multitarget environment. *IEEE Transactions on Automatic Control* ac-23(4):618–626.

Kalman, R. E., and Bucy, R. S. 1961. New results in linear filter and prediction theory. *Journal of Basic Engineering* 95–108.

Kalman, R. E. 1960. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering* 35–45.

Kitano, H.; Tambe, M.; Stone, P.; and Veloso, M. M. 1997. The robocup'97 synthetic agents challenge. In *Proceedings of IJCAI-97*.

Sohota, M. A.; Mackworth, A. K.; Kingdon, S. J.; and Barman, R. A. 1995. Real-time control of soccer-playing robots using off-board vision: The dynamite testbed. *IEEE Systems, Man and Cybernetics*.

Stone, P., and Veloso, M. 1997. A layered approach to learning client behaviors in the robocup soccer server. *Journal of Applied Artificial Intelligence (AAI)*. To appear.

Veloso, M.; Stone, P.; and Achim, S. 1997. A layered approach for an autonomous robotic soccer system. In *Proceedings of the First International Conference on Autonomous Agents*.

Yao, B. 1992. Data association methods for tracking systems. In Blake, A., and Yuille, A., eds., *Active Vision*. MIT Press. chapter 6.