Multi-Tracking of Moving Objects with Unreliable Sensors for Mobile Robotic Platforms

Extended abstract from Master Thesis

Roberta Pigliacampo
University of Rome "La Sapienza"
E-mail: roberta.pigliacampo@alice.it

28 July 2006

Abstract

This work presents a method for tracking multiple moving objects from noisy and unreliable data taken by a mobile robotic platform. We develop a multi-hypothesis tracking algorithm based on Kalman filtering. The approach is implemented on a four-legged AIBO robot and tested in the context of multiple-balls tracking in the RoboCup domain, with features extracted from the vision module.

1 Introduction

The successful operations of a robotic agent are strictly dependent on the knowledge about the world that the agent is able to acquire. An accurate knowledge of the surrounding environment can be used to improve the robot's behavior and make better decisions. The kind of knowledge that a robot require depends on the kind of environment in which it has to operate and the kind of tasks it has to accomplish.

For many application domains, the robot is required to estimate and update the state of several objects present in the scene. For example, choosing the right object to grasp in the presence of a set of objects in the environment, requires knowledge of all the objects' locations and trajectories.

The process of estimating object' s properties over time is usually referred to as *object tracking*. The difficulty of the object tracking problem depends on a number of factors, such as how accurately the robot can estimate its own motion, the predictability of the object' s motion, the accuracy of the sensors being used. Moreover the data acquired are often incomplete, because of physical limitations of the sensors, but also because often some of the components of interest of the vector state are not directly measurable and have to been calculated. Particularly, in the case of multi-tracking, there are further difficulties due to tracks management and the problem of establishing which is the object which determined the received report (*Data Association*).
Object tracking is a very well known and deeply studied area, and it has several application domains such as security, surveillance, military applications, etc...

This work focuses on the problem of multi-object tracking with unreliable sensors. In particular our goal is to track the location of n balls with a four-legged AIBO robot in the RoboCup domain, which aims at playing soccer with teams of mobile robots. This domain poses highly challenging target tracking problems due to:

- the dynamics of the soccer game: the ball frequently bounces off the borders of the field or gets kicked by other robots; such interactions between target and environment result in non-linear motion of the ball;
- limited processing power, which poses computational constraints on the tracking problem and requires an efficient solution;
- low sensors' quality, which provides unreliable and noisy distance measurements for the ball;
- presence of systematic and not predictable errors.

Our approach is a multi-hypothesis Kalman Filter. The input to our method consists of features extracted from the vision system of the robot. The technique used applies Kalman filters to estimate the state of each target. This provides highly efficient state estimates, satisfying the computational constraints. Although Kalman Filter is restricted to representing unimodal probability distributions and estimating the state of linear systems, a multiple-hypothesis tracking allows to represent multi-modal distributions and to manage the non-linearities present in the ball' s motion. The choice of this approach derives from the requirements of our application. In particular, our aim is not to obtain a perfectly accurate target' s state estimate, but to develop an efficient solution to the multi-tracking problem for an high number of targets and that allows to manage the unpredictable systematic errors, which can occur during the game.
In addition, notice that since we apply the fusion process at feature level, the proposed solution is general and can be easily applied to other domains or other kinds of objects (e.g., opponents' position in the robocup field).

This paper is organized as follows. In Section 2 we describe in details the technique used; Section 3 presents the implementation choices in the application scenario and the experimental results.

2 Multiple-Hypothesis Kalman Filter

When developing a multi-object tracking method, one usually has to deal with track initiation, track update including prediction and data association and track deletion. The process is divided into two fundamental steps:

1. *association*: assignment of each incoming report to a specific target track;
2. *estimate*: the received report is used to provide a state estimate of the associated track.

In what follows we will first describe the state estimate using a Kalman Filter and then its extension with a Data Association algorithm and tracks management methods.

2.1 Kalman filtering

A Kalman Filter is an optimal recursive data processing algorithm [1]. Such a filter [2] represent an efficient solution to the general problem of estimating the state $x \in \Re^n$ of a discrete-time controlled process that is governed by the linear stochastic difference equation:

$$x_t = Fx_{t-1} + Gu_{t-1} + w_{t-1} \quad (1)$$

where x_t is the vector state and u_t is a known input vector, with a measurement $z \in \Re^m$ that is:

$$z_t = H_t x_t + v_t \quad (2)$$

The random variables w_t and v_t represent the process and measurement noise (respectively). They are assumed to be independent, white and with normal probability distributions

$$p(w) \sim \mathcal{N}(0, Q) \quad (3)$$

$$p(v) \sim \mathcal{N}(0, R) \quad (4)$$

where Q and R are, respectively, the process noise covariance and the measurement noise covariance.
The Kalman Filter algorithm consist of essentially two stages:

1. *prediction* (or *time update*):

$$\hat{x}_t^- = F\hat{x}_{t-1} + Gu_{t-1} \quad (5)$$

$$P_t^- = FP_{t-1}F^T + Q \quad (6)$$

   where $\hat{x}_t^-$ and P_t^- are the vector state and covariance error *a priori* estimates for the next time step.

2. *update* (or *measurement update*): the new report z_t is incorporated into the *a priori* estimate to obtain an improved *a posteriori* estimate, according to the following equations.

$$K_t = P_t^- H^T (HP_t^- H^T + R)^{-1} \quad (7)$$

$$\hat{x}_t = \hat{x}_t^- + K_t(z_t - H\hat{x}_t^-) \quad (8)$$

$$P_t = (I - K_t H)P_t^- \quad (9)$$

K_t is the filter' s *gain*, which represent the degree to which the measurement is incorporated in the new estimate [3], providing an indicative measure of the filter' s accuracy. After each time and measurement update pair, the process is repeated with the previous *a posteriori* estimates used to project or predict the new *a priori* estimates. This recursive nature of the filter allows a sequential processing of the received data, so that it is not necessary to store the complete data set nor to reprocess existing data if a new measurement becomes available [4].

2.2 Tracks management

To keep track of a variable ad unknown number of moving objects we use a set of Kalman Filters. Each time a new observation is received, it must be associate to the correct track among the set of the existing tracks, or, if it represents a new target, a new track must be created. Thus, the tracking system needs some mechanisms of Data Association and tracks management (see [5]), that we will describe in this section.

2.2.1 Data Association

The technique used for the data association is the *Nearest Neighbors rule*, which is the simplest approach for determining which tracked object produced a given sensor report. When a new report is received, all existing tracks are projected forward to the time of the new measurement. Then the report is assigned to the nearest such estimate. More generally, the distance calculation is computed to reflect the relative uncertainties (covariances) associated with each track and report. The most widely used measure of the correlation between two mean and covariance pair $\{\mathbf{x}_1, \mathbf{P}_1\}$ and $\{\mathbf{x}_2, \mathbf{P}_2\}$, which are assumed to be Gaussian-distributed random variables, is:

$$P_{ass}(\mathbf{x}_1, \mathbf{x}_2) = \frac{\exp\left(-\frac{1}{2}(\mathbf{x}_1 - \mathbf{x}_2)(\mathbf{P}_1 + \mathbf{P}_2)^{-1}(\mathbf{x}_1 - \mathbf{x}_2)^T\right)}{\sqrt{2\pi \mid (\mathbf{P}_1 + \mathbf{P}_2) \mid}} \quad (10)$$

If this quantity is above a given threshold, the two estimates are considered to be feasibly correlated. A report is assigned to the track with which it has the highest association ranking. In this way, a multiple-target problem can be decomposed into a set of single-target problems.

2.2.2 Track formation

The nearest-neighbors rule is very simple and intuitive, but presents some difficulties. A first problem is in creating initial tracks for multiple targets, because some components of the vector state are not directly measurable. In the case of a single target, two reports can be accumulated to derive an estimate of such components. For multiple target, there' s no obvious way to deduce such initial values: the first two reports could represent successive position of a single object or the initial detection of two distinct objects. Every subsequent report could be the continuation of a known track or the start of a new one.

So when a new report is obtained, if it is not highly correlated with an existing track, a new track is created and a new Kalman filter is initialized with the position (x, y) observed and giving to all the not observed components (e.g., velocity) a null value with a relatively high covariance. If the subsequent reports will confirm the track existence, the filter will converge to the real state.

2.2.3 Track deletion

In many cases, some objects are not observed for a while, with the uncertainty in the state estimate increasing. Moreover the presence of noisy sensors can determine spurious reports, which give rise to spurious tracks. Thus, the tracking system needs an

additional mechanism to recognize and delete tracks that do not receive any subsequent confirming reports.
We have considered, as indicative measure of the uncertainty in the state estimate of each target, the filter' s gain relative to the track:

$$K_t = P_t^- H^T (H P_t^- H^T + R)^{-1} \tag{11}$$

and experimentally established a threshold for the track deletion: if the received reports do not confirm a certain track for a period of time the gain' s value grows exceeding the threshold and determining the track deletion.

2.2.4 Track splitting

When two object are sufficiently close together, the observations are highly correlated with more than one track. In these cases a missassignment can cause the Kalman-filtering process to converge very slowly, or fail to converge altogether. Moreover, tracks updated with missassigned reports (or not updated at all) will tend to correlate poorly with subsequent reports and may, therefore, be mistaken as spurious by the track deletion mechanism; mistakenly deleted track then necessitate subsequent track initiation and a possible repetition of the process.
The choice of a *multi-hypothesis tracking* has been made to give a solution to the problem of assignment ambiguity: when the correct association is not known, more association hypothesis are created. The new observation received is used to update all the tracks with which it has a probability association that exceed the threshold value. A copy of the not updated track is also maintained (*track splitting*). Subsequent reports can be used to determine which assignment is correct.

2.2.5 Track merging

One important issue of the track splitting technique is a proliferation in the number of tracks. Because track splitting does not decompose a multiple-target tracking into independent single-target problems, the deletion mechanism described in section 2.2.3 is not sufficient. For example, two hypothesis tracks may lock onto the trajectory of a single object; because both tracks are valid, the standard track-deletion mechanism cannot eliminate either of them.
The deletion procedure has to be modified to detect redundant tracks and, therefore, cannot look at just one track at a time. At each step, for each track the correlation with all the other tracks is calculated using equation (10). If the association probability between two tracks exceeds a threshold (experimentally established), one of the two tracks is deleted, keeping only the most significant hypothesis.

3 Experimental Results

To evaluate the effectiveness of our approach, we implemented our algorithm on the Sony AIBO Ers7 for a multiple-balls tracking in the Robocup domain and carried out a series of experiment with real data. These experiments demonstrate that our tracking system can keep track of all the ball in the field, even in situation of association ambiguity and in presence of non-linearities of the ball' s motion.

3.1 Implementation Details

As we said we considered the tracking of an unknown and variable number of balls by a robot.

The observation model is a random variable $x_s = (x, y, \theta, v, w)$ with mean $\hat{x}_s$ and covariance Σ_s, where (x, y) is the position, θ is the heading, v is the translational velocity and w is the rotational velocity (which is zero in the case of the ball). We assume that Σ_s is a constant diagonal matrix

$$\Sigma_s = diag(\sigma_x^2, \sigma_y^2, \sigma_\theta^2, \sigma_v^2, \sigma_w^2)$$

where $\sigma_x^2, \sigma_y^2, \sigma_\theta^2, \sigma_v^2$ and σ_w^2 are constant standard deviation for position, heading and velocity, determined through experiments. When initiating a new track a new Kalman Filter is created, setting its state to:

$$\hat{x}_r = (x, y, 0, 0, 0) \tag{12}$$

$$\Sigma_r = \Sigma_{r_0} \tag{13}$$

where

$$\Sigma_{r_0} = diag(\sigma_x^2, \sigma_y^2, \sigma_{\theta_0}^2, \sigma_{v_0}^2, \sigma_{w_0}^2)$$

with $\sigma_{\theta_0}^2, \sigma_{v_0}^2$ and $\sigma_{w_0}^2$ are relatively high initial standard deviation for heading and velocity.

For predicting the state of a track we use a simple motion model, where we assume that the ball moves with constant speed. Given the time interval t, between two frames, the track is projected according to:

$$\hat{x}_r \longleftarrow F_s(\hat{x}_r, t) = \begin{pmatrix} \hat{x}_r + \cos(\hat{\theta}_r)\hat{v}_r t \\ \hat{y}_r + \sin(\hat{\theta}_r)\hat{v}_r t \\ \hat{\theta}_r + \hat{w}_r t \\ \hat{v}_r \\ \hat{w}_r \end{pmatrix}$$

$$\Sigma_r \longleftarrow \nabla F_s \Sigma_r \nabla F_s^{\,T} + \Sigma_a(t)$$

where ∇F_s is the Jacobian of F_s and $\Sigma_a(t)$ is the covariance of some additive Gaussian noise with zero mean:

$$\Sigma_a(t) = diag(\sigma_{x_a}^2 t, \sigma_{y_a}^2 t, \sigma_{\theta_a}^2 t, \sigma_{v_a}^2 t, \sigma_{w_a}^2 t)$$

with $\sigma_{x_a}^2, \sigma_{y_a}^2, \sigma_{\theta_a}^2, \sigma_{v_a}^2$ and $\sigma_{w_a}^2$ being some constant standard deviation determined through experiments.

When a new measurement $\hat{x}_s$ arrives which correspond to the track $\hat{x}_r$ we fuse observation and track according to:

$$\hat{x}_r \longleftarrow (\Sigma_r^{-1} + \Sigma_s^{-1})^{-1} (\Sigma_r^{-1} \hat{x}_r + \Sigma_s^{-1} \hat{x}_s) \tag{14}$$

$$\Sigma_r \longleftarrow (\Sigma_r^{-1} + \Sigma_s^{-1})^{-1} \tag{15}$$

In Table 1 is shown the tracking algorithm in detail.

Inputs:	
$T = (t_0, t_1, ..., t_M)$	set of tracks
z_k	observation

while (no observations)
- $\forall t_i \in T$:
  - evolve t_i;
  - calculate K_{t_i};
  - **if** ($K_{t_i} >$ gain threshold) kill t_i;
- $\forall t_i \in T$:
  - $\forall t_j \in T$:
    - **if** $(i \neq j)\{$ calculate $P_{ass}(t_i, t_j)$;
      if $(P_{ass}(t_i, t_j) >$ redundant track threshold) kill t_i;$\}$

if (z_k received):
- **if** $(T = \emptyset)$ create a new track;
- **else**
- $T_{z_k} := \emptyset$; // set of tracks associated with z_k
- $\forall t_i \in T$:
  - calculate $P_{ass}(z_k, t_i)$;
  - **if** $(P_{ass}(z_k, t_i) >$ pa-threshold) add t_i to T_{z_k}
- **if** $(|T_{z_k}| > 1) \forall t_i \in T_{z_k}$:
  - splitting: not updated copy of t_i added to T_k;
  - t_i update with z_k;
- **else if** $(|T_{z_k}| = 1)$ t_i update with z_k
  - **else** create a new track;

Table 1: Multi-hypothesis multi-tracking algorithm

3.2 Experiments

The tracking system has been tested on real data extracted with the vision software of the robot. We will present in this section some significant situations.

In Fig. (1) we have the representation of the x values over time of two balls. The experiment demonstrate that the tracking system can keep track of more moving objects, also in presence of a slight non-linearity. Moreover it shows how spurious tracks are managed with the creation of new tracks early deleted because of the absence of confirming reports.

The experiment in Fig. (2) shows two crossing balls (x over time). In this case the multi-hypothesis approach allows a correct tracking also in the crossing point where there's a high association ambiguity. Moreover we can see how the non linearity is managed: two tracks, corresponding to the two linear parts, are created.

Finally the Fig. (3) shows on the xy plane a not moving ball and two other balls with crossing trajectories. The system is able to keep track of all the three balls creating more association hypothesis in the crossing point. In this case the stopped ball could represent a systematic error (e.g., a robot of the red team seen as an orange object, and thus recognized as a ball, because of changes in the lighting conditions): with the knowledge of all the three trajectories, it can decide to consider the 'more dangerous' ball.

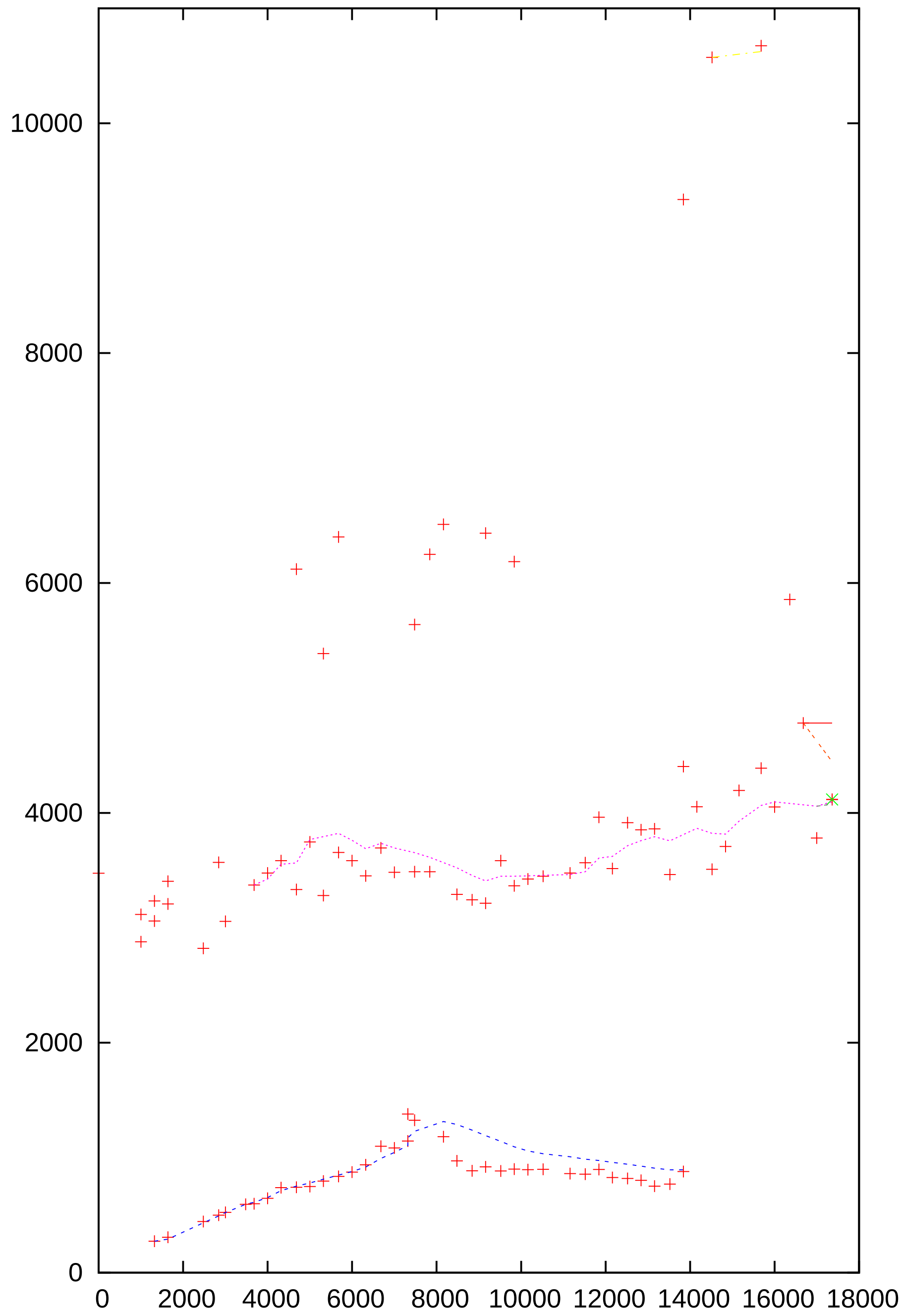

Figure 1: Multi-tracking

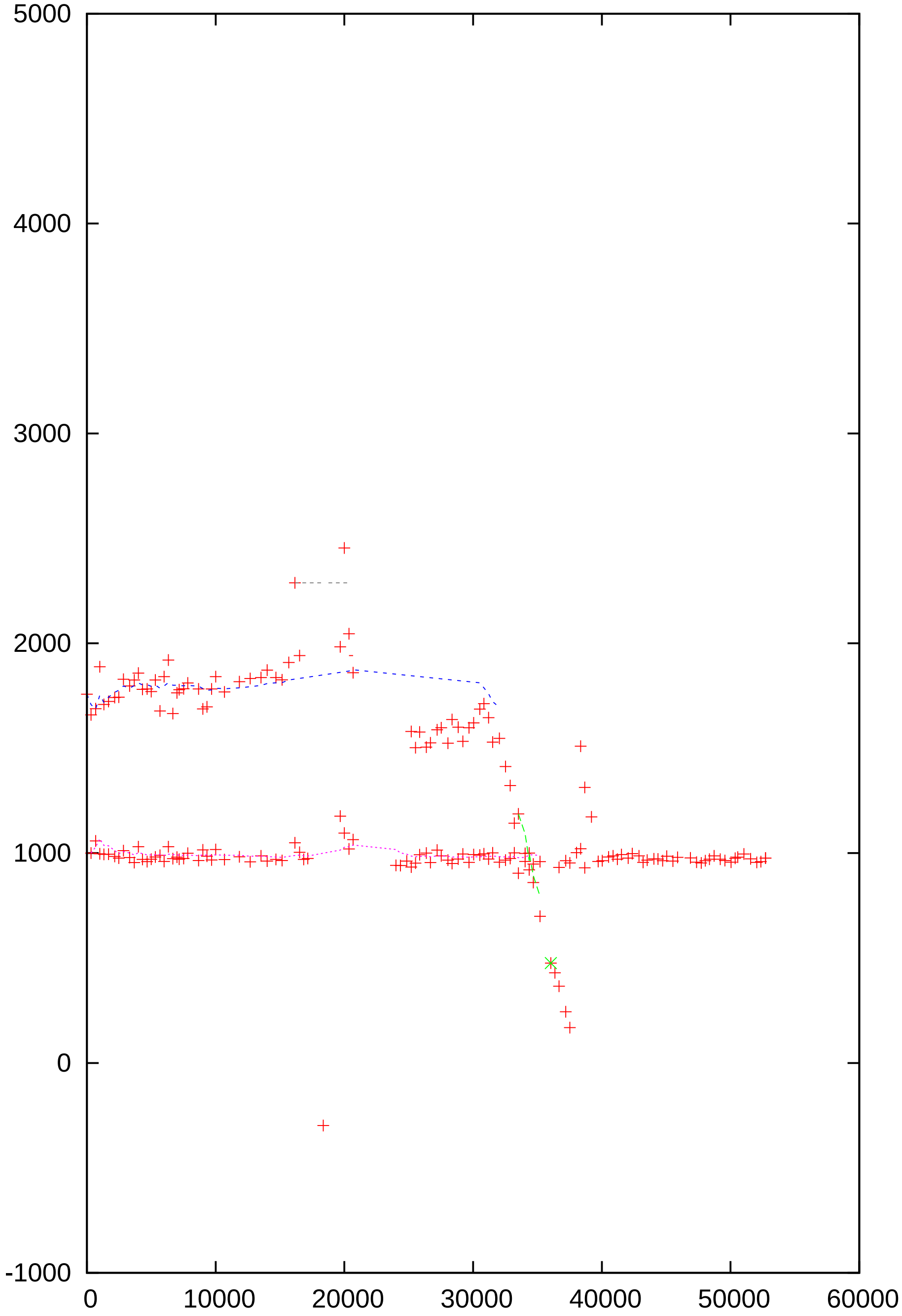

Figure 2: Non-linearities management

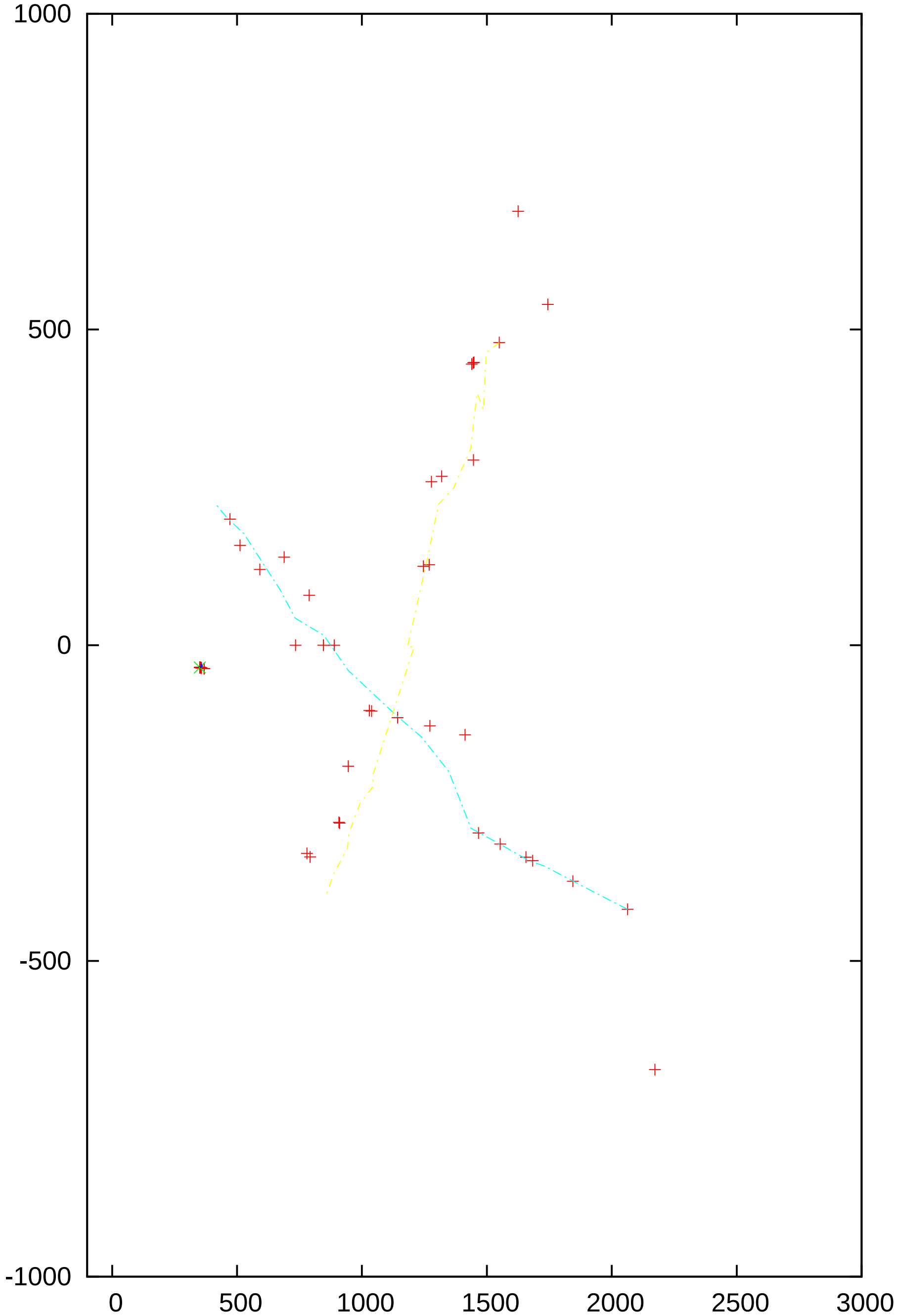

Figure 3: Crossing trajectories

In conclusion, the experiments showed that:

- the implemented system keeps track of the state of all the objects for which observations are received;
- the spurious reports due to noisy sensors are filtered;
- the mechanism of tracks management compensates for Kalman Filtering limitations in presence of non-linearities;
- the tracking system allows systematic errors management.

References

[1] P. S. Maybeck. *Stochastic models, estimation and control. vol I.* ACADEMIC PRESS, 1979.

[2] G. Welch and G. Bishop. An introduction to the kalman filter. Technical report, University of North Carolina at Chapel Hill, NC 27599-3175, 2004.

[3] W. Burgar S. Thrun and D. Fox. *Probabilistic Robotics*. The MIT Press, 2005.

[4] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for on-line nonlinear/non-gaussian bayesian tracking. In *IEEE Transactions on signal processing*, 2002.

[5] D. L. Hall and J. Llinas. *Handbook for multisensor data fusion (Electrical engineering and applied signal processing)*. CRC Press LLC, 2001.

[6] S. Russel and P. Norvig. *Artificial Intelligence - A Modern Approach*. Prentice Hall, second edition, 2003.

[7] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 1960.

[8] Y. C. Ho and R. C. K. Lee. *A Bayesian approach to problems in stochastic estimation and control*. IEEE Trans. Automat. Contr., 1964.

[9] X.-R. Li Y. Bar-Shalom and T. Kirubarajan. *Estimation with Applications to Tracking and Navigation*. John Wiley, 2001.

[10] Y. Bar-Shalom and T. E. Fortmann. *Tracking and Data Association*. Academic Press, 1988.

[11] Y. Bar-Shalom and X.-R. Li. *Multitarget-Multisensor Tracking: principles and techniques*. Yaakov Bar-Shalom, 1995.

[12] E. A. Wan and R. van der Merwe. The unscented kalman filter for nonlinear estimation. Technical report, Oregon Graduate Institute of Science and Technology, 2001.

[13] A. Doucet, J. F. G. de Freitas, and N. J. Gordon. *An introduction to sequential Monte Carlo methods*. Springer-Verlag, 2001.

[14] A. Doucet, S. Godsill, and C. Andrieu. On sequential monte carlo sampling methods for bayesian filtering. *Statist. Comput.*, 2001.

[15] S. Arulampalam and B. Ristic. *Comparison of the particle filter with range parameterized and modified polar EKF*. Proc. SPIE, 2000.

[16] M. R. Neal. Probabilistic inference using markov chain monte carlo methods. Technical report, Department of Computer Science, University of Toronto, 1993.

[17] L. Marchetti. Localizzazione in ambiente robotico tramite l'uso di filtri particellari. Master's thesis, Università di Roma "La Sapienza", 2005.

[18] R. Tramma. Riconoscimento di oggetti colorati mediante un approccio gerarchico. Master's thesis, Università di Roma "La Sapienza", 2006.

[19] C. Kwok and D. Fox. Map-based multiple model tracking of a moving object. Robocup Symposium, 2004.

[20] R. Mohr and B. Triggs. Projective geometry for image analysis. *ISPRS*, 1996.

[21] T. Röfer et al. German team robocup 2004. Technical report, Available from http://www.germanteam.org/, 2004.

[22] www.robocup.org.

[23] www.aibo.com.