Team-Driven Multi-Model Motion Tracking with Communication

Yang Gu
School of Computer Science
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213
Email: guyang@cs.cmu.edu

Manuela Veloso
School of Computer Science
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213
Email: mmv@cs.cmu.edu

***Abstract*— Interactions are frequently seen between the robot and the targets being tracked within the robotics community. Modeling the interactions using knowledge of robot cognition improves the performance of the tracker. Communication improves the performance of a multi-agent system. The focus of this paper is to present our solution to integrate the communication information into our team-driven multi-model motion tracking. We present the probabilistic tracking algorithm in detail and present empirical results both in simulation and in a Segway soccer team. The information from team communication allows the robot to much more effectively track mobile targets.**

I. Introduction

Target tracking is widely used in robot applications, e.g. [1]. We identify two kinds of tracking problems in which: (i) the tracker is static or does not actuate on the tracked target and (ii) the tracker actuates on the target. As mobile robots become more and more useful in everyday life, the scenarios are more frequently seen where they actuate on the targets. It is important to predict the location and velocity of targets in the robot's vicinity after executing actions over them [2]. When tracking is performed by a robot executing specific tasks actuating on the target being tracked, such as a soccer robot grabbing a ball and passing it to a teammate, the motion model of the target relies on the robot's own actions. The robot's tactic provides valuable information in terms of the target behavior, and we introduced a tactic-based motion modeling and tracking algorithms [3]. Further, when a team of robots are cooperating on a common task, any robot can actuate on the targets, which makes the motion model of the target even more complex. We proposed an extension to the tactic-based tracking to solve the plan-dependent multi-target tracking problem [4]. The dynamic multiple motion models are based on the predefined team coordination plan. Communication between robots is not used in their approach and every robot coordinates based on observation only. In some cases, when the tactics of the team member are not exactly determined, the motion model of the target is unprecise so that the tracking performance will be affected seriously.

However, communication improves the performance of such a multi-agent system [5]. When communication is enabled, a shared world model that stores the state of the team can be constructed. The focus of this paper is to present our solution to integrate the communication information into our team-driven multi-model motion tracking. The techniques that we describe are applicable to any domain where a team of agents are cooperating on a specific task and any of them can actuate on the targets in the field. Actuation on the targets is sent as communication message to a specified team member to update and synchronize the motion model of the tracked target.

The paper is organized as follows. We first give a brief description of the Segway RMP soccer robot, which is the testing environment of our approach. Next we show our team-driven multi-model tracking scheme which incorporates the team communication information. We then describe our tracking algorithm in detail, leading to our experimental results, related work and conclusions.

II. Segway RMP Soccer Robot

The Segway platform is unique due to its combination of wheel actuators and dynamic balancing [6]. Segway RMP, or Robot Mobility Platform, provides an extensible control platform for robotics research. It imbues the robot with the novel characteristics of a fast platform and travel long ranges, able to carry significant payloads, able to navigate in relatively tight spaces for its size, and provides the opportunity to mount sensors at a height comparable to human eye level.

In our previous work, we have developed a Segway RMP robot base capable of playing Segway soccer. We briefly describe the two major components of the control architecture, the sensor and the robot cognition, which are highly related to our team-driven motion tracking. We give an example to motivate our purpose to include the team actuation models and communication to help tracking.

A. Vision Sensor and Infrared Sensors

The goal of vision is to provide as many valid estimates of targets as possible. Tracking then fuses this information to track the most interesting targets of relevance to the robot. We use one pan-tilt camera as the vision sensor. We do not discuss the localization of the robot in the sense that a lot of soccer tasks can be done by the Segway RMP robot without localization knowledge. Also we use global reference for position and velocity in this paper which means it is

Fig. 1. A sequence of views from Segway Robot's pan-tilt camera.

relative to the reference point where the robot starts to do dead reckoning.

We have equipped each robot with infrared sensors (IR) to reliably detect the objects located in the catchable area of the robot. Its measurement is a binary value indicating whether or not an object is there. In most cases, this is the blind area of the pan-tilt camera. Therefore, the infrared sensor is particularly useful when the robot is grabbing the ball.

B. Robot Cognition

A control architecture, called Skills-Tactics-Plays, was proposed in [7] to achieve the goals of responsive, adversarial team control. The key component of STP is the division between single robot behavior and team behavior.

We construct the robot cognition using a similar architecture. Plays, tactics, and skills, form a hierarchy for team control. Plays control the team behavior through tactics, while tactics encapsulate individual robot behavior and instantiate actions through sequences of skills. Skills implement the focused control policy for actually generating useful actions. Figure 2 shows the $SSMs$ and transitions for an example tactic: *CatchKickToTeammate*, which contains six skills. Each node in the figure is a skill and the edges show the transition between skills.

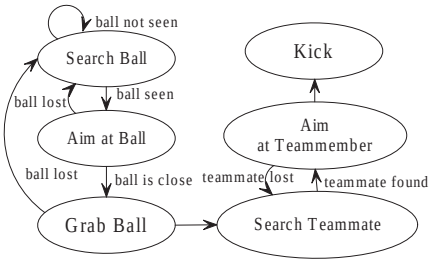

Fig. 2. Skill state machines for an example tactic: *CatchKickToTeammate*.

C. Team Actuation Models Help Tracking

In order to motivate our purpose to include the team actuation models and communication to help tracking, let us first look at what the robot sees from its own camera. Figure 1 shows a sequence of views from a Segway Robot's pan-tilt camera. Each picture corresponds to one frame identified by the frame id shown at the right-lower corner. The current frame rate is approximately 30 frames/sec. Note that the camera has a limited field of view (FOV), which makes tracking more useful. Because the estimation from the tracker is translated into a command to guide the camera where to look, which makes it possible that the target consistently located in the camera's FOV.

The robot is trying to track the ball with its pan-tilt camera. Hence the role of the robot is to keep the ball in its view without losing it. At the beginning, the ball is static and the robot finds the ball at frame# 72.At frame# 431, the ball is kicked towards the robot. Since the ball moves fast, the robot with the limited and narrow FOV loses the ball in its vision but quickly finds the ball again at frame# 445 and keeps tracking it well. In this experiment, the robot knows the team member will pass the ball towards him. So when the ball is lost, the robot calculate the estimated ball position with the help of motion model (teammate actuation), trying to find the ball at the estimated position which is right beneath its kicker. Otherwise, if the robot cognition is not integrated, the robot might have great difficulty locating the ball in a short time because the actuation at frame# 431 makes the motion of the ball highly discontinuous and nonlinear. Furthermore, should the teammate announces the actuation as soon as he kicks the ball, the robot would be able to select a correct motion model and track the ball even better.

III. Team-Driven Motion Modelling

In this section, we take a multi-model tracking problem as a detailed example to show our team-driven motion modeling method. First we give an introduction of the environment and targets under the Segway soccer setup. Second, we define the communication information and we include it into the tracking. Finally, we give the multi-model regularized particle filtering algorithm.

A. Tracking Scenario

In a Segway soccer game, there are multiple moving objects on the field. e.g, the ball, the human team member and the two opponents. In this paper, we focus on the ball tracking problem.

The general parameterized state-space system for the target $\mathbf{x}_t$ at time t is given by:

$$\mathbf{x}_t = f^m(\mathbf{x}_{t-1}, \mathbf{u}^m_{t-1}, \mathbf{v}^m_{t-1}) \quad (1)$$

$$\mathbf{z}_t = h^m(\mathbf{x}_t, \mathbf{n}^m_t) \quad (2)$$

where f^m and h^m are the parameterized state transition and measurement functions for the mth model of the target; $\mathbf{x}, \mathbf{u}, \mathbf{z}$ are the state, input and measurement vectors; $\mathbf{v}, \mathbf{n}$ are the process and measurement noise vectors of known statistics; m is the model index that can take any one of N values, where N is the number of models of the target being tracked;

In our Segway RMP soccer robot environment, we define five motion models (state and measurement transition functions), namely *Free-Ball*, *Robot-Grab-Ball*, *Robot-Kick-Ball*, *Teammate-Grab-Ball*, *Teammate-Kick-Ball* to model the ball motion (for the rest of this paper, for simplicity, we use $\mathbf{x}_t$ to represent the ball state). They are similar models as introduced in [4]. Generally, *Free-Ball* is a motion model that the ball moves without external actuation. *Robot-Grab-Ball* and *Robot-Kick-Ball* are the two motion models that describe the robot's own actuation effects on the ball. *Teammate-Kick-Ball* and *Teammate-Grab-Ball* are the two motion models that describe the teammate's actuation effects on the ball.

Though we are using the similar models as [4], we have different strategy on how to infer which model to use and how to transits from one model to another. Briefly speaking, their approach assumes the robot infers teammate's tactic from team play, and assign transitional probabilities based on the assumed teammate's tactic. While we obtain teammate's tactic directly from the communication message, which will be further discussed in the following sections.

B. Communication Information

Current communication between robots is through peer-to-peer UDP sockets. Each announcement will be repeated for the several following time steps to avoid possible data loss in transimition. We define three kinds of communication messages in terms of different actuation model. Each kind of message has a unique message ID.

- *Hold*: After grabbing the ball, robot announces "hold" indicating the ball is under its kicker.
- *Shoot*: Robot announces "shoot" when it kicks the ball to the opponent's goal.
- *Pass*: Robot announces "pass" when it decides to pass the ball to its teammate.

With the communication information, robots do not need to infer teammate's tactic from the team play any more [4]. Instead, every actuation on the ball is announced to keep the ball motion updated among team members. we can use dynamic Bayesian network (DBN) to represent the whole system for the ball tracking in a natural and compact way as shown in Figure 3. In this graph, the system state is represented by variables (robot's own tactic T, communication message C, infrared sensor measurement s, ball state $\mathbf{x}$, ball motion model index m, vision sensor measurement $\mathbf{z}$). The variables change over time in discrete intervals. The edges indicate dependencies between the variables. For instance, in Figure 3 the ball motion model index m_t depends on $m_{t-1}, T_{t-1}, C_{t-1}, s_t$ and $\mathbf{x}_{t-1}$, hence there are edges coming from the latter five variables to m_t. Note that the motion model is not dependent on teammate's play or tactics, since the communication message contains all the information presented by teammate's tactics.

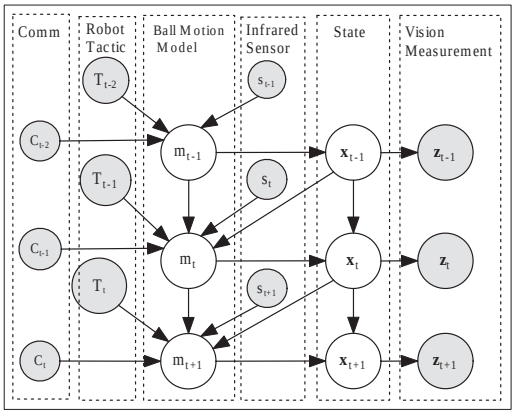

Fig. 3. A DBN for ball tracking with a Segway RMP robot.

C. Team-Driven Model Transitions

Given the communication information ($C_{t-1} = \{None, Hold, Shoot, Pass\}$), the robot can infer which motion model the ball should take. The motion model of the ball (m_t) is therefore affected by what tactic the robot (T_{t-1}) is executing and what actuation the teammate is doing.

We know that the model index m determines the present motion model being used. For our ball tracking example, $m_t = i, i = 1, \cdots, 5$. In our approach, it is assumed that the team member motion model index, m_t, conditioned on the previous tactic executed T_{t-1} by the robot, the communication message C_{t-1} which indication teammate actuation, and other useful information v_t (such as ball state and infrared sensor reading), is governed by an underlying Markov process, such that, the conditioning parameter can branch at the next time-step with probability.

$$h_{i,j} = p(m_t = i | m_{t-1} = j, T_{t-1}, C_{t-1}, v_t) \quad (3)$$

Note that $C_{t-1} = None$ indicates no message is received in the current time step. While $C_{t-1} \neq None$ indicates a message is received concerned with a teammate actuation.

IV. Multi-Model Motion Tracking

We give the ball-tracking algorithm following Figure 3. We use the sequential Monte Carlo method to track the

motion model m and the object state $\mathbf{x}$. Particle filtering is a general purpose Monte Carlo scheme for tracking in a dynamic system. It maintains the belief state at time t as a set of particles $p_t^{(1)}, p_t^{(2)}, \cdots, p_t^{(N_s)}$, where each $p_t^{(i)}$ is a full instantiation of the tracked variables $\{p_t^{(i)}, w_t^{(i)}\}$, $w_t^{(i)}$ is the weight of particle $p_t^{(i)}$ and N_s is the number of particles. In our case, $p_t^{(i)} = \langle \mathbf{x}_t^{(i)}, m_t^{(i)} \rangle$.

The equations below follow from the ball-tracking DBN.

$$m_t^{(i)} \sim p(m_t | m_{t-1}^{(i)}, \mathbf{x}_{t-1}^{(i)}, s_t, T_{t-1}, C_{t-1}) \quad (4)$$

$$\mathbf{x}_t^{(i)} \sim p(\mathbf{x}_t | m_t^{(i)}, \mathbf{x}_{t-1}^{(i)}) \quad (5)$$

Note that Equation 4 is implemented as a regime transition [8]. Also note in Equation 5, the ball state is conditioned on the ball motion model $m_t^{(i)}$ sampled from Equation 4.

Then we use the regularized particle filter to update the state estimate and resampling from a continuous approximation of the posterior density. The sampling algorithm is as follows:

$[\{\mathbf{x}_t^{(i)}, m_t^{(i)}, w_t^{(i)}\}_{i=1}^{N_s}] = \textbf{\textit{RPF}}[\{\mathbf{x}_{t-1}^{(i)}, m_{t-1}^{(i)}, w_{t-1}^{(i)}\}_{i=1}^{N_s}, \mathbf{z}_t, s_t, T_{t-1}, C_{t-1}]$

for $i = 1 : N_s$
  draw $m_t^{(i)} \sim p(m_t | m_{t-1}^{(i)}, \mathbf{x}_{t-1}^{(i)}, s_t, T_{t-1}, C_{t-1})$.
  draw $\mathbf{x}_t^{(i)} \sim p(\mathbf{x}_t | m_t^{(i)}, \mathbf{x}_{t-1}^{(i)})$.
  set $w_t^{(i)} = w_{t-1}^{(i)} \cdot p(\mathbf{z}_t | \mathbf{x}_t^{(i)})$
end for
Calculate total weight: $w = \sum[\{w_t^i\}_{i=1}^{N_s}]$
for $i = 1 : N_s$
  Normalize: $w_t^i = w_t^i / w$
end for
if $\hat{N}_{eff} < N_{thr}$
  Calculate covariance matrix S_k of $\{\mathbf{x}_k^i, w_k^i\}_{i=1}^N$
  Computer D_k such that $D_k D_k^T = S_k$
  Resample
  for $i = 1 : N_s$
    draw $\epsilon^i \sim K$ from Gaussian kernel
    $x_k^{i^*} = x_k^i + h_{opt} D_k \epsilon^i$
  end for
end if

The inputs of the algorithm are samples drawn from the previous posterior $\langle \mathbf{x}_{t-1}^{(i)}, m_{t-1}^{(i)}, w_{t-1}^{(i)} \rangle$, the present vision and infrared sensory measurement $\mathbf{z}_t, s_t$, the robot's tactic T_{t-1}, and the communication message C_{t-1}. The outputs are the updated weighted samples $\langle \mathbf{x}_t^{(i)}, m_t^{(i)}, w_t^{(i)} \rangle$. In the sampling algorithm, first, a new ball motion model index, $m_t^{(i)}$, is sampled according to Equation 4. Then given the model index, and previous ball state, a new ball state is sampled according to Equation 5. The importance weight of each sample is given by the likelihood of the vision measurement given the predicted new ball state. Finally, each weight is normalized and the samples are resampled. Then we can estimate the ball state based on the mean of all the $\mathbf{x}_t^{(i)}$. Ball motion model can be estimated based on $m_t^{(i)}$. For details about the regularized particle filter, refer to [9].

V. Experiment

In this section, we evaluate the effectiveness of our tracking system in both simulated and real-world tests.

A. Simulation Experiments

It is difficult to know the ground truth of the target's position and velocity in the real robot test. We do the simulation experiments to evaluate the performance of tracking using various metrics.

A tracking scenario with multi-agent actuators is selected to illustrate the communication-based multi-model tracking. A static robot is observing the target, which starts from the point (-36.5, -67) of the x-y plane at a speed of $0.3m/s$ and a heading of $1.31rad$. After 78 seconds, another robot actuates on the target which accelerate it for 52 seconds. The next actuation on the target begins at 130 seconds and ends at 180 seconds which finalize the simulated run. Figure 4 shows the target trajectory. Whenever an actuation happens, the robot who executes the action send message to others informing the actuation model ID. In all simulated tests, the sample time is taken as 1.0 s.

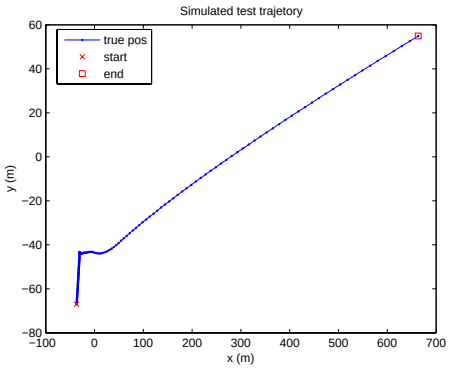

Fig. 4. Simulated target trajectory. The cross indicates the starting point, and the square indicates the ending point.

Three types of motion models are employed. The first assumes constant velocity in the Cartesian frame with small deviations in velocity $\mathbf{v}_t$ on x-y axis by zeros-mean, Gaussian white noise $\mathbf{w}_t$ with covariance Q_{cv}. The second and third model, employed to track the motion actuated by a robot, assumes constant acceleration motion on x-y axis with small deviations in acceleration $\mathbf{a}_t$ by zeros-mean, Gaussian white noise $\mathbf{w}_t$ with covariance Q_{ca}. The second model uses a high disturbance covariance, the third model uses a low disturbance covariance. The Markov transition probability matrix is defined as

$$H_1 = \begin{bmatrix} 0.95 & 0.33 & 0 \\ 0.05 & 0.34 & 0.05 \\ 0 & 0.33 & 0.95 \end{bmatrix}$$

and the initial probabilities are set to

$$p(m_0) = \begin{bmatrix} 0.6 & 0.2 & 0.2 \end{bmatrix}^T$$

When there is no communication about actuation model, H_1 is the only transition matrix we use. When communication

is enabled, we have two more transition matrix defined corresponding to receiving message of actuation model 2 and 3 respectively, where

$$H_2 = \begin{bmatrix} 0.05 & 0.03 & 0.05 \\ 0.9 & 0.9 & 0.85 \\ 0.05 & 0.07 & 0.1 \end{bmatrix}, H_3 = \begin{bmatrix} 0.05 & 0.03 & 0.02 \\ 0.15 & 0.2 & 0.1 \\ 0.8 & 0.77 & 0.88 \end{bmatrix}$$

In all scenarios, observation are taken of the Cartesian position of the target, corrupted by zero-mean, Gaussian noise with variance R = 0.01 on each axis.

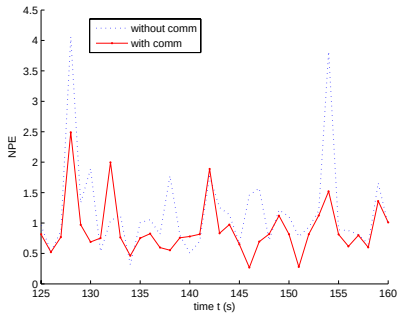

Fig. 5. Variation in NPE with and without communication information.

We simulate the experiment for 50 runs. For comparison purposes, the normalized position error (NPE), defined as the ratio of the mean-square position estimation error to the mean-square measurement error over M simulations [8] is applied. Figure 5 plots the variation in NPE for tracking with and without communication. With communication enabled, the position estimation is better, which is expected because the estimated motion model is more precise with communication enabled. To illustrate the superior tracking of the tracking with communication, Figure 6 and 7 plots the corresponding model weighting from each tracker. We can clearly see that the model transition in Figure 7 exactly describes what happens in real-time (motion model transits from 1 to 2, then 2 to 3).

B. Real Robot Test

In the real-world test, we do experiments on the Segway RMP soccer robot executing the $Catch$ tactic to receive ball from robot teammate. The teammate robot is executing the $GrabKickToTeammate$ tactic. When the kick motion is done, the teammate announces the "pass" message through peer-to-peer communication. We implement multi-model tracker with and without including communication information to compare the performance.

Figure 8 plots the ball speed estimation results from each tracker in one of the experiment. As is shown in the figure, the estimation without communication lags from the true value about 0.5 s, while the estimation with communication is well predicted because the announcement is received right after the actuation is made. To illustrate the superior tracking of the tracking with communication, Figure 9 and 10 plots the corresponding model weighting from each tracker. We can clearly see that the model transition in Figure 7 exactly

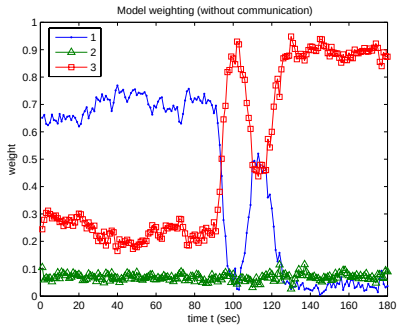

Fig. 6. Model weightings when communication is disabled.

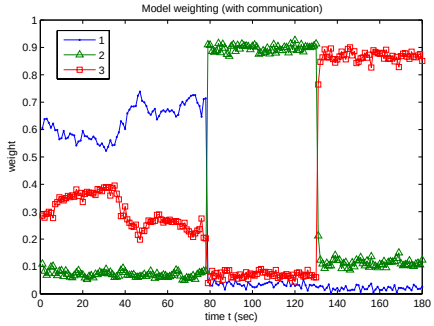

Fig. 7. Model weightings when communication is enabled.

describes the real world testing scenario in which motion model transits from *Free-Ball* to *Teammate-Kick-Ball*, then *Free-Ball* and at last *Robot-Grab-Ball*.

VI. Related Work

There are several approaches incorporating some kind of prior knowledge related to the general problem of tracking under no actuation. For example, hard constraints on target position, speed or acceleration have been considered in tracking problems to improve tracking performance [10]. This kind of information is simple and easy to be represented as a truncated density. The only thing we need to do is to sample from a truncated density using rejection sampling technique.

Another example is the situation where a number of targets are moving in formation, and there is a strong dependency between the individual sensor measurements, which provides valuable information on target behavior. Actually this problem can be modelled as independent individual target motions superimposed on a common group effect. A model of this type was introduced in [11], in which the motion of the group and disposition of the measurement sources relative to the group are modelled as two separate components.

In terrain-aided tracking problem, using the ground moving target indicator (GMTI), one may have some prior information of the terrain, road maps, and visibility conditions [12]. The algorithm is referred to as the variable structure multiple-model particle filter, since it adaptively selects a subsets of modes that are active at a particular time. This approach outperformed normal tracking method without integrating the

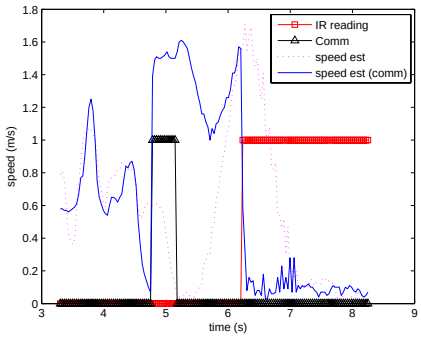

Fig. 8. Ball speed estimation results from each tracker.

prior information due to the better dynamics models, which capture the motion dynamics with terrain information in an intricate but accurate manner.

All the above approaches deal with the problem of tracking under no actuation. The first effort reporting a tracking approach concerned with our problem of actuation over tracked objects is presented in [2]. In this approach, the author acclaims that the physical interaction between observer and target is one of the difficult issues. One of the novel contributions of this approach is its idea to model different interactions between the ball and the environment. The transition probabilities between the states are fixed. Their values were tuned manually. By additionally sampling the non-linear parts (robot motion and ball-environment interaction) of the observer motion, the target and its motion can be estimated accurately using Kalman filters conditioned on these samples. Then a tracking algorithm incorporating single and multi-robot control and coordination knowledge is proposed by [3] and [4], in which the transition probabilities between the states are dependent on the robot's tactic and team plan. If the teammate is a human, not a robot, the certainty that the teammate is executing the expected play or tactic could be reduced. That is, the human teammate could fail to execute the desired play or tactic, which cause the motion model to be unprecise.

In this paper, communication information between multiple agents is recognized as another source of prior knowledge. We integrate this information to our team-driven multi-model motion tracking scheme and obtain better performance.

VII. Conclusions

Motivated by the interactions between a team and the tracked object, we contribute a method to achieve efficient tracking through using a communication-based motion model and combined vision and infrared sensory information. This method gives the robot a more exact task-specific motion model when executing different tactics over the tracked object. Then we represent the system in a compact dynamic Bayesian network and use particle filter to keep track of the motion model and target state through sampling. The empirical results from the simulated and the real experiments show the efficiency of the multi-model tracking when communication is enabled and integrated.

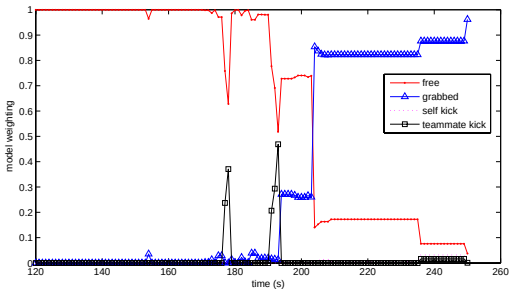

Fig. 9. Model weightings when communication is disabled.

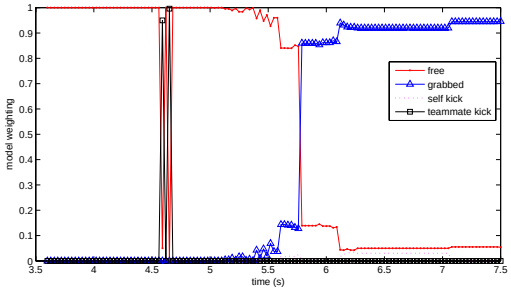

Fig. 10. Model weightings when communication is enabled.

References

[1] D. Schulz, W. Burgrad, and D. Fox, "People tracking with mobile robots using sample-based joint probabilistic data association filters," *International Journal of Robotics Research*, vol. 22(2), 2003.

[2] C. Kwok and D. Fox, "Map-based multiple model tracking of a moving object," *Proceedings of eight RoboCup International Symposium*, July 2004.

[3] Y. Gu, "Tactic-based motion modelling and multi-sensor tracking," *Proceedings of Twentieth National Conference on Artificial Intelligence*, 2005.

[4] Y. Gu and M. Veloso, "Multi-model tracking using team actuation models," in *Proceedings of the 2006 International Conference on Robotics and Automation*, 2006.

[5] K.-C. Jim and C. L. Giles, "How communication can improve the performance of multi-agent systems," in *AGENTS '01: Proceedings of the fifth international conference on Autonomous agents*. New York, NY, USA: ACM Press, 2001, pp. 584–591.

[6] B. Browning, J. Searock, P. E. Rybski, and M. Veloso, "Turning segways into soccer robots," *Industrial Robot*, vol. 32, no. 2, pp. 149–156, 2005.

[7] B. Browning, J. Bruce, M. Bowling, and M. Veloso, "Stp: Skills, tactics and plays for multi-robot control in adversarial environments," *IEEE Journal of Control and Systems Engineering*, vol. 219, pp. 33–52, 2005.

[8] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter, Particle Filters for Tracking Applications*. Artech House Publishers, 2004.

[9] A. Doucet, N. D. Freitas, and N. Gordon, Eds., *Sequential Monte Carlo Methods in Practice*. New York: Springer-Verlag, 2001.

[10] L.-S. Wang, Y.-T. Chiang, and F.-R. Chang, "Filtering method for nonlinear systems with constraints," *IEEE Proc. - Control Theory and Appl.*, pp. 525–531, November 2002.

[11] M. Morelande and S.Challa, "An algorithm for tracking group targets," *Proc. Workshopp on Multiple Hypothsis Tracking: A Tribute to S.Blackman*, May 2003.

[12] M. Arulampalam, N. Gordon, M. Orton, and B. Ristic, "A variable structure multiple model particle filter for gmti tracking," *Proc. 5th Int. Conf. Information Fusion*, July 2002.