Multi-Robot Tracking of a Moving Object Using Directional Sensors

Manuel Mazo Jr., Alberto Speranzon, Karl H. Johansson
Dept. of Signals, Sensors & Systems
Royal Institute of Technology
SE-100 44 Stockholm, Sweden
manuelme@ieee.org, {albspe,kallej}@s3.kth.se

Xiaoming Hu
Dept. of Mathematics
Royal Institute of Technology
SE-100 44 Stockholm, Sweden
hu@math.kth.se

Abstract— The problem of estimating and tracking the motion of a moving target by a team of mobile robots is studied in this paper. Each robot is assumed to have a directional sensor with limited range, thus more than one robot (sensor) is needed for solving the problem. A sensor fusion scheme based on inter-robot communication is proposed in order to obtain accurate real-time information of the target's position and motion. Accordingly a hierarchical control scheme is applied, in which a consecutive set of desired formations is planned through a discrete model and low-level continuous-time controls are executed to track the resulting references. The algorithm is illustrated through simulations and on an experimental platform.

I. INTRODUCTION

Multi-robot systems are used in many situations in order to improve performance, sensing ability and reliability, in comparison to single-robot solutions. For example, in applications like exploration, surveillance and tracking, we want to control a team of robots to keep specific formations in order to achieve better overall performance. Formation control is a particularly active area of multi-robot systems, e.g., [1], [2], [3], [4], [5]. Most of the work in the literature, however, is focused on the problem of designing a controller for maintaining a preassigned formation. The issue of sensors limitations seems to be more or less overlooked. In this paper we consider the problem of localizing and tracking a moving object using directional sensors that are mounted on mobile robots. When the range is far and the resolution becomes low, many visual sensors are in effect reduced to only directional sensors, since the depth information is then hard to recover. In situations like this, sensing and estimation become central and an integrated solution for control and sensing desirable. Since we need to have sufficient separation for the sensor channels, it is reasonable that we mount the sensors on different robots. Thus the goal of the control design is not only that the robots should track the target, but also that the robots should coordinate their motion so that the sensing and localizing of the target is not lost.

The main contribution of our work is a hierarchical algorithm for localizing and tracking a moving target. We study a prototype of a distributed mobile sensing system, namely, a team of two nonholonomic robots with sensors for the environment. These tracking robots have hard sensor constraints, as they can just obtain relative angular position of the target within a limited field of vision, and relative positions of each other within short distance. Our solution provides a cooperative scheme in which the higher level of the algorithm plans a formation for the robots to follow in order to track the target. The robots exchange sensor information to estimate the target's position by triangulation. In particular, since the motion of the target is unknown and thus can not be planned, the motion planning for the formation must be done on-the-fly and based on the actual sensor readings, which is quite different from many formation control algorithms in the literature where all agents' motion can be planned. In the lower level of our algorithm, for each robot we use a tracking controller that is based on the so-called virtual vehicle approach [6], which turns to be quite robust with respect to uncertainties and disturbances.

The outline of the paper is as follows. The problem formulation for collaborative tracking is presented in Section II. The hierarchical solution is described in Section III. Supporting simulation results are shown in Section IV, and ongoing experimental work is presented in Section V. The conclusions are given in Section VI.

II. PROBLEM FORMULATION

Consider two robots tracking a moving target as shown in Figure 1. The robots are positioned at (x_1, y_1) and (x_2, y_2), respectively, while the target is at (x_T, y_T). The robots are modeled as two unicycles

$$\dot{x}_i = v_i \cos \theta_i$$
$$\dot{y}_i = v_i \sin \theta_i \qquad i = 1, 2$$
$$\dot{\theta}_i = \omega_i$$

with controls v_i and ω_i. The motion of the target is not a priori given. Each robot has a directional sensor, which provides an estimate of the direction $\alpha_i = \beta_i - \theta_i$, $i = 1, 2$, to the target from robot i, where

$$\beta_i = \arctan \frac{y_T - y_i}{x_T - x_i}$$

From the estimates of α_1, α_2 and the robot positions, estimates of target position and velocity are derived. Note that $\alpha_i = 0$ corresponds to the target being in the heading direction θ_i of robot i. The sensors are assumed to have a constant limited
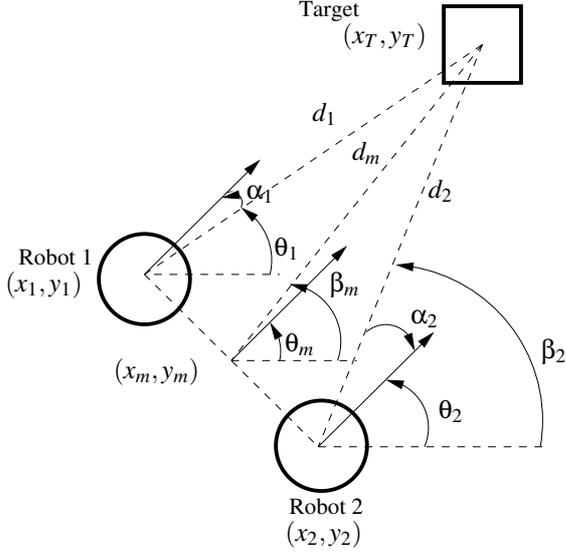
Fig. 1. Two robots tracking a target under constrained angular sensing.

angular range of $\alpha_{\max} \in (0, \pi)$, so that the estimated angle to the target from robot i is given by

$$\hat{\alpha}_i = \begin{cases} \alpha_i, & |\alpha_i| \leq \alpha_{\max} \\ \infty, & \text{otherwise} \end{cases}$$

where $\hat{\alpha}_i = \infty$ denotes that the target is out of range. We introduce the distance $d_i = \|(x_T, y_T) - (x_i, y_i)\|$, $i = 1, 2$. Suppose the robot positions (x_i, y_i) and headings θ_i, $i = 1, 2$, are known and that the angles α_i, $i = 1, 2$, are within the sensor range, then the target position can be estimated as

$$\begin{pmatrix} \hat{x}_T \\ \hat{y}_T \end{pmatrix} = \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} + \hat{d}_1 \begin{pmatrix} \cos \beta_1 \\ \sin \beta_1 \end{pmatrix}$$

or, equivalently, as

$$\begin{pmatrix} \hat{x}_T \\ \hat{y}_T \end{pmatrix} = \begin{pmatrix} x_2 \\ y_2 \end{pmatrix} + \hat{d}_2 \begin{pmatrix} \cos \beta_2 \\ \sin \beta_2 \end{pmatrix}$$

where the distance estimates are given by

$$\begin{pmatrix} \hat{d}_1 \\ \hat{d}_2 \end{pmatrix} = \begin{pmatrix} -\cos \beta_1 & \cos \beta_2 \\ -\sin \beta_1 & \sin \beta_2 \end{pmatrix}^{-1} \begin{pmatrix} x_1 - x_2 \\ y_1 - y_2 \end{pmatrix}$$

provided that the inverse exists. The estimation problem is thus how to obtain a good estimate $(\hat{x}_T, \hat{y}_T)$, under the constraints on the directional sensors. An example of a directional sensor is the linear video sensor of the Khepera II robot as illustrated in Figure 9 and further described in Section V.

Remark: For the sake of simplicity, we present all the variables in a global or inertially fixed coordinate system. We can easily recast our results in the moving frame fixed on the target. By this way, we would not need to know the global coordinates of the robots.

Given that the robots follow the target on a certain distance, there is a trade-off between the robustness of tracking on the target (for both sensors or robots) and the robustness of estimates. When the robots are very close to each other, the

target will be safely within the sensor "view field" but the estimate will be very sensitive to measurement inaccuracies since the two directions toward the target are almost parallel; When the robots are very far away from each other, any motion of the target can lead to no angular measurements since it will be outside the view field. However, if the sensor readings are available in this case, the target position estimate by triangulation will be quite robust to measurement errors.

III. HIERARCHICAL TRACKING ALGORITHM

The trade-off between having guaranteed position estimates of the target and having accurate estimates leads to imposing a desired formation for the two robots following the target. The formation is chosen such that the target is within the angular limits of the robot sensors, and the sensor readings give a well-conditioned estimate of (x_T, y_T). The evolution of the formation is defined in a discrete set of points, while lower-level continuous-time controls make the robots tracking the formation. The resulting hierarchical control structure has an upper level in which the evolution of the formation is updated at discrete events and a lower level dedicated to the tracking by the robots of the waypoints defined by the formation. This section describes both the high-level formation planning and the low-level tracking control in detail.

A. Formation Planning

The desired robot formation is shown in Figure 1. The distance between the robots at (x_1, y_1) and (x_2, y_2) is denoted p, and the distances to the target are d_1 and d_2. The desired orientation of the two robots is fixed and corresponds to that the robot headings should be perpendicular to the axis that connects the two robots. Thus the formation is maintained as long as $\dot{p} = \dot{d}_1 = \dot{d}_2 = 0$ and $\dot{\theta}_1 = \dot{\theta}_2$. Let (x_m, y_m) denote the point half-way between the robots. Then the position of this point together with the orientation of the axis θ_m decides the state of the formation. The dynamics of these variables are described by

$$\dot{x}_m = v_m \cos \theta_m$$
$$\dot{y}_m = v_m \sin \theta_m$$
$$\dot{\theta}_m = \omega_m$$

Let

$$v_m = \frac{\dot{x}_T \cos \beta_m + \dot{y}_T \sin \beta_m}{\cos \alpha_m}$$
$$\omega_m = \frac{v_m \sin \alpha_m + \dot{y}_T \cos \beta_m - \dot{x}_T \sin \beta_m}{d_m}$$

with

$$\beta_m = \arctan \frac{y_T - y_m}{x_T - x_m}, \qquad \alpha_m = \beta_m - \theta_m$$

It is easy to show that if $v_1 = v_m + \omega_m p/2$, $v_2 = v_m - \omega_m p/2$, and $\omega_1 = \omega_2 = \omega_m$, then $\dot{p} = \dot{d}_1 = \dot{d}_2 = 0$ and $\dot{\alpha}_1 = \dot{\alpha}_2 = 0$, i.e., that the desired formation is maintained. The evolution of the formation is updated at discrete time instances t_k, $k = 0, 1, \ldots$. Given an estimate of the target position at time t_k, the target

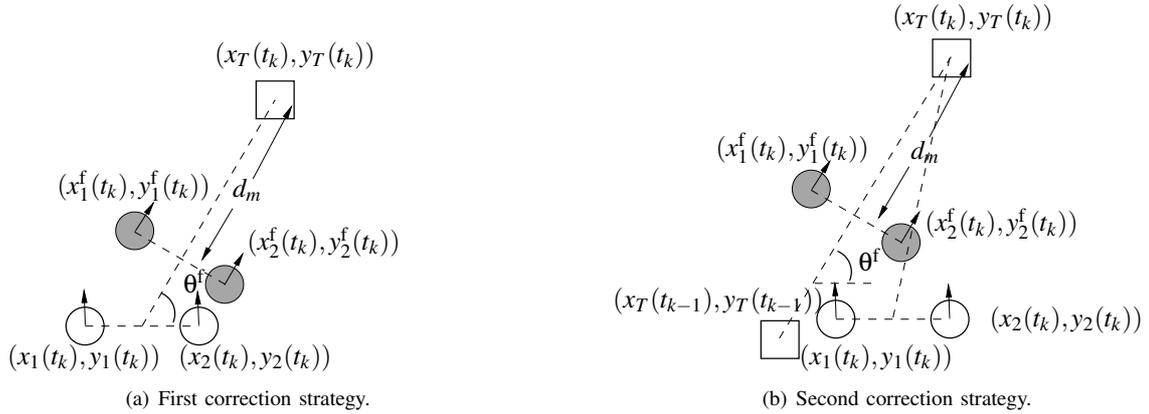(a) First correction strategy. (b) Second correction strategy.

Fig. 2. Correction strategies for the initial points at each iteration in the Formation Control level.

position at t_{k+1} is estimated. We suppose that no model of the target is available, so a simple estimate is linear extrapolation

$$\hat{x}_T(t_{k+1}) = \hat{x}_T(t_k) + (t_{k+1} - t_k)\hat{v}_T(t_k)\cos\hat{\theta}_p(t_k)$$
$$\hat{y}_T(t_{k+1}) = \hat{y}_T(t_k) + (t_{k+1} - t_k)\hat{v}_T(t_k)\sin\hat{\theta}_p(t_k)$$

where

$$\hat{v}_T(t_k) = \frac{\|(\hat{x}_T(t_k),\hat{y}_T(t_k)) - (\hat{x}_T(t_{k-1}),\hat{y}_T(t_{k-1}))\|}{t_k - t_{k-1}}$$
$$\hat{\theta}_T(t_k) = \arctan\frac{\hat{y}_T(t_k) - \hat{y}_T(t_{k-1})}{\hat{x}_T(t_k) - \hat{x}_T(t_{k-1})}$$

The reference path provided to the low-level motion control is given by trajectories generated from the controls v_i, ω_i, $i = 1, 2$, where v_m and ω_m are continuous-time controls defined over (t_k, t_{k+1}), such that the corresponding way point for $(x_m(t_{k+1}), y_m(t_{k+1}))$ is reached. If v_i, ω_i, $i = 1, 2$, are constant over an interval, they generate the following reference trajectories:

$$x_i^{\text{ref}}(t) = x_i^{\text{f}}(t_k) + \frac{v_i(t_k)}{\omega_i(t_k)}\left[\sin(\theta_i^{\text{f}}(t_k) + \omega_i(t_k)t) - \sin(\theta_i^{\text{f}}(t_k))\right]$$
$$y_i^{\text{ref}}(t) = y_i^{\text{f}}(t_k) - \frac{v_i(t_k)}{\omega_i(t_k)}\left[\cos(\theta_i^{\text{f}}(t_k) + \omega_i(t_k)t) - \cos(\theta_i^{\text{f}}(t_k))\right]$$

The initial reference points at each step $(x_i^{\text{f}}(t_k), y_i^{\text{f}}(t_k), \theta_i^{\text{f}}(t_k))$, $i = 1, 2$, are set to fulfill the desired formation. Those points are calculated based on the estimations of the moving target from sensors readings. Two different strategies are proposed for choosing those points. In the first, shown in Figure 2(a), we compute the new initial reference points as

$$\theta_i^{\text{f}}(t_k) = \arctan\frac{y_T(t_k) - y_m(t_k)}{x_T(t_k) - x_m(t_k)}$$
$$x_i^{\text{f}}(t_k) = x_m^{\text{f}}(t_k) \pm p^{\text{f}}\cos(\theta_i^{\text{f}}(t_k))$$
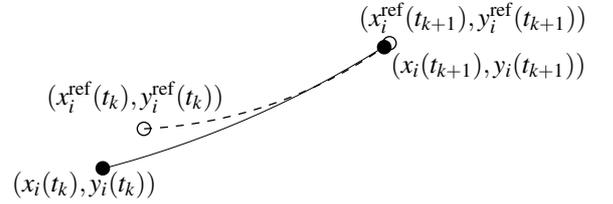$$y_i^{\text{f}}(t_k) = y_m^{\text{f}}(t_k) \mp p^{\text{f}}\sin(\theta_i^{\text{f}}(t_k))$$

Fig. 3. Tracking of a trajectory generated between t_k and t_{k+1}.

while the second one, see Figure 2(b), we use the following equations

$$\theta_i^{\text{f}}(t_k) = \arctan\frac{y_T(t_k) - y_T(t_{k-1})}{x_T(t_k) - x_T(t_{k-1})}$$
$$x_i^{\text{f}}(t_k) = x_m^{\text{f}}(t_k) \pm p^{\text{f}}\cos(\theta_i^{\text{f}}(t_k))$$
$$y_i^{\text{f}}(t_k) = y_m^{\text{f}}(t_k) \mp p^{\text{f}}\sin(\theta_i^{\text{f}}(t_k))$$

where $(x_m(t_k), y_m(t_k))$ and $(x_m^{\text{f}}(t_k), y_m^{\text{f}}(t_k))$ are defined as

$$\begin{pmatrix}x_m(t_k)\\y_m(t_k)\end{pmatrix} = \frac{1}{2}\left[\begin{pmatrix}x_1(t_k)\\y_1(t_k)\end{pmatrix} + \begin{pmatrix}x_2(t_k)\\y_2(t_k)\end{pmatrix}\right]$$

$$\begin{pmatrix}x_m^{\text{f}}(t_k)\\y_m^{\text{f}}(t_k)\end{pmatrix} = \begin{pmatrix}\hat{x}_T(t_k)\\\hat{y}_T(t_k)\end{pmatrix} - d_m^{\text{f}}\begin{pmatrix}\cos(\theta_i^{\text{f}}(t_k))\\\sin(\theta_i^{\text{f}}(t_k))\end{pmatrix}$$

and d_m^{f} and p^{f} depend on the desired formation.

Remark: Note that using the second strategy here proposed would require a global or inertially fixed coordinate system. In the following we use the first proposed strategy to compute the initial reference points.

B. Tracking Control

The reference trajectories $(x_i^{\text{ref}}, y_i^{\text{ref}})$, $i = 1, 2$, generated by the high-level formation planning are tracked by the robots using the virtual vehicle approach [6]. The idea is to let the motion of a reference point on the reference path $(x_i^{\text{ref}}(t), y_i^{\text{ref}}(t))$, $t \in (t_k, t_{k+1})$, be governed by a differential equation with error feedback, see Figure 3. Consider a reference trajectory $(x_i^{\text{ref}}(t), y_i^{\text{ref}}(t))$, $i = 1, 2$, on an interval (t_k, t_{k+1}). Parameterize
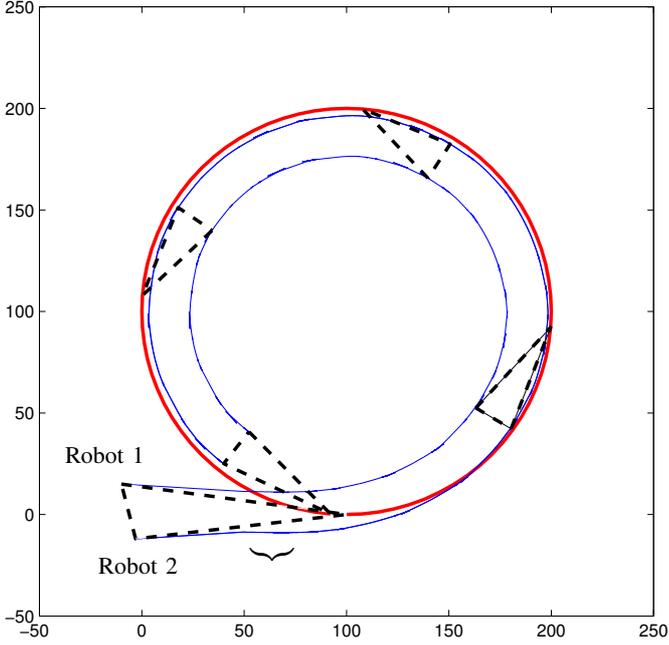
Fig. 4. Robot formation tracking a target that follows a circle. The target is shown as a thick line and the robots' trajectories as thin. Despite the errors in the initial positions of the robots, they converge to the desired formation after a few updates of the formation. Note the difference in shape between the initial formation and the following four.

the trajectories as $p_i(s_i) := x_i^{\mathrm{ref}}(s_i), q_i(s_i) := y_i^{\mathrm{ref}}(s_i)$, $i = 1,2$, where $\dot{s}_i$ is chosen as

$$\dot{s}_i = \frac{ce^{-a\rho}v_i^0}{\sqrt{p_i'^2(s_i) + q_i'^2(s_i)}}$$

where v_i^0 is the desired speed at which one wants robot i to track its path (a natural choice is $v_i^0 = v_i(t_k)$), and a and c are appropriate positive constants.

The low-level tracking control for robot i, $i = 1,2$, is given by

$$v_i(t) = \gamma \rho_i(t) \cos[\phi_i^d(t) - \theta_i(t)]$$
$$\omega_i(t) = k[\phi_i^d(t) - \theta_i(t)] + \dot{\phi}_i^d(t)$$

where γ and k are positive tuning parameters, and

$$\rho_i(t) = \sqrt{(x_i^{\mathrm{ref}}(s_i) - x_i(t))^2 + (y_i^{\mathrm{ref}}(s_i) - y_i(t))^2}$$
$$\phi_i^d(t) = \arctan \frac{x_i^{\mathrm{ref}}(s_i) - x_i(t)}{y_i^{\mathrm{ref}}(s_i) - y_i(t)}$$

Note that when $\rho_i(t) = 0$, the angle $\phi_i^d(t)$ is not well-defined. This problem can be easily solved by re-defining ϕ_i^d when ρ_i is small, see [7].

IV. SIMULATION RESULTS

The hierarchical control algorithm developed in previous section is now evaluated through a simulated example. We let the target follow a circular trajectory and the robots have an
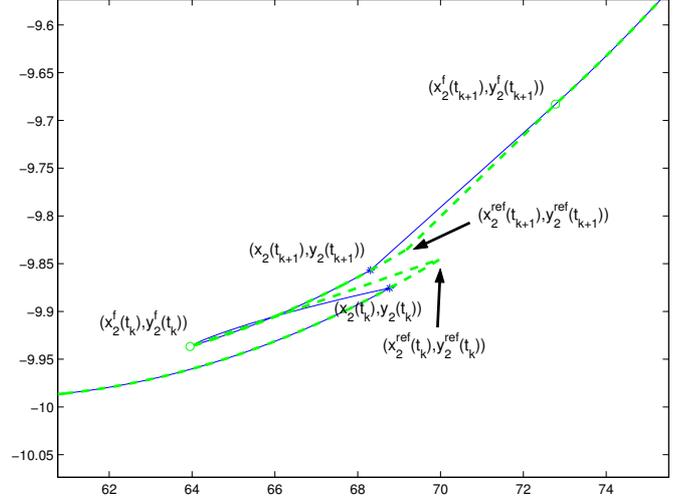

Fig. 5. A closer look to the trajectory of Robot 2 in the region marked with a big under-brace in Figure 4. The circles denote $(x_i^{\mathrm{f}}(t_k), y_i^{\mathrm{f}}(t_k))$, asterisks denote $(x_i(t_k), y_i(t_k))$ and the arrows point to $(x_i^{\mathrm{ref}}(t_k), y_i^{\mathrm{ref}}(t_k))$. Note that the scales of the x-axis and y-axis are different.

error in the measurement of β_i with uniform distribution in $(-0.02, 0.02)$. Figure 4 shows the target as a thick line and with thin lines the trajectories of the robots. Note that despite errors in the initial positions of the robots, they converge to the desired formation after a few formation updates. The formation at the initial state and at four other locations are indicated with dashed triangles.

Figure 5 shows a closer look of one section of the trajectory of Robot 2. The zoom is taken in the beginning of the trajectory, and thus the robot is trying to converge to the high-level planner reference. The circles in Figure 5 indicate the corrected initial points $(x_i^{\mathrm{f}}(t_k), y_i^{\mathrm{f}}(t_k))$ at each step of the high-level planning algorithm. The stars show the final position of the tracker robots after each step, denoted $(x_i(t_k), y_i(t_k))$. The arrows are pointing at the planned final points $(x_i^{\mathrm{ref}}(t_k), y_i^{\mathrm{ref}}(t_k))$ of the trajectory at the previous step. Figure 6 shows the evolution of target distance d_2 and target angle α_2 for Robot 2, corresponding to the first quarter of circle in Figure 4. The estimated and the real position of the target is shown in Figure 7.

V. EXPERIMENTAL RESULTS

The new multi-robot algorithm presented in the paper was experimentally evaluated on a team of two Khepera II robots [8] tracking a target LEGO Mindstorms robot. The setup is illustrated in Figure 8. Khepera II is a small self-contained wheeled robot with micro-processor and basic sensors (infra-red proximity sensors and encoders). Its diameter is about 70 mm, it has a precise odometry and a linear speed in the range of 0.02–1.00 m/s. Our Khepera II robots are provided with a linear video sensor, see Figure 9. It gives a directional measurement up to a distance of about 25 cm. The sensor consists of a linear light-sensitive array of 64×1 pixels, which gives an image with 256 gray levels. A major
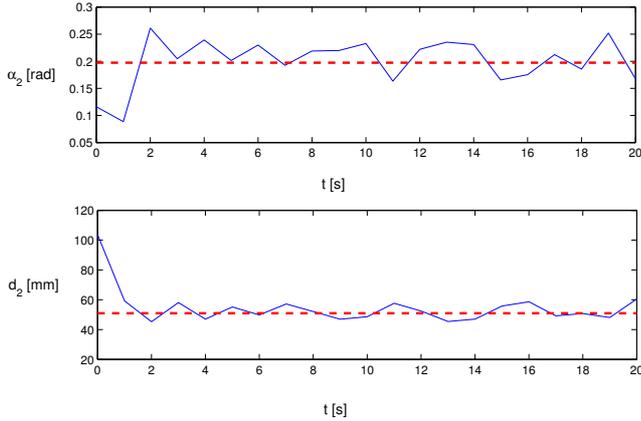
Fig. 6. Evolution of d_2 and α_2 (Robot 2) corresponding to the first quarter of circle in Figure 4. The dashed thick line shows the desired values of (d_2, α_2) to fulfill the desired formation.
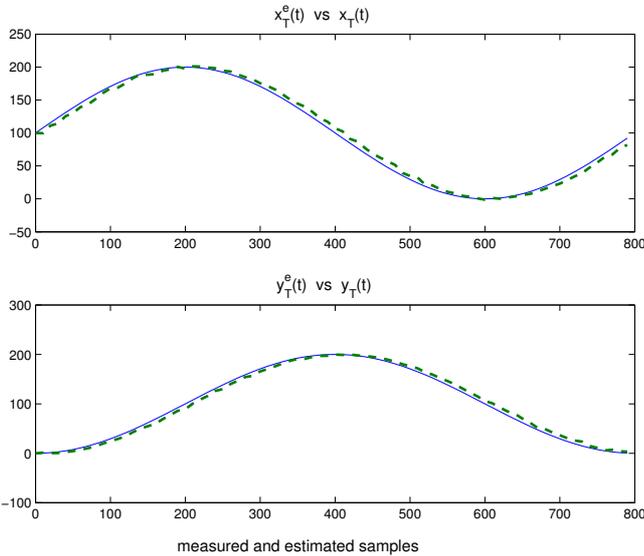

Fig. 7. Comparison of the position of the target estimated by the robots based on their sensor measurements (x_T^e, y_T^e) (thick dashed line) vs. the real position of the target (x_T, y_T) (thin line).
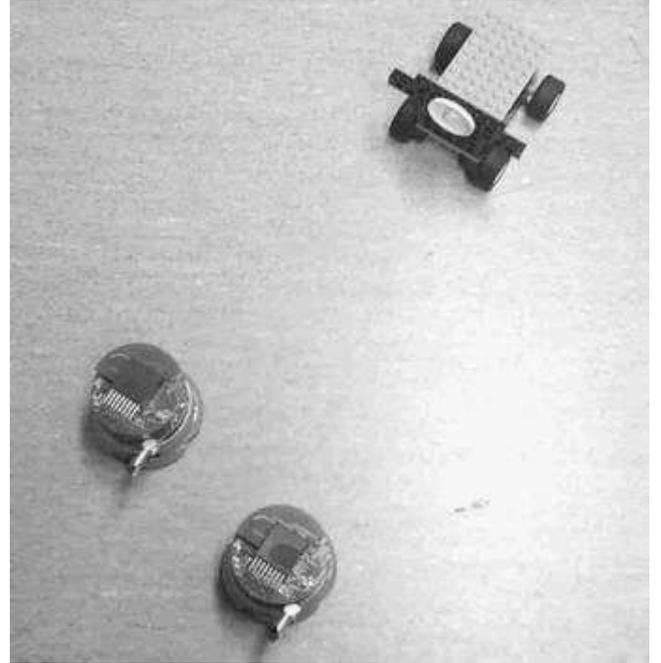

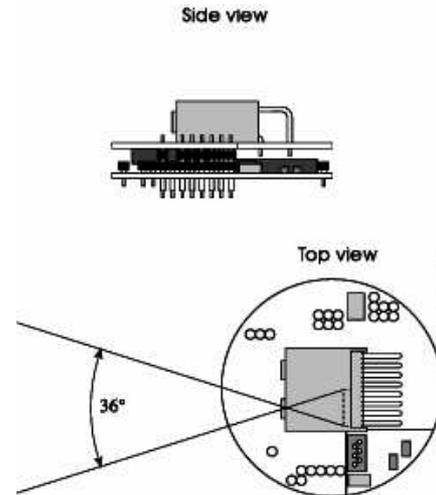Fig. 8. Two Khepera II robots tracking a LEGO Mindstorms robot.

Fig. 9. Khepera II linear vision sensor. The angular range is limited to $\alpha_{\max} = \pi/10$. *(Illustration from [8].)*

constraint of the linear video sensor is a limited horizontal field of view of 36 degrees, i.e., the angular range is limited to $\alpha_{\max} = \pi/10$. For data exchange, the two robots use an on-board radio communication system.

An multi-robot tracking experiment is shown in Figure 10. The target is following a smooth curve. As indicated in the figure, the desired formation is reached already after one step of the high-level planner. Figure 11 shows a closer look on how the formation is kept by the robots following the trajectories generated by the path-planner. Finally, Figure 12 presents a closer look on one of the Khepera II robots following the planned trajectory using the controller presented in Section III.

VI. CONCLUSIONS

Multi-robot estimation and tracking of a moving target was discussed in the paper. An integrated approach to sensing and control of two robots was presented, when each robot is equipped with a directional sensor with limited angular range. The sensor readings were fused in order to get an estimate of the targets motion. A hierarchical control strategy was developed and tested, in which high-level commands were issued to plan a series of desired formations for the robots. Low-level tracking of paths connecting waypoints defined by the formations was specified according to the recent virtual vehicle method. Ongoing work includes a systematic treatment of communication limitations in the system.
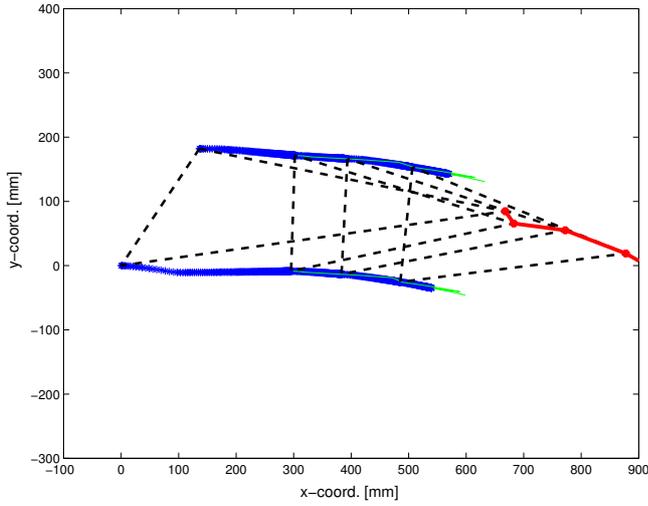
Fig. 10. An experiment showing two Khepera II robots tracking a LEGO Mindstorms robot moving on a smooth trajectory. The lighter thick line represents the estimated trajectory of the LEGO robot, while the thicker dark lines the trajectories of the tracking robots. The evolution of the formation is shown by thick dashed triangles.
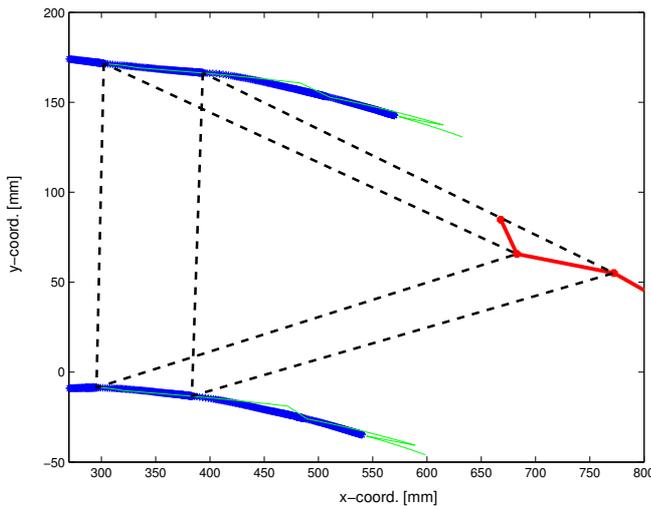

Fig. 11. Zoom of Figure 10 showing how the formation evolves and is kept between two steps of the high-level planner.
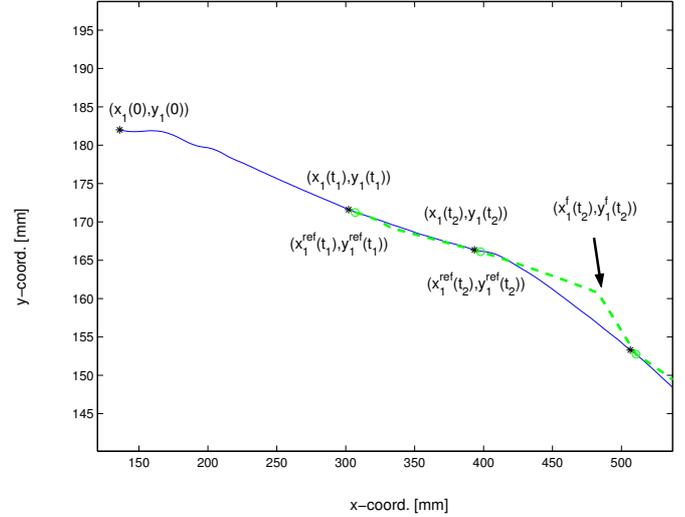

Fig. 12. Closer look of Figure 10 showing how one of the Khepera II robots follows the trajectory derived by the formation planner. The figure presents the planned path as a dashed line and as a thin line the robot actual motion. Circles mark the desired ending points from the planned path, and asterisks mark the position of the points when the robot estimates the position of the target. Corrected points are indicated by an arrow.

REFERENCES

[1] T. Balch and R. Arkin, "Behavior-based formation control for multi-robot teams," *IEEE Transaction on Robotics and Automation*, vol. 14, no. 6, pp. 926–938, 1998.

[2] B. Dunbar and R. Murray, "Model predictive control of coordinated multi-vehicle formations," in *IEEE Conference on Decision and Control*, 2002.

[3] P. Ögren, E. Fiorelli, and N. Leonard, "Formations with a mission: Stable coordination of vehicle group maneuvers," in *International Symposium on Mathematical Theory of Networks and Systems*, 2002.

[4] N. Leonard and E. Fiorelli, "Virtual leaders, artificial potentials and coordinated control of groups," in *IEEE Conference on Decision and Control*, 2001, pp. 2968–2973.

[5] H. G. Tanner, G. J. Pappas, and V. Kumar, "Leader-to-formation stability," in *IEEE Transactions on Robotics and Automation*, 2004,(to appear).

[6] M. Egerstedt, X. Hu, and A. Stotsky, "Control of mobile platforms using a virtual vehicle approach," *IEEE Transaction on Automatic Control*, vol. 46, no. 11, pp. 1777–1782, 2001.

[7] X. Hu, , D. Fuentes, and T. Gustavi, "Sensor-based navigation coordination for mobile robots," in *IEEE Conference on Decision and Control*, 2003.

[8] K-Team S.A. [Online]. Available: http://www.k-team.com