

Multi-Robot Tracking of a Moving Object Using Directional Sensors

Xiaoming Hu[†], Karl H. Johansson[‡], Manuel Mazo Jr.[‡], Alberto Speranzon[‡]

[‡]Dept. of Signals, Sensors & Systems

Royal Institute of Technology, SE-100 44 Stockholm, Sweden

[†]Optimization and Systems Theory

Royal Institute of Technology, SE-100 44 Stockholm, Sweden

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 position and motion of the target. Accordingly a hierarchical control scheme is developed, 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 experiments

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. 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.

Since we need to have sufficient separation for the sensor channels, it is reasonable that we mount the sensors on different robots. Since one sensor is obviously not enough to localize the target, a network of sensors is needed. 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. In situations like this, sensing and estimation become central and an integrated solution of control for sensing, or active sensing, is a must.

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 network, namely, a team of nonholonomic robots with exteroceptive sensors. 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 position of the target 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. We should emphasize that in order to implement our control algorithms, only local information is needed.

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 some preliminary experimental work is presented in Section V. The conclusions are given in Section VI.

II. PROBLEM FORMULATION

Consider N robots tracking a moving target as shown in Figure 1 where $N = 2$. The robots are positioned at (x_i, y_i) $i = 1, \dots, N$, respectively, while the target is at (x_T, y_T) . The motion of the target is not a priori given. Each robot has a directional sensor, which provides an estimate of the direction $\alpha_i = \beta_i - \theta_i$, $i = 1, \dots, N$, to the target from robot i , where

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

From the estimates of α_i and the relative positions of the robots, estimates of the target position and velocity relative to the robots can be 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 angular range of $\alpha_{\max} \in (0, \pi)$, so that the estimated angle to the target from

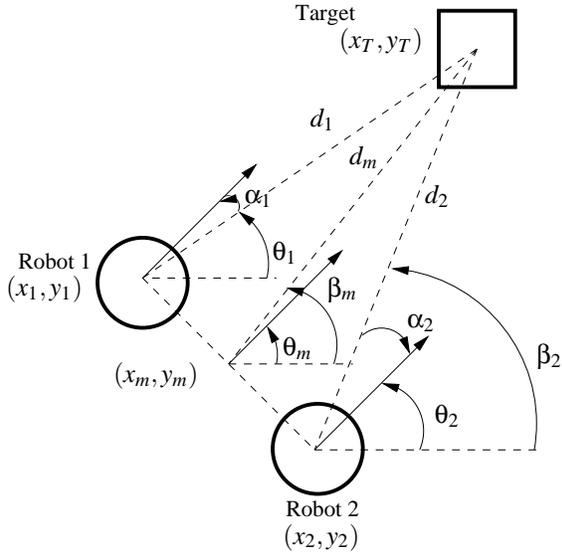


Fig. 1. Two robots tracking a target under constrained angular sensing.

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 $d_i = \|(x_T, y_T) - (x_i, y_i)\|$, $i = 1, \dots, N$. Suppose that the angles α_i , $i = 1, \dots, N$, are within the sensor range, then the target position is related to the global coordinates of the robot as

$$\begin{pmatrix} \hat{x}_T \\ \hat{y}_T \end{pmatrix} = \begin{pmatrix} x_i \\ y_i \end{pmatrix} + d_i \begin{pmatrix} \cos \beta_i \\ \sin \beta_i \end{pmatrix}, \quad i = 1, \dots, N \quad (1)$$

From (1) one can derive:

$$\begin{pmatrix} x_i - x_j \\ y_i - y_j \end{pmatrix} + d_i \begin{pmatrix} \cos \beta_i \\ \sin \beta_i \end{pmatrix} - d_j \begin{pmatrix} \cos \beta_j \\ \sin \beta_j \end{pmatrix} = 0, \quad i, j = 1, \dots, N. \quad (2)$$

Multiplying (2) by $(\cos \beta_i \ \sin \beta_i)$ we obtain

$$d_i = d_j \cos(\beta_i - \beta_j) - \frac{1}{p_{ij}} \cos(\beta_j - \psi_{ij}), \quad (3)$$

where, $p_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$ and $\psi_{ij} = \arctan(\frac{y_i - y_j}{x_i - x_j})$, both of which can be measured locally without knowing the global coordinates.

Clearly when $N > 2$, there is redundant information in (3) and this can be used, for example, to improve the accuracy and robustness of the estimation. In the rest of the paper, we only consider the case $N = 2$, where the distance estimates are given by

$$\begin{pmatrix} d_1 \\ d_2 \end{pmatrix} = \begin{pmatrix} -\cos \beta_1 & \cos \beta_2 \\ -\sin \beta_1 & \sin \beta_2 \end{pmatrix}^{-1} \begin{pmatrix} x_1 - x_2 \\ y_1 - y_2 \end{pmatrix}$$

provided that the inverse exists. The estimation problem is thus how to obtain a good estimate (\hat{d}_1, \hat{d}_2) , 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.

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.

In this paper we assume that each robot is modeled as unicycle

$$\begin{aligned} \dot{x}_i &= v_i \cos \theta_i \\ \dot{y}_i &= v_i \sin \theta_i \quad i = 1, 2 \\ \dot{\theta}_i &= \omega_i \end{aligned} \quad (4)$$

where v_i and ω_i are the controls, and the directional sensor is mounted along the orientation axis.

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 as 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,

$$x_m = \frac{x_1 + x_2}{2}, \quad y_m = \frac{y_1 + y_2}{2}.$$

Under the ideal condition that the formation is maintained, the position of this point together with the heading of the axis θ_m decides the state of the formation for the two robots, where

$\theta_m = \psi_{12} - \frac{\pi}{2}$. The evolution of the ideal formation for the two robots is described by

$$\begin{aligned}\dot{x}_m &= v_m \cos \theta_m \\ \dot{y}_m &= v_m \sin \theta_m \\ \dot{\theta}_m &= \omega_m,\end{aligned}\quad (5)$$

where v_m is the speed of (x_m, y_m) , and ω_m the rotational speed of the axis. Accordingly we have to have

$$\begin{aligned}v_1 &= v_m - \omega_m p/2 \\ v_2 &= v_m + \omega_m p/2 \\ \omega_1 &= \omega_m \\ \omega_2 &= \omega_m,\end{aligned}\quad (6)$$

in order to keep $\dot{p} = 0$ and the heading perpendicular to the axis.

Now let

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

with

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

It is easy to show that with the above defined v_m and ω_m $\dot{p} = \dot{d}_1 = \dot{d}_2 = 0$ and $\dot{\alpha}_1 = \dot{\alpha}_2 = 0$, i.e., the desired formation is maintained.

Equation (5) will be used to plan the reference path, where no dynamics of the actual robots is considered. The evolution of the reference formation is updated at discrete time instances $t_k, k = 0, 1, \dots$. Given an estimate of the target position at time t_k , the target position at t_{k+1} is estimated. We suppose that no model of the target is available, so a simple estimate is linear extrapolation

$$\begin{aligned}\tilde{x}_T(t_{k+1}) &= \hat{x}_T(t_k) + (t_{k+1} - t_k) \hat{v}_T(t_k) \cos \hat{\theta}_T(t_k) \\ \tilde{y}_T(t_{k+1}) &= \hat{y}_T(t_k) + (t_{k+1} - t_k) \hat{v}_T(t_k) \sin \hat{\theta}_T(t_k)\end{aligned}$$

where

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

The reference path provided to the low-level motion control is given by trajectories generated from the controls $v_i, \omega_i, i = 1, 2$ in (6), 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 $v_i, \omega_i, i = 1, 2$, are constant over an interval, they generate the following reference trajectories:

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

Now the question is how to choose the initial reference points at each step $(x_i^f(t_k), y_i^f(t_k), \theta_i^f(t_k)), i = 1, 2$, such that the desired formation is fulfilled. 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

$$\begin{aligned}\theta_i^f(t_k) &= \arctan \frac{\hat{y}_T(t_k) - y_m(t_k)}{\hat{x}_T(t_k) - x_m(t_k)} \\ x_i^f(t_k) &= x_m^f(t_k) \pm p^f \cos(\theta_i^f(t_k)) \\ y_i^f(t_k) &= y_m^f(t_k) \mp p^f \sin(\theta_i^f(t_k)),\end{aligned}$$

where $x_m(t_k) = \frac{1}{2}(x_1(t_k) + x_2(t_k))$, $y_m(t_k) = \frac{1}{2}(y_1(t_k) + y_2(t_k))$, and p^f specifies the desired distance between the robots.

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

$$\begin{aligned}\theta_i^f(t_k) &= \hat{\theta}_T(t_k) \\ x_i^f(t_k) &= x_m^f(t_k) \pm p^f \cos(\theta_i^f(t_k)) \\ y_i^f(t_k) &= y_m^f(t_k) \mp p^f \sin(\theta_i^f(t_k)),\end{aligned}$$

where

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

and d_m^f specifies the desired distance to the target formation.

Remark: Note that the reference trajectory (x_i^f, y_i^f) can easily be expressed in the robot-fixed coordinate systems. During each updating interval (t_k, t_{k+1}) , the odometry can be used for localization with respect to the initial frame at t_k . In the following we use only 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], [7]. The idea is briefly summarized as follows.

When one uses directional sensors, it is important that the relative orientation of the robot is known. This naturally could be achieved if we could specify at what relative orientation the look-ahead distance should be kept. Here we want the mobile robot at all instants to be oriented towards the target and therefore we simply choose the reference point (x_L^i, y_L^i) to be on the robot's axis of orientation at a distance L from the center of the robot. Then

$$\begin{aligned}x_L^i &= x_i + L \cos \theta \\ y_L^i &= y_i + L \sin \theta.\end{aligned}\quad (7)$$

When choosing L in implementation, one has to make a compromise between performance and computation.

Proposition Let v_r and θ_r be the speed and orientation of the point (x_r, y_r) to be tracked by the robot. Then for each robot

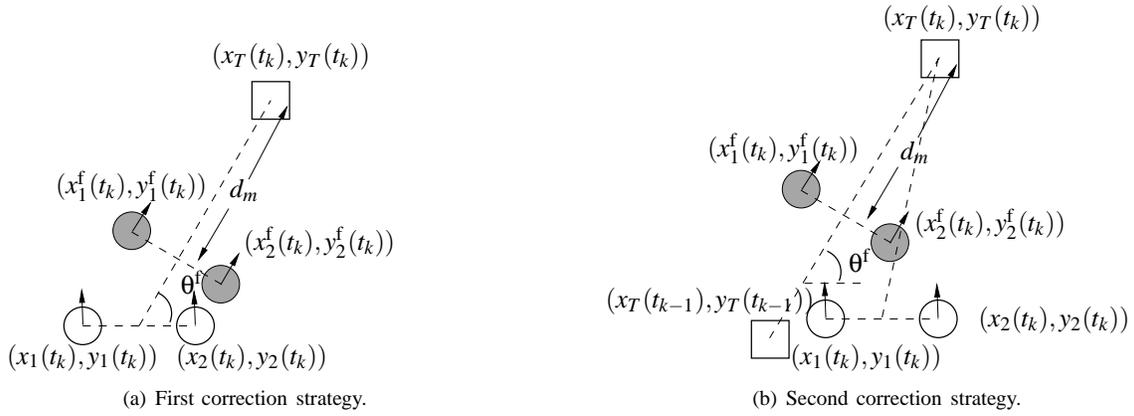


Fig. 2. Correction strategies for the initial points at each iteration in the Formation Control level.

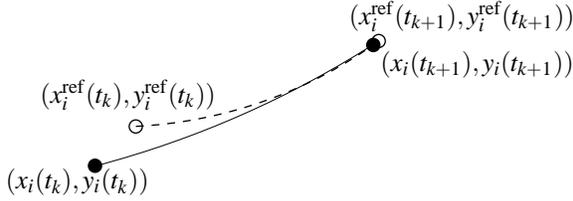


Fig. 3. Tracking of a trajectory generated between t_k and t_{k+1} .

in (4), as $t \rightarrow \infty$, $(x_L^i(t), y_L^i(t))$ converges to $(x_r(t), y_r(t))$ with the following control

$$\begin{aligned} v_i &= -k(L - \rho_i \cos \Delta\phi) + v_r \cos(\theta_r - \theta_i) \\ \omega_i &= \frac{k\rho_i}{L} \sin \Delta\phi_i + \frac{v_r}{L} \sin(\theta_r - \theta_i) \end{aligned}$$

where $\Delta\phi_i = \arctan(\frac{y_r - y_i}{x_r - x_i}) - \theta_i$ is the relative angle to the target measured by the robot, ρ_i is the distance of the robot to the reference point and k is any positive constant.

In our case (x_r, y_r) for each robot is specified by 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 the trajectories as $p_i(s_i) := x_i^{\text{ref}}(s_i)$, $q_i(s_i) := y_i^{\text{ref}}(s_i)$, $i = 1, 2$. The parameter, s_i , is defined in order to adjust the speed of the point according to the tracking error so that the robot can keep up with it (see Figure 3). Here we choose to define s_i as

$$s_i = \frac{v_i^0}{\sqrt{p_i^2 + q_i^2}} e^{-a\rho_i}$$

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 is an appropriate positive constant. We then get

$$\begin{cases} v_i &= k(\rho_i \cos \Delta\phi_i - L) + v_i^0 e^{-a\rho_i} \cos(\theta_T - \theta_i) \\ \omega_i &= \frac{k\rho_i}{L} \sin \Delta\phi_i + \frac{v_i^0}{L} e^{-a\rho_i} \sin(\theta_T - \theta_i). \end{cases} \quad (8)$$

In implementation, we can let $\theta_T = \hat{\theta}_T(t_k)$.

IV. SIMULATION RESULTS

The hierarchical control algorithms developed in previous section are now evaluated through a simulated example. In

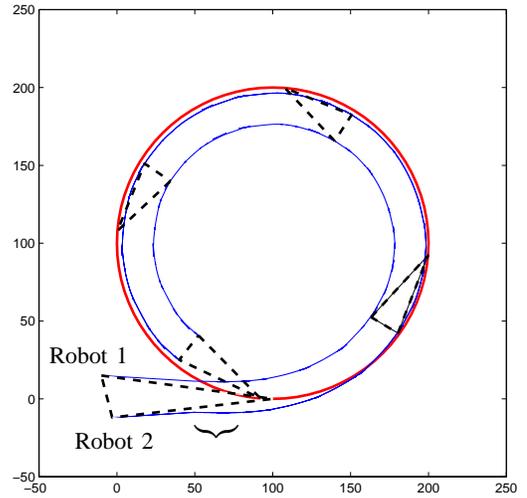


Fig. 4. The target is shown as a thick line and with thin lines the trackers trajectories. Despite the errors in the initial positions of the robots, they converge to the desired formation after a few events (note the difference between the initial state formation triangle and the posterior four ones).

terms of formation planning, only the first strategy is tested. Let the target follow a circular trajectory and the trackers have an error in the measurement of their β_i angles with distribution $U(-0.02, 0.02)$. Figure 4 shows the target as a thick line and with thin lines the trackers trajectories. Note that despite errors in the initial positions of the robots, they converge to the desired formation after a few events, see Figure 6. The formations at the initial state and four other instants are indicated with dashed triangles. Figure 5 shows a zoomed view of a part of Robot 2's trajectory, taken from Figure 4, with added information. The zoom is taken in the beginning of the trajectory and thus the robot is still trying to converge to the high-level planner reference, as also is shown in Figure 6. In Figure 5 with circles are marked the corrected initial points at each step of the high-level planning algorithm, it is $(x_i^f(t_k), y_i^f(t_k))$. The stars denote the final position of the tracker robots after an step, namely $(x_i(t_k), y_i(t_k))$. And the arrows are pointing at the positions designed as final points of

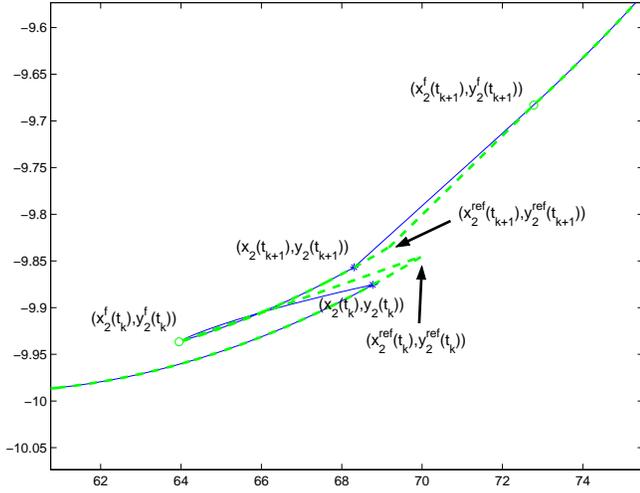


Fig. 5. A closer look to the Robot 2's trajectory in the region marked with a big under-brace in Figure 4. Some extra information has been added about the planned path (thick dashed line). The circles denote $(x_2^f(t_k), y_2^f(t_k))$, asterisks $(x_2^{\text{ref}}(t_k), y_2^{\text{ref}}(t_k))$ and the arrows point to $(x_2^{\text{ref}}(t_k), y_2^{\text{ref}}(t_k))$. Note that the x-axis and y-axis scale are different.

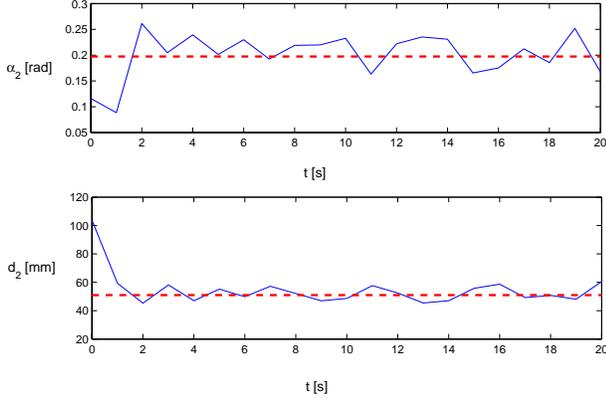


Fig. 6. Evolution of d_2 and α_2 (Robot 2) corresponding to the first quarter of circle in Figure 4. The dashed thick line shows the desired values of (d_2, α_2) to fulfill the desired formation.

the trajectory at the previous step, i.e. $(x_i^{\text{ref}}(t_k), y_i^{\text{ref}}(t_k))$. The estimated position of the target compared to the real position of the target in this simulation is shown in Figure 7.

V. EXPERIMENTAL RESULTS

Currently we are implementing the algorithms on a team of two Khepera II robots [8] tracking a target robot. The setup is illustrated in Figure 8. Khepera II is a small self-contained wheeled robot with micro-processor and basic sensors (infrared 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 constraint of the linear video sensor is a limited horizontal field of view of

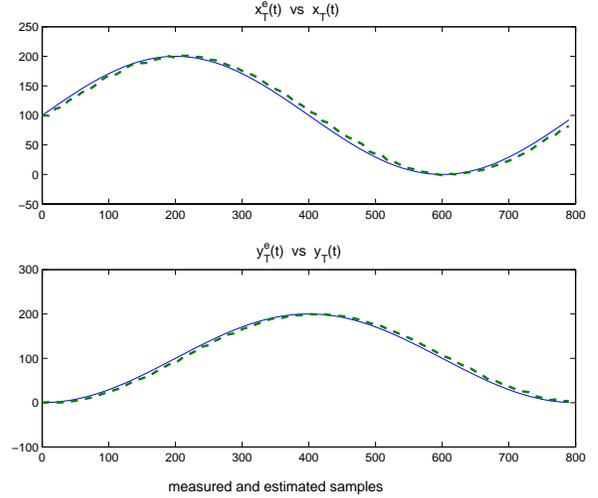


Fig. 7. Comparison of the position of the target estimated by the robots based on their sensor measurements (x_T^e, y_T^e) (thick dashed line) vs the real position of the target (x_T, y_T) (thin line).

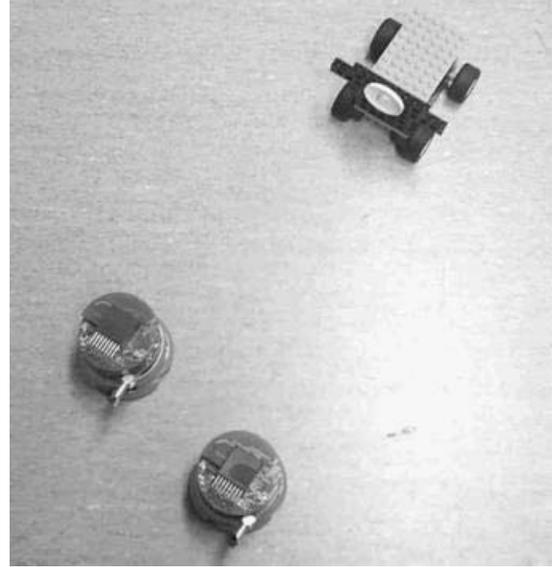


Fig. 8. Two Khepera II robots tracking a target.

36 degrees, i.e., the angular range is limited to $\alpha_{\text{max}} = 18^\circ$. For data exchange, the two robots use an on-board radio communication system.

In this testing benchmark some experiments have been performed as the one presented in Figure 10 with a LEGO robot acting as target and moving in a slow and smooth curved trajectory. As can be seen in that figure the starting formation is not the desired one, and after one step of the high-level planner the desired formation is reached. Moreover, in Figure 11 is shown in a closer look how the formation is kept by driving the robots following the trajectories generated by the path-planning level. Finally, Figure 12 presents a zoom of one of the Khepera II robots following the desired planned trajectories using the controller presented in Section III.B.

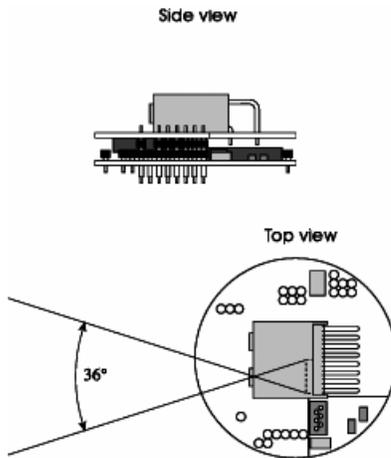


Fig. 9. Khepera II linear vision sensor. The angular range is limited to $\alpha_{\max} = 18^\circ$. (Illustration from [8].)

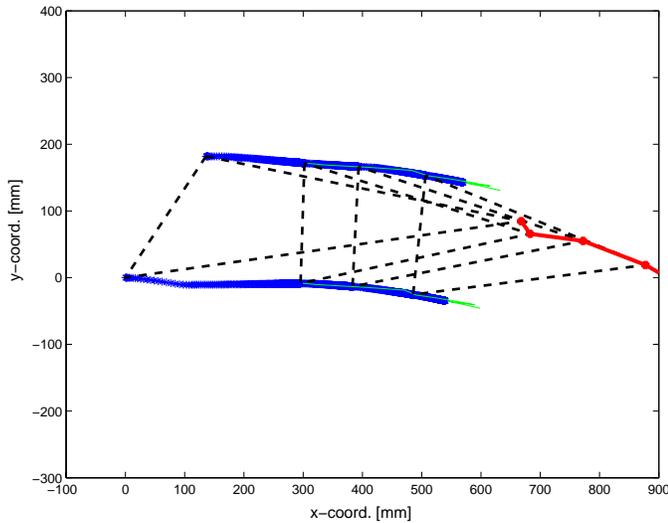


Fig. 10. An experiment showing two Khepera II robots tracking a LEGO Mindstorms robot moving on a smooth curving trajectory. The thick line represents the estimated trajectory of the LEGO robot, and the thicker lines (sequence of asterisks) the trackers trajectories. With thick dashed triangles is marked the formation evolution.

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.

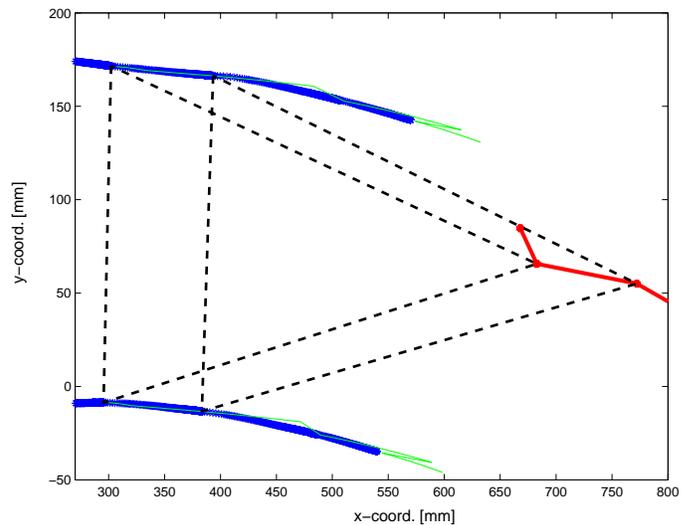


Fig. 11. Zoom from Figure 10 showing how the formation evolves and is kept between two sampling points.

ACKNOWLEDGMENT

This work was supported by the European Commission through the RECSYS IST project, the Swedish Research Council and the Swedish Foundation for Strategic Research through its Centre for Autonomous Systems at the Royal Institute of Technology.

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] T. Gustavi and X. hu, "Formation control for mobile robots with limited sensor information," in *IEEE Conference on Robotics and automation*, 2005.
- [8] K-Team S.A. [Online]. Available: <http://www.k-team.com>

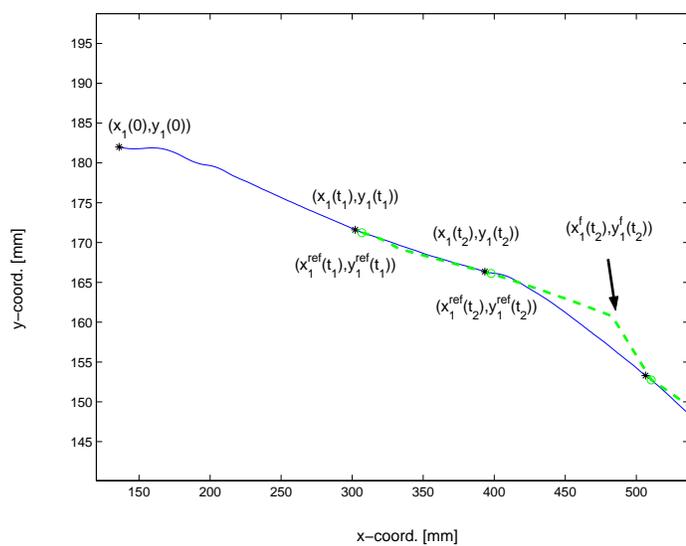


Fig. 12. Big zoom from Figure 10, in a similar manner to Figure 5, showing how one Khepera II robot follows the trajectories designed at the path planning level. The figure presents with a thick dash line the path planner trajectory and with a thin line the robot motion. Circles mark the desired ending points from the path planning, and with asterisks the position of the robot at the sampling instants. Also pointed by an arrow a corrected point is remarked.