The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

Distributed Minimax Filter for Tracking and Flocking

Dongbing Gu and Huosheng Hu

Abstract— **In this paper, we investigate a moving target tracking problem with mobile sensor networks. The moving target is assumed to be an intelligent agent, which is 'smart' enough to escape from the detection. We formulate this target estimation problem as a zero-sum game in this paper and use a so-called minimax filter to estimate the target position. The minimax filter is a robust filter that minimizes the estimation error by considering the worst case noise. Furthermore we develop a distributed version of the minimax filter for multiple sensor nodes. The distributed computation is implemented via a consensus filter. Finally, the mobile sensor nodes need to control their motions to move towards the estimated target position and avoid collisions with neighbors. A flocking algorithm is developed for this purpose. The simulation results show that the target tracking algorithm proposed in this paper provides a satisfactory result.**

I. INTRODUCTION

The problem tracking a moving target in a mobile sensor network can be solved in two steps: estimating the target position by using readings from sensors, and moving towards the target by controlling mobile node's motion. Each sensor node has a local sensor to detect the moving target. An estimation filter needs to be developed in each node to estimate the target position. A flocking algorithm needs to be developed in each node to control itself to move towards the estimated target position.

One challenging problem in this tracking task is that the moving target is very likely an intelligent agent, which is 'smart' enough to increase the tracking difficulty. Basically it can maximize the estimation error and therefore has the potential to lead to the failure of the tracking task. Another challenging problem in this tracking task is that mobile sensor nodes only have limited wireless communication capability and there is a need to find a way to cooperate efficiently in target estimation.

In this paper, we solve the first challenging problem by applying the differential game theory approach [1] to the estimation of target position. A zero-sum game is used to model the estimation problem. By solving the saddle-point equilibrium of the game, an iterative minimax filter is developed. The minimax filter is robust to the adversary moving target since it is obtained by minimizing the estimation error under the worst case noise. The minimax filter is then formatted into the information form [2], which helps developing the distributed implementation.

We solve the second challenging problem by developing a distributed version of the minimax filter. The cooperation in the estimation problem is established by exchanging sensor readings and sensor related information between nodes. However, a sensor node is difficult to communicate with all other nodes in the sensor network because it has limited wireless communication range. The distributed version of the minimax filter only requires a sensor node to communicate with its neighbors that are located in the range of wireless communication. To obtain sensor readings and sensor related information from all other sensor nodes over the entire network, a consensus filter is employed in the minimax filter. The function of the consensus filter is to diffuse the local information over the entire network via local communication.

Once the estimated target position is obtained, a flocking algorithm can be developed in a distributed way: each mobile node controls its motors to move towards the estimated target position. Since all of them try to move to the same target position, it is very likely to collide with each other. Thus, a mobile node also needs a collision avoiding mechanism to avoid collisions with its neighbors. When mobile sensor nodes can move to the estimated target position and avoid collisions with neighbors simultaneously, a balance will be kept where their distances between each other will keep constant and all of them move towards the target as a flock. Consequently the local communication is maintained.

Using a sensor network to estimate the position of a moving target has been investigated recently by some researchers. The distributed Kalman filter is one of the popular approaches for such a purpose. The work in [3] [4] [5] [6] [7] [8] employed a consensus filter to implement the distributed version of Kalman filter to estimate the position of a moving target in a static sensor network. In [9], the problem is solved by communicating estimates between neighbors and then forming a weighted average as the new estimate. The weights are optimized off-line to yield a small estimation error covariance. In [10], they adopted a strategy to reduce information transmitted for sensor networks. They demonstrated that only a single bit per sensor reading will be good enough for the Kalman filter. In [11], they developed an adaptive dynamic strategy for sensor selection and fusion location using a certainty equivalence approach that seeks to optimize a tradeoff between tracking error and communications cost. In [12], a cluster of sensor nodes is self-organized and a cluster head is elected to implement the Kalman filter. The cluster will vary with the moving target.

For the problems where the noise does not possess the Gaussian distribution, the distributed particle filters have been developed via message passing method [13] or consensus filter [14]. In [15], an information utility of data is optimized to route the communication and tracking path in

Dongbing Gu and Huosheng Hu are with School of Computer Science and Electronic Engineering, University of Essex, Wivenhoe Park, Colchester, UK dgu@essex.ac.uk

978-1-4244-6676-4/10/$25.00 ©2010 IEEE

a static sensor network.

In the rest of this paper, the tracking problem with an adversary moving target and multiple sensors is formulated in section II. The distributed minimax filter is developed in section III. The simulation is given in section IV. The conclusion and future work are summarized in section V.

II. Game Model and Minimax Filter

A. Target and Measurement Models

The minimax filter is used to estimate states of a dynamic system based on the measurements related to the estimated states, the measurement model, and the system state model. It looks the same as other state estimators, such as the Kalman filter. However the difference is that the system state model includes a fictitious adversary disturbance which is used to model an adversary target, i.e. the noise in the system state model is not only limited to the Gaussian noise, but also includes some partially unknown noise. The adversary target is assumed 'smart' enough to deliberately maximize the estimation state error. The minimax filter is a minimized estimator in the case of the worst adversary disturbance.

The centralized minimax filter consists of multiple sensors and a centralized node to where all sensors send their readings in order to execute a minimax filter to estimate the target position. The target is modelled as a discrete time linear time-invariant system defined by the system equation:

$$x_{t+1} = Ax_t + Bw_t + d_t \tag{1}$$

where x_t is the system state and w_t is the system noise with zero mean and covariance matrix $Q \geq 0$. A and B are the matrices of the appropriate dimensions with bounded entries. d_t is the adversary disturbance. Assume that the target is intelligent and can maximize the estimation error. Let $\hat{x}$ denote the estimated state, the estimation error is $x_t - \hat{x}_t$. The adversary disturbance is modelled as:

$$d_t = L(C(x_t - \hat{x}_t) + n_t) \tag{2}$$

where L is a gain to be determined, n_t is the Gaussian noise with zero mean and diagonal covariance matrix $S > 0$. C is a matrix that is the same as the one used in the measurement equation.

Assume that N sensors are used, each of which has a measurement y_t^i at time t. For sensor i, the measurement equation is defined as

$$y_t^i = C^i x_t + v_t^i \tag{3}$$

where v_t^i is the Gaussian noise with zero mean and covariance matrix $R^i > 0$.

Multiple measurements are stacked into a measurement vector and the compact measurement equation is

$$y_t = Cx_t + v_t \tag{4}$$

where

$$y_t = \begin{bmatrix} y_t^1 \\ \vdots \\ y_t^N \end{bmatrix}, C = \begin{bmatrix} C^1 \\ \vdots \\ C^N \end{bmatrix}, v_t = \begin{bmatrix} v_t^1 \\ \vdots \\ v_t^N \end{bmatrix}$$

R is the covariance matrix of v_t:

$$R = \begin{bmatrix} R^1 & & \\ & \ddots & \\ & & R^N \end{bmatrix}$$

The noises w_t, v_t, and n_t are mutually uncorrelated and also uncorrelated with x_0.

B. Zero-Sum Game

In general, unbiased estimators are preferred over biased estimators due to the mathematical tractability. The minimax filter is also confined to the unbiased estimator with the following form:

$$\hat{x}_{t+1} = A\hat{x}_t + K(y_t - C\hat{x}_t) \tag{5}$$

where K is the gain of the filter. The estimation error is defined by

$$e_t = x_t - \hat{x}_t \tag{6}$$

The estimation error is evolved by the following form:

$$\begin{aligned} e_{t+1} &= (A - KC + LC)e_t + Bw_t + Ln_t - Kv_t \\ &= Fe_t + Bw_t + Ln_t - Kv_t \end{aligned}$$

where $F = A - KC + LC$. In order to form a minimax problem, the estimation error is decomposed into two parts:

$$e_t = e_t^K + e_t^L$$

where

$$\begin{aligned} e_{t+1}^K &= Fe_t^K + Bw_t - Kv_t, \qquad & e_0^K = x_0 \\ e_{t+1}^L &= Fe_t^L + Ln_t, \qquad & e_0^L = 0 \end{aligned}$$

The cost function in the minimax problem is then defined by

$$J(K, L) = tr \sum_{t=0}^{H} g_t E[||e_t^K||^2 - ||e_t^L||^2] \tag{7}$$

where H is the time horizon, g_t is the weight parameter, and K, L are the filter gains to be optimized. By denoting $P_{t+1} = FP_tF' + BQB' + KRK' - LSL'$ and $F = A - KC + LD$, the cost function J is given by the following [16] [17]:

$$J(K, L) = tr \sum_{t=0}^{H} g_t P_t \tag{8}$$

C. Minimax Filter

With the given cost function and the error state equation, we can find the worst case adversary performance by maximizing the cost function with respect to L and find the best performance given the worst case adversary performance by minimizing the cost function with respect to K. By denoting the optimized gains as K^* and L^* respectively, the equilibrium of the zero-sum game is satisfied with

$$J(K^*, L) \leq J(K^*, L^*) \leq J(K, L^*) \tag{9}$$

By solving this zero-sum game, the game equilibrium is found as follows:

$$K^* = A\Sigma_t C' R^{-1} \tag{10}$$

$$L^* = A\Sigma_t C' S^{-1} \tag{11}$$

and

$$\Sigma_t^{-1} = P_t^{-1} + C'(R^{-1} - S^{-1})C \tag{12}$$

The minimax filter algorithm can be summarized as:

$$\begin{array}{l} \hat{x}_{t+1} = A\hat{x}_t + K^*(y_t - C\hat{x}_t) \\ \Sigma_t^{-1} = P_t^{-1} + C'(R^{-1} - S^{-1})C \\ P_{t+1} = A\Sigma_t A' + BQB' \\ K^* = A\Sigma_t C' R^{-1} \end{array} \tag{13}$$

with $\hat{x}_0 = x_0$ and P_0.

III. Distributed Minimax Filter and Flocking

It is not scalable and robust to receive readings from all other sensor nodes in order to use the minimax filter to estimate the result. The distributed filter advocates each sensor node only communicates with its neighbors to exchange sensor readings and executes the estimation using its own readings and neighbor's readings.

A. Consensus Filter

We will use the information filter form [2] to deduce the distributed algorithm. The centralized minimax filter in the information form can be written as:

$$\Sigma_t^{-1} = P_t^{-1} + \sum_{j=1}^{N} C^{j\prime}(R^{j^{-1}} - S^{-1})C^j$$

$$P_{t+1} = A\Sigma_t A' + BQB'$$

$$\hat{x}_{t+1} = A\hat{x}_t + A\Sigma_t\Big(\sum_{j=1}^{N} C^{j\prime} R^{j^{-1}} y_t^j - \sum_{j=1}^{N} C^{j\prime} R^{j^{-1}} C^j \hat{x}_t\Big)$$

Let

$$\begin{aligned} \chi_t^{j,1} &= C^{j\prime} R^{j^{-1}} C^j \\ \chi_t^{j,2} &= C^{j\prime} R^{j^{-1}} y_t^j \\ \chi_t^{j,3} &= C^{j\prime}(R^{j^{-1}} - S^{-1})C^j \end{aligned}$$

And $\chi_t^j = \left[\chi_t^{j,1}, \chi_t^{j,2}, \chi_t^{j,3}\right]$. It can be seen from the centralized minimax filter that the estimation requires the sum calculation of χ_t^j from all network members. To develop a distributed filter, the sum calculation of χ_t^j has to be calculated in a distributed way. A consensus filter can be used to implement the distributed sum calculation of χ_t^j.

Before we present the consensus filter, the wireless communication network should be modelled as a graph. A graph represents interconnections between sensor nodes. A vertex of the graph corresponds to a node and edges of the graph correspond to interconnections between nodes. Formally, a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consists of a set of vertices $\mathcal{V} = \{\nu^1, ..., \nu^N\}$, indexed by nodes in the network, and a set of edges $\mathcal{E} = \{(\nu^i, \nu^j) \in \mathcal{V} \times \mathcal{V}\}$, containing unordered pairs of distinct vertices. Assume that the graph has no loops, i.e. $(\nu^i, \nu^j) \in \mathcal{E}$ implies $\nu^i \neq \nu^j$.

Let E denote the distance that a node can communicate via wireless radio links. Edge (ν^i, ν^j) is connected if the Euclidean distance between nodes i and j is less than or equal to E. Node j is a neighbor of node i. Node i makes sensor measurement y_t^i, send it to its neighbors, and also receives the measurement y_t^j from its neighbor j.

A graph is connected if for any vertices $(\nu^i, \nu^j) \in \mathcal{V}$, there exists a path of edges in $\mathcal{E}$ from ν^i to ν^j. The set of neighbors of vertex i is defined as $\mathcal{N}_i = \{j \in \mathcal{V} : (i,j) \in \mathcal{E}\}$. The degree of vertex i is defined as $\delta_i = N_i$ and the maximum degree is $\delta_{max} = \max_i \delta_i$. Let Δ be the degree matrix, $\Delta = diag(\delta^i)$. The adjacency matrix $\mathcal{A}$ is the integer matrix with rows and columns indexed by the vertices, such as the ij-entry of $\mathcal{A}$ is equal to the number of edges from i to j. Following [18], the Laplacian matrix of a graph $\mathcal{G}$ is defined as L:

$$\mathcal{L} = \Delta - \mathcal{A} \tag{15}$$

Let the initial value χ_t^i be given

$$\chi_0^i = \left[\begin{array}{l} C^{i\prime} R^{i^{-1}} C^i \\ C^{i\prime} R^{i^{-1}} y_t^i \\ C^{i\prime}(R^{i^{-1}} - S^{-1})C^i \end{array}\right]$$

The consensus filter is in the following form:

$$\chi_{t+1}^i = \chi_t^i + \eta \sum_{j \in \mathcal{N}_i} (\chi_t^j - \chi_t^i) \tag{16}$$

where η is the updating rate and should be constrained by:

$$\eta \leq \frac{1}{\delta_{max}}$$

When the sensor nodes are in motion, the graph is changed with time. Given the dynamic graph is connected throughout the whole tracking process, the above constraint guarantees the stability of the consensus filter according to the Gershgorin theorem. The connectivity of the graph will be secured by using the flocking algorithm with a potential function introduced in section III-C. The filter output asymptotically converges to:

$$\chi_t^i \rightarrow \frac{1}{N}\sum_{j=1}^{N} \chi_t^j \tag{17}$$

Finally $N\hat{\chi}_t^i$ can be used as an approximation to the sum calculation of χ_t^j.

B. Distributed Filter

In the distributed way, the neighbor nodes exchange their information via wireless communication. The information sent out and received in node i are χ_t^i and χ_t^j where $j \in \mathcal{N}_i$, respectively. Each node i has an estimation result and the estimation result is denoted as $\hat{x}_t^i$ and P_t^i. Figure 1 shows the structure of the algorithm. The distributed minimax filter for node i is as follows:

$$\chi_0^i = \begin{bmatrix} \chi_t^{i,1} \\ \chi_t^{i,2} \\ \chi_t^{i,3} \end{bmatrix} = \begin{bmatrix} C^{i\prime} R^{i^{-1}} C^i \\ C^{i\prime} R^{i^{-1}} y_t^i \\ C^{i\prime} (R^{i^{-1}} - S^{-1}) C^i \end{bmatrix} \tag{18a}$$

$$\chi_{t+1}^i = \chi_t^i + \eta \sum_{j \in N_i} (\chi_t^j - \chi_t^i) \tag{18b}$$

$$\Sigma_t^{i^{-1}} = P_t^{i^{-1}} + N\chi_t^{i,3}$$

$$\hat{x}_{t+1}^i = A\hat{x}_t^i + AP_t^i \left(N\chi_t^{i,2} - N\chi_t^{i,1}\hat{x}_t^i \right)$$

$$\bar{P}_{t+1}^i = A\Sigma_t^i A' + BQB'$$

In the above, the equations (18a) and (18b) are the consensus filter. They constitute an inner loop used to calculate the approximated sum calculation of χ_t^j.

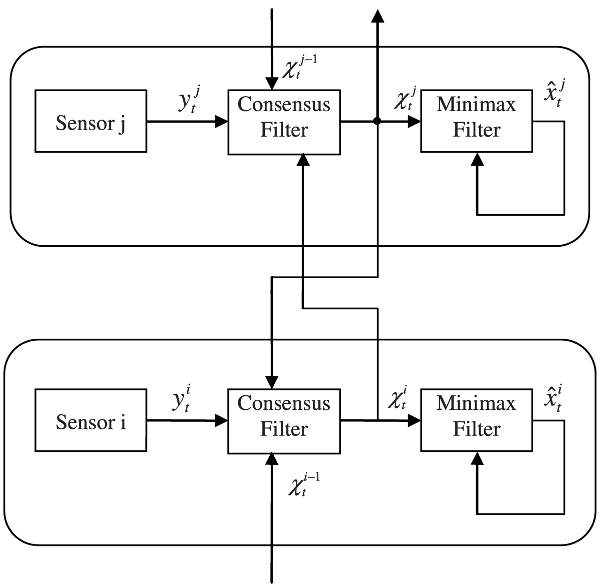

Fig. 1. Distributed algorithm architecture

C. Flocking Controller

There are N mobile sensor nodes in a network, each of which is described by its double integrator dynamics. For a node i with $2D$ dimensional coordinates q^i, the state and control vectors are $z_t^i = [q_t^i, \dot{q}_t^i]^T$ and u_t^i. The dynamics is:

$$z_{t+1}^i = Az_t^i + Bu_t^i \tag{19}$$

where

$$A = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, B = \begin{bmatrix} T^2/2 & 0 \\ 0 & T^2/2 \\ T & 0 \\ 0 & T \end{bmatrix},$$

where T is the sample interval. When the estimated target position is available, each mobile nodes should move towards to the target and avoid collisions with neighbors. This behavior requires that the flocking controller to be designed consist of two basic components. The first one is the tracking control. The estimated target position plays a key role in this component. When all mobile nodes move towards the target, they will have potential to converge to the target and achieve the cohesive property. Let q_i denote the current position of node i. The cohesive potential function is defined as $H_c = 1/2||q_t^i - \hat{x}_t^i||^2$, where $||q_t^i - \hat{x}_t^i||$ is the Euclidian norm of $q_t^i - \hat{x}_t^i$.

The second component is the separation control. Adjacent mobile nodes are required to keep a specific distance. If the distance between adjacent nodes is too small, they attempt to separate. Let H_s denote the separation potential function between nodes i and j:

$$H_s = h_1 e^{-||q_t^i - q_t^j||/20} - h_2 e^{-||q_t^i - q_t^j||/60}$$

where h_1 and h_2 are two parameters.

The flocking controller should be designed as:

$$u_t^i = -k_c \frac{\partial H_c}{\partial q_i} - k_r \sum_{j \in N_i} \frac{\partial H_s}{\partial q_i} \tag{20}$$

where k_c and k_r are the positive gains.

IV. Simulations

In our simulations, three moving targets are used to test the estimation performance. The first one is a 'normal target' and its model is exactly the same as the equation (1) with the noise covariance

$$Q = \begin{bmatrix} 0.01 & 0 \\ 0 & 0.01 \end{bmatrix}$$

The second one is an 'intelligent target' that is to maximize the estimation error and its model is:

$$x_{t+1} = Ax_t + k_e(x_t - \hat{x}_t)$$

where $k_e = 0.04$ is the gain.

In the simulations, the minimax filter is compared with a standard Kalman filter. The Kalman filter is obtained by taking $S \to \infty$ in both centralized and distributed versions. All other parameters of the filters are the same.

The number of mobile sensors is 10 and each of them observe the target via a sensor with

$$C^i = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, R^i = \begin{bmatrix} 0.04 & 0 \\ 0 & 0.04 \end{bmatrix}$$

and

$$S = \begin{bmatrix} 0.09 & 0 \\ 0 & 0.09 \end{bmatrix}$$

Firstly the centralized Kalman filter and minimax filter are used to estimte a 'normal target'. The estimation trajectory of the Kalman filter and the target true trajectory are shown in figure 2. The estimation trajectory of the minimax filter and the target true trajectory are shown in figure 3. It can be seen that both filters can estimate the target with reasonable accuracy.

The distributed Kalman filter and minimax filter are used to estimate the 'intelligent target' and the estimation trajectories in one of sensor nodes are shown in figure 4 for the

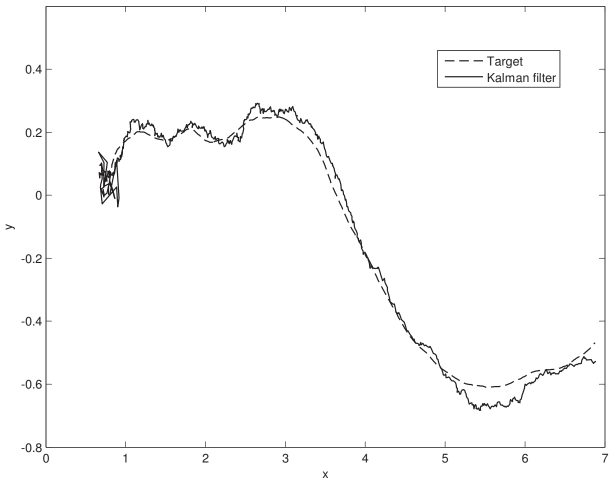

Fig. 2. ‘Normal target’ estimation trajectory by the Kalman filter

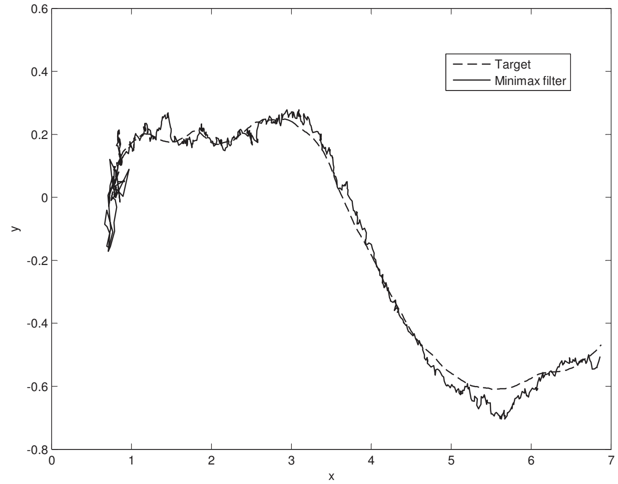

Fig. 3. ‘Normal target’ estimation trajectory by the minimax filter

Kalman filter and figure 5 for the minimax filter. Note that the target trajectories are different for two filters due to the ‘intelligent’ nature of the target. As the target deliberately maximizes the estimation error, the Kalman filter can not overcome such adversary noise and gradually the estimation error is increased. However, the minimax filter takes the adversary noise into consideration and can always predict the next-step position to generate good estimation results. The difference can be clearly seen from their estimation errors shown in figure 6 where the estimation error of the minimax filter is bounded with a small value while the estimation error of the Kalman filter goes without bound. The distributed minimax filter and the flocking controller are tested in the final simulation. The estimation error of the minimax filter is shown in figure 7. Although the error experiences an increase during the first half period when the ‘intelligent target’ turns in a sharp angle, it is kept low and stable afterwards. This shows again the minimax filter works fine with the ‘intelligent target’. All mobile sensor nodes move towards the estimated target positions and avoid collision using the flocking controller. The flocking behavior can be observed from figure 8 where all mobile nodes marked with circles are able to track the target and separate with each other.

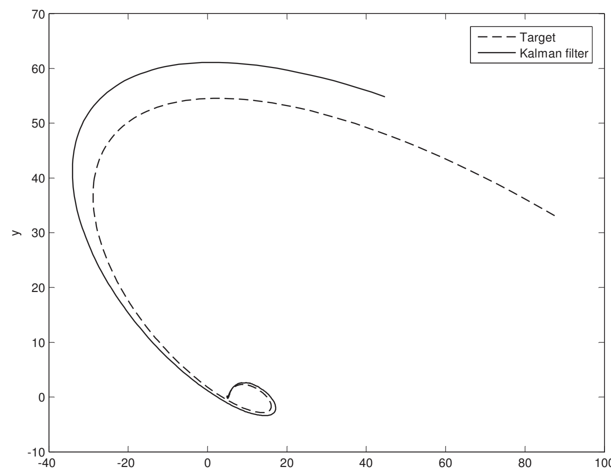

Fig. 4. ‘Intelligent target’ estimation trajectory by the Kalman filter

V. Conclusions

This paper mainly concentrates on a distributed minimax filter developed to estimate a moving target position. The moving target is assumed to be an adversary agent which can move away from the estimated position by maximizing the estimation error. A game theory approach is employed to develop the minimax filter. The filter is optimized by minimizing the estimation error and maximizing the worst case adversary noise. In order to apply the minimax filter to mobile sensor networks, a distributed version is developed by using a consensus filter. The distributed minimax filter only requires local communication between neighbor sensor nodes. The mobile sensor network also has the ability to move towards the estimated target by using a flocking

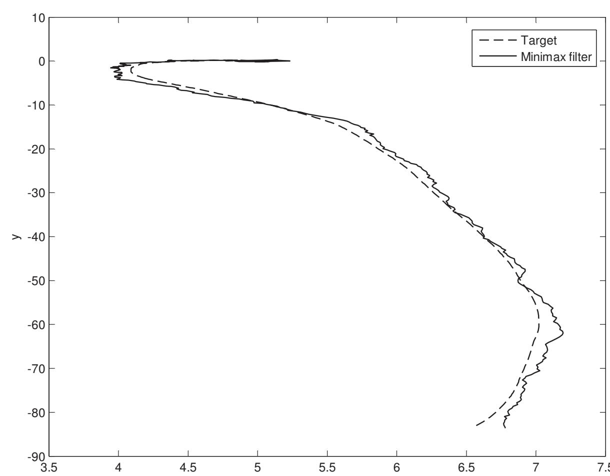

Fig. 5. ‘Intelligent target’ estimation trajectory by the minimax filter

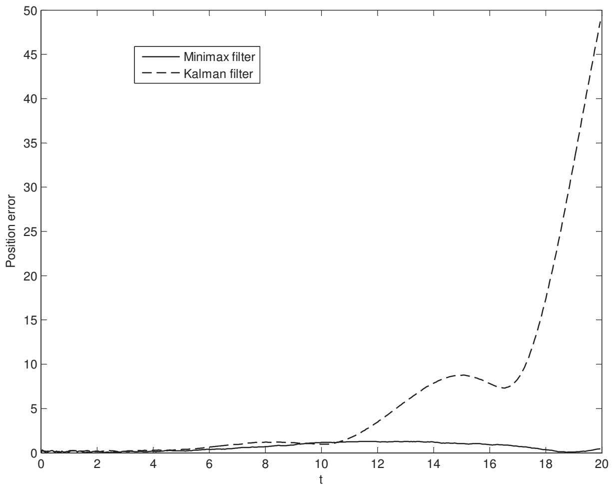

Fig. 6. ‘Intelligent target’ estimation errors

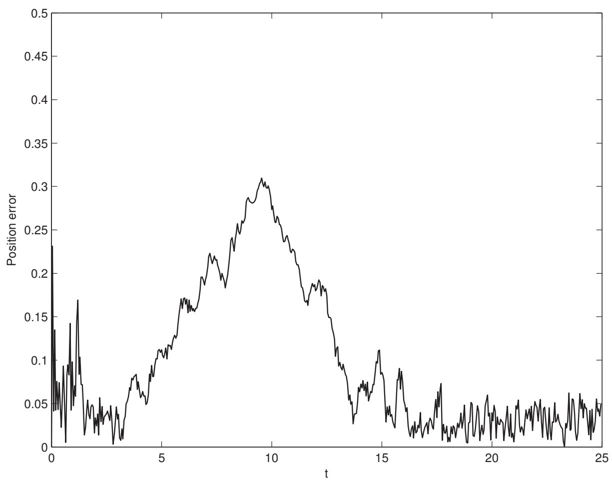

Fig. 7. ‘Intelligent target’ tracking error by the minimax filter

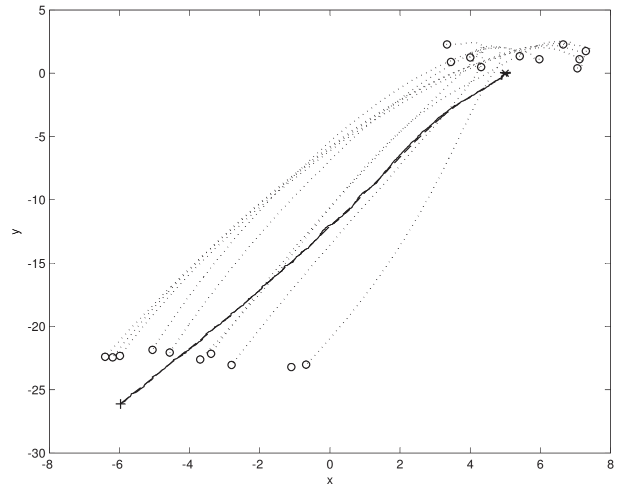

Fig. 8. ‘Intelligent target’ tracking trajectory by the minimax filter and flocking trajectories

algorithm. In own further work, we will develop a real system to test the algorithms.

ACKNOWLEDGEMENT

This research has been financially supported by European Union FP7 program, ICT-231646, SHOAL: Search and monitoring of Harmful contaminants, Other pollutants And Leaks in vessels in port using a swarm of robotic fish.

REFERENCES

[1] T. Basar and G. Olsder, *Dynamic noncooperative game theory*. SIAM, 1995.

[2] B. Rao and H. Durrant-Whyte, “Fully decentralized algorithm for multisensor Kalman filtering,” *IEE Proceedings-Control Theory and Applications*, vol. 138, pp. 413–420, 1991.

[3] R. Olfati-Saber, “Distributed Kalman filtering for sensor networks,” in *Proc. of the 44th IEEE Conference on Decision and Control*, Dec. 2007.

[4] D. P. Spanos, R. Olfati-Saber, and R. M. Murray, “Dynamic consensus on mobile networks,” in *Proc. of the 16th IFAC World Congress*, Prague, Czech, Apr. 2005.

[5] A. Speranzon, C. Fischione, and K. Johansson, “Distributed and collaborative estimation over wireless sensor networks,” in *Proc. of the 43th IEEE Conference on Decision and Control*, Dec. 2006.

[6] R. Carli, A. Chiuso, L. Schenato, and S. Zampieri, “Distributed Kalman filtering using consensus strategies,” in *Proc. of the 44th IEEE Conference on Decision and Control*, Dec. 2007.

[7] K. M. Lynch, I. B. Schwartz, P. Yang, and R. A. Freeman, “Decentralized environmental modeling by mobile sensor networks,” *IEEE Trans. on Robotics*, vol. 24, no. 3, pp. 710–724, 2008.

[8] R. Rahman, M. Alanyali, and V. Saligrama, “Distributed tracking in multihop sensor networks with communication delays,” *IEEE Trans. on Signal Processing*, vol. 55, no. 9, pp. 4656–4668, 2007.

[9] P. Alriksson and A. Rantzer, “Experimental evaluation of a distributed Kalman filter algorithm,” in *Proc. of the 44th IEEE Conference on Decision and Control*, Dec. 2007.

[10] A. Ribeiro, G. Giannakis, and S. Roumeliotis, “SOI-KF: distributed Kalman filtering with low-cost communications using the sign of innovations,” *IEEE Trans. on Signal Processing*, vol. 54, no. 12, pp. 4782–4795, 2006.

[11] S. Aeron, V. Saligrama, and D. Castanon, “Energy efficient policies for distributed target tracking in multihop sensor networks,” in *Proc. of the 43th IEEE Conference on Decision and Control*, Dec. 2006.

[12] H. Medeiros, J. Park, and A. Kak, “Distributed object tracking using a cluster-based Kalman filter in wireless camera networks,” *IEEE J. of Selected Topics in Signal Processing*, accepted.

[13] Y. Sheng, X. Hu, and P. Ramanathan, “Distributed particle filter with GMM approximation for multiple targets localization and tracking in wireless sensor networks,” in *Proc. of the 4th Int. Symposium on Information Processing in Sensor Networks*, Apr. 2005.

[14] D. Gu, “Distributed particle filter for target tracking,” in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Rome, Italy, April 2007.

[15] F. Zhao, J. Shin, and J. Reich, “Information-driven dynamic sensor collaboration for tracking applications,” *IEEE Signal Processing Magazine*, vol. 19, pp. 61–72, 2002.

[16] D. Simon, “A game theory approach to constrained minimax state estimation,” *IEEE Trans. on Signal Processing*, vol. 54, no. 2, pp. 405–412, 2006.

[17] I. Yaesh and U. Shaked, “Game theory approach to state estimation of linear discrete-time processes and its relation to H_∞ optimal estimation,” *International Journal of Control*, vol. 55, no. 6, pp. 1443–1452, 1992.

[18] C. Godsil and G. Royle, *Algebraic graph theory*. Springer-Verlag, 2001.