2007 IEEE International Conference on
Robotics and Automation
Roma, Italy, 10-14 April 2007

Distributed Cooperative Active Sensing Using Consensus Filters

Peng Yang Randy A. Freeman Kevin M. Lynch
Northwestern University
Evanston, Illinois, USA

Abstract— **We consider the problem of multiple mobile sensor agents tracking the position of one or more moving targets. In our formulation, each agent maintains a target estimate, and each agent moves so as to maximize the expected information from its sensor, relative to the current uncertainty in the estimate. The novelty of our approach is that each agent need only communicate with one-hop neighbors in a communication network, resulting in a fully distributed and scalable algorithm, yet the performance of the system approximates that of a centralized optimal solution to the same problem. We provide two fully distributed algorithms based on one-time measurements and a Kalman filter approach, and we validate the algorithms with simulations.**

I. Introduction

In this paper we use a group of mobile sensors to cooperatively track the location of a dynamic target. Each sensor can take measurements and fuse its local information with others to get a better estimate of the target position. Moreover, the additional mobility of the sensors enables them to move in such a way as to minimize the uncertainty in the fused sensor reading. The sensors may be identical, or heterogeneous with regard to their sensor types.

Recent advancement in wireless communications and electronics has enabled the development of low-cost sensor networks. Exemplary applications include target tracking [21], [13], formation and coverage control [1], [4], [5], [7], environmental monitoring [11], [17], [20], and many others. Most approaches to the target tracking problem rely on stationary sensors, and *active sensing* aims to leverage the mobility of the sensors to get better tracking performance, typically in error estimates, and this requires additional motion controller design. Given the estimation error covariance matrix P, we can use any of $\text{tr}(P)$, $\det(P)$ $\lambda_{\max}(P)$ as the performance measure. Entropy-based information measures are also available [9]. A very similar class of problems is *distributed localization*, where a group of mobile robots try to localize themselves by measuring distances to other agents [16] or landmarks [6]. Empirical tests in [8] showed the localization accuracy increases dramatically by actively controlling each robot's motion direction and sensor heading.

This work was supported in part by NSF grants IIS-0308224 and ECS-0115317, and ECS-0601661.

Peng Yang and Kevin Lynch are with the Department of Mechanical Engineering, Northwestern University, Evanston, IL 60208, USA p-yang@northwestern.edu, kmlynch@northwestern.edu.

Randy Freeman is with the Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL 60208, USA freeman@ece.northwestern.edu.

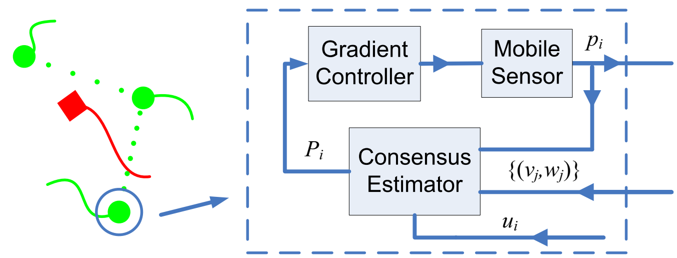

Fig. 1. Three mobile sensors collaboratively track one moving target. By taking it own sensory information u_i and obtaining the internal estimator states (v_j, w_j) from its one-hop neighbors $\{j, j \in \mathcal{N}_i\}$, each agent i implements a distributed Kalman filter to obtain a near-global estimate of the target and the associated uncertainty covariance matrix P_i. Each agent also calculates the gradient of $\det(P_i)$ with respect to its own position p_i and moves to reduce the uncertainty in the target estimate.

One approach to *active sensing* problems is to separate the estimation process from motion controller design: Given the estimation task, the optimal sensor configuration is obtained first. From there the controller design is reduced to a formation control problem. The optimal configuration is hard to obtain in general and a special case is solved in [13].

The second approach is online: By communicating, each sensor obtains a global estimate of the target and then moves to improve this estimate. For range-bearing sensors that estimate the target location from current measurement, a centralized gradient controller is derived in [3]. The range-only sensor case was discussed in [22].

In this paper we extend the results in [3] in several ways. First, by constructing a *dynamic average consensus estimator* [18], [8], we make the control algorithm in [3] distributedly implementable: Each agent communicates only with one-hop neighbors in a communication network, and the amount of data transmitted is independent of the number of agents. Another decentralized architecture for data fusion and control was described by Durrant-Whyte et al. [10], [12], and they use a general Bayesian framework for data fusion. Second, we develop a new controller design based on a better sensor fusion technique, the recently developed *distributed Kalman filter* [14]. Third, we allow heterogeneous sensors and derive motion controllers for range-only sensors.

In the following section, we detail the sensor models, the general motion controller design method, and the distributed estimator we use throughout the paper. Then we describe how to fuse measurements of individual sensors and give our first controller design in Section III. A better sensor fusion technique based on the Kalman filter and its induced controller are given in Section IV. The simulations in Section V validate our approach and illustrates the benefits of incorporating mobility in sensor networks. Finally we discuss the heterogeneous sensors and multiple targets cases in Section VI and the summary is provided in Section VII.

II. FORMULATION

A. Measurement Model

We consider n sensors and one target moving in the plane, having positions $p_1, \cdots, p_n \in \mathbb{R}^2$ and $x_t \in \mathbb{R}^2$, respectively. The observation made by the i^{th} sensor is given by

$$z_i = H_i x_t + v_i\,, \qquad i = 1, \ldots, n\,, \tag{1}$$

where the measurement noise v_i is a continuous-time Gaussian noise with zero mean. This measurement model can include several different types of sensors, and in this paper we focus on range-bearing sensors and range-only sensors as illustrative examples.

In a standard linear range-finding sensor model [15], [3], $H_i = I_2$ (the 2×2 identity matrix) and its covariance matrix R_i assumes a diagonal structure in the sensor's local range/bearing frame:

$$R_i = \begin{bmatrix} (\sigma^i_{\text{range}})^2 & 0 \\ 0 & (\sigma^i_{\text{bearing}})^2 \end{bmatrix}. \tag{2}$$

The range measurement noise variance $(\sigma^i_{\text{range}})^2$ is commonly represented by a function $f_r(r_i)$ of the distance r_i from the target to sensor i. The bearing noise variance $(\sigma^i_{\text{bearing}})^2$ also depends on the range and can be modeled as $f_b(r_i)$. We use the following simple yet representative forms of these functions:

$$(\sigma^i_{\text{range}})^2 = f_r(r_i) = a_2(r_i - a_1)^2 + a_0 \tag{3}$$

$$(\sigma^i_{\text{bearing}})^2 = f_b(r_i) = \alpha f_r(r_i)\,, \tag{4}$$

where a_0, a_1, a_2, α are model parameters. This measurement uncertainty model assumes the existence of a "sweet spot" location $r_i = a_1$ at which the noise is at its minimum value. In practice, when the target is out of the sensing range, we can initialize the diagonal entries of R_i to be ∞.

If the sensor being used takes a nonlinear measurement of the state, we will use its linearized approximate model. For example, given a range-only sensor i:

$$z_i = \|x_t - p_i\|_2 + v_i \tag{5}$$

with the Gaussian noise level $R_i = f_r(r_i)$ (as in (3)), we can linearize it around the point $x_{t0} = (x_0, y_0)$:

$$\widetilde{z}_i = -H_i x_t + v_i \tag{6}$$

with

$$\begin{aligned} H_i &= \begin{bmatrix} \frac{p_i^x - x_0}{\sqrt[2]{(x_0-p_i^x)^2+(y_0-p_i^y)^2}} & \frac{p_i^y - y_0}{\sqrt[2]{(x_0-p_i^x)^2+(y_0-p_i^y)^2}} \end{bmatrix} \\ &= \begin{bmatrix} \cos(\theta_{i0}) & \sin(\theta_{i0}) \end{bmatrix} \end{aligned} \tag{7}$$

and $\widetilde{z}_i = z_i - ||x_{t0} - p_i||_2 - H_i x_{t0}$ is our modified measurement to take into account the linearization effect. Here both H_i and $\widetilde{z}_i$ can be obtained by sensor i locally. In this paper, we don't deal with the sensor i's self-localization error and assume p_i can be measured perfectly.

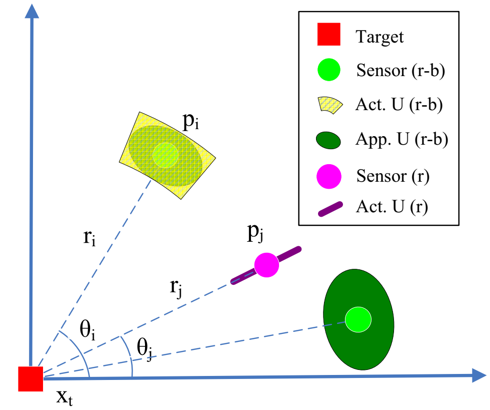

Fig. 2. Schematic of the measurement models for range-bearing sensors (r-b) and range-only sensors (r). For the range-bearing sensor, the segment of the annulus shows the one-sigma uncertainty of each sensor's estimate and the ellipse is the approximation that we use.

B. Gradient Controller Design

We consider two different ways of fusing the local target position measurements z_i and error covariances R_i to obtain a global target position estimate $\hat{x}_{\text{global}}$ and global error covariance P_{global}. The first method, described in Section III and based on the work in [3], uses only current measurements to obtain $\hat{x}_{\text{global}}$ and P_{global}. The second method, described in Section IV, defines $\hat{x}_{\text{global}}$ and P_{global} by means of a Kalman filter. In either case, the matrix P_{global} depends on the sensor and target locations, which means the sensors can move to reduce the uncertainty P_{global}. To formulate a proper cost function, we can use either

$$J = \det(P_{\text{global}}) \tag{8}$$

or

$$J = \text{tr}(P_{\text{global}}) \tag{9}$$

In optimal experiments theory, they are referred to as *D-optimal design* and *A-optimal design*, respectively. For simplicity, we assume all agents are kinematic and fully actuated so that $\dot{p}_i = u_i$, and we use the gradient controller

$$u_i = K^{\text{initial}}(\cdot) = -\Gamma T_i^T \begin{bmatrix} \dfrac{\partial J}{\partial r_i} \\ \dfrac{1}{r_i}\dfrac{\partial J}{\partial \theta_i} \end{bmatrix}, \tag{10}$$

where $\Gamma > 0$ is a gain matrix, $\theta_i = \angle(p_i - x_t)$ is the angle from the target to sensor i, and T_i is the rotation matrix

$$T_i = \begin{bmatrix} \cos(\theta_i) & \sin(\theta_i) \\ -\sin(\theta_i) & \cos(\theta_i) \end{bmatrix}. \tag{11}$$

We also use T_i to transform R_i, the covariance matrix in the local frame, to $T_i R_i T_i^T$, the covariance matrix in the global Cartesian frame. In the face of the nonlinear transformation from polar frames to Cartesian frames, this is a convenient approximation (Fig. 2). Furthermore, we define

$$P_i^r \triangleq \frac{\partial P_{\text{global}}}{\partial r_i}\,, \qquad P_i^\theta \triangleq \frac{\partial P_{\text{global}}}{\partial \theta_i}\,, \tag{12}$$

and use the following facts from matrix calculus [2]:

$$\frac{\partial}{\partial x} f(A(x)) = \text{tr}[\frac{\partial f}{dA}\frac{\partial A}{\partial x}] \tag{13}$$

$$\frac{\partial}{\partial A}\det(A) = |A|A^{-T} = |A|A^{-1} \tag{14}$$

$$\frac{\partial}{\partial A}\text{tr}(A) = I \tag{15}$$

$$\frac{\partial}{\partial x}A^{-1} = -A^{-1}(\frac{\partial A}{\partial x})A^{-1} \tag{16}$$

From above we can calculate the gradients in (10) as

$$\frac{\partial J}{\partial r_i} = J \cdot \text{tr}\big[P_{\text{global}}^{-1} P_i^r\big] \tag{17}$$

$$\frac{\partial J}{\partial \theta_i} = J \cdot \text{tr}\big[P_{\text{global}}^{-1} P_i^\theta\big] \tag{18}$$

when we use the D-optimal design (8) or the alternative form

$$\frac{\partial J}{\partial r_i} = \text{tr}\big[P_i^r\big] \tag{19}$$

$$\frac{\partial J}{\partial \theta_i} = \text{tr}\big[P_i^\theta\big] \tag{20}$$

when we use the A-optimal design (9).

In general the controller in (17)– (20) is centralized because $P_{\text{global}}, P_i^\theta, P_i^r$ each contains information from all sensors. We will obtain the implementable, decentralized local controller $u_i = K(\cdot)$ from (10) by replacing any unavailable global quantities with local estimates.

C. Distributed Estimator Design

In both the sensor fusion schemes in Sections III and IV, the sum of the information from each individual sensors is used to calculate the global information P_{global} (also P_i^θ, P_i^r). This motivates the usage of a *PI dynamic average consensus estimator* [8] for our task of estimating the global information. For n agents, assume each agent i has an input $u_i(t) \in \mathbb{R}^{k\times r}$, internal states $v_i, w_i \in \mathbb{R}^{k\times r}$ and output $y_i = v_i$. The PI estimator is given by the following equations (see [8] for details):

$$\begin{aligned}\dot{v}_i =& -\gamma v_i - K_p \sum_{j\in N_i}\big[v_i - v_j\big] \\ &+ K_i \sum_{j\in N_i}\big[w_i - w_j\big] + \gamma u_i\end{aligned} \tag{21}$$

$$\dot{w}_i = -K_i \sum_{j\in N_i}\big[v_i - v_j\big]. \tag{22}$$

Here $\gamma > 0$ is a design parameter, $\mathcal{N}_i$ contains all one-hop neighbors of agent i in the communication network, and K_p, K_i are estimator gains. When the network is connected over time, each estimator output y_i will track the global signal $\frac{1}{n}\sum_{i=1}^n u_i$ asymptotically (see [8] for more rigorous discussions). Since each agent only transmits its estimator states v_i, w_i, this is a fully scalable distributed design.

The following two sections describe two sensor fusion schemes to obtain P_{global}. In each one the explicit form of the derived motion controller depends on the choice of cost functions (D-optimal or A-optimal) and sensor models (range-bearing, range-only). In Sections III and IV we derive these equations for range-bearing sensors. The more complicated case with the range-only sensors are dealt with in Section VI.

III. One-Time Measurement Approach

An instantaneous fusion of current sensor readings leads to the following relations [3], [15]:

$$P_{\text{global}}^{-1}\hat{x}_{\text{global}} = \sum_{i=1}^n H_i^T (T_i R_i T_i^T)^{-1} z_i = \sum_{i=1}^n H_i^T T_i R_i^{-1} T_i^T z_i \tag{23}$$

$$P_{\text{global}}^{-1} = \sum_{i=1}^n H_i^T (T_i R_i T_i^T)^{-1} H_i = \sum_{i=1}^n H_i^T T_i R_i^{-1} T_i^T H_i\,, \tag{24}$$

We further use the rules in (13) – (16) to find P_i^r, P_i^θ:

$$\begin{aligned}P_i^r &= \frac{\partial}{\partial r_i}(\sum_{i=1}^n H_i^T T_i R_i^{-1} T_i^T H_i)^{-1} \\ &= -P_{\text{global}}\frac{\partial}{\partial r_i}(\sum_{i=1}^n H_i^T T_i R_i^{-1} T_i^T H_i)P_{\text{global}} \\ &= -P_{\text{global}}\frac{\partial}{\partial r_i}(H_i^T T_i R_i^{-1} T_i^T H_i)P_{\text{global}}\end{aligned} \tag{25}$$

and similarly

$$P_i^\theta = -P_{\text{global}}\frac{\partial}{\partial \theta_i}(H_i^T T_i R_i^{-1} T_i^T H_i)P_{\text{global}}. \tag{26}$$

For range-bearing sensors, we plug in $H_i = I_2$ and (3), (11) into (25), (26):

$$P_i^r = 2a_2(r_i - a_1)P_{\text{global}} T_i R_i^{-2}\begin{bmatrix}1 & 0\\ 0 & \alpha\end{bmatrix} T_i^T P_{\text{global}} \tag{27}$$

$$P_i^\theta = P_{\text{global}}\big(A_i + A_i^T\big)P_{\text{global}} \tag{28}$$

with

$$A_i = \begin{bmatrix}0 & -1\\ 1 & 0\end{bmatrix} T_i R_i^{-1} T_i^T\,. \tag{29}$$

We implement a decentralized version of the resulting gradient control law (10) as follows. Each agent runs a PI average consensus estimator with local matrix input $u_i = nT_iR_i^{-1}T_i^T$ (for a total of 3 scalar estimators due to the symmetry of this 2×2 matrix), but with the unknown quantities r_i and θ_i replaced by the measurements

$$r_i \approx |p_i - z_i|\,, \qquad \theta_i \approx \angle(p_i - z_i)\,. \tag{30}$$

The inverse of the output of this estimator is P_i, the local estimate of P_{global}. Each agent runs a second average consensus estimator with local vector input $nT_iR_i^{-1}T_i^Tz_i$ (for a total of 2 scalar estimators), again with the replacements (30). The output of this second estimator, when multiplied by P_i, yields $\hat{x}_i$, the local estimate of $\hat{x}_{\text{global}}$. We now evaluate the expressions (8), (27), and (28) by replacing P_{global}

with P_i and using the following filtered versions of the replacements (30):

$$r_i \approx |p_i - \hat{x}_i|\,, \qquad \theta_i \approx \angle(p_i - \hat{x}_i)\,. \tag{31}$$

These replacements lead to the decentralized version of the control law (10) with gradients (17), (18) The same approach applies when we use the control law (19), (20). This implementation assumes the sensor model parameters a_0, a_1, a_2, and α are known to each agent.

IV. Kalman Filter Approach

The approach in Section III fuses sensor readings from current measurements only. To make use of past measurements as well, we can adopt a Kalman filter approach to defining $\hat{x}_{\text{global}}$ and P_{global}. We begin with a linear target model

$$\dot{x}_t = F x_t + G u_t + w\,, \tag{32}$$

where u_t is an exogenous input and w is a continuous-time Gaussian noise with zero mean and covariance matrix Q. We consider the centralized Kalman-Bucy filter [19]:

$$\dot{P}_{\text{global}} = F P_{\text{global}} + P_{\text{global}} F^T + Q - n P_{\text{global}} C P_{\text{global}} \tag{33}$$

$$\dot{\hat{x}}_{\text{global}} = F \hat{x}_{\text{global}} + G u_t + n P_{\text{global}} (y - C \hat{x}_{\text{global}})\,, \tag{34}$$

where C and y are the fused measurements

$$C = \frac{1}{n}\sum_{i=1}^{n} H_i^T T_i R_i^{-1} T_i^T H_I\,, \quad y = \frac{1}{n}\sum_{i=1}^{n} H_i^T T_i R_i^{-1} T_i^T z_i \tag{35}$$

and initial conditions are given by the one-time measurements (23) and (24). The partial derivatives in (12) can be obtained by taking partial derivatives on both sides of (33):

$$\dot{P}_i^r = F P_i^r + P_i^r F^T - n P_i^r C P_{\text{global}} - n P_{\text{global}} C P_i^r + P_{\text{global}} \frac{\partial}{\partial r_i}(H_i^T T_i R_i^{-1} T_i^T H_i) P_{\text{global}} \tag{36}$$

$$\dot{P}_i^\theta = F P_i^\theta + P_i^\theta F^T - n P_i^\theta C P_{\text{global}} - n P_{\text{global}} C P_i^\theta + P_{\text{global}} \frac{\partial}{\partial \theta_i}(H_i^T T_i R_i^{-1} T_i^T H_i) P_{\text{global}} \tag{37}$$

For range-bearing sensors, we plug in $H_i = I_2$ and (3), (11) into (36), (37):

$$\dot{P}_i^r = F P_i^r + P_i^r F^T - n P_i^r C P_{\text{global}} - n P_{\text{global}} C P_i^r + 2a_2(r_i - a_1) P_{\text{global}} T_i R_i^{-2} \begin{bmatrix} 1 & 0 \\ 0 & \alpha \end{bmatrix} T_i^T P_{\text{global}} \tag{38}$$

$$\dot{P}_i^\theta = F P_i^\theta + P_i^\theta F^T - n P_i^\theta C P_{\text{global}} - n P_{\text{global}} C P_i^\theta + P_{\text{global}} \big(A_i + A_i^T\big) P_{\text{global}} \tag{39}$$

with initial conditions calculated according to the one-time measurements (27) and (28).

We implement a decentralized version of the resulting gradient control law (10) as follows. Each agent runs two average consensus estimators, one with local matrix input $T_i R_i^{-1} T_i^T$ and local output C_i, and the other with local vector input $T_i R_i^{-1} T_i^T z_i$ and local output y_i. Each agent also maintains estimates P_i and $\hat{x}_i$ of P_{global} and $\hat{x}_{\text{global}}$ (respectively) by means of the differential equations

$$\dot{P}_i = F P_i + P_i F^T + Q - n P_i C_i P_i \tag{40}$$

$$\dot{\hat{x}}_i = F \hat{x}_i + G u_t + n P_i (y_i - C_i \hat{x}_i) \tag{41}$$

with initial conditions

$$P_i(0) = (T_i R_i T_i^T)(0)\,, \qquad \hat{x}_i(0) = z_i(0)\,. \tag{42}$$

Finally, each agent maintains local copies of the gradients P_i^r and P_i^θ (which we again name P_i^r and P_i^θ with a slight abuse of notation) by means of the differential equations

$$\begin{aligned} \dot{P}_i^r &= \frac{\partial}{\partial r_i}(F P_i + P_i F^T + Q - n P_i C_i P_i) \\ &= F P_i^r + P_i^r F^T - n P_i^r C_i P_i - n P_i C_i P_i^r \\ &\quad + 2a_2(r_i - a_1) P_i T_i R_i^{-2} \begin{bmatrix} 1 & 0 \\ 0 & \alpha \end{bmatrix} T_i^T P_i \end{aligned} \tag{43}$$

$$\begin{aligned} \dot{P}_i^\theta &= \frac{\partial}{\partial \theta_i}(F P_i + P_i F^T + Q - n P_i C_i P_i) \\ &= F P_i^\theta + P_i^\theta F^T - n P_i^\theta C_i P_i - n P_i C_i P_i^\theta \\ &\quad + P_i\big(A_i + A_i^T\big) P_i \end{aligned} \tag{44}$$

with initial conditions given by (27) and (28) but with $P_i(0)$ replacing $P_{\text{global}}(0)$. In all of these equations we use the replacements (31), and we arrive at an implementable version of the local controller (10). This implementation assumes that F, G, Q, u_t, n, and the sensor model parameters a_0, a_1, a_2, and α are known to each agent.

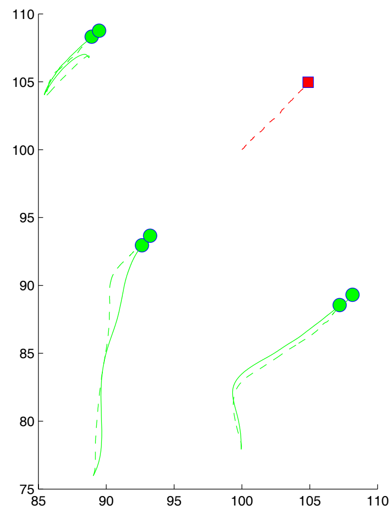

Fig. 3. Trajectories of sensors with a moving target starting from $(100, 100)$. The solid lines denote the Kalman filter scheme and the dashed lines denote the one-time measurement scheme.

V. Simulation Results

We use three range-bearing sensors starting from $(88.73, 106.76)$, $(89.05, 75.98)$, $(99.94, 77.93)$ and a moving target starting from $(100, 100)$. The dynamic model of the target is $\dot{x}_t = u_t + w$ with $u_t = \begin{bmatrix} 0.1 & 0.1 \end{bmatrix}^T$ and $Q = \text{diag}(0.05\ 0.05)$. The sensor model parameters are $a_0 = 0.3528$, $a_1 = 15.625$, $a_2 = 0.0008$, and $\alpha = 5$.

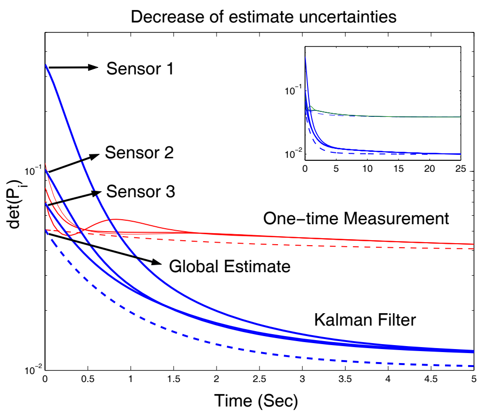

Fig. 4. Comparison of the individual belief uncertainty matrices P_i. The centralized versions P_{global} for each scheme are shown as dotted lines.

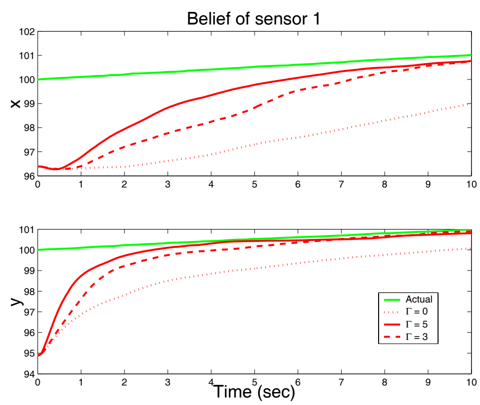

Fig. 5. Comparison between static sensors (Γ=0) and mobile sensors ($\Gamma = 3I, 5I$).

Here we use a radius based communication model. The communication radius is set at $r = 50$ to guarantee the connectedness of the network. We choose a controller gain of $\Gamma = 20I$.

Figure 3 shows the actual trajectories of the sensors. In both sensor fusion schemes, the sensors space themselves from others by 60 degrees (relative to the target). Previous analysis shows this is the optimal collaborative configuration to do a one-measurement sensor fusion [3]. In Figure 4 we compare the performance of these decentralized algorithms with each other and with the centralized versions, where each sensor has access to the correct centralized computation of P_{global}. In both cases, the decentralized schemes recover the results of their centralized counterparts after an initial transient.

Figure 5 compares the performance of static and mobile sensor fusion schemes. Sensors start from the same positions, and in this simulation we changed the parameter a_2 of the sensor model to 0.5 to increase the spatial influence on the measurement noise level. We see that the moving sensors more quickly obtain accurate estimates of the target position. For control gains $\Gamma > 5I$, we start to see oscillatory behaviors in sensor motions and the dynamics of sensor 1's estimate looks almost the same as the case when $\Gamma = 5I$.

VI. Heterogeneous Sensors and Multiple Targets

A. *Heterogeneous Sensors*

Here we address the cases where range-bearing sensors need to collaborate with range-only sensors for the estimation task. The control laws for the range-bearing sensors remain as they are in (27), (28), (38), (39). All we need to do is to derive the controllers for those range-only sensors. We derive the explicit form of P_i^r, P_i^θ for the one-time measurement case, and the Kalman filter case can be done in a similar manner.

From the schematic of the linearized measurement model (Fig. 6), we have:

$$\cos\theta_{i0} = \frac{a + r_i\cos\theta_i}{\sqrt{(a + r_i\cos\theta_i)^2 + (r_i\sin\theta_i - b)^2}} \quad (45)$$

$$\sin\theta_{i0} = \frac{r_i\sin\theta_i - b}{\sqrt{(a + r_i\cos\theta_i)^2 + (r_i\sin\theta_i - b)^2}} \quad (46)$$

Then based on (25) we have

$$P_i^r = P_{\text{global}}\Big(\frac{2a_2(r_i - a_1)}{a_2(r_i - a_1)^2 + a_0}H_i^T H_i - R_i^{-1}\begin{bmatrix}\zeta & \eta \\ \eta & -\zeta\end{bmatrix}\Big)P_{\text{global}} \quad (47)$$

with

$$\zeta = \frac{-2\cos\theta_{i0}\sin\theta_{i0}}{||p_i - x_{t0}||_2}(\sin\theta_i\cos\theta_{i0} - \sin\theta_{i0}\cos\theta_i)$$

$$\eta = \frac{\cos^2\theta_{i0} - \sin^2\theta_{i0}}{||p_i - x_{t0}||_2}(\sin\theta_i\cos\theta_{i0} - \sin\theta_{i0}\cos\theta_i)$$

and similarly

$$P_i^\theta = -\frac{1}{R_i}P_{\text{global}}\begin{bmatrix}\zeta_2 & \eta_2 \\ \eta_2 & -\zeta_2\end{bmatrix}P_{\text{global}} \quad (48)$$

with

$$\zeta_2 = \frac{-2r_i\cos\theta_{i0}\sin\theta_{i0}}{||p_i - x_{t0}||_2}(\sin\theta_i\sin\theta_{i0} + \cos\theta_i\cos\theta_{i0})$$

$$\eta_2 = \frac{r_i\cos(2\theta_{i0})}{||p_i - x_{t0}||_2}(\sin\theta_i\sin\theta_{i0} + \cos\theta_i\cos\theta_{i0}).$$

Now we can finish the distributed design by replacing P_{global} with P_i and using the approximation (30) or (31).

For range-only sensors, singularity issues can arise when implementing these control algorithms. This is because we need to use the invert the estimator output to calculate P_i and the estimator input $u_i = H_i^T T_i R_i^{-1} T_i^T H_i$ has 0 determinant. One solution is to let the estimator run a short time before inverting its output to calculate the control effort; the problematic matrix will become nonsingular when a range-only sensor fuses information with other sensors.

B. *Extension to Multiple Targets*

Here we only consider the case when each sensor is capable of taking multiple measurements at a time and able to distinguish different targets. Otherwise some dynamic *sensor scheduling* and *target association* algorithms need to be developed, which is outside the scope of this paper.

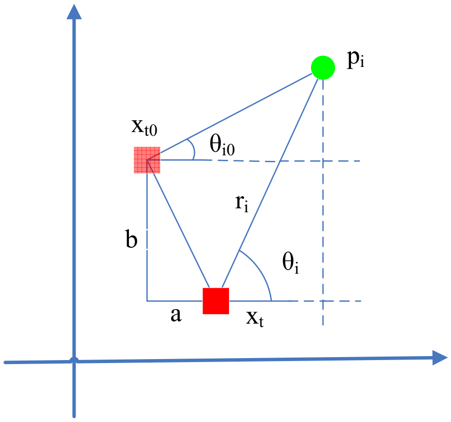

Fig. 6. The linearized measurement model for range-only sensors.

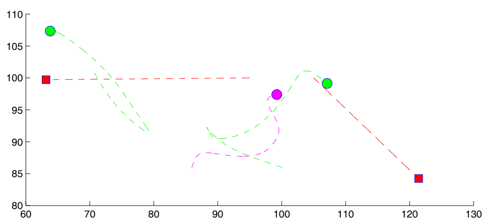

Fig. 7. Two range-bearing sensors (green circle) and one range-only sensor (magenta circle) collaborate to track two moving targets (red square).

Assume we have M targets in total. For each target j, the previous calculation gives a control vector u_{ij} for sensor i, and we can simply add them up in a weighted manner:

$$u_i = \sum_{j=1}^{M} w_{ij} u_{ij}, w_{ij} > 0, \sum_{j=1}^{M} w_{ij} = 1 \tag{49}$$

This approach will retain the distributed nature of the algorithm. In essence, this approach is the same as in [3], which took an algebraic approach by redefining the target state: $X_t = \left[\begin{array}{ccc} x_t^1 & \ldots & x_t^M \end{array} \right]$ and assume the measurement noise for each target is uncorrelated from others. In this approach the number of consensus estimators being used is proportional to the number of targets, and future research effort is needed to deal with this communication constraint.

Here is an illustrative example: two different range-bearing sensors collaborate with a range-only sensor to track the trajectories of two moving targets. Mobile sensors start from $(70.7, 100.8)$, $(100.0, 86.0)$, $(86.0, 85.9)$ and their measurement models are given below:

	a_0^j	a_1^j	a_2^j	α_j
$j=1$	0.1528	15.625	0.0008	5
$j=2$	0.0166	10.800	0.0010	3
$j=3$	0.1100	15.396	0.001	n/a

We set the control gain $\Gamma = 50I$, estimator gain $K_p = 50I, K_i = 0.5I$. The range-only sensor measurement model is linearized around the point $(100, 100)$. Figure 7 shows how sensors divide into groups to track individual targets.

VII. Conclusion

We proposed two distributed solutions for the active sensing problem. By communicating and fusing information with nearest neighbors each sensor gets a global estimate of the target and a local velocity vector field which the mobile sensor can follow to maximize its sensory information. This approach can be extended to deal with the multiple target and heterogeneous sensors cases. Future work includes a stability analysis of the full feedback system and incorporating the sensor localization inaccuracy in the problem formulation.

References

[1] C. Belta and V. Kumar. Abstraction and control for groups of robots. *IEEE Transactions on Robotics*, 20(5):865–875, Oct. 2004.

[2] M. Brookes. Matrix reference manual. In *http://www.ee.ic.ac.uk/hp/staff/dmb/matrix/calculus.html*.

[3] T. H. Chung, J. W. Burdick, and R. M. Murray. Decentralized motion control of mobile sensing agents in a network. In *Intl. Conf. on Decision and Control*, 2005.

[4] J. Cortés, S. Martínez, T. Karatas, and F. Bullo. Coverage control for mobile sensing networks. *IEEE Transactions on Robotics and Automation*, 20(2):243–255, 2004.

[5] J. A. Fax and R. M. Murray. Information flow and cooperative control of vehicle formations. *IEEE Transactions on Automatic Control*, 49(9):1465–1476, Sep 2004.

[6] D. Fox, W. Burgard, and S. Thrun. Active Markov localization for mobile robots. *Robotics and Autonomous Systems*, 25:195–207, 1998.

[7] R. A. Freeman, P. Yang, and K. M. Lynch. Distributed estimation and control of swarm formation statistics. In *American Control Conference*, 2006.

[8] R. A. Freeman, P. Yang, and K. M. Lynch. Stability and convergence properties of dynamic consensus estimators. In *IEEE International Conference on Decision and Control*, 2006.

[9] B. Grocholsky. *Information-Theoretic Control of Multiple Sensor Platforms*. PhD thesis, University of Sidney, 2002.

[10] B. Grocholsky, A. Makarenko, T. Kaupp, and H. Durrant-Whyte. Scalable control of decentralised sensor platforms. In *Int. Workshop on Info. Processing in Sensor Networks*, pages 96–112, 2003.

[11] N. E. Leonard, D. Paley, F. Lekien, R. Sepulchre, D. Fratantoni, and R. Davis. Collective motion, sensor networks, and ocean sampling. *To appear in Proceedings of the IEEE, Special Issue on Networked Control Systems*, 2006.

[12] A. Makarenko and H. Durrant-Whyte. Decentralized data fusion and control in active sensor networks. In *Int. Conf. on Info. Fusion*, 2004.

[13] S. Martínez and F. Bullo. Optimal sensor placement and motion coordination for target tracking. *Automatica*, 2006. To appear.

[14] R. Olfati-Saber. Distributed Kalman filter with embedded consensus filters. In *IEEE International Conference on Decision and Control*, 2005.

[15] K. Ramachandra. *Kalman Filtering Techniques for Radar Tracking*. Marcel Dekker, New York, NY, 2000.

[16] S. Roumeliotis and G. Bekey. Distributed multirobot localization. *IEEE Transactions on Robotics and Automation*, 18(5):781–795, 2002.

[17] S. Simic and S. Sastry. Distributed environmental monitoring using random sensor networks. In *Proceedings of the 2nd International Workshop on Information Processing in Sensor Networks*, pages 582–592, Palo Alto, California, 2003.

[18] D. P. Spanos, R. Olfati-Saber, and R. M. Murray. Dynamic consensus on mobile networks. In *IFAC World Congress*, 2005.

[19] R. F. Stengel. *Optimal control and estimation*. Dover Publications, New York, 1994.

[20] S. Susca, S. Martínez, and F. Bullo. Monitoring environmental boundaries with a robotic sensor network. In *American Control Conference*, pages 2072–2077, 2006.

[21] F. Zhao, J. Shin, and J. Reich. Information-driven dynamic sensor collaboration for tracking applications. *IEEE Signal Processing Magazine*, 19(2):61–72, March 2002.

[22] K. X. Zhou and S. I. Roumeliotis. Optimal motion strategies for distributed range-only tracking. In *Proc. 2006 American Control Conference (ACC'06)*, 2006.