CLUSTER MANAGEMENT IN BEARING-ONLY TRACKING

Özgür Özfidan[1], **Hakan A. Çırpan**[2], **Uluğ Bayazıt**[3] **and** **Azime Can**[4]

[1]PAVO Electronic, Gebze, Kocaeli
e-mail:ozgur.ozfidan@pavo.com.tr

[2,4]Istanbul University, Department of Electrical Engineering,Istanbul
e-mail: {hcirpan,azimecan}@istanbul.edu.tr

[3]Işık University, Department of Electronics Engineering, Istanbul
e-mail: bayazit@isikun.edu.tr

ABSTRACT

Sensor scheduling problem in a classical bearing-only target tracking application is addressed in this paper. In particular, particle filtering algorithm is employed for the tracking, and clustering method is utilized for the scheduling. Cluster scheduling is applied instead of scheduling the individual sensor nodes. Cramér-Rao Lower Bound is used as a decision criteria for the estimation performance of each cluster. Due to non-linearity of the problem, proposed solution for tracking is presented in the framework of non-linear Bayesian estimation.

I. INTRODUCTION

With the advances in semiconductor technology, sensor devices today tend to have smaller form factor and lower prices compared to the past. Large amount of sensor nodes with wireless communication capabilities can now be densely distributed over the area of interest, both in military and civil environments [1]. Bayesian techniques have lately been extensively used in target tracking applications[2], [3]. Due to non-linearity and possibly non-gaussianity of the problem, Sequential Monte Carlo (SMC) Methods [4], [5], are widely used for target tracking applications in Multi-Sensor systems. In [6] interacting multiple model (IMM)-based tracker is presented for the problem of bearing only tracking. In [7] different particle filters for the bearing only tracking of maneuvering target are proposed and Cramér-Rao Lower Bound (CRLB) is employed for the performance criteria of these filters.

In the centralized architecture, sensor nodes share their observation with a data fusion center directly, which requires relatively higher bandwidth compared to the decentralized (Cluster Based) architecture [8], [9]. In the latter case many of the management activities such as sharing the radio resources, collecting, storing and processing the data and several monitoring tasks can be distributed efficiently in the network. In this approach, each sensor passes its own observation about the target state to a local leader node. One immediate result of this choice is the extra processing power requirement of these local leaders. Selection of the leader node is another parameter that can affect the performance of overall network. Leader nodes can be fixed through the lifetime of a network or they can be updated periodically or in a more intelligent fashion by considering the extra power consumption. In the former case, leaders can be equipped with longer life batteries and higher output transmit power RF communication ICs. [10] and [11] introduce Distance-based Scheduling (DS) and Balanced-energy scheduling (BS) schemes, and choose the sensor nodes to be kept in sleeping state with the assumption that energy consumption of the nodes farther away from the cluster leader is more than the others.

In this paper, we present a novel sensor scheduling method based on partitioning the sensors into clusters. Our clustering algorithm depends on associating each slave sensor to its nearest master node. Initially, we assumed a totally random deployment strategy into the region of interest. After grouping the sensors, we used the information collected by the sensors in the same cluster to track the position of our target. Figure 1 describes such a tracking scenario.

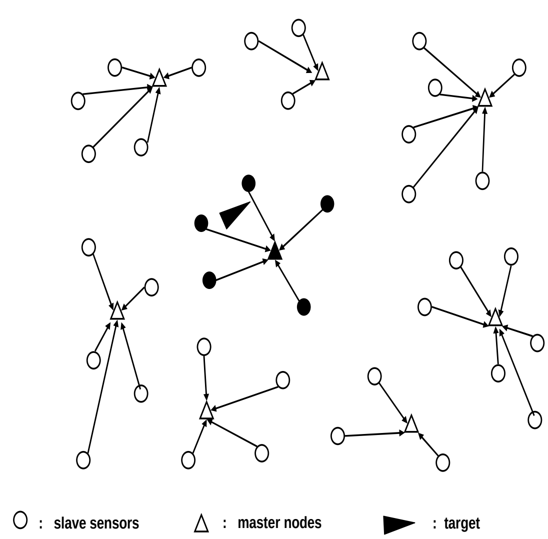

Figure 1. *Tracking Scenario: Black triangle and circles represent the sensors in the active cluster. Arrows indicate the master node that each slave report.*

II. SCENARIO

In this section, we introduce a general model for our tracking scenario. We consider the task of tracking a moving vehicle through our two dimensional stationary sensor field under surveillance while conserving power by minimizing the number of active sensors. Before we run our tracking algorithm there is a set-up procedure which works as follows : First, we randomly distribute both the slave sensors and the master nodes into our region of interest. Master nodes are basically responsible for communicating with the data fusion center. Remaining sensors will be called slaves. Slave sensors report the position of the target to their master periodically or if there is no target detected, they report this situation as well . After randomly distributing both type of sensors, we associate each slave with a master by running our master-slave association algorithm. Basic criteria for this process is the Cartesian distance between the master nodes and the slave sensors. Each slave is associated with its closest master. Another practical real world constraint that we take into account at this point is the service capacity of a master node. Maximum number of slaves that we can associate with each master is defined. During the set-up process if this limit is exceeded for a master, than the remaining slaves are associated with another master.

As mentioned before, our main objective is to accurately track the target while minimizing the number of active sensors. Only the active sensors provide observation about target position, otherwise they are configured to remain in sleep mode to reduce the power consumption. Thus, activation of sensors within a specified distance from the current target position estimate is quite important. Several different formulations of this problem are possible as target of interest moves through our randomly distributed sensors. Our approach at this point is simply to compare the current position estimate of the target with the position of each master node at every time step and to activate the associated slaves of the closest master for the next epoch. Here, we are using the assumption that master nodes are always active and thus leadership can be immediately transferred from one master to another. Every master can activate its own slaves whenever needed.

III. TRACKING AND SCHEDULING

In this section, we introduce a general model for our multi-sensor, single target system. Target velocity is assumed to be constant during the tracking phase. Sensors are assumed to be bearing only sensors. After running our clustering algorithm we employ particle filter to estimate the position of our target based on reported bearing information by the slave sensors in the same cluster.

III-A. Dynamics

Now we define the system and observation models for our target in a detailed manner. For the 2-dimensional case, state vector $\mathbf{X}_k$ at time step k contains four elements: positions in the x and y directions and velocities in the x and y directions:

$$\mathbf{X}_k = [x_k, y_k, \dot{x}_k, \dot{y}_k]^T \tag{1}$$

Kinematics for the target can be written as

$$x_k = x_{k-1} + \dot{x}_{k-1}\Delta t + \frac{1}{2}\ddot{x}_{k-1}\Delta t^2 + \frac{1}{3}\dddot{x}_{k-1}\Delta t^3 \tag{2}$$

$$y_k = y_{k-1} + \dot{y}_{k-1}\Delta t + \frac{1}{2}\ddot{y}_{k-1}\Delta t^2 + \frac{1}{3}\dddot{y}_{k-1}\Delta t^3 \tag{3}$$

where Δt is the time difference between state transitions or simply the sampling period. The parameters $\ddot{x}_k$ and $\ddot{y}_k$ represent the acceleration in the x and y directions, respectively. Finally, $\dddot{x}_k$ and $\dddot{y}_k$ are to represent the variations in the acceleration in two directions again. We model the acceleration components using random noise. Assuming target moves with a constant velocity, using (2) and (3), the state equation can be written as

$$\mathbf{X}_k = F\mathbf{X}_{k-1} + Q^{1/2}\mathbf{V}_{k-1} \tag{4}$$

where

$$F = \begin{pmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{5}$$

$$Q = q \begin{pmatrix} \Delta t^3/3 & 0 & \Delta t^2/2 & 0 \\ 0 & \Delta t^3/3 & 0 & \Delta t^2/2 \\ \Delta t^2/2 & 0 & \Delta t & 0 \\ 0 & \Delta t^2/2 & 0 & \Delta t \end{pmatrix} \tag{6}$$

where Q is the state error covariance matrix and models the acceleration terms in the x and y directions. The vector $\mathbf{V}_k$ is a Gaussian random vector of zero mean, unit variance and independent components. Finally, q is used to control the intensity of the process noise.

The observation vector can simply be related to the state vector as

$$\mathbf{Z}_k = \mathbf{\Theta}_k + R^{1/2}\mathbf{n}_k. \tag{7}$$

$\mathbf{\Theta}_k$ is an L x 1 vector whose elements are the angle between the target and the each slave sensor used to generate observations at time step k, where L is the number of slaves in the corresponding cluster. The ith element of $\mathbf{\Theta}_k$ is

$$\Theta_{ik} = tan^{-1}\frac{(y_k - y^{s_i})}{(x_k - x^{s_i})} \quad i = 1, 2, ...L \tag{8}$$

where (x_k, y_k) is the target position and (x^{s_i}, y^{s_i}) is the position of ith sensor in the corresponding cluster. R denotes the measurement error covariance matrix and $\mathbf{n}_k$ is an L x 1 vector whose elements are generated by a Gaussian random variable of zero mean and unit variance.

Intuitively, it can be said that Signal to Noise Ratio (SNR) will decrease with increasing slave-target distance. Defining

Received and Transmitted Signal Powers as P_r and P_t respectively, we can write

$$P_r = KP_t. \tag{9}$$

K represents both the attenuation due to the channel characteristics and the reflection depending on the target material. Then, SNR can simply be expressed as

$$SNR = P_r/\sigma^2. \tag{10}$$

This situation can be modeled in the observation vector above, by increasing the variance of measurement error with sensor-target distance. Furthermore, if we assume that the noise components for each sensor are independent, R becomes an LxL diagonal matrix whose elements are directly proportional to the slave-target distance, where L is the number of slaves in the active cluster.

$$R_{ii} \propto \sqrt{(y_k - y^{s_i})^2 + (x_k - x^{s_i})^2} \tag{11}$$

III-B. Clustering

For the maneuvering target tracking application we can assume our region of interest under surveillance is quite a large area and only a small portion of deployed sensors can provide useful information at a specific time instance. That is why, clustering the bearing only sensors is one of the most reasonable strategy that can be applied. By clustering, we can reduce the spatial coverage of sensors considerably, which means higher quality of data reported by each sensor.

Now we present unbalanced clustering algorithm for the efficient data collection. As mentioned before, initially both Master nodes and Slave sensors are distributed into the region of interest randomly.

```
S ≡ Number of slave sensors
M ≡ Number of master nodes
C ≡ Max service capacity of a master
m ≡ Master nodes
s ≡ Slave sensors

Calculate distance from all slaves to all masters
for i = 1, 2, ··· , M
   for j = 1, 2, ··· , S
      D[i, j] = ||m_i − s_j||
   end
end

Assign each slave to its closest master
for j = 1, 2, ··· , S
   [mindist, i] = min(D[:, j])
   if capacity of ith master is smaller than C
      Assign slave j to master i
   else
      Assign slave j to another master
   end
end
```

Table I. Master-Slave Association Algorithm

III-C. Particle Filtering

We start filtering with the data reported by the slaves in the cluster whose master is closest to this initial position. Given $p(\mathbf{X}_k|\mathbf{X}_{k-1})$ and $p(\mathbf{Z}_k|\mathbf{X}_k)$ Generic Particle Filter and Resampling algorithms [2] are used to recursively estimate the state of moving target.

We approximate the posterior density $p(\mathbf{X}_k|\mathbf{Z}_{1:k})$ at time step k by a set of particles $\{\boldsymbol{x}_k^i, i = 1, 2, ...N\}$ and associated weights $\{\omega_k^i, i = 1, 2, ...N\}$ where $\sum_{i=1}^N \omega_k^i = 1$. We draw the particles from a proposal distribution $q(\boldsymbol{x}_k^i|\boldsymbol{x}_{k-1}^i) = p(\boldsymbol{x}_k^i|\boldsymbol{x}_{k-1}^i)$ and assign each particle a weight using the weight update equation $\omega_k^i \propto \omega_{k-1}^i p(\mathbf{z}_k|\boldsymbol{x}_k^i)$. Approximation to the posterior density is then

$$p(\boldsymbol{x}_k|\mathbf{z}_{1:k}) \approx \sum_{i=1}^N \omega_k^i \delta(\boldsymbol{x}_k - \boldsymbol{x}_k^i) \tag{12}$$

And finally the state estimation is

$$\hat{\boldsymbol{x}}_k \approx \sum_{i=1}^N \omega_k^i \boldsymbol{x}_k^i. \tag{13}$$

III-D. Posterior Cramér-Rao Bounds

Let $\hat{\boldsymbol{x}}_k$ be an unbiased estimator of $\boldsymbol{x}_k$. Cramér-Rao Lower Bound (CRLB) on the error covariance of estimator is defined to be the inverse of Fisher Information Matrix (FIM).

$$\mathsf{E}[(\hat{\boldsymbol{x}}_k - \boldsymbol{x}_k)(\hat{\boldsymbol{x}}_k - \boldsymbol{x}_k)^T] \geq \mathbf{J}_k{}^{-1} \tag{14}$$

$\mathbf{J}_k$ can be calculated recursively as follows[12].

$$\mathbf{J}_{k+1} = D_k^{33} - D_k^{21}[\mathbf{J}_k + D_k^{11}]^{-1} D_k^{12} + \mathbf{J}_z(k+1) \tag{15}$$

where

$$D_k^{11} = \mathsf{E}[(\nabla_{\boldsymbol{x}_k} \ln p(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k))(\nabla_{\boldsymbol{x}_k} \ln p(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k))^T] \tag{16}$$

$$D_k^{12} = \mathsf{E}[(\nabla_{\boldsymbol{x}_k} \ln p(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k))(\nabla_{\boldsymbol{x}_{k+1}} \ln p(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k))^T] \tag{17}$$

$$D_k^{33} = \mathsf{E}[(\nabla_{\boldsymbol{x}_{k+1}} \ln p(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k))(\nabla_{\boldsymbol{x}_{k+1}} \ln p(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k))^T] \tag{18}$$

$$D_k^{21} = (D_k^{12})^T \tag{19}$$

$$\mathbf{J}_z(k+1) = \mathsf{E}[(\nabla_{\boldsymbol{x}_{k+1}} \ln p(\mathbf{z}_{k+1}|\boldsymbol{x}_{k+1}))(\nabla_{\boldsymbol{x}_{k+1}} \ln p(\mathbf{z}_{k+1}|\boldsymbol{x}_{k+1}))^T] \tag{20}$$

Initial FIM can be written as

$$\mathbf{J}_0 = \mathsf{E}[(\nabla_{\boldsymbol{x}_0} \ln p(\boldsymbol{x}_0))(\nabla_{\boldsymbol{x}_0} \ln p(\boldsymbol{x}_0))^T] \tag{21}$$

If process model is linear it can be shown that recursive formulation of $\mathbf{J}_k$ reduces to

$$\mathbf{J}_{k+1} = (Q_k + F_k \mathbf{J}_k {F_k}^T)^{-1} + \mathbf{J}_z(k+1) \tag{22}$$

In (22) computation of $(Q_k + F_k \mathbf{J}_k {F_k}^T)^{-1}$ is trivial. Furthermore, if measurement error is zero mean Gaussian with covariance R_k it can be shown that

$$\mathbf{J}_z(k+1) = \mathsf{E}[{H_{k+1}}^T {R_{k+1}}^{-1} H_{k+1}] \tag{23}$$

where H_k is the Jacobian of the nonlinear function $h_k(.)$

$$H_k = [\nabla_{\boldsymbol{x}_k}[h_k(\boldsymbol{x}_k)]^T]^T \tag{24}$$

and for our range only sensors

$$h^i{}_k(\boldsymbol{x}_k) = tan^{-1}\frac{(y_k - y^{s_i})}{(x_k - x^{s_i})} \tag{25}$$

where (x^{s_i}, y^{s_i}) is the position of ith slave in the corresponding cluster.

$$H_k{}^T = \begin{pmatrix} \frac{\partial h^1{}_k(\boldsymbol{x}_k)}{\partial x_k} & \frac{\partial h^2{}_k(\boldsymbol{x}_k)}{\partial x_k} & .. & \frac{\partial h^L{}_k(\boldsymbol{x}_k)}{\partial x_k} \\ \frac{\partial h^1{}_k(\boldsymbol{x}_k)}{\partial y_k} & \frac{\partial h^2{}_k(\boldsymbol{x}_k)}{\partial y_k} & .. & \frac{\partial h^L{}_k(\boldsymbol{x}_k)}{\partial y_k} \\ 0 & 0 & .. & 0 \\ 0 & 0 & .. & 0 \end{pmatrix} \tag{26}$$

$H_k{}^T$ is 4xL matrix where L is the number of slave sensors in a cluster. Since we don't have observation about velocities in x and y directions, last two elements of each column is zero.

III-E. Cluster Scheduling

In this section, we develop a sensor scheduling method for our randomly distributed sensors. Note that sensors were initially partitioned into clusters by running Master-Slave association algorithm given in Table I. Then we estimate the position of our target by using the sensors in each cluster with the particle filtering algorithm. We activate the cluster whose calculated inverse Fisher Information Matrix is smallest compared to other clusters. In this case cost function is

$$C_k^i = \mathbf{tr}(\mathbf{J}^i{}_k)^{-1} \tag{27}$$

Finally our scheduling decision is that we choose the master node for which the cost function is minimized.

$$m_{opt} = argmin_i C_k^i \tag{28}$$

IV. RESULTS AND DISCUSSION

In this section, we discuss an example of target tracking using our proposed sensor scheduling algorithm. For the simulations, the trajectory for a target was generated in a 2-dimensional cartesian coordinate system.

Initially, 64 master nodes and 256 slave sensors were distributed randomly in the area x = (-1500,1500) and y = (-1500,1500). Then, our Master-Slave Association algorithm was applied. Maximum number of slaves that a master can give service was assumed to be 5. Sampling period, Δt was chosen to be 2 seconds. The process noise intensity factor q was taken as 0.01 and the initial position of target was taken to be $(x, y) = (0, 0)$. For the particle filter algorithm we used a total of 200 particles and 500 time steps. Resampling applied when $\widehat{N_{eff}}$ is below the threshold 40.

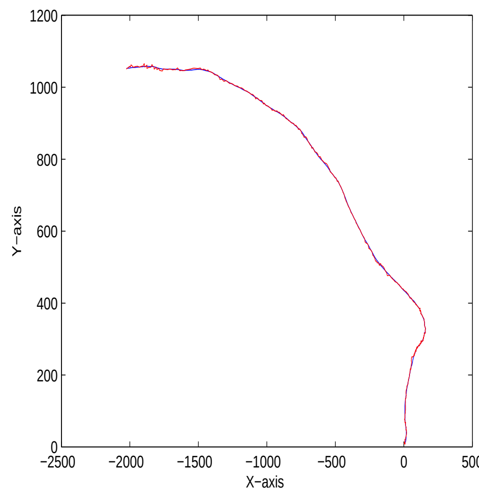

Figure 2. *True and Estimated trajectories.*

During the tracking phase, at each time step k, we have updated measurement error covariance matrix R by calculating the distance between each slave and the target position.

$$R = R_{coeff} R' \tag{29}$$

$$R' = \frac{1}{d_{max}} diag([d_1, d_2, ...d_L]^T) \tag{30}$$

where

$$d_i = \sqrt{(y_k - y^{s_i})^2 + (x_k - x^{s_i})^2}, i = 1, 2, ...L \tag{31}$$

d_{max} is the normalizing constant and L is the number of slaves in the active cluster. Constant R_{coeff} was set to 100.

True and Estimated target trajectories using Particle Filter are shown in Figure 2. Activated sensors throughout the tracking phase are shown in Figure 3. Rms position errors in x and y directions are shown in Figure 4.

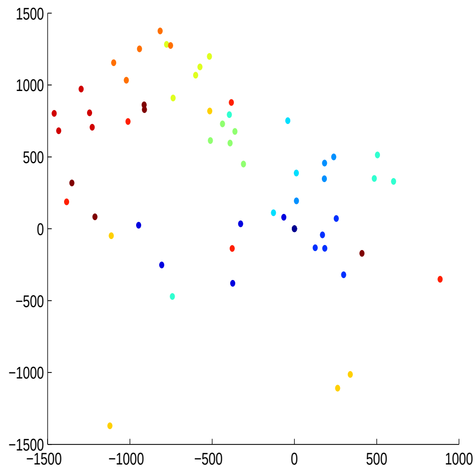

Figure 3. *Activated sensors.*

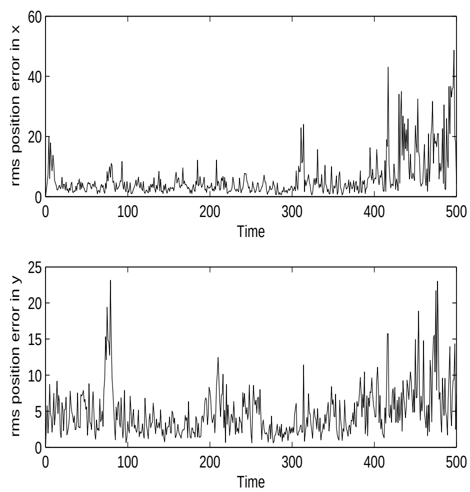

Figure 4. *rms position errors in x and y directions.*

V. CONCLUSION

We have presented a recursive Bayesian formulation for target tracking and proposed a simple cluster scheduling technique in order to reduce power consumption of the system. In particular, we have formulated the target tracking problem using state-space equations. Tracking was considered as a sequential estimation problem and particle filtering algorithm was implemented. In order to schedule the sensors in our region of interest we have utilized Cramér-Rao Lower Bound criteria on the error performance of particle filter. We observed that our scheduling results are still quite satisfactory when we take into account the decreasing detection quality of bearing only sensors with increasing distance. It is evident that, over all power consumption of the system is extremely low when compared to the case where no scheduling is done.

REFERENCES

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, “A survey on sensor networks,” *IEEE Communications Magazine* , vol. 40, No. 8, Aug. 2002.

[2] M. S. Arulampalam, S. Maskell, N. Gordon,and T. Clapp, “A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking,” *IEEE Trans. on Signal Processing.*, Vol 50,No. 2, Feb. 2002.

[3] N. Gordon, D. Salmond, and A. Smith, “A Novel Approach to Non-Linear and Non-Gaussian Bayesian State Estimation,” *IEEE Proc.-F*, vol. 140, pp. 107-113, 1993.

[4] C. Kreucher, K. Kastella, and A. O. Hero, “Multitarget Tracking using the Joint Multitarget Probabilty Density,” *IEEE Trans. on Aerospace and Electronic Systems* , vol. 41, No. 4, October 2005.

[5] C. Hue, J.P. Le Cadre, and P. Pérez, “Sequential Monte Carlo Methods for Multiple Target Tracking and Data Fusion,” *IEEE Trans. on Signal Processing.*, Vol 50,No. 2, Feb. 2002.

[6] T. Kirubarajan, Y. Bar-Shalom, and D. Lerro, “Bearings-only tracking of maneuvering targets using a batch-recursive estimator,” *IEEE Transactions on Aerospace and Electronic Systems* , vol. 37, no. 3, pp. 770-780, 2001.

[7] M. S. Arulampalam, B. Ristic, N. Gordon, T. Mansell, “Bearings-Only Tracking of Manoeuvring Targets Using Particle Filters,” *EURASIP Journal on Applied Sig Proc.* ,vol. 2004 (15), pp. 2351-2365, 2004.

[8] S. Ghiasi, A. Srivastava, X. Yang, and M. Sarrafzadeh, “Optimal Energy Aware Clustering in Sensor Networks,” *Sensors 2002,2,258-269*, July 2002.

[9] X. Sheng, Y-H. Hu, “Distributed Particle Filter with GMM Approximation for multiple target localization and tracking in wireless sensor networks,” *Proc. IEEE/ACM Int. Symp. IPSN, LA, CA* , Apr 2005.

[10] J. Deng, Y.S. Han, W.B. Heinzelman, P.K. Varshney, “Scheduling sleeping nodes in high density cluster-based sensor networks,” *ACM/Kluwer Mobile Networks and Applications (MONET) Special Issue on Energy Constraints and Lifetime Performance in Wireless Sensor Networks*, April 2005.

[11] J. Deng, Y.S. Han, W.B. Heinzelman and P.K. Varshney, “Balanced-energy sleep scheduling scheme for high-density cluster-based sensor networks,” *Computer Communications*, vol. 28, pp. 1631-1642, Apr. 2005.

[12] P. Tichavsky, C.H. Muravchik, and A. Nehorai, “Posterior Cramér-Rao Bounds for Discrete-Time Nonlinear Filtering,” *IEEE Trans. on Signal Processing.*, 46(5),pp. 1386-1396, 1998.