Article

Energy Efficient Sensor Scheduling with a Mobile Sink Node for the Target Tracking Application

Suhinthan Maheswararajah [1,*], **Saman Halgamuge** [1] and **Malin Premaratne** [2]

[1] Department of Mechanical Engineering, Melbourne School of Engineering, University of Melbourne, Australia; E-mail: saman@unimelb.edu.au (S.H.)

[2] Advanced Computing and Simulation Laboratory, Department of Electrical and Computer Systems Engineering, Monash University, Clayton, Victoria, Australia; E-mail: malin@eng.monash.edu.au (M.P.)

[*] Author to whom correspondence should be addressed; E-mail: s.maheswararajah@pgrad.unimelb.edu.au (S.M.)

Abstract: Measurement losses adversely affect the performance of target tracking. The sensor network's life span depends on how efficiently the sensor nodes consume energy. In this paper, we focus on minimizing the total energy consumed by the sensor nodes whilst avoiding measurement losses. Since transmitting data over a long distance consumes a significant amount of energy, a mobile sink node collects the measurements and transmits them to the base station. We assume that the default transmission range of the activated sensor node is limited and it can be increased to maximum range only if the mobile sink node is out-side the default transmission range. Moreover, the active sensor node can be changed after a certain time period. The problem is to select an optimal sensor sequence which minimizes the total energy consumed by the sensor nodes. In this paper, we consider two different problems depend on the mobile sink node's path. First, we assume that the mobile sink node's position is known for the entire time horizon and use the dynamic programming technique to solve the problem. Second, the position of the sink node is varied over time according to a known Markov chain, and the problem is solved by stochastic dynamic programming. We also present sub-optimal methods to solve our problem. A numerical example is presented in order to discuss the proposed methods' performance

1. Introduction

In a sensor network, a large number of small, low cost sensing devices called sensor nodes are deployed and interconnected to gather information from the field of interest and transmit it to the base station. The gathered information may be used to perform various kinds of tasks, such as environmental monitoring, habit monitoring, intelligent building and military applications. Each sensor node in the sensor field is powered by a battery and consumes energy for various purposes such as sensing, on board signal processing and transmitting. Energy required for transmission is significantly more than other purposes and is a function of the distance between sender and receiver. In general, sensor nodes are stationary and unattended. It is not easy and in most cases impractical to replace or recharge the battery in hazardous environments [1, 2]. Once the battery runs out, the sensor node becomes inoperable, increasing the coverage hole and decreasing the connectivity of the whole network [2, 3], on which the sensor network's performance depends highly. Therefore, efficient use of the sensor node battery's energy is an important aspect of sensor networks.

The life span of a sensor network is determined by the time duration until the sensor network fails to function due to inadequate number of sensor nodes [2, 4]. Many strategies have been proposed to prolong the life span of sensor networks [5, 6, 7, 8]. Activating a set of sensor nodes and putting the rest of the sensor nodes in *sleep* mode is a technique widely used to prevent unwanted energy consumption. In [9], the low-energy adaptive clustering hierarchy (LEACH) was proposed to distribute the energy consumption evenly throughout the sensor nodes, doubling the life span of the network. The alternative strategies for long distance transmission, such as multi-hop transmission and using a mobile sink node to perform the transmission [10, 11] also save significant energy. Data latency is experienced in multi-hop transmission, and the lower the latency the better the performance, especially in time-critical applications. Generally a mobile sink node has more battery energy, but navigating the mobile sink node is a challenging task. Many solutions have been proposed to solve this navigation problem [12, 13].

In target tracking, continuous collection of information about the target improves the tracking quality. The loss of the target and the measurement losses seriously affect the tracking quality [14]. Moreover in [15], an information theoretic approach was proposed to estimate the number of targets and their states simultaneously, where the expected information from the targets are maximized by selecting appropriate sensor nodes. In this paper, we show that for a linear Gaussian dynamics, the measurement losses adversely affect the tracking quality. In literature, the sensor scheduling was mainly studied in two different ways: first, scheduling the sensor nodes to minimize the tracking error subject to limited sensor usage and communication costs. Second, scheduling the sensor nodes to minimize the cost of sensor usage and communication subject to constraints on tracking quality. Our previous work [16] focused on maximizing the tracking quality, subject to limited battery energy available at the sensor nodes and we used several techniques to solve this constrained problem. In [17], the energy constraint was relaxed using Lagrangian multipliers and the problem was solved using an approximate dynamic programming technique. Predicting the target and activating a set of sensor nodes closer to the target, rather than activating all the sensor nodes, conserves energy in the sensor network and energy efficient sensor scheduling algorithm was presented in [18]. The optimal sensor scheduling was considered as minimizing the usage of

sensor resources (i.e., battery energy and number of sensor nodes) whilst keeping the tracking accuracy at a desired level [19, 20, 21].

In this study, we minimize the total energy consumed by the sensor nodes whilst avoiding the additional error in the estimation due to the measurement losses. A mobile sink node is used to collect the measurements and transmit them to the base station (BS). The BS schedules the sensor nodes to send the measurements to the mobile sink node without loss considering the minimal energy consumption. In many situations, multiple sensor nodes cannot be activated at the same time due to the limited bandwidth or avoid interferences between sensor nodes. For example sonar sensor nodes cannot be operated simultaneously in the same frequency band in order to avoid interferences [22] and, therefore, a single sensor node is allowed to activate at each time step. In [23], it was assumed that once a sensor node is activated it remains active for a certain time period and can only be changed to another sensor node if the active time period of a sensor node is elapsed. Moreover switching between different types of sensor nodes caused additional cost [24]. In this paper, we assume that a single sensor node can be activated at any time step and each sensor node has an active time period. Therefore, a sensor node cannot be activated during the active period of a previously activated sensor node. We use deterministic and stochastic dynamic programming to solve this problem optimally when the path of the mobile sink node is known fully and partially, respectively. Moreover, we propose sub-optimal methods for both cases, and discuss the performance of the proposed methods with respect to the end result as well as to the computational cost.
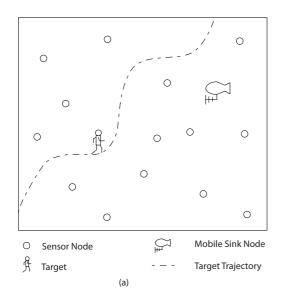
The remainder of this paper is organized as follows. Section II constitutes the overview of our problem and analyzes the effect of measurement losses on tracking quality for a linear Gaussian dynamics. In section III, the assumptions and constraints made for this study are presented to formulate the sensor management problem. We describe optimal and sub-optimal methods to solve the sensor management problem for fully and partially known mobile sink node's paths. In Section IV, a numerical example of our problem is presented to analyze the performance and limitations of the proposed methods. Finally, our conclusion and future work are presented in section V.
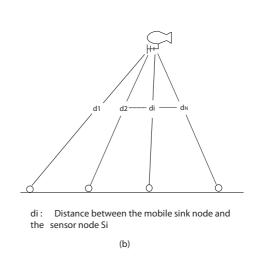
2.   Problem description and Models

Figure 1 shows a wireless sensor network which is located far form the base station. Several sensors are deployed to track a target which moves in the field. In each time step a known set of sensor nodes cover the target and the sensor measurements are transmitted to a mobile sink node, for example an unmanned aerial vehicle (UAV) is flying over the sensor field. The mobile sink node transmits the collected measurements directly to the base station. The energy required for a sensor node to transmit the measurement depends on the distance between the sensor node and the mobile sink node. The distance is illustrated in Figure 1(b). An activated sensor node far away from the mobile sink node consumes more energy to transmit the measurement, compared to one closer to the mobile sink node and therefore the energy is wasted. Alternatively we may follow two approaches when the activated sensor node is far away from the mobile sink node: first, activating another sensor node closer to the mobile sink node: Second, not transmitting the measurement over a long distance. Once a sensor node is activated it cannot be changed to another one for a certain time period and therefore the first approach

is void. Since measurement losses degrade the tracking quality, the second approach is strictly avoided. Therefore, the base station should manage the sensor nodes to minimize the total waste of energy without degrading the tracking quality. The energy model of the sensor node and the importance of availability of measurements for target tracking are presented below.

Figure 1. (a) Tracking a single target using multiple sensor nodes and a mobile sink node in the sensor network. (b) The Euclidean distances between the sensor nodes and the mobile sink node.

2.1. Sensor Model

The base station (BS) schedules the sensor nodes to either *sleep* or *active* mode to perform the tasks efficiently. In order to satisfy the limited bandwidth, one sensor node (or generally a set of sensor nodes) is in $active$ mode at any given instant. We assume that the position of the target belongs to a region denoted by Ω_k at time step k and it is known a priori. Let $\{S_1, S_2, ..., S_{N_k}\}$ denote the set of feasible sensor nodes for scheduling at time step k and sensor node S_i is an element in the set $\{S_1, S_2, ..., S_{N_k}\}$ only if its sensing range covers the region Ω_k. Here, N_k is the total number of feasible sensor nodes available for the scheduling at time step k. If a sensor node S_i is activated at time step k, the BS cannot activate another sensor node S_j for a period of t_i time steps (ie, BS can only activate the S_j sensor node at $k + t_i$ time step or later). Each sensor node consumes energy for sensing, signal processing and transmitting. The energy consumption of the sensor node during its *active* and *sleep* modes is assumed as follows:

$$\psi = \begin{cases} \psi_1 + \psi_t & \text{if } active \\ \psi_2 & \text{if } sleep \end{cases} \tag{1}$$

The power consumption by transmission is denoted by ψ_t and ψ_1 represents the power required for sensing and signal processing. The power consumption due to its own timer during the *sleep* mode

is denoted by ψ_2. In *active* mode the sensor node consumes more energy than in *sleep* mode, due to particularly the transmission. In order to transmit r bits of data to a distance of d, the sensor node requires $(\alpha_1 + \alpha_2 d^2)r$ energy at each time step [25], where α_1 denotes the electronic energy required to transmit one bit of data and α_2 is a constant related to the radio energy. The transmission range of the sensor node s_i is set at default range r_i. However, it can be set at $r_i^{\max}$ $(> r_i)$ only when the mobile sink node is not reachable by the sensor node with r_i. The intuition behind this method is to prolong the life span of the sensor network by avoiding unnecessary long distance transmissions.

Moreover, the sensor nodes' measurements are linearly related to the state of the target and corrupted by white Gaussian noise. The measurement from the sensor node s_i at time step k is given by:

$$y_k^i = \mathrm{H}^i \mathrm{X}_k + v_k^i \tag{2}$$

The column vector $\mathrm{X}_k = \begin{bmatrix} \chi_k & \dot{\chi}_k & \xi_k & \dot{\xi}_k \end{bmatrix}'$ denotes the state of the target at time step k where χ_k and ξ_k represent its positions in the x and y directions, $\dot{\chi}_k$ and $\dot{\xi}_k$ represent its velocities in the x and y directions. The observation matrix is denoted as H^i. The measurement noise v_k^i for each sensor node is assumed to be independent of each other and $v_k^i \sim N(0, \mathrm{R}_i)$. Here R_i denotes the error covariance of the sensor node s_i. In this study, we use homogeneous sensor nodes and therefore, the error covariance R_i and the observation matrix H^i are the same for all the sensor nodes and are known a priori.

2.2. Tracking Quality and Measurement availability

In this sub-section, we analyze the consequences of measurement losses in target tracking applications. Let X_k represent the state of the target at time step k. Then the motion of the target can be modeled by a stochastic equation. In many applications, the stochastic equation is assumed linear [16, 26], non linear [27] and Markov process [28, 29]. In this study, we assume that the target evolves according to the linear stochastic equation impaired by white Gaussian noise and given by:

$$\mathrm{X}_{k+1} = \mathrm{F}\mathrm{X}_k + w_k \tag{3}$$

where w_k is the white Gaussian process noise with a known covariance Q and the intensity of the process noise is determined by the scalar quantity q. F denotes the system matrix which models the state kinematics of the target. For a nearly constant velocity model of the target, F and Q are given by [30]:

$$\mathrm{F} = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad \mathrm{Q} = q \begin{bmatrix} \frac{\Delta t^3}{3} & \frac{\Delta t^2}{2} & 0 & 0 \\ \frac{\Delta t^2}{2} & \Delta t & 0 & 0 \\ 0 & 0 & \frac{\Delta t^3}{3} & \frac{\Delta t^2}{2} \\ 0 & 0 & \frac{\Delta t^2}{2} & \Delta t \end{bmatrix}.$$

The state of the target evolves after Δt time steps. Since the state of the target and the measurements are assumed to have linear Gaussian dynamics, the Kalman filter is used to optimally estimate the state and calculate its error covariance [31]. If the initial state of the target X_0 is known with the error covariance P_0, the estimated state of the target $\xi_{k|k} = \mathbb{E}\{\mathrm{X}_k\}$ and its error covariance $\mathrm{P}_{k|k} = \mathbb{E}\{(\mathrm{X}_k - \xi_{k|k})(\mathrm{X}_k - \xi_{k|k})'\}$ at time step k are given by:

$$\xi_{k|k} = \xi_{k|k-1} + \mathrm{K}_k\left(y_k - z_{k|k-1}\right) \tag{4}$$

$$\mathrm{P}_{k|k} = \left(\mathbb{I} - \mathrm{K}_k\mathrm{H}\right)\mathrm{P}_{k|k-1} \tag{5}$$

where

$$\xi_{k|k-1} = \mathrm{F}\xi_{k-1|k-1}$$

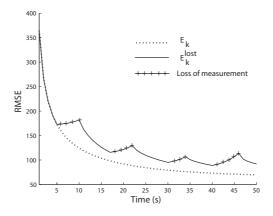$$z_{k|k-1} = \mathrm{H}\xi_{k-1|k-1}.$$

The predicted state and the predicted measurement at time step $k-1$ are denoted as $\xi_{k|k-1}$ and $z_{k|k-1}$ respectively. The predicted error of the estimated state $\mathrm{P}_{k|k-1}$ and the Kalman gain K_k at time steps $k-1$ and k are given as:

$$\mathrm{P}_{k|k-1} = \mathrm{F}\mathrm{P}_{k-1|k-1}\mathrm{F}' + \mathrm{Q}, \tag{6}$$

$$\mathrm{K}_k = \mathrm{P}_{k|k-1}\mathrm{H}'\left(\mathrm{R} + \mathrm{H}\mathrm{P}_{k|k-1}\mathrm{H}'\right)^{-1}. \tag{7}$$

It can be inferred from (5) and (7) that the error covariance of the estimated state is influenced by the error covariance R of the sensor node. In [16], we studied the problem of sensor management to enhance the tracking quality, and showed that the tracking quality depends on the sequence of the activated sensor nodes where the error covariances of the sensor nodes are different. However, the tracking quality does not depend on the sequence of activated sensor nodes if the error covariances of the sensor nodes are the same and the measurements are available at each time step.

Figure 2. Variation of the RMSE of the estimated state with and without measurement loss.

In state estimation, measurements may be absent for many reasons, such as that the target is not covered by any sensor nodes or the sensed measurement is lost. Considering the limited transmission range of each sensor node, it is possible that measurements may not be collected by the mobile sink node when it is outside the transmission range of the activated sensor node. Let E_k and $\mathrm{E}_k^{\mathrm{lost}}$ denote the RMSE (root mean square error) of the estimated state at time step k without and with measurement loss, respectively. Thus E_k is the square root of the estimated error for availability of continuous measurements

and E_k^{lost} is the square root of estimation error for availability of noncontinuous measurements. We can calculate E_k and E_k^{lost} as follows:
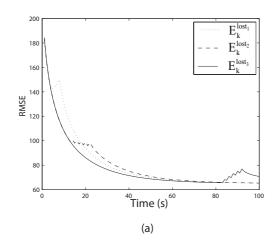
$$E_k = \sqrt{P_{k|k}} \tag{8}$$

$$E_k^{lost} = \begin{cases} \sqrt{P_{k|k}} & \text{if} \quad \text{measurement is received} \\ \sqrt{P_{k|k-1}} & \text{if} \quad \text{measurement is lost.} \end{cases} \tag{9}$$
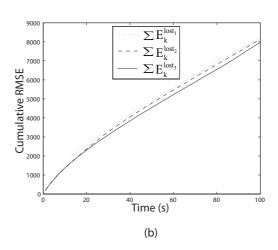
For observable $[F, H]$ and controllable $[F, Q^{\frac{1}{2}}]$ the RMSE of the estimated state E_k asymptotically reaches a steady state if there is no measurement loss. The variation of E_k and E_k^{lost} with time are illustrated in Figure 2, and it can be seen that E_k reaches to a steady state error, but not E_k^{lost}. Here, we used $\Delta t = 1s$, $q = 10$, observation matrix as 4×4 identity matrix and the error covariance of the sensor nodes as:

$$R = \text{diag} \begin{bmatrix} 2.25 \times 10^3 \\ 100 \\ 1.5 \times 10^5 \\ 100 \end{bmatrix}$$

Since $P_{k|k-1} \geq P_{k|k}$, we can say that $E_k^{lost} \geq E_k$ at any time step k as it can be seen in Figure 2. The additional $E_k^{lost} - E_k$ error in estimation is due to the measurement loss. Moreover, we consider three different cases where the measurements are lost at different time steps and the number of measurement losses are not equal. The measurement losses occurred at $k = 5, 6, 7, 8$, $k = 14, 16, 18, 20, 22$ and $k = 84, 86, 88, 90, 92$ for case 1, case 2 and case 3 respectively. We denote the RMSE of the estimated error as $E_k^{lost_1}$, $E_k^{lost_2}$ and $E_k^{lost_3}$ for case 1, case 2 and case 3 respectively.

Figure 3. (a) Variation of the RMSE of the estimated state with measurement losses happening at different times. (b) Variation of cumulative RMSE of the estimated state.

(a)   (b)

The variation in RMSE and cumulative RMSE are illustrated in Figure 3. Even though the number of measurement losses are equal for case 2 and case 3, the cumulative RMSE are not the same. Furthermore, the number of measurement losses for case 1 is less than that of case 2 and case 3, but produces a higher cumulative RMSE than case 2 and case 3. This indicates that the time steps of measurement losses are more important than the number of measurement losses for target tracking. Moreover, it can be

seen in Figures 2 and 3 that measurement losses creates unnecessary error in estimation resulting in degraded tracking quality. Therefore, we make sure that the measurements are not lost at any time step to avoid the additional error in estimation. We increase the transmission range of the sensor nodes to avoid measurement loss only if necessary as this has to be done with minimal energy consumption of the sensor nodes. The following section describes the sensor management strategies according to the mobile sink node positions.

3. Sensor Management with Mobile Sink Node

Let $L_i = (l_{x_i}, l_{y_i}, l_{z_i})$ and $M_k = (m_{x_k}, m_{y_k}, m_{z_k})$ denote the position of the sensor node S_i and the mobile sink node in 3-dimensional Cartesian space at time step k respectively. The assumptions and constraints made in this study are as follows:

1. The position of the target belongs to a known region Ω_k at time step k and the sensor set $\{S_1, S_2, ..., S_{N_k}\}$ which covers the region Ω_k is known a priori

2. Only a single sensor node can be *active* mode at any time step.

3. A sensor node can not be activated during the *active* period t_i of the previously activated sensor node S_i.

4. The transmission range of the activated sensor node S_i is set at $r_i^{\max}$ only if $\|L_i - M_k\| > r_i$ otherwise it remains at default range r_i.

5. $r_i^{\max}$ is selected in such a way that the mobile sink node is always within the range. Thus, $\|L_i - M_k\| < r_i^{\max} \,\forall\, S_i$ and $\forall\, k$.

Our objective is to minimize the total energy consumed by the sensor nodes in order to send the measurements to the mobile sink node without loss. We define the cost function J_T for the entire time horizon $\{1, 2, .., T\}$ as follows:

$$J_T = \sum_{k=1}^{T} \left\{ \delta_{u_k}^{M_k}(\psi_{u_k}^{\max} - \psi_{u_k}) + \psi_{u_k} \right\} \tag{10}$$

where u_k denotes the sensor node in *active* mode at time step k and $u_k \in \{S_1, S_2, ..., S_{N_k}\}$. The measurement-loss function with the default transmission range is denoted as $\delta_{u_k}^{M_k}$ and given by:

$$\delta_{u_k}^{M_k} = \begin{cases} 0 & \text{if } \|L_{u_k} - M_k\| \le r_{u_k} \\ 1 & \text{if } \|L_{u_k} - M_k\| > r_{u_k}. \end{cases}$$

This problem would be easily solved if the mobile sink node were always reachable by a single sensor node with a lower transmission range than the rest of the sensor nodes. However, it may be impossible to find such a sensor node in practical terms. Our aim is to find the optimal sequence of activated sensor nodes $\{u_1^*, u_1^*, ...u_T^*\}$ to minimize the total accumulated cost J_T from time 1 to T:

$$\{u_1^*, u_1^*, ...u_T^*\} = \arg\min_{\forall u_k} \sum_{k=1}^{T} \left\{ \delta_{u_k}^{M_k}(\psi_{u_k}^{\max} - \psi_{u_k}) + \psi_{u_k} \right\}. \tag{11}$$

Since J_T is a function of the relative position of the mobile sink node, we solve this optimization problem for cases when the mobile sink node's position is known both fully and partially.

3.1. *Position of the mobile sink node is known*

Let $\{M_1, M_2, ..., M_\text{T}\}$ denote the sequence of the position of the mobile sink node. We assume that it is known a priori. The base station (BS) collects information about the sensor nodes, such as location and the energy availability. We solve this problem using a dynamic programming technique and rollout algorithm as follows.

a. Dynamic Programming Approach

We used deterministic backward dynamic programming (DP) to find the optimal sequence of the activated sensor nodes. DP is a recursive technique which divides the problem into a sequence of sub-problems using principle of optimality. In this paper, *state* refers the position of the mobile sink node, *stage* refers the time step and *decision* refers the activated sensor node. Since the *state* M_k is known at the kth *stage*, this problem has only a single *state* at each *stage*. The *decision* at time step k when the position of the mobile sink node is at M_k is denoted as $\mu_k(M_k)$ and the number of *decision*s are equal to the number of available sensor nodes.

If the *decision* at time step k is made to activate $\mu_k(M_k) = u_k$ (i.e, to activate the sensor node u_k at time step k), then the period cost (please note that it is not the instantaneous cost but the period cost because once the sensor node has been activated it remains in *active* mode for t_i time steps) from k to $k + t_i - 1$ is given by:

$$g_k(u_k) \;=\; \sum_{t=k}^{\min(k+t_{u_k}-1,\text{T})} \left\{ \delta_{u_k}^{M_t}(\psi_{u_k}^{\max} - \psi_{u_k}) + \psi_{u_k} \right\}. \tag{12}$$

We can rewrite our objective function J_T as:

$$J_\text{T} \;=\; \sum g_k(\mu_k(M_k)) \quad k = 1 : t_{\mu_k(M_k)} : \text{T}. \tag{13}$$

The DP proceeds backwards from *stage* T to *stage* 1 and the "cost-to-go" functions are defined as follows:

$$J_\text{T}(M_\text{T}) \;=\; \min_{\forall \mu_\text{T}(M_\text{T})} \left\{ \delta_{\mu_\text{T}(M_\text{T})}^{M_\text{T}}(\psi_{\mu_\text{T}(M_\text{T})}^{\max} - \psi_{\mu_\text{T}(M_\text{T})}) + \psi_{\mu_\text{T}(M_\text{T})} \right\} \quad \text{at} \quad stage \text{ T} \tag{14}$$

$$J_k(M_k) \;=\; \min_{\forall \mu_k(M_k)} \left\{ g_k(\mu_k(M_k)) + J_{k+\text{K}}(M_{k+\text{K}}) \right\} \quad \text{at} \quad stage \text{ T}-1, ...2, 1 \tag{15}$$

where

$$\text{K} \;=\; t_{\mu_k(M_k)}.$$

DP solves the sub-problems from $k = \text{T}$ to 1 and stores the optimal value for $J_k(\text{M}_k)$ and $\mu_k^*(\text{M}_k)$. Once all the sub-problems have been solved, DP backtracks the optimal sensor nodes from the stored values. The optimal total cost is given by:

$$\min J_\text{T} \;=\; J_1(\text{M}_1). \tag{16}$$

The optimal sensor node to be activated at time step 1 and k are given by:

$$u_1^* \;=\; \arg\{J_1(\text{M}_1)\} \tag{17}$$

$$u_k^* \;=\; \begin{cases} \text{S}_j & \text{if} \quad \text{S}_j \text{ sensor node is in } active \text{ mode} \\ \arg\{J_k(\text{M}_k)\} & \text{if} \quad \text{otherwise} \end{cases} \tag{18}$$

where S_j denotes the previously activated optimal sensor node. For this problem, the computational cost of DP only depends on the number of *stages* (total time horizon) and the number of admissible *decisions* (available feasible sensor nodes). Since the DP calculations are done off-line, it is assumed that at each time step set of available feasible sensor nodes are known a priori. In real situations, the available feasible number of sensor nodes can vary with time, for example sensor nodes may be deleted due to dead battery or malfunction. In this study, we assume that the set of available feasible sensor nodes at each time step are known and the sensor nodes work without malfunction.

b. Rollout Approach

The rollout algorithm (RA) is an approximate dynamic programming technique to overcome the curse of dimensionality in DP. The algorithm performs according to the approximate cost-to-go function given by a sub-optimal base heuristic policy. It produces a solution not worse than the solution given by the base heuristic. The computational cost of the RA depends on the problem size and the base heuristic method. In general, RA is computationally more tractable than DP. Since M_k has a single value at each time step, known in our problem, the number of *states* at each *stage* is one and therefore, the computational cost of DP is not an issue. However, DP may not be used if the *decision* variables are varying with time. RA updates *stages*, *states* and *decisions* at each time step and performs the optimization. The major steps involved in RA for our problem are as follows:

Step 1: At $k = 1$, update $\{\text{S}_1, \text{S}_2, ..., \text{S}_{\text{N}_1}\}$ and calculate $g_1(\text{S}_i) + J_{1+t_i}(\text{M}_{1+t_i})$, $\forall\ \text{S}_i \in \{\text{S}_1, \text{S}_2, ..., \text{S}_{\text{N}_1}\}$. Here, $J_{1+t_i}(\text{M}_{1+t_i})$ is solved by the base heuristic method.

Step 2: Choose the sensor node $\tilde{u}_1^*$ which minimizes $g_1(i) + J_{1+t_i}(\text{M}_{1+t_i})$.

Step 3: At time k, update $\{\text{S}_1, \text{S}_2, ..., \text{S}_{\text{N}_k}\}$.

Step 4: If a sensor node S_j is in *active* mode set $\tilde{u}_k^* = \text{S}_j$ and go to Step 3 with replace k by $k + 1$, else go to Step 5.

Step 5: Calculate $g_k(\mathrm{s}_i) + J_{k+t_i}(\mathrm{M}_{k+t_i})$ for all $\forall\, \mathrm{s}_i \in \{1, 2, .., \mathrm{N}_k\}$ and $J_{k+t_i}(\mathrm{M}_{k+t_i})$ is solved by the base heuristic method.

Step 6: choose the sensor node $\tilde{u}_k^*$ which minimizes $g_k(\mathrm{s}_i) + J_{k+t_i}(\mathrm{M}_{k+t_i})$.

Step 7: If $k > \mathrm{T}$ go to Step 8 , else go to Step 3, replacing k with $k + 1$.

Step 8: The sub-optimal sensor node sequence is given by $\{\tilde{u}_1^*, \tilde{u}_2^*, ..., \tilde{u}_\mathrm{T}^*\}$.

We use the one-step-look ahead (OSLA) [32] method as our base heuristic method for RA. Since OSLA algorithm optimizes only the cost occurred at the current time step, the results are sub-optimal and the computational cost required is comparatively very small. The OSLA algorithm produces the sub-optimal sensor sequence $\{\breve{u}_1^*, \breve{u}_2^*, ..., \breve{u}_\mathrm{T}^*\}$ and given by:

$$\breve{u}_1^* = \arg\min_{\forall \mu_1(\mathrm{M}_1)} \left\{ \delta_{\mu_1(\mathrm{M}_1)}^{\mathrm{M}_1} (\psi_{\mu_1(\mathrm{M}_1)}^{\max} - \psi_{\mu_1(\mathrm{M}_1)}) + \psi_{\mu_1(\mathrm{M}_1)} \right\} \tag{19}$$

$$\breve{u}_k^* = \begin{cases} \mathrm{s}_j & \text{if} \quad \mathrm{s}_j \text{ sensor node is in } \textit{active} \text{ mode} \\ \arg\min\limits_{\forall \mu_k(\mathrm{M}_k)} \left\{ \delta_{\mu_k(\mathrm{M}_k)}^{\mathrm{M}_k} (\psi_{\mu_k(\mathrm{M}_k)}^{\max} - \psi_{\mu_k(\mathrm{M}_k)}) + \psi_{\mu_k(\mathrm{M}_k)} \right\} & \text{if} \quad \text{otherwise} \end{cases} \tag{20}$$

where s_j denotes the previously activated optimal sensor node.

3.2. Position of the mobile sink node is partially known

In this section, we assume that the exact position of the mobile sink node M_k is known only at time step k otherwise it is only partially known. In other words, M_k is known $\forall k \leq c$ and partially known $\forall k > c$ at current time step c. We assume that the BS immediately gets to know the current position of the mobile sink node at time step k.

The position of the mobile sink node varies with time according to a known Markov chain. Assume that M_k is an S-state Markov chain with the state space $\{e_1, ..., e_S\}$. Here, e_s denotes a position in 3-dimensional Cartesian space. The transition probability matrix A and the initial probability vector π_1 of the mobile sink node's position are defined as:

$$\mathrm{A} = [a_{mn}]_{S \times S} \quad \text{where} \quad a_{mn} = P(\mathrm{M}_k = e_n | \mathrm{M}_{k-1} = e_m), \quad m, n \in \{1, ..., S\}.$$
$$\pi_1 = [\pi_1(m)]_{S \times 1} \quad \text{where} \quad \pi_1(m) = P(\mathrm{M}_1 = e_m), \quad m \in \{1, ..., S\}.$$

The Markov chain parameters A and π_1 are assumed to be known.
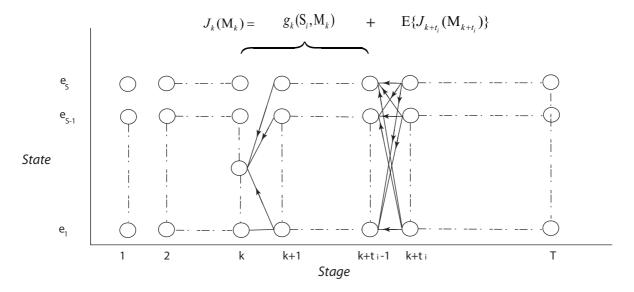
a. Stochastic Dynamic Programming Approach

Since the exact future position of the mobile sink node is unknown at the current time step, we cannot use the deterministic dynamic programming for this problem. We present the stochastic dynamic

programming (SDP) technique to produce the optimal sequence of activated sensor nodes when the mobile sink node' future position is uncertain. Let $g_k(S_i, M_k)$ denotes the expected period cost from k to $k + t_i - 1$ when the position of the mobile sink node is in M_k and the activated sensor node at kth time step is S_i. The period cost $g_k(S_i, M_k)$ is given by:

$$g_k(S_i, M_k) = \sum_{\substack{t=k}}^{\min(k+t_i-1, T)} \mathbb{E}_{t \neq k} \left\{ \delta_i^{M_t}(\psi_i^{\max} - \psi_i) + \psi_i \right\}. \tag{21}$$

The future position of mobile sink node M_t is unknown $\forall t > k$ at time step k and therefore, we take the expectation to calculate the period cost $g_k(S_i, M_k)$.

Figure 4. *state-stage* diagram of SDP with S number of *states* and T number of *stages*.

For this problem M_t can take any state in S-state Markov chain and therefore, SDP has S number of *states* at each *stage*. Figure 4 shows the $state - stage$ diagram of the SDP for this sensor management problem. The SDP proceeds backwards from *stage* T to *stage* 1 and the "cost-to-go" functions are defined as follows:

$$J_T(M_T) = \min_{\forall \mu_T(M_T)} \left\{ \delta_{\mu_T(M_T)}^{M_T}(\psi_{\mu_T(M_T)}^{\max} - \psi_{\mu_T(M_T)}) + \psi_{\mu_T(M_T)} \right\} \quad at \quad stage \; T \tag{22}$$

$$J_k(M_k) = \min_{\substack{\forall \mu_k(M_k) \\ \forall M_{k+K}}} \left\{ g_k(\mu_k(M_k), M_k) + \mathbb{E} \left\{ J_{k+K}(M_{k+K}) \right\} \right\} \quad at \quad stage \; T-1, ...2, 1 \tag{23}$$

where

$$K = t_{\mu_k(M_k)}.$$

SDP solves the sub-problems from $k = T$ to 1 and stores the values for $J_k(M_k)$ and $\mu_k^*(M_k)$. The optimal expected total cost J_T is given by:

$$\min \mathbb{E} \{ J_T \} = \sum_{a=1}^{S} \pi_1(e_a) J_1(e_a). \tag{24}$$

The optimal sensor node to be activated at time steps 1 and k depend on the position of the mobile sink node and is given by:

$$\mu_1^*(M_1) \;\;=\;\; \arg\left\{J_1(M_1)\right\} \tag{25}$$

$$\mu_k^*(M_k) \;\;=\;\; \begin{cases} j & \text{if} \quad j\,\text{th sensor node is in } active \text{ mode} \\ \arg\left\{J_k(M_k)\right\} & \text{if} \quad \text{otherwise} \end{cases} \tag{26}$$

where j denotes the previously activated optimal sensor node. These calculations are done at the BS off-line. At each time step, the BS updates the current position of the mobile sink node and activates the optimal sensor node using the look up table, where the SDP stores $J_k(M_k)$ and $\mu_k^*(M_k)$. The computational cost of SDP for this problem depends on the number of *stages*, number of *states* and number of admissible *decisions*. Furthermore, we assume that the number of available sensor nodes (admissible *decisions*) at each time step are known a priori and the sensor nodes work without malfunction.

b. One-step-look-ahead method

For a large problem (where the number of *states*, *stages* and *decisions* are larger), SDP may not solve the problem with-in an acceptable time period, due to its huge computational cost. Consider a scenario where the parameters required for the SDP are known just before the application (target started moving) and the problem is large enough that we may not have enough time to solve it before the application. Moreover, if the set of available sensor nodes varies with time, SDP cannot be used. We use the one-step-look-ahead (OSLA) algorithm as a sub-optimal method to solve this scheduling problem. The OSLA finds the best sensor node which minimizes only the energy consumption at the current time step for a known M_k since M_k is known at the current time step k. For $k = 1, 2, ...T$, the OSLA produces the sub-optimal sensor sequence as $\{\tilde{u}_1^*, \tilde{u}_2^*, ..., \tilde{u}_T^*\}$ where the sub-optimal sensor node at time step 1 and k are given by:

$$\tilde{u}_1^* \;\;=\;\; \arg\min_{\forall u_1}\left\{\delta_{u_1}^{M_1}(\psi_{u_1}^{\max} - \psi_{u_1}) + \psi_{u_1}\right\} \tag{27}$$

$$\tilde{u}_k^* \;\;=\;\; \begin{cases} j & \text{if} \quad j\,\text{th sensor node is in } active \text{ mode} \\ \arg\min_{\forall u_k}\left\{\delta_{u_k}^{M_k}(\psi_{u_k}^{\max} - \psi_{u_k}) + \psi_{u_k}\right\} & \text{if} \quad \text{otherwise} \end{cases} \tag{28}$$

Although OSLA algorithm produces a sub-optimal solution for the sensor management problem, it is suitable when the availability of the feasible sensor nodes are unknown a priori or when we need to solve the problem within a short time period.

4. Results and Discussions

In this section, we present a numerical example of single target tracking with noisy sensor nodes and the BS is located far away from the sensor field. A mobile sink node flies around the sensor field, collects the measurements from the sensor nodes and transmits them to the BS in order to prolong the life span of the sensor network. We simulate a single target moving in a 2-dimensional Cartesian space according to (3). The system matrix F and the system noise covariance Q are the same as in section 2.2.
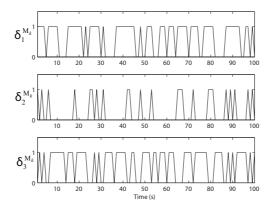
4.1. Parameter Settings for the Sensor Network

In our simulation, we use only 3 sensor nodes to enhance the clarity and simplicity of the problem. Sensor nodes S_1, S_2 and S_3 are deployed in a sensor field of 500 m $\times$ 500 m and for the experimental purpose, we assume that each sensor node covers the target such that the feasible sensor set has S_1, S_2 and S_3 sensor nodes at each time step. A mobile sink node collects the measurements and immediately transmits to the BS. The properties of the sensor nodes are given in Table 1.

Table 1. Properties of the sensor nodes.

Sensor Node (S_i)	Coordinate (L_i)	Transmission Range (r_i)	Active time period (t_i)
S_1	$(150, 150, 0)$m	200m	5s
S_2	$(350, 200, 0)$m	300m	4s
S_3	$(250, 400, 0)$m	200m	2s

It is assumed that the measurements are sent to the mobile sink node in a single-hop communication without any time delay. The observation matrix H is a unit matrix, and therefore the measurements are linearly related to the state of the target. The time interval between the measurements is considered as $\Delta t = 1$s. We assumed the measurement data size as 1 MB and the data rate as 8 Mbps. Without loss of generality, we neglect the energy consumption ψ_1 and ψ_2 and only consider ψ_{tx}. The parameters in ψ_{tx} are set $\alpha_1 = 50$ nJ/b and $\alpha_2 = 100$ pJ/bm^2. The maximum transmission range of $r_i^{\max} = 500$m was chosen for all sensor nodes. The target starts at X_0 with the known initial error covariance P_o and moves for T s.

Figure 5. Measurement-loss functions with default transmission ranges of the sensor nodes with time.
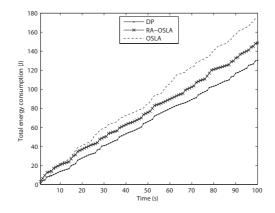
4.2.   Sensor management with known mobile sink node's path

In this section, we assume that the position of the mobile sink node is known at each time step and the mobile sink node flies horizontally at a height of 100 m above the ground. We plot the measurement-loss function $\delta_i^{M_k}$ with the r_i for the total time horizon of 100s, shown in Figure 5.

Table 2. Comparison of cumulative cost J_T obtained by OSLA, RA-OSLA and DP for different total time horizons.

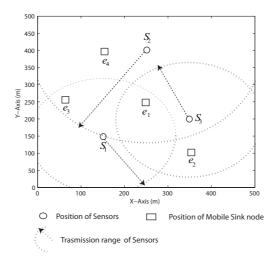| T (s)	OSLA (J)	RA-OSLA (J)	DP (J)
30	59.0	48.8	40.8
40	72.4	62.4	49.6
50	84.0	74.4	63.2
60	105.6	89.6	76.8
70	123.2	101.6	89.2
80	141.6	118.4	103.2
90	157.2	130.0	114.8
100	175.2	148.8	130.0

Figure 6. Variation of total energy consumption with time obtained by OSLA, RA-OSLA and DP.

We compare the sub-optimal solutions obtained by the rollout algorithm (RA) and one-step-look-ahead (OSLA) method with the optimal solution obtained by dynamic programming (DP). We use OSLA method as the base heuristic method for RA. We increase the total time horizon from T = 30s to 100s. The results for all three methods are shown in Table 2. The variation of total energy consumption with time is shown in Figure 6 for T = 100s. We can see from the results in Table 2 that RA always produces better results than the base heuristic method (OSLA). Since the positions of the mobile sink node are

known a priori, the computational cost of DP is not large and therefore DP is best suited for this particular problem.

Figure 7. Probable positions of the mobile sink and the default transmission ranges and positions of the sensor nodes.

4.3. Sensor management with partially known mobile sink node's path

In this section, we assume that the position of the mobile sink node varies according to a known Markov chain and that the exact position of the mobile sink node is known only at the current time step but not before. We have chosen a 4-State Markov chain with the state space $\{e_1, e_2, e_3, e_4\}$ for the position of the mobile sink node. Here $e_1 = (250, 250, 100)$m, $e_2 = (350, 100, 100)$m, $e_3 = (75, 250, 100)$m and $e_4 = (150, 400, 100)$m. Figure 7 shows the probable positions of the mobile sink node and of the sensor nodes, and the default transmission ranges of the sensor nodes

The initial probability vector of the mobile sink node's position is assumed as $\pi_1 = [0.2 \; 0.4 \; 0.2 \; 0.2]$. We choose two different transition probabilities to analyze the performance of our methods as follows:

$$
A_1 = \begin{bmatrix} 0.2 & 0.3 & 0.2 & 0.3 \\ 0.3 & 0.3 & 0.2 & 0.2 \\ 0.2 & 0.3 & 0.2 & 0.3 \\ 0.3 & 0.1 & 0.2 & 0.4 \end{bmatrix}, \qquad A_2 = \begin{bmatrix} 0.1 & 0.5 & 0.1 & 0.3 \\ 0.1 & 0.3 & 0.1 & 0.5 \\ 0.3 & 0.1 & 0.5 & 0.1 \\ 0.1 & 0.3 & 0.1 & 0.5 \end{bmatrix}
$$

The optimal expected total energy consumption of the sensor nodes with A_1 and A_2 for $T = 100$s are calculated off-line using stochastic dynamic programming (SDP), and are given by:

$$
\mathbb{E}\{J_{100}^{A_1}\} = \sum_{a=1}^{4} \pi_1(a) J_1(e_a) = 142.54 \text{ J}, \quad \mathbb{E}\{J_{100}^{A_2}\} = \sum_{a=1}^{4} \pi_1(a) J_1(e_a) = 154.45 \text{J}
$$

Since the future positions of the mobile sink node are uncertain, the OSLA schedules the sensor nodes on-line. Therefore, we simulate the mobile sink node's path according to the known Markov parameters
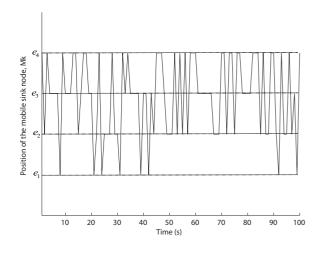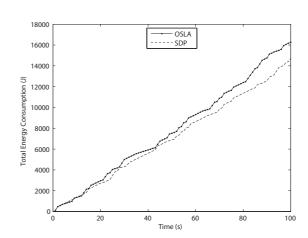
to obtain the results for OSLA.

Table 3. Total average energy consumption obtained by OSLA and SDP with different transition matrices.

T (s)	J_{T} with A_1		J_{T} with A_2	
	OSLA (J)	SDP (J)	OSLA (J)	SDP (J)
30	47.80±5.48	42.50±5.01	51.33±6.26	46.12±5.16
40	64.12±6.53	56.61±5.77	67.86±7.43	61.50±5.92
50	80.24±7.36	70.70±6.41	85.19±8.56	77.10±6.75
60	96.45±7.87	85.14±7.05	101.79±8.80	92.67±7.27
70	112.55±8.89	99.46±7.63	118.97±9.27	107.80±8.04
80	128.34±8.93	113.26±7.98	136.77±9.99	123.38±8.48
90	144.87±9.64	127.84±8.48	153.43±10.91	138.69±9.01
100	160.90±10.15	143.27±9.05	170.82±11.35	155.74±9.65

The results in Table 3 were averaged over 100 independently simulated paths for the mobile sink node using the known Markov parameters. The mean and standard deviation of the total energy consumption are shown in Table 3. The results in Table 3 obtained from SDP become equal to $\mathbb{E}\{J_{\mathrm{T}}\}$ for very many simulated paths of the mobile sink node. For example, we can see that $J_{100} = 141.27$ with A_1 whereas $\mathbb{E}\{J_{\mathrm{T}}^{A_1}\} = 142.54$. Figure 8 shows a simulated path for the mobile sink node using the known Markov parameters and the total energy consumption obtained by OSLA and SDP.

Figure 8. (a) Simulated position of the mobile sink node with time. (b) Total energy consumption obtained by OSLA and SDP for $\mathrm{T} = 100$ s.
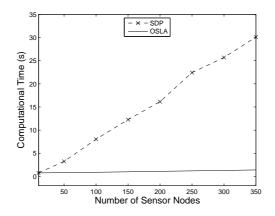

Since OSLA is a sub-optimal method, on average it consumes more energy than SDP. Even though the results are not promising, the computational cost of the OSLA is very low compared to that of SDP.

The number of probable positions (e_a) of the mobile sink node does not affect the computational cost of the OSLA whereas the computational cost of the SDP increases. In order to analyze the effect of the number of sensor nodes on the computational cost of the algorithms, we increase the sensor nodes from 3 to 350 whilst keeping a constant number of probable positions of the mobile sink node. It can be seen from Figure 9 that the computational cost of SDP increases as the number of sensor nodes increases, whereas no significant variation for OSLA was observed.

Figure 9. Variation of the computational cost required by the algorithms when the number of sensor nodes increases with 4-State Markov chain.

5. Conclusion

In this paper, we studied optimal and sub-optimal methods for scheduling the sensor nodes with a mobile sink node to minimize the total energy consumption for target tracking. An activated sensor node increases its transmission range to maximum only if the measurement could not reach the mobile sink node. Moreover, a sensor node cannot be activated while a previously activated sensor node remains active. We studied the sensor management problem when the path of the mobile sink node is known both fully and partially. We solved the problem optimally using deterministic and stochastic dynamic programming techniques and sub-optimally using rollout and one-step-look-ahead algorithms. The deterministic dynamic programming is best suited for the problem where the path of the mobile sink node is fully known. The limitation of dynamic programming technique is that it cannot be used to solve the problem when the available feasible number of sensor nodes are unknown a priori or varying with time. In such cases, rollout and one-step-look-ahead algorithms are useful. Furthermore, for a big problem, stochastic dynamic programming technique does not solve the problem within a feasible time period due to its high computational cost.

6. Notes Added in Proof

1. In page 5, last paragraph:
   Original version
   where w_k is the white Gaussian process noise with a known covariance Q and the intensity of the

process noise is determined by the matrix B.

Corrected version

where w_k is the white Gaussian process noise with a known covariance Q and the intensity of the process noise is determined by the scalar quantity q.

2. In page 15, Figure 5: y-axis texts are enlarged to make it more clearer.

References and Notes

1. Mahfoudh, S.; Minet, P. Survey of Energy Efficient Strategies in Wireless Ad Hoc and Sensor Networks. In *Seventh International Conference on Networking ICN 2008*, IEEE Computer Society: Washington, DC, USA, 2008

2. Tang, X.; Xu, J. Extending Network Lifetime for Precision-Constrained Data Aggregation in Wireless Sensor Networks. In *25th IEEE International Conference on Computer Communications IN-FOCOM 2006*, Barcelona, Spain, April 23-29, 2006; pp. 1-12.

3. Chiasserini, C.-F.; Garetto, M. Modeling the performance of wireless sensor networks. In *Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies INFOCOM 2004*, Hong Kong, China, November 22, 2004.

4. Akyildiz, I.F.; Weilian Su; Sankarasubramaniam, Y.; Cayirci, E. A survey on sensor networks. *IEEE Commun. Mag.* **2002**, *40*, 102-114.

5. Lee, D.; Kaliappan, V.K.; Chung, D.; Min, D. An energy efficient dynamic routing scheme for clustered sensor network using a ubiquitous robot. In *IEEE International Conference on Research, Innovation and Vision for the Future*, Ho Chi Minh, July 13-17, 2008

6. Ingelrest, F.; Simplot-Ryl, D.; Stojmenovic, I. Optimal transmission radius for energy efficient broadcasting protocols in ad hoc and sensor networks. *IEEE T. Parall. Distr.* **2006**, *17*, 536-547.

7. Supriyo, C.; Tim, N.; Nirvana, M.; Paul, H. A distributed and self-organizing scheduling algorithm for energy-efficient data aggregation in wireless sensor networks. *ACM Trans. Sens. Netw.* **2008**, *4*,1-48.

8. Bugra, G.; Ling, L.; Philip, S.Y. ASAP: An Adaptive Sampling Approach to Data Collection in Sensor Networks. *IEEE T. Parall. Distr.* **2007**, *18*, 1766-1783.

9. Heinzelman, W.R.; Chandrakasan, A.; Balakrishnan, H. Energy-efficient communication protocol for wireless microsensor networks. In *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, ACM: New York, NY, USA, 2000

10. Tirta, Y.; Li, Zhiyuan; Lu, Y.; Bagchi, S. Efficient collection of sensor data in remote fields using mobile collectors. In *13th International Conference on Computer Communications and Networks ICCCN*, ACM: New York, NY, USA, 2004; pp. 515-519.

11. Wu, Y.; Zhang, L.; Wu, Y.; Niu, Z. Interest dissemination with directional antennas for wireless sensor networks with mobile sinks. In *Proceedings of the 4th international conference on Embedded networked sensor systems*, New York, NY, USA, 2006; pp. 99-111.

12. Qin, Y.; Sun, D.; Li, N.. Cen, Y. Path planning for mobile robot using the particle swarm optimization with mutation operator. In em Proceedings of 2004 International Conference on Machine

Learning and Cybernetics, Shanghai, China, August 26-29, 2004; pp. 2473-2478.

13. Verma, A.; Sawant, H.; Tan, J. Selection and Navigation of Mobile Sensor Nodes Using a Sensor Network. In *Third IEEE International Conference on Pervasive Computing and Communications*, Kauai Island, HI, USA, March 8-12, 2005; pp.41-50.

14. Xiao, W.; Xie, L.; Lin, J.; Li, J. Multi-Sensor Scheduling for Reliable Target Tracking in Wireless Sensor Networks. In *6th International Conference on ITS Telecommunications Proceedings*, Chengdu, China, June 2006; pp.41-50.

15. Kreucher, C.; Hero, A.; Kastella, K.; Shapo, B. Information-based Sensor Management for Simultaneous Multitarget Tracking and Identification. In *the Proceedings of the Thirteenth Annual Conference on Adaptive Sensor Array Processing (ASAP)*, Lexington, MA, June 7-8, 2005.

16. Maheswararajah, S.; Halgamuge, S.; Premaratne, M. Sensor Scheduling For Target Tracking by Sub-optimal Algorithms. *IEEE T. Veh. Technol.* **2009** Accepted for publication.

17. Williams, J.L.; Fisher, J.W.; Willsky, A.S. Approximate Dynamic Programming for Communication-Constrained Sensor Network Management. *IEEE T. Signal Proces.* **2007**, *55*, 4300-4311.

18. Xue W.; Jun-Jie M.; Sheng W.; Dao-Wei B. Cluster-based Dynamic Energy Management for Collaborative Target Tracking in Wireless Sensor Networks. *Sensors* **2007**, *7*, 1193-1215.

19. Hernandez, M.L.; Kirubarajan, T.; Bar-Shalom, Y. Multisensor resource deployment using posterior Cramer-Rao bounds. *IEEE T. Aero. Elec. Sys.* **2004**, *40*, 399-416.

20. Chhetri, A.S.; Morrell, D.; Papandreou-Suppappola, A. Energy efficient target tracking in a sensor network using non-myopic sensor scheduling. In *8th International Conference on Information Fusion*, Philadelphia, USA, July 25-28, 2005; p. 8.

21. Shah, H.; Morrell, D. Non-myopic sensor scheduling for a distributed sensor network. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, Las Vegas, March 30-April 4, 2008; pp. 2541-2544.

22. Gupta, V.; Chung, T.; Hassibi, B.; Murray, R.M. On a Stochastic Sensor Selection Algorithm with Applications in Sensor Scheduling and Sensor Coverage. *Automatica* **2006** , *42*, 251-260.

23. Tharmarasa, R.; Kirubarajan, T.; Hernandez, M.L.; Sinha, A. PCRLB-based multisensor array management for multitarget tracking. *IEEE T. Aero. Elec. Sys.* **2007**, *43*, 539-55.

24. John, S.B.; Alain, B. Sensor scheduling problems. In *Proceedings of 27 th IEEE International Conference on Decision and Control*, Austin, Texas, December 1998; pp. 2342-2345.

25. Bhardwaj, M.; Chandrakasan, A.P. Bounding the lifetime of sensor networks via optimal role assignments. In *Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies*, New York, USA, June 23-27, 2002; pp.1587-1596.

26. Gupta, V.; Chung, T.; Hassibi, B.; Murray, R.M. Sensor scheduling algorithms requiring limited computation [vehicle sonar range-finder example]. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Montreal, CA, May 17-21, 2004; pp. iii-825-8.

27. Tichavsky, P.; Muravchik, C.H.; Nehorai, A. Posterior Cramer-Rao bounds for discrete-time nonlinear filtering. *IEEE T. Signal Proces.* **1998**, *46*, 1386-1396.

28. Krishnamurthy, V. Algorithms for optimal scheduling and management of hidden Markov model sensors. *IEEE T. Signal Proces.* **2002**, *50*, 1382-1397

29. Evans, J.; Krishnamurthy, V. Optimal sensor scheduling for Hidden Markov models. In *Proceedings of the 1998 IEEE International Conference on Acoustics*, Speech, and Signal Processing, Seattle, WA, USA, 1998; pp.2161-2164

30. Yaakov, B.; Thiagalingam, K.; Rong, L. *Estimation with Applications to Tracking and Navigation*. John Wiley & Sons, Inc.: New York, NY, USA, 2002.

31. Kalman, R.E. A New Approach to Linear Filtering and Prediction Problems. *Trans. ASME, J. Basic Eng.* **1960**, 35-45.

32. Bertsekas, D.P. *Dynamic Programming and Optimal Control,* third edition, Volume 3, Athena Scientific, 2005.