Dynamic Bit Allocation for Object Tracking in Wireless Sensor Networks

Engin Masazade, *Member, IEEE*, Ruixin Niu, *Senior Member, IEEE*, and Pramod K. Varshney, *Fellow, IEEE*

***Abstract*—In this paper, we study the target tracking problem in wireless sensor networks (WSNs) using quantized sensor measurements where the total number of bits that can be transmitted from sensors to the fusion center is limited. At each time step of tracking, a total of R available bits need to be distributed among the N sensors in the WSN for the next time step. The optimal solution for the bit allocation problem can be obtained by using a combinatorial search which may become computationally prohibitive for large N and R. Therefore, we develop two new suboptimal bit allocation algorithms which are based on convex optimization and approximate dynamic programming (A-DP). We compare the mean squared error (MSE) and computational complexity performances of convex optimization and A-DP with other existing suboptimal bit allocation schemes based on generalized Breiman, Friedman, Olshen, and Stone (GBFOS) algorithm and greedy search. Simulation results show that, A-DP, convex optimization and GBFOS yield similar MSE performance, which is very close to that based on the optimal exhaustive search approach and they outperform greedy search and nearest neighbor based bit allocation approaches significantly. Computationally, A-DP is more efficient than the bit allocation schemes based on convex optimization and GBFOS, especially for a large sensor network.**

***Index Terms*—Convex optimization, dynamic bit allocation, dynamic programming, posterior Cramér–Rao lower bound, target tracking, wireless sensor networks.**

I. Introduction

A wireless sensor network (WSN) consists of a large number of spatially distributed sensors which are tiny, battery-powered devices, and have limited on-board energies. When properly programmed and networked, WSNs perform different tasks that are useful in a wide range of applications such as battlefield surveillance, environment and health monitoring, and disaster relief operations. In a WSN, dense sensor deployment may introduce redundancy in coverage, so selecting a subset of sensors may still provide information with the desired quality. Adaptive sensor management policies select a subset of active sensors to meet the application requirements in terms of quality of service while minimizing the use of resources.

In the literature, there exist many sensor selection algorithms (see [1] and references therein). In [1], the sensor selection problem, an integer programming problem, has been relaxed and solved through convex optimization. One popular strategy for sensor selection is to use information driven methods [2]–[5] where the main idea is to select the sensors that provide the most useful information, which is quantified by entropy or mutual information. The posterior Cramér-Rao lower bound (PCRLB) is also a very important tool because it provides a theoretical performance limit for a Bayesian estimator [6] and has been recently considered as a metric for adaptive sensor management [7]–[10]. As we have shown in our previous paper [8], for sensor selection, the complexity to compute the mutual information increases exponentially with the number of sensors to be selected, whereas the computational complexity of Fisher information, which is the inverse of the PCRLB, increases linearly with the number of sensors to be selected. For target tracking in a bearing-only sensor network, a sensor selection approach which minimizes the PCRLB on the estimation error has been proposed in [9] and [10], where the selected sensors transmit either analog or quantized data to the fusion center.

In this paper, the task of the WSN is to track a moving object emitting energy in a given region of interest (ROI). In the bit allocation problem, we assume that the total number of bits for the channels between sensors and the fusion center is limited per iteration. Note that, in the sensor selection problem, a sensor is either activated or not to transmit its measurement under the constraint on the total number of selected sensors, while in the bit allocation problem, the fusion center not only needs to make a decision about activation but also needs to distribute the available bits optimally and dynamically among the activated sensors. Therefore, the sensor selection problem is a special case of the bit allocation problem. For a given total number of bits per time step that can be transmitted from sensors to the fusion center, dynamic bit allocation provides better estimation performance as compared to the sensor selection schemes since it distributes the resources more efficiently. We consider a myopic (one-step ahead) scenario, where at a given time step, the fusion center only decides on the bit allocations of the next time step.

In the literature, the bit allocation problem for WSNs has been studied in the context of data gathering [11], [12], distributed sequential detection [13] and target tracking [14]. In [11], the bit allocation problem has been studied in the context of maximizing the amount of information collected by all the nodes in the network over the course of network lifetime. In [12], the authors proposed an energy efficient data gathering scheme where

Manuscript received September 21, 2011; revised February 15, 2012 and May 23, 2012; accepted June 05, 2012. Date of publication June 11, 2012; date of current version September 11, 2012. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Maciej Niedzwiecki. This work was supported by the U.S. Air Force Office of Scientific Research (AFOSR) by Grant FA9550-10-1-0263. Part of this work was presented at the Fusion'11 Conference, Chicago, IL, July 5-8,2011.

E. Masazade and P. K. Varshney are with the Department of Electrical Engineering and Computer Science, Syracuse University, Syracuse, NY 13244 USA (e-mail: emasazad@syr.edu; varshney@syr.edu).

R. Niu is with the Department of Electrical and Computer Engineering, Virginia Commonwealth University, Richmond, VA 23284 USA (e-mail: rniu@vcu.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2012.2204257

the sensor data are gathered sequentially. First, the data rate of the most energy efficient sensor, the sensor which is closest to the sink, is determined by its entropy. Having received the data of the sensors, using Slepian-Wolf coding, the data rates of the remaining sensors are determined sequentially. For distributed sequential detection, the authors in [13] studied the optimal allocation of bits to different sensors if a fixed number of bits are available for the channels between sensors and the fusion center. In our paper, since the task is to localize and track a moving object emitting energy, only a subset of sensors receives signals with significant amplitudes from the object and the subset of informative sensors keeps changing over time. In this scenario, a fixed number of bits for the channels between sensors and the fusion center need to be distributed optimally and dynamically among the sensors in the network. This is an important problem since given a constraint on the total number of bits at each reporting instant, dynamic bit allocation uses the available resources more efficiently while providing better estimation performance as compared to only sensor selection which is a special case of the bit allocation problem. This is in contrast to the data gathering problems solved in [11], [12] which put no constraint on the total number of transmitted bits from sensors to the fusion center.

Given the constraint on total number of bits, R, at each time step during tracking, the fusion center should determine the optimal bit allocations for the channels between N sensors and the fusion center by optimizing the target tracking performance in the WSN. This one-step ahead bit allocation problem can be solved by using an exhaustive search which enumerates all possible bit allocations and decides on the solution that maximizes the determinant of the Fisher information matrix (FIM) which is the inverse of the PCRLB. Under Gaussian assumption, maximizing the determinant of the FIM is equivalent to minimizing the volume of the uncertainty ellipsoid [15]. Since the FIM can be written as the summation of individual sensor FIMs and the Fisher information of the prior, by using a FIM based bit allocation metric, we are able to generalize [1] for the bit allocation problem. However, for metrics other than FIM, such as mutual information, the solution to the sensor selection problem can not be easily generalized for the bit allocation problem [8]. The bit allocation problem is a combinatorial problem where the explicit enumeration of all bit allocation solutions is computationally prohibitive for large values of N and R. Therefore, computationally efficient suboptimal methods are required. The generalized Breiman, Friedman, Olshen, and Stone (GBFOS) algorithm, first proposed in [16] for vector quantization, has been employed for dynamic bit allocation for source localization [17] and target tracking [14]. Results in [14] show that dynamic bit allocation significantly outperforms a static equal bit allocation scheme in terms of tracking performance. But still, as we show later in the paper, the GBFOS algorithm may become computationally costly with increasing values of N.

By generalizing the sensor selection method proposed in [1], we first propose the convex optimization based bit allocation as a benchmark method against which performance of other methods can be compared. We formulate the bit allocation problem as a constrained optimization problem with binary-valued decision variables and equality constraints. Then, we relax it by replacing the Boolean variables with their continuous counterparts and solve this relaxed problem optimally using an interior point method [18], which may still become computationally costly with increasing values of N and R. In our preliminary work [19], we proposed a computationally efficient dynamic programming (DP) based bit allocation scheme. Due to the fact that the total Fisher information is the summation of each sensor's individual Fisher information, for a scalar-valued parameter estimation problem, a DP recursion can be easily formulated to find the optimal bit allocation at each time step by maximizing the Fisher information [20]. For target tracking, even though the Fisher information is in a matrix form and the objective is to maximize the determinant of the FIM, we can still formulate a DP recursion which would yield a suboptimal solution. We refer to this scheme as approximate DP (A-DP).

In this paper, we compare the bit allocation schemes based on convex optimization, A-DP, GBFOS and greedy search in terms of their mean squared error and computational load under different process noise parameters. Simulation results show that convex optimization, A-DP and GBFOS yield similar tracking performance and among these three suboptimal schemes, A-DP has the least computational load, when the sensor network is large. The convex optimization, A-DP and GBFOS algorithms yield similar tracking performance, which is also similar to that provided by the optimal exhaustive search approach, and they outperform the greedy search and nearest neighbor approach significantly.

The rest of the paper is organized as follows. In Section II, we introduce the target tracking problem, and describe the optimization of the quantization thresholds and particle filtering in target tracking. In Section III, we describe the bit allocation schemes based on convex optimization, A-DP, GBFOS and greedy search. In Section IV, we present numerical examples and compare the performances of the considered bit allocation schemes in terms of their computational load and MSEs. Finally, we conclude our work in Section V and discuss some future research directions.

II. Target Tracking in Wireless Sensor Networks

The problem we seek to solve is to track a moving target using a WSN where N sensors are grid deployed in a square surveillance area of size b^2. The assumption of grid layout is not necessary but has been made here for convenience. Target tracking based on sensor readings can be performed for an arbitrary network layout if sensor placements are known in advance. All the sensors that are assigned bits report to a central fusion center, which estimates the target state, i.e., the position and the velocity of the target based on quantized sensor measurements. We assume that the target (e.g., an acoustic or an electromagnetic source) emits a signal from the location (x_t, y_t) at time t. We assume that the target is based on flat ground and all the sensors and target have the same height so that a 2-D model is sufficient to formulate the problem.

At time t, the target dynamics are defined by a 4-dimensional state vector $\mathbf{x}_t = [x_t \quad y_t \quad \dot{x}_t \quad \dot{y}_t]^T$ where $\dot{x}_t$ and $\dot{y}_t$ are the target velocities in the horizontal and the vertical directions, respectively. Target motion is defined by the following white noise acceleration model:

$$\mathbf{x}_{t+1} = \mathbf{F}\mathbf{x}_t + \boldsymbol{\upsilon}_t \tag{1}$$

where $\mathbf{F}$ models the state dynamics and $\boldsymbol{\upsilon}_t$ is the process noise which is assumed to be white, zero-mean and Gaussian with the following covariance matrix $\mathbf{Q}$:

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & \mathcal{D} & 0 \\ 0 & 1 & 0 & \mathcal{D} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{Q} = \rho \begin{bmatrix} \frac{\mathcal{D}^3}{3} & 0 & \frac{\mathcal{D}^2}{2} & 0 \\ 0 & \frac{\mathcal{D}^3}{3} & 0 & \frac{\mathcal{D}^2}{2} \\ \frac{\mathcal{D}^2}{2} & 0 & \mathcal{D} & 0 \\ 0 & \frac{\mathcal{D}^2}{2} & 0 & \mathcal{D} \end{bmatrix}. \tag{2}$$

In (2), $\mathcal{D}$ and ρ denote the time interval between adjacent sensor measurements and the process noise parameter, respectively. It is assumed that the fusion center has perfect information about the target state-space model (1) as well as the process noise statistics (2).

The target is assumed to be an acoustic or an electromagnetic source that follows the power attenuation model provided below [21]. At any given time t, the signal power received at the sensor i is given as

$$a_{i,t}^2 = \frac{P_0}{1 + \alpha d_{i,t}^n}. \tag{3}$$

By adopting this model, we prevent the receiver amplifier from saturation and the regularity conditions for PCRLB hold when the target is very close to a sensor. In (3), P_0 denotes the signal power of the target, n is the signal decay exponent and α is a scaling parameter. $d_{i,t}$ is the distance between the target and the i^{th} sensor, $d_{i,t} = \sqrt{(x_i - x_t)^2 + (y_i - y_t)^2}$, where (x_i, y_i) are the coordinates of the i^{th} sensor. Without loss of generality, α and n are assumed to be unity and 2, respectively. At time t, the received signal at sensor i is given by

$$z_{i,t} = a_{i,t} + n_{i,t} \tag{4}$$

where $n_{i,t}$ is the noise term modeled as additive white Gaussian noise (AWGN), i.e., $n_{i,t} \sim \mathcal{N}(0, \sigma^2)$, which represents the cumulative effects of sensor background noise and the modeling error of signal parameters.

Rather than transmitting analog sensor observations to the fusion center, transmitting a quantized version of sensor measurements decreases the amount of communication and therefore reduces the energy consumption. A sensor measurement $z_{i,t}$ at sensor i is locally quantized before its transmission to the fusion center using $R_{i,t}$ bits. Let $\mathbf{R}_t \triangleq [R_{1,t}, \ldots, R_{N,t}]$ be the vector of quantization rates used by the N sensors in the network. For the bit allocation problem, at each time step of tracking, $R_{i,t}$ can take values $m,\ m \in \{0, 1, \ldots, R\}$ where R is the maximum number of bits to be transmitted to the fusion center collectively by all the sensors. Let $L_m \triangleq 2^m - 1$ be the number of decision intervals for transmitting m bits to the fusion center and $D_{i,t}$ be the m-bit observation of sensor i quantized with rate $R_{i,t} = m$ at time step t, then

$$D_{i,t} = \begin{cases} 0 & -\infty < z_{i,t} < \eta_1^m \\ 1 & \eta_1^m < z_{i,t} < \eta_2^m \\ \vdots & \\ L_m - 1 & \eta_{L_m - 1}^m < z_{i,t} < \infty \end{cases} \tag{5}$$

where $\boldsymbol{\eta}^m = [\eta_0^m\ \eta_1^m\ \ldots\ \eta_{L_m}^m]$ with $\eta_0^m = -\infty$ and $\eta_{L_m}^m = \infty$. The quantization thresholds are assumed to be identical at each sensor for simplicity. We explain the selection of the quantization thresholds for each data rate $R_{i,t} = m$ later in this section. Given $\mathbf{x}_t$ and m, it is easy to show that the probability of a particular quantization output l is

$$P(D_{i,t} = l | \mathbf{x}_t, R_{i,t} = m) = Q\left(\frac{\eta_l^m - a_{i,t}}{\sigma}\right) - Q\left(\frac{\eta_{l+1}^m - a_{i,t}}{\sigma}\right) \tag{6}$$

where $Q(.)$ is the complementary distribution function of the standard Gaussian distribution with zero mean and unit variance

$$Q(x) = \int_x^{\infty} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{t^2}{2}\right) dt. \tag{7}$$

At time t, let the fusion center receive the data vector $\mathbf{D}_t = [D_{1,t}, \ldots, D_{N,t}]$ from the N sensors with the corresponding quantization rate vector $\mathbf{R}_t = [R_{1,t}, \ldots, R_{N,t}]$, then

$$p(\mathbf{D}_t | \mathbf{x}_t, \mathbf{R}_t) = \prod_{i=1}^{N} p(D_{i,t} | \mathbf{x}_t, R_{i,t}) \tag{8}$$

where we assume $p(D_{i,t} | \mathbf{x}_t, R_{i,t} = 0) = 1$.

A. PCRLB With Quantized Data

PCRLB provides the theoretical performance limit for a Bayesian estimator. Let $p(\mathbf{D}_t, \mathbf{x}_t)$ be the joint probability density of received data, $\mathbf{D}_t$ and the unknown state $\mathbf{x}_t$, and $\hat{\mathbf{x}}_t$ be an estimate of $\mathbf{x}_t$ at time step t. Based on $\mathbf{D}_t$, quantized with rate vector $\mathbf{R}_t$, and the prior probability distribution function of $\mathbf{x}_t$, $p(\mathbf{x}_t)$, the PCRLB on the mean squared estimation error has the form

$$E\left\{[\hat{\mathbf{x}}_t - \mathbf{x}_t][\hat{\mathbf{x}}_t - \mathbf{x}_t]^T | \mathbf{R}_t\right\} \geq \mathbf{J_t^{-1}}(\mathbf{R_t}) \tag{9}$$

where $\mathbf{J}_t(\mathbf{R}_t)$ is the 4×4 Fisher information matrix (FIM) with the elements

$$\mathbf{J}_t(\mathbf{R}_t)(i,j) = E_{p(\mathbf{D}_t, \mathbf{x}_t | \mathbf{R}_t)}\left[-\frac{\partial^2 \log p(\mathbf{D}_t, \mathbf{x}_t | \mathbf{R}_t)}{\partial \mathbf{x}_t(i) \partial \mathbf{x}_t(j)}\right] \quad i,j \in \{1, \ldots, 4\} \tag{10}$$

where $\mathbf{J}_t(\mathbf{R}_t)(i,j)$ denotes the i^{th} row, j^{th} column element of the matrix $\mathbf{J}_t(\mathbf{R}_t)$ and $\mathbf{x}_t(i)$ denotes the i^{th} element of vector $\mathbf{x}_t$ and $E_{p(\mathbf{D}_t, \mathbf{x}_t | \mathbf{R}_t)}[.]$ denotes the expectation with respect to $p(\mathbf{D}_t, \mathbf{x}_t | \mathbf{R}_t)$. Let $\nabla_{\mathbf{x}_t}^{\mathbf{x}_t} \triangleq \nabla_{\mathbf{x}_t} \nabla_{\mathbf{x}_t}^T$ denote the second order partial derivative operator with respect to $\mathbf{x}_t$. Using this notation, (10) can be rewritten in a more compact fashion as

$$\mathbf{J}_t(\mathbf{R}_t) = E_{p(\mathbf{D}_t, \mathbf{x}_t | \mathbf{R}_t)}\left[-\nabla_{\mathbf{x}_t}^{\mathbf{x}_t} \log p(\mathbf{D}_t, \mathbf{x}_t | \mathbf{R}_t)\right]. \tag{11}$$

Since $p(\mathbf{D}_t, \mathbf{x}_t | \mathbf{R}_t) = p(\mathbf{D}_t | \mathbf{x}_t, \mathbf{R}_t) p(\mathbf{x}_t)$, $\mathbf{J}_t(\mathbf{R}_t)$ can be decomposed into two parts as,

$$\mathbf{J}_t(\mathbf{R}_t) = \mathbf{J}_t^D(\mathbf{R}_t) + \mathbf{J}_t^P \tag{12}$$

where

$$\mathbf{J}_t^D(\mathbf{R}_t) \triangleq E_{p(\mathbf{D}_t | \mathbf{x}_t, \mathbf{R}_t) p(\mathbf{x}_t)}\left[-\nabla_{\mathbf{x}_t}^{\mathbf{x}_t} \log p(\mathbf{D}_t | \mathbf{x}_t, \mathbf{R}_t)\right]$$
$$\mathbf{J}_t^P \triangleq E_{p(\mathbf{x}_t)}\left[-\nabla_{\mathbf{x}_t}^{\mathbf{x}_t} \log p(\mathbf{x}_t)\right].$$

Note that $\mathbf{J}_t^D(\mathbf{R}_t)$ represents the Fisher information obtained from the data averaged over the prior distribution $p(\mathbf{x}_t)$ and $\mathbf{J}_t^P$ represents the *a priori* Fisher information. $E_{p(\mathbf{D}_t|\mathbf{x}_t)p(\mathbf{x}_t)}[.]$ and $E_{p(\mathbf{x}_t)}[.]$ denote expectations with respect to $p(\mathbf{D}_t|\mathbf{x}_t)p(\mathbf{x}_t)$ and $p(\mathbf{x}_t)$, respectively.

Given the vector of quantization rates $\mathbf{R}_t = [R_{1,t}, \ldots, R_{N,t}]$ and using (8) in (12), the data part of the Fisher information can be written as shown in (13) at the bottom of the page. For a given $\mathbf{x}_t$, let us define $\mathbf{J}_{i,t}^S(R_{i,t}|\mathbf{x}_t)$, as the Fisher information of sensor i

$$\begin{aligned}\mathbf{J}_{i,t}^S(R_{i,t}|\mathbf{x}_t) &\triangleq E_{p(\mathbf{D}_t|\mathbf{x}_t,\mathbf{R}_t)}\left[-\nabla_{\mathbf{x}_t}^{\mathbf{x}_t}\log p(D_{i,t}|\mathbf{x}_t)\right]\\ &= \sum_{l=0}^{2^{R_{i,t}}-1}\Bigg\{-\nabla_{\mathbf{x}_t}^{\mathbf{x}_t}\log p(D_{i,t}=l|\mathbf{x}_t,R_{i,t})\\ &\qquad\qquad \times p(D_{i,t}=l|\mathbf{x}_t,R_{i,t})\Bigg\}. \qquad (14)\end{aligned}$$

Then combining (13) and (14), sensor i's contribution to the Fisher information $\mathbf{J}_{i,t}^D(R_{i,t})$ can be stated as

$$\mathbf{J}_{i,t}^D(R_{i,t}) \triangleq \int_{\mathbf{x}_t}\mathbf{J}_{i,t}^S(R_{i,t}|\mathbf{x}_t)p(\mathbf{x}_t)d\mathbf{x}_t. \qquad (15)$$

Given $\mathbf{R}_t$, the Fisher information at time t can be written as

$$\mathbf{J}_t(\mathbf{R}_t) = \sum_{i=1}^{N}\mathbf{J}_{i,t}^D(R_{i,t}) + \mathbf{J}_t^P. \qquad (16)$$

From (14), after straightforward calculations, the (1,1) term of $\mathbf{J}_{i,t}^S(R_{i,t}|\mathbf{x}_t)$ can be derived as

$$\begin{aligned}&E\left[-\frac{\partial^2\log p(D_{i,t}|\mathbf{x}_t)}{\partial x_t^2}\right]\\ &= \sum_{l=0}^{2^{R_{i,t}}-1}\frac{1}{p(D_{i,t}=l|\mathbf{x}_t,R_{i,t})}\left(\frac{\partial p(D_{i,t}=l|\mathbf{x}_t,R_{i,t})}{\partial x_t}\right)^2. \qquad (17)\end{aligned}$$

The rest of the terms can be derived similarly. Using the procedures similar to [22], $\mathbf{J}_{i,t}^S(R_{i,t}|\mathbf{x}_t)$ can be obtained as follows:

$$\begin{aligned}&\mathbf{J}_{i,t}^S(R_{i,t}=m|\mathbf{x}_t) = n^2\kappa_{i,t}(m,x_i,y_i,x_t,y_t)\frac{a_{i,t}^2\alpha^2 d_{i,t}^{2n-4}}{(1+\alpha d_{i,t}^n)^2}\\ &\times\begin{bmatrix}(x_i-x_t)^2 & (x_i-x_t)(y_i-y_t) & 0 & 0\\ (x_i-x_t)(y_i-y_t) & (y_i-y_t)^2 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\end{bmatrix} \qquad (18)\end{aligned}$$

where

$$\begin{aligned}&\kappa_{i,t}(m,x_i,y_i,x_t,y_t)\\ &= \frac{1}{8\pi\sigma^2}\left\{\sum_{l=0}^{2^m-1}\frac{\left[e^{-\frac{(\eta_l^m-a_{i,t})^2}{2\sigma^2}} - e^{-\frac{(\eta_{l+1}^m-a_{i,t})^2}{2\sigma^2}}\right]^2}{p(D_i=l|\mathbf{x}_t)}\right\}. \qquad (19)\end{aligned}$$

Detailed derivation of (18) can be found in Appendix A. Note that in (18) and (19), $d_{i,t}$ and $a_{i,t}$ are functions of the sensor location (x_i, y_i) and target location (x_t, y_t). Note that the Fisher information depends on the quantization thresholds. In this paper we assume that quantization thresholds of all the sensors are identical and designed in advance. Next subsection introduces the optimization of quantization thresholds.

B. Optimization of Quantization Thresholds

Since the Fisher information and hence the PCRLB are functions of the quantization thresholds corresponding to each data rate $R_{i,t} = m$, the quantization thresholds need to be designed optimally to achieve better estimation accuracy. An algorithm to obtain the optimal quantization thresholds that minimizes the variance of the estimation errors has been proposed in [22]. If we assume that (x_i, y_i) and (x_t, y_t) are uniformly distributed in a region, we can minimize the sum of two diagonal elements of the CRLB matrix, after averaging the CRLB matrix over all the random parameters which may result in a large computational load since it requires a multiple fold integration. To alleviate this problem, some alternative methods to design the quantization thresholds were developed in [22].

Note that all the information about $[x_t, y_t]^T$ is contained in sensors' signal amplitudes $(a_{i,t})$'s. If all the signal amplitudes can be recovered from their quantized data $D_{i,t}$ accurately, an accurate estimate of $[x_t, y_t]^T$ can be obtained. In this paper, we use the Fisher information based heuristic quantization method [22] which maximizes the Fisher information about the signal amplitude $a_{i,t}$ contained in the quantized data $D_{i,t}$. We define $F_a(\boldsymbol{\eta}|x_i, y_i, x_t, y_t, R_{i,t} = m)$ as the Fisher information of the signal amplitude contained in quantized m-bit data, $D_{i,t}$, using a threshold $\boldsymbol{\eta}$. Note that $a_{i,t}$ is a function of $d_{i,t}$ for fixed P_0, α and n as defined in (3). Then given $R_{i,t} = m$, sensor location (x_i, y_i) and source location (x_t, y_t), it has been derived in [22] that $F_a(\boldsymbol{\eta}|x_i, y_i, x_t, y_t, R_{i,t} = m) = 4\kappa_{i,t}(m, x_i, y_i, x_t, y_t)$.

$$\begin{aligned}\mathbf{J}_t^D(R_{1,t},\ldots,R_{N,t}) &= \int_{\mathbf{x}_t}E_{p(\mathbf{D}_t|\mathbf{x}_t,\mathbf{R}_t)}\left[-\nabla_{\mathbf{x}_t}^{\mathbf{x}_t}\log p(\mathbf{D}_t|\mathbf{x}_t,\mathbf{R}_t)\right]p(\mathbf{x}_t)d\mathbf{x}_t\\ &= \sum_{i=1}^{N}\int_{\mathbf{x}_t}\left\{\sum_{l=0}^{2^{R_{i,t}-1}}-\nabla_{\mathbf{x}_t}^{\mathbf{x}_t}\log p(D_{i,t}=l|\mathbf{x}_t,R_{i,t})p(D_{i,t}=l|\mathbf{x}_t,R_{i,t})\right\}p(\mathbf{x}_t)d\mathbf{x}_t. \qquad (13)\end{aligned}$$

The Fisher information based heuristic quantization method [22] finds the decision thresholds that maximize

$$\begin{aligned} F_a(\boldsymbol{\eta}|R_{i,t}=m) &= E[-\nabla_{a_{i,t}}^{a_{i,t}} \log p(D_{i,t}|a_{i,t}(x_i,y_i,x_t,y_t))] \\ &= \int_{x_i,y_i,x_t,y_t} 4\kappa(m,x_i,y_i,x_t,y_t,) \\ &\quad dx_i dy_i dx_t dy_t \\ &= \int_u 4\kappa(m|u)p(u)du \end{aligned} \tag{20}$$

where $u = d_{i,t}^2$ and the Fisher information about the signal amplitude is averaged over the probability density function of u, $p(u)$, under the assumption that (x_i, y_i) and (x_t, y_t) are independent and identically distributed and follow a uniform distribution $U[\frac{-b}{2}, \frac{b}{2}]$. Derivation of $p(u)$ and other details of this quantizer design approach can be found in [22].

C. Particle Filtering With Quantized Data

The target tracking problem requires estimation of the state of a system that changes over time using a sequence of measurements. It is known that Kalman Filter provides the optimal solution to the Bayesian sequential estimation problem for linear and Gaussian systems. In nonlinear systems, the extended Kalman filter (EKF) can be used to provide a suboptimal solution by linearizing the nonlinear state dynamics and/or nonlinear measurement equations locally. However, it has been shown [23] that, even for linear and Gaussian systems, when the sensor measurements are quantized, the EKF fails to provide an acceptable performance especially when the number of quantization levels is small. Therefore, we propose to employ a particle filter to solve the Bayesian sequential estimation problem. Particle filters are sequential Monte Carlo methods based on particle representations of probability densities, which can be applied to any state-space model and generalize the Kalman filtering methods [24].

Let $\mathbf{D}_{1:t} = [\mathbf{D}_1, \ldots, \mathbf{D}_t]$ be the received sensor data up to time t which are obtained according to the data rates $\mathbf{R}_{1:t} = [\mathbf{R}_1, \ldots, \mathbf{R}_t]$. In particle filtering, the main idea is to find a discrete representation of the posterior distribution $p(\mathbf{x}_t|\mathbf{D}_{1:t})$ by using a set of particles $\{\mathbf{x}_t^s;\ s = 1, \ldots, N_s\}$ with associated weights $\{w_t^s;\ s = 1, \ldots, N_s\}$. The posterior density at t can be approximated as,

$$p(\mathbf{x}_t|\mathbf{D}_{1:t}) \approx \sum_{s=1}^{N_s} w_t^s \delta(\mathbf{x}_t - \mathbf{x}_t^s) \tag{21}$$

where N_s denotes the total number of particles. In this paper, we employ sequential importance resampling (SIR) particle filtering algorithm [24] to solve the nonlinear Bayesian filtering problem. In Algorithm 1, we provide a summary of the SIR based particle filtering rather than discussing the details. Note that T_S in Algorithm 1 denotes the number of time steps over which the target is tracked. A more detailed treatment of particle filtering can be found in a wide variety of publications such as [24].

Algorithm 1: SIR based Particle Filtering for Target Tracking

```
Set $t = 0$. Generate initial particles $\mathbf{x}_0^s \sim p(\mathbf{x}_0)$ with $\forall s$ , $w_0^s = N_s^{-1}$.
while $t \le T_S$ do
    (A1.1) $\mathbf{x}_{t+1}^s = \mathbf{F}\mathbf{x}_t^s + \boldsymbol{\upsilon}_t$ (Propagating particles)
    (A1.2) $p(\mathbf{x}_{t+1}|\mathbf{D}_{1:t}) = \frac{1}{N_s}\sum_{s=1}^{N_s} \delta(\mathbf{x}_{t+1} - \mathbf{x}_{t+1}^s)$
    (A1.3) Bit Allocation: Decide $\mathbf{R}_{t+1}$ and obtain sensor data $\mathbf{D}_{t+1}$
    (A1.4) $w_{t+1}^s \propto p(\mathbf{D}_{t+1}|\mathbf{x}_{t+1}^s, \mathbf{R}_{t+1})$ (Updating weights)
    $w_{t+1}^s = \frac{w_{t+1}^s}{\sum_{j=1}^{N_s} w_{t+1}^j}$ (Normalizing weights)
    $\hat{\mathbf{x}}_{t+1} = \sum_{s=1}^{N_s} w_{t+1}^s \mathbf{x}_{t+1}^s$
    (A1.5) $\{\mathbf{x}_{t+1}^s, N_s^{-1}\} = \text{Resampling}(\mathbf{x}_{t+1}^s, w_{t+1}^s)$
    (A1.6) $t = t + 1$
end while
```

In Algorithm 1, $p(\mathbf{D}_{t+1}|\mathbf{x}_{t+1}^s, \mathbf{R}_{t+1})$ is obtained according to (6) and (8). Resampling step avoids the situation that all but one of the importance weights are close to zero [24].

By using (10) to (16), at time t, one can compute the PCRLB on the estimation error and the corresponding FIM, for a given bit allocation scheme $\mathbf{R}_t$ and prior distribution $p(\mathbf{x}_t)$. For the bit allocation problem, at time t, from (A1.2), we first get the prior $p(\mathbf{x}_{t+1}|\mathbf{D}_{1:t})$ for state $\mathbf{x}_{t+1}$ based on data received up to time t.

Under the Gaussian assumption, maximizing the determinant of the FIM is equivalent to minimizing the volume of the uncertainty ellipsoid [15]. Therefore, we determine bit allocation scheme for time $t+1$, $\mathbf{R}_{t+1}$, by maximizing the determinant of the Fisher information about $\mathbf{x}_{t+1}$ as

$$\begin{aligned} \max_{\mathbf{R}_{t+1}} \quad & \det(\mathbf{J}_{t+1}(\mathbf{R}_{t+1})) \\ \text{s.t.} \quad & \sum_{i=1}^{N} R_{i,t+1} = R. \end{aligned} \tag{22}$$

The fusion center then informs the sensors about $\mathbf{R}_{t+1}$ and sensors transmit their quantized measurements $\mathbf{D}_{t+1}$ accordingly. The Fisher information, $\mathbf{J}_{t+1}(\mathbf{R}_{t+1})$ is written as

$$\begin{aligned} \mathbf{J}_{t+1}(\mathbf{R}_{t+1}) = E_{p(\mathbf{D}_{t+1},\mathbf{x}_{t+1}|\mathbf{D}_{1:t},\mathbf{R}_{t+1})} \\ \left[-\nabla_{\mathbf{x}_{t+1}}^{\mathbf{x}_{t+1}} \log p(\mathbf{D}_{t+1}, \mathbf{x}_{t+1}|\mathbf{D}_{1:t}, \mathbf{R}_{t+1}))\right]. \end{aligned}$$

Following the derivations from (10) to (16), the Fisher information, $\mathbf{J}_{t+1}(\mathbf{R}_{t+1})$, is obtained as,

$$\mathbf{J}_{t+1}(\mathbf{R}_{t+1}) = \sum_{i=1}^{N} \mathbf{J}_{t+1}^{D}(R_{i,t+1}) + \mathbf{J}_{t+1}^{P}. \tag{23}$$

Using the particle approximation

$$p(\mathbf{x}_{t+1}|\mathbf{D}_{1:t}) \approx \frac{1}{N_S}\sum_{s=1}^{N_S} \delta(\mathbf{x}_{t+1} - \mathbf{x}_{t+1}^s) \tag{24}$$

$\mathbf{J}^D_{t+1}(R_{i,t+1})$ is found from

$$\mathbf{J}^D_{t+1}(R_{i,t+1}) = \frac{1}{N_S}\sum_{s=1}^{N_S}\mathbf{J}^S_{t+1}(R_{i,t+1}|\mathbf{x}^s_{t+1}). \tag{25}$$

As in (12), $\mathbf{J}^P_{t+1} = E_{p(\mathbf{x}_{t+1}|\mathbf{D}_{1:t})}[-\nabla^{\mathbf{x}_{t+1}}_{\mathbf{x}_{t+1}}\log p(\mathbf{x}_{t+1}|\mathbf{D}_{1:t})]$ has been defined as the prior Fisher information of $\mathbf{x}_{t+1}$. According to (24), $p(\mathbf{x}_{t+1}|\mathbf{D}_{1:t})$ has a non-parametric representation by a set of random particles with associated weights, so it is very difficult to calculate the exact $\mathbf{J}^P_{t+1}$[25]. Instead, we use a Gaussian approximation such that $p(\mathbf{x}_{t+1}|\mathbf{D}_{1:t}) \approx \mathcal{N}(\boldsymbol{\mu}_{t+1}, \boldsymbol{\Sigma}_{t+1})$, where

$$\boldsymbol{\mu}_{t+1} = \frac{1}{N_s}\sum_{s=1}^{N_s}\mathbf{x}^s_{t+1}$$

and

$$\boldsymbol{\Sigma}_{t+1} = \frac{1}{N_s}\sum_{s=1}^{N_s}(\mathbf{x}^s_{t+1}-\boldsymbol{\mu}_{t+1})(\mathbf{x}^s_{t+1}-\boldsymbol{\mu}_{t+1})^T.$$

Given the Gaussian approximation, it is easy to show that $\mathbf{J}^P_{t+1} = \boldsymbol{\Sigma}^{-1}_{t+1}$.

Having introduced the system model, next section presents the dynamic bit allocation methods.

III. Dynamic Bit Allocation for Target Tracking

In order to maximize (22), an exhaustive search can be employed to find the optimal bit allocations. For a network of N sensors and the total bits constraint R, there are a total of $\binom{R+N-1}{N-1} = \frac{(N+R-1)!}{(N-1)!R!}$ possible bit allocation solutions. For large N and R, such an exhaustive search may not be feasible in real time. Therefore suboptimal but computationally more efficient algorithms are required which we explore in this section.

A. Convex Optimization Based Dynamic bit Allocation

In this section, we generalize the sensor selection method proposed in [1] and consider the convex optimization based bit allocation method as a benchmark before developing other suboptimal methods. We maximize the determinant of the FIM by maximizing its log determinant. Using Boolean variables $q_{i,m} \in \{0,1\}$ where $i \in \{1,\ldots,N\}$ and $m \in \{0,1,\ldots,R\}$, the bit allocation problem can be explicitly formulated as follows: [see (26) at the bottom of the page]. In the above formulation, $\mathbf{q}_{t+1} = [q_{1,0}, q_{2,0},\ldots,q_{N,0},\ldots,q_{1,R},q_{2,R},\ldots,q_{N,R}]^T$ denotes the bit allocation scheme for time $t+1$ where $q_{i,m} = 1$ when sensor i transmits its measurement in m bits and $\mathbf{J}^D_{i,t+1}(m)$ is the corresponding FIM of sensor i. Note that we drop the time index $t+1$ from the elements of vector $\mathbf{q}_{t+1}$ to simplify the notation. All constraints are equality constraints where the first N constraints guarantee that each sensor can transmit using only one of the quantization rates. If $q_{i,0} = 1$ is selected, the quantized measurement of the sensor is not transmitted to the fusion center. The $(N+1)^{th}$ equality constraint satisfies the total number of bits constraint and the last $N(R+1)$ constraints restrict $q_{i,m}$ to be Boolean.

Similar to the convex optimization approach presented in [1], the last $N(R+1)$ constraints can be relaxed by replacing the Boolean variables $q_{i,m} \in \{0,1\}$ with their continuous counterparts, $\hat{q}_{i,m} \in [0,1]$. Then the problem becomes [see (27) at the bottom of the next page], where the last $2N(R+1)$ inequality constraints ensure $0 \leq \hat{q}_{i,m} \leq 1$. The Lagrangian function for the relaxed problem can be written and optimal solution can be found from the Karush-Kuhn-Tucker (KKT) conditions of the relaxed problem. However, finding the optimal values of each $\hat{q}_{i,m}$ from the KKT conditions is neither efficient since it requires the solutions of tedious equations, nor necessary. Since the relaxed problem is convex with linear equality and inequality constraints, interior-point methods reach the optimal solution of (27) by traversing the interior of the feasible region [18].

We solve the problem in (27) using a particular interior point method, the barrier method, which approximately formulates the inequality constrained problem (27) as an equality constrained problem where the inequality constraints are included implicitly to the objective specified by the parameter ζ. The resulting optimization problem with equality constraints can be solved with Newton's method [18].

$$\begin{aligned}\min_{\hat{\mathbf{q}}_{t+1}} \quad & -\log\det\left(\sum_{m=0}^{R}\sum_{i=1}^{N}\hat{q}_{i,m}\mathbf{J}^D_{i,t+1}(m) + \mathbf{J}^P_{t+1}\right)\\ & -\frac{1}{\zeta}\sum_{m=0}^{R}\sum_{i=1}^{N}\Big(\log(\hat{q}_{i,m}) + \log(1-\hat{q}_{i,m})\Big)\\ \text{subject to} \quad & \mathcal{A}\hat{\mathbf{q}}_{t+1} = \mathbf{b}.\end{aligned} \tag{28}$$

$$\begin{aligned}\min_{\mathbf{q}_{t+1}} \quad & -\log\det(\mathbf{J}_{t+1}(\mathbf{q}_{t+1})) = -\log\det\left(\sum_{m=0}^{R}\sum_{i=1}^{N}q_{i,m}\mathbf{J}^D_{i,t+1}(R_{i,t+1}=m) + \mathbf{J}^P_{t+1}\right)\\ \text{subject to} \quad & \sum_{m=0}^{R}q_{i,m} = 1 \quad i \in \{1,\ldots,N\}\\ & \sum_{m=0}^{R}\sum_{i=1}^{N} m\, q_{i,m} \quad = R \quad q_{i,m} \in \{0,1\} \quad m \in \{0,1,\ldots,R\} \quad i \in \{1,\ldots,N\}.\end{aligned} \tag{26}$$

Note that the equality constraints in (27) can be written in a matrix equality form where (see the equation at the bottom of the page). Let us define

$$\phi(\hat{\mathbf{q}}_{t+1},\zeta) \triangleq -\log\det\left(\sum_{m=0}^{R}\sum_{i=1}^{N}\hat{q}_{i,m}\mathbf{J}_{i,t+1}^{D}(m)-\mathbf{J}_{t+1}^{P}\right) \\ -\frac{1}{\zeta}\sum_{m=0}^{R}\sum_{i=1}^{N}\Big(\log\left(\hat{q}_{i,m}\right)+\log\left(1-\hat{q}_{i,m}\right)\Big)$$

where the gradient vector $\nabla_{\hat{q}_{t+1}}\phi(\hat{\mathbf{q}}_{t+1},\zeta)$, and the Hessian matrix $\nabla_{\hat{q}_{t+1}}^{\hat{q}_{t+1}}\phi(\hat{\mathbf{q}}_{t+1},\zeta)$ of $\phi(\hat{\mathbf{q}}_{t+1},\zeta)$ are provided in Appendix B.

Note that in [1], the relaxed problem similar to (27) has been solved approximately for a particular value of ζ. In general, the barrier method solves (27) for a sequence of iterations corresponding to increasing values of ζ until a stopping criteria is met [18]. The barrier method is simplified in Algorithm 2 [18],

Algorithm 2: Barrier Method—Outer Iterations

```
Find a feasible starting point $\hat{\mathbf{q}}_{t+1}$ and set $\zeta := \zeta^0 > 0, \beta > 1$
and precision $\epsilon_1 > 0$.
Repeat
(A2.1) Obtain $\hat{\mathbf{q}}_{t+1}^*$ from the Newton iterations as in Alg. 3
designed for the KKT system of $\phi(\hat{\mathbf{q}}_{t+1},\zeta)$ (Inner iterations)
(A2.2) Update $\hat{\mathbf{q}}_{t+1} = \hat{\mathbf{q}}_{t+1}^*$
(A2.3) Quit if $\frac{2N(R+1)}{\zeta} < \epsilon_1$.
(A2.4) Increase $\zeta = \zeta\beta$.
```

In order to initialize the barrier method, an initial feasible point for (28) needs to be found. The underdetermined system $\mathcal{A}\hat{\mathbf{q}}_{t+1}=\mathbf{b}$ has infinite number of solutions but there is only a subset of solutions which are feasible satisfying $\mathbf{0}\leq\hat{\mathbf{q}}_{t+1}\leq\mathbf{1}$ where $\mathbf{0}$ and $\mathbf{1}$ are the all zero and all one vectors, respectively. We formulate the following linear optimization sub-problem to find an initial feasible solution:

$$\begin{aligned} \min_{\hat{\mathbf{q}}_{t+1}} \quad & -\sum_{m=0}^{R}\sum_{i=1}^{N}\hat{q}_{i,m} \\ \text{subject to} \quad & \mathcal{A}\hat{\mathbf{q}}_{t+1}=\mathbf{b} \\ & 0\leq\hat{q}_{i,m}\leq 1 \;\; m\in\{0,\ldots,R\} \;\; i\in\{1,\ldots,N\}. \end{aligned} \tag{29}$$

For step (A2.1) of Algorithm 2, for a particular value of ζ, (28) needs to be solved from its KKT system

$$\begin{pmatrix} \nabla_{\hat{q}_{t+1}}^{\hat{q}_{t+1}}\phi(\hat{\mathbf{q}}_{t+1},\zeta) & \mathcal{A}^T \\ \mathcal{A} & 0 \end{pmatrix}\begin{pmatrix} \Delta\mathbf{Z} \\ \omega \end{pmatrix} = \begin{pmatrix} -\nabla_{\hat{q}_{t+1}}\phi(\hat{\mathbf{q}}_{t+1},\zeta) \\ 0 \end{pmatrix} \tag{30}$$

where $\Delta\mathbf{Z}$ is the Newton Step, ω is the optimal dual variable, i.e., the Lagrange multiplier of the equality constraint. The optimal solution to (A2.1) can be found from Newton's method as described in Algorithm 3. We refer to this algorithm as inner iterations, since it has to be carried out to find the optimal solution at a particular ζ.

$$\begin{aligned} \min_{\hat{\mathbf{q}}_{t+1}} \quad & -\log\det(\mathbf{J}_{t+1}(\hat{\mathbf{q}}_{t+1})) = -\log\det\left(\sum_{m=0}^{R}\sum_{i=1}^{N}\hat{q}_{i,m}\mathbf{J}_{i,t+1}^{D}(R_{i,t+1}=m)+\mathbf{J}_{t+1}^{P}\right) \\ \text{subject to} \quad & \sum_{m=0}^{R}\hat{q}_{i,m}=1 \quad i\in\{1,\ldots,N\} \\ & \sum_{m=0}^{R}\sum_{i=1}^{N}m\,\hat{q}_{i,m}=R \\ & -\hat{q}_{i,m}\leq 0 \quad m\in\{0,1,\ldots,R\} \quad i\in\{1,\ldots,N\} \\ & \hat{q}_{i,m}-1\leq 0 \quad m\in\{0,1,\ldots,R\} \quad i\in\{1,\ldots,N\}. \end{aligned} \tag{27}$$

$$\mathcal{A} \triangleq \begin{pmatrix} 1 & 0 & \ldots & 0 & 1 & 0 & \ldots & 0 & \ldots & 1 & \ldots & 0 & 0 \\ 0 & 1 & \ldots & 0 & 0 & 1 & \ldots & 0 & \ldots & 0 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \ldots & 1 & 0 & 0 & \ldots & 1 & \ldots & 0 & \ldots & 0 & 1 \\ 0 & 0 & \ldots & 0 & 1 & 1 & \ldots & 1 & \ldots & R & \ldots & R & R \end{pmatrix}$$

$$\hat{\mathbf{q}}_{t+1} \triangleq \left(\hat{q}_{1,0} \;\; \hat{q}_{2,0} \;\; \cdots \;\; \hat{q}_{N,0} \;\; \hat{q}_{1,1} \;\; \hat{q}_{2,1} \;\; \cdots \;\; \hat{q}_{N,1} \;\; \cdots \;\; \hat{q}_{1,R} \;\; \cdots \;\; \hat{q}_{N-1,R} \;\; \hat{q}_{N,R}\right)^T$$

and

$$\mathbf{b} \triangleq \left(1 \;\; 1 \;\; \ldots \;\; 1 \;\; R\right)^T.$$

Algorithm 3: Newton's Method—Inner Iterations

```
Get the feasible starting point $\hat{\mathbf{q}}_{t+1}$ from the outer iteration and set precision $\epsilon_2 > 0$
Repeat
(A3.1) Compute the Newton Step $\Delta\mathbf{Z}$ from Alg. 4 and Newton decrement $\lambda = \{[-\nabla_{\hat{q}_{t+1}}\phi(\zeta)]^T \Delta\mathbf{Z}\}^{\frac{1}{2}}$
(A3.2) Choose Step size $s$ by backtracking line search [18].
(A3.3) Update $\hat{\mathbf{q}}_{t+1} = \hat{\mathbf{q}}_{t+1} + s\Delta\mathbf{Z}$
(A3.4) Quit if $\frac{\lambda^2}{2} \leq \epsilon_2$, then $\hat{\mathbf{q}}^*_{t+1} = \hat{\mathbf{q}}_{t+1}$, else go to Step (A3.1).
```

At step (A3.1), the Newton step $\Delta\mathbf{Z}$ is obtained from the KKT system (30) by block elimination as given in Algorithm 4,

Algorithm 4: Solving KKT System by Block Elimination

```
(A4.1) Form $(\nabla^{\hat{q}_{t+1}}_{\hat{q}_{t+1}}\phi)^{-1}\mathcal{A}^T$ and $(\nabla^{\hat{q}_{t+1}}_{\hat{q}_{t+1}}\phi)^{-1}(\nabla_{\hat{q}_{t+1}}\phi)$.
(A4.2) Form $\mathcal{S} = -\mathcal{A}(\nabla^{\hat{q}_{t+1}}_{\hat{q}_{t+1}}\phi)^{-1}\mathcal{A}^T$
(A4.3) Determine $\omega$ by solving $\mathcal{S}\omega = \mathcal{A}(\nabla^{\hat{q}_{t+1}}_{\hat{q}_{t+1}}\phi)^{-1}(\nabla_{\hat{q}_{t+1}}\phi)$.
(A4.4) Determine $\Delta\mathbf{Z}$ by solving $(\nabla^{\hat{q}_{t+1}}_{\hat{q}_{t+1}}\phi)^{-1}\Delta\mathbf{Z} = \mathcal{A}^T\omega - (\nabla_{\hat{q}_{t+1}}\phi)$.
```

In order to compute the complexity of the convex optimization based bit allocation method, we ignore the complexity for computing $\nabla_{\hat{q}_{t+1}}\phi(\zeta)$ and $\nabla^{\hat{q}_{t+1}}_{\hat{q}_{t+1}}\phi(\zeta)$. The cost of block elimination to solve (28) is dominated by the Cholesky decomposition of $\nabla^{\hat{q}_{t+1}}_{\hat{q}_{t+1}}\phi$ which is used to find $(\nabla^{\hat{q}_{t+1}}_{\hat{q}_{t+1}}\phi)^{-1}$. Let us define $\mathcal{V} \triangleq N(R+1)$. Then, in order to compute the Cholesky decomposition of the matrix $(\nabla^{\hat{q}_{t+1}}_{\hat{q}_{t+1}}\phi)$, we require a total of $\frac{1}{6}(\mathcal{V}^3 - \mathcal{V})$ summations and multiplications and $\frac{1}{6}(3\mathcal{V}^2 - 3\mathcal{V})$ divisions [26]. Thus, the complexity of bit allocation based on convex optimization increases with $\mathcal{O}(N^3(R+1)^3)$ at each iteration.

1) The Probabilistic Transmission Scheme: From the optimal solution of the relaxed problem $\hat{\mathbf{q}}^*_{t+1}$, the bit allocation for the next time step needs to be determined. For the sensor selection problem, the authors in [1] employed a simple scheme in which k sensors are selected out of N sensors by first sorting $\hat{\mathbf{q}}^*_{t+1}$ in descending order and then setting k largest elements of $\hat{\mathbf{q}}^*_{t+1}$ to one. For the bit allocation problem, a similar solution is to sort the probabilities $\hat{q}_{i,m}$ in descending order and then to assign 1 starting from the largest probability until the total bit constraint is satisfied. However, in this paper, we consider a randomized scheme similar to the ones used in [1] and [27]. Since the elements of $\hat{\mathbf{q}}^*_{t+1}$ are within the range $[0, 1]$, we can consider each $\hat{q}_{i,m}$ as the transmission probability of sensor i, transmitting information in m bits. Instead of putting a strict total bit constraint, i.e., $\sum_{i=1}^{N} R_{i,t+1} = R$, the probabilistic transmission puts a weak constraint on the total bit constraint and ensures that the sensors on the average transmit R bits to the fusion center. We present a numerical example on the probabilistic transmission scheme in Section IV.

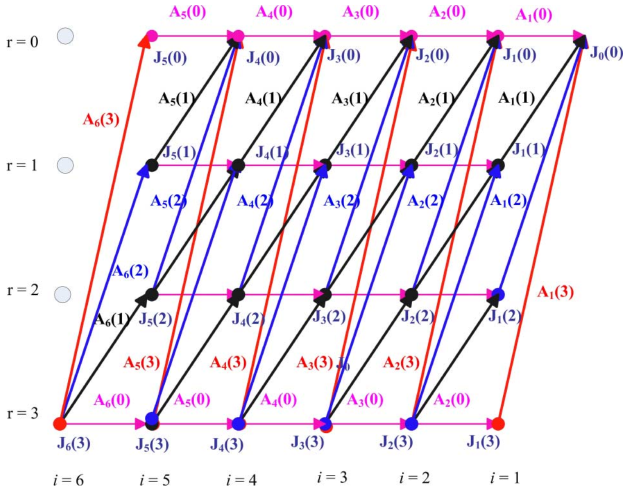


Fig. 1. Trellis of the DP for tracking time step t. $(N = 6,\ R = 3)$.

B. Approximate Dynamic Programming Based Bit Allocation

In this section, we present the bit allocation algorithm based on A-DP which will be shown to provide near optimal solution but requires much less computation time than the convex optimization approach. Note that the Fisher information matrix can be expressed as the summation of each sensor's individual Fisher information matrices as defined in (23). In this section, we formulate an approximate DP recursion in tracking applications where we maximize the determinant of the Fisher information matrix subject to the total bit constraint.

Typically DP involves progression along time. But in our problem formulation, the DP progresses across sensors and is executed at each time step of tracking to determine the bit allocations of the next time step. Since A-DP is performed at each time step, for simplicity, the time index $t+1$ for Fisher information matrix is dropped. Instead an index for the stages in DP is adopted. Let $\mathbf{J}_N = \mathbf{J}_{t+1}$ and $\mathbf{A}_i(R_i) = \mathbf{J}^D_{t+1}(R_i)$ be the reward in terms of Fisher information when sensor i quantizes its measurement in R_i bits $(R_i \in \{0, 1, \ldots, R\})$. While constructing the DP trellis, the bit allocation problem is first divided into $N+1$ stages which correspond to N sensors and a termination stage. We define the state of a stage as the remaining bits for the usage of sensor i. So each stage has $R+1$ states associated with it. The bit allocation chosen at any sensor (stage) determines the feasible states at the next sensor. An example DP trellis is shown in Fig. 1 with $N = 6$ and $R = 3$ which implies a total of 7 stages and 4 states in the DP trellis. As an example, sensor 1 is at state $r = 1$ means 2 bits have already been used by the other $N-1$ sensors and 1 bit is available for sensor 1. Then, sensor 1 can only take the action $\mathbf{A}_1(1)$ and the DP goes to the termination stage (stage 0) which has only the 0 bit available state.

For such a DP trellis, we have

$$\begin{aligned} \mathbf{J}_N &= \mathbf{A}_N(R_N) + \{\mathbf{A}_{N-1}(R_{N-1}) + \cdots + \mathbf{A}_1(R_1) + \mathbf{J}_0\} \\ &= \mathbf{A}_N(R_N) + \mathbf{J}_{N-1} \\ &\vdots \\ \mathbf{J}_1 &= \mathbf{A}_1(R_1) + \mathbf{J}_0 \end{aligned} \tag{31}$$

where $\mathbf{J}_0 = \mathbf{\Sigma}_{t+1}^{-1}$ and $\sum_{i=1}^{N} R_i = R$. According to the matrix determinant lemma [28],

$$\det(\mathbf{X}+\mathbf{A}) = \det(\mathbf{X}+\mathbf{AI}) = \det(\mathbf{X})\det(\mathbf{I}+\mathbf{X}^{-1}\mathbf{A}).$$

With $\mathbf{X} = \mathbf{J}_{i-1}$, $\mathbf{A} = \mathbf{A}_i(R_i)$, and $\mathbf{I}$ being the identity matrix, we have

$$\begin{aligned}\log\Big\{\det(\mathbf{J}_N)\Big\} &= \log\Big\{\det(\mathbf{J}_{N-1})\Big\}\\ &\quad + \log\Big\{\det\left[\mathbf{I}+\mathbf{J}_{N-1}^{-1}\mathbf{A}_N(R_N)\right]\Big\}\\ &\vdots\\ \log\Big\{\det(\mathbf{J}_1)\Big\} &= \log\Big\{\det(\mathbf{J}_0)\Big\}\\ &\quad + \log\Big\{\det(\mathbf{I}+\mathbf{J}_0^{-1}\mathbf{A}_1(R_1))\Big\}.\end{aligned}$$

We can maximize $\det(\mathbf{J}_N)$, by maximizing $\log\Big\{\det(\mathbf{J}_N)\Big\}$. The DP recursion at each stage is formulated as follows: the trellis starts from $\mathbf{J}_0(0)$ and for the first stage $(i = 1)$ and for all $r \in \{0, 1, \ldots, R\}$

$$\begin{aligned}&\log\Big[\det[\mathbf{J}_1(r)]\Big]\\ &\quad = \log\Big[\det[\mathbf{I}+\mathbf{J}_0^{-1}(0)\mathbf{A}_1(r)]\Big] + \log\Big[\det[\mathbf{J}_0(0)]\Big].\end{aligned} \tag{32}$$

Then for all the intermediate stages $i \in \{2, \ldots, N-1\}$ and for all $r \in \{0, 1, \ldots, R\}$

$$\begin{aligned}&\log\Big[\det[\mathbf{J}_i(r)]\Big]\\ &\quad = \max_{k=0,1,\ldots,r}\Bigg\{\log\Big[\det[\mathbf{I}+\mathbf{J}_{i-1}^{-1}(r-k)\mathbf{A}_i(k)]\Big]\\ &\qquad + \log\Big[\det[\mathbf{J}_{i-1}(r-k)]\Big]\Bigg\}.\end{aligned} \tag{33}$$

Finally for the last stage $i = N$

$$\begin{aligned}&\log\Big[\det[\mathbf{J}_N(R)]\Big]\\ &\quad = \max_{k=0,1,\ldots,R}\Bigg\{\log\Big[\det[\mathbf{I}+\mathbf{J}_{N-1}^{-1}(R-k)\mathbf{A}_i(k)]\Big]\\ &\qquad + \log\Big[\det[\mathbf{J}_{N-1}(R-k)]\Big]\Bigg\}.\end{aligned} \tag{34}$$

In (32), (33), and (34), the reward of sensor i's transmission in R_i bits depends not only on $\mathbf{A}_i(R_i)$ but also on the FIM of the previous stage $\mathbf{J}_{i-1}^{-1}$. So at each stage i, the FIM, $\mathbf{J}_i(r)$, which has the maximum determinant should be stored in a memory for its use at the next recursion. Note that the proposed A-DP may not yield the maximum matrix determinant at the final stage. The suboptimality of the A-DP recursions is discussed later in this section.

We analyze the computational complexity of A-DP in terms of the number of matrix summations. Note that the number of element-wise summations is a scaled version of number of matrix summations. The first stage needs R matrix summations to compute the FIM at all states. For all the intermediate stages, at each state r, $(r \in \{1, \ldots, R\})$, r different matrix summations are required to find the FIM with the maximum determinant. Finally, at stage N, A-DP again needs R matrix summations in order to maximize the determinant of $\det \mathbf{J}_N$. So, the A-DP totally requires

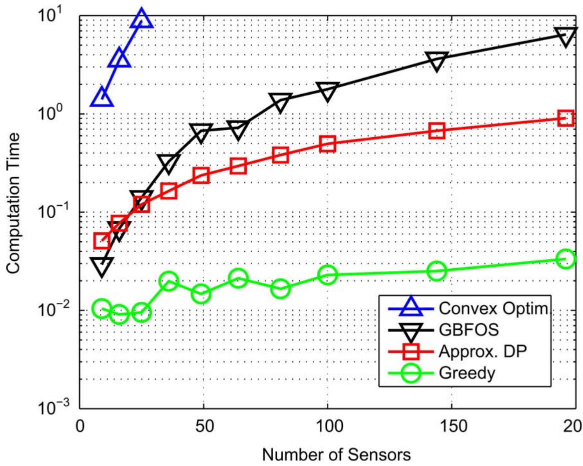


Fig. 2. Computation time in seconds for convex optimization, A-DP, GBFOS, and greedy search $(R = 5)$.

$$R + (N-2)\left[\sum_{r=1}^{R} r\right] + R = 2R + (N-2)\frac{R(R+1)}{2}$$

matrix summations which is linear in N and quadratic in R.

1) Suboptimality of the A-DP Recursions: For a given state of a stage, we choose the path with the maximum determinant of the FIM and dismiss all the other paths arriving at this state. The proposed DP recursions would yield the optimal solution to maximize the determinant of the FIM, if the following property were satisfied

$$\begin{aligned}\text{if}\quad &\det\{\mathbf{J}'\} \geq \det\{\mathbf{J}''\}\\ \text{then}\quad &\det\{\mathbf{A}+\mathbf{J}'\} \geq \det\{\mathbf{A}+\mathbf{J}''\}\end{aligned} \tag{35}$$

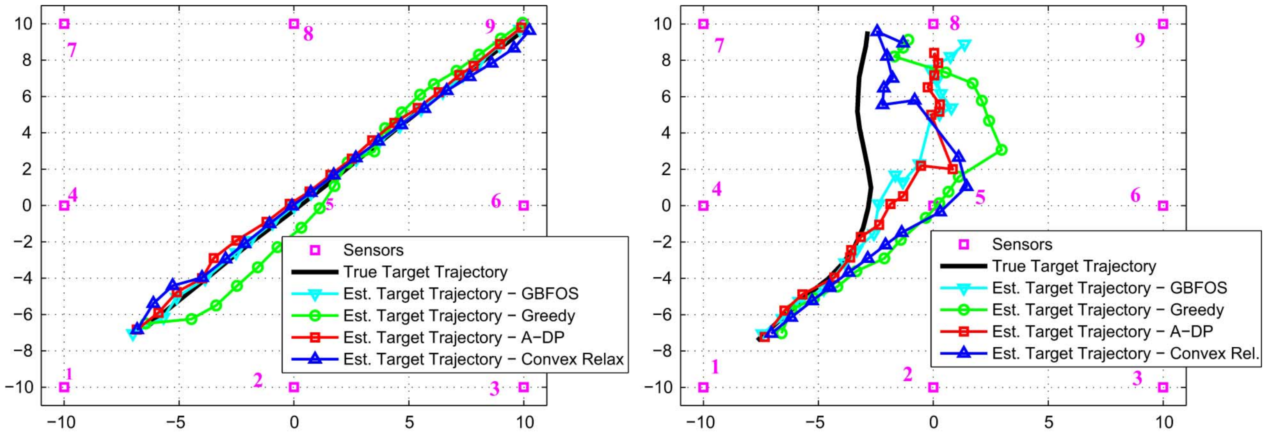
for some positive semidefinite matrices $\mathbf{J}'$, $\mathbf{J}''$ and $\mathbf{A}$. Unfortunately, the above property is not necessarily true. Consider the simple example, $\mathbf{J}' = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ and $\mathbf{J}'' = \begin{pmatrix} 1 & -0.1 \\ -0.1 & 1 \end{pmatrix}$ where $\det\{\mathbf{J}'\} > \det\{\mathbf{J}''\}$. Let $\mathbf{A} = \begin{pmatrix} 1 & 0.1 \\ 0.1 & 1 \end{pmatrix}$. Then $\det\{\mathbf{A}+\mathbf{J}'\} < \det\{\mathbf{A}+\mathbf{J}''\}$.

At each stage of DP, we only store the FIM with the maximum determinant. Therefore, the final solution obtained by the DP recursions becomes suboptimal since not all the feasible solutions are enumerated.

C. Existing Suboptimal Bit Allocation Methods

In this section, we review some existing suboptimal methods that are suitable for solving the bit allocation problem in target tracking applications.

1) GBFOS Algorithm: This algorithm has been first proposed in [14] for dynamic bit allocation in target tracking. The GBFOS algorithm starts by assigning the maximum number of bits, R to each sensor in the network and then reduces the number of bits one bit at a time until the sum rate constraint is satisfied. The GBFOS algorithm can be stated as in Algorithm 5. Note that in order to simplify the notation, we drop the time



(a) (b)

Fig. 3. A WSN with $N = 9$ sensors tracking sample targets. (a) $\rho = 2.5 \times 10^{-3}$. (b) $\rho = 0.1$.

index $t+1$ in the algorithm. As shown in Step (A5.1), at each iteration, the GBFOS algorithm searches the N sensors and reduces the bits of the sensor by one which ensures the minimum reduction of the determinant of the FIM. An efficient implementation of the GBFOS algorithm and its complexity analysis are given as follows:

Algorithm 5: GBFOS—Bit Allocation Algorithm

```
Set $\mathbf{R}_0 = [R_1 = R, \ldots, R_N = R]$ and $\mathbf{J}^0 = \mathbf{J}^P_{t+t} + \mathbf{A}_1(R) + \mathbf{A}_2(R) + \cdots + \mathbf{A}_N(R)$.
FOR $i = 1 : (N-1)R$
    (A5.1) FOR $k = 1 : N$
        IF $R_k > 0$
            Reduce one bit from sensor $k$ and compute $\det(\mathbf{J}^i(k))$ where
            $\mathbf{J}^i(k) = \mathbf{J}^{i-1} + \mathbf{A}_k(R_k - 1) - \mathbf{A}_k(R_k)$.
        ENDIF
    ENDFOR
    (A5.2) $\forall$ $k$ with $R_k > 0$, find the sensor $p^*$ for which $\det(\mathbf{J}^i(k))$ is the maximum:
    $p^* = \arg\max_{k \text{ where } R_k > 0} \det(\mathbf{J}^i(k))$.
    (A5.3) Decrement $R_{p^*} = R_{p^*} - 1$, update $\mathbf{R}_i = [R_1, \ldots, R_{p^*}, \ldots, R_N]$ and set $\mathbf{J}^i = \mathbf{J}^i(p^*)$.
ENDFOR
```

Let us define $\mathbf{J}^i$ as the FIM after the i^{th} iteration, and $\mathbf{A}_k(R_k) \triangleq \mathbf{J}^D_{t+1}(R_k)$ as sensor k's contribution to the FIM using R_k bits. In the beginning, we need to calculate $\mathbf{J}^0 = \mathbf{J}^P_{t+1} + \mathbf{A}_1(R) + \mathbf{A}_2(R) + \cdots + \mathbf{A}_N(R)$. As a result, a total of N matrix summations are needed. At the i-th iteration, where $i = 1, \ldots, (N-1)R$, there are at most N different ways to reduce 1 bit. Assuming one particular solution is to reduce one bit at the k-th sensor, ($k \in \{1, 2, \ldots, N\}$), and $\mathbf{J}^i(k) = \mathbf{J}^{i-1} + \mathbf{A}_k(R_k - 1) - \mathbf{A}_k(R_k)$, which requires two matrix summations. Hence at each iteration, at most $2N$ matrix summations are required. At the end of the i^{th} iteration, we store $\mathbf{J}^i(k)$ with the maximum determinant. In summary, we need at most $N + 2N(N-1)R$ matrix summations, which is quadratic in N and linear in R. Note that this is an upper bound on complexity.

2) Greedy Algorithm: Basically, greedy search is the reverse of the GBFOS method which makes the algorithm much faster. The greedy algorithm can be stated as in Algorithm 6. The greedy algorithm starts by assigning 0 bits to each sensor in the network and then increases the number of bits one bit at a time until the sum rate constraint is satisfied in R iterations. At each iteration, greedy algorithm searches the N sensors and a single bit is added to the sensor which maximizes the determinant of the resulting FIM.

Algorithm 6: Greedy Bit Allocation Algorithm

```
Set $\mathbf{R}_0 = [R_1 = 0, \ldots, R_N = 0]$ and $\mathbf{J}^0 = \mathbf{J}^P_{t+1}$.
FOR $i = 1 : R$
    (A6.1) FOR $k = 1 : N$
        Add one bit to sensor $k$ and compute $\det(\mathbf{J}^i(k))$ where
        $\mathbf{J}^i(k) = \mathbf{J}^{i-1} + \mathbf{A}_k(R_k + 1) - \mathbf{A}_k(R_k)$.
    ENDFOR
    (A6.2) Find the sensor $p^*$ for which $\det(\mathbf{J}^i(k))$ is the maximum:
    $p^* = \arg\max_k \det(\mathbf{J}^i(k))$.
    (A6.3) Increment $R_{p^*} = R_{p^*} + 1$, update $\mathbf{R}_i = [R_1, \ldots, R_{p^*}, \ldots, R_N]$ and set $\mathbf{J}^i = \mathbf{J}^i(p^*)$.
ENDFOR
```

The implementation of greedy search and its complexity can be stated as follows: At the first iteration, there are N different ways to add 1 bit. For the k-th way of adding 1 bit, $\mathbf{J}^1(k) = \mathbf{J}^P_{t+1} + \mathbf{A}_k(1)$. Then we set $\mathbf{J}^1 = \max_k \det(\mathbf{J}^1(k))$. Hence,

TABLE I
TRANSMISSION PROBABILITIES OF EACH QUANTIZATION RATE FOR $N = 9$ AND $R = 5$ AT $t = 1$ FOR THE EXAMPLE ILLUSTRATED IN FIG. 3(A)

	$m=0$	$m=1$	$m=2$	$m=3$	$m=4$	$m=5$
$i=1$	0.0000	0.0000	0.0000	0.0000	**1.0000**	0.0000
$i=2$	**0.6565**	0.0000	**0.3435**	0.0000	0.0000	0.0000
$i=3$	**1.0000**	0.0000	0.0000	0.0000	0.0000	0.0000
$i=4$	**0.8435**	0.0000	**0.1565**	0.0000	0.0000	0.0000
$i=5$	**1.0000**	0.0000	0.0000	0.0000	0.0000	0.0000
$i=6$	**1.0000**	0.0000	0.0000	0.0000	0.0000	0.0000
$i=7$	**1.0000**	0.0000	0.0000	0.0000	0.0000	0.0000
$i=8$	**1.0000**	0.0000	0.0000	0.0000	0.0000	0.0000
$i=9$	**1.0000**	0.0000	0.0000	0.0000	0.0000	0.0000

a total of N matrix summations are required at the first iteration. At the i-th iteration, for $i = 2, \ldots, R$, there are still N different ways to add 1 bit. For the k-th way of adding 1 bit, $\mathbf{J}^i(k) = \mathbf{J}^{i-1} + \mathbf{A}_k(R_k+1) - \mathbf{A}_k(R_k)$. Note that $\mathbf{A}_k(R_k)$ could be a zero matrix since R_k could be zero. Therefore, for each iteration, at most a total of $2N$ matrix summations are required. In summary, we need at most $N + (R-1)2N = N(2R-1)$ matrix summations which is an upper bound for the complexity of greedy search.

IV. Simulation Results

In this section, we illustrate the performance of different dynamic bit allocation methods with numerical examples. We evaluate the computation time of each bit allocation approach by using the "etime" function of MATLAB averaged over 100 trials. For the convex optimization based bit allocation scheme, the parameters of the barrier method are selected as $\zeta^0 = 100$, $\beta = 10$, $\epsilon_1 = 10^{-6}$, and $\epsilon_2 = 10^{-3}$. The linear programming problem in (29) is solved by using the "linprog" routine in MATLAB . In Fig. 2, the mean computation times of the considered suboptimal bit allocation schemes are compared. Since the number of summations for A-DP increases linearly with N, for large number of sensors, the computation time of A-DP is less than those of GBFOS and convex optimization where the numbers of summations increase quadratically and cubically, respectively.

We next compare the MSE performances of the bit allocation schemes based on the optimal exhaustive search, A-DP, convex optimization, GBFOS and greedy search. In addition, we analyze the MSE performance of nearest neighbor bit allocation, where all the bits are assigned to the sensor which is nearest to the predicted target location. In our simulations, we assume that N sensors are grid deployed in a $b^2 = 20m \times 20m$ surveillance area as shown in Fig. 3(a) and (b). We select $P_0 = 10^3$ and sensor observation noise $\sigma^2 = 1$. The probability density function of the target's initial state, $p(\mathbf{x}_0)$, is assumed to be Gaussian with mean $\mu_0 = [-8 \; -8 \; 2 \; 2]$ and covariance $\Sigma_0 = diag[\sigma_\theta^2 \; \sigma_\theta^2 \; 0.01 \; 0.01]$ where we select $3\sigma_\theta = 2$ so the initial point of the target remains in the ROI with very high probability. The target motion follows a white noise acceleration model and we consider two process noise parameters $\rho = 2.5 \times 10^{-3}$ and $\rho = 0.1$. Measurements are assumed to be taken at regular intervals of $\mathcal{D} = 0.5$ seconds and the observation length is 10 seconds. Namely, we perform target tracking over $T_S = 20$ time steps for each Monte Carlo trial. The number of particles used in the particle filter is $N_s = 5000$. We assume $R = 5$ bits are available at each time step for data transmission. The MSE at each time step is averaged over $T_{trials} = 500$ trials as

$$\mathrm{MSE}(t) = \frac{1}{T_{trials}} \sum_{v=1}^{T_{trials}} \left[(\mathbf{x}_t^v(1) - \hat{\mathbf{x}}_t^v(1))^2 + (\mathbf{x}_t^v(2) - \hat{\mathbf{x}}_t^v(2))^2 \right] \quad (36)$$

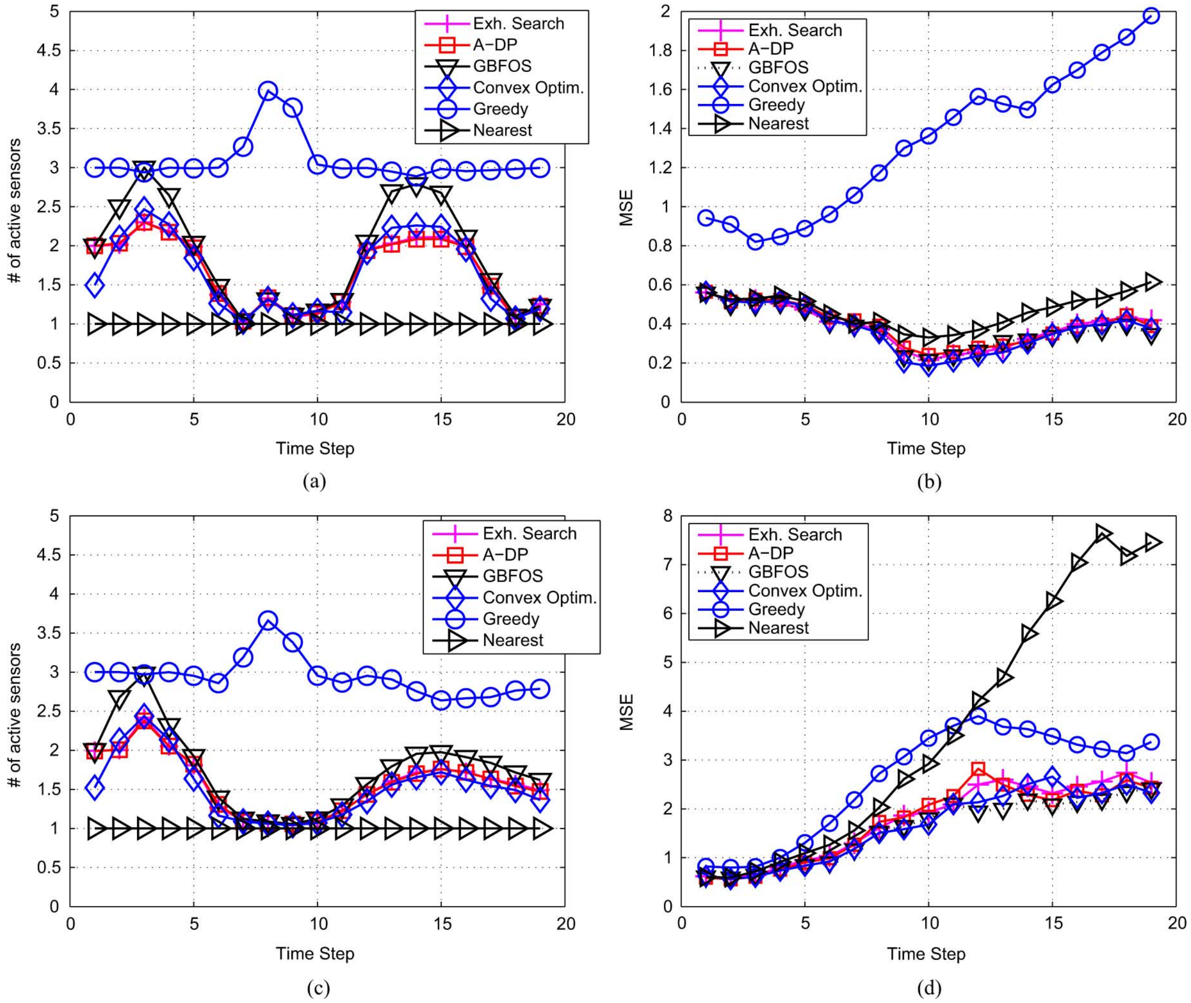
where in the v^{th} trial $\mathbf{x}_t^v$ and $\hat{\mathbf{x}}_t^v$ are the actual and estimated target states at time t, respectively.

TABLE II
MEAN AND STANDARD DEVIATION OF THE TOTAL NUMBER OF TRANSMITTED BITS USING CONVEX OPTIMIZATION BASED BIT ALLOCATION METHOD, $R = 5$

	Mean	Standard Deviation
$N = 9$, $\rho = 2.5 \times 10^{-3}$	4.9989	0.6897
$N = 9$, $\rho = 0.1$	4.9918	0.7702
$N = 25$, $\rho = 2.5 \times 10^{-3}$	4.9825	1.2494
$N = 25$, $\rho = 0.1$	4.9822	1.2915

In Fig. 3(a) and (b), a WSN is illustrated where $N = 9$ sensors track a target under the process noise parameters $\rho = 2.5 \times 10^{-3}$ and $\rho = 0.1$ respectively. For $\rho = 2.5 \times 10^{-3}$, the process noise is relatively small and the target trajectory is almost deterministic. For $\rho = 0.1$, the target trajectory has relatively large uncertainty. For the first time step of tracking, Table I presents each sensor's transmission probability for each quantization rate for the convex optimization based bit allocation scheme with $R = 5$. Note that at $t = 1$, the target is relatively close to sensor 1 located at $(-10\ m., -10\ m.)$. Then it is very likely that sensor 1 transmits its measurement using $m = 4$ bits because of the probability, $\hat{q}_{1,4} \approx 1.00$. Rest of the sensors tend to remain silent since their transmission probabilities are small. As seen in Table II, the probabilistic transmission introduces a weak constraint on the total number of bits and on the average sensors transmit R bits to the fusion center.

For $N = 9$ sensors, Fig. 4(a) and (c) show the average number of sensors activated and Fig. 4(b) and (d) show the MSE at each time step of tracking averaged over 500 Monte Carlo trials. Simulation results show that when $\rho = 2.5 \times 10^{-3}$, convex optimization, A-DP and GBFOS yield similar tracking performance to that of exhaustive search in terms of MSE. For $\rho = 2.5 \times 10^{-3}$, between the time steps 8 and 10, the target is relatively close to sensor 5 located at $(0, 0)$. Hence, using exhaustive search, A-DP, convex optimization, and GBFOS based bit allocation schemes, almost all the bits are allocated to sensor 5. When the target is not relatively close to any of the sensors, as in time steps 2–6 and 12–17, the fusion center has relatively large uncertainty about the target location, so multiple sensors are activated with relatively coarse information which increases the estimation error as shown in Fig. 4(b). After time step 17, the target approaches sensor 9 and by using exhaustive search, convex optimization, A-DP and GBFOS, all



(a)

(b)

(c)

(d)

Fig. 4. $N = 9, R = 5, T_{\mathrm{trial}} = 500$ (a) Average number of active sensors, $\rho = 2.5 \times 10^{-3}$ (b) MSE at each time step, $\rho = 2.5 \times 10^{-3}$ (c) Average number of active sensors, $\rho = 0.1$ (d) MSE at each time step, $\rho = 0.1$.

the bits are assigned to sensor 9 and then estimation error reduces again. The greedy bit allocation scheme tends to activate more sensors all the time with relatively coarse information as compared to the other bit allocation algorithms. With small process noise parameter $(\rho = 2.5 \times 10^{-3})$, nearest neighbor based bit allocation becomes more accurate than greedy search since the target trajectory is highly deterministic and there is a small uncertainty on the predicted target location. However, the tracking performance of the nearest neighbor approach is still not as good as those for exhaustive search, convex optimization, GBFOS, and A-DP. For $\rho = 0.1$, the uncertainty on target trajectory is relatively large and we observe a worse tracking performance as compared to the $\rho = 2.5 \times 10^{-3}$ case. On the other hand, still A-DP, convex optimization, and GBFOS perform equally well as exhaustive search in terms of MSE and outperform greedy search. For $\rho = 0.1$, nearest neighbor based bit allocation introduces much larger estimation errors which are sometimes even greater than those obtained by the greedy search based dynamic bit allocation scheme.

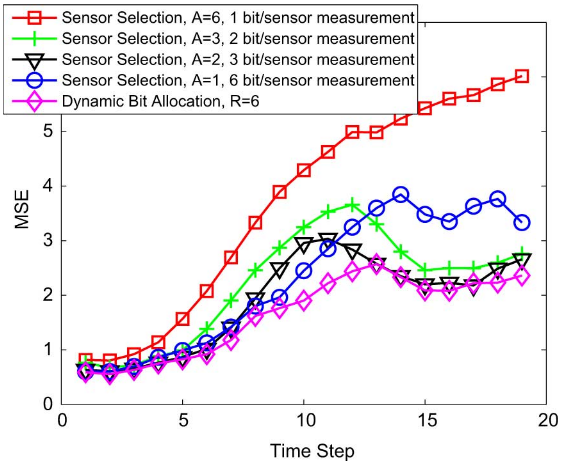
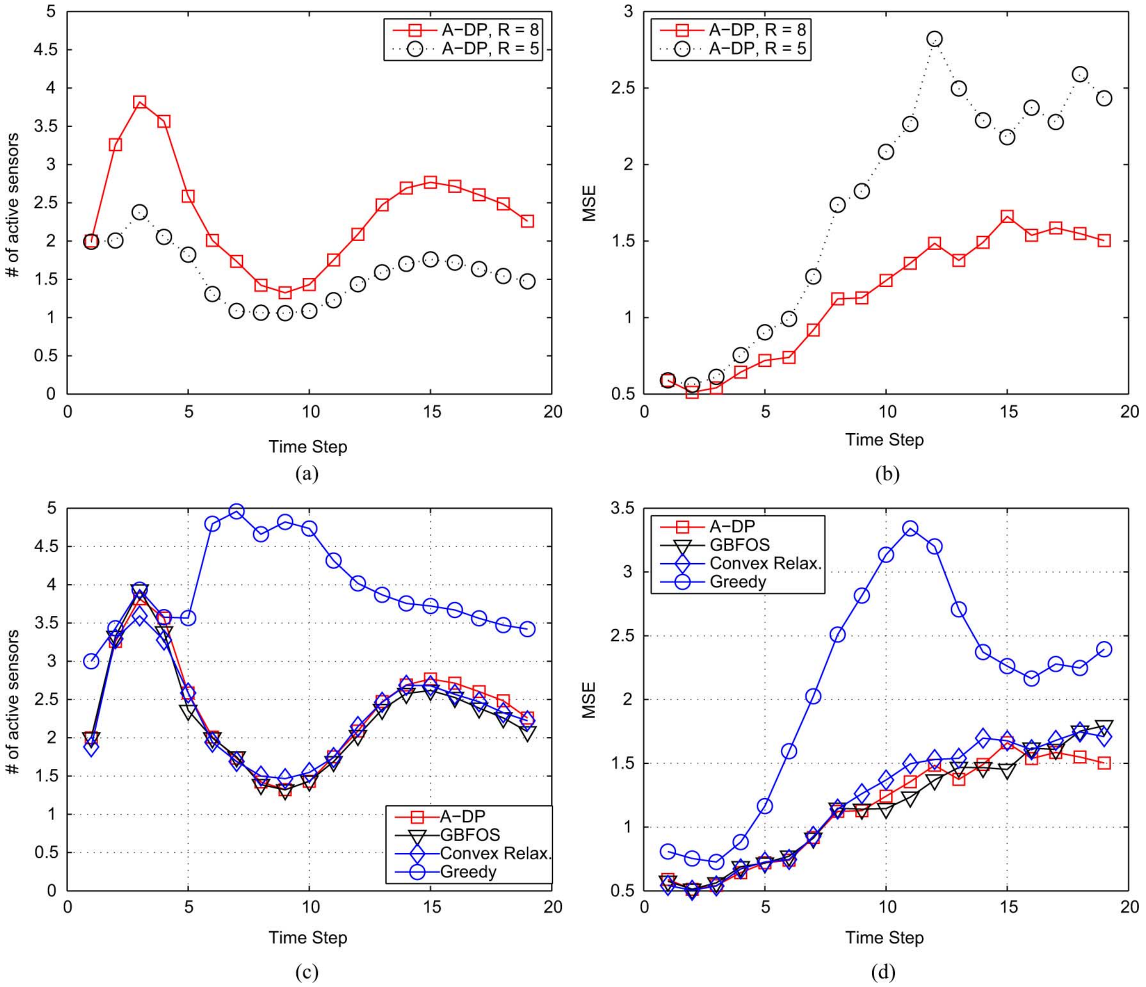


Fig. 5. MSE comparison of sensor selection and dynamic bit allocation schemes $(R = 6,\ \rho = 0.1)$.

In Fig. 5, we compare the estimation performance of the sensor selection scheme based on the method described in [1] and the bit allocation scheme using convex optimization described in this paper. We assume that the total number of bits that can be sent from sensors to the fusion center is $R = 6$. For the sensor selection problem, A sensors are selected among $N = 9$ sensors in the network. In the sensor selection examples, depending on the value of A, each selected sensor quantizes its



(a) (b) (c) (d)

Fig. 6. (a) Average number of active sensors with $R=5$ and $R=8$ with A-DP based bit allocation, $N=9$. (b) MSE at each time step with $R=5$ and $R=8$ with A-DP based bit allocation $N=9$. (c) Average number of active sensors with $N=9$ and $R=8$ (d) MSE at each time step with $N=9$ and $R=8$.

data into $M=\frac{R}{A}$ bits. Simulation results show that dynamic bit allocation uses the total number of available bits more efficiently than the sensor selection approaches with various A-M combinations.

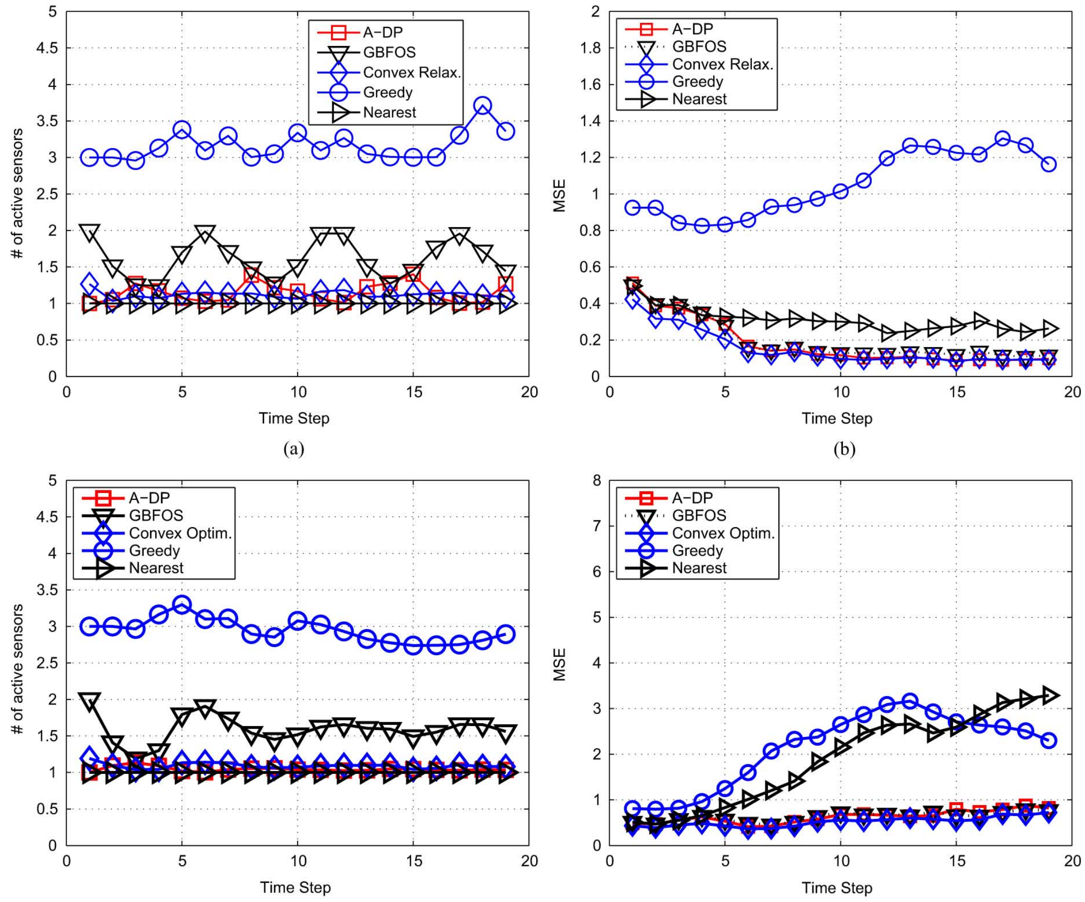
Because of (18) and (19), for large values of R, the computation of the Fisher information matrix takes a lot of time. With $\rho=0.1$, we compare the cases where $R=5$ and $R=8$ and the simulation results are shown in Fig. 6. As the intuition suggests, by increasing the available bits ($R=8$), more sensors are activated as compared to the $R=5$ case [Fig. 6(a)] and because of the more information bits, the estimation performance improves [Fig. 6(b)]. Similar to the previous results, for $R=8$ case, convex optimization, A-DP and GBFOS yield similar estimation performance and outperform the estimation performance of greedy search.

For $N=25$ sensors, Fig. 7(a) and (c) show the average number of sensors activated and Fig. 7(b) and (d) show the MSE at each time step of tracking. Since the sensor density is increased, the bit allocation schemes tend to assign all the available bits to a single sensor which has more precise information about the target. This improves the tracking performance at each time step. For $\rho=2.5\times10^{-3}$ and $\rho=0.1$ cases, convex optimization, A-DP and GBFOS yield similar estimation performances and they significantly outperform the greedy search and nearest neighbor based bit allocation approaches in terms of the MSE.

V. Conclusion

In this paper, we studied the dynamic bit allocation problem for target tracking in a WSN with quantized measurements. We proposed two bit allocation schemes which are based on convex optimization and approximate DP respectively to maximize the determinant of the FIM. Simulation results show that convex optimization, A-DP and GBFOS algorithms yield similar tracking performance, which is close to that provided by the optimal exhaustive search approach, and they outperform the greedy search and nearest neighbor approaches significantly. Using the optimal solution of the convex optimization problem as the probability of transmission at each data rate, convex optimization based bit allocation satisfies the total bit constraint on the average while the other bit allocation methods put a strict constraint on the total number of bits. In terms of computational complexity, A-DP is computationally more efficient than GBFOS and convex optimization methods especially for a large sensor network with a large N.

In this paper, we have applied the proposed approaches to a particular type of sensor network, where sensors transmit quantized signal amplitudes. Based on simulation results, we have



(a) (b) (c) (d)

Fig. 7. $N = 25, R = 5, T_{\text{trial}} = 500$. (a) Average number of active sensors $\rho = 2.5 \times 10^{-3}$. (b) MSE at each time step $\rho = 2.5 \times 10^{-3}$. (c) Average number of active sensors $\rho = 0.1$ (d) MSE at each time step, $\rho = 0.1$.

shown that the convex optimization method, the GBFOS algorithm, and the A-DP algorithm provide very similar estimation performance. However, the proposed approaches are very general and their applications are not limited to any particular type of sensor network. For example, one can apply them to sensor networks where sensors quantize their bearing readings. For different types of sensor networks, it is difficult to predict whether or not these approaches will lead to similar results without simulation experiments. Therefore, it is worthwhile to introduce all the approaches with different complexities, which may potentially lead to different tradeoffs between estimation performance and computational complexity.

In this work, we developed and compared bit allocation schemes in target tracking for one step ahead only. Our future work will cover extensions of proposed schemes to non-myopic scenarios. Multi-target tracking by dynamic bit allocation will also be considered as a future research direction.

Appendix A

It is easy to show that

$$\frac{\partial}{\partial x_t} Q\left(\frac{\eta_l^m - a_{i,t}}{\sigma}\right) = \frac{a_{i,t} n \alpha d_{i,t}^{n-2}(x_i - x_t)}{2\sqrt{2\pi\sigma^2}(1+\alpha d_{i,t}^n)} e^{-\frac{(\eta_l^m - a_{i,t})^2}{2\sigma^2}}. \tag{37}$$

Then, substituting (37) in (17), we have

$$\begin{aligned} E\left[-\frac{\partial^2}{\partial x_t^2} p(D_{i,t} = l|\mathbf{x}_t, R_{i,t} = m)\right] &= \left(\sum_{l=0}^{2^m-1} \frac{\left[e^{-\frac{(\eta_l^m - a_{i,t})^2}{2\sigma^2}} - e^{-\frac{(\eta_{l+1}^m - a_{i,t})^2}{2\sigma^2}}\right]^2}{8\pi\sigma^2 p(D_{i,t} = l|\mathbf{x}_t, R_{i,t} = m)}\right) \\ &\quad \times \frac{a_{i,t}^2 n^2 \alpha^2 d_{i,t}^{2n-4}(x_i - x_t)^2}{(1+\alpha d_{i,t}^n)^2} \\ &= \kappa_{i,t}(m, x_i, y_i, x_t, y_t) \frac{a_{i,t}^2 n^2 \alpha^2 d_{i,t}^{2n-4}}{(1+\alpha d_{i,t}^n)^2}(x_i - x_t)^2. \end{aligned} \tag{38}$$

Due to the symmetry between elements x_t and y_t

$$\begin{aligned} E\left[-\frac{\partial^2}{\partial y_t^2} p(D_{i,t} = l|\mathbf{x}_t, R_{i,t} = m)\right] \\ = \kappa_{i,t}(m, x_i, y_i, x_t, y_t) \frac{a_{i,t}^2 n^2 \alpha^2 d_{i,t}^{2n-4}}{(1+\alpha d_{i,t}^n)^2}(y_i - y_t)^2 \end{aligned} \tag{39}$$

and

$$E\left[-\frac{\partial^2}{\partial x_t \partial y_t} p(D_{i,t}=l|\mathbf{x}_t, R_{i,t}=m)\right] = \kappa_{i,t}(m, x_i, y_i, x_t, y_t)\frac{a_{i,t}^2 n^2 \alpha^2 d_{i,t}^{2n-4}}{(1+\alpha d_{i,t}^n)^2}(x_i - x_t)(y_i - y_t) \quad (40)$$

Appendix B

The $(i,m)^{th}$ element of the gradient vector is,

$$(\nabla_{\hat{q}_{t+1}}\phi)_{i,m} = \frac{\partial}{\partial \hat{q}_{i,m}}\left\{ -\log\det\left(\sum_{m=0}^{R}\sum_{i=1}^{N}\hat{q}_{i,m}\mathbf{J}_{i,t+1}^{D}(m) + \mathbf{J}_{t+1}^{P}\right) - \frac{1}{\zeta}\sum_{m=0}^{R}\sum_{i=1}^{N}\left(\log\left(\hat{q}_{i,m}\right) + \log\left(1-\hat{q}_{i,m}\right)\right)\right\}. \quad (41)$$

Let $\mathbf{X}$ be an invertible matrix and x be a scalar. Using the property

$$\frac{\partial \log\det\left(\mathbf{X}\right)}{\partial x} = tr\left\{\mathbf{X}^{-1}\frac{\partial \mathbf{X}}{\partial x}\right\}$$

and the definition

$$\mathbf{W} \triangleq \left(\sum_{m=0}^{R}\sum_{i=1}^{N}\hat{q}_{i,m}\mathbf{J}_{i,t+1}^{D}(m) + \mathbf{J}_{t+1}^{P}\right)$$

each element of the gradient vector is obtained as

$$(\nabla_{\hat{q}_{t+1}}\phi)_{i,m}(\zeta) = -tr\{\mathbf{W}^{-1}\mathbf{J}_{i,t+1}^{D}(m)\} - \frac{1}{\zeta\hat{q}_{i,m}} + \frac{1}{\zeta(1-\hat{q}_{i,m})}. \quad (42)$$

In order to compute $\nabla_{\hat{q}_{t+1}}^{\hat{q}_{t+1}}\phi_{\hat{\mathbf{q}}_{t+1}}(\zeta)$, for $i, i^* \in \{1,2,\ldots,N\}$ and $m, m^* \in \{0,1,\ldots,R\}$, we define

$$\boldsymbol{\psi}_{(i^*,m^*),(i,m)} \triangleq \frac{\partial}{\partial \hat{q}_{i^*,m^*}}\left\{-tr(\mathbf{W}^{-1}\mathbf{J}_{i,t+1}^{D}(m))\right\}. \quad (43)$$

Using the properties

$$\frac{\partial}{\partial x}tr\{\mathbf{X}\} = tr\left\{\frac{\partial \mathbf{X}}{\partial x}\right\}$$

and

$$\frac{\partial \mathbf{X}^{-1}}{\partial x} = -\mathbf{X}^{-1}\frac{\partial \mathbf{X}}{\partial x}\mathbf{X}^{-1}$$

we get

$$\boldsymbol{\psi}_{(i^*,m^*),(i,m)} = tr\{\mathbf{W}^{-1}\mathbf{J}_{i^*,t+1}^{D}(m^*)\mathbf{W}^{-1}\mathbf{J}_{i,t+1}^{D}(m)\}.$$

Finally the Hessian matrix is obtained as

$$(\nabla_{\hat{q}_{t+1}}^{\hat{q}_{t+1}}\phi(\zeta)) = \boldsymbol{\psi} + \frac{1}{\zeta}\,diag\left(\frac{1}{\hat{q}_{1,0}^2} + \frac{1}{(1-\hat{q}_{1,0})^2};\ldots;\frac{1}{\hat{q}_{N,R}^2} + \frac{1}{(1-\hat{q}_{N,R})^2}\right) \quad (44)$$

References

[1] S. Joshi and S. Boyd, "Sensor selection via convex optimization," *IEEE Trans. Signal Process.*, vol. 57, no. 2, pp. 451–462, Feb. 2009.

[2] F. Zhao, J. Shin, and J. Reich, "Information-driven dynamic sensor collaboration," *IEEE Signal Process. Mag.*, vol. 19, no. 2, pp. 61–72, Mar. 2002.

[3] G. M. Hoffmann and C. J. Tomlin, "Mobile sensor network control using mutual information methods and particle filters," *IEEE Trans. Autom. Control*, vol. 55, no. 1, pp. 32–47, Jan. 2010.

[4] J. Williams, J. Fisher, and A. Willsky, "Approximate dynamic programming for communication-constrained sensor network management," *IEEE Trans. Signal Process.*, vol. 55, no. 8, pp. 4300–4311, Aug. 2007.

[5] C. Kreucher, A. Hero, K. Kastella, and M. Morelande, "An information-based approach to sensor management in large dynamic networks," *IEEE Proc.*, vol. 95, no. 5, pp. 978–999, May 2007.

[6] H. L. V. Trees, *Detection, Estimation and Modulation Theory, Part I*. New York: Wiley, 2001.

[7] M. Hernandez, T. Kirubarajan, and Y. Bar-Shalom, "Multisensor resource deployment using posterior Cramér-Rao bounds," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 40, no. 2, pp. 399–416, Apr. 2004.

[8] E. Masazade, R. Niu, P. K. Varshney, and M. Keskinoz, "Energy aware iterative source localization schemes for wireless sensor networks," *IEEE Trans. Signal Process.*, vol. 58, no. 9, pp. 4824–4835, Sep. 2010.

[9] L. Zuo, R. Niu, and P. K. Varshney, "Posterior CRLB based sensor selection for target tracking in sensor networks," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2007, vol. 2, pp. II–1041.

[10] L. Zuo, R. Niu, and P. K. Varshney, "A sensor selection approach for target tracking in sensor networks with quantized measurements," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Apr. 2008, pp. 2521–2524.

[11] Y. Hou, Y. Shi, and H. Sherali, "Rate allocation and network lifetime problems for wireless sensor networks," *IEEE/ACM Trans. Netw.*, vol. 16, no. 2, pp. 321–334, Apr. 2008.

[12] X. Zhang, L.-L. Xie, and X. Shen, "An energy-efficient bit allocation scheme in wireless sensor networks," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, Dec. 2010, pp. 1–5.

[13] Q. Cheng, P. K. Varshney, K. G. Mehrotra, and C. K. Mohan, "Bandwidth management in distributed sequential detection," *IEEE Trans. Inf. Theory*, vol. 51, no. 8, pp. 2954–2961, 2005.

[14] O. Ozdemir, R. Niu, and P. K. Varshney, "Dynamic bit allocation for target tracking in sensor networks with quantized measurements," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2010, pp. 2906–2909.

[15] Y. Bar-Shalom, P. K. Willett, and X. Tian, *Tracking and Data Fusion: A Handbook of Algorithms*. Storrs, CT: YBS Publishing, 2011.

[16] E. Riskin, "Optimal bit allocation via the generalized bfos algorithm," *IEEE Trans. Inf. Theory*, vol. 37, no. 2, pp. 400–402, Mar. 1991.

[17] Y. Kim and A. Ortega, "Quantizer design and distributed encoding algorithm for source localization in sensor networks," in *Proc. 4th Int. Symp. Inf. Process. Sensor Netw. (IPSN 2005)*, Apr. 2005, pp. 231–238.

[18] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[19] E. Masazade, R. Niu, and P. K. Varshney, "Dynamic bandwidth allocation for target tracking in wireless sensor networks," in *Proc. 14th Int. Conf. Inf. Fusion (FUSION'11)*, Jul. 2011.

[20] D. P. Bertsekas, *Dynamic Programming and Optimal Control, vol. I and II*. New York: Athena Scientific, 2007.

[21] R. Niu and P. K. Varshney, "Distributed detection and fusion in a large wireless sensor network of random size," *EURASIP J. Wireless Commun. Netw.*, vol. 2005, no. 4, pp. 462–472, 2005.

[22] R. Niu and P. K. Varshney, "Target location estimation in sensor networks with quantized data," *IEEE Trans. Signal Process.*, vol. 54, no. 12, pp. 4519–4528, Dec. 2006.

[23] Y. Ruan, P. Willett, A. Marrs, F. Palmieri, and S. Marano, "Practical fusion of quantized measurements via particle filtering," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 44, no. 1, pp. 15–29, Jan. 2008.

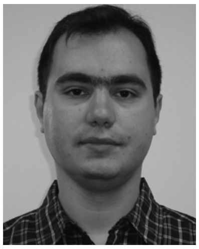
[24] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–188, Feb. 2002.

[25] O. Ozdemir, R. Niu, P. K. Varshney, and A. Drozd, "Modified Bayesian Cramér-Rao lower bound for nonlinear tracking," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2011.

[26] G. Hämmerlin and K. Hoffmann, *Numerical Mathematics*. New York: Springer-Verlag, 1991.

[27] E. Masazade, R. Niu, P. K. Varshney, and M. Keskinoz, "A probabilistic transmission scheme for distributed estimation in wireless sensor networks," in *Proc. 44th Ann. Conf. Inf. Sci. Syst. (CISS'10)*, Mar. 2010, pp. 1–6.

[28] K. B. Petersen and M. S. Pedersen, The Matrix Cookbook, version 20081110, Oct. 20081110 [Online]. Available: http://www2.imm.dtu.dk/pubdb/p.php?3274



Engin Masazade (S'03–M'10) received the B.S. degree from the Electronics and Communications Engineering Department, Istanbul Technical University, Turkey, in 2003. He received the M.S. and Ph.D. degrees from Sabanci University, Electronics Engineering Program, Istanbul, in 2006 and 2010, respectively.

Currently, he is a Postdoctoral Research Associate with the Department of Electrical Engineering and Computer Science, Syracuse University, Syracuse, NY. His research interests include distributed detection, localization, and tracking for wireless sensor networks, dynamic resource management in sensor/communication networks.

Ruixin Niu (M'04–SM'11) received the B.S. degree from Xian Jiaotong University, Xian, China, in 1994, the M.S. degree from the Institute of Electronics, Chinese Academy of Sciences, Beijing, in 1997, and the Ph.D. degree from the University of Connecticut, Storrs, in 2001, all in electrical engineering.

He is currently an Assistant Professor with the Department of Electrical and Computer Engineering, Virginia Commonwealth University (VCU), Richmond. Before joining VCU, he was a Research Assistant Professor with Syracuse University, Syracuse, NY. His research interests are in the areas of statistical signal processing and its applications, including detection, estimation, data fusion, sensor networks, communications, and image processing.

Dr. Niu received the Best Paper Award at the Seventh International Conference on Information Fusion in 2004. He is a coauthor of the paper that won the Best Student Paper Award at the Thirteenth International Conference on Information Fusion in 2010. He is an Associate Editor of the IEEE TRANSACTIONS ON SIGNAL PROCESSING and the Associate Administrative Editor of the *Journal of Advances in Information Fusion*. He was an Associate Editor of the *International Journal of Distributed Sensor Networks* between 2010 and 2012.

Pramod K. Varshney (S'72–M'77–SM'82–F'97) was born in Allahabad, India, on July 1, 1952. He received the B.S. degree in electrical engineering and computer science (with highest honors), and the M.S. and Ph.D. degrees in electrical engineering from the University of Illinois at Urbana-Champaign in 1972, 1974, and 1976 respectively.

From 1972 to 1976, he held teaching and research assistantships at the University of Illinois. Since 1976, he has been with Syracuse University, Syracuse, NY, where he is currently a Distinguished Professor of Electrical Engineering and Computer Science and the Director of CASE: Center for Advanced Systems and Engineering. He served as the Associate Chair of the department from 1993 to 1996. He is also an Adjunct Professor of Radiology at Upstate Medical University, Syracuse. His current research interests are in distributed sensor networks and data fusion, detection and estimation theory, wireless communications, image processing, radar signal processing, and remote sensing. He has published extensively. He is the author of *Distributed Detection and Data Fusion* (New York: Springer-Verlag, 1997). He has served as a consultant to several major companies.

Dr. Varshney was a James Scholar, a Bronze Tablet Senior, and a Fellow while with the University of Illinois. He is a member of Tau Beta Pi and is the recipient of the 1981 ASEE Dow Outstanding Young Faculty Award. He was elected to the grade of Fellow of the IEEE in 1997 for his contributions in the area of distributed detection and data fusion. He was the Guest Editor of the Special Issue on Data Fusion of the PROCEEDINGS OF THE IEEE, January 1997. In 2000, he received the Third Millennium Medal from the IEEE and Chancellor's Citation for exceptional academic achievement at Syracuse University. He is the recipient of the IEEE 2012 Judith A. Resnik Award. He serves as a Distinguished Lecturer for the IEEE Aerospace and Electronic Systems (AES) Society. He is on the Editorial Board of the *Journal on Advances in Information Fusion*. He was the President of International Society of Information Fusion during 2001.