DECENTRALIZED COMPUTATION OF THE CONDITIONAL POSTERIOR CRAMÉR-RAO LOWER BOUND: APPLICATION TO ADAPTIVE SENSOR SELECTION

Arash Mohammadi and Amir Asif

Computer Science and Engineering
York University, Toronto, ON, Canada M3J 1P3
Email:{marash, asif}@cse.yorku.ca

ABSTRACT

Motivated by the problem of adaptive resource management in decentralized sensor networks, the paper derives an algorithm for the distributed computation of the *conditional* posterior Cramér-Rao lower bound (PCRLB) for nonlinear tracking applications as an alternative to the *non-conditional* (conventional) PCRLB. Using the proposed conditional bound, a decentralized adaptive sensor-selection algorithm is then developed with the objective of dynamically activating a subset of observation nodes to optimize the network's performance. Our Monte Carlo simulations verify the superiority of the proposed decentralized PCRLB based sensor selection approach in bearing only tracking applications over its conventional counterparts.

Index Terms— Data fusion, Distributed estimation, Multisensor tracking, Particle filters, and Sensor Selection.

1. INTRODUCTION

Adaptive sensor resource management is an important task in geographically distributed sensor networks, where power and bandwidth limitations constraint the number of sensors active at a particular time. This paper focuses on decentralized networks [1], where there is no central fusion node and tracking is performed in a distributed fashion across the network. We derive a decentralized algorithm for computing the *conditional* PCRLB for tracking sensor networks as an alternative to the *non-conditional* (conventional) PCRLB. Using the proposed bound, we develop a decentralized sensor-selection algorithm and apply it for tracking targets in large sensor networks.

The conventional PCRLB [2] considers observations and state variables as random. Consequently, the conventional PCRLB is determined primarily from the state model, observation model, and prior knowledge of the initial state of the system leading to an *offline* bound that makes it inefficient for adaptive resource management. An alternative is the conditional PCRLB [3], which is a function of the past history of observations. The online conditional PCRLB leads to a more accurate representation of the current systems's performance and is, therefore, a better criteria for adaptive sensor-selection. Existing conditional PCRLB expressions [3]-[5] are, however, limited to centralized architectures utilizing a fusion centre, which make them inappropriate for the decentralized topologies. The paper derives the decentralized PCRLB (dPCRLB) and computes it distributively without requiring a fusion algorithm.

Prior Work: Decentralized adaptive sensor selection arises in several applications, e.g., cellular networks [20], decentralized tracking in wireless Ad hoc sensor networks [21], robotic localization and underwater acoustics [22]. In principle, sensor selection [6]-[10] is a stochastic problem that involves optimization of a pre-defined cost function. Reference [11], for example, minimizes the uncertainty ellipsoid function associated with every possible sensor configurations using convex optimization. Other cost functions considered previously are based on the mean square error (MSE) [12] of the state estimates and overall entropy of the system [13]. For adaptive sensor resource management, the PCRLB [14]-[19] has been recently shown as an effective criteria for selecting sensors. The PCRLB provides a near-optimal bound of the achievable tracker's performance and can be calculated predictively. Further, the PCRLB is independent and not constrained by the estimation methodology employed. Existing PCRLB-based selection techniques are, however, limited to centralized and hierarchical architectures [16], and when extended to distributed/decentralized topologies use suboptimal expressions [15] for computing the PCRLB. Our previous work [23, 24] improves on [15] by deriving the exact expression for computing the non-conditional dPCRLB distributively for full-order [23] and reduced-order [24] decentralized state estimation. In [14], we have proposed a dPCRLB-based sensor selection algorithm that uses conditional PCRLB as a selection criteria. This paper couples our previous work by extending our non-conditional dPCRLB framework [23, 24] to conditional PCRLB and applies the later to full-order adaptive sensor selection in tracking applications. Through Monte Carlo simulations, we verify the superiority of the conditional dPCRLB sensor selection over its existing counterparts including the non-conditional PCRLB.

The rest of the paper is organized as follows. In Section 2, we formulate the problem and review non-conditional dPCRLB. Section 3 derives the conditional dPCRLB, while the conditional dPCRLB based sensor selector is presented in Section 4. The effectiveness of the proposed conditional dPCRLB in tracking applications is illustrated through simulations in Section 5. Section 6 then concludes the paper.

2. PROBLEM FORMULATION

A decentralized network consisting of N_f processing nodes (local fusion nodes) and N_s sensor nodes, and observing a set of n_x state variables based on the following state-space model is considered

$$\text{State model:} \qquad \boldsymbol{x}(k) = \boldsymbol{f}(\boldsymbol{x}(k-1)) + \boldsymbol{\xi}(k) \tag{1}$$

$$\text{Observation model:} \quad \underbrace{\begin{bmatrix} \boldsymbol{z}^{(1)}(k) \\ \vdots \\ \boldsymbol{z}^{(N_s)}(k) \end{bmatrix}}_{\boldsymbol{z}(k)} = \underbrace{\begin{bmatrix} \boldsymbol{g}^{(1)}(\boldsymbol{x}(k)) \\ \vdots \\ \boldsymbol{g}^{(N_s)}(\boldsymbol{x}(k)) \end{bmatrix}}_{\boldsymbol{g}(\boldsymbol{x}(k))} + \underbrace{\begin{bmatrix} \boldsymbol{\zeta}^{(1)}(k) \\ \vdots \\ \boldsymbol{\zeta}^{(N_s)}(k) \end{bmatrix}}_{\boldsymbol{\zeta}(k)}, \tag{2}$$

where $\boldsymbol{z}^{(m)}(k)$ is the measurement made by sensor node m at time instant k, $(1 \leq k)$, for $(1 \leq m \leq N_s)$. Terms $\boldsymbol{\xi}(\cdot)$ and $\boldsymbol{\zeta}(\cdot)$ are, respectively, the global uncertainties in the process and observation models. The local fusion nodes are modeled as vertices of the fusion graph $\boldsymbol{\mathcal{G}}_f = (\boldsymbol{\nu}_f, \boldsymbol{\mathcal{E}}_f)$, while the sensor nodes are modeled as the vertices of the observation graph $\boldsymbol{\mathcal{G}}_s = (\boldsymbol{\nu}_s, \boldsymbol{\mathcal{E}}_s)$. The edge set $\boldsymbol{\mathcal{E}}_s \subseteq \boldsymbol{\nu}_s \times \boldsymbol{\nu}_f$ represents the sensor to fusion node communication constraints, i.e., if observation node m_l can send information to fusion node l then $(m_l, l) \in \boldsymbol{\mathcal{E}}_s$. The edge set $\boldsymbol{\mathcal{E}}_f \subseteq \boldsymbol{\nu}_f \times \boldsymbol{\nu}_f$ represents the fusion communication constraints, i.e., if fusion node q can communicate with fusion node i then $(q, i) \in \boldsymbol{\mathcal{E}}_f$. We also define a fusion-to-fusion neighbourhood $\aleph_{\text{fuse}}^{(l)}$ that includes the set of fusion nodes $m \neq l$ connected to fusion node l. In the decentralized sensor selection scenario, each local fusion node can communicate only with sensors and other fusion nodes within its immediate neighbourhood referred to as the local surveillance region. Due to physical limitations, only a maximum number N_{sl} of sensors can be activated by fusion node l at each iteration. The total number $N_{\max}$ of simultaneously active sensors is also limited, i.e., $\sum_{l=1}^{N_f} N_{sl} \leq N_{\max} < N_s$. The entire state $\boldsymbol{x}(k)$ is estimated by running localized filter at fusion node l, for $(1 \leq l \leq N_f)$, with the reduced state-space model

$$\boldsymbol{x}(k) \quad = \quad \boldsymbol{f}(\boldsymbol{x}(k-1)) + \boldsymbol{\xi}(k) \tag{3}$$

$$\boldsymbol{z}^{(\aleph_{\text{obs}}^{(l)})}(k) \quad = \quad \boldsymbol{g}^{(\aleph_{\text{obs}}^{(l)})}(\boldsymbol{x}(k)) + \boldsymbol{\zeta}^{(\aleph_{\text{obs}}^{(l)})}(k), \tag{4}$$

where local observations are restricted to $\boldsymbol{z}^{(\aleph_{\text{obs}}^{(l)})}(k)$ collected from the sensors in the neighbourhood $\aleph_{\text{obs}}^{(l)}$ selected by fusion node l. Since $\aleph_{\text{obs}}^{(l)}$ varies with time, dimension of $\boldsymbol{z}^{(\aleph_{\text{obs}}^{(l)})}(k)$ is also time-varying. The following subsection reviews key concepts from [3, 26] needed to understand the rest of the paper.

A. Posterior Cramér Rao Lower Bound (PCRLB)

The conventional PCRLB inequality [2] lower bounds the mean square error (MSE) of the estimate $\hat{\boldsymbol{x}}(0\!:\!k\!+\!1)$ to

$$\mathbb{E}\{(\hat{\boldsymbol{x}}(0\!:\!k)-\boldsymbol{x}(0\!:\!k))(\hat{\boldsymbol{x}}(0\!:\!k)-\boldsymbol{x}(0\!:\!k))^T\} \geq [\boldsymbol{J}(0\!:\!k)]^{-1}, \tag{5}$$

where the FIM $\boldsymbol{J}(0\!:\!k)$ is defined as

$$\boldsymbol{J}(0\!:\!k) = \mathbb{E}_{P(\boldsymbol{x}(0:k),\boldsymbol{z}(1:k))}\{-\Delta_{\boldsymbol{x}(0:k)}^{\boldsymbol{x}(0:k)} \log P(\boldsymbol{x}(0\!:\!k)|\boldsymbol{z}(1\!:\!k))\}. \tag{6}$$

Notation $\Delta_{\boldsymbol{x}(k-1)}^{\boldsymbol{x}(k)} = \nabla_{\boldsymbol{x}(k-1)} \nabla_{\boldsymbol{x}(k)}^T$ denotes the second order partial derivative with the first order partial derivative $\nabla_{\boldsymbol{x}(k)} = [\frac{\partial}{\partial X_1(k)}, \ldots, \frac{\partial}{\partial X_{n_x}(k)}]^T$. The FIM $\boldsymbol{J}(k)$ associated with the estimate $\boldsymbol{x}(k)$ is obtained from the inverse of the $(n_x \times n_x)$ right-lower square block of the inverse of $\boldsymbol{J}(0:k)$. Proposition 1, given below, computes the global FIM at time $k\!+\!1$ as a function of local filtering FIM $\boldsymbol{J}^{(l)}(\boldsymbol{x}(k\!+\!1))$ and local predictive FIM $\boldsymbol{J}^{(l)}(\boldsymbol{x}(k\!+\!1|k))$.

Proposition 1. *The sequence $\{\boldsymbol{J}(\boldsymbol{x}(k))\}$ corresponding to the global information submatrix (dPCRLB) follows the recursion*

$$\boldsymbol{J}(\boldsymbol{x}(k\!+\!1)) = \boldsymbol{C}^{22}(k) - \boldsymbol{D}^{21}(k)\big(\boldsymbol{J}(\boldsymbol{x}(k)) + \boldsymbol{D}^{11}(k)\big)^{-1}\boldsymbol{D}^{12}(k) \tag{7}$$

where $\boldsymbol{D}^{11}(k)$, $\boldsymbol{D}^{21}(k)$, $\boldsymbol{D}^{12}(k)$, and $\boldsymbol{C}^{22}(k)$ are given by

$$\boldsymbol{D}^{11}(k) = \mathbb{E}\{-\Delta_{\boldsymbol{x}(k)}^{\boldsymbol{x}(k)} \log P(\boldsymbol{x}(k+1)|\boldsymbol{x}(k))\} \tag{8}$$

$$\boldsymbol{D}^{12}(k) = (\boldsymbol{D}^{21}(k))^T = \mathbb{E}\{-\Delta_{\boldsymbol{x}(k)}^{\boldsymbol{x}(k+1)} \log P(\boldsymbol{x}(k+1)|\boldsymbol{x}(k))\} \tag{9}$$

$$\boldsymbol{C}^{22}(k\!+\!1) \approx \sum_{l=1}^{N_f} \boldsymbol{J}^{(l)}(\boldsymbol{x}(k\!+\!1)) - \sum_{l=1}^{N_f} \boldsymbol{J}^{(l)}(\boldsymbol{x}(k\!+\!1|k)) + \mathbb{E}\{-\Delta_{\boldsymbol{x}(k+1)}^{\boldsymbol{x}(k+1)} \log P(\boldsymbol{x}(k+1)|\boldsymbol{x}(k))\}. \tag{10}$$

We proved Proposition 1 in [26]. Note that the expectations in Proposition 1 are with respect to $P(\boldsymbol{x}(0:k), \boldsymbol{z}(1:k))$.

Next, the auxiliary FIM is defined in terms of the posterior distribution $P_a(k) \triangleq P(\boldsymbol{x}(0\!:\!k)|\boldsymbol{z}(1\!:\!k))$ as follows

$$\boldsymbol{J}_{\text{AUX}}(0\!:\!k) \triangleq \mathbb{E}_{P_a(k)}\{-\Delta_{\boldsymbol{x}(0:k)}^{\boldsymbol{x}(0:k)} \log P_a(k)\}. \tag{11}$$

Reference [3] has derived centralized recursive expressions for computing $\boldsymbol{J}_{\text{AUX}}(k)$ (the inverse of $(n_x \times n_x)$ right-lower square block of the inverse of $\boldsymbol{J}_{\text{AUX}}(0\!:\!k)$). Alternatively, $\boldsymbol{J}_{\text{AUX}}(k)$ can be derived using Proposition 1 except for the expectation, which is now taken with respect to $P_a(k)$. The conditional PCRLB provides a bound on the performance of estimating $\boldsymbol{x}(0:k)$ given that the past observations $\boldsymbol{z}(1:k-1)$ are known [3]. The conditional MSE of the state vector is lower bounded by

$$\mathbb{E}_{P_c(k)}\{\hat{\boldsymbol{x}}(0\!:\!k) - \boldsymbol{x}(0\!:\!k))(\hat{\boldsymbol{x}}(0\!:\!k) - \boldsymbol{x}(0\!:\!k))^T\} \geq [\boldsymbol{I}(0\!:\!k)]^{-1}, \tag{12}$$

where $P_c(k) \triangleq P(\boldsymbol{x}(0:k), \boldsymbol{z}(k)|\boldsymbol{z}(1:k-1))$ and $\boldsymbol{I}(0:k) \triangleq \mathbb{E}_{P_c(k)}\{-\Delta_{\boldsymbol{x}(0:k)}^{\boldsymbol{x}(0:k)} \log P_c(k)\}$. Proposition 1 [3] updates the conditional FIM denoted by $\boldsymbol{L}(k+1)$ (the inverse of the $(n_x \times n_x)$ right-lower block of $[\boldsymbol{I}(0\!:\!k\!+\!1)]^{-1}$) for estimating $\boldsymbol{x}(k\!+\!1)$.

Proposition 2. *The centralized conditional FIM $\boldsymbol{L}(k+1)$ associated with the filtering estimate $\hat{\boldsymbol{x}}(k\!+\!1)$ follows the recursion*

$$\boldsymbol{L}(k\!+\!1) = \boldsymbol{B}^{22}(k) - \boldsymbol{B}^{21}(k)\big(\boldsymbol{J}_{AUX}(k) + \boldsymbol{B}^{11}(k)\big)^{-1}\boldsymbol{B}^{12}(k), \tag{13}$$

where $$\boldsymbol{B}^{11}(k) = \mathbb{E}_{P_c(k+1)}\{-\Delta_{\boldsymbol{x}(k)}^{\boldsymbol{x}(k)} \log P(\boldsymbol{x}(k+1)|\boldsymbol{x}(k))\}, \tag{14}$$

$$\boldsymbol{B}^{12}(k) = [\boldsymbol{B}^{21}(k)]^T = \mathbb{E}_{P_c(k+1)}\{-\Delta_{\boldsymbol{x}(k)}^{\boldsymbol{x}(k+1)} \log P(\boldsymbol{x}(k\!+\!1)|\boldsymbol{x}(k))\}, \tag{15}$$

$$\boldsymbol{B}^{22}(k) = \mathbb{E}_{P_c(k+1)}\{-\Delta_{\boldsymbol{x}(k+1)}^{\boldsymbol{x}(k+1)} \log P(\boldsymbol{x}(k+1)|\boldsymbol{x}(k))\} + \mathbb{E}_{P_c(k+1)}\{-\Delta_{\boldsymbol{x}(k+1)}^{\boldsymbol{x}(k+1)} \log P(\boldsymbol{z}(k\!+\!1)|\boldsymbol{x}(k\!+\!1))\}. \tag{16}$$

where $P_c(k\!+\!1) \triangleq P(\boldsymbol{x}(0\!:\!k\!+\!1), \boldsymbol{z}(k\!+\!1)|\boldsymbol{z}(1\!:\!k))$.

3. DECENTRALIZED CONDITIONAL DPCRLB

This section presents the decentralized algorithm for computing the global conditional FIM from the local conditional FIMs. The local conditional FIMs are defined first.

Definition 1. *The local conditional FIM $\boldsymbol{I}^{(l)}(0\!:\!k+1)$ corresponding to the local estimate $\hat{\boldsymbol{x}}^{(l)}(0\!:\!k\!+\!1)$, for $(1 \leq l \leq N)$, is defined as*

$$\boldsymbol{I}^{(l)}(0\!:\!k\!+\!1) \triangleq \mathbb{E}_{P_c^{(l)}(k+1)}\{-\Delta_{\boldsymbol{x}(0:k+1)}^{\boldsymbol{x}(0:k+1)} \log P_c^{(l)}(k\!+\!1)\}, \tag{17}$$

where $P_c^{(l)}(k+1) \triangleq P(\boldsymbol{x}(0:k+1), \boldsymbol{z}^{(l)}(k+1)|\boldsymbol{z}^{(l)}(1:k))$. The local bound $\boldsymbol{L}^{(l)}(k\!+\!1)$ on $\hat{\boldsymbol{x}}^{(l)}(k\!+\!1)$, is given by the inverse of the $(n_x \times n_x)$ right-lower block of $[\boldsymbol{I}^{(l)}(0\!:\!k\!+\!1)]^{-1}$.

Definition 2. *The local predictive conditional FIM $\boldsymbol{I}^{(l)}(0\!:\!k\!+\!1|k)$ is defined as follows*

$$\boldsymbol{I}^{(l)}(0\!:\!k\!+\!1|k) \triangleq \mathbb{E}_{P_p^{(l)}(k+1)}\{-\Delta_{\boldsymbol{x}(0:k+1)}^{\boldsymbol{x}(0:k+1)} \log P_p^{(l)}(k+1)\}, \tag{18}$$

where $P_p^{(l)}(k+1) \triangleq P(\boldsymbol{x}(0:k+1)|\boldsymbol{z}^{(l)}(1:k))$. The local bound $\boldsymbol{L}^{(l)}(k\!+\!1|k)$ on $\hat{\boldsymbol{x}}^{(l)}(k\!+\!1|k)$ is given by the inverse of the $(n_x \times n_x)$ right-lower block of $[\boldsymbol{I}^{(l)}(0\!:\!k\!+\!1|k)]^{-1}$. Note that Proposition 1 can be used to compute both $\boldsymbol{L}^{(l)}(k\!+\!1)$ and $\boldsymbol{L}^{(l)}(k\!+\!1|k)$ with relevant local distributions replacing the global ones. Theorem 1 provides a recursive expression for computing the overall conditional FIM as a function of local FIMs.

Theorem 1. *The sequence $\{\boldsymbol{L}(\boldsymbol{x}(k))\}$ of the global information sub-matrices follows the recursion*

$$\boldsymbol{L}(k+1)=\boldsymbol{C}^{22}(k)-\boldsymbol{C}^{21}(k)\Big(\boldsymbol{J}_{AUX}(k)+\boldsymbol{C}^{11}(k)\Big)^{-1}\boldsymbol{C}^{12}(k) \quad (19)$$

$$\boldsymbol{C}^{11}(k)=\mathbb{E}_{P_c(k+1)}\big\{-\Delta_{\boldsymbol{x}(k)}^{\boldsymbol{x}(k)}\log P\big(\boldsymbol{x}(k+1)|\boldsymbol{x}(k)\big)\big\}, \quad (20)$$

$$\boldsymbol{C}^{12}(k)=\mathbb{E}_{P_c(k+1)}\big\{-\Delta_{\boldsymbol{x}(k)}^{\boldsymbol{x}(k+1)}\log P\big(\boldsymbol{x}(k+1)|\boldsymbol{x}(k)\big)\big\}, \quad (21)$$

$$\boldsymbol{C}^{22}(k)\approx\sum_{l=1}^{N}\boldsymbol{L}^{(l)}(k+1)-\sum_{l=1}^{N}\boldsymbol{L}^{(l)}(k+1|k) \quad (22)$$
$$+\mathbb{E}_{P_c(k+1)}\big\{-\Delta_{\boldsymbol{x}(k+1)}^{\boldsymbol{x}(k+1)}\log P\big(\boldsymbol{x}(k+1)|\boldsymbol{x}(k)\big)\big\}.$$

Theorem 1 is similar in structure to the conventional dPCRLB (Proposition 1) with two main differences: (i) The local conditional FIMs ($\boldsymbol{L}^{(l)}(k+1)$ and $\boldsymbol{L}^{(l)}(k+1|k)$) are used instead of their non-conditional counterparts, and; (ii) The global FIM for previous time $\boldsymbol{J}(k)$ is replaced by the global auxiliary FIM $\boldsymbol{J}_{\text{AUX}}(k)$.

Proof of Theorem 1: Decomposing $\boldsymbol{x}(0:k+1)=[\boldsymbol{x}^T(0:k-1),\boldsymbol{x}^T(k),\boldsymbol{x}^T(k+1)]^T$, Eq. (17) for iteration $k+1$ reduces to

$$\boldsymbol{I}(0{:}k{+}1)=\mathbb{E}\left\{-\left[\begin{array}{c:c:c}\Delta_{\boldsymbol{x}(0:k-1)}^{\boldsymbol{x}(0:k-1)} & \Delta_{\boldsymbol{x}(0:k-1)}^{\boldsymbol{x}(k)} & \Delta_{\boldsymbol{x}(0:k-1)}^{\boldsymbol{x}(k+1)}\\ \hdashline \Delta_{\boldsymbol{x}(k)}^{\boldsymbol{x}(0:k-1)} & \Delta_{\boldsymbol{x}(k)}^{\boldsymbol{x}(k)} & \Delta_{\boldsymbol{x}(k)}^{\boldsymbol{x}(k+1)}\\ \hdashline \Delta_{\boldsymbol{x}(k+1)}^{\boldsymbol{x}(0:k-1)} & \Delta_{\boldsymbol{x}(k+1)}^{\boldsymbol{x}(k)} & \Delta_{\boldsymbol{x}(k+1)}^{\boldsymbol{x}(k+1)}\end{array}\right]\log P_c(k{+}1)\right\}$$
$$\triangleq\begin{bmatrix}\boldsymbol{A}^{11}(k) & \boldsymbol{A}^{12}(k) & \boldsymbol{0}\\ \boldsymbol{A}^{21}(k) & \boldsymbol{A}^{22}(k)+\boldsymbol{C}^{11}(k) & \boldsymbol{C}^{12}(k)\\ \boldsymbol{0} & \boldsymbol{C}^{21}(k) & \boldsymbol{C}^{22}(k+1)\end{bmatrix}. \quad (23)$$

Block $\boldsymbol{0}$ stands for a block of all zeros. Terms $\boldsymbol{C}^{11}(k)$, $\boldsymbol{C}^{12}(k)$ and $\boldsymbol{C}^{21}(k)$ are defined as in Eqs. (20)-(21). Terms $\boldsymbol{A}^{11}(k)$, $\boldsymbol{A}^{12}(k)$, $\boldsymbol{A}^{21}(k)$, and $\boldsymbol{A}^{22}(k)$ are derived as follows

$$\begin{bmatrix}\boldsymbol{A}^{11}(k) & \boldsymbol{A}^{12}(k)\\ \boldsymbol{A}^{21}(k) & \boldsymbol{A}^{22}(k)\end{bmatrix}=\mathbb{E}\left\{-\left[\begin{array}{c:c}\Delta_{\boldsymbol{x}(0:k-1)}^{\boldsymbol{x}(0:k-1)} & \Delta_{\boldsymbol{x}(0:k-1)}^{\boldsymbol{x}(k)}\\ \hdashline \Delta_{\boldsymbol{x}(k)}^{\boldsymbol{x}(0:k-1)} & \Delta_{\boldsymbol{x}(k)}^{\boldsymbol{x}(k)}\end{array}\right]\log P_a(k)\right\} \quad (24)$$

where $P_a(k)=P(\boldsymbol{x}(0{:}k)|\boldsymbol{z}(1{:}k))$ and $\boldsymbol{J}_{\text{AUX}}(k)$ the inverse of the $(n_x\times n_x)$ right-lower block of (24), i.e.,

$$\boldsymbol{J}_{\text{AUX}}(k)=\boldsymbol{A}^{22}(k)-\boldsymbol{A}^{21}(k)\big[\boldsymbol{A}^{11}(k)\big]^{-1}\boldsymbol{A}^{12}(k). \quad (25)$$

The decentralized computation of $\boldsymbol{C}^{22}(k{+}1)=\mathbb{E}\{-\Delta_{\boldsymbol{x}(k+1)}^{\boldsymbol{x}(k+1)}\log P_c(k{+}1)\}$ is based on Lemma 1 below, which is an extension of the Chong-Mori-Chang track-fusion theorem [25] for the conditional posterior.

Lemma 1. *Assuming that the observations conditioned on the state variables are independent, the global posterior for a N-sensor network is factorized as follows*

$$P_c(k{+}1)=P\big(\boldsymbol{x}(k{+}1)|\boldsymbol{x}(k)\big)P\big(\boldsymbol{x}(k)|\boldsymbol{x}(k{-}1)\big)P_a(k-1) \quad (26)$$
$$\times\frac{\prod_{l=1}^{N}P\big(\boldsymbol{x}(k{+}1),\boldsymbol{z}^{(l)}(k{+}1)|\boldsymbol{z}^{(l)}(1{:}k)\big)\prod_{l=1}^{N}P\big(\boldsymbol{x}(k)|\boldsymbol{z}^{(l)}(1{:}k)\big)}{\prod_{l=1}^{N}P\big(\boldsymbol{x}(k{+}1)|\boldsymbol{z}^{(l)}(1{:}k)\big)\prod_{l=1}^{N}P\big(\boldsymbol{x}(k)|\boldsymbol{z}^{(l)}(1{:}k{-}1)\big)}$$

The proof of Lemma 1 is not included to save on space. Based on Lemma 1, $\boldsymbol{C}^{22}(k+1)=\mathbb{E}\{-\Delta_{\boldsymbol{x}(k+1)}^{\boldsymbol{x}(k+1)}\log P_c(k+1)\}$ is

$$\boldsymbol{C}^{22}(k{+}1)=\mathbb{E}_{P_c(k+1)}\Big\{-\Delta_{\boldsymbol{x}(k+1)}^{\boldsymbol{x}(k+1)}\log\big(P\left(\boldsymbol{x}(k{+}1)|\boldsymbol{x}(k)\right)\big)\Big\}$$
$$+\sum_{l=1}^{N}\mathbb{E}_{P_c(k+1)}\Big\{-\Delta_{\boldsymbol{x}(k+1)}^{\boldsymbol{x}(k+1)}\log\big(P(\boldsymbol{x}(k{+}1),\boldsymbol{z}^{(l)}(k{+}1))|\boldsymbol{z}^{(l)}(1:k))\big)\Big\}$$
$$-\sum_{l=1}^{N}\mathbb{E}_{P_c(k+1)}\Big\{-\Delta_{\boldsymbol{x}(k+1)}^{\boldsymbol{x}(k+1)}\log\big(P(\boldsymbol{x}(k{+}1)|\boldsymbol{z}^{(l)}(1{:}k))\big)\Big\}. \quad (27)$$

Approximating the summation terms on the right hand side of Eq. (27) with $\boldsymbol{L}^{(l)}(k{+}1)$ and $\boldsymbol{L}^{(l)}(k{+}1|k)$, term $\boldsymbol{C}^{22}(k)$ reduces to Eq (22). Finally $\boldsymbol{L}(k{+}1)$ is calculated as the inverse of the right lower $(n_x\times n_x)$ sub-matrix of $[\boldsymbol{I}(0{:}k{+}1)]^{-1}$, i.e.,

$$\boldsymbol{L}(k{+}1)=\boldsymbol{C}^{22}(k{+}1)- \quad (28)$$
$$\begin{bmatrix}\boldsymbol{0} & \boldsymbol{C}^{21}(k)\end{bmatrix}\begin{bmatrix}\boldsymbol{A}^{11}(k) & \boldsymbol{A}^{12}(k)\\ \boldsymbol{A}^{21}(k) & \boldsymbol{A}^{22}(k)+\boldsymbol{C}^{11}(k)\end{bmatrix}^{-1}\begin{bmatrix}\boldsymbol{0}\\ \boldsymbol{C}^{12}(k)\end{bmatrix}$$
$$=\boldsymbol{C}^{22}(k{+}1)-\boldsymbol{C}^{21}(k)\big(\boldsymbol{J}_{\text{AUX}}(k)+\boldsymbol{C}^{11}(k)\big)^{-1}\boldsymbol{C}^{12}(k). \qquad \square$$

The conditional dPCRLB is computed distributively using Theorem 1, which instead requires $\boldsymbol{J}_{\text{AUX}}(k)$ obtained from Proposition 1.

4. CONDITIONAL DPCRLB BASED SENSOR SELECTION

The cost function for sensor selection is based on the conditional dPCRLBs related to the (x,y) coordinates of the target as follows

$$\mathcal{C}(k+1)=[\boldsymbol{L}(\boldsymbol{x}(k{+}1))]_{xx}^{-1}+[\boldsymbol{L}(\boldsymbol{x}(k{+}1))]_{yy}^{-1}. \quad (29)$$

where $[\boldsymbol{L}(\boldsymbol{x}(k{+}1))]_{xx}^{-1}$ and $[\boldsymbol{L}(\boldsymbol{x}(k{+}1))]_{yy}^{-1}$ are the conditional dPCRLBs corresponding to the x and y coordinates at iteration $k{+}1$. Our observation node selection is carried out in several iterations. During initialization at each iteration, the best observation node for each fusion centre is picked. One observation node among N_f selected sensors forms the initial neighbourhood. The process is repeated till the desired number of observation nodes is included in the neighbourhood set. To select the best observation node at each fusion centre, we use the following local cost function (expressed in terms of fusion-node-observation-node (l,m_l) combination)

$$\mathcal{C}^{(l,m_l)}(k{+}1)=[\boldsymbol{L}^{(\aleph_{\text{obs}}^{(l)})}(\boldsymbol{x}(k{+}1))]_{xx}^{-1}+[\boldsymbol{L}^{(\aleph_{\text{obs}}^{(l)})}(\boldsymbol{x}(k{+}1))]_{yy}^{-1}. \quad (30)$$

where $[\boldsymbol{L}^{(\aleph_{\text{obs}}^{(l)})}(\boldsymbol{x}(k{+}1))]_{xx}^{-1}$ and $[\boldsymbol{L}^{(\aleph_{\text{obs}}^{(l)})}(\boldsymbol{x}(k{+}1))]_{yy}^{-1}$ are the conditional dPCRLB corresponding to the x and y-coordinates in

$$\boldsymbol{L}^{(\aleph_{\text{obs}}^{(l)})}\big(\boldsymbol{x}(k{+}1)\big)=[\boldsymbol{C}^{22}(k{+}1)]^{(l,m_l)} \quad (32)$$
$$-\boldsymbol{D}^{21}(k)\Big(\boldsymbol{J}_{\text{AUX}}^{(\min)}(x(k))+\boldsymbol{D}^{11}(k)\Big)^{-1}\boldsymbol{D}^{12}(k),$$

with
$$[\boldsymbol{C}^{22}(k{+}1)]^{(l,m_l)}=\boldsymbol{L}^{(l,m_l)}(\boldsymbol{x}(k{+}1))+\boldsymbol{L}_{k+1|k+1}^{\min}(t)$$
$$-\boldsymbol{L}^{(l,m_l)}(\boldsymbol{x}(k{+}1|k))-\boldsymbol{L}_{k+1|k}^{(\min)}(t)+\boldsymbol{Q}^{-1}(k). \quad (33)$$

Note that Eqs. (32) and (33) are representations of Eqs. (10) and (7) for a single fusion-node-observation-node (l,m_l) combination. Notation $\boldsymbol{L}^{(\aleph_{\text{obs}}^{(l)})}(\boldsymbol{x}(k+1))$ correspond to the FIM for estimates obtained from the iterating neighbourhood $\aleph_{\text{obs}}^{(l)}(t)$ as it is being optimized. Once optimized, $\aleph_{\text{obs}}^{(l)}(k+1)=\aleph_{\text{obs}}^{(l)}(t)$. Parameters $\boldsymbol{D}^{21}(k)=(\boldsymbol{D}^{12})^T(k)$ and $\boldsymbol{D}^{11}(k)$ are available from the conditional dPCRLB computation block and are fixed for various iterations of the senor selector. Parameter $\boldsymbol{J}_{\text{AUX}}^{(\min)}(x(k))$ corresponds to the dPCRLB from the previously optimized neighbourhood in the last k iteration. Parameter $[\boldsymbol{C}^{22}(k{+}1)]^{(l,m_l)}$ is local for the (l,m_l) fusion-node-observation-node combination and is obtained from Eq. (33). Parameter $\boldsymbol{L}^{(l,m_l)}(\boldsymbol{x}(k+1))$ and $\boldsymbol{L}^{(l,m_l)}(\boldsymbol{x}(k{+}1|k))$ are the conditional dPCRLBs corresponding to the filtering and prediction estimates obtained at fusion node l from a single observation at observation node m_l. Finally, $\boldsymbol{L}_{k+1|k}^{(\min)}(t)$ and $\boldsymbol{L}_{k+1|k+1}^{(\min)}(t)$ are the conditional FIMs corresponding to the filtered and predicted estimates obtained from the iterating neighbourhood $\aleph_{\text{obs}}^{(l)}(t)$. Our sensor selection approach is based on Steps 1 and 2.

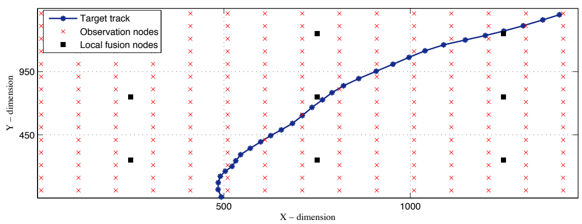

Fig. 1. Actual target's track along with the estimated track.

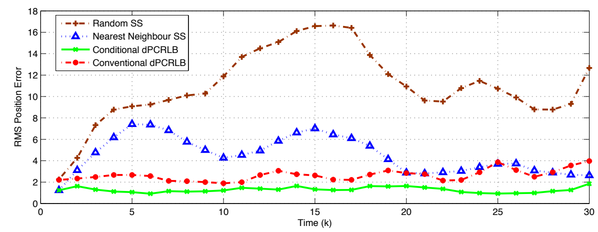

Fig. 2. RMSE for target's position averaged over all fusion nodes.

1. **Initial Selection:** has the following sub-steps:

a. At fusion node l, for $(1 \leq l \leq N_f)$ the conditional FIMs $\boldsymbol{L}^{(l,m_l)}(\boldsymbol{x}(k+1))$ and the cost function $\mathcal{C}^{(l,m_l)}(k+1)$ corresponding to the fusion-node-observation-node (l, m_l) combination are computed based on (30)-(33).
b. From all (l, m_l) combinations, the fusion node l selects one observation node for which $\mathcal{C}^{(l,m_l)}(k+1)$ is minimum. In other words, a single observation node is selected by each fusion node that provides the optimal performance at that node when at the most one observation is used.
c. At this stage, a complete enumeration encompassing all fusion nodes $(1 \leq l \leq N_f)$ is performed. We select one fusion-node-observation-node combination $(q = l, m_q = m_l)$ with the minimum cost function associated to it across the network. A minimum consensus algorithm accomplishes Step 1(c).
d. Matrices $\boldsymbol{L}^{(q,m_q)}(\boldsymbol{x}(k+1)) \triangleq \boldsymbol{L}^{(\min)}_{k+1|k}$ and $\boldsymbol{L}^{(q,m_q)}(\boldsymbol{x}(k+1|k)) \triangleq \boldsymbol{L}^{(\min)}_{k+1|k+1}$ corresponding to the FIMs for the combination (q, m_q) are communicated across the network. The neighbourhood structure is given by $\aleph(1) = \{\aleph^{(l)}_{\text{obs}}(1)\}^{N_f}$. After the initial selection, all $\aleph^{(l)}_{\text{obs}}(1) = \{\}$ (i.e., empty sets) except for $l = q$ where $\aleph^{(q)}_{\text{obs}} = \{m_q\}$. Note that we have added time index $t = 1$ to each neighbourhood to indicate the iteration number for the fusion selection stage. The FIMs $\boldsymbol{L}^{(l,m_l)}(\boldsymbol{x}(k+1))$ computed in Step 1a. are limited to the sensors within the neighbourhood of fusion node l.

2. **Subsequent Selection:** is based on the following substeps: Each fusion node l, $(1 \leq l \leq N_f)$, selects an observation node in its immediate neighbourhood and for it computes the cost function taking into account the previously selected neighbourhood ($\aleph^{(l)}_{\text{obs}}(t)$) and the associated FIMs $\boldsymbol{L}^{(\min)}_{k+1|k}(t)$ and $\boldsymbol{L}^{(\min)}_{k+1|k+1}(t)$.

a. Fusion node l computes $[\boldsymbol{C}^{22}(k+1)]^{(l,m_l)}$, for $(m_l \notin \aleph^{(l)}_{\text{obs}}(t))$, using (32) and (33).
b. Given $\boldsymbol{L}^{(\aleph^{(l)}_{\text{obs}})}(\boldsymbol{x}(k+1))$, Eq. (30) is used to compute the local cost function $\mathcal{C}^{(l,m_l)}(k+1)$.
c. Select the fusion node $\mathcal{L}$ and observation node $m_{\mathcal{L}}$ combination corresponding to the minimum overall cost function using a minimum consensus algorithm.
d. Append the neighbourhood structure to include the new combination $N^{(l)}_{\text{obs}}(t+1) = \{N^{(l)}_{\text{obs}}(t)\}$, appended with the new combination. The overall FIM corresponding to the appended neighbourhood combination is denoted by $\boldsymbol{L}^{(\mathcal{L},m_{\mathcal{L}})}(k+1)$.
e. Matrix $\boldsymbol{L}^{(\min)}(\boldsymbol{x}(k+1))$ now equals to $\boldsymbol{L}^{(\mathcal{L},m_{\mathcal{L}})}(k+1)$, which now corresponds to the overall conditional FIM corresponding to the selected sensors. The new value of matrix $\boldsymbol{L}^{(\min)}(\boldsymbol{x}(k+1))$ is communicated across the network.

3. *Termination:* Check if $N_{\max}$ has been reached. Else, go to Step 2.

5. EXPERIMENTAL RESULTS

A large-scale distributed bearing-only tracking (BOT) application [27] is simulated. A sensor network consisting of $N_s = 225$ sensor nodes and $N_f = 9$ fusion nodes scattered in a square region of dimension (1500×1500) m^2 is considered. The target starts its maneuver from coordinates $(1400, 1400)$. The initial course is set at -140° with the standard deviation of the process noise $\sigma_v = 1.6 \times 10^0$. For simplicity, the observation nodes are assumed distributed uniformly with the fusion node at the centre of its rectangular (500×500)m neighbourhood. Each fusion node communicates only with selected observation nodes within its rectangular (500×500)m neighbourhood and other fusion nodes within a connectivity radius of 550 m. A fusion node is linked to at least one other fusion node in the network. The maximum number of active observation nodes at each iteration is $N_{\max} = 18$ with the constraint that each fusion node (shown as '■') can select four sensors at the most. Fig. 1 shows the target tracks along with locations of observation nodes and local fusion nodes. The observation model is

$$Z^{(l)}(k) \quad = \quad \text{atan}\left(\frac{X(k) - X^{(l)}}{Y(k) - Y^{(l)}}\right) + \zeta^{(l)}(k), \qquad (34)$$

where $(X^{(l)}, Y^{(l)})$ are the coordinates of node l. Target moves according to a clockwise coordinated turn kinematic motion model [27] with maneuver acceleration parameter A_m set to 1.08×10^{-5}km/s^2. Since the distributed dynamical system is non-linear, we use the distributed particle filter implementation (CF/DPF) [28] to track the targets and compute the local FIMs. Our conditional dPCRLB sensor selection approach is compared with other distributed approaches [14, 15]: 1. *Non-conditional dPCRLB-based sensor selection:* where the conventional PCRLB is the selection criteria. 2. *Random-sensor approach:* Observation nodes are selected randomly by each fusion node from within its neighbourhood. 3. *Closest-sensor approach:* where the observation nodes closest to the estimated location of the target are selected. Fig. 2 shows the position root mean square error (RMSE) for the four approaches, where the conditional dPCRLB based sensor selection approach outperforms the other methods and provides the minimum RMSE.

6. SUMMARY

In this paper, a consensus-based sensor selection approach based on the conditional dPCRLB is proposed for a decentralized sensor network with two types of nodes: *sensor nodes* with limited power, no processing ability, which only make observations, and; *local fusion nodes* without any power constraints for processing and communication. The online conditional PCRLB leads to a more accurate representation of the systems's current performance. Our numerical simulations verify the efficiency of the proposed decentralized sensor selection approach over its non-conditional counterpart.

7. REFERENCES

[1] M.E. Liggins II, C-Y Chong, I. Kadar, M.G. Alford, V. Vannicola, and S. Thomopoulos. "Distributed fusion architectures and algorithms for target tracking," in *Proc. of the IEEE*, vol. 85, no. 1, pp. 95–107, 1997.

[2] P. Tichavsky, C.H. Muravchik, and A. Nehorai, "Posterior Cramér-Rao bounds for discrete-time nonlinear filtering," *IEEE Trans. Sig. Proc.*, vol. 46, no. 5, pp. 1386–1396, 1998.

[3] L. Zuo, R. Niu, and P.K. Varshney, "Conditional Posterior Cramér-Rao Lower Bounds for Nonlinear Sequential Bayesian Estimation," *IEEE Tran. on Sig. Pro.*, vol. 59, no. 1, 2011.

[4] M.L. Hernandez, T. Kirubarajan, Y. Bar-Shalom, "Multisensor resource deployment using posterior Cramér-Rao bounds," *IEEE Trans. Aerospace & Elec. Sys.*, vol. 40, no. 2, pp. 399–416, 2004.

[5] O. Ozdemir, R. Niu, P.K. Varshney, A.L. Drozd, "Modified Bayesian Cram R-rao lower bound for nonlinear tracking," *ICASSP*, pp.3972-3975, 2011.

[6] H. Rowaihy, S. Eswaran, M. Johnson, D. Verma, A. Bar-Noy, T. Brown, and T. L. Porta, "A survey of sensor selection schemes in wireless sensor networks, *in Proc. SPIE*, 2007.

[7] C. Kreucher, K. Kastella, and A. Hero III, "Sensor management using an active sensing approach," *Signal Processing*, vol. 85, no. 3, pp. 607-624, 2005.

[8] F. Zhao, 1. Shin, and J. Reich, "Information-driven dynamic sensor collaboration," *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 61-72, 2002.

[9] M. I. Smith, C. R. Angell, M. L. Hernandez, and W. J. Oxford, "Improved data fusion through intelligent sensor management," in *Proceedings of the SPIE*, vol. 5096, 2003.

[10] D. Smith and S.Singh, "Approaches to multisensor data fusion in target tracking: A survey". *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, pp. 1696–1710, 2006.

[11] G. M. Hoffmann and C. 1. Tomlin, "Mobile sensor network control using mutual information methods and particle filters," *IEEE Transactions on Automatic Control*, vol. 55, no. 1, pp. 32-47, 2010.

[12] A.S. Chhetri, D. Morrell, and A.P. Suppappola, "The use of particle filter with the unscented transform to schedule sensors," *ICASSP*, 2004.

[13] D. Guo and X. Wang, "Dynamic sensor collaboration via sequential monte carlo," *IEEE J. Sel. Areas in Comm.*, vol. 22, pp. 1037–1047,2004.

[14] A. Mohammadi and A. Asif, "Decentralized sensor selection based on the distributed posterior Cramr-Rao lower bound," *Information Fusion (FUSION)*, pp.1668-1675, 2012.

[15] R. Tharmarasa, T. Kirubarajan, A. Sinha, and T. Lang, "Decentralized Sensor Selection for Large-Scale Multisensor-Multitarget Tracking," *IEEE Trans. Aero. & Elec. Sys.*, vol. 47, no. 2, pp. 1307-1324, 2011.

[16] R. Tharmarasa, T. Kirubarajan, P. Jiming, and T. Lang, "Optimization-Based Dynamic Sensor Management for Distributed Multitarget Tracking," *IEEE Trans. Sys, Man, & Cybernetics*, vol. 39, pp. 534–546, 2009.

[17] L. Zuo, R. Niu, and P. K. Varshney, "Posterior CRLB based sensor selection for target tracking in sensor networks," *Proc. ICASSP*, vol. 2, pp. 1041–1044,2007.

[18] R. Tharmarasa, "PCRLB based multisensor array management for multitarget tracking." *IEEE Transactions on Aerospace and Electronic Systems*, vol. 43, pp. 539–555, 2007.

[19] X. Zhang, and P.K. Willett, "Cramér-Rao bounds for discrete time linear filtering with measurement origin uncertainty" *Workshop on Estimation, Track. & Fusion: A Tribute to Yaakov Bar-Shalom*, 2001.

[20] M.M. Olama, S.M. Djouadi, C.D. Charalambous, and I.G. Papageorgiou, "Position and velocity tracking in mobile networks using particle and Kalman filtering with comparison", *IEEE Tran. on Vehicle Tech.*, Vol. 57, No. 2, pp. 1001–101, 2008.

[21] C. Chong, F. Zhao, S. Mori and S. Kumar, "Distributed tracking in wire-less ad hoc sensor networks", *Int. Conf. Info. Fusion*, pp. 431-438, 2003.

[22] A. Simonetto, T. Keviczky, and R. Babuska, "Distributed Nonlinear Estimation for Robot Localization using Weighted Consensus ", In *IEEE Inter. Con. on Robotics and Automation*, pp.3026–3031, 2010.

[23] A. Mohammadi and A. Asif, "Distributed Posterior Cramer-Rao Lower Bound for Nonlinear Sequential Bayesian Estimation," *IEEE SAM*, 2012.

[24] A. Mohammadi and A. Asif, "Theoretical Performance Bounds for Reduced-order Linear and Nonlinear Distributed Estimation," accepted in *IEEE Global communiactions conference (GLOBECOM)*, 2012.

[25] C.Y. Chong, S. Mori, and K.C. Chang, *Multi-target Multi-sensor Tracking*, Artech House, pp. 248–295, 1990.

[26] A. Mohammadi and A. Asif, "Distributed Posterior Cramer-Rao Lower Bound for Nonlinear Sequential Bayesian Estimation," *IEEE SAM*, 2012.

[27] M.S. Arulampalam, B. Ristic, N. Gordon, and T. Mansell, "Bearings-only tracking of maneuvering targets using particle filters," *Appl. Sig. Process.*, vol. 15, pp. 2351–2365, 2004.

[28] A. Mohammadi and A.Asif, "Consensus-based Particle Filter Implementations for Distributed Non-linear Systems", chapter 24 in *Nonlinear Est. & Applications to Industrial Sys. Control*, Editor G. Rigatos, 2011.

[29] C.H. Papadimitriou and K. Steiglitz, "Combinatorial Optimization Algorithms and Complexity" Mineola, NY: Dover Publication, 1998.