Cognitive Radar Framework for Target Detection and Tracking

Kristine L. Bell, *Fellow, IEEE*, Christopher J. Baker, *Senior Member, IEEE*, Graeme E. Smith, *Senior Member, IEEE*, Joel T. Johnson, *Fellow, IEEE*, and Muralidhar Rangaswamy, *Fellow, IEEE*

***Abstract*—Most radar systems employ a feed-forward processing chain in which they first perform some low-level processing of received sensor data to obtain target detections and then pass the processed data on to some higher-level processor such as a tracker, which extracts information to achieve a system objective. System performance can be improved using adaptation between the information extracted from the sensor/processor and the design and transmission of subsequent illuminating waveforms. As such, cognitive radar systems offer much promise. In this paper, we develop a general cognitive radar framework for a radar system engaged in target tracking. The model includes the higher-level tracking processor and specifies the feedback mechanism and optimization criterion used to obtain the next set of sensor data. Both target detection (track initiation/termination) and tracking (state estimation) are addressed. By separating the general principles from the specific application and implementation details, our formulation provides a flexible framework applicable to the general tracking problem. We demonstrate how the general framework may be specialized for a particular problem using a distributed sensor model in which system resources (observation time on each sensor) are allocated to optimize tracking performance. The cognitive radar system is shown to offer significant performance gains over a standard feed-forward system.**

***Index Terms*—Cognitive radar, fully adaptive radar, target tracking, track initiation, track termination, sensor management.**

I. Introduction

MOST radar systems employ a feed-forward processing chain in which they first perform some low-level processing of received sensor data to obtain target detections and then pass the processed data on to some higher-level processor (such as a tracker or classifier), which extracts information (such as target kinematic parameters and target type) to achieve a system objective. Tracking and classification can be improved using adaptation between the information extracted from the sensor/processor and the design and transmission of subsequent illuminating waveforms. As such, cognitive radar (CR) systems offer much promise for improved performance [1]–[4].

The heart of cognition is the *perception-action cycle* that both informs and is informed by *memory* and uses *attention* to focus sensing resources on relevant observations [2]–[4]. Cognition requires stimulation by *sensors*. In the human this is via hearing, touch, smell, vision, and taste. The nervous system *processes* the sensed stimuli and converts them into a *perception* of the world. We are able to take informed *action* by interpreting our perception of the world and making decisions. As a consequence, the nervous system sends signals that activate our muscles, thus enabling the desired action to take place. Informed *decision-making* is a key feature of the perception-action cycle. It requires the establishment of choices and the selection of one according to a desired goal. There is an element of *prediction* that arises in which perceptual information is combined with a model of the environment to predict the effect actions may have on it.

Artificial cognition in a cognitive radar system attempts to mimic the perception-action cycle to make the best use of system resources for the situation at hand. Haykin introduced the concept of cognitive radar in [1], however the motivation and many of the underlying ideas grew out of the fields of knowledge-aided radar detection and tracking [5], [6], agile waveform design [7]–[11], and sensor management [12]–[18]. Artificial cognition has also been used to optimize the processing rather than the sensing in [19], [20]. In [2], [3], Haykin makes the distinction between traditional feed-forward radar, fully adaptive radar, and truly cognitive radar. While a fully adaptive radar [5], [6] may employ feedback and use prior knowledge stored in memory, a cognitive radar predicts the consequences of actions, performs explicit decision-making, learns from the environment, and uses memory to store the learned knowledge [3]. Haykin's approach uses Bayesian filtering [21], [22] as the basis for the artificial perception-action cycle. It provides the processing, perception, and memory inherent in a cognitive system. The Bayesian filter is augmented with a decision-making controller that uses perception from the filter and prediction of future outcomes to determine the next action taken by the sensor. The Bayesian approach has been used in [7]–[20] for a variety of applications.

In this work, we generalize and formalize the work in [1]–[3], [7]–[20] and develop a general cognitive radar framework for a system engaged in target tracking. We begin with a general sensor/processor system that includes the lower-level radar

Manuscript received January 29, 2015; revised May 11, 2015; accepted July 20, 2015. Date of publication August 06, 2015; date of current version November 17, 2015. This work was supported by the U.S. Air Force Research Laboratory under contracts FA8650-13-M-1656 and FA8650-14-C-1825. Opinions, interpretations, conclusions, and recommendations are those of the authors and not necessarily endorsed by the U. S. Government. The guest editor coordinating the review of this manuscript and approving it for publication was Dr. Maria Greco.

K. Bell is with Metron, Inc., Reston, VA 20190 USA (e-mail: bell@metsci.com).

C. Baker, G. Smith, and J. Johnson are with the Department of Electrical and Computer Engineering and ElectroScience Laboratory, The Ohio State University, Columbus, OH 43210 USA (e-mail: baker.1891@osu.edu; smith.8347@osu.edu; johnson.1374@osu.edu).

M. Rangaswamy is with the Sensors Directorate, Radar Signal Processing Branch, U.S. Air Force Research Laboratory, Wright-Patterson AFB, OH 45433 USA (e-mail: muralidhar.rangaswamy@us.af.mil).

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/JSTSP.2015.2465304

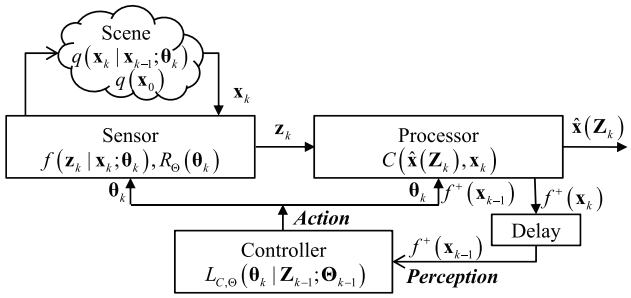

Fig. 1. Cognitive sensor/processor system.

sensor and the higher-level task processor. The model specifies the feedback mechanism and optimization criterion used to obtain the next set of sensor data to optimize the performance of the task processor. In this paper, the model is specialized for a task processor whose objective is single target tracking, as well as for target detection and track initiation/termination. By separating the general principles from the specific application and implementation details, our formulation provides a flexible framework applicable to the general tracking problem. We demonstrate how the general framework may be specialized for a particular problem using a distributed sensor system similar to the cognitive radar networks in [17], [18], in which system resources (observation time on each sensor) are allocated to optimize tracking performance. We show that the cognitive radar system offers significant performance gains over a standard feed-forward system. Preliminary versions of this work appeared in [23], [24] and follow-on work with a different application appears in [25].

II. Cognitive Sensor/Processor System Model

The basic mathematical model of a cognitive sensor/processor system is shown in Fig. 1. The system consists of four components: the scene, which includes the target and the environment, the sensor that observes the scene, the processor that converts the observed data into a perception of the scene, and the controller that determines the next actions taken by the sensor and processor. The controller is the novel component that distinguishes a cognitive system from a traditional feed-forward sensor/processor system.

We assume that the objective of the system is to estimate the state of a target in the scene. The target state at time t_k is denoted as $\mathbf{x}_k$. The sensor observes the scene and produces a measurement vector $\mathbf{z}_k$ that depends on the target state $\mathbf{x}_k$ and the sensor parameters $\boldsymbol{\theta}_k$. We assume that the estimate of the target state at time t_k is a function of the observations up to time t_k, which in turn depend on the sensor parameters up to time t_k, which we denote as $\mathbf{Z}_k \equiv \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_k\}$ and $\boldsymbol{\Theta}_k \equiv \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_k\}$, respectively.

We assume a Markov motion model with initial target state probability density function (pdf) $q(\mathbf{x}_0)$ and the transition pdf $q(\mathbf{x}_k \mid \mathbf{x}_{k-1};\boldsymbol{\theta}_k)$. The transition density may depend on the sensor parameters. This will occur, for example, when the choice of sensor parameters affects the time difference $\Delta t_k = t_k - t_{k-1}$, as in the example considered in [25].

The sensor measurement model is described by the conditional pdf, or likelihood function, $f(\mathbf{z}_k \mid \mathbf{x}_k;\boldsymbol{\theta}_k)$. The cost of obtaining an observation and any constraints on the sensor parameters are modeled by the sensor cost function $R_\Theta(\boldsymbol{\theta}_k)$.

The processor processes the data and produces an estimate of the target state $\hat{\mathbf{x}}_k(\mathbf{Z}_k)$ by minimizing the expected value of the processor cost function $C(\hat{\mathbf{x}}_k(\mathbf{Z}_k), \mathbf{x}_k)$.

The controller decides on the next value for the sensor parameters $\boldsymbol{\theta}_k$ by minimizing a loss function $L_{C,\Theta}(\cdot)$ that balances the performance of the processor via the processor cost function $C(\cdot,\cdot)$ and the cost of using the sensor via the sensor cost function $R_\Theta(\cdot)$.

For the Markov motion model, the conditional or posterior pdf of $\mathbf{x}_k$ conditioned on $\mathbf{Z}_k$ may be obtained from the standard Bayes-Markov recursion [21], [22], [26], in which we first compute the predicted density $f^-(\mathbf{x}_k) \equiv f(\mathbf{x}_k \mid \mathbf{Z}_{k-1};\boldsymbol{\Theta}_k)$ using the Chapman-Kolmogorov equation, and then the posterior density $f^+(\mathbf{x}_k) \equiv f(\mathbf{x}_k|\mathbf{Z}_k;\boldsymbol{\Theta}_k)$ using Bayes rule. The recursion is initialized with

$$f^+(\mathbf{x}_0) = q(\mathbf{x}_0), \tag{1}$$

and then proceeds according to:

$$\begin{aligned} f^-(\mathbf{x}_k) &\equiv f(\mathbf{x}_k \mid \mathbf{Z}_{k-1};\boldsymbol{\Theta}_k) \\ &= \int q(\mathbf{x}_k \mid \mathbf{x}_{k-1};\boldsymbol{\theta}_k) f^+(\mathbf{x}_{k-1}) d\mathbf{x}_{k-1} \end{aligned} \tag{2}$$

$$\begin{aligned} f^+(\mathbf{x}_k) &\equiv f(\mathbf{x}_k \mid \mathbf{Z}_k;\boldsymbol{\Theta}_k) \\ &= \frac{f(\mathbf{z}_k \mid \mathbf{x}_k;\boldsymbol{\theta}_k) f^-(\mathbf{x}_k)}{\int f(\mathbf{z}_k \mid \mathbf{x}_k;\boldsymbol{\theta}_k) f^-(\mathbf{x}_k) d\mathbf{x}_k}. \end{aligned} \tag{3}$$

The *conditional Bayes risk* is the expected value of the processor cost function with respect to the conditional pdf of $\mathbf{x}_k$ given $\mathbf{Z}_k$,

$$R_C^+(\mathbf{Z}_k;\boldsymbol{\Theta}_k) \equiv E_k^+\{C(\hat{\mathbf{x}}(\mathbf{Z}_k), \mathbf{x}_k)\}, \tag{4}$$

where $E_k^+\{\cdot\}$ denotes expectation with respect to $f^+(\mathbf{x}_k)$. The estimator is found by minimizing the conditional Bayes risk [27]:

$$\hat{\mathbf{x}}(\mathbf{Z}_k) = \underset{\hat{\mathbf{x}}(\mathbf{Z}_k)}{\arg\min}\ R_C^+(\mathbf{Z}_k;\boldsymbol{\Theta}_k). \tag{5}$$

In the controller, we assume that we have received observations up to time t_{k-1} and want to find the next set of sensor parameters $\boldsymbol{\theta}_k$ to optimize the performance of the state estimator that will include the next observation $\mathbf{z}_k$ as well as the previous observations $\mathbf{Z}_{k-1}$. We define the *joint conditional pdf* of $\mathbf{x}_k$ and $\mathbf{z}_k$ conditioned on $\mathbf{Z}_{k-1}$ as:

$$\begin{aligned} f^\uparrow(\mathbf{x}_k, \mathbf{z}_k) &\equiv f(\mathbf{x}_k, \mathbf{z}_k \mid \mathbf{Z}_{k-1};\boldsymbol{\Theta}_k) \\ &= f(\mathbf{z}_k \mid \mathbf{x}_k;\boldsymbol{\theta}_k) f(\mathbf{x}_k \mid \mathbf{Z}_{k-1};\boldsymbol{\Theta}_k) \\ &= f(\mathbf{z}_k \mid \mathbf{x}_k;\boldsymbol{\theta}_k) f^-(\mathbf{x}_k). \end{aligned} \tag{6}$$

We define the *predicted conditional Bayes risk* for the estimator $\hat{\mathbf{x}}(\mathbf{Z}_k)$ by taking the expectation of the processor cost function with respect to the joint conditional pdf $f^\uparrow(\mathbf{x}_k, \mathbf{z}_k)$,

$$R_C^\uparrow(\boldsymbol{\theta}_k \mid \mathbf{Z}_{k-1};\boldsymbol{\Theta}_{k-1}) \equiv E_k^\uparrow\{C(\hat{\mathbf{x}}(\mathbf{Z}_k), \mathbf{x}_k)\}. \tag{7}$$

It is important to emphasize that the predicted conditional Bayes risk is a function of the known past observations $\mathbf{Z}_{k-1}$ but not the unknown next observation $\mathbf{z}_k$ since it is averaged over both

$\mathbf{z}_k$ and $\mathbf{x}_k$. It is also function of all the sensor parameters in $\boldsymbol{\Theta}_k$, however we separate the dependence on the unknown next sensor parameter $\boldsymbol{\theta}_k$ from the known past sensor parameters $\boldsymbol{\Theta}_{k-1}$ so that we may optimize over $\boldsymbol{\theta}_k$.

The next value of $\boldsymbol{\theta}_k$ is chosen to minimize a loss function that balances the predicted conditional Bayes risk and the sensor cost,

$$L_{C,\Theta}(\boldsymbol{\theta}_k \mid \mathbf{Z}_{k-1}; \boldsymbol{\Theta}_{k-1}) \equiv L\Big\{ R_C^{\uparrow}(\boldsymbol{\theta}_k \mid \mathbf{Z}_{k-1}; \boldsymbol{\Theta}_{k-1}), R_{\Theta}(\boldsymbol{\theta}_k) \Big\}. \tag{8}$$

The controller optimization problem is then given by

$$\boldsymbol{\theta}_k = \arg\min_{\boldsymbol{\theta}} \; L_{C,\Theta}\left(\boldsymbol{\theta} \mid \mathbf{Z}_{k-1}; \boldsymbol{\Theta}_{k-1}\right). \tag{9}$$

The specific form of the loss function in (8) depends on the nature of the sensor cost function and is chosen to provide a tractable controller optimization problem in (9). It is often the case that the sensor parameters are restricted to an allowable set, i.e., $\boldsymbol{\theta}_k \in \boldsymbol{\Theta}$, and/or must satisfy a constraint of the form $g(\boldsymbol{\theta}_k) = 0$. The sensor cost function for this case is equal to zero when the constraints are met and infinite otherwise,

$$R_{\Theta}(\boldsymbol{\theta}_k) = \begin{cases} 0 & \boldsymbol{\theta}_k \in \boldsymbol{\Theta}; g(\boldsymbol{\theta}_k) = 0 \\ \infty & \text{otherwise.} \end{cases} \tag{10}$$

If we define the controller loss function to be the sum of the predicted conditional Bayes risk and the sensor cost, we have

$$L_{C,\Theta}(\boldsymbol{\theta}_k \mid \mathbf{Z}_{k-1}; \boldsymbol{\Theta}_{k-1}) = R_C^{\uparrow}(\boldsymbol{\theta}_k \mid \mathbf{Z}_{k-1}; \boldsymbol{\Theta}_{k-1}) + R_{\Theta}(\boldsymbol{\theta}_k). \tag{11}$$

Substituting (10) and (11) into (9), the controller optimization problem becomes:

$$\boldsymbol{\theta}_k = \arg\min_{\boldsymbol{\theta}} R_C^{\uparrow}(\boldsymbol{\theta} \mid \mathbf{Z}_{k-1}; \boldsymbol{\Theta}_{k-1}) + \begin{cases} 0 & \boldsymbol{\theta} \in \boldsymbol{\Theta}; g(\boldsymbol{\theta}) = 0 \\ \infty & \text{otherwise,} \end{cases} \tag{12}$$

which may be written as the equivalent constrained optimization problem:

$$\boldsymbol{\theta}_k = \arg\min_{\boldsymbol{\theta}} R_C^{\uparrow}(\boldsymbol{\theta} \mid \mathbf{Z}_{k-1}; \boldsymbol{\Theta}_{k-1}) \quad s.t.\ \boldsymbol{\theta} \in \boldsymbol{\Theta}; g(\boldsymbol{\theta} = 0. \tag{13}$$

The cognitive sensor/processor system model described by Fig. 1 and (1)–(9) is quite general and can be applied to many problems. In Sections III and IV, we specialize the framework for systems whose objectives are single target tracking and track initiation/termination, respectively.

III. Single Target Tracking

To specialize the framework for single target tracking, we specify the processor cost function and derive the corresponding state estimator and predicted conditional Bayes risk function used by the controller.

For parameter estimation, a commonly used cost function is the sum of the squared estimation errors, which is given by:

$$C\left(\hat{\mathbf{x}}(\mathbf{Z}_k), \mathbf{x}_k\right) = \operatorname{tr}\left\{ \left[\hat{\mathbf{x}}(\mathbf{Z}_k) - \mathbf{x}_k\right] \left[\hat{\mathbf{x}}(\mathbf{Z}_k) - \mathbf{x}_k\right]^T \right\}. \tag{14}$$

For the cost function in (14), the solution to (5) is the minimum mean-square error (MMSE) estimator, which is the conditional mean [27]:

$$\hat{\mathbf{x}}(\mathbf{Z}_k) = \boldsymbol{\mu}_k^{+} \equiv E_k^{+}\{\mathbf{x}_k\}. \tag{15}$$

The predicted conditional Bayes risk is given by:

$$R_C^{\uparrow}(\boldsymbol{\theta}_k \mid \mathbf{Z}_{k-1}; \boldsymbol{\Theta}_{k-1}) = \operatorname{tr}\left\{ \boldsymbol{\Sigma}_k^{\uparrow}(\boldsymbol{\theta}_k \mid \mathbf{Z}_{k-1}; \boldsymbol{\Theta}_{k-1}) \right\}, \tag{16}$$

where

$$\boldsymbol{\Sigma}_k^{\uparrow}(\boldsymbol{\theta}_k \mid \mathbf{Z}_{k-1}; \boldsymbol{\Theta}_{k-1}) \equiv E_k^{\uparrow}\left\{ \left[\hat{\mathbf{x}}(\mathbf{Z}_k) - \mathbf{x}_k\right] \left[\hat{\mathbf{x}}(\mathbf{Z}_k) - \mathbf{x}_k\right]^T \right\} \tag{17}$$

is the predicted conditional mean square error (PC-MSE) matrix. The trace is one of several scalar performance metrics that can be applied to the PC-MSE matrix. Other possibilities include the weighted trace, the determinant, and the maximum eigenvalue. Good discussions are provided in [3] and [13].

In most cases, it is not possible to evaluate $\boldsymbol{\Sigma}_k^{\uparrow}(\boldsymbol{\theta}_k \mid \mathbf{Z}_{k-1}; \boldsymbol{\Theta}_{k-1})$ analytically. However, the *Bayesian Cramér-Rao lower bound* (BCRLB), which is the inverse of the *Bayesian information matrix* (BIM), provides a matrix lower bound[1] on the MSE matrix of any estimator [27], [28] and is usually analytically tractable. It is frequently used as a tool for system analysis in place of the MSE matrix. An overview of the BCRLB and BIM is given in Appendix A.

For target tracking, application of the BCRLB theory yields the *posterior Cramér-Rao lower bound* (PCRLB) [29], [28]. The PCRLB provides a lower bound on the global MSE that has been averaged over $\mathbf{x}_k$ and $\mathbf{Z}_k$, thus it characterizes tracker performance for all possible data that might have been received. Here we develop a *predicted conditional Cramér-Rao lower bound* (PC-CRLB) to bound the PC-MSE matrix in (17), which is averaged over the joint density of $\mathbf{x}_k$ and $\mathbf{z}_k$ conditioned on $\mathbf{Z}_{k-1}$. The PC-CRLB differs from the PCRLB in that it characterizes performance conditioned on the actual data that has been received.

Let $\nabla_{\mathbf{x}} g(\mathbf{x})$ denote the gradient of a function $g(\mathbf{x})$ with respect to the components of the N-dimensional vector $\mathbf{x}$, and $\boldsymbol{\Delta}_{\mathbf{x}}^{\mathbf{x}} g(\mathbf{x})$ denote the matrix of second-order partial derivatives, i.e.,

$$\nabla_{\mathbf{x}} g(\mathbf{x}) \equiv \left[\frac{\partial g(\mathbf{x})}{\partial x_1} \quad \frac{\partial g(\mathbf{x})}{\partial x_2} \quad \cdots \quad \frac{\partial g(\mathbf{x})}{\partial x_N} \right]^T \tag{18}$$

$$\boldsymbol{\Delta}_{\mathbf{x}}^{\mathbf{x}} g(\mathbf{x}) \equiv \nabla_{\mathbf{x}} \left[\nabla_{\mathbf{x}} g(\mathbf{x})\right]^T. \tag{19}$$

Following Appendix A, we define the *predicted conditional Bayesian information matrix* (PC-BIM) as:

$$\mathbf{B}_k^{\uparrow}(\boldsymbol{\theta}_k \mid \mathbf{Z}_{k-1}; \boldsymbol{\Theta}_{k-1}) = -E_k^{\uparrow}\left\{ \boldsymbol{\Delta}_{\mathbf{x}_k}^{\mathbf{x}_k} \ln f^{\uparrow}(\mathbf{x}_k, \mathbf{z}_k) \right\}. \tag{20}$$

The PC-CRLB is the inverse of the PC-BIM and has the property:

$$R_C^{\uparrow}(\boldsymbol{\theta}_k \mid \mathbf{Z}_{k-1}; \boldsymbol{\Theta}_{k-1}) \geq \operatorname{tr}\left\{ \mathbf{B}_k^{\uparrow}(\boldsymbol{\theta}_k \mid \mathbf{Z}_{k-1}; \boldsymbol{\Theta}_{k-1})^{-1} \right\}. \tag{21}$$

[1] A matrix lower bound of the form $\mathbf{B} \succeq \mathbf{A}$ means that $\mathbf{B} - \mathbf{A}$ is a positive semidefinite matrix, or equivalently that $\operatorname{tr}[\mathbf{C}^T\mathbf{B}\mathbf{C}] \geq \operatorname{tr}[\mathbf{C}^T\mathbf{A}\mathbf{C}]$ for any non-zero matrix $\mathbf{C}$.

As shown in Appendix A, the PC-BIM may be expressed as the sum of a prior term and a data term as follows [27]:

$$\mathbf{B}_k^{\uparrow}(\boldsymbol{\theta}_k \mid \mathbf{Z}_{k-1}; \boldsymbol{\Theta}_{k-1}) = \mathbf{B}_k^{-}(\mathbf{Z}_{k-1}; \boldsymbol{\Theta}_{k-1}) + \mathbf{J}_k^{-}(\boldsymbol{\theta}_k \mid \mathbf{Z}_{k-1}; \boldsymbol{\Theta}_{k-1}). \quad (22)$$

From (6), the predicted pdf $f^-(\mathbf{x}_k)$ serves as the prior pdf of $\mathbf{x}_k$ in the calculation of the PC-BIM, therefore we call the prior term in (22) the *predicted information matrix* (PIM). It is defined as:

$$\mathbf{B}_k^{-}(\boldsymbol{\theta}_k \mid \mathbf{Z}_{k-1}; \boldsymbol{\Theta}_{k-1}) \equiv -E_k^{-}\left\{\boldsymbol{\Delta}_{\mathbf{x}_k}^{\mathbf{x}_k} \ln f^-(\mathbf{x}_k)\right\}, \quad (23)$$

where the expectation is with respect to $f^-(\mathbf{x}_k)$. Evaluation of the PIM analytically is difficult in many cases. One case where it can be evaluated is for a Gaussian density, where it is equal to the inverse of the covariance matrix. Using the intuition gained from the Gaussian case, we approximate the PIM by the inverse of the predicted covariance matrix, which is straightforward to evaluate numerically. In the Kalman filter family of trackers, the predicted density is presumed to be Gaussian and the predicted covariance matrix is available from the recursion. In particle filter or grid-based trackers, the predicted covariance matrix can be calculated numerically from the set of particles/grid points and weights that represent the predicted density. The approximation is:

$$\mathbf{B}_k^{-}(\mathbf{Z}_{k-1}; \boldsymbol{\Theta}_{k-1}) \approx \boldsymbol{\Sigma}_k^{-}(\boldsymbol{\theta}_k \mid \mathbf{Z}_{k-1}; \boldsymbol{\Theta}_{k-1})^{-1}, \quad (24)$$

where

$$\boldsymbol{\Sigma}_k^{-}(\boldsymbol{\theta}_k \mid \mathbf{Z}_{k-1}; \boldsymbol{\Theta}_{k-1}) \equiv E_k^{-}\left\{\left[\mathbf{x}_k - \boldsymbol{\mu}_k^{-}\right]\left[\mathbf{x}_k - \boldsymbol{\mu}_k^{-}\right]^T\right\} \quad (25)$$

and

$$\boldsymbol{\mu}_k^{-} \equiv E_k^{-}\{\mathbf{x}_k\}. \quad (26)$$

The data term in (22) is the expected value of the Fisher information matrix (FIM) with respect to the predicted density $f^-(\mathbf{x}_k)$. The expected FIM (EFIM) is given by:

$$\mathbf{J}^{-}(\boldsymbol{\theta}_k \mid \mathbf{Z}_{k-1}; \boldsymbol{\Theta}_{k-1}) \equiv E_k^{-}\{\mathbf{J}_{\mathbf{x}}(\mathbf{x}_k; \boldsymbol{\theta}_k)\}, \quad (27)$$

where $\mathbf{J}_{\mathbf{x}}(\mathbf{x}_k; \boldsymbol{\theta}_k)$ is the standard FIM [27], [28]:

$$\mathbf{J}_{\mathbf{x}}(\mathbf{x}_k; \boldsymbol{\theta}_k) \equiv -E_{\mathbf{z}_k \mid \mathbf{x}_k; \boldsymbol{\theta}_k}\left\{\boldsymbol{\Delta}_{\mathbf{x}_k}^{\mathbf{x}_k} \ln f(\mathbf{z}_k \mid \mathbf{x}_k; \boldsymbol{\theta}_k)\right\}. \quad (28)$$

We assume that we have an analytical expression for the likelihood function $f(\mathbf{z}_k \mid \mathbf{x}_k; \boldsymbol{\theta}_k)$ and can evaluate the FIM $\mathbf{J}_{\mathbf{x}}(\mathbf{x}_k; \boldsymbol{\theta}_k)$. The expectation in (27) can be accomplished analytically or numerically.

The expressions in (1)–(3), (15), and (21)–(28) provide the Bayes-Markov tracking recursions, the state estimate, and the predicted conditional Bayes risk expressions for a cognitive sensor/processor system whose objective is single target tracking. The formulation in this paper provides a generalization and formalism to the cognitive radar tracking formulations in [1]–[3], [7]–[20]. By separating the general principles from the specific application and implementation details, our formulation provides a general framework that can be applied to a wide variety of tracking problems.

An important component of our formulation is the predicted conditional Bayes risk used in the controller, which is a generalization of the posterior expected covariance matrix discussed in [3] and [7]. For a squared error cost function, it can be lower bounded by the PC-CRLB, which is a more suitable bound than the PCRLB. The PCRLB is the proper tool for characterizing the average MSE performance of the tracker in a static system, but the PC-CRLB is more appropriate for cognitive systems because it more accurately bounds the PC-MSE performance of the current realization of the tracker.

The current literature is not clear on this point. While most formulations recognize that the tracker performance is characterized by the PC-MSE matrix, the PCRLB is widely used to bound the performance. Exceptions are [11], in which the PCRLB is modified to calculate an EFIM with respect to the predicted density, and [12], in which a "predictive PCRLB" is developed that is essentially the same as the PC-CRLB developed here. A "conditional PCRLB" is also developed in [30] which has an elegant recursive form, however it is a joint bound on the past and current target states, and is therefore a weaker bound than the PC-CRLB and predictive PCRLB.[2] However, we note that in some problems, such as the linear Gaussian model, this point is moot since the PC-MSE matrix, posterior covariance matrix of the tracker, and the different bounds do not depend on the data and therefore are all the same.

IV. Track Initiation and Termination

For track initiation and termination, the goal is to minimize the time to detect the presence or absence of a target. We do this by maximizing the probability of making a correct decision at each time. To specialize the general cognitive sensor/processor framework for this problem, we follow the likelihood ratio detection and tracking (LRDT) methodology of ([21], Ch. 7). On H_1 when the target is present, $\mathbf{x}_k \in \mathbb{X}$, where $\mathbb{X}$ is the target-present state space. We define a null (target-absent) state $\varnothing$ so that $\mathbf{x}_k = \varnothing$ on H_0. We define the augmented state space $\mathbb{X}^{\varnothing} \equiv \mathbb{X} \cup \varnothing$ and develop the Bayes-Markov recursions for this model. We then specify the processor cost function and derive the corresponding state estimator and predicted conditional Bayes risk function used by the controller.

A. Bayes-Markov Recursion and State Estimator

Let $\tilde{f}(\mathbf{x}_k)$ denote a pdf on the augmented state space $\mathbb{X}^{\varnothing}$. It is characterized by two components $P(\varnothing_k)$ and $f(\mathbf{x}_k)$, where

$$P(\varnothing_k) \equiv \Pr(\mathbf{x}_k = \varnothing), \quad (29)$$

and $f(\mathbf{x}_k)$ is the conditional pdf of $\mathbf{x}_k$ given that $\mathbf{x}_k \in \mathbb{X}$. The pdf $\tilde{f}(\mathbf{x}_k)$ has the form

$$\tilde{f}(\mathbf{x}_k) = P(\varnothing_k)\delta(\mathbf{x}_k - \varnothing) + [1 - P(\varnothing_k)]\, f(\mathbf{x}_k), \quad (30)$$

where $\delta(\cdot)$ is the Dirac delta function.

The initial target pdf $\tilde{q}(\mathbf{x}_0)$ is characterized by $P(\varnothing_0)$ and $q(\mathbf{x}_0)$ and is given by:

$$\tilde{q}(\mathbf{x}_0) = P(\varnothing_0)\delta(\mathbf{x}_0 - \varnothing) + [1 - P(\varnothing_0)]\, q(\mathbf{x}_0). \quad (31)$$

[2]Consider the current target state as the desired parameter to be estimated and the past target states as nuisance parameters. It is shown in [31] (see also [28]) that a bound on the desired parameter is greater than or equal to a joint bound on the desired and nuisance parameters.

Let $\tilde{q}(\mathbf{x}_k \mid \mathbf{x}_{k-1}; \boldsymbol{\theta}_k)$ denote a transition density on the augmented state space. It is characterized by four components:

$$\begin{array}{ll} P(\varnothing_k \mid \varnothing_{k-1}) \equiv \Pr(\mathbf{x}_k = \varnothing \mid \mathbf{x}_{k-1} = \varnothing) & \mathbf{x}_k = \varnothing, \mathbf{x}_{k-1} = \varnothing \\ P(\varnothing_k \mid \mathbb{X}_{k-1}) \equiv \Pr(\mathbf{x}_k = \varnothing \mid \mathbf{x}_{k-1} \in \mathbb{X}) & \mathbf{x}_k = \varnothing, \mathbf{x}_{k-1} \in \mathbb{X} \\ q(\mathbf{x}_k \mid \varnothing_{k-1}) & \mathbf{x}_k \in \mathbb{X}, \mathbf{x}_{k-1} = \varnothing \\ q(\mathbf{x}_k \mid \mathbf{x}_{k-1}; \boldsymbol{\theta}_k) & \mathbf{x}_k \in \mathbb{X}, \mathbf{x}_{k-1} \in \mathbb{X}, \end{array} \tag{32}$$

and has the form:

$$\tilde{q}(\mathbf{x}_k \mid \mathbf{x}_{k-1}; \boldsymbol{\theta}_k) = \begin{cases} P(\varnothing_k \mid \varnothing_{k-1}) \delta(\mathbf{x}_k - \varnothing) \\ \quad + [1 - P(\varnothing_k \mid \varnothing_{k-1})]\, q(\mathbf{x}_k \mid \varnothing_{k-1}) & \mathbf{x}_{k-1} = \varnothing \\ P(\varnothing_k \mid \mathbb{X}_{k-1}) \delta(\mathbf{x}_k - \varnothing) \\ \quad + [1 - P(\varnothing_k \mid \mathbb{X}_{k-1})]\, q(\mathbf{x}_k \mid \mathbf{x}_{k-1}; \boldsymbol{\theta}_k) & \mathbf{x}_{k-1} \in \mathbb{X}. \end{cases} \tag{33}$$

Let $\tilde{f}(\mathbf{z}_k \mid \mathbf{x}_k; \boldsymbol{\theta}_k)$ denote the likelihood function on the augmented state space. It is characterized by the likelihood functions on H_0 and H_1, which are denoted by $f(\mathbf{z}_k \mid \varnothing_k; \boldsymbol{\theta}_k)$ and $f(\mathbf{z}_k \mid \mathbf{x}_k; \boldsymbol{\theta}_k)$, respectively.

Let $\tilde{f}^-(\mathbf{x}_k)$ denote the predicted density on the augmented state space, with components $P^-(\varnothing_k)$ and $f^-(\mathbf{x}_k)$, and $\tilde{f}^+(\mathbf{x}_k)$ denote the posterior density on the augmented state space, with components $P^+(\varnothing_k)$ and $f^+(\mathbf{x}_k)$. Let $\tilde{f}^{\uparrow}(\mathbf{x}_k, \mathbf{z}_k)$ denote the joint conditional density on the augmented state space. It has the form:

$$\begin{aligned} \tilde{f}^{\uparrow}(\mathbf{x}_k, \mathbf{z}_k) &= \tilde{f}(\mathbf{z}_k \mid \mathbf{x}_k; \boldsymbol{\theta}_k) \tilde{f}^-(\mathbf{x}_k) \\ &= P^-(\varnothing_k) f(\mathbf{z}_k \mid \varnothing_k; \boldsymbol{\theta}_k) \delta(\mathbf{x}_k - \varnothing) \\ &\quad + \left[1 - P^-(\varnothing_k)\right] f(\mathbf{z}_k \mid \mathbf{x}_k; \boldsymbol{\theta}_k) f^-(\mathbf{x}_k). \end{aligned} \tag{34}$$

We express the Bayes-Markov recursions for $\tilde{f}^-(\mathbf{x}_k)$ and $\tilde{f}^+(\mathbf{x}_k)$ as recursions on $P^-(\varnothing_k), f^-(\mathbf{x}_k), P^+(\varnothing_k)$, and $f^+(\mathbf{x}_k)$. Following [21], the Bayes-Markov recursions are initialized with:

$$P^+(\varnothing_0) = P(\varnothing_0) \tag{35}$$

$$f^+(\mathbf{x}_0) = q(\mathbf{x}_0). \tag{36}$$

The predicted probability of being in the null state is:

$$\begin{aligned} P^-(\varnothing_k) = & P^+(\varnothing_{k-1}) P(\varnothing_k \mid \varnothing_{k-1}) \\ & + \left[1 - P^+(\varnothing_{k-1})\right] P(\varnothing_k \mid \mathbb{X}_{k-1}), \end{aligned} \tag{37}$$

and the predicted pdf on H_1 is found from:

$$\begin{aligned} f^-(\mathbf{x}_k) = & \frac{P^+(\varnothing_{k-1})\,[1 - P(\varnothing_k \mid \varnothing_{k-1})]}{[1 - P^-(\varnothing_k)]} q(\mathbf{x}_k \mid \varnothing_{k-1}) \\ & + \frac{[1 - P^+(\varnothing_{k-1})]\,[1 - P(\varnothing_k \mid \mathbb{X}_{k-1})]}{[1 - P^-(\varnothing_k)]} \bullet \\ & \int q(\mathbf{x}_k \mid \mathbf{x}_{k-1}; \boldsymbol{\theta}_k) f^+(\mathbf{x}_{k-1}) d\mathbf{x}_{k-1}. \end{aligned} \tag{38}$$

The information update is given by:

$$\begin{aligned} D(\mathbf{Z}_k) = & P^-(\varnothing_k) f(\mathbf{z}_k \mid \varnothing_k; \boldsymbol{\theta}_k) \\ & + \left[1 - P^-(\varnothing_k)\right] \int f(\mathbf{z}_k \mid \mathbf{x}_k; \boldsymbol{\theta}_k) f^-(\mathbf{x}_k) d\mathbf{x}_k \end{aligned} \tag{39}$$

$$P^+(\varnothing_k) = \frac{P^-(\varnothing_k) f(\mathbf{z}_k \mid \varnothing_k; \boldsymbol{\theta}_k)}{D(\mathbf{Z}_k)} \tag{40}$$

$$f^+(\mathbf{x}_k) = \frac{f(\mathbf{z}_k \mid \mathbf{x}_k; \boldsymbol{\theta}_k) f^-(\mathbf{x}_k)}{\int f(\mathbf{z}_k \mid \mathbf{x}_k; \boldsymbol{\theta}_k) f^-(\mathbf{x}_k) d\mathbf{x}_k}. \tag{41}$$

For target detection, we assume that the state estimate takes one of two values, $\hat{\mathbf{x}}(\mathbf{Z}_k) = \varnothing$ (H_0) or $\hat{\mathbf{x}}(\mathbf{Z}_k) \in \mathbb{X}$ (H_1), and the estimation problem becomes a binary detection problem. We assume the standard binary detection cost function [21], [27]:

$$\begin{aligned} C\left(\hat{\mathbf{x}}(\mathbf{Z}_k), \mathbf{x}_k\right) = & C_{00} \mathbb{I}\left\{\hat{\mathbf{x}}(\mathbf{Z}_k) = \varnothing, \mathbf{x}_k = \varnothing\right\} \\ & + C_{10} \mathbb{I}\left\{\hat{\mathbf{x}}(\mathbf{Z}_k) \in \mathbb{X}, \mathbf{x}_k = \varnothing\right\} \\ & + C_{01} \mathbb{I}\left\{\hat{\mathbf{x}}(\mathbf{Z}_k) = \varnothing, \mathbf{x}_k \in \mathbb{X}\right\} \\ & + C_{11} \mathbb{I}\left\{\hat{\mathbf{x}}(\mathbf{Z}_k) \in \mathbb{X}, \mathbf{x}_k \in \mathbb{X}\right\}, \end{aligned} \tag{42}$$

where $\mathbb{I}\{\cdot\}$ is the indicator function that is equal to one when its argument is true and zero otherwise, and C_{ij} is the cost for deciding H_i when H_j is true. The optimal state estimator (decision rule), found by minimizing the conditional Bayes risk in (4), is the *Bayesian integrated likelihood ratio test* (BLRT) [21]:

$$\overline{\Lambda}(\mathbf{Z}_k) \underset{\hat{\mathbf{x}} = \varnothing}{\overset{\hat{\mathbf{x}} \in \mathbb{X}}{\gtrless}} \tau = \frac{C_{10} - C_{00}}{C_{01} - C_{11}}, \tag{43}$$

where $\overline{\Lambda}(\mathbf{Z}_k)$ is the *Bayesian integrated likelihood ratio* (BLR):

$$\overline{\Lambda}(\mathbf{Z}_k) = \frac{[1 - P^-(\varnothing_k)] \int f(\mathbf{z}_k \mid \mathbf{x}_k; \boldsymbol{\theta}_k) f^-(\mathbf{x}_k) d\mathbf{x}_k}{P^-(\varnothing_k) f(\mathbf{z}_k \mid \varnothing_k; \boldsymbol{\theta}_k)}. \tag{44}$$

Using (39) and (40), the BLR may also be written as:

$$\overline{\Lambda}(\mathbf{Z}_k) = \frac{1 - P^+(\varnothing_k)}{P^+(\varnothing_k)}. \tag{45}$$

Rearranging (45), we can express $P^+(\varnothing_k)$ in terms of $\overline{\Lambda}(\mathbf{Z}_k)$ as follows:

$$P^+(\varnothing_k) = (1 + \overline{\Lambda}(\mathbf{Z}_k))^{-1}. \tag{46}$$

The BLR is computed after the motion update using (44). Processing continues with the information update, which can be computed from (46) and (41).

In the BLRT, the BLR is compared to a threshold τ. If it exceeds the threshold, we say the target is present and if it is below the threshold, we say the target is absent. Note that if we define the posterior probability of the target being present as:

$$P^+(\mathbb{X}_k) \equiv 1 - P^+(\varnothing_k) = \frac{\overline{\Lambda}(\mathbf{Z}_k)}{1 + \overline{\Lambda}(\mathbf{Z}_k)}, \tag{47}$$

then an equivalent test to the BLRT in (43) is:

$$P^+(\mathbb{X}_k) \underset{\hat{\mathbf{x}} = \varnothing}{\overset{\hat{\mathbf{x}} \in \mathbb{X}}{\gtrless}} \gamma = \frac{\tau}{1 + \tau}. \tag{48}$$

Thus, the BLRT is equivalent to computing the posterior probability of the target being present and comparing it to a threshold.

B. Simplified Recursion

If we assume that the transition probabilities are defined such that the motion update does not affect the probability of being in the null state, then (37) becomes:

$$P^-(\varnothing_k) = P^+(\varnothing_{k-1}). \tag{49}$$

Under this assumption, we obtain the "simplified recursion" in [21]. The predicted pdf in (38) simplifies to:

$$f^-(\mathbf{x}_k) = P(\varnothing_k \,|\, \mathbb{X}_{k-1}) q(\mathbf{x}_k \,|\, \varnothing_{k-1}) + [1 - P(\varnothing_k \,|\, \mathbb{X}_{k-1})] \bullet \int q(\mathbf{x}_k \,|\, \mathbf{x}_{k-1}; \boldsymbol{\theta}_k) f^+(\mathbf{x}_{k-1}) d\mathbf{x}_{k-1}, \quad (50)$$

and the BLR in (44) simplifies to:

$$\overline{\Lambda}(\mathbf{Z}_k) = \overline{\Lambda}(\mathbf{Z}_{k-1}) \overline{\mathcal{L}}(\mathbf{z}_k \,|\, \mathbf{Z}_{k-1}), \quad (51)$$

where $\overline{\mathcal{L}}(\mathbf{z}_k \,|\, \mathbf{Z}_{k-1})$ is the *integrated likelihood ratio* (ILR) for the current data, defined as:

$$\overline{\mathcal{L}}(\mathbf{z}_k \,|\, \mathbf{Z}_{k-1}) = \int \frac{f(\mathbf{z}_k \,|\, \mathbf{x}_k; \boldsymbol{\theta}_k)}{f(\mathbf{z}_k \,|\, \varnothing_k; \boldsymbol{\theta}_k)} f^-(\mathbf{x}_k) d\mathbf{x}_k. \quad (52)$$

Substituting (49) into (37), the simplified recursion assumption in (49) is satisfied when

$$P^+(\varnothing_{k-1}) [1 - P(\varnothing_k \,|\, \varnothing_{k-1})] = \left[1 - P^+(\varnothing_{k-1})\right] P(\varnothing_k \,|\, \mathbb{X}_{k-1}), \quad (53)$$

or equivalently when[3]

$$\frac{1 - P(\varnothing_k \,|\, \varnothing_{k-1})}{P(\varnothing_k \,|\, \mathbb{X}_{k-1})} = \frac{1 - P^+(\varnothing_{k-1})}{P^+(\varnothing_{k-1})} = \overline{\Lambda}(\mathbf{Z}_{k-1}). \quad (54)$$

If the BLR $\overline{\Lambda}(\mathbf{Z}_{k-1}) \geq 1$, we can satisfy (54) by setting $P(\varnothing_k \,|\, \varnothing_{k-1}) = 0$ and $P(\varnothing_k \,|\, \mathbb{X}_{k-1}) = \overline{\Lambda}(\mathbf{Z}_{k-1})^{-1}$. However, if $\overline{\Lambda}(\mathbf{Z}_{k-1}) < 1$, this would make $P(\varnothing_k \,|\, \mathbb{X}_{k-1})$ greater than one. If we restrict $P(\varnothing_k \,|\, \mathbb{X}_{k-1})$ to be less than some upper limit $\overline{P}_\varnothing$, then when $\overline{\Lambda}(\mathbf{Z}_{k-1}) \leq 1/\overline{P}_\varnothing$, (54) can be satisfied by setting $P(\varnothing_k \,|\, \mathbb{X}_{k-1}) = \overline{P}_\varnothing$ and $P(\varnothing_k \,|\, \varnothing_{k-1}) = 1 - \overline{P}_\varnothing \overline{\Lambda}(\mathbf{Z}_{k-1})$. Thus, the simplified recursion assumption can be satisfied at each time step by adjusting $P(\varnothing_k \,|\, \varnothing_{k-1})$ and $P(\varnothing_k \,|\, \mathbb{X}_{k-1})$ according to:

$$P(\varnothing_k \,|\, \varnothing_{k-1}) = \max\left\{0, 1 - \overline{P}_\varnothing \overline{\Lambda}(\mathbf{Z}_{k-1})\right\} \quad (55)$$

$$P(\varnothing_k \,|\, \mathbb{X}_{k-1}) = \min\left\{\overline{\Lambda}(\mathbf{Z}_{k-1})^{-1}, \overline{P}_\varnothing\right\}. \quad (56)$$

While it might seem improper to adjust the probability of transitioning to the null state during the recursion, it is actually quite a reasonable thing to do. When the target is absent and the BLRT is small, $P(\varnothing_k \,|\, \varnothing_{k-1})$ will be close to one. However, if the BLRT is not small, the probability drops, and the tracker is less inclined to stay in the null state. Similarly, when the target is present and the BLRT is large, $P(\varnothing_k \,|\, \mathbb{X}_{k-1})$ will be close to zero. If the BLRT is not large, the probability increases, and the tracker is more inclined to transition to the null state.

To summarize, (35), (36), (41), (46), (49)–(52), and (56) specify the simplified Bayes-Markov recursion and state estimator (decision rule) for the target detection and track initiation problem.

[3]Equation (53) is also satisfied when $P(\varnothing_k \,|\, \varnothing_{k-1}) = 1$ and $P(\varnothing_k \,|\, \mathbb{X}_{k-1}) = 0$, i.e., when the probability of transitioning between H_0 and H_1 is zero.

C. Predicted Conditional Bayes Risk

If we assume $C_{00} = C_{11} = 0$ (i.e., there is no cost for making a correct decision), the predicted conditional Bayes risk in (7) is given by:

$$\begin{aligned} R_C^\uparrow(\boldsymbol{\theta}_k \,|\, \mathbf{Z}_{k-1}; \boldsymbol{\Theta}_{k-1}) &= E^\uparrow\{C(\hat{\mathbf{x}}(\mathbf{Z}_k), \mathbf{x}_k)\} \\ &= C_{10} P^-(\varnothing_k) P_F\left(\boldsymbol{\theta}_k \,|\, \varnothing_k; \tau \overline{\Lambda}(\mathbf{Z}_{k-1})^{-1}\right) \\ &\quad + C_{01}\left[1 - P^-(\varnothing_k)\right] P_M\left(\boldsymbol{\theta}_k \,|\, \mathbb{X}_k; \tau \overline{\Lambda}(\mathbf{Z}_{k-1})^{-1}\right), \end{aligned} \quad (57)$$

where $P_F\left(\boldsymbol{\theta}_k \,|\, \varnothing_k; \tau \overline{\Lambda}(\mathbf{Z}_{k-1})^{-1}\right)$ is the probability of false alarm for the current data ILR test when the threshold is $\tau \overline{\Lambda}(\mathbf{Z}_{k-1})^{-1}$, i.e.,

$$\begin{aligned} &P_F\left(\boldsymbol{\theta}_k \,|\, \varnothing_k; \tau \overline{\Lambda}(\mathbf{Z}_{k-1})^{-1}\right) \\ &\quad = E_{\mathbf{z}_k \,|\, \varnothing_k; \boldsymbol{\theta}_k}\left[\mathbb{I}\left\{\overline{\mathcal{L}}(\mathbf{z}_k \,|\, \mathbf{Z}_{k-1}) \geq \frac{\tau}{\overline{\Lambda}(\mathbf{Z}_{k-1})}\right\}\right] \\ &\quad = \Pr\left[\overline{\mathcal{L}}(\mathbf{z}_k \,|\, \mathbf{Z}_{k-1}) \geq \frac{\tau}{\overline{\Lambda}(\mathbf{Z}_{k-1})} \,\middle|\, \varnothing_k; \boldsymbol{\theta}_k\right] \end{aligned} \quad (58)$$

and $P_M\left(\boldsymbol{\theta}_k \,|\, \mathbb{X}_k; \tau \overline{\Lambda}(\mathbf{Z}_{k-1})^{-1}\right)$ is the probability of missed detection, averaged over the predicted density,

$$\begin{aligned} &P_M\left(\boldsymbol{\theta}_k \,|\, \mathbb{X}_k; \tau \overline{\Lambda}(\mathbf{Z}_{k-1})^{-1}\right) \\ &\quad = E_k^-\left\{E_{\mathbf{z}_k \,|\, \mathbf{x}_k; \boldsymbol{\theta}_k}\left[\mathbb{I}\left\{\overline{\mathcal{L}}(\mathbf{z}_k \,|\, \mathbf{Z}_{k-1}) < \frac{\tau}{\overline{\Lambda}(\mathbf{Z}_{k-1})}\right\}\right]\right\} \\ &\quad = E_k^-\left\{\Pr\left[\overline{\mathcal{L}}(\mathbf{z}_k \,|\, \mathbf{Z}_{k-1}) < \frac{\tau}{\overline{\Lambda}(\mathbf{Z}_{k-1})} \,\middle|\, \mathbf{x}_k; \boldsymbol{\theta}_k\right]\right\}. \end{aligned} \quad (59)$$

These probabilities are generally difficult to calculate and we must resort to an approximation or surrogate function to perform the controller optimization. Some possibilities include the expected Kullback-Leibler and Renyi divergences [14], [15] and the entropy [18]. We take an alternative approach and use the same criterion that we used in the single target tracking problem, namely to minimize the trace of the PC-CRLB. This maximizes Bayesian information, which is useful for making statistical inferences about $\mathbf{x}_k$ in both the estimation and detection settings. Thus,

$$R_C^\uparrow(\boldsymbol{\theta}_k \,|\, \mathbf{Z}_{k-1}; \boldsymbol{\Theta}_{k-1}) \Rightarrow \mathrm{tr}\left\{\mathbf{B}_k^\uparrow(\boldsymbol{\theta}_k \,|\, \mathbf{Z}_{k-1}; \boldsymbol{\Theta}_{k-1})^{-1}\right\}, \quad (60)$$

where we use the notation $\Rightarrow$ to denote replacement by a surrogate function.

D. Track Initiation and Termination

For track initiation, we initially assume the target is absent and initialize $\overline{\Lambda}(\mathbf{Z}_0)$ to some small value well below the target-present detection threshold τ_P. If a target is present, then the BLR will grow over time as evidence in favor of H_1 is accumulated. Eventually it will cross the threshold and the target will be declared present. If the target is absent, the BLR will decrease over time. However, if we allow the BLR to decrease below the initial value during periods when there is no target, then when a target appears and the BLR starts to grow, it will take longer to detect. To prevent this, we restrict the BLR to stay at or above the initial value, which we denote as $\overline{\Lambda}_{\min}$.

Once the target is declared present, processing continues in the same manner, except now we assume the target is present and restrict the BLR to stay at or below some maximum value $\overline{\Lambda}_{\max}$, which is well above the target-absent detection threshold τ_A. To allow some robustness in the system, we set τ_A to be less

than τ_P. This way, once the BLR exceeds τ_P and the target is declared present, sufficient evidence in favor of H_0 must accumulate before the BLR drops below τ_A and the target is declared absent again.

Once the target is declared present, we can extract a target state estimate after the information update. Thus, the single target tracking recursion can be performed simultaneously with the track initiation/termination recursion. The cognitive radar tracking recursion is summarized in Table I.

E. Summary

The cognitive radar recursion summarized in Table I provides a general framework for target tracking that can be applied to a wide variety of tracking problems. The framework mimics the perception-action cycle and includes sensing in the transmitter and receiver; processing in the detector and tracker; perception in the conversion of sensor data to the posterior density of the state vector; memory of all the past data in the posterior density; prediction in the PC-CRLB, which predicts the performance of the next measurement; decision-making in the controller, which decides on the next values for the sensor parameters based on the predicted performance; and attention in the sensor as a result of optimal sensor parameter selection.

For a specific application, we would need to specify the components of the state vector and the motion and measurement models, and evaluate the FIM from the measurement model. We would also need to specify the sensor parameters being controlled, their costs or constraints, and the form of the controller loss function that includes the PC-BIM and sensor costs. Finally, we would need to specify the implementation details that include the type of tracker used to implement the Bayes-Markov recursions and the method for solving the controller optimization problem. These details are provided for specific problems in [2], [3], [8]–[20]. Although these problems differ in the target state and motion models, sensor parameters being controlled, and Bayesian filter implementations, they are all examples that fit within the general framework.

V. Distributed Sensor Example

In this section, we demonstrate the how the general framework can be applied to a distributed sensor problem. We first define the model for this problem, then discuss the implementation, and finally provide some simulation results.

A. Model

In this example, the target state is the two-dimensional position in the xy-plane,

$$\mathbf{x}_k = \begin{bmatrix} x_k & y_k \end{bmatrix}^T. \tag{61}$$

The "sensor" consists of N independent sensors which produce noisy target angle of arrival estimates. Let (x_n, y_n) denote the position of the nth sensor.

The angle and range to the target are:

$$\phi_n(\mathbf{x}_k) = \arctan\left(\frac{y_k - y_n}{x_k - x_n}\right) \tag{62}$$

$$r_n(\mathbf{x}_k) = \sqrt{(x_k - x_n)^2 + (y_k - y_n)^2}. \tag{63}$$

A typical scenario is shown in Fig. 2.

TABLE I
Cognitive Radar Tracking Recursion

Initialization

Declare *target absent*

$\overline{\Lambda}(\mathbf{Z}_0) = \overline{\Lambda}_{\min}$

$P^+(\varnothing_0) = (1 + \overline{\Lambda}(\mathbf{Z}_0))^{-1}$

$f^+(\mathbf{x}_0) = q(\mathbf{x}_0)$

Motion Update - Part I

$P(\varnothing_k|\mathbb{X}_{k-1}) = \min\left\{\overline{\Lambda}(\mathbf{Z}_{k-1})^{-1}, \overline{P}_{\varnothing}\right\}$

$P^-(\varnothing_k) = P^+(\varnothing_{k-1})$

Controller Optimization

$$f^-(\mathbf{x}_k;\boldsymbol{\theta}) = P(\varnothing_k|\mathbb{X}_{k-1})q(\mathbf{x}_k|\varnothing_{k-1}) + [1 - P(\varnothing_k|\mathbb{X}_{k-1})] \cdot \int q(\mathbf{x}_k|\mathbf{x}_{k-1};\boldsymbol{\theta}) f^+(\mathbf{x}_{k-1}) d\mathbf{x}_{k-1}$$

$$\mathbf{B}_k^-(\boldsymbol{\theta}|\mathbf{Z}_{k-1};\boldsymbol{\Theta}_{k-1}) \cong \boldsymbol{\Sigma}_k^-(\boldsymbol{\theta}|\mathbf{Z}_{k-1};\boldsymbol{\Theta}_{k-1})^{-1}$$

$$\mathbf{J}_{\mathbf{x}}(\mathbf{x}_k;\boldsymbol{\theta}) = -E_{\mathbf{z}_k|\mathbf{x}_k;\boldsymbol{\theta}}\left\{\boldsymbol{\Delta}_{\mathbf{x}_k}^{\mathbf{x}_k} \ln f(\mathbf{z}_k|\mathbf{x}_k;\boldsymbol{\theta})\right\}$$

$$\mathbf{J}_k^-(\boldsymbol{\theta}|\mathbf{Z}_{k-1};\boldsymbol{\Theta}_{k-1}) = E_k^-\{\mathbf{J}_{\mathbf{x}}(\mathbf{x}_k;\boldsymbol{\theta})\}$$

$$\mathbf{B}_k^{\uparrow}(\boldsymbol{\theta}|\mathbf{Z}_{k-1};\boldsymbol{\Theta}_{k-1}) = \mathbf{B}_k^-(\boldsymbol{\theta}|\mathbf{Z}_{k-1};\boldsymbol{\Theta}_{k-1}) + \mathbf{J}_k^-(\boldsymbol{\theta}|\mathbf{Z}_{k-1};\boldsymbol{\Theta}_{k-1})$$

$$\boldsymbol{\theta}_k = \arg\min_{\boldsymbol{\theta}} L\left(\text{tr}\left\{\mathbf{B}_k^{\uparrow}(\boldsymbol{\theta}|\mathbf{Z}_{k-1};\boldsymbol{\Theta}_{k-1})^{-1}\right\}, R_{\Theta}(\boldsymbol{\theta})\right)$$

Motion Update - Part II

$$f^-(\mathbf{x}_k) = P(\varnothing_k|\mathbb{X}_{k-1})q(\mathbf{x}_k|\varnothing_{k-1}) + [1 - P(\varnothing_k|\mathbb{X}_{k-1})] \cdot \int q(\mathbf{x}_k|\mathbf{x}_{k-1};\boldsymbol{\theta}_k) f^+(\mathbf{x}_{k-1}) d\mathbf{x}_{k-1}$$

Measurement

Obtain measurement $\mathbf{z}_k$ according to θ_k

BLR

$$\overline{\mathcal{L}}(\mathbf{z}_k|\mathbf{Z}_{k-1}) = \int \frac{f(\mathbf{z}_k|\mathbf{x}_k;\boldsymbol{\theta}_k)}{f(\mathbf{z}_k|\varnothing_k;\boldsymbol{\theta}_k)} f^-(\mathbf{x}_k) d\mathbf{x}_k$$

$$\overline{\Lambda}(\mathbf{Z}_k) = \max\left\{\overline{\Lambda}_{\min}, \min\left\{\overline{\Lambda}_{\max}, \overline{\Lambda}(\mathbf{Z}_{k-1})\overline{\mathcal{L}}(\mathbf{z}_k|\mathbf{Z}_{k-1})\right\}\right\}$$

Information Update

$$P^+(\varnothing_k) = (1 + \overline{\Lambda}(\mathbf{Z}_k))^{-1}$$

$$f^+(\mathbf{x}_k) = \frac{f(\mathbf{z}_k|\mathbf{x}_k;\boldsymbol{\theta}_k) f^-(\mathbf{x}_k)}{\int f(\mathbf{z}_k|\mathbf{x}_k;\boldsymbol{\theta}_k) f^-(\mathbf{x}_k) d\mathbf{x}_k}$$

BLRT

if *target absent* and $\overline{\Lambda}(\mathbf{Z}_k) \geq \tau_P$
  Declare *target present*, initiate track
elseif *target present* and $\overline{\Lambda}(\mathbf{Z}_k) < \tau_A$
  Declare *target absent*, terminate track
end

Track Estimate (if *target present*)

$$\hat{\mathbf{x}}_k^+ = \boldsymbol{\mu}_k^+ = E_k^+\{\mathbf{x}_k\}$$

$$\boldsymbol{\Sigma}_k^+ = E_k^+\left\{\left[\mathbf{x}_k - \boldsymbol{\mu}_k^+\right]\left[\mathbf{x}_k - \boldsymbol{\mu}_k^+\right]^T\right\}$$

We assume that there is a fixed amount of observation time that can be allocated among the sensors. Let $\theta_{n,k}$ denote the

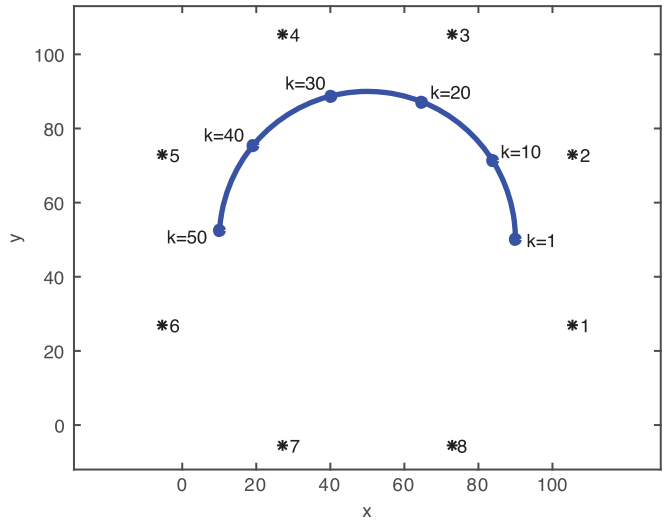

Fig. 2. Sensor placement and target path.

fraction of observation time allocated to the nth sensor and let $\boldsymbol{\theta}_k \equiv [\,\theta_{1,k} \quad \cdots \quad \theta_{N,k}\,]^T$ denote the sensor parameter vector. Each $\theta_{n,k}$ must be between zero and one, inclusive, and their sum must be less than or equal to one. We do not impose any cost to using the sensors within these constraints, thus the cost function has the form shown in (10) and is given by:

$$R_\Theta(\boldsymbol{\theta}_k) = \begin{cases} 0 & 0 \le \theta_{n,k} \le 1; n = 1, \ldots, N \\ & \sum_{n=1}^{N} \theta_{n,k} \le 1 \\ \infty & \text{otherwise.} \end{cases} \tag{64}$$

The motion model consists of a prior density characterized by $P(\varnothing_0)$ and $q(\mathbf{x}_0)$ and a transition density with four components, given in (32). We assume an initial state distribution of the form $\mathbf{x}_0 \sim N(\bar{\mathbf{x}}_0, \boldsymbol{\Omega}_0)$, therefore

$$q(\mathbf{x}_0) = \frac{\exp\left\{-\frac{1}{2}(\mathbf{x}_0 - \bar{\mathbf{x}}_0)^T \boldsymbol{\Omega}_0^{-1}(\mathbf{x}_0 - \bar{\mathbf{x}}_0)\right\}}{2\pi\sqrt{|\boldsymbol{\Omega}_0|}}, \tag{65}$$

and a "nearly constant" motion model,[4]

$$\mathbf{x}_k = \mathbf{x}_{k-1} + \mathbf{v}_k, \tag{66}$$

where $\mathbf{v}_k \sim N(\mathbf{0}, \mathbf{Q})$ and $\mathbf{Q} = \sigma_x^2 \mathbf{I}$, so that

$$q(\mathbf{x}_k \,|\, \mathbf{x}_{k-1}) = \frac{\exp\left\{-\|\mathbf{x}_k - \mathbf{x}_{k-1}\|^2 / 2\sigma_x^2\right\}}{2\pi\sigma_x^2}. \tag{67}$$

In this model, transition density does not depend on $\boldsymbol{\theta}_k$, so we have dropped it from the notation. We define $q(\mathbf{x}_k \,|\, \varnothing_{k-1})$ to be the same as the initial target present distribution $q(\mathbf{x}_k \,|\, \varnothing_{k-1}) = q(\mathbf{x}_0) \sim N(\bar{\mathbf{x}}_0, \boldsymbol{\Omega}_0)$. $P(\varnothing_0)$ is set to some initial value and $P(\varnothing_k \,|\, \varnothing_{k-1})$ and $P(\varnothing_k \,|\, \mathbb{X}_{k-1})$ are adjusted at each time according to (55) and (56) so that the simplified recursion assumption in (49) is satisfied.

[4]The "nearly constant" motion model does not match the target motion shown in Fig. 2. The tracker rarely has knowledge of the actual motion model of the target and must specify a motion model that is a reasonable approximation to the presumed motion model, with enough process noise to accommodate model mismatch. We chose the "nearly constant" motion model for its simplicity.

The measurement model consists of the likelihood functions on H_0 and H_1. When $\theta_{n,k} > 0$, the observation at the nth sensor is the noisy angle estimate

$$z_{n,k} = \phi_n(\mathbf{x}_k) + w_{n,k}, \tag{68}$$

where $w_{n,k}$ is zero-mean additive white Gaussian noise whose variance is inversely proportional to the observation time on that sensor and whose value is truncated so that $z_{n,k}$ is restricted to the 180° (π radians) sector $[\phi_{n,0}, \phi_{n,0} + \pi)$ looking into the field of view from that sensor. The likelihood function for the nth sensor is a truncated Gaussian density when $\theta_{n,k} > 0$,

$$f_n(z_{n,k} \,|\, \mathbf{x}_k; \theta_{n,k}) = c_{n,k} \exp\left\{-\frac{(z_{n,k} - \phi_n(\mathbf{x}_k))^2}{2\sigma_n^2/\theta_{n,k}}\right\}, \tag{69}$$

where $c_{n,k}$ is the normalization constant,

$$c_{n,k}^{-1} = \int_{\phi_{n,0}}^{\phi_{n,0}+\pi} \exp\left\{-\frac{(z - \phi_n(\mathbf{x}_k))^2}{2\sigma_n^2/\theta_{n,k}}\right\} dz, \tag{70}$$

and a uniform density when $\theta_{n,k} = 0$:

$$f_n(z_{n,k} \,|\, \mathbf{x}_k; 0) = \frac{1}{\pi} \quad z_{n,k} \in [\phi_{n,0}, \phi_{n,0} + \pi)\,. \tag{71}$$

On H_0, the likelihood function for the nth sensor is uniform on $[\phi_{n,0}, \phi_{n,0} + \pi)$:

$$f_n(z_{n,k} \,|\, \varnothing_k; \theta_{n,k}) = \frac{1}{\pi} \quad z_{n,k} \in [\phi_{n,0}, \phi_{n,0} + \pi)\,. \tag{72}$$

The joint likelihood functions of the observation vector $\mathbf{z}_k \equiv [\,z_{1,k} \quad \cdots \quad z_{N,k}\,]^T$ are the products of the individual likelihood functions:

$$f(\mathbf{z}_k \,|\, \mathbf{x}_k; \boldsymbol{\theta}_k) = \prod_{n=1}^{N} f_n(z_{n,k} \,|\, \mathbf{x}_k; \theta_{n,k}) \tag{73}$$

$$f(\mathbf{z}_k \,|\, \varnothing_k; \boldsymbol{\theta}_k) = \prod_{n=1}^{N} f_n(z_{n,k} \,|\, \varnothing_k; \theta_{n,k}). \tag{74}$$

The FIM can be evaluated from the joint likelihood function on H_1. The FIM is the sum of sensor FIMs,

$$\mathbf{J}(\mathbf{x}_k; \boldsymbol{\theta}_k) = \sum_{n=1}^{N} \mathbf{J}_n(\mathbf{x}_k; \theta_{n,k}), \tag{75}$$

where $\mathbf{J}_n(\mathbf{x}_k; \theta_{n,k})$ is the FIM for the nth sensor. As shown in Appendix B, it has the form:

$$\mathbf{J}_n(\mathbf{x}_k; \theta_{n,k}) = \frac{\theta_{n,k}}{\sigma_n^2 r_n(\mathbf{x}_k)^2} \mathbf{u}_n(\mathbf{x}_k) \mathbf{u}_n(\mathbf{x}_k)^T, \tag{76}$$

where $\mathbf{u}_n(\mathbf{x}_k)$ is the unit vector perpendicular to the vector pointing from the sensor to the target,

$$\mathbf{u}_n(\mathbf{x}_k) \equiv \frac{1}{r_n(\mathbf{x}_k)} \begin{bmatrix} -(y_k - y_n) \\ (x_k - x_n) \end{bmatrix}. \tag{77}$$

The nth sensor FIM in (76) can be expressed as the product of the sensor parameter $\theta_{n,k}$ and a unit sensor FIM, $\mathbf{J}_n(\mathbf{x}_k)$, defined as the sensor FIM when $\theta_{n,k} = 1$,

$$\mathbf{J}_n(\mathbf{x}_k) \equiv \frac{1}{\sigma_n^2 r_n(\mathbf{x}_k)^2} \mathbf{u}_n(\mathbf{x}_k) \mathbf{u}_n(\mathbf{x}_k)^T. \tag{78}$$

The total FIM can then be expressed as a weighted sum of unit sensor FIMs,

$$\mathbf{J}(\mathbf{x}_k;\boldsymbol{\theta}_k)=\sum_{n=1}^{N}\theta_{n,k}\mathbf{J}_n(\mathbf{x}_k). \tag{79}$$

Consequently, the EFIM can also be expressed as a weighted sum of unit sensor EFIMs,

$$\mathbf{J}^{-}(\boldsymbol{\theta}_k\,|\,\mathbf{Z}_{k-1};\boldsymbol{\Theta}_{k-1})=\sum_{n=1}^{N}\theta_{n,k}\mathbf{J}_n^{-}(\mathbf{Z}_{k-1};\boldsymbol{\Theta}_{k-1}), \tag{80}$$

where the unit sensor EFIM is defined as

$$\mathbf{J}_n^{-}(\mathbf{Z}_{k-1};\boldsymbol{\Theta}_{k-1})\equiv E^{-}\left\{\mathbf{J}_n(\mathbf{x}_k)\right\}. \tag{81}$$

The PC-BIM is therefore the PIM plus a weighted sum of unit EFIMS,

$$\begin{aligned}\mathbf{B}_k^{\uparrow}(\boldsymbol{\theta}_k\,|\,\mathbf{Z}_{k-1};\boldsymbol{\Theta}_{k-1})=&\mathbf{B}_k^{-}(\mathbf{Z}_{k-1};\boldsymbol{\Theta}_{k-1})\\&+\sum_{n=1}^{N}\theta_{n,k}\mathbf{J}_n^{-}(\mathbf{Z}_{k-1};\boldsymbol{\Theta}_{k-1}).\end{aligned} \tag{82}$$

We assume the controller loss function is the sum of the predicted conditional Bayes risk and the sensor cost as in (11). Substituting (60) and (64) into (11), the controller optimization problem in (9) becomes a constrained optimization problem as in (13):

$$\begin{aligned}\boldsymbol{\theta}_k=\arg\min_{\boldsymbol{\theta}}\ &\operatorname{tr}\left\{\mathbf{B}_k^{\uparrow}(\boldsymbol{\theta}\,|\,\mathbf{Z}_{k-1};\boldsymbol{\Theta}_{k-1})^{-1}\right\}\\&s.t.\ 0\le\theta_n\le 1;n=1,\ldots,N;\sum_{n=1}^{N}\theta_n\le 1.\end{aligned} \tag{83}$$

B. Implementation

In this section, we provide the implementation details that include the tracking algorithm used to implement the Bayes-Markov recursions and the method for solving the controller optimization problem.

We start with the controller optimization problem in (83). It doesn't have a convenient closed form solution due to the non-negativity constraints, therefore we develop an iterative solution. Let $G(\boldsymbol{\theta})$ denote the objective function in (83),

$$G(\boldsymbol{\theta})\equiv\operatorname{tr}\left\{\mathbf{B}_k^{\uparrow}(\boldsymbol{\theta}\,|\,\mathbf{Z}_{k-1};\boldsymbol{\Theta}_{k-1})^{-1}\right\}, \tag{84}$$

where $\mathbf{B}_k^{\uparrow}(\boldsymbol{\theta}\,|\,\mathbf{Z}_{k-1};\boldsymbol{\Theta}_{k-1})$ is given by (82). We compute the gradient using the property ([32], App. A):

$$\frac{\partial}{\partial\theta_n}\operatorname{tr}\left\{\mathbf{B}(\boldsymbol{\theta})^{-1}\right\}=-\operatorname{tr}\left\{\mathbf{B}(\boldsymbol{\theta})^{-2}\frac{\partial}{\partial\theta_n}\mathbf{B}(\boldsymbol{\theta})\right\} \tag{85}$$

to obtain

$$\begin{aligned}g_n(\boldsymbol{\theta})\equiv&\frac{\partial}{\partial\theta_n}G(\boldsymbol{\theta})\\=&-\operatorname{tr}\left\{\mathbf{B}_k^{\uparrow}(\boldsymbol{\theta}\,|\,\mathbf{Z}_{k-1};\boldsymbol{\Theta}_{k-1})^{-2}\mathbf{J}_n^{-}(\mathbf{Z}_{k-1};\boldsymbol{\Theta}_{k-1})\right\}.\end{aligned} \tag{86}$$

The gradients are always nonpositive, thus the objective function decreases when any θ_n is made larger and increases when it is made smaller. The sensor parameters will sum to one in an optimal solution since there is always incentive to make the sensor parameters as large as possible. Let $\boldsymbol{\theta}^{(i)}$ denote the value of the sensor parameter vector at the ith iteration. We initialize with the previous optimal sensor parameter vector, $\boldsymbol{\theta}^{(0)}=\boldsymbol{\theta}_{k-1}$. At the ith iteration, we assume that $\boldsymbol{\theta}^{(i-1)}$ satisfies the constraints and adjust two components at a time by an equal amount Δ_i (one up and one down) so that the constraints remain satisfied and the objective function is decreased. We do this by choosing the sensors with the largest and smallest gradients so that the net change in the objective function is negative. At each iteration, Δ_i is chosen to balance convergence behavior with the requirement that all sensor parameters stay between zero and one.

We implement the tracker on a discrete 2D grid with M points, denoted as $\boldsymbol{\chi}_m; m=1,\ldots,M$. Let $q_0(\boldsymbol{\chi}_m)$, $q(\boldsymbol{\chi}_m\,|\,\boldsymbol{\chi}_r)$, and $q(\boldsymbol{\chi}_m\,|\,\varnothing)$ denote the discrete approximations to the prior and transition densities, evaluated at the grid points $\boldsymbol{\chi}_m; m=1,\ldots,M$ and $\boldsymbol{\chi}_r; r=1,\ldots,M$, respectively. They are computed from:

$$q_0(\boldsymbol{\chi}_m)=c_0\ q(\mathbf{x}_0)|_{\mathbf{x}_0=\boldsymbol{\chi}_m} \tag{87}$$

$$q(\boldsymbol{\chi}_m\,|\,\boldsymbol{\chi}_r)=c_r\ q(\mathbf{x}_k\,|\,\mathbf{x}_{k-1})|_{\mathbf{x}_k=\boldsymbol{\chi}_m,\mathbf{x}_{k-1}=\boldsymbol{\chi}_r} \tag{88}$$

$$q(\boldsymbol{\chi}_m\,|\,\varnothing)=c_t\ q(\mathbf{x}_k\,|\,\varnothing_{k-1})|_{\mathbf{x}_k=\boldsymbol{\chi}_m}, \tag{89}$$

where c_0, c_r, and c_t are normalization constants so that the discrete densities sum to one over $\boldsymbol{\chi}_m$.

The integrals on lines 7, 13, 15, and 18 and the expectations on lines 10, 20, and 21 in Table I are implemented as a discrete sums. Since the motion model does not depend on the sensor parameters, the motion updates on lines 7 and 13 of Table I can be combined into a single motion update prior to controller optimization.

Let $\hat{f}_k^{-}(\boldsymbol{\chi}_m)$ and $\hat{f}_k^{+}(\boldsymbol{\chi}_m)$ denote the discrete predicted and posterior densities at time k evaluated at the grid point $\boldsymbol{\chi}_m$, and let $\hat{\mathbf{x}}_k^{-}$, $\hat{\boldsymbol{\Sigma}}_k^{-}$, $\hat{\mathbf{x}}_k^{+}$, $\hat{\boldsymbol{\Sigma}}_k^{+}$, $\hat{\mathbf{B}}_k^{-}$, $\hat{\mathbf{J}}_{n,k}^{-}$, and $\hat{\mathbf{B}}_k^{\uparrow}(\boldsymbol{\theta})$ denote the discrete sum approximations to the predicted mean and covariance matrix, posterior mean and covariance matrix, PIM, sensor n EFIM, and PC-BIM, respectively. The implementation is summarized as follows:

Initialization

$$P^{+}(\varnothing_0)=P(\varnothing_0) \tag{90}$$

$$\hat{f}_0^{+}(\boldsymbol{\chi}_m)=q_0(\boldsymbol{\chi}_m) \tag{91}$$

$$\overline{\Lambda}(\mathbf{Z}_0)=\frac{1-P^{+}(\varnothing_0)}{P^{+}(\varnothing_0)} \tag{92}$$

Motion Update

$$P(\varnothing_k\,|\,\mathbb{X}_{k-1})=\min\left\{\overline{\Lambda}(\mathbf{Z}_{k-1})^{-1},\overline{P}_{\varnothing}\right\} \tag{93}$$

$$P^{-}(\varnothing_k)=P^{+}(\varnothing_{k-1}) \tag{94}$$

$$\begin{aligned}\hat{f}_k^{-}(\boldsymbol{\chi}_m)=&P(\varnothing_k\,|\,\mathbb{X}_{k-1})q(\boldsymbol{\chi}_m\,|\,\varnothing)\\&+[1-P(\varnothing_k\,|\,\mathbb{X}_{k-1})]\sum_{r=1}^{M}q(\boldsymbol{\chi}_m\,|\,\boldsymbol{\chi}_r)\hat{f}_k^{+}(\boldsymbol{\chi}_r)\end{aligned} \tag{95}$$

Controller Optimization

- Compute

$$\hat{\mathbf{x}}_k^{-}=\sum_{m=1}^{M}\boldsymbol{\chi}_m\hat{f}_k^{-}(\boldsymbol{\chi}_m) \tag{96}$$

$$\hat{\boldsymbol{\Sigma}}_k^{-}=\sum_{m=1}^{M}\left[\boldsymbol{\chi}_m-\hat{\mathbf{x}}_k^{-}\right]\left[\boldsymbol{\chi}_m-\hat{\mathbf{x}}_k^{-}\right]^{T}\hat{f}_k^{-}(\boldsymbol{\chi}_m) \tag{97}$$

$$\hat{\mathbf{B}}_k^- = \left[\hat{\mathbf{\Sigma}}_k^-\right]^{-1} \tag{98}$$

$$\hat{\mathbf{J}}_{n,k}^- = \sum_{m=1}^{M} \mathbf{J}_n(\boldsymbol{\chi}_m)\hat{f}_k^-(\boldsymbol{\chi}_m). \tag{99}$$

- Initialize: $\boldsymbol{\theta}^{(0)} = \boldsymbol{\theta}_{k-1}$
- For $i = 1, 2, \ldots$ until stopping criterion reached,
  1) Compute

$$\hat{\mathbf{B}}_k^{\uparrow}\left(\boldsymbol{\theta}^{(i-1)}\right) = \hat{\mathbf{B}}_k^- + \sum_{n=1}^{N} \theta_n^{(i-1)} \hat{\mathbf{J}}_{n,k}^- \tag{100}$$

$$g_n\left(\boldsymbol{\theta}^{(i-1)}\right) = -\mathrm{tr}\left\{\hat{\mathbf{B}}_k^{\uparrow}\left(\boldsymbol{\theta}^{(i-1)}\right)^{-2} \hat{\mathbf{J}}_{n,k}^-\right\} \tag{101}$$

  2) Choose n_1 to be the index of the sensor with the largest negative gradient
  3) Among sensors whose parameter value $\theta_n^{(i-1)} > 0$, choose n_2 to be the index of the sensor with the smallest negative gradient
  4) Update

$$\begin{aligned} \theta_{n_1}^{(i)} &= \theta_{n_1}^{(i-1)} + \Delta_i \\ \theta_{n_2}^{(i)} &= \theta_{n_2}^{(i-1)} - \Delta_i \end{aligned} \tag{102}$$

- Set $\boldsymbol{\theta}_k = \boldsymbol{\theta}^{(i)}$

Measurement

- Obtain $\mathbf{z}_k$ according to the allocation in $\boldsymbol{\theta}_k$

BLRT

$$\overline{\mathcal{L}}(\mathbf{z}_k \,|\, \mathbf{Z}_{k-1}) = \sum_{m=1}^{M} \frac{f(\mathbf{z}_k \,|\, \boldsymbol{\chi}_m; \boldsymbol{\theta}_k)}{f(\mathbf{z}_k \,|\, \varnothing_k; \boldsymbol{\theta}_k)} f_k^-(\boldsymbol{\chi}_m) \tag{103}$$

$$\overline{\Lambda}(\mathbf{Z}_k) = \overline{\Lambda}(\mathbf{Z}_{k-1})\overline{\mathcal{L}}(\mathbf{z}_k \,|\, \mathbf{Z}_{k-1}). \tag{104}$$

$$\overline{\Lambda}(\mathbf{Z}_k) \underset{\hat{\mathbf{x}}=\varnothing}{\overset{\hat{\mathbf{x}}\in\mathbb{X}}{\gtrless}} \gamma, \tag{105}$$

Information Update

$$P^+(\varnothing_k) = (1 + \overline{\Lambda}(\mathbf{Z}_k))^{-1} \tag{106}$$

$$\hat{f}_k^+(\boldsymbol{\chi}_m) = \frac{f(\mathbf{z}_k \,|\, \boldsymbol{\chi}_m; \boldsymbol{\theta}_k)\hat{f}_k^-(\boldsymbol{\chi}_m)}{\sum_{m'=1}^{M} f(\mathbf{z}_k \,|\, \boldsymbol{\chi}_{m'}; \boldsymbol{\theta}_k)\hat{f}_k^-(\boldsymbol{\chi}_{m'})}. \tag{107}$$

C. Simulation Results

The locations of the $N = 8$ sensors and the target path for $k = 0$ to 50 are shown in Fig. 2. At time $k = 50$, the target disappears. The parameter space is the square $[0, 100] \times [0, 100]$. The angles that define the field of view for the sensors are $\phi_{1,0} = \phi_{2,0} = \pi/2$, $\phi_{3,0} = \phi_{4,0} = -\pi$, $\phi_{5,0} = \phi_{6,0} = -\pi/2$, $\phi_{7,0} = \phi_{8,0} = 0$. For the prior distribution, $P(\varnothing_0) = 0.9$, and $q(\mathbf{x}_0) \sim N(\bar{\mathbf{x}}_0, \mathbf{\Omega}_0)$ with $\bar{\mathbf{x}}_0 = [50\ 50]^T$, and $\mathbf{\Omega}_0 = (30)^2\mathbf{I}$. The transition covariance matrix is $\mathbf{Q} = (16)^2\mathbf{I}$. There are 101 grid points in each dimension with a spacing of one unit for a total of $M = 10201$ grid points. We set the initial components of $\boldsymbol{\theta}_0$ all to be equal to 1/8.

To determine values for τ_A, τ_P, $\overline{\Lambda}_{\min}$, and $\overline{\Lambda}_{\max}$, we considered the equivalent test to the BLRT given in (48), in which the posterior probability that the target is present is compared to a threshold $\gamma = \tau/(1+\tau)$. We chose the target absent threshold as $\gamma_A = 0.5$ and the target present threshold as $\gamma_P \approx 1 - 10^{-6}$, which gives $\tau_A = 1$ and $\tau_P = 10^6$. We then chose $\overline{\Lambda}_{\min}$ and $\overline{\Lambda}_{\max}$ to be one order of magnitude less than τ_A and greater than τ_P, respectively, i.e., $\overline{\Lambda}_{\min} = 0.1$ and $\overline{\Lambda}_{\max} = 10^7$. $\overline{P}_{\varnothing}$ should be close to one and was chosen to be $\overline{P}_{\varnothing} = 0.9$.

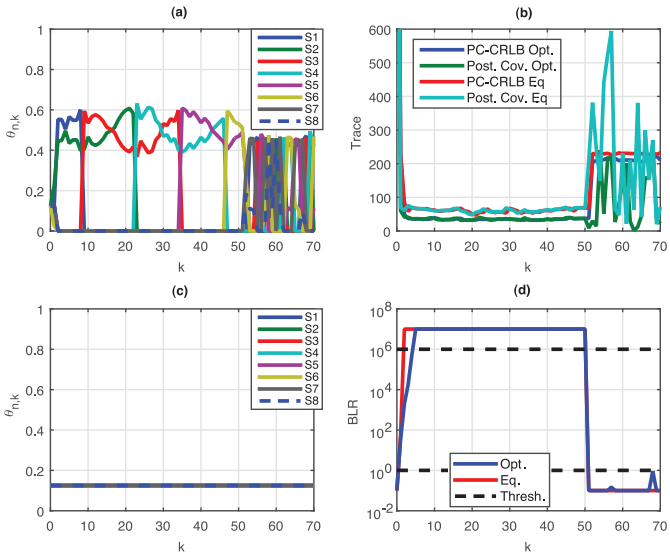

Fig. 3. Sensor parameters, PC-CRLB, and BLR for high resolution case. (a) Optimal Sensor Parameters, (b) Trace of PC-CRLB & Posterior Covariance Matrix, (c) Equal Sensor Parameters, (d) BLR.

We first consider a high resolution case where the unit measurement variance of the sensors is $\sigma_n^2 = (0.04\pi)^2$. Fig. 3(a) shows the optimal allocation of sensor resources over time and Fig. 3(c) shows an equal allocation of sensor resources. The trace of the PC-CRLB is shown in Fig. 3(b) and the BLR is shown in Fig. 3(d). Before any data is received, the optimal allocation is equal across all eight sensors. After a few iterations, the controller is able to extract enough information from the data to focus resources on sensors 1 and 2, which are closest to the target. As the target moves past sensors 2, 3, 4, and 5, the allocation shifts to an approximately equal allocation between the closest two sensors, thus the system is directing its attention to the target.

The PC-CRLB represents the theoretical performance before a measurement is taken. The actual trace of the posterior covariance matrix after the data is received is also plotted and shows good agreement with the PC-CRLB. For this high resolution case, the trace of the PC-CRLB decreases rapidly and the BLR increases rapidly as the first few measurements are taken. The BLR crosses the target-present threshold at $k = 5$. While the target is present, the trace of the PC-CRLB remains fairly constant and the BLR is capped at $\overline{\Lambda}_{\max}$. When the target disappears at $k = 50$, the BLR immediately drops below the target-absent threshold at $k = 51$, and the trace of the PC-CRLB increases significantly. In Fig. 3(b) we also show the traces of the PC-CRLB and posterior covariance matrix when the sensor allocation is not optimized, but remains constant throughout the track, and in Fig. 3(d), we also show the BLR of the static system. The BLR of the static system crosses the detection threshold earlier at $k = 2$, remains capped at $\overline{\Lambda}_{\max}$, and then drops immediately below the target-absent threshold at $k = 51$. Thus the track initiation and termination performance is similar, or even a little better than the cognitive

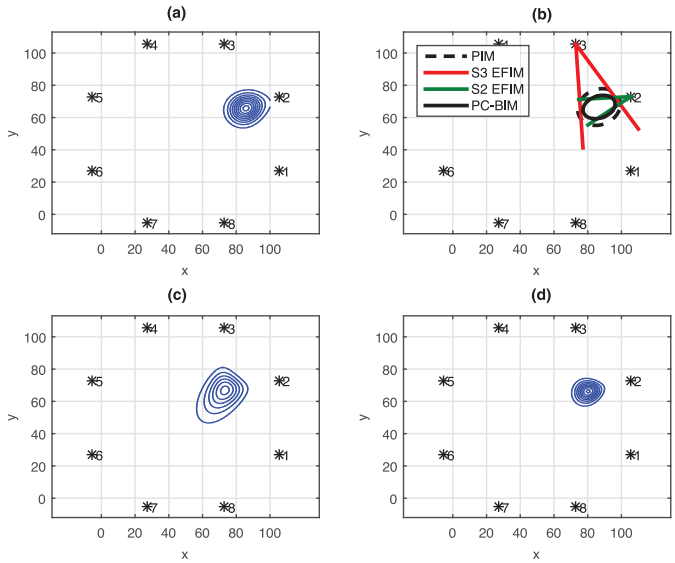

Fig. 4. Controller optimization for high resolution case at $k = 9$. (a) Predicted, (b) PC-BIM, $\theta_3 = 0.60$, $\theta_2 = 0.40$, (c) Likelihood function $\theta_3 = 0.60$, $\theta_2 = 0.40$, (d) Posterior.

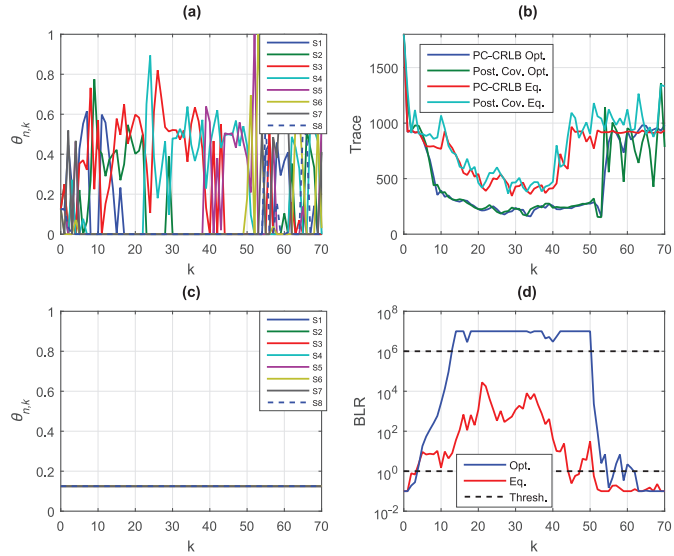

Fig. 5. Sensor parameters, PC-CRLB, and BLR for low resolution case. (a) Optimal Sensor Parameters, (b) Trace of PC-CRLB and Posterior Covariance Matrix, (c) Equal Sensor Parameters, (d) BLR.

system. However, the trace of the PC-CRLB is higher for the static system than the cognitive system while the target is under track.

Fig. 4 illustrates the controller optimization at time $k = 9$. Prior to obtaining the next sample, the predicted density is shown in Fig. 4(a). The PIM is equal to the inverse of the predicted covariance matrix and is represented by 2σ error ellipse shown in dashed black in Fig. 4(b). The controller allocates resources to sensors 2 and 3 with $\theta_2 = 0.40$ and $\theta_3 = 0.60$. This produces the 2σ resolution sectors determined by the sensor EFIMs shown in green and red, and the 2σ error ellipse of the PC-BIM shown in black. The sensor measurements are obtained and the resulting likelihood function is shown in Fig. 4(c). The posterior density is shown in Fig. 4(d).

Next we consider a low resolution case where the unit measurement variance of the sensors is $\sigma_n^2 = (0.18\pi)^2$. Fig. 5 shows the allocation of sensor resources, the trace of the PC-CRLB and posterior covariance matrices, and the BLR vs. time. In this case, it takes longer for the trace of the PC-CRLB to decrease and the BLR to increase. By about $k = 5$, enough information has accumulated that the controller can focus resources on sensors 1, 2, and 3, and the BLR increases rapidly. The target-present threshold is crossed at $k = 13$. While the target is present, the trace of the PC-CRLB remains fairly constant and the BLR is capped at $\overline{\Lambda}_{\max}$. When the target disappears, the BLR drops rapidly and crosses the target-absent threshold at $k = 55$, and the trace of the PC-CRLB increases. In this case, the static system shows significantly poorer performance. The trace of the PC-CRLB is much higher than for the cognitive system and the BLR never crosses the target-present threshold. These two examples show that the cognitive system makes better use of the system resources and achieves better tracking performance than the static system.

VI. Summary

In this paper, we have developed a general cognitive radar framework for a radar system engaged in target tracking. The model includes the radar sensor and the tracking processor and specifies the feedback mechanism and optimization criterion used to obtain the next set of sensor data. Both target detection and tracking were addressed. By separating the general principles from the specific application and implementation details, our formulation provides a flexible framework applicable to the general tracking problem. The performance advantage of the cognitive system over a standard static system was demonstrated on a distributed sensor system.

Appendix A
Bayesian Cramér-Rao Lower Bound

Following [27] and [28] (see also [29]), suppose we wish to estimate a random vector $\mathbf{x}$ based on an observation vector $\mathbf{z}$. Let $f(\mathbf{x})$, $f(\mathbf{z}\,|\,\mathbf{x})$, and $f(\mathbf{x},\mathbf{z}) = f(\mathbf{z}\,|\,\mathbf{x})f(\mathbf{x})$ denote the prior density of $\mathbf{x}$, the conditional density of $\mathbf{z}$ given $\mathbf{x}$, and the joint density of $\mathbf{x}$ and $\mathbf{z}$, respectively. Let $\hat{\mathbf{x}}(\mathbf{z})$ denote an estimate of $\mathbf{x}$. The MSE matrix is defined as

$$\boldsymbol{\Sigma} \equiv E_{\mathbf{x},\mathbf{z}}\left\{[\hat{\mathbf{x}}(\mathbf{z}) - \mathbf{x}]\,[\hat{\mathbf{x}}(\mathbf{z}) - \mathbf{x}]^T\right\}, \tag{108}$$

where the expectation is taken with respect to $f(\mathbf{x},\mathbf{z})$.

The *Bayesian Cramér-Rao lower bound (BCRLB)* provides a lower bound on the MSE matrix of the form:

$$\boldsymbol{\Sigma} \succeq \mathbf{B}, \tag{109}$$

where $\mathbf{B}$ is the Bayesian information matrix (BIM) and $\mathbf{B}^{-1}$ is the BCRLB matrix. The BIM is defined as:

$$\mathbf{B} \equiv -E_{\mathbf{x},\mathbf{z}}\{\boldsymbol{\Delta}_{\mathbf{x}}^{\mathbf{x}} \ln f(\mathbf{x},\mathbf{z})\}. \tag{110}$$

The BIM can be expressed as the sum of two terms:

$$\begin{aligned}\mathbf{B} &= -E_{\mathbf{x},\mathbf{z}}\{\boldsymbol{\Delta}_{\mathbf{x}}^{\mathbf{x}} \ln f(\mathbf{x})\} - E_{\mathbf{x},\mathbf{z}}\{\boldsymbol{\Delta}_{\mathbf{x}}^{\mathbf{x}} \ln f(\mathbf{z}\,|\,\mathbf{x})\}\\ &= -E_{\mathbf{x}}\{\boldsymbol{\Delta}_{\mathbf{x}}^{\mathbf{x}} \ln f(\mathbf{x})\} - E_{\mathbf{x}}\left\{E_{\mathbf{z}\,|\,\mathbf{x}}\{\boldsymbol{\Delta}_{\mathbf{x}}^{\mathbf{x}} \ln f(\mathbf{z}\,|\,\mathbf{x})\}\right\}\\ &= \mathbf{B}_P + \mathbf{B}_D,\end{aligned} \tag{111}$$

where

$$\mathbf{B}_P \equiv -E_{\mathbf{x}}\{\mathbf{\Delta}_{\mathbf{x}}^{\mathbf{x}} \ln f(\mathbf{x})\} \tag{112}$$

and

$$\mathbf{B}_D \equiv E_{\mathbf{x}}\left\{-E_{\mathbf{z}\,|\,\mathbf{x}}\{\mathbf{\Delta}_{\mathbf{x}}^{\mathbf{x}} \ln f(\mathbf{z}\,|\,\mathbf{x})\}\right\}. \tag{113}$$

The term $\mathbf{B}_P$ represents prior information and the term $\mathbf{B}_D$ represents information in the data.

The inner term in (113) is the standard FIM, defined as [27]:

$$\mathbf{J}(\mathbf{x}) \equiv -E_{\mathbf{z}\,|\,\mathbf{x}}\{\mathbf{\Delta}_{\mathbf{x}}^{\mathbf{x}} \ln f(\mathbf{z}\,|\,\mathbf{x})\}. \tag{114}$$

Thus the data term can be written as:

$$\mathbf{B}_D = E_{\mathbf{x}}\{\mathbf{J}(\mathbf{x})\}. \tag{115}$$

Appendix B
Sensor FIM Derivation

The sensor FIM is defined as [27]:

$$\mathbf{J}_n(\mathbf{x};\theta_n) = -E_{z_n\,|\,\mathbf{x};\theta_n}\left\{\mathbf{\Delta}_{\mathbf{x}}^{\mathbf{x}} \ln f_n(z_n\,|\,\mathbf{x};\theta_n)\right\}, \tag{116}$$

where $f_n(z_n\,|\,\mathbf{x};\theta_n)$ is the sensor likelihood function on H_1, defined in (69)–(70), and we have dropped the k subscript for simplicity.

When $\theta_n > 0$, the likelihood function is a truncated Gaussian density defined on the interval $[\phi_{n,0}, \phi_{n,0} + \pi)$. We assume that the variance of the density is sufficiently small that the main body of the Gaussian density is well contained within the interval. Then the mean is given to good approximation by $E_{z_n\,|\,\mathbf{x}}\{z_n\} \approx \phi_n(\mathbf{x})$ and the normalization constant is approximately one.

Taking the derivatives, we have

$$\nabla_{\mathbf{x}} \ln f_n(z_n\,|\,\mathbf{x};\theta_n) = \frac{(z_n - \phi_n(\mathbf{x}))}{\sigma_n^2/\theta_n}\nabla_{\mathbf{x}}\phi_n(\mathbf{x}) \tag{117}$$

$$\mathbf{\Delta}_{\mathbf{x}}^{\mathbf{x}} \ln f_n(z_n\,|\,\mathbf{x};\theta_n) = -\frac{1}{\sigma_n^2/\theta_n}\nabla_{\mathbf{x}}\phi_n(\mathbf{x})\left[\nabla_{\mathbf{x}}\phi_n(\mathbf{x})\right]^T + \frac{(z_n - \phi_n(\mathbf{x}))}{\sigma_n^2/\theta_n}\mathbf{\Delta}_{\mathbf{x}}^{\mathbf{x}}\phi_n(\mathbf{x}). \tag{118}$$

Substituting (118) into (116) and using the property that the expected value of the second term in (118) is approximately zero, we obtain

$$\mathbf{J}_n(\mathbf{x};\theta_n) = \frac{\theta_n}{\sigma_n^2}\nabla_{\mathbf{x}}\phi_n(\mathbf{x})\left[\nabla_{\mathbf{x}}\phi_n(\mathbf{x})\right]^T. \tag{119}$$

Evaluating the gradient of $\phi_n(\mathbf{x})$, we have

$$\nabla_{\mathbf{x}}\phi_n(\mathbf{x}) = \frac{1}{r_n(\mathbf{x})^2}\begin{bmatrix}-(y - y_n)\\ (x - x_n)\end{bmatrix} = \frac{1}{r_n(\mathbf{x})}\mathbf{u}_n(\mathbf{x}), \tag{120}$$

where $\mathbf{u}_n(\mathbf{x})$ is the unit vector perpendicular to the vector pointing from the sensor to the target,

$$\mathbf{u}_n(\mathbf{x}) \equiv \frac{1}{r_n(\mathbf{x})}\begin{bmatrix}-(y - y_n)\\ (x - x_n)\end{bmatrix}. \tag{121}$$

Substituting (120) into (119), we obtain

$$\mathbf{J}_n(\mathbf{x};\theta_n) = \frac{\theta_n}{\sigma_n^2 r_n(\mathbf{x})^2}\mathbf{u}_n(\mathbf{x})\mathbf{u}_n(\mathbf{x})^T. \tag{122}$$

When $\theta_n = 0$, the likelihood function in (71) does not depend on $\mathbf{x}$ and the FIM is equal to zero, thus the expression in (122) is valid for $0 \le \theta_n \le 1$.

References

[1] S. Haykin, "Cognitive radar: A way of the future," *IEEE Signal Process. Mag.*, vol. 23, no. 1, pp. 30–40, Jan. 2006.

[2] S. Haykin, Y. Xue, and M. P. Setoodeh, "Cognitive radar: Step toward bridging the gap between neuroscience and engineering," *Proc. IEEE*, vol. 100, no. 11, pp. 3102–3130, Nov. 2012.

[3] S. Haykin, *Cognitive Dynamic Systems (Perception-Action Cycle, Radar, and Radio)*. Cambridge, U.K.: Cambridge Univ. Press, 2012.

[4] S. Haykin, Ed., "Cognitive dynamic systems," *Proc. IEEE*, vol. 102, no. 4, pp. 1369–1372, Apr. 2014.

[5] *Knowledge Based Radar Detection, Tracking, and Classification*, F. Gini and M. Rangaswamy, Eds. Hoboken, NJ, USA: Wiley, 2008.

[6] J. R. Guerci, *Cognitive Radar: The Knowledge Aided Fully Adaptive Approach*. Reading, MA, USA: Artech House, 2010.

[7] S. P. Sira, Y. Li, A. Papandreou-Suppappola, D. Morrell, D. Cochran, and M. Rangaswamy, "Waveform-agile sensing for tracking," *IEEE Signal Process. Mag.*, vol. 26, no. 1, pp. 53–64, Jan. 2009.

[8] D. J. Kershaw and R. J. Evans, "Optimal waveform selection for tracking systems," *IEEE Trans. Inf. Theory*, vol. 40, no. 5, pp. 1536–1550, Sep. 1994.

[9] D. Fuhrmann, "Active-testing surveillance systems, or, playing twenty questions with radar," in *Proc. 11th Annu. Adaptive Sens. Array Process. (ASAP) Workshop*, Lexington, MA, USA, Mar. 2003, MIT Lincoln Lab.

[10] S. P. Sira, A. Papandreou-Suppappola, and D. Morrell, "Dynamic configuration of time-varying waveforms for agile sensing and tracking in clutter," *IEEE Trans. Signal Process.*, vol. 55, no. 7, pp. 3207–3217, Jul. 2007.

[11] M. Hurtado, T. Zhao, and A. Nehorai, "Adaptive polarized waveform design for target tracking based on sequential Bayesian inference," *IEEE Trans. Signal Process.*, vol. 56, no. 13, pp. 1120–1133, Mar. 2008.

[12] M. L. 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.

[13] R. Tharmarasa, T. Kirubarajan, and M. L. Hernandez, "Large-scale optimal sensor array management for multitarget tracking," *IEEE Trans. Syst., Man, Cybern. C: Appl. Rev.*, vol. 37, no. 5, pp. 803–814, Sep. 2007.

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

[15] A. O. Hero and C. M. Kreucher, "Network sensor management for tracking and localization," in *Proc. IEEE Conf. Inf. Fusion*, Quebec, QC, Canada, Jul. 2007.

[16] A. Saksena and I.-J. Wang, "Dynamic ping optimization for surveillance in multistatic sonar buoy networks with energy constraints," in *Proc. 47th IEEE Conf. Decision Control*, Cancun, Mexico, Dec. 2008, pp. 1109–1114.

[17] P. Chavali and A. Nehorai, "Scheduling and power allocation in a cognitive radar network for multiple-target tracking," *IEEE Trans. Signal Process.*, vol. 60, no. 2, pp. 715–729, Feb. 2012.

[18] R. A. Romero and N. A. Goodman, "Cognitive radar network: Cooperative adaptive beamsteering for integrated search-and-track application," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 49, no. 2, pp. 915–931, Apr. 2013.

[19] C. Kreucher and K. Carter, "An information theoretic approach to processing management," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Las Vegas, NV, USA, May 2008, pp. 1869–1872.

[20] U. Güntürkün, "Toward the development of a radar scene analyzer for cognitive radar," *IEEE J. Ocean. Eng.*, vol. 35, no. 2, pp. 303–313, Feb. 2010.

[21] L. D. Stone, R. L. Streit, T. L. Corwin, and K. L. Bell, *Bayesian Multiple Target Tracking*, 2nd ed. Norwood, MA, USA: Artech House, 2014.

[22] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Boston, MA, USA: Artech House, 2004.

[23] K. L. Bell, C. J. Baker, G. E. Smith, J. T. Johnson, and M. Rangaswamy, "Fully adaptive radar for target tracking Part I: Single target tracking," in *Proc. IEEE Radar Conf.*, Cincinnati, OH, USA, May 2014, pp. 303–308.

[24] K. L. Bell, C. J. Baker, G. E. Smith, J. T. Johnson, and M. Rangaswamy, "Fully adaptive radar for target tracking Part II: Target detection and track initiation," in *Proc. IEEE Radar Conf.*, Cincinnati, OH, USA, May 2014, pp. 309–314.
[25] K. L. Bell, J. T. Johnson, G. E. Smith, C. J. Baker, and M. Rangaswamy, "Cognitive radar for target tracking using a software defined radar system," in *Proc. IEEE Radar Conf.*, Washington, DC, USA, May 2015, pp. 1394–1399.
[26] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation With Applications to Tracking and Navigation*. New York, NY, USA: Wiley, 2001.
[27] H. L. Van Trees, K. L. Bell, and Z. Tian, *Detection, Estimation, and Modulation Theory, Part I*, 2nd ed. Hoboken, NJ, USA: Wiley, 2013.
[28] , H. L. Van Trees and K. L. Bell, Eds., *Bayesian Bounds for Parameter Estimation and Nonlinear Filtering/Tracking*. Piscataway, NJ, USA: Wiley-IEEE Press, 2007.
[29] P. Tichavský, C. H. Muravchik, and A. Nehorai, "Posterior Cramér-Rao bounds for discrete-time nonlinear filtering," *IEEE Trans. Signal Process.*, vol. 46, no. 5, pp. 1386–1396, May 1998.
[30] L. Zuo, R. Niu, and P. K. Varshney, "Conditional posterior Cramér-Rao lower bounds for nonlinear sequential Bayesian estimation," *IEEE Trans. Signal Process.*, vol. 59, no. 1, pp. 1–14, Jan. 2011.
[31] B. Z. Bobrovsky, E. Mayer-Wolf, and M. Zakai, "Some classes of global Cramér-Rao bounds," *Ann. Statist.*, vol. 15, no. 4, pp. 1421–1438, 1987.
[32] H. L. Van Trees, *Optimum Array Processing*. New York, NY, USA: Wiley, 2002.

Kristine L. Bell (M'88–S'91–M'96–SM'01–F'15) is a Senior Scientist at Metron, Inc. She received the B.S. in electrical engineering from Rice University in 1985, and the M.S. in electrical engineering and Ph.D. in information technology from George Mason University (GMU) in 1990 and 1995. Her technical expertise is in the area of statistical signal processing and multi-target tracking with applications in radar, sonar, aeroacoustics, and satellite communications. From 1996–2009, Dr. Bell was an Associate/Assistant Professor in the Statistics Department and C4I Center at GMU. Dr. Bell is an IEEE Fellow, and was an Associate Editor of the IEEE TRANSACTIONS ON SIGNAL PROCESSING and chair of the IEEE Signal Processing Society's Sensor Array and Multichannel (SAM) Technical Committee. In 2009, she received the George Mason University Volgenau School of Engineering Outstanding Alumnus Award.

Christopher J. Baker (SM'06) is the Ohio State Research Scholar in Integrated Sensor Systems at The Ohio State University. Until June 2011 he was the Dean and Director of the College of Engineering and Computer Science at the Australian National University (ANU). Prior to this he held the Thales-Royal Academy of Engineering Chair of intelligent radar systems based at University College London. He has been actively engaged in radar systems research since 1984 and is the author of over two hundred and fifty publications. His research interests include coherent radar techniques, radar signal processing, radar signal interpretation, electronically scanned radar systems, radar imaging, natural and cognitive echo locating systems. He is the recipient of the IEE Mountbatten premium (twice), the IEE Institute premium and is a Fellow of the IET. He is a visiting Professor at the University of Cape Town, Cranfield University, University College London, Wright State University, Nanyang Technical University and Strathclyde University.

Graeme E. Smith (M'10–SM'11) is a Research Scientist at The Ohio State University and is a Visiting Scholar at University College London. His pertinent research interests include: cognitive/fully adaptive radar processing; flow fields for echolocation; multistatic radar clutter, radar target recognition/classification; micro-Doppler signatures; and passive radar systems. Before joining the team at The Ohio State University, Dr. Smith worked at Villanova University where his research focused on through-the-wall radar imaging. Prior to this he completed his Ph.D. and first post-doctoral position at University College London. Between 1999 and 2004 he worked as a Lead Systems Engineer for BAE SYSTEMS developing radar warning receivers. Dr. Smith is the author of three book chapters and over sixty journal and conference papers. He is a member of the IET and a Senior Member of the IEEE.

Joel T. Johnson (S'88–M'96–SM'03–F'08) received the bachelor of electrical engineering degree from the Georgia Institute of Technology in 1991 and the S.M. and Ph.D. degrees from the Massachusetts Institute of Technology in 1993 and 1996, respectively. He is currently Professor and Department Chair in the Department of Electrical and Computer Engineering and ElectroScience Laboratory of The Ohio State University. His research interests are in the areas of microwave remote sensing, propagation, and electromagnetic wave theory. Dr. Johnson is an IEEE Fellow, a member of commissions B and F of the International Union of Radio Science (URSI), and a member of Tau Beta Pi, Eta Kappa Nu, and Phi Kappa Phi. He received the 1993 best paper award from the IEEE Geoscience and Remote Sensing Society, was named an Office of Naval Research Young Investigator, National Science Foundation Career awardee, and PECASE award recipient in 1997, and was recognized by the U. S. National Committee of URSI as a Booker Fellow in 2002.

Muralidhar Rangaswamy (S'89–M'93–SM'98–F'06) received the B.E. degree in electronics engineering from Bangalore University, Bangalore, India, in 1985 and the M.S. and Ph.D. degrees in electrical engineering from Syracuse University, Syracuse, NY, in 1992. He is presently employed as the Senior Advisor for Radar Research at the RF Exploitation Branch within the Sensors Directorate of the Air Force Research Laboratory (AFRL). Prior to this he has held industrial and academic appointments. His research interests include radar signal processing, spectrum estimation, modeling non-Gaussian interference phenomena, and statistical communication theory. He has co-authored more than 200 refereed journal and conference record papers in the areas of his research interests. Additionally, he is a contributor to 8 books and is a co-inventor on 3 U.S. patents. He received the IEEE Warren White Radar Award in 2013, the 2013 Affiliate Societies Council Dayton (ASC-D) Outstanding Scientist and Engineer Award, the 2007 IEEE Region 1 Award, the 2006 IEEE Boston Section Distinguished Member Award, and the 2005 IEEE-AESS Fred Nathanson memorial outstanding young radar engineer award. He was elected as a Fellow of the IEEE in January 2006 with the citation for contributions to mathematical techniques for radar space-time adaptive processing. He received the 2012 and 2005 Charles Ryan basic research award from the Sensors Directorate of AFRL, in addition to more than 40 scientific achievement awards.