



# CHALMERS

## Chalmers Publication Library

### **Recent results on Bayesian Cramér-Rao bounds for jump Markov systems**

This document has been downloaded from Chalmers Publication Library (CPL). It is the author's version of a work that was accepted for publication in:

**FUSION 2016 - 19th International Conference on Information Fusion, Proceedings**

Citation for the published paper:

Fritsche, C. ; Orguner, U. ; Svensson, L. et al. (2016) "Recent results on Bayesian Cramér-Rao bounds for jump Markov systems". FUSION 2016 - 19th International Conference on Information Fusion, Proceedings pp. 512-520.

Downloaded from: <http://publications.lib.chalmers.se/publication/245135>

Notice: Changes introduced as a result of publishing processes such as copy-editing and formatting may not be reflected in this document. For a definitive version of this work, please refer to the published source. Please note that access to the published version might require a subscription.

Chalmers Publication Library (CPL) offers the possibility of retrieving research publications produced at Chalmers University of Technology. It covers all types of publications: articles, dissertations, licentiate theses, masters theses, conference papers, reports etc. Since 2006 it is the official tool for Chalmers official publication statistics. To ensure that Chalmers research results are disseminated as widely as possible, an Open Access Policy has been adopted. The CPL service is administrated and maintained by Chalmers Library.

(article starts on next page)

# Recent Results on Bayesian Cramér-Rao Bounds for Jump Markov Systems

Carsten Fritsche, Umut Orguner<sup>‡</sup>, Lennart Svensson\*, and Fredrik Gustafsson

Department of Electrical Engineering, Linköping University, Linköping, Sweden

<sup>‡</sup>Department of Electrical and Electronics Engineering, Middle East Technical University, Ankara, Turkey

\*Department of Signals and Systems, Chalmers University of Technology, Gothenburg, Sweden

**Abstract**—In this paper, recent results on the evaluation of the Bayesian Cramér-Rao bound for jump Markov systems are presented. In particular, previous work is extended to jump Markov systems where the discrete mode variable enters into both the process and measurement equation, as well as where it enters exclusively into the measurement equation. Recursive approximations are derived with finite memory requirements as well as algorithms for checking the validity of these approximations are established. The tightness of the bound and the validity of its approximation is investigated on a couple of examples.

## I. INTRODUCTION

Jump Markov systems (JMSs) are nowadays widely used to model systems in various disciplines, such as target tracking [1], [2], econometrics [3] and control [4], [5] to name only a few. Compared to the nonlinear filtering framework, estimators for JMSs have to additionally estimate the discrete state (or mode) of a Markov chain that allow a switching between different state-space models, for which various estimation algorithms have been proposed e.g. [2], [6]–[9].

The computation of performance bounds for JMSs has also evolved over the past few years. To date, various Bayesian Cramér-Rao bounds (BCRBs) for JMSs have been proposed that generally differ from each other in terms of tightness and computational complexity. The perhaps least computationally complex bound for JMSs is the enumeration BCRB (EBCRB). It is derived from a bound on the mean square error (MSE) conditioned on the entire mode sequence, and an unconditional bound is generally obtained by averaging over all possible mode sequences [1], [10]. A bound that is at least as tight as the EBCRB, but which is significantly more computational complex to evaluate since it relies on running particle filters, is the marginal EBCRB (M-EBCRB). The M-EBCRB is derived from the same principles as the EBCRB, but evaluates a different information matrix, see [11] for the details. A third bound that is directly bounding the unconditional MSE has been presented in [12], and is hereinafter termed BCRB. This bound cannot be related in terms of tightness to the EBCRB or M-EBCRB via an inequality, as explained in [12], [13]. However, its computation is in many cases (e.g. nonlinear models or time varying models) only slightly more complex than the computation of the EBCRB. Another bound that has been proposed in the literature, is the so-called marginal BCRB (M-BCRB) [14]. It is also directly bounding the unconditional MSE but similar to the M-EBCRB evaluates a different

information matrix. It has been shown that the M-BCRB is at least as tight as the BCRB, but it is much more complex to evaluate.

In this paper, we focus on the evaluation of the BCRB proposed in [12]. In particular, this bound is useful in situations when it is tighter than the EBCRB and when computational resources are not available for evaluating the M-BCRB or M-EBCRB. The main contributions are as follows. We generalize the approach presented in [12] to the important cases, where the discrete mode variable enters either exclusively into the measurement model or into both the process and measurement models, see e.g. [13], [15]–[17] and [18], [19] for application examples. For both cases we additionally derive recursions for approximately computing the BCRB. These recursions depend on a conditional independence assumption between temporal random variables within a certain time interval that has been chosen in [12] empirically. We present recursive algorithms with linear in time complexity that can be used to specify this time interval.

The rest of the paper is organized as follows. In Section II, the system model is presented together with some definitions used in the paper. Section III gives a brief background overview on the BCRB, and Section IV provides the main results for computing the BCRB. The algorithms for conditional independence assumption verification are presented in Section V, and the simulation results are summarized in Section VI. Section VII finally concludes this work.

## II. SYSTEM MODEL

Consider the discrete-time JMS, that is described by the following process and measurement equation

$$\mathbf{x}_k = \mathbf{f}_{k-1}(\mathbf{x}_{k-1}, r_k, \mathbf{v}_{k-1}), \quad (1a)$$

$$\mathbf{z}_k = \mathbf{h}_k(\mathbf{x}_k, r_k, \mathbf{w}_k), \quad (1b)$$

where  $\mathbf{z}_k \in \mathbb{R}^{n_z}$  is the measurement vector at discrete time instant  $k$  and  $\mathbf{x}_k \in \mathbb{R}^{n_x}$  is the state vector and  $\mathbf{f}_{k-1}$  and  $\mathbf{h}_k$  are arbitrary nonlinear functions. The process and measurement noise vectors  $\mathbf{v}_{k-1} \in \mathbb{R}^{n_v}$  and  $\mathbf{w}_k \in \mathbb{R}^{n_w}$  are assumed mutually independent white processes, with noise densities  $p_{\mathbf{v}, r_k}(\mathbf{v}_{k-1})$  and  $p_{\mathbf{w}, r_k}(\mathbf{w}_k)$  that are assumed known. The mode variable  $r_k$  denotes a discrete-time Markov chain with  $s$  states and transition probability matrix  $\Pi$  with elements  $\pi_{ij} = \Pr\{r_k = j | r_{k-1} = i\}$ . At time instances  $k = 0$  and  $k = 1$ , prior information about the state  $\mathbf{x}_0$  and mode  $r_1$

is available in terms of the probability density function (pdf)  $p_{\mathbf{x}_0}(\mathbf{x}_0)$  and probability mass function (pmf)  $\pi_i^1 = \Pr\{r_1 = i\}$ . The initial state  $\mathbf{x}_0$  and mode  $r_1$  are mutually independent and also independent of  $\mathbf{w}_k$  and  $\mathbf{v}_{k-1}$ .

In the following, let  $\mathbf{x}_{0:k} = [\mathbf{x}_0^\top, \dots, \mathbf{x}_k^\top]^\top$  and  $\mathbf{z}_{1:k} = [\mathbf{z}_1^\top, \dots, \mathbf{z}_k^\top]^\top$  denote the collection of states and measurement vectors up to time  $k$ . Furthermore, let  $\hat{\mathbf{x}}_{0:k}(\mathbf{z}_{1:k}) = [\hat{\mathbf{x}}_0^\top(\mathbf{z}_{1:k}), \dots, \hat{\mathbf{x}}_k^\top(\mathbf{z}_{1:k})]^\top$  denote the estimator of the state sequence, and let the sequence of mode variables at time  $k$  be given by  $r_{1:k}^i = (r_1^i, r_2^i, \dots, r_k^i)$ , where  $i = 1, \dots, s^k$ . Whenever possible, and when there is no risk of ambiguity, the estimator's dependency on the measurements  $\mathbf{z}_{1:k}$  is omitted in the following. Let us further introduce the gradient operator  $\nabla_{\mathbf{s}} = [\partial/\partial s_1, \dots, \partial/\partial s_n]^\top$  and Laplace operator  $\Delta_{\mathbf{s}}^t = \nabla_{\mathbf{s}} \nabla_{\mathbf{t}}^\top$  for any vectors  $\mathbf{s}$  and  $\mathbf{t}$ , and let  $\mathbb{E}_{p(x)}\{\cdot\}$  denote expectation with respect to the pdf (or pmf)  $p(x)$ .

### III. BACKGROUND ON BCRB

The BCRB provides a lower bound on the MSE matrix  $\mathcal{M}(\hat{\mathbf{x}}_{0:k}(\mathbf{z}_{1:k}))$  of any estimator  $\hat{\mathbf{x}}_{0:k}(\mathbf{z}_{1:k})$ . Assuming that suitable regularity conditions hold [20], the BCRB for estimating the state sequence  $\mathbf{x}_{0:k}$  is defined as the inverse of the Bayesian information matrix (BIM)  $\mathbf{J}_{0:k}$ , bounding

$$\mathcal{M}(\hat{\mathbf{x}}_{0:k}) \triangleq \mathbb{E}_{p(\mathbf{x}_{0:k}, \mathbf{z}_{1:k})} \{ [\hat{\mathbf{x}}_{0:k}(\mathbf{z}_{1:k}) - \mathbf{x}_{0:k}] [\cdot]^\top \} \geq [\mathbf{J}_{0:k}]^{-1}, \quad (2)$$

where  $[\mathbf{A}][\cdot]^\top$  stands for  $[\mathbf{A}][\mathbf{A}]^\top$  and where the matrix inequality  $\mathbf{A} \geq \mathbf{B}$  means that the difference  $\mathbf{A} - \mathbf{B}$  is a positive semi-definite matrix [21]. The BIM is defined as

$$\mathbf{J}_{0:k} = \mathbb{E}_{p(\mathbf{x}_{0:k}, \mathbf{z}_{1:k})} \{ -\Delta_{\mathbf{x}_{0:k}}^{\mathbf{x}_{0:k}} \log p(\mathbf{x}_{0:k}, \mathbf{z}_{1:k}) \}, \quad (3)$$

with dimension given by  $(k+1)n_x \times (k+1)n_x$ .

In the following, we are interested in computing the BCRB of the MSE matrix for estimating  $\mathbf{x}_k$ . Generally, this can be achieved by taking the  $(n_x \times n_x)$  lower-right submatrix of  $[\mathbf{J}_{0:k}]^{-1}$ , which can be expressed mathematically for  $k \geq 1$  as

$$\begin{aligned} \mathcal{M}(\hat{\mathbf{x}}_k) &= \mathbb{E}_{p(\mathbf{x}_k, \mathbf{z}_{1:k})} \{ [\hat{\mathbf{x}}_k(\mathbf{z}_{1:k}) - \mathbf{x}_k] [\cdot]^\top \} \\ &= \mathbf{U} \mathcal{M}(\hat{\mathbf{x}}_{0:k}) \mathbf{U}^\top \\ &\geq \mathbf{U} [\mathbf{J}_{0:k}]^{-1} \mathbf{U}^\top \triangleq [\mathbf{J}_k]^{-1}, \end{aligned} \quad (4)$$

with mapping matrix

$$\mathbf{U} = [\mathbf{0}, \mathbf{I}_{n_x}], \quad (5)$$

and where  $\mathbf{I}_{n_x}$  is the  $(n_x \times n_x)$  identity matrix and  $\mathbf{0}$  is a matrix of zeros of appropriate size. The matrix  $\mathbf{J}_k$  is denoted as the filtering information matrix, whose inverse gives the BCRB for estimating  $\mathbf{x}_k$  we seek to derive.

### IV. COMPUTING THE BCRB

#### A. Jump Markov System Models

Depending on how the mode variable  $r_k$  enters into the system equations, different JMSs will result. In total, three different system models can be identified which are summarized in Table I. When both the process and measurement model are independent of  $r_k$ , we arrive at systems without Markovian

switching structure and the BCRB for this case was presented in [22]. In this paper, only algorithms for computing the BCRB for Model 1 and 2 are presented. The reader interested in the BCRB for Model 3 is referred to [12].

The approach followed in this paper for computing the BCRB is to numerically evaluate the BIM  $\mathbf{J}_{0:k}$  of the complete state trajectory  $\mathbf{x}_{0:k}$  using Monte Carlo methods. In many cases, the expression inside the expectation of (3) is difficult to evaluate directly and it may then be easier to evaluate the equivalent expression

$$\mathbf{J}_{0:k} = \mathbb{E}_{p(\mathbf{x}_{0:k}, \mathbf{z}_{1:k})} \left\{ \frac{[\nabla_{\mathbf{x}_{0:k}} p(\mathbf{x}_{0:k}, \mathbf{z}_{1:k})][\cdot]^\top}{p(\mathbf{x}_{0:k}, \mathbf{z}_{1:k})^2} \right\}. \quad (6)$$

If the mode variable enters only into one of the system equations, structure inherent in the BIM can be exploited. In these cases, it is convenient to decompose the BIM as follows

$$\mathbf{J}_{0:k} = \mathbf{J}_{\mathbf{x}_{0:k}} + \mathbf{J}_{\mathbf{z}_{1:k}}, \quad (7)$$

where  $\mathbf{J}_{\mathbf{x}_{0:k}}$  denotes the BIM of the prior and process model:

$$\mathbf{J}_{\mathbf{x}_{0:k}} = \mathbb{E}_{p(\mathbf{x}_{0:k})} \left\{ \frac{[\nabla_{\mathbf{x}_{0:k}} p(\mathbf{x}_{0:k})][\cdot]^\top}{p(\mathbf{x}_{0:k})^2} \right\} \quad (8)$$

and  $\mathbf{J}_{\mathbf{z}_{1:k}}$  denotes the BIM of the data:

$$\mathbf{J}_{\mathbf{z}_{1:k}} = \mathbb{E}_{p(\mathbf{x}_{0:k}, \mathbf{z}_{1:k})} \left\{ \frac{[\nabla_{\mathbf{x}_{0:k}} p(\mathbf{z}_{1:k} | \mathbf{x}_{0:k})][\cdot]^\top}{p(\mathbf{z}_{1:k} | \mathbf{x}_{0:k})^2} \right\}. \quad (9)$$

In particular, different algorithms will be provided on how  $\mathbf{J}_{0:k}$  or  $\mathbf{J}_{\mathbf{x}_{0:k}}$  and  $\mathbf{J}_{\mathbf{z}_{1:k}}$  can be evaluated for the different models presented in Table I. The results can be then used to compute the BCRB for the current state  $\mathbf{x}_k$  according to (4). The general algorithmic structure for computing the BIM  $\mathbf{J}_{0:k}$  for the different models is presented in Algorithm 1.

---

**Algorithm 1** Computation of the BIM  $\mathbf{J}_{0:k}$  for different JMS models

---

- (1) At time  $k = 0$ , generate  $\mathbf{x}_0^{(i)} \sim p(\mathbf{x}_0)$  for  $i = 1, \dots, N$ , and define  $\mathbf{x}_{0:0}^{(i)} = \mathbf{x}_0^{(i)}$ .
    - For each  $i$ , evaluate  $\nabla_{\mathbf{x}_{0:0}} p(\mathbf{x}_{0:0}^{(i)}) = \nabla_{\mathbf{x}_0} p(\mathbf{x}_0^{(i)})$  and  $p(\mathbf{x}_{0:0}^{(i)}) = p(\mathbf{x}_0^{(i)})$ .
    - Compute the initial BIM  $\mathbf{J}_0 = \mathbf{J}_{\mathbf{x}_0}$  and store the results of  $\mathbf{J}_{\mathbf{x}_0}$ .
  - (2) For  $k = 1, 2, \dots$ , do:
    - If  $k = 1$ , generate  $r_1^{(i)} \sim \Pr\{r_1\}$ , otherwise generate  $r_k^{(i)} \sim \Pr\{r_k | r_{k-1}^{(i)}\}$  for  $i = 1, \dots, N$
    - Compute the BIM  $\mathbf{J}_{0:k}$ :
      - \* Model 1: from (6) using Algorithm 2
      - \* Model 2: from (7) by determining:
        - $\mathbf{J}_{\mathbf{x}_{0:k}}$  using Algorithm 3
        - $\mathbf{J}_{\mathbf{z}_{1:k}}$  using Algorithm 4
      - \* Model 3: from (7) by determining:
        - $\mathbf{J}_{\mathbf{x}_{0:k}}$ , see (23)-(29) in [12]
        - $\mathbf{J}_{\mathbf{z}_{1:k}}$ , see (9) in [12]
-

TABLE I  
JUMP MARKOV SYSTEM MODELS

	$\mathbf{x}_k = \mathbf{f}_{k-1}(\mathbf{x}_{k-1}, r_k, \mathbf{v}_{k-1})$	$\mathbf{x}_k = \mathbf{f}_{k-1}(\mathbf{x}_{k-1}, \mathbf{v}_{k-1})$
$\mathbf{z}_k = \mathbf{h}_k(\mathbf{x}_k, r_k, \mathbf{w}_k)$	Model 1	Model 2
$\mathbf{z}_k = \mathbf{h}_k(\mathbf{x}_k, \mathbf{w}_k)$	Model 3, see [12]	[22]

### B. BIM Computation for Model 1

For nonlinear JMSs as given by (1) closed-form solutions for computing  $\mathbf{J}_{0:k}$  generally do not exist. In the following, Monte Carlo integration is used to approximate (6)

$$\mathbf{J}_{0:k} \approx \frac{1}{N} \sum_{i=1}^N \frac{[\nabla_{\mathbf{x}_{0:k}} p(\mathbf{x}_{0:k}^{(i)}, \mathbf{z}_{1:k}^{(i)})][\cdot]^\top}{[p(\mathbf{x}_{0:k}^{(i)}, \mathbf{z}_{1:k}^{(i)})]^2}, \quad (10)$$

where  $\mathbf{x}_{0:k}^{(i)}$  and  $\mathbf{z}_{1:k}^{(i)}$ ,  $i = 1, \dots, N$ , are independent and identically distributed (i.i.d.) vectors such that  $(\mathbf{x}_{0:k}^{(i)}, \mathbf{z}_{1:k}^{(i)}) \sim p(\mathbf{x}_{0:k}, \mathbf{z}_{1:k})$ . We introduce the intermediate quantities  $p(\mathbf{x}_{0:k}, \mathbf{z}_{1:k}|r_k)$  and  $\nabla_{\mathbf{x}_{0:k}} p(\mathbf{x}_{0:k}, \mathbf{z}_{1:k}|r_k)$  which can be computed recursively as shown in the following lemma.

**Lemma 1.** For JMSs as given by Model 1 in Table I, the pdf  $p(\mathbf{x}_{0:k}, \mathbf{z}_{1:k}|r_k)$  and the gradient  $\nabla_{\mathbf{x}_{0:k}} p(\mathbf{x}_{0:k}, \mathbf{z}_{1:k}|r_k)$  can be updated recursively as follows:

If  $k = 1$ :

$$p(\mathbf{x}_{0:1}, \mathbf{z}_1|r_1) = p(\mathbf{z}_1|\mathbf{x}_1, r_1)p(\mathbf{x}_1|\mathbf{x}_0, r_1)p(\mathbf{x}_0), \quad (11a)$$

$$\begin{aligned} \nabla_{\mathbf{x}_{0:1}} p(\mathbf{x}_{0:1}, \mathbf{z}_1|r_1) &= \left[ \nabla_{\mathbf{x}_{0:1}} p(\mathbf{z}_1|\mathbf{x}_1, r_1) \right] p(\mathbf{x}_1|\mathbf{x}_0, r_1) \\ &\quad + p(\mathbf{z}_1|\mathbf{x}_1, r_1) \left[ \nabla_{\mathbf{x}_{0:1}} p(\mathbf{x}_1|\mathbf{x}_0, r_1) \right] p(\mathbf{x}_0) \\ &\quad + p(\mathbf{z}_1|\mathbf{x}_1, r_1) p(\mathbf{x}_1|\mathbf{x}_0, r_1) \left[ \nabla_{\mathbf{x}_{0:1}} p(\mathbf{x}_0) \right]. \end{aligned} \quad (11b)$$

If  $k \neq 1$ :

$$\begin{aligned} p(\mathbf{x}_{0:k}, \mathbf{z}_{1:k}|r_k) &= p(\mathbf{z}_k|\mathbf{x}_k, r_k)p(\mathbf{x}_k|\mathbf{x}_{k-1}, r_k) \\ &\quad \times \sum_{r_{k-1}} \Pr\{r_{k-1}|r_k\} p(\mathbf{x}_{0:k-1}, \mathbf{z}_{1:k-1}|r_{k-1}), \end{aligned} \quad (12a)$$

$$\begin{aligned} \nabla_{\mathbf{x}_{0:k}} p(\mathbf{x}_{0:k}, \mathbf{z}_{1:k}|r_k) &= \left[ \nabla_{\mathbf{x}_{0:k}} p(\mathbf{z}_k|\mathbf{x}_k, r_k) \right] p(\mathbf{x}_k|\mathbf{x}_{k-1}, r_k) \\ &\quad + p(\mathbf{z}_k|\mathbf{x}_k, r_k) \left[ \nabla_{\mathbf{x}_{0:k}} p(\mathbf{x}_k|\mathbf{x}_{k-1}, r_k) \right] \sum_{r_{k-1}} \Pr\{r_{k-1}|r_k\} \\ &\quad \times p(\mathbf{x}_{0:k-1}, \mathbf{z}_{1:k-1}|r_{k-1}) + p(\mathbf{z}_k|\mathbf{x}_k, r_k) p(\mathbf{x}_k|\mathbf{x}_{k-1}, r_k) \\ &\quad \times \sum_{r_{k-1}} \Pr\{r_{k-1}|r_k\} \left[ \nabla_{\mathbf{x}_{0:k}} p(\mathbf{x}_{0:k-1}, \mathbf{z}_{1:k-1}|r_{k-1}) \right], \end{aligned} \quad (12b)$$

where  $\Pr\{r_{k-1}|r_k\} = \Pr\{r_k|r_{k-1}\} \Pr\{r_{k-1}\} / \Pr\{r_k\}$ .

*Proof:* Due to space limitations, the proof is provided in an accompanying technical report [23]. ■

Consequently, the BIM  $\mathbf{J}_{0:k}$  can be computed recursively which is summarized in Algorithm 2.

### C. BIM Computation for Model 2

For JMSs where  $r_k$  enters exclusively into the measurement equation, structure in the BIM can be exploited by making

### Algorithm 2 Computation of BIM $\mathbf{J}_{0:k}$ for Model 1 and $k \geq 1$

(1) For  $i = 1, \dots, N$  do:

- Generate  $\mathbf{x}_k^{(i)} \sim p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(i)}, r_k^{(i)})$  and set  $\mathbf{x}_{0:k}^{(i)} = [\mathbf{x}_{0:k-1}^{(i)}, \mathbf{x}_k^{(i)}]$ . Generate  $\mathbf{z}_k^{(i)} \sim p(\mathbf{z}_k|\mathbf{x}_k^{(i)}, r_k^{(i)})$  and set  $\mathbf{z}_{1:k}^{(i)} = [\mathbf{z}_{1:k-1}^{(i)}, \mathbf{z}_k^{(i)}]$ .
- If  $k = 1$ , then evaluate the quantities  $p(\mathbf{x}_{0:1}, \mathbf{z}_1|r_1)$  and  $\nabla_{\mathbf{x}_{0:1}} p(\mathbf{x}_{0:1}, \mathbf{z}_1|r_1)$  using (11).
- If  $k \neq 1$ , then update the stored quantities  $\Pr\{r_{k-1}\}$ ,  $\nabla_{\mathbf{x}_{0:k-1}} p(\mathbf{x}_{0:k-1}, \mathbf{z}_{1:k-1}|r_{k-1})$  and  $p(\mathbf{x}_{0:k-1}, \mathbf{z}_{1:k-1}|r_{k-1})$ , using (12) and

$$\Pr\{r_k\} = \sum_{r_{k-1}} \Pr\{r_k|r_{k-1}\} \Pr\{r_{k-1}\}.$$

- Evaluate  $p(\mathbf{x}_{0:k}, \mathbf{z}_{1:k}^{(i)})$  and  $\nabla_{\mathbf{x}_{0:k}} p(\mathbf{x}_{0:k}, \mathbf{z}_{1:k}^{(i)})$  as follows:

$$p(\mathbf{x}_{0:k}, \mathbf{z}_{1:k}^{(i)}) = \sum_{r_k} p(\mathbf{x}_{0:k}, \mathbf{z}_{1:k}^{(i)}|r_k) \Pr\{r_k\}$$

$$\nabla_{\mathbf{x}_{0:k}} p(\mathbf{x}_{0:k}, \mathbf{z}_{1:k}^{(i)}) = \sum_{r_k} \left[ \nabla_{\mathbf{x}_{0:k}} p(\mathbf{x}_{0:k}, \mathbf{z}_{1:k}^{(i)}|r_k) \right] \Pr\{r_k\}$$

(2) Evaluate the BIM  $\mathbf{J}_{0:k}$  according to (10).

use of (7). In fact, it is easy to verify that when the process equation satisfies

$$\mathbf{x}_k = \mathbf{f}_{k-1}(\mathbf{x}_{k-1}, \mathbf{v}_{k-1}), \quad (13)$$

then the BIM of the prior  $\mathbf{J}_{\mathbf{x}_{0:k}}$  will be independent of  $r_k$ . More specifically, for Model 2 the state vector  $\mathbf{x}_k$  is a Markov process, i.e.  $p(\mathbf{x}_k|\mathbf{x}_{0:k-1}) = p(\mathbf{x}_k|\mathbf{x}_{k-1})$  holds, and the BIM of the prior  $\mathbf{J}_{\mathbf{x}_{0:k}}$  can be computed according to the following lemma.

**Lemma 2.** For a mode-independent process equation as given by (13), the BIM of the prior can be computed according to:

$$\begin{aligned} \mathbf{J}_{\mathbf{x}_{0:k}} &= \delta_{k+1}(1, 1) \otimes \mathbf{J}_{\mathbf{x}_0} + \sum_{n=1}^k \left[ \delta_{k+1}(n, n) \otimes \mathbf{D}_n^{11} \right. \\ &\quad + \delta_{k+1}(n, n+1) \otimes \mathbf{D}_n^{12} + \delta_{k+1}(n+1, n) \otimes \mathbf{D}_n^{21} \\ &\quad \left. + \delta_{k+1}(n+1, n+1) \otimes \mathbf{D}_n^{22,a} \right], \end{aligned} \quad (14)$$

with

$$\mathbf{D}_n^{11} = \mathbb{E}_{p(\mathbf{x}_{0:n})} \{ -\Delta_{\mathbf{x}_{n-1}}^{\mathbf{x}_{n-1}} \log p(\mathbf{x}_n|\mathbf{x}_{n-1}) \}, \quad (15a)$$

$$\mathbf{D}_n^{12} = \mathbb{E}_{p(\mathbf{x}_{0:n})} \{ -\Delta_{\mathbf{x}_{n-1}}^{\mathbf{x}_{n-1}} \log p(\mathbf{x}_n|\mathbf{x}_{n-1}) \} = [\mathbf{D}_n^{21}]^\top, \quad (15b)$$

$$\mathbf{D}_n^{22,a} = \mathbb{E}_{p(\mathbf{x}_{0:n})} \{ -\Delta_{\mathbf{x}_n}^{\mathbf{x}_n} \log p(\mathbf{x}_n|\mathbf{x}_{n-1}) \}, \quad (15c)$$

where  $\otimes$  denotes the Kronecker product and  $\delta_k(i, j)$  denotes a  $(k \times k)$  dimensional matrix whose elements are all zero except at the  $i$ -th row and the  $j$ -th column which is one.

*Proof:* See technical report [23]. ■

The expectations in (15) generally cannot be solved analytically for nonlinear models as given by (13). In this case, the expectations can be converted into a different standard form, as in (6), and then may be approximated using Monte Carlo integration, yielding

$$\mathbf{D}_n^{11} \approx \frac{1}{N} \sum_{i=1}^N \frac{[\nabla_{\mathbf{x}_{n-1}} p(\mathbf{x}_n^{(i)} | \mathbf{x}_{n-1}^{(i)})][\cdot]^\top}{[p(\mathbf{x}_n^{(i)} | \mathbf{x}_{n-1}^{(i)})]^2}, \quad (16a)$$

$$\begin{aligned} \mathbf{D}_n^{12} &\approx \frac{1}{N} \sum_{i=1}^N \frac{[\nabla_{\mathbf{x}_n} p(\mathbf{x}_n^{(i)} | \mathbf{x}_{n-1}^{(i)})][\nabla_{\mathbf{x}_{n-1}} p(\mathbf{x}_n^{(i)} | \mathbf{x}_{n-1}^{(i)})]^\top}{[p(\mathbf{x}_n^{(i)} | \mathbf{x}_{n-1}^{(i)})]^2}, \\ &= [\mathbf{D}_n^{21}]^\top \end{aligned} \quad (16b)$$

$$\mathbf{D}_n^{22,a} \approx \frac{1}{N} \sum_{i=1}^N \frac{[\nabla_{\mathbf{x}_n} p(\mathbf{x}_n^{(i)} | \mathbf{x}_{n-1}^{(i)})][\cdot]^\top}{[p(\mathbf{x}_n^{(i)} | \mathbf{x}_{n-1}^{(i)})]^2}, \quad (16c)$$

where  $\mathbf{x}_{0:n}^{(i)}$ ,  $i = 1, \dots, N$  are i.i.d. vectors such that  $\mathbf{x}_{0:n}^{(i)} \sim p(\mathbf{x}_{0:n})$ . A method for numerically approximating  $\mathbf{J}_{\mathbf{x}_{0:k}}$  is given in Algorithm 3.

---

**Algorithm 3** Computation of BIM of the prior  $\mathbf{J}_{\mathbf{x}_{0:k}}$  for Model 2 and  $k \geq 1$

---

- (1) For  $i = 1, \dots, N$  do:
    - Generate  $\mathbf{x}_k^{(i)} \sim p(\mathbf{x}_k | \mathbf{x}_{k-1}^{(i)})$
    - Evaluate  $\nabla_{\mathbf{x}_{k-1}} p(\mathbf{x}_k^{(i)} | \mathbf{x}_{k-1}^{(i)})$ ,  $\nabla_{\mathbf{x}_k} p(\mathbf{x}_k^{(i)} | \mathbf{x}_{k-1}^{(i)})$  and  $p(\mathbf{x}_k^{(i)} | \mathbf{x}_{k-1}^{(i)})$ .
  - (2) Evaluate  $\mathbf{D}_k^{11}$ ,  $\mathbf{D}_k^{12}$ ,  $\mathbf{D}_k^{21}$  and  $\mathbf{D}_k^{22,a}$  according to (16) and store the results.
  - (3) Evaluate the BIM of the prior  $\mathbf{J}_{\mathbf{x}_{0:k}}$  according to (14).
- 

It is easy to verify that the BIM for the prior (14) is the same as in [22]. However, we still cannot develop a recursive algorithm which is equally simple because of the correlation caused by the mode  $r_k$  affecting the BIM of the data  $\mathbf{J}_{\mathbf{z}_{1:k}}$ . For nonlinear models as given by (1b),  $\mathbf{J}_{\mathbf{z}_{1:k}}$  is generally not tractable analytically. In the following, Monte Carlo integration is used to numerically approximate (9) according to:

$$\mathbf{J}_{\mathbf{z}_{1:k}} \approx \frac{1}{N} \sum_{i=1}^N \frac{[\nabla_{\mathbf{x}_{0:k}} p(\mathbf{z}_{1:k}^{(i)} | \mathbf{x}_{0:k}^{(i)})][\cdot]^\top}{[p(\mathbf{z}_{1:k}^{(i)} | \mathbf{x}_{0:k}^{(i)})]^2}, \quad (17)$$

where  $\mathbf{x}_{0:k}^{(i)}$  and  $\mathbf{z}_{1:k}^{(i)}$ ,  $i = 1, \dots, N$ , are i.i.d. vectors such that  $(\mathbf{x}_{0:k}^{(i)}, \mathbf{z}_{1:k}^{(i)}) \sim p(\mathbf{x}_{0:k}, \mathbf{z}_{1:k})$ . We introduce the intermediate quantities  $p(\mathbf{z}_{1:k}, r_k | \mathbf{x}_{0:k})$  and  $\nabla_{\mathbf{x}_{0:k}} p(\mathbf{z}_{1:k}, r_k | \mathbf{x}_{0:k})$ , which can be computed recursively as stated in the following lemma.

**Lemma 3.** For JMSs as given by Model 2 in Table I, the pdf  $p(\mathbf{z}_{1:k}, r_k | \mathbf{x}_{0:k})$  and the gradient  $\nabla_{\mathbf{x}_{0:k}} p(\mathbf{z}_{1:k}, r_k | \mathbf{x}_{0:k})$  can be updated recursively as follows:

If  $k = 1$ :

$$p(\mathbf{z}_1, r_1 | \mathbf{x}_{0:1}) = p(\mathbf{z}_1 | \mathbf{x}_1, r_1) \Pr\{r_1\}, \quad (18a)$$

$$\nabla_{\mathbf{x}_{0:1}} p(\mathbf{z}_1, r_1 | \mathbf{x}_{0:1}) = [\nabla_{\mathbf{x}_{0:1}} p(\mathbf{z}_1 | \mathbf{x}_1, r_1)] \Pr\{r_1\} \quad (18b)$$

If  $k \neq 1$ :

$$\begin{aligned} p(\mathbf{z}_{1:k}, r_k | \mathbf{x}_{0:k}) &= p(\mathbf{z}_k | \mathbf{x}_k, r_k) \\ &\times \sum_{r_{k-1}} \Pr\{r_k | r_{k-1}\} p(\mathbf{z}_{1:k-1}, r_{k-1} | \mathbf{x}_{0:k-1}), \end{aligned} \quad (19a)$$

$$\begin{aligned} \nabla_{\mathbf{x}_{0:k}} p(\mathbf{z}_{1:k}, r_k | \mathbf{x}_{0:k}) &= \sum_{r_{k-1}} \Pr\{r_k | r_{k-1}\} \\ &\times \left[ [\nabla_{\mathbf{x}_{0:k}} p(\mathbf{z}_k | \mathbf{x}_k, r_k)] p(\mathbf{z}_{1:k-1}, r_{k-1} | \mathbf{x}_{0:k-1}) \right. \\ &\left. + p(\mathbf{z}_k | \mathbf{x}_k, r_k) [\nabla_{\mathbf{x}_{0:k}} p(\mathbf{z}_{1:k-1}, r_{k-1} | \mathbf{x}_{0:k-1})] \right]. \end{aligned} \quad (19b)$$

*Proof:* See technical report [23]. ■

Then, the BIM of the data  $\mathbf{J}_{\mathbf{z}_{1:k}}$  can be computed recursively as summarized in Algorithm 4. Note, that a similar but computationally more complex algorithm has appeared in our previous work [24].

---

**Algorithm 4** Computation of BIM of the data  $\mathbf{J}_{\mathbf{z}_{1:k}}$  for Model 2 and  $k \geq 1$

---

- (1) For  $i = 1, \dots, N$  do:
  - Generate  $\mathbf{z}_k^{(i)} \sim p(\mathbf{z}_k | \mathbf{x}_k^{(i)}, r_k^{(i)})$  and set  $\mathbf{z}_{1:k}^{(i)} = [\mathbf{z}_{1:k-1}^{(i)}, \mathbf{z}_k^{(i)}]$ .
  - If  $k = 1$ , then evaluate the quantities  $p(\mathbf{z}_1^{(i)}, r_1 | \mathbf{x}_{0:1}^{(i)})$  and  $\nabla_{\mathbf{x}_{0:1}} p(\mathbf{z}_1^{(i)}, r_1 | \mathbf{x}_{0:1}^{(i)})$  using (18).
  - If  $k \neq 1$ , update the stored quantities  $p(\mathbf{z}_{1:k-1}^{(i)}, r_{k-1} | \mathbf{x}_{0:k-1}^{(i)})$  and  $\nabla_{\mathbf{x}_{0:k-1}} p(\mathbf{z}_{1:k-1}^{(i)}, r_{k-1} | \mathbf{x}_{0:k-1}^{(i)})$  using (19).
  - Evaluate  $p(\mathbf{z}_{1:k}^{(i)} | \mathbf{x}_{0:k}^{(i)})$  and  $\nabla_{\mathbf{x}_{0:k}} p(\mathbf{z}_{1:k}^{(i)} | \mathbf{x}_{0:k}^{(i)})$  as follows:

$$p(\mathbf{z}_{1:k}^{(i)} | \mathbf{x}_{0:k}^{(i)}) = \sum_{r_k} p(\mathbf{z}_{1:k}, r_k | \mathbf{x}_{0:k}^{(i)}),$$

$$\nabla_{\mathbf{x}_{0:k}} p(\mathbf{z}_{1:k}^{(i)} | \mathbf{x}_{0:k}^{(i)}) = \sum_{r_k} \nabla_{\mathbf{x}_{0:k}} p(\mathbf{z}_{1:k}, r_k | \mathbf{x}_{0:k}^{(i)})$$

- (2) Evaluate the BIM of the data  $\mathbf{J}_{\mathbf{z}_{1:k}}$  according to (17).
- 

#### D. Recursive Computation of the BCRB

The algorithm presented so far requires the computation of the matrix inverse  $[\mathbf{J}_{0:k}]^{-1}$ , see (4). This approach eventually becomes impractical in situations when  $k$  is large, due to its computational complexity which is in the order of  $\mathcal{O}(((k+1)n_x)^3)$ . In these situations, recursive algorithms are sought after that avoid inverting  $\mathbf{J}_{0:k}$ .

1) *Model 1:* The recursive algorithm presented in [12] can be generalized to Model 1 as described in the following.

For nonlinear JMSs, a recursive calculation of the filtering information matrix  $\mathbf{J}_k$  is generally not possible without introducing further approximations. This is due to the fact that the state vector  $\mathbf{x}_k$  is not a Markov process<sup>1</sup>, i.e. conditionally it

<sup>1</sup>In order to obtain a Markov process, we have to augment the state vector  $\mathbf{x}_k$  with the discrete mode variable  $r_k$ .

depends on the entire state sequence  $\mathbf{x}_{0:k-1}$ , or equivalently

$$p(\mathbf{x}_k, \mathbf{z}_k | \mathbf{x}_{0:k-1}, \mathbf{z}_{1:k-1}) \neq p(\mathbf{x}_k, \mathbf{z}_k | \mathbf{x}_{k-1}, \mathbf{z}_{1:k-1}). \quad (20)$$

Nevertheless, it can be assumed that given the measurement sequence  $\mathbf{z}_{1:k-1}$ , the dependence between  $(\mathbf{x}_k, \mathbf{z}_k)$  and  $\mathbf{x}_{k-l}$  decreases rather quickly, especially when conditioned on the state vector of all intermediate times,  $\mathbf{x}_{k-l+1:k-1}$ . Thus, it is reasonable to assume that there exists an integer  $d$ , such that

$$p(\mathbf{x}_k, \mathbf{z}_k | \mathbf{x}_{0:k-1}, \mathbf{z}_{1:k-1}) \approx p(\mathbf{x}_k, \mathbf{z}_k | \mathbf{x}_{k-d:k-1}, \mathbf{z}_{1:k-1}), \quad (21)$$

i.e.  $(\mathbf{x}_k, \mathbf{z}_k)$  and  $\mathbf{x}_{k-l}$  given  $\mathbf{z}_{1:k-1}$  are approximately independent for all  $l > d$ , when we condition on the state vectors  $\mathbf{x}_{k-d:k-1}$ . The above assumption will result in two important properties for  $\mathbf{J}_{0:k}$  as stated in the following lemma.

**Lemma 4.** *Suppose that given the measurements  $\mathbf{z}_{1:k-1}$ , the joint vector  $(\mathbf{x}_k, \mathbf{z}_k)$  and  $\mathbf{x}_{0:k-d-1}$  are conditionally independent in the sense that  $p(\mathbf{x}_k, \mathbf{z}_k | \mathbf{x}_{0:k-1}, \mathbf{z}_{1:k-1}) = p(\mathbf{x}_k, \mathbf{z}_k | \mathbf{x}_{k-d:k-1}, \mathbf{z}_{1:k-1})$ . It then follows that*

$$[\mathbf{J}_{0:k}]_{0:k_1-d-1 \times k_1:k_1} = ([\mathbf{J}_{0:k}]_{k_1:k_1 \times 0:k_1-d-1})^\top = \mathbf{0} \quad (22a)$$

$$[\mathbf{J}_{0:k+1}]_{0:k-d \times 0:k-d} = [\mathbf{J}_{0:k}]_{0:k-d \times 0:k-d} \quad (22b)$$

for any  $k$  and  $k_1$  such that  $k \geq k_1 > d$ .

*Proof:* See technical report [23]. ■

Here we have used the following notation

$$[\mathbf{J}_{0:k}]_{t_1:t_2 \times t_3:t_4} = \mathbb{E}_{p(\mathbf{x}_{0:k}, \mathbf{z}_{1:k})} \{-\Delta_{t_1:t_2}^{t_3:t_4} \log(p(\mathbf{x}_{0:k}, \mathbf{z}_{1:k}))\}, \quad (23)$$

where  $[\mathbf{J}_{0:k}]_{t_1:t_2 \times t_3:t_4}$  denotes the submatrix of  $\mathbf{J}_{0:k}$  that contains the rows that correspond to time  $t_1$  and  $t_2$  and the columns that correspond to time  $t_3$  to  $t_4$ . Note, that the dimension of  $[\mathbf{J}_{0:k}]_{t_1:t_2 \times t_3:t_4}$  is  $n_x(t_2-t_1+1) \times n_x(t_4-t_3+1)$ , whereas that of  $\mathbf{J}_{0:k}$  is  $n_x(k+1) \times n_x(k+1)$ .

The above lemma basically states that the matrix  $\mathbf{J}_{0:k}$  becomes block tri-diagonal, a property required for developing a recursive algorithm for  $\mathbf{J}_k$ .

**Proposition 1.** *Suppose that the conditional independence assumption of Lemma 4 holds. Then, the  $(n_x \times n_x)$  filtering information matrix  $\mathbf{J}_k$  can be computed from the following relation:*

$$\mathbf{J}_k = \mathbf{E}_k - \mathbf{D}_k^\top [\mathbf{H}_k]^{-1} \mathbf{D}_k, \quad (24)$$

with

$$\mathbf{E}_k = [\mathbf{J}_{0:k}]_{k:k \times k:k}, \quad (25a)$$

$$\mathbf{D}_k = [\mathbf{J}_{0:k}]_{k-d:k-1 \times k:k}, \quad (25b)$$

where  $\mathbf{E}_k$  and  $\mathbf{D}_k$  have size  $(n_x \times n_x)$  and  $(n_x d \times n_x)$ , respectively. The  $(n_x d \times n_x d)$  matrix  $\mathbf{H}_k$  can be updated

recursively according to the following relations

$$\mathbf{H}_k = \begin{bmatrix} \tilde{\mathbf{H}}_{22} & \tilde{\mathbf{D}}_k^2 \\ (\tilde{\mathbf{D}}_k^2)^\top & \tilde{\mathbf{E}}_k \end{bmatrix} - \begin{bmatrix} \tilde{\mathbf{H}}_{12}^\top \\ (\tilde{\mathbf{D}}_k^1)^\top \end{bmatrix} [\tilde{\mathbf{H}}_{11}]^{-1} \begin{bmatrix} \tilde{\mathbf{H}}_{12} & \tilde{\mathbf{D}}_k^1 \end{bmatrix} \quad (26a)$$

with

$$\tilde{\mathbf{H}}_k = \begin{bmatrix} \tilde{\mathbf{H}}_{11} & \tilde{\mathbf{H}}_{12} \\ (\tilde{\mathbf{H}}_{12})^\top & \tilde{\mathbf{H}}_{22} \end{bmatrix} = \begin{cases} \tilde{\mathbf{C}}_k - \mathbf{B}_{k-1}^\top \mathbf{A}_{k-1}^{-1} \mathbf{B}_{k-1}, & k = d+1 \\ \mathbf{H}_{k-1} + \tilde{\mathbf{C}}_k - \mathbf{C}_{k-1}, & k > d+1 \end{cases} \quad (26b)$$

and

$$\tilde{\mathbf{D}}_k = \begin{bmatrix} \tilde{\mathbf{D}}_k^1 \\ \tilde{\mathbf{D}}_k^2 \end{bmatrix}, \quad (26c)$$

where the different matrices are defined as follows:

$$\mathbf{A}_k = [\mathbf{J}_{0:k}]_{0:k-d-1 \times 0:k-d-1}, \quad (27a)$$

$$\mathbf{B}_k = [\mathbf{J}_{0:k}]_{0:k-d-1 \times k-d:k-1}, \quad (27b)$$

$$\mathbf{C}_k = [\mathbf{J}_{0:k}]_{k-d:k-1 \times k-d:k-1}, \quad (27c)$$

$$\tilde{\mathbf{C}}_k = [\mathbf{J}_{0:k}]_{k-d-1:k-2 \times k-d-1:k-2}, \quad (27d)$$

$$\tilde{\mathbf{D}}_k = [\mathbf{J}_{0:k}]_{k-d-1:k-2 \times k-1:k-1}, \quad (27e)$$

$$\tilde{\mathbf{E}}_k = [\mathbf{J}_{0:k}]_{k-1:k-1 \times k-1:k-1}, \quad (27f)$$

and where  $\tilde{\mathbf{D}}_k^1$  and  $\tilde{\mathbf{H}}_{11}$  are of dimension  $(n_x \times n_x)$ ,  $\tilde{\mathbf{D}}_k^2$  and  $\tilde{\mathbf{H}}_{12}^\top$  are of dimension  $(n_x(d-1) \times n_x)$ , whereas  $\tilde{\mathbf{H}}_{22}$  is a  $(n_x(d-1) \times n_x(d-1))$  dimensional matrix.

*Proof:* See technical report [23]. ■

For the computation of  $\mathbf{J}_k$ , it is required to compute  $[\mathbf{J}_{0:k}]_{k-d-1:k \times k-d-1:k}$  at each recursion. As this is a submatrix of  $\mathbf{J}_{0:k}$  it can be easily computed from the techniques introduced to compute the full matrix  $\mathbf{J}_{0:k}$ . Thus, we only have to compute an approximation of

$$[\mathbf{J}_{0:k}]_{k-d-1:k \times k-d-1:k} \approx \frac{1}{N} \sum_{i=1}^N \frac{[\nabla_{\mathbf{x}_{k-d-1:k}} p(\mathbf{x}_{0:k}^{(i)}, \mathbf{z}_{1:k}^{(i)})][\cdot]^\top}{[p(\mathbf{x}_{0:k}^{(i)}, \mathbf{z}_{1:k}^{(i)})]^2}, \quad (28)$$

which compared to (10) requires to store and update the much shorter vector  $\nabla_{\mathbf{x}_{k-d-1:k}} p(\mathbf{x}_{0:k}^{(i)}, \mathbf{z}_{1:k}^{(i)})$ . Further, instead of having to invert the  $(n_x(k+1) \times n_x(k+1))$  matrix  $\mathbf{J}_{0:k}$ , whose dimension grows at each time step, it is only required to invert matrices of constant size that do not exceed  $(n_x(d+1) \times n_x(d+1))$ . The method to recursively compute the BCRB for Model 1 is summarized in Algorithm 5.

2) *Model 2:* For Model 2, finding a recursion for  $\mathbf{J}_k$  also requires to introduce approximations. Even though the state vector  $\mathbf{x}_k$  for this model is a Markov process, i.e.  $p(\mathbf{x}_k | \mathbf{x}_{0:k-1}) = p(\mathbf{x}_k | \mathbf{x}_{k-1})$  holds, this property cannot be exploited in the pdf of the current measurement given all previous states and measurements, i.e.

$$p(\mathbf{x}_k, \mathbf{z}_k | \mathbf{x}_{0:k-1}, \mathbf{z}_{1:k-1}) = p(\mathbf{z}_k | \mathbf{x}_{0:k}, \mathbf{z}_{1:k-1}) p(\mathbf{x}_k | \mathbf{x}_{k-1}) \neq p(\mathbf{z}_k | \mathbf{x}_k, \mathbf{z}_{1:k-1}) p(\mathbf{x}_k | \mathbf{x}_{k-1}). \quad (29)$$

---

**Algorithm 5** Recursive computation of BCRB
 

---

- (1) At time  $k = 0$  do:
    - Compute the initial filtering information matrix  $\mathbf{J}_0$ , see Algorithm 1, and its inverse  $[\mathbf{J}_0]^{-1}$  which gives the BCRB for estimating  $\mathbf{x}_0$ .
  - (2) For  $k = 1, 2, \dots, d$ , do:
    - Compute the full BIM  $\mathbf{J}_{0:k}$  using Algorithm 1.
    - Compute  $\mathbf{U}[\mathbf{J}_{0:k}]^{-1}\mathbf{U}^\top$  which gives the BCRB for estimating  $\mathbf{x}_k$ .
    - If  $k = d$ , then extract from  $\mathbf{J}_{0:k}$  the submatrices  $\mathbf{A}_k = [\mathbf{J}_{0:k}]_{0:k-1 \times 0:k-1}$  and  $\mathbf{B}_k = [\mathbf{J}_{0:k}]_{0:k-1 \times k:k}$  and store them.
  - (3) For  $k = d + 1, d + 2, \dots$ , do:
    - Compute the Bayesian information submatrix  $[\mathbf{J}_{0:k}]_{k-d-1:k \times k-d-1:k}$  using Algorithm 1, but replace  $\nabla_{\mathbf{x}_{0:k}}$  by  $\nabla_{\mathbf{x}_{k-d-1:k}}$ .
    - Extract the matrices  $\mathbf{C}_k, \tilde{\mathbf{C}}_k, \mathbf{D}_k, \tilde{\mathbf{D}}_k, \mathbf{E}_k$  and  $\tilde{\mathbf{E}}_k$  which are defined in (25) and (27) and store them.
    - Evaluate  $\tilde{\mathbf{H}}_k$  from (26b).
    - Evaluate  $\mathbf{H}_k$  from (26a) and store the result.
    - Evaluate  $\mathbf{J}_k$  from (24), and compute the inverse  $[\mathbf{J}_k]^{-1}$  which gives the BCRB for estimating  $\mathbf{x}_k$ .
- 

However, we can assume that given  $\mathbf{z}_{1:k-1}$ , the dependence between  $\mathbf{z}_k$  and  $\mathbf{x}_{k-l}$  decreases rather quickly, especially when conditioned on  $\mathbf{x}_{k-l+1:k-1}$ . Thus, we can assume that there exists an integer  $d$ , such that

$$p(\mathbf{z}_k | \mathbf{x}_{0:k}, \mathbf{z}_{1:k-1}) \approx p(\mathbf{z}_k | \mathbf{x}_{k-d:k}, \mathbf{z}_{1:k-1}), \quad (30)$$

i.e.  $\mathbf{z}_k$  and  $\mathbf{x}_{k-l}$  given  $\mathbf{z}_{1:k-1}$  are approximately independent for all  $l > d$ , when we condition on  $\mathbf{x}_{k-d:k}$ .

The above assumption results again in favorable properties for  $\mathbf{J}_{0:k}$  as stated in the following lemma.

**Lemma 5.** *Suppose that given the measurements  $\mathbf{z}_{1:k-1}$ , the current measurement  $\mathbf{z}_k$  and  $\mathbf{x}_{0:k-d-1}$  are conditionally independent in the sense that  $p(\mathbf{z}_k | \mathbf{x}_{0:k-1}, \mathbf{z}_{1:k-1}) = p(\mathbf{z}_k | \mathbf{x}_{k-d:k}, \mathbf{z}_{1:k-1})$ . It then follows that*

$$[\mathbf{J}_{0:k}]_{0:k_1-d-1 \times k_1:k_1} = ([\mathbf{J}_{0:k}]_{k_1:k_1 \times 0:k_1-d-1})^\top = \mathbf{0} \quad (31a)$$

$$[\mathbf{J}_{0:k+1}]_{0:k-d \times 0:k-d} = [\mathbf{J}_{0:k}]_{0:k-d \times 0:k-d} \quad (31b)$$

for any  $k$  and  $k_1$  such that  $k \geq k_1 > d$ .

*Proof:* See technical report [23]. ■

The conditional independence assumption of Lemma 5 results in that  $\mathbf{J}_{0:k}$  has a block tri-diagonal structure that is needed for a recursive evaluation of the filtering information matrix  $\mathbf{J}_k$ .

**Proposition 2.** *Suppose that the conditional independence assumption of Lemma 5 holds. Then, the  $(n_x \times n_x)$  filtering information matrix  $\mathbf{J}_k$  can be computed from the recursion presented in Proposition 1.*

*Proof:* Since Proposition 1 requires Lemma 4 to hold and Lemma 5 contains the same conditions on the structure

of  $\mathbf{J}_{0:k}$  as Lemma 4, it follows that both lemmas will yield the same recursion as given in Proposition 1. ■

Hence, the algorithm to recursively compute the BCRB for Model 2 is essentially the same as for Model 1, which is summarized in Algorithm 5.

## V. ALGORITHMS FOR CONDITIONAL INDEPENDENCE ASSUMPTION VERIFICATION

In Section IV.D approximations have been introduced that allow a recursive computation of the filtering information matrix  $\mathbf{J}_k$  for different depths  $d$ . In the following, algorithms are presented to quantify  $d$  such that the conditional independence approximation in (21) and (30) hold.

### A. Model 1

In order to find a metric to quantify  $d$ , we decompose the conditional density as follows

$$p(\mathbf{x}_k, \mathbf{z}_k | \mathbf{x}_{0:k-1}, \mathbf{z}_{1:k-1}) = \sum_{r_k} p(\mathbf{z}_k | \mathbf{x}_k, r_k) \times p(\mathbf{x}_k | \mathbf{x}_{k-1}, r_k) \Pr\{r_k | \mathbf{x}_{0:k-1}, \mathbf{z}_{1:k-1}\}. \quad (32)$$

Of particular importance is the probability  $\Pr\{r_k | \mathbf{x}_{0:k-1}, \mathbf{z}_{1:k-1}\}$ , which tells us how well  $r_k$  can be predicted based on the information that is contained in the past states  $\mathbf{x}_{0:k-1}$  and measurements  $\mathbf{z}_{1:k-1}$ . For the approximation introduced in (21), a similar expression can be derived which is given by

$$p(\mathbf{x}_k, \mathbf{z}_k | \mathbf{x}_{k-d:k-1}, \mathbf{z}_{1:k-1}) = \sum_{r_k} p(\mathbf{z}_k | \mathbf{x}_k, r_k) \times p(\mathbf{x}_k | \mathbf{x}_{k-1}, r_k) \Pr\{r_k | \mathbf{x}_{k-d:k-1}, \mathbf{z}_{1:k-1}\}, \quad (33)$$

i.e. the two expressions differ only in their prediction probabilities. We introduce the abbreviations  $P(\ell) \triangleq \Pr\{r_k = \ell | \mathbf{x}_{0:k-1}, \mathbf{z}_{1:k-1}\}$  and  $Q(\ell) \triangleq \Pr\{r_k = \ell | \mathbf{x}_{k-d:k-1}, \mathbf{z}_{1:k-1}\}$ , and define an average Kullback-Leibler type divergence (AKLD)

$$D_{\text{AKL}}(P||Q) \triangleq \int D_{\text{KL}}(P||Q) p(\mathbf{x}_{0:k-1}, \mathbf{z}_{0:k-1}) d\mathbf{x}_{0:k-1} d\mathbf{z}_{1:k-1} \quad (34)$$

with

$$D_{\text{KL}}(P||Q) = \sum_{\ell} P(\ell) \log \left( \frac{P(\ell)}{Q(\ell)} \right), \quad (35)$$

which is equal to zero when the probabilities are equal. Note that we have introduced an average divergence in order to get rid of the conditional dependency on  $(\mathbf{x}_{0:k-1}, \mathbf{z}_{0:k-1})$ . We further introduce the average Jensen-Shannon divergence (AJSD), which is defined as

$$D_{\text{AJS}}(P||Q) = 0.5 \cdot D_{\text{AKL}}(P|(P+Q)/2) + 0.5 \cdot D_{\text{AKL}}(Q|(P+Q)/2). \quad (36)$$

In contrast to the AKLD, the AJSD is symmetric and bounded as  $0 \leq D_{\text{AJS}}(P||Q) \leq 1$ , but requires that (35) is defined with respect to the binary logarithm, and is used in the following

to quantify the depth  $d$  of the BCRB recursions. The AJSD generally cannot be computed in closed-form, due to the integral in the expression for the AKLD. We therefore resort to Monte Carlo integration techniques to approximate

$$D_{\text{AKL}}(P||Q) \approx \frac{1}{N} \sum_{i=1}^N D_{\text{KL}}(P^{(i)}||Q^{(i)}) \quad (37)$$

with  $P^{(i)} \triangleq \Pr\{r_k = \ell | \mathbf{x}_{0:k-1}^{(i)}, \mathbf{z}_{1:k-1}^{(i)}\}$  and  $Q^{(i)}$  defined accordingly, and where  $(\mathbf{x}_{0:k-1}^{(i)}, \mathbf{z}_{1:k-1}^{(i)})$ ,  $i = 1, \dots, N$ , are i.i.d. vectors such that  $(\mathbf{x}_{0:k-1}^{(i)}, \mathbf{z}_{1:k-1}^{(i)}) \sim p(\mathbf{x}_{0:k-1}, \mathbf{z}_{1:k-1})$ . For the evaluation of the AKLD, it is required to have closed-form expressions for the prediction pmfs  $\Pr\{r_k | \mathbf{x}_{0:k-1}, \mathbf{z}_{1:k-1}\}$  and  $\Pr\{r_k | \mathbf{x}_{k-d:k-1}, \mathbf{z}_{1:k-1}\}$ . These probabilities can be computed recursively using the following two lemmas.

**Lemma 6.** *The prediction pmf  $\Pr\{r_k | \mathbf{x}_{0:k-1}, \mathbf{z}_{1:k-1}\}$  can be computed for  $n = 1, \dots, k-1$  from the following recursion*

$$\Pr\{r_{n+1} | \mathbf{x}_{0:n}, \mathbf{z}_{1:n}\} = \frac{\sum_{r_n} \left( \Pr\{r_{n+1} | r_n\} p(\mathbf{z}_n | \mathbf{x}_n, r_n) p(\mathbf{x}_n | \mathbf{x}_{n-1}, r_n) \times \Pr\{r_n | \mathbf{x}_{0:n-1}, \mathbf{z}_{1:n-1}\} \right)}{\sum_{r_{n+1}} \sum_{r_n} \left( \Pr\{r_{n+1} | r_n\} p(\mathbf{z}_n | \mathbf{x}_n, r_n) p(\mathbf{x}_n | \mathbf{x}_{n-1}, r_n) \times \Pr\{r_n | \mathbf{x}_{0:n-1}, \mathbf{z}_{1:n-1}\} \right)}, \quad (38)$$

which is initialized with  $\Pr\{r_1 | \mathbf{x}_0, \mathbf{z}_{1:0}\} = \Pr\{r_1\}$ .

*Proof:* See technical report [23]. ■

**Lemma 7.** *The prediction pmf  $\Pr\{r_k | \mathbf{x}_{k-d:k-1}, \mathbf{z}_{1:k-1}\}$  can be computed for  $n = k-d+1, \dots, k-1$  from the following recursion*

$$\Pr\{r_{n+1} | \mathbf{x}_{k-d:n}, \mathbf{z}_{1:n}\} = \frac{\sum_{r_n} \left( \Pr\{r_{n+1} | r_n\} p(\mathbf{z}_n | \mathbf{x}_n, r_n) p(\mathbf{x}_n | \mathbf{x}_{n-1}, r_n) \times \Pr\{r_n | \mathbf{x}_{k-d:n-1}, \mathbf{z}_{1:n-1}\} \right)}{\sum_{r_{n+1}} \sum_{r_n} \left( \Pr\{r_{n+1} | r_n\} p(\mathbf{z}_n | \mathbf{x}_n, r_n) p(\mathbf{x}_n | \mathbf{x}_{n-1}, r_n) \times \Pr\{r_n | \mathbf{x}_{k-d:n-1}, \mathbf{z}_{1:n-1}\} \right)}, \quad (39)$$

which is initialized with  $\Pr\{r_{k-d+1} | \mathbf{x}_{k-d}, \mathbf{z}_{1:k-d}\}$ .

*Proof:* See technical report [23]. ■

The only unknown in the latter recursion is the initial pmf  $\Pr\{r_{k-d+1} | \mathbf{x}_{k-d}, \mathbf{z}_{1:k-d}\}$ , which is obtained by integrating out the past states  $\mathbf{x}_{0:k-d-1}$  from  $\Pr\{r_{k-d+1} | \mathbf{x}_{0:k-d}, \mathbf{z}_{1:k-d}\}$ . For nonlinear JMSSs, closed-form expressions for  $\Pr\{r_{k-d+1} | \mathbf{x}_{k-d}, \mathbf{z}_{1:k-d}\}$  generally do not exist. However, we can rewrite

$$\begin{aligned} \Pr\{r_{k-d+1} | \mathbf{x}_{k-d}, \mathbf{z}_{1:k-d}\} &\propto \\ &\sum_{r_{k-d}} \sum_{r_{k-d-1}} \Pr\{r_{k-d+1} | r_{k-d}\} \Pr\{r_{k-d} | r_{k-d-1}\} \\ &\times p(\mathbf{z}_{k-d} | \mathbf{x}_{k-d}, r_{k-d}) \int p(\mathbf{x}_{k-d} | \mathbf{x}_{k-d-1}, r_{k-d}) \\ &\times p(\mathbf{x}_{k-d-1}, r_{k-d-1} | \mathbf{z}_{1:k-d-1}) d\mathbf{x}_{k-d-1} \end{aligned} \quad (40)$$

and approximate  $p(\mathbf{x}_{k-d-1}, r_{k-d-1} | \mathbf{z}_{1:k-d-1})$  using Rao-Blackwellized particle filters (RBPFs), for details see [23].

## B. Model 2

Similarly to Model 1, we can decompose the conditional densities  $p(\mathbf{z}_k | \mathbf{x}_{0:k}, \mathbf{z}_{1:k-1})$  and  $p(\mathbf{z}_k | \mathbf{x}_{k-d:k}, \mathbf{z}_{1:k-1})$  to obtain expressions depending on  $\Pr\{r_k | \mathbf{x}_{0:k-1}, \mathbf{z}_{1:k-1}\}$  and  $\Pr\{r_k | \mathbf{x}_{k-d:k-1}, \mathbf{z}_{1:k-1}\}$ . Hence, we can use AJSD to quantify the depth  $d$ . The prediction probabilities can be computed from Lemma 6 and Lemma 7, with the exception that we have to replace  $p(\mathbf{x}_n | \mathbf{x}_n, r_n)$  with  $p(\mathbf{x}_n | \mathbf{x}_n)$ . Since  $p(\mathbf{x}_n | \mathbf{x}_n)$  appears in both numerator and denominator and is independent of  $r_n$ , the density  $p(\mathbf{x}_n | \mathbf{x}_n)$  cancels out, and the expressions simplify accordingly, see [23] for further details.

## VI. SIMULATION RESULTS

We assume the following jump Markov linear Gaussian system

$$\mathbf{x}_k = \mathbf{F} \mathbf{x}_{k-1} + \mathbf{v}_k(r_k), \quad (41a)$$

$$\mathbf{z}_k = \mathbf{H}(r_k) \mathbf{x}_k + \mathbf{w}_k(r_k), \quad (41b)$$

with mapping matrices  $\mathbf{F}$  and  $\mathbf{H}(r_k)$ , process noise distributed according to  $\mathbf{v}_k(r_k) \sim \mathcal{N}(\boldsymbol{\mu}_v(r_k), \mathbf{Q}(r_k))$  and measurement noise distributed according to  $\mathbf{w}_k(r_k) \sim \mathcal{N}(\boldsymbol{\mu}_w(r_k), \mathbf{R})$ . We investigate for each model (i.e. Model 1 and Model 2) an example, and assume that for both examples, the discrete mode  $r_k$  evolves according to a 2-component time-homogeneous Markov chain with initial mode probabilities  $\pi_1^1 = \pi_1^2 = 0.5$ , and transition probability matrix  $\Pi$  with elements  $\pi_{11} = \pi_{22} = 0.95$ . We further assume that the initial state for both examples is zero-mean Gaussian distributed  $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{P}_{0|0})$  with covariance matrix  $\mathbf{P}_{0|0} = \text{diag}([0.5, 0.5])$ .

We compare the following bounds and filter performances: 1) Optimal filter (in MSE sense) [6], [25], 2) Interacting multiple model Kalman filter (IMM-KF) [2], [7], 3) M-BCRB using a RBPF with optimal importance density and  $N_p = 50$  particles [14], 4) enumeration BCRB (EBCRB) [1], [10], and 5) BCRB computed from Algorithm 1 (BCRB(non-recursive)) and Algorithm 5 (BCRB(recursive)). We perform in total  $N = 50.000$  Monte Carlo runs (100.000 for Model 2) and compute the root mean square error (RMSE) according to

$$\text{RMSE}_k = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_{1,k} - \hat{x}_{1,k})^2 + (x_{2,k} - \hat{x}_{2,k})^2}, \quad (42)$$

with true state  $\mathbf{x}_k = [x_{1,k}, x_{2,k}]^T$  and estimated state  $\hat{\mathbf{x}}_k = [\hat{x}_{1,k}, \hat{x}_{2,k}]^T$ . Accordingly, every bound is computed by taking the square root of the trace of the corresponding  $(2 \times 2)$  BCRB matrix. We further compute the AJSD using a RBPF with optimal importance density [26] and  $N_p = 50$  particles from  $N = 10.000$  Monte Carlo runs (even though 1.000 runs already yielded acceptable results).

*Model 1)* We assume the following mapping matrix

$$\mathbf{F} = \begin{bmatrix} 1 & 0.632 \\ 0 & 0.368 \end{bmatrix}, \quad (43)$$



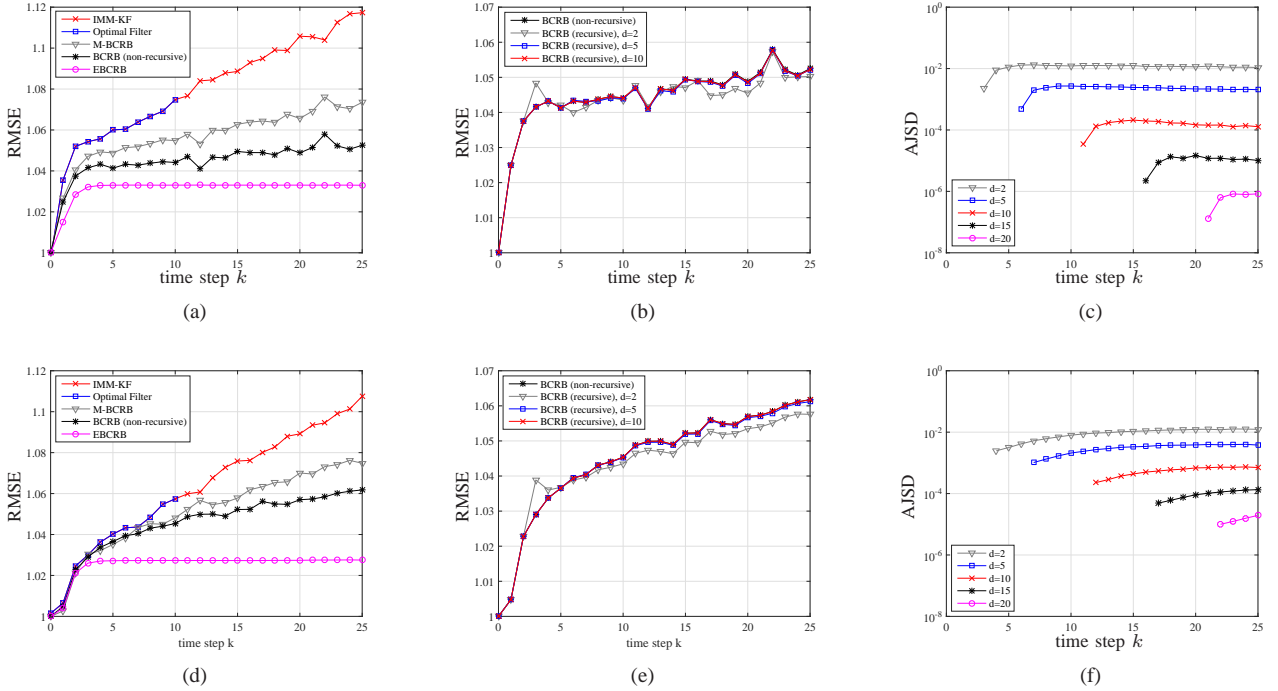


Fig. 1. Simulation results for the two examples. RMSE vs. time step  $k$  for the different bounds and algorithms are shown in (a) for Model 1 and in (d) for Model 2. A comparison of BCRB approximations with zoomed in RMSE scale is shown in (b) for Model 1 and in (e) for Model 2. The average Jensen-Shannon divergence (logarithmic scale) vs. time step  $k$  for different recursion depths  $d$  is shown in (c) for Model 1 and in (f) for Model 2.

and process noise with mean vector  $\boldsymbol{\mu}_v(1) = [0, -0.1]^T$  and  $\boldsymbol{\mu}_v(2) = [0, 0.1]^T$ , and covariance matrix  $\mathbf{Q}(1) = \text{diag}([0.5, 1])$  and  $\mathbf{Q}(2) = \text{diag}([0.35, 0.35])$ . The measurement model is defined by a mode-dependent mapping matrix  $\mathbf{H}(1) = \text{diag}([1, 1])$  and  $\mathbf{H}(2) = \text{diag}([0.8, 0.5])$ , and measurement noise with mean vector  $\boldsymbol{\mu}_w(1) = [0, 0]^T$  and  $\boldsymbol{\mu}_w(2) = [0.2, 0.5]^T$ , and covariance matrix  $\mathbf{R} = \text{diag}([1, 1])$ . *Model 2*) We assume a mode independent process model (41a), with mapping matrix  $\mathbf{F}$  defined as in (43) and process noise with mean vector  $\boldsymbol{\mu}_v = \mathbf{0}$  and covariance matrix  $\mathbf{Q} = \text{diag}([0.4, 0.4])$ . The measurement model is mode-dependent with mapping matrix

$$\mathbf{H}(1) = \begin{bmatrix} 1 & -0.2 \\ 0 & 0.5 \end{bmatrix}, \quad \mathbf{H}(2) = \begin{bmatrix} 0.8 & 0 \\ 0 & 0.5 \end{bmatrix}, \quad (44)$$

and measurement noise with mean vector  $\boldsymbol{\mu}_w(1) = [0, 0]^T$  and  $\boldsymbol{\mu}_w(2) = [0, 0.25]^T$ , and covariance matrix given by  $\mathbf{R} = \text{diag}([1, 1])$ .

The simulation results for both examples are summarized in Fig. 1. For Model 1, we can observe that the different bounds fail to predict the performance of the optimal filter, see Fig. 1(a), i.e. all bounds are rather loose. Here, the M-BCRB is the tightest bound, followed by the BCRB and the EBCRB. It is worth noting that tightness relations between the M-BCRB and BCRB have been established [13], [14], i.e. the M-BCRB is at least as tight as the BCRB. Such tightness relations cannot be established in general between the EBCRB and the M-BCRB or BCRB, i.e. there are certain problem instances where the EBCRB is tighter than the M-BCRB and/or the BCRB,

whereas for other problem instances the reverse is true. This depends on the informativeness of the model as was explained in [12], [13]. Even though the M-BCRB is the tightest bound in this example, its computation requires to run a RBPF for each Monte Carlo trial, which is much more expensive than the computation of the BCRB and EBCRB using Monte Carlo integration. In Fig. 1(b) the recursive BCRB approximations for different depths  $d$  using Algorithm 5 are compared to the BCRB obtained from Algorithm 1 (note the different RMSE scale compared to Fig. 1(a)). It can be seen that even choosing  $d = 2$  yields a fairly well approximation of the BCRB. However, differences to the BCRB are clearly visible. Increasing the depth  $d$  yields better approximations of the BCRB, with less differences, which is also reflected in the AJSD as shown in Fig. 1(c). Note, that the AJSD curves for  $k < d + 1$  are not shown since in this case the prediction pmf and its approximation are equivalent, i.e. AJSD = 0. AJSD values of  $10^{-2}$  seem to be insufficient to obtain an excellent approximation of the original BCRB. However, the more we increase  $d$ , the better this approximation becomes and an AJSD value smaller than  $3 \cdot 10^{-4}$  (or  $d = 10$ ) seems to be an appropriate choice for this example. In order to obtain a better understanding of the AJSD, consider a true probability  $p$  and an approximation  $q$  which is 1% smaller in probability than  $p$  (e.g.  $p = 0.45$  and  $q = 0.44$ ), then the JSD averaged over all possible  $p$  is  $\text{JSD}_{\text{avg.}} \approx 1.94 \cdot 10^{-4}$ . Similarly, if we assume that  $q$  is 0.1% smaller in probability than  $p$  (e.g.  $p = 0.45$  and  $q = 0.499$ ), then  $\text{JSD}_{\text{avg.}} \approx 1.7 \cdot 10^{-6}$ .

For Model 2, we can observe that the different bounds are rather loose bounds since they are relatively far away from the optimal filter performance as time increases, see Fig. 1(d). As in the previous example, the M-BCRB is the tightest bound in this setting followed by the BCRB and EBCRB. For the recursive BCRB approximations with different depths  $d$  as shown in Fig. 1(e), conclusions similar to that of Model 1 can be drawn. The AJSD values for this example are different to that of Model 1 and one should better choose a larger depth  $d$  in order to obtain an excellent approximation of the BCRB, as depicted in Fig. 1(f). It is worth noting that compared to Model 1, the AJSD curves start at one time instance later, i.e. at  $k = d + 2$ , which is a result of the fact that in Model 2 the transition pdf  $p(\mathbf{x}_k | \mathbf{x}_{k-1})$  is independent of the mode variable  $r_k$ , and the corresponding pmf and its approximation for  $k = d + 1$  are equivalent.

Note, that the AJSD is one indicator to assess the quality of the bound approximations. By simulating many other examples we found out that an AJSD value smaller than  $10^{-6}$  generally yields very good approximations with almost no differences to the original BCRB. In many other examples, values smaller than  $10^{-4}$  were already sufficient to obtain excellent approximations, but then other factors, such as the shape of the mixture density (32), i.e. if the mixture components overlap or not, play an important role. As a final remark we want to stress that the true benefit of the proposed approach is to adaptively change the depth  $d$  depending on the result of the AJSD value, as we run the algorithm for the bound computations.

## VII. CONCLUSION

In this paper, we have developed algorithms to compute the BCRB for a wide class of jump Markov systems. Our work extends previous algorithms to models where the discrete mode enters the measurement model. We have presented recursive algorithms to compute the desired bound for both the general case where the discrete mode also enters the motion model and the special case where it does not. The calculation of the BCRB involves a design parameter that determines an independency approximation and we also provide a strategy for how to select this parameter. Simulations indicate that the BCRB may provide a suitable trade-off between tractability and tightness compared to other bounds that have appeared in the literature.

## ACKNOWLEDGMENT

This work was supported in part by the Excellence Center at Linköping and Lund in Information Technology (ELLIIT).

## REFERENCES

- [1] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Boston, MA, USA: Artech-House, 2004.
- [2] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*. New York, NY, USA: Wiley-Interscience, 2001.
- [3] S. Chib and M. Dueker, "Non-Markovian regime switching with endogenous states and time-varying state strengths," *Econometric Society, Econometric Society 2004 North American Summer Meetings 600*, Aug. 2004.
- [4] F. Gustafsson, *Adaptive Filtering and Change Detection*. New York, NY, USA: John Wiley & Sons, 2000.
- [5] O. L. V. Costa, M. D. Fragoso, and R. P. Marques, *Discrete-Time Markov Jump Linear Systems*, ser. Probability and Its Applications, J. Gani, C. C. Heyde, P. Jagers, and T. G. Kurtz, Eds. London, UK: Springer-Verlag, 2005.
- [6] G. A. Ackerson and K. S. Fu, "On state estimation in switching environments," *IEEE Trans. Autom. Control*, vol. 15, no. 1, pp. 10–17, 1970.
- [7] H. A. P. Blom and Y. Bar-Shalom, "The interacting multiple model algorithm for systems with Markovian switching coefficients," *IEEE Trans. Autom. Control*, vol. 33, no. 8, pp. 780–783, 1988.
- [8] S. McGinnity and G. W. Irwin, "Multiple model bootstrap filter for maneuvering target tracking," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 36, no. 3, pp. 1006–1012, 2000.
- [9] E. Özkan, F. Lindsten, C. Fritsche, and F. Gustafsson, "Recursive maximum likelihood identification of jump Markov nonlinear systems," *IEEE Trans. Signal Process.*, vol. 63, no. 3, pp. 754–765, Mar. 2015.
- [10] A. Bessell, B. Ristic, A. Farina, X. Wang, and M. S. Arulampalam, "Error performance bounds for tracking a manoeuvring target," in *Proc. of the International Conference on Information Fusion*, vol. 1, Cairns, Queensland, Australia, Jul. 2003, pp. 903–910.
- [11] C. Fritsche, U. Orguner, L. Svensson, and F. Gustafsson, "The marginal enumeration Bayesian Cramér-Rao bound for jump Markov systems," *IEEE Signal Process. Lett.*, vol. 21, no. 4, pp. 464–468, Apr. 2014.
- [12] L. Svensson, "On the Bayesian Cramér-Rao bound for Markovian switching systems," *IEEE Trans. Signal Process.*, vol. 58, no. 9, pp. 4507–4516, Sept. 2010.
- [13] C. Fritsche and F. Gustafsson, "Bounds on the optimal performance for jump Markov linear Gaussian systems," *IEEE Trans. Signal Process.*, vol. 61, no. 1, pp. 92–98, Jan. 2013.
- [14] —, "The marginal Bayesian Cramér-Rao bound for jump Markov systems," *IEEE Signal Process. Lett.*, 2016, accepted.
- [15] F. Caron, M. Davy, E. Duflos, and P. Vanheeghe, "Particle filtering for multisensor data fusion with switching observation models: Application to land vehicle positioning," *IEEE Trans. Signal Process.*, vol. 55, no. 6, pp. 2703–2719, Jun. 2007.
- [16] M. Nicoli, C. Morelli, and V. Rampa, "A jump markov particle filter for localization of moving terminals in multipath indoor scenarios," *IEEE Trans. Signal Process.*, vol. 56, no. 8, pp. 3801–3809, Aug. 2008.
- [17] N. Viandier, D. F. Nahimana, J. Marais, and E. Duflos, "GNSS performance enhancement in urban environment based on pseudo-range error model," in *Position, Location and Navigation Symposium, 2008 IEEE/ION*, Monterey, CA, USA, May 2008, pp. 377–382.
- [18] R. J. Elliott, F. Dufour, and W. P. Malcolm, "State and mode estimation for discrete-time jump Markov systems," *SIAM Journal on Control and Optimization*, vol. 44, no. 3, pp. 1081–1104, Mar. 2005. [Online]. Available: <http://dx.doi.org/10.1137/S0363012904442628>
- [19] L. Blackmore, S. Rajamanoharan, and B. C. Williams, "Active estimation for jump Markov linear systems," *IEEE Trans. Autom. Control*, vol. 53, no. 10, pp. 2223–2236, Nov. 2008.
- [20] R. D. Gill and B. Y. Levit, "Applications of the van Trees inequality: a Bayesian Cramér-Rao bound," *Bernoulli*, vol. 1, no. 1/2, pp. 59–79, 1995.
- [21] H. L. van Trees, *Detection, Estimation and Modulation Theory Part I*. New York, NY, USA: John Wiley & Sons, 1968.
- [22] 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.
- [23] C. Fritsche and U. Orguner, "Supplementary material for 'Recent results on Bayesian Cramér-Rao bounds for jump Markov systems'," Linköping University, Linköping, Sweden, Tech. Rep. LiTH-ISY-R-3091, Feb. 2016.
- [24] C. Fritsche, F. Gustafsson, and A. Klein, "Bayesian Cramér-Rao bound for mobile terminal tracking in mixed LOS/NLOS environments," *IEEE Wireless Comm. Lett.*, vol. 2, no. 3, pp. 335–338, Mar. 2013.
- [25] F. Gustafsson, *Statistical Sensor Fusion*. Lund, Sweden: Studentlitteratur AB, 2010.
- [26] A. Doucet, N. J. Gordon, and V. Krishnamurthy, "Particle filters for state estimation of jump Markov linear systems," *IEEE Trans. Signal Process.*, vol. 49, no. 3, pp. 613–624, 2001.