COMPUTING THE RECURSIVE POSTERIOR CRAMER-RAO BOUND FOR A NONLINEAR NONSTATIONARY SYSTEM

Robert M. Taylor Jr., Brian P. Flanagan, and John A. Uber

The MITRE Corporation 7525 Colshire Drive McLean, VA 22102
{rtaylor, bflan, johnuber}@mitre.org

ABSTRACT

The recursive Posterior Cramer-Rao Bound (PCRB) has recently been shown to be the information-theoretic mean square error (MSE) bound for an unbiased sequential Bayesian estimator. The expectation integrals for the Fisher information components, which arise out of the recursive PCRB formulation, are intractable in general and must be approximated numerically. We introduce a sequential Monte Carlo method for computing the PCRB in a nonlinear nonstationary dynamic system. To validate the bound accuracy, we run a particle filter on a nonstationary logistic function and see how the MSE compares to the PCRB.

1. INTRODUCTION

The goal of this work is to derive a precise algorithm for computing the mean square error (MSE) bound on an optimal Bayesian tracker for a general system. We focus attention on the nonlinear, non-stationary system model and restrict attention to the Gaussian case only for simulation convenience. Van Trees [7] introduced the batch form of a posterior Cramer-Rao bound (PCRB) for random parameter vectors, but his formulation does not allow for a recursive implementation and suffers from computational complexity as the state vector grows linearly with time. The recursive PCRB derived in [6] gives us a formula for updating the posterior Fisher information matrix (FIM) from one time instance to the next while keeping the FIM constant in size. An alternative formulation of the recursive PCRB was done by Bergman [1], where his recursion required modifying the measurement function to be bijective. Zhang and Willet [8] extend the work of [6] to derive a PCRB expression for estimating a state with unknown measurement origin. Hue et al. [3] extend the PCRB formulation to include data association for multiple target tracking. None of these discusses how to accurately compute the expectation integrals in the recursive PCRB formulation for the general nonlinear, nonstationary, and non-Gaussian system model. In this study, we show how sequential Monte Carlo methods can be applied to estimate the FIM components in the PCRB equation for the general case. We apply our algorithm to a nonstationary logistic function as an example. In [5] we apply a special case of the algorithm to single-target sensor-sensor radar tracking.

The rest of the paper is organized as follows. We describe the PCRB for sequential Bayesian estimation in section 2 and derive methods for computing the PCRB in section 3. We show simulation results in section 5 and summarize results in section 6.

2. POSTERIOR CRAMER-RAO BOUND FOR SEQUENTIAL BAYESIAN ESTIMATION

The general form for the process and observation models with additive noise statistics can be written as

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{f}_k(\mathbf{x}_k) + \mathbf{w}_k \\ \mathbf{z}_k &= \mathbf{h}_k(\mathbf{x}_k) + \mathbf{v}_k \end{aligned} \tag{1}$$

where $\mathbf{x}_k$ is the state vector at time k we wish to estimate, $\mathbf{f}_k$ is the process model mapping the state at the current time k to the next time $k+1$, $\mathbf{z}_k$ is the measurement vector at time k, and $\mathbf{h}_k$ is the observation model mapping the current state vector to the observed measurement. The process noise and measurement noise are modeled by the random vectors $\mathbf{w}_k$ and $\mathbf{v}_k$ which can assume any distribution in principle but are generally Gaussian. The first equation gives the necessary information for the process model density function $p(\mathbf{x}_{k+1}|\mathbf{x}_k)$, and the second equation provides the observation model density function $p(\mathbf{z}_k|\mathbf{x}_k)$.

We desire a lower bound on the covariance of a tracker defined by (1). The trackers are state estimators $\hat{\mathbf{x}}_k$ for the true target state $\mathbf{x}_k$. Since we are interested in the class of trackers that are unbiased, bounding the MSE can be achieved by the PCRB alone. Assuming regularity holds for the probability density functions, the estimator covariance is:

$$P_k = E\{[\hat{\mathbf{x}}_k - \mathbf{x}_k][\hat{\mathbf{x}}_k - \mathbf{x}_k]^T\} \geq J_k^{-1} \tag{2}$$

Work supported under MITRE internal research and development.

in which J_k is the posterior FIM defined as

$$J_k = E\{-\nabla_{\mathbf{x}_k}\nabla^T_{\mathbf{x}_k}\log p(\mathbf{x}_k,\mathbf{z}_k)\}. \quad (3)$$

Tichavsky et al. [6] show that from the previous Fischer information J_k we have:

$$J_{k+1} = D_k^{22} - D_k^{21}(J_k + D_k^{11})^{-1}D_k^{12} \quad (4)$$

where

$$D_k^{11} = E_{p(\mathbf{x}_{k+1}|\mathbf{z}_{k+1})}\{-\nabla_{\mathbf{x}_k}\nabla^T_{\mathbf{x}_k}\log p(\mathbf{x}_{k+1}|\mathbf{x}_k)\} \quad (5)$$

$$D_k^{12} = E_{p(\mathbf{x}_{k+1}|\mathbf{z}_{k+1})}\{-\nabla_{\mathbf{x}_k}\nabla^T_{\mathbf{x}_{k+1}}\log p(\mathbf{x}_{k+1}|\mathbf{x}_k)\} = [D_k^{21}]^T \quad (6)$$

$$D_k^{22} = E_{p(\mathbf{x}_{k+1}|\mathbf{z}_{k+1})}\{-\nabla_{\mathbf{x}_{k+1}}\nabla^T_{\mathbf{x}_{k+1}}\log p(\mathbf{x}_{k+1}|\mathbf{x}_k)\} + E_{p(\mathbf{x}_{k+1}|\mathbf{z}_{k+1})}\{-\nabla_{\mathbf{x}_{k+1}}\nabla^T_{\mathbf{x}_{k+1}}\log p(\mathbf{z}_{k+1}|\mathbf{x}_{k+1})\} \quad (7)$$

and we initialize the recursion of (4) with:

$$J_0 = E\{-\nabla_{\mathbf{x}_0}\nabla^T_{\mathbf{x}_0}\log p(\mathbf{x}_0)\}. \quad (8)$$

3. COMPUTING THE RECURSIVE PCRB

In general, the expectations in (5)-(7) have no closed-form analytical solution and must be approximated. If we do not attempt to linearize the system model, but instead keep the nonlinear equations as they are, then we can solve for the expectation integrals using Monte Carlo integration. If either the process model or observation model is already linear, then some of the terms in (5)-(7) will simply be products of matrices. As a first step in applying Monte Carlo integration, we will need to define the following matrix functions:

$$\begin{aligned}
\Lambda^{11}(\mathbf{x}_k,\mathbf{x}_{k+1}) &= -\nabla_{\mathbf{x}_k}\nabla^T_{\mathbf{x}_k}\log p(\mathbf{x}_{k+1}|\mathbf{x}_k)\\
\Lambda^{12}(\mathbf{x}_k,\mathbf{x}_{k+1}) &= -\nabla_{\mathbf{x}_k}\nabla^T_{\mathbf{x}_{k+1}}\log p(\mathbf{x}_{k+1}|\mathbf{x}_k)\\
\Lambda^{22,a}(\mathbf{x}_k,\mathbf{x}_{k+1}) &= -\nabla_{\mathbf{x}_{k+1}}\nabla^T_{\mathbf{x}_{k+1}}\log p(\mathbf{x}_{k+1}|\mathbf{x}_k)\\
\Lambda^{22,b}(\mathbf{x}_{k+1},\mathbf{z}_{k+1}) &= -\nabla_{\mathbf{x}_{k+1}}\nabla^T_{\mathbf{x}_{k+1}}\log p(\mathbf{z}_{k+1}|\mathbf{x}_{k+1}).
\end{aligned} \quad (9)$$

We can now write equations (5)-(7) using the following:

$$\begin{aligned}
D_k^{11} &= \int \Lambda^{11}(\mathbf{x}_k,\mathbf{x}_{k+1})p(\mathbf{x}_{k+1}|\mathbf{z}_{k+1})d\mathbf{x}_{k+1}\\
D_k^{12} &= \int \Lambda^{12}(\mathbf{x}_k,\mathbf{x}_{k+1})p(\mathbf{x}_{k+1}|\mathbf{z}_{k+1})d\mathbf{x}_{k+1}\\
D_k^{22} &= \int (\Lambda^{22,a}(\mathbf{x}_k,\mathbf{x}_{k+1}) + \Lambda^{22,b}(\mathbf{x}_{k+1},\mathbf{z}_{k+1}))\times\\
&\quad p(\mathbf{x}_{k+1}|\mathbf{z}_{k+1})d\mathbf{x}_{k+1}.
\end{aligned} \quad (10)$$

These expectation integrals can be evaluated with a sample mean approximation once we have a sample representation of the posterior density. We can obtain this sample-based representation of the posterior pdf $p(\mathbf{x}_{k+1}|\mathbf{z}_{k+1})$ by borrowing on the work done in particle filtering (see [2]). Fig. 1, adapted from Isard and Blake [4], gives an illustration of how the samples in a sequential Monte Carlo process are used to represent the various density functions necessary for Bayesian tracking. The *a posteriori* samples at time

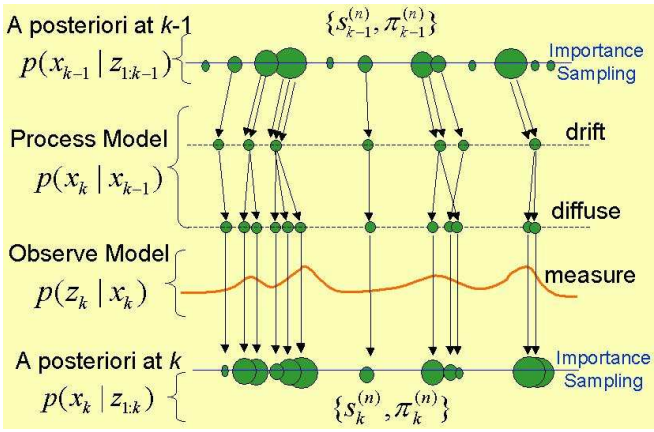

Fig. 1. Illustration of how the samples in a sequential Monte Carlo method are used to represent the various density functions necessary for Bayesian tracking. Figure is adapted from Isard and Blake [4].

$k-1$ denoted as $\mathbf{s}_{k-1}^{(n)}$ with weight $\pi_{k-1}^{(n)}$ are passed through the process model to experience drift and diffusion. The output samples of the process model are then fed into the measurement model to assign new weights to the samples. The weighted samples are then used in the sequential importance resampling (SIR) step to produce the *a posteriori* samples at time k. Therefore, using the process model density $p(\mathbf{x}_{k+1}|\mathbf{x}_k)$ and likelihood density $p(\mathbf{z}_{k+1}|\mathbf{x}_{k+1})$ we can generate weighted samples on a stochastic grid to represent the posterior density and estimate the Fisher component matrices with the empirical averages

$$\begin{aligned}
D_k^{11} &\cong \tfrac{1}{N_s}\sum_{n=1}^{N_s}\Lambda^{11}(\mathbf{s}_k^{(n)},\mathbf{s}_{k+1}^{(n)})\\
D_k^{12} &\cong \tfrac{1}{N_s}\sum_{n=1}^{N_s}\Lambda^{12}(\mathbf{s}_k^{(n)},\mathbf{s}_{k+1}^{(n)})\\
D_k^{22} &\cong \tfrac{1}{N_s}\sum_{n=1}^{N_s}(\Lambda^{22,a}(\mathbf{s}_k^{(n)},\mathbf{s}_{k+1}^{(n)}) + \Lambda^{22,b}(\mathbf{s}_{k+1}^{(n)},\mathbf{z}_{k+1}))
\end{aligned} \quad (11)$$

where $\mathbf{s}_{k+1}^{(n)}\ \forall n = 1,...,N_s$ are the *a posteriori* samples representing the density $p(\mathbf{x}_{k+1}|\mathbf{z}_{k+1})$ and N_s is the number of samples. From the Strong Law of Large Numbers we know that the empirical averages converge almost surely to the expectations in (5)-(7). We will designate the algorithm for computing the PCRB via sequential Monte Carlo integration to be PCRB-SMC and summarize it here.

PCRB-SMC

1. Initialize samples $\mathbf{s}_0^{(n)}\ \forall n = 1,...,N_s$ from $p(\mathbf{x}_0)$, compute J_0 from (8), set $\pi_0^{(n)} = \frac{1}{N_s}$, $c_0^{(n)} = c_0^{(n-1)} + \pi_0^{(n)}$, and set $k = 0$.
2. Predict by sampling $\forall n = 1,...,N_s$ from $p(\mathbf{x}_{k+1}|\mathbf{x}_k = \mathbf{x}_k^{TRUTH})$ to choose $\mathbf{s}\prime_{k+1}^{(n)}$

3. Weight samples from measurement $\mathbf{z}_{k+1}$.
$\pi_{k+1}^{(n)} = p(\mathbf{z}_{k+1}|\mathbf{x}_{k+1} = \mathbf{s\prime}_{k+1}^{(n)})$ and then normalize so that $\sum_k \pi_{k+1}^{(n)} = 1$ and store cumulative probabilities so that we propagate the triplet $(\mathbf{s\prime}_{k+1}^{(n)}, \pi_{k+1}^{(n)}, c_{k+1}^{(n)})$ where $c_{k+1}^{(0)} = 0; c_{k+1}^{(n)} = c_{k+1}^{(n-1)} + \pi_{k+1}^{(n)} (n = 1, ..., N)$.

4. For each of N_s samples select $\mathbf{s}_{k+1}^{(n)}$ as follows:
1) Generate uniform r.v. $r \in [0, 1]$.
2) Find smallest l for which $c_{k+1}^{(l)} \geq r$
3) Set $\mathbf{s}_{k+1}^{(n)} = \mathbf{s\prime}_{k+1}^{(l)}$

5. Compute $\Lambda^{11}(\mathbf{s}_k^{(n)}, \mathbf{s}_{k+1}^{(n)})$, $\Lambda^{12}(\mathbf{s}_k^{(n)}, \mathbf{s}_{k+1}^{(n)})$, $\Lambda^{22,a}(\mathbf{s}_k^{(n)}, \mathbf{s}_{k+1}^{(n)})$, and $\Lambda^{22,b}(\mathbf{s}_{k+1}^{(n)}, \mathbf{z}_{k+1})$ $\forall n = 1, ..., N_s$ from (9).

6. Estimate D_k^{11}, D_k^{12}, and D_k^{22} from (11) and set $D_k^{21} = [D_k^{12}]^T$.

7. Compute $J_{k+1} = D_k^{22} - D_k^{21}(J_k + D_k^{11})^{-1} D_k^{12}$

8. Set $k = k+1$, save the sample set $\mathbf{s}_k^{(n)}, \pi_k^{(n)}, c_k^{(n)}, \{n = 1, ..., N_s\}$, and go to step 2.

4. VALIDATING THE PCRB

One way to verify the accuracy of the PCRB is by comparing the MSE of various Bayesian trackers to the bound. The MSE matrix of an optimal Bayesian tracker should exactly match the inverse FIM matrix at each time instance. The particle filter (PF) is one type of suboptimal Bayesian estimator that performs well in nonlinear, nonstationary, and non-Gaussian systems. It also has conditions under which it converges to the optimal Bayesian tracker. There have been many variations and improvements on this fundamental approach including auxiliary particle filters, regularized particle filters, and Rao-Blackwellized particle filters. (See [2] for a thorough introduction to the theory and application of SMC methods.) For the purposes at hand, a basic PF will provide a sufficient baseline for comparison of MSE to the PCRB. Here, we simply provide the PF algorithm as given in [4]. Note that the PF algorithm is identical to the PCRB-SMC algorithm of section 3 in the resampling and weighting stages (steps 2 and 4 respectively).

PF (Isard and Blake [4])

1. Initialize samples $\mathbf{s}_0^{(n)}$ $\forall n = 1, ..., N_s$ from $p(\mathbf{x}_0)$, set $\pi_0^{(n)} = \frac{1}{N_s}$, $c_0^{(n)} = c_0^{(n-1)} + \pi_0^{(n)}$, and set $k = 1$.

2. From the previous sample set $\mathbf{s}_{k-1}^{(n)}, \pi_{k-1}^{(n)}, c_{k-1}^{(n)}, \{n = 1, ..., N\}$ at time step $k-1$

3. Select a sample $\mathbf{s\prime}_{k+1}^{(n)}$ as follows for each of N new samples:
1) Generate uniform r.v. $r \in [0, 1]$.
2) Find smallest l for which $c_{k-1}^{l} \geq r$
3) Set $\mathbf{s\prime}_k^{(n)} = \mathbf{s}_{k-1}^{(l)}$

4. Choose $\mathbf{s}_k^{(n)}$ by sampling from $p(\mathbf{x}_k|\mathbf{x}_{k-1} = \mathbf{s\prime}_k^{(n)})$

5. Measure and weight new positions in terms of measured features $\mathbf{z}_k$. $\pi_k^{(n)} = p(\mathbf{z}_k|\mathbf{x}_k = \mathbf{s}_k^{(n)})$ and then normalize so that $\sum_k \pi_k^{(n)} = 1$ and store cumulative probabilities so that we propagate the triplet $(\mathbf{s}_k^{(n)}, \pi_k^{(n)}, c_k^{(n)})$ where $c_k^{(0)} = 0; c_k^{(n)} = c_k^{(n-1)} + \pi_k^{(n)} (n = 1, ..., N)$.

6. Compute state estimate at time k as:
$\hat{\mathbf{x}}_k = E[\mathbf{x}_k] \cong \sum_{n=1}^{N} \pi_k^{(n)} \mathbf{s}_k^{(n)}$

7. Set $k = k + 1$ and go to step 2.

5. SIMULATION RESULTS

For the purpose of verifying the results by simulation we will examine the scalar nonstationary logistic function

$$\begin{aligned} x_{k+1} &= \alpha x_k(1 - x_k) + \beta \cos k + w_k \\ z_k &= \gamma x_k^3 + v_k \end{aligned} \tag{12}$$

where we choose $\alpha = 4$, $\beta = .0001$, and $\gamma = 10$ such that the evolution of this chaotic system produces states that are roughly distributed according to the arcsine distribution. To simplify calculations we let the noise statistics be Gaussian. We set $w_k \sim \mathcal{N}(\mathbf{0}, \sigma_{w_k}^2)$ and $v_k \sim \mathcal{N}(\mathbf{0}, \sigma_{v_k}^2)$ with $\sigma_{w_k} = .01$ and $\sigma_{v_k} = 1$. Fig. 2 compares the true state to the estimated state coming from the particle filter of section 4. The system is sufficiently nonlinear to cause the extended Kalman filter (not shown) to fail. The particle filter does not diverge but maintains an unbiased estimate.

We can now apply the PCRB-SMC algorithm described in section 3 to bound the MSE of the particle filter. The terms from (9) are easily computed as:

$$\begin{aligned} \Lambda^{11}(\mathbf{x}_k, \mathbf{x}_{k+1}) &= \tfrac{1}{\sigma_{w_k}^2}[2\alpha x_{k+1} + \alpha^2 - 6\alpha^2 x_k + \\ &\qquad 6\alpha^2 x_k^2 - 2\alpha\beta \cos k] \\ \Lambda^{12}(\mathbf{x}_k, \mathbf{x}_{k+1}) &= \tfrac{1}{\sigma_{w_k}^2}[-\alpha + 2\alpha x_k] \\ \Lambda^{22,a}(\mathbf{x}_k, \mathbf{x}_{k+1}) &= \tfrac{1}{\sigma_{w_k}^2} \\ \Lambda^{22,b}(\mathbf{x}_{k+1}, \mathbf{z}_{k+1}) &= \tfrac{1}{\sigma_{v_k}^2}[12\gamma z_{k+1} x_{k+1} + 30\gamma^2 x_{k+1}^4] \end{aligned} \tag{13}$$

Fig. 3 shows the MSE of the particle filter run over 100 experiments compared with the PCRB. We estimate the terms in (11) using $N_s = 500$ samples. We see that the bound generally is lower or matches the MSE for most points in time.

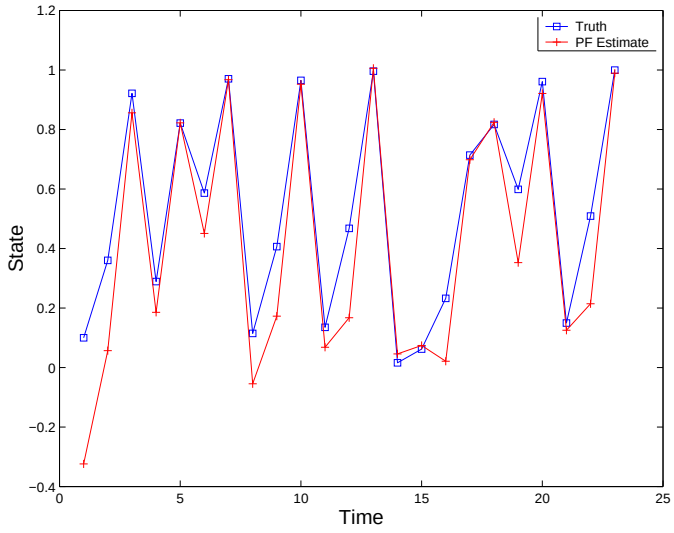

Fig. 2. True and particle filter (PF) estimated state for a nonstationary logistic function

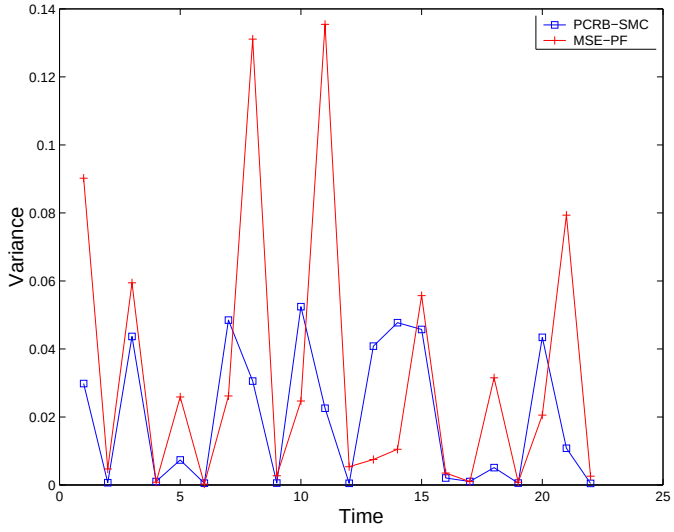

Fig. 3. Comparison of the mean square error of a particle filter (MSE-PF) to the posterior Cramer-Rao bound as computed by sequential Monte Carlo integration (PCRB-SMC).

6. CONCLUSION

The PCRB accurately bounds the mean square error for unbiased sequential Bayesian estimators as shown theoretically and demonstrated against a nonlinear MMSE estimator–namely a particle filter. Since the PCRB can only be approximated due to the intractable expectation integrals, we derived a method for approximating the PCRB. The method uses sequential Monte Carlo integration and converges almost surely to the PCRB. The formulation of the bound is general enough to account for any arbitrary system model. We designed a nonstationary logistic function to test our results. We ran the PCRB computation algorithm and the particle filter on the same system model and showed that the PCRB does indeed bound the MSE of a nonlinear nonstationary Bayesian estimator.

7. REFERENCES

[1] N. Bergman, Recursive Bayesian Estimation: Navigation and Tracking Applications, Ph.D. thesis, Department of Electrical Engineering. Linköping Studies in Science and Technology. Dissertations No. 579.

[2] A. Doucet, N. de Freitas, N. Gordon, *Sequential Monte Carlo Methods in Practice*, Springer-Verlag New York, Inc., 2001.

[3] C. Hue, J. Le Cadre, P. Perez, Performance analysis of two sequential Monte Carlo methods and posterior Cramer-Rao bounds for multi-target tracking, *5th Int. Conf. on Information Fusion, FUSION'02*, Annapolis, Maryland, July 2002.

[4] M. Isard, A. Blake, CONDENSATION – conditional density propagation for visual tracking, *Int. J. Computer Vision*, 29, 1, 5–28, 1998.

[5] Taylor, R., Flanagan, B., Uber, J., Computing the Recursive Posterior Cramer-Rao Bound for a Radar Tracking System, to be submitted to *IEEE Transactions on Aerospace and Electronic Systems*

[6] P. Tichavsky, C. Muravchik, A. Nehorai, Posterior Cramer-Rao Bounds for Discrete-Time Nonlinear Filtering. *IEEE Transactions on Signal Processing*, Vol. 46, No. 5, May 1998.

[7] H. Van Trees, *Detection, Estimation, and Modulation Theory: Part 1*, John Wiley and Sons, Inc., New York, 1968.

[8] X. Zhang, P. Willett, Cramer-Rao Bounds for Discrete-Time Linear Filtering with Measurement Origin Uncertainties. Workshop on Estimation, Tracking, and Fusion: A Tribute to Yaakov Bar-Shalom, May 2001.