Sensor Validation for Flight Control by Particle Filtering

Tao Wei, Yufei Huang and Philip Chen
Department of Electrical Engineering
The University of Texas at San Antonio
San Antonio, TX 78249 US
Email: twei@lonestar.utsa.edu, Yufei.Huang@utsa.edu, Philip.Chen@utsa.edu

***Abstract*— In this paper, we address the problem of adaptive sensor validation for flight control. The model-based approaches are developed, where the sensor system is modeled by a Markov switch dynamic state-space model. To handle the nonlinearity of the problem, two different particle filters: mixture Kalman filter (MKF) and stochastic M-algorithm (SMA) are proposed. Simulation results are presented to compare the effectiveness and complexity of MKF and SMA methods.**

Index Terms—Mixture Kalman Filter (MKF), Stochastic M-Algorithm (SMA), sensor validation, sensor failure, Fault Detection and Isolation (FDI), Monte-Carlo technique, particle filter (PF)

I. Introduction

Fault detection and isolation (FDI) is an important procedure for safe flight operation and, due to the complicate nature of FDI, much research has been devote to improving its performance [1][2]. Usually, FDI approaches fall into two major categories: the model-based and the knowledge-based. The knowledge-based approaches use qualitative models based on the available information and knowledge of the flight system to be monitored. When the dynamic behavior of systems can be well-described quantitatively by mathematical models, the model-based methods are more powerful for FDI. Kalman filters are commonly used for state estimation in the model-based methods, when the system is linear, and the noise and disturbance are Gaussian [2]. When, in more realistic cases, nonlinear models are used, extended Kalman filters (EKF) [3] are proposed, where linearization of the nonlinear system model is required. Although EKF is straightforward to implement, there is no guarantee that it works well in most cases.

A special topic of FDI for flight control systems is the sensor validation (SV) and it caught great attentions from researchers recently, owing mainly to the nonlinear nature of the problem. To its solutions, model-based approaches have been arguably more favorable and this is especially true for SV with a stochastic hybrid system that has both continuous and discrete random variables, since only model-based SV methods are suitable for the solution. In this paper, we focus on the model-based methods. For model-based methods, when systems are with an unknown but constant structure, the "noninteracting" MM (NMM) methods can be used for the optimal solution [4][5], in which multiple filters run independently in parallel without mutual interactions. However, for most of the SV problems, models are interactive and have transitions in between over time. That is to say the structure of systems changes and thereby the "noninteracting" MM methods are no longer suitable. Instead, "interacting" Multiple-model (IMM) was developed for improved performance [6]. The IMM enables a set of single-model-based filters interacting between each other in a remarkably cost-effective way and thus provides significant improvement over "noninteracting" MM methods [7]. From signal processing point of view, SV with a hybrid model resembles the problem of symbol decoding over flat-fading channels, where one can treat the sensor state as unknown "transmitted symbol" and impute all possibilities of it on the current observation. Such analogy motivates us to propose a mixture Kalman filter (MKF) solution in this paper, a unique particle filtering detector developed for the aforementioned problem of symbol decoding[8]. As an additional improvement to the MKF, we further propose a more efficient particle filtering algorithm known as the stochastic M-Algorithm (SMA), which was introduced in [9]. With the SMA, we are able to achieve better performance than the MKF using much lower computation.

The rest of the paper is organized as follows. In section II, the problem of sensor failures on an F/A aircraft model is formulated. In section III, we develop an MKF-based solution and in section IV an SMA solution is discussed. Simulations are provided in section V to compare the performance and complexity of the MKF and SMA. Concluding remarks is given in section VI.

II. Problem Formulation

We adopt an F/A aircraft model from [7]. Under the normal (no fault) sensor condition, an aircraft system can be formulated by a dynamic state-space model (DSSM):

$$\begin{aligned} \mathbf{x}_t &= \mathbf{F}\mathbf{x}_{t-1} + \mathbf{G}\mathbf{u}_t + \boldsymbol{\omega}_t \\ \mathbf{y}_t &= \mathbf{H}\mathbf{x}_t + \boldsymbol{\nu}_t. \end{aligned} \tag{1}$$

In (1), $\mathbf{x}_t = [u_t\ w_t\ q_t\ \theta_t]^\top$ is a state vector with u_t and w_t representing the velocities in forward and vertical directions of the body axes, q_t representing the pitch angular rate, and

This work was supported by the AFOSR under Contract No. FA9550-04-1-0254.

θ_t representing the pitch angle; the input vector $\mathbf{u_t}$ is a vector of four different longitudinal control inputs: $\mathbf{u}_t = [\delta_e\ \delta_{sle}\ \delta_{ste}\ \delta_{ast}]^\top$; $\mathbf{F}$ and $\mathbf{G}$ are all known coefficient matrices of the state equation; y_t is the sensor reading; $\mathbf{H}$ is the measurement matrix; $\boldsymbol{\omega}_t$ and $\boldsymbol{\nu}_t$ are all white Gaussian noise vectors, i.e., $\boldsymbol{\omega}_t \sim \mathcal{N}(0, \sigma_\omega^2 \mathbf{I})$; $\boldsymbol{\nu}_k \sim \mathcal{N}(0, \sigma_\nu^2 \mathbf{I})$.

To formulate sensor failure, we define a discrete sensor state variable $s_t \in \{0, 1, \cdots, M-1\}$. In particular, when $s_t = 0$, the system is in the normal state (no fault), and when $s_t = i > 0$, the system is in the ith faulty state, which represents that the ith sensor reading is a fault. Here, we only consider a simple scenario: one sensor fails at a time. We further assume that the transitions between states follow a first order Markov chain as presented in [7]. The resulting formulation is a Markov switch system described by the following new hybrid state-space equations

$$\mathbf{x}_t = \mathbf{F}\mathbf{x}_{t-1} + \mathbf{G}\mathbf{u} + \boldsymbol{\omega}_t \tag{2}$$

$$\mathbf{p}(s_t = n | s_{t-1} = m) = \boldsymbol{\Pi}_{mn} \tag{3}$$

$$\mathbf{y}_t = \mathbf{H}_{s_t} x_t + \boldsymbol{\nu}_t \tag{4}$$

where $\boldsymbol{\Pi}_{mn}$ denotes the (m, n)th element of the transitional matrix $\boldsymbol{\Pi}$ and in this model, the measurement matrix $\mathbf{H}_{s_t}$ is different for different sensor state. For the normal sensor state, $\mathbf{H_{s_t}}$ is an identity matrix, but, for the ith sensor failure, the elements in the ith row of the measurement matrix $\mathbf{H_{s_t}}$ are zeros. Based on the sensor readings up to time t, our objective is to detect at time t whether there is a faulty reading and then isolate the faulty sensor, or in another words, to estimate the sensor state s_t. Note the system state vector $\mathbf{x}_t$ is also unknown.

To form the solution, we adopt the maximum *a posteriori* (MAP) criterion, which is expressed as

$$\{\hat{s}_t\} = \arg\max_{s_t}\ p(s_t | \mathbf{y}_{1:t}) \tag{5}$$

where we use subscript $_{1:t}$ to denote a collection of variables from 1 to t. Note $p(s_t|\mathbf{y}_{1:t})$ is the marginalized *a posteriori* probabilities (APP), which is calculated according to

$$p(s_t|\mathbf{y}_{1:t}) = \sum_{\mathbf{s}_{1:t-1}} \int_{\mathbf{x}_{1:t}} p(\mathbf{s}_{1:t}, \mathbf{x}_{1:t} | \mathbf{y}_{1:t}) d\mathbf{x}_{1:t} \tag{6}$$

The difficulty in obtaining the MAP solution is the calculation of the APP by (6), where the size of the summation increases exponentially with t, making the problem NP hard.

III. Sensor validation with MKF

The mixture Kalman filter (MKF), proposed in [10], is an efficient particle filtering algorithm for the conditional dynamic linear models (CDLMs) such as equation (4), where, given the discrete variable s_t, $\mathbf{y}_t$ is linear in the continuous state vector $\mathbf{x}_t$. The MKF can be used for online filtering and prediction of the CDLMs and thereby is suitable for sensor validation.

The MKF is a Rao-Blackwellized particle filter, where samples are drawn only for sensor state s_t and a mixture of Gaussian distribution is used instead of approximating the posterior distribution of $\mathbf{x}_t$. Let us start first with the derivation of the importance function. For the MKF, the optimal importance function $p(s_t|\mathbf{s}^{(j)}_{1:t-1}, \mathbf{y}_{1:t})$ is used, which can be obtained by

$$\begin{aligned} & p(s_t|\mathbf{s}^{(j)}_{1:t-1}, \mathbf{y}_{1:t}) \\ & \quad \propto \quad p(\mathbf{y}_t|s_t^{(j)}, \mathbf{s}^{(j)}_{1:t-1}, \mathbf{y}_{1:t-1}) \cdot p(s_t|\mathbf{s}^{(j)}_{1:t-1}, \mathbf{y}_{1:t-1}) \\ & \quad = \quad \int p(\mathbf{y}_t, \mathbf{x}_t | s_t, \mathbf{s}^{(j)}_{1:t-1}, \mathbf{y}_{1:t-1}) d\mathbf{x}_t \cdot p(s_t|\mathbf{s}^{(j)}_{1:t-1}) \\ & \quad = \quad \int p(\mathbf{y}_t|\mathbf{x}_t, s_t) \\ & \qquad\qquad \cdot p(\mathbf{x}_t|s_t, \mathbf{s}^{(j)}_{1:t-1}, \mathbf{y}_{1:t-1}) d\mathbf{x}_t \cdot p(s_t|s^{(j)}_{t-1}) \end{aligned} \tag{7}$$

where $^{(j)}$ denotes the jth sample trajectory. In (7), $p(\mathbf{y}_t|\mathbf{x}_t, s_t)$ is the conditional likelihood, which is a Gaussian distribution, i.e.,

$$p(\mathbf{y}_t|\mathbf{x}_t, s_t) \quad = \quad \mathcal{N}(\mathbf{H}_{s_t}\mathbf{x}_t, \sigma_\nu^2 \mathbf{I}).$$

Furthermore, $p(\mathbf{x}_t|s_t, \mathbf{s}^{(j)}_{1:t-1}, \mathbf{y}_{1:t-1})$ is the predictive density, which can be obtained from the Kalman filtering as a Gaussian

$$p(\mathbf{x}_t|s_t, \mathbf{s}^{(j)}_{1:t-1}, \mathbf{y}_{1:t-1}) = \mathcal{N}(\mu^{(j)}_{x_{t|t-1}}, \mathbf{C}^{(j)}_{x_{t|t-1}}) \tag{8}$$

where

$$\mu^{(j)}_{x_{t|t-1}} \quad = \quad \mathbf{F}\mu_{x_{t-1}} + \mathbf{G}\mathbf{u} \tag{9}$$

$$\mathbf{C}^{(j)}_{x_{t|t-1}} \quad = \quad \mathbf{F}\mathbf{C}_{x_{t-1}}\mathbf{F}^\top + \sigma_\omega^2 \mathbf{I}. \tag{10}$$

and $\mu_{x_{t-1}}$ and $\mathbf{C}_{x_{t-1}}$ are the mean and the variance of the posterior distribution $p(\mathbf{x}_{t-1}|\mathbf{s}^{(j)}_{1:t-1}, \mathbf{y}_{1:t-1})$. Consequently, the importance function in (7) can be expressed as

$$p(s_t|\mathbf{s}^{(j)}{}_{1:t-1}, \mathbf{y}_{1:t}) \quad \propto \quad \mathcal{N}(\mu^{(j)}_{y_t}(s_t), \mathbf{C}^{(j)}_{y_t}(s_t)) \cdot \boldsymbol{\Pi}_{mn} \tag{11}$$

where

$$\mu^{(j)}_{y_t}(s_t) \quad = \quad \mathbf{H}_{s_t}\mu^{(j)}_{x_{t|t-1}} \tag{12}$$

$$\mathbf{C}^{(j)}_{y_t}(s_t) \quad = \quad \mathbf{H}_{s_t}\mathbf{C}^{(j)}_{x_{t|t-1}}\mathbf{H}_{s_t}^\top + \sigma_\nu^2 \mathbf{I} \tag{13}$$

Having derived the importance function, we implement the MKF in the following five major steps:

1) Draw samples;
2) Calculate weights;
3) Calculate the MAP solution;
4) Resample the weight samples;
5) Update state information.

In the first step, we need to draw samples from the importance function. Notice that the importance function is a discrete distribution on, say, M possible sensor states and thus sampling can be easily performed. Next, in the second step, after we obtain the samples, a weight needs to calculate for each sample. Suppose that for the jth trajectory we have a weight at time $t-1$ as $w^{(j)}_{t-1}$. The weight $w^{(j)}_t$ at t can be then computed by

$$w^{(j)}_t = u^{(j)}_t w^{(j)}_{t-1}, \tag{14}$$

where the incremental weight $u^{(j)}_t$ is defined as

$$u^{(j)}_t \propto \sum_{s_t=0}^{M-1} p(\mathbf{y}_t|s_t, \mathbf{s}^{(j)}_{1:t-1}, \mathbf{y}_{1:t-1}) p(s^{(j)}_t|s^{(j)}_{t-1}) \tag{15}$$

In the third step, once we have the samples and weights, the MAP solution in (5) can be easily approximated by the state that has the largest cumulative weights. After these steps, we might observe a large amount of relatively small weights. Small weight means that the sample is drawn far from the mean of the posterior distribution and thus has small contribution to the final MAP estimation. Therefore, the subsequent MKF implementation will be inefficient when where is a lot of small weighted samples and this effect is called sample degeneracy. A useful method to reduce sample degeneracy is called resampling. In the fourth step, resampling is performed on the samples and the weights obtained in Step 1 and 2. However, resampling will be performed only at the instances when

$$\bar{J}_t \leq \frac{J}{10} \tag{16}$$

where J is the total number of samples, $\bar{J}_t$ is the effective sample size and defined as $\bar{J}_t \triangleq \frac{J}{1+v_t^2}$, and v_t is a coefficient of variation and defined as

$$v_t^2 \;=\; \frac{1}{J}\sum_{j=1}^{J}\left(\frac{w_t^{(j)}}{\hat{w}_t}-1\right)^2 \tag{17}$$

with $\hat{w}_t = \sum_{j=1}^{J} w_t^{(j)}/J$.

In the last step, we need to update the state information by calculating the posterior distribution of $\mathbf{x}_t$, to be prepared for the next MKF iteration at $t+1$. This step is exactly like the one-step filtering update in the Kalman filter and can be more explicitly expressed by

$$p(\mathbf{x}_t|\mathbf{s}_{1:t}^{(j)}, \mathbf{y}_{1:t}) = \mathcal{N}(\mu_{x_t}^{(j)}, \mathbf{C}_{x_t}^{(j)}), \tag{18}$$

where

$$\mu_{x_t}^{(j)} \;=\; \mu_{x_{t|t-1}}^{(j)} + \mathbf{K}(\mathbf{y}_t - \mathbf{H}_{s_t}\mu_{x_{t|t-1}}^{(j)}) \tag{19}$$

$$\mathbf{C}_{x_t}^{(j)} \;=\; (\mathbf{I}-\mathbf{K}\mathbf{H}_{s_t})\mathbf{C}_{x_{t|t-1}}^{(j)} \tag{20}$$

$$\mathbf{K}^{(j)} \;=\; \mathbf{C}_{x_{t|t-1}}^{(j)}\mathbf{H}_{s_t}^{\top}(\mathbf{H}_{s_t}\mathbf{C}_{x_{t|t-1}}^{(j)}\mathbf{H}_{s_t}^{\top}+\sigma_v^2)^{-1}, \tag{21}$$

where $\mathbf{K}$ is called the Kalman filter gain.

After the above five steps, current iteration is finished and we obtain the estimated sensor state s_t, and the mean and covariance matrix of $\mathbf{x}_t$ for the next iteration. The MKF is summarized as follows:

Algorithm: Mixture Kalman Filter (MKF)

- For $j = 1$ to J,
  1) **Predictive step:**
     Calculate $\mu_{x_{t|t-1}}^{(j)}$ from (9) and $\mathbf{C}_{x_{t|t-1}}^{(j)}$ from (10)
  2) **Sampling step:**
     a) For $s_t = 1$ to M, calculate
        – $\mu_{y_t}^{(j)}(s_t)$ from (12)
        – $\mathbf{C}_{y_t}^{(j)}(s_t)$ from (13)
        – $\mathbf{\Pi}_{s_{t-1},s_t}$ from (3)
        – $q(s_t^{(j)}|\mathbf{s}^{(j)}{}_{1:t-1}, \mathbf{y}_{1:t})$ from (11).
     b) Sample $n \in \{0, 1, 2, ..., M-1\}$ with probability proportional to $q(s_t^{(j)}|\mathbf{s}^{(j)}{}_{1:t-1}, \mathbf{y}_{1:t})$;
     c) Set $s_t^{(j)} = n$;
     d) Calculate incremental $u_t^{(j)}$ from (15) and the unnormalized weight $w_t^{(j)}$ from (14).
  3) **Updating step:** Calculate
     – $\mathbf{K}^{(j)}$ from (21);
     – $\mu_{x_t}^{(j)}$ from (19);
     – $\mathbf{C}_{x_t}^{(j)}$ from (20).
- Form the new trajectories $\{s_t^{(j)}, \mathbf{s}_{1:t-1}^{(j)}\}_{j=1}^{J}$;
- Normalize the weight as $\bar{w}_t^{(j)} = w_t^{(j)} / \sum_{j=1}^{J} w_t^{(j)}$.

IV. Sensor validation with SMA

A more efficient particle filter than the MKF for online filtering and prediction of the CDLMs is called stochastic M-algorithm (SMA). The SMA was proposed in [9]. Compared with the MKF, it can provide similar performance but with much reduced computation. SMA uses discrete Delta functions as importance functions. Unlike the MKF, which produces only one sample in each trajectory, the SMA takes all possible states as samples and then extends each of the sample into a separate trajectory. One of the key features of the SMA is that no two trajectories are identical. This gives SMA more diversities even with less trajectories than the MKF. The SMA also adopts optimal resampling scheme [11] to control the exponential expansion of the trajectories. To guarantee dissimilar trajectories, the optimal resampling uses sampling without replacement that prevents sample replication. The SMA for this problem is outlined as below:

Algorithm: Stochastic M-Algorithm (SMA)

- **Trajectory expansion**
  For $j = 1$ to J
  – Perform *Predictive step* in Algorithm MKF;
  – Perform 2).a) in in Algorithm MKF;
  – For n = 1:M Set $s_t^{(M*(j-1)+n)} = n$ and calculate the the weight by $\bar{w}_t^{(M*(j-1)+n)} = p(s_t^{(M*(j-1)+n)}|s^{(j)}{}_{1:t-1}, \mathbf{y}_{1:t})w_{t-1}^{(j)}$;
  – Form $M * J$ new trajectories by setting $\mathbf{s}_t^{(M*(j-1)+n)} = \{s_t^{(M*(j-1)+n)}, s_{1:t}^{(j)}\}$.
- Normalize the weights $\bar{w}_t^{(j)}$;
- **Trajectory selection:** Select J trajectories from $M * J$ trajectories using the optimal resampling algorithm.
- **Updating step:** For $j = 1$ to J;
  Performance the *Updating step* in Algorithm MKF.

V. Simulation Results

In the section, we compared the performance of the MKF and the SMA through computation simulation. The setting of transition matrix Π and the coefficient matrices $\mathbf{F}$ and $\mathbf{G}$

TABLE I. Performance of MKF with different sample size

SNR(dB)	-30	-25	-20	-15	-10	-5	0
# of errors (15 samples)	51.23	26.57	18.89	11.53	8.49	6.13	2.27
# of errors (50 samples)	50.8	30.57	20.8	13.94	8.66	4.56	2.07

TABLE II. Performance of SMA and MKF with the same sample size

SNR(dB)	-30	-25	-20	-15	-10	-5	0
# of errors (SMA)	29.27	17.33	14.63	10.54	7.06	5.5	1.58
# of errors (MKF)	51.23	26.57	18.89	11.53	8.49	6.13	2.27

can be found in [7]. In addition, we chose constant input $\mathbf{u} = [-5.0\ 0.0\ 0.0\ 0.0]^\top$. In Table I, we show the performance of the MKF at two different sample sizes: 15 samples and 50 samples. The results are in terms of errors per 1000 time stamps at different SNRs. The signal power was chosen as the smallest average state power in the the state vector. We notice that the performance does not change much when sample size is increased from 15 to 50. Therefore, we claim that 15 samples are good enough to show the performance of the MKF. Next, in Table II, we compared the performance between the SMA and the MKF with the same criterion in Table I except that, this time, we used the same sample size (15 samples) for both the SMA and the MKF. From the result, we can tell that the over all performance of SMA is superior to that of the MKF.

To demonstrate ability of the SMA in tracking the failure variation, we showed in Figure 1 and 2 the real sensor state and the estimate sensor state, respectively, over 500 time stamps at SNR = -20dB. The only differences between these two figures are at the first seven and the 395th time stamps. We can see that, even if the SMA made a wrong decision at the 395th time stamp, it can jump back to the right state in the next time stamp.

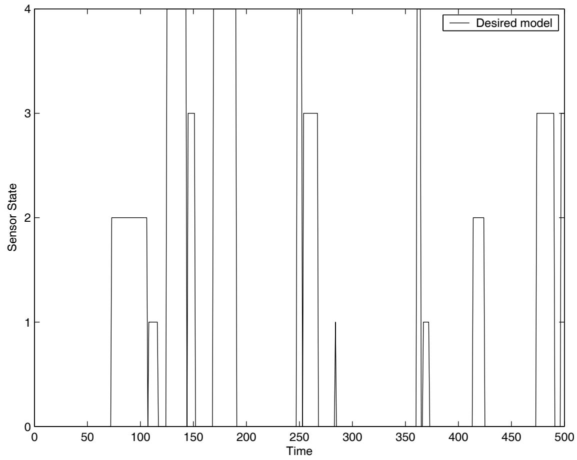

Fig. 1. Desired Sensor State

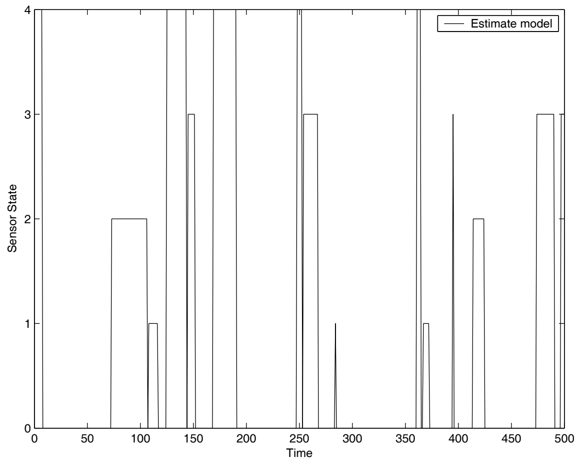

Fig. 2. Estimate Sensor State

VI. CONCLUSIONS

We have developed two particle filtering methods to solve the sensor validation problems in flight control. Both of these methods can on-line track sensor state efficiently. Between these two methods, the SMA has less complexity. Also under the same sample size, the SMA produces less errors than the MKF.

REFERENCES

[1] M. Basseville, "Detecting changes in signals and systems–a survey," *Automatica*, vol. 24, no. 3, pp. 309–326, 1988.

[2] A. S. Willsky, "A survey of design methods for failure detection in dynamic systems," in *Automatica,*, vol. 12, pp. 601-611, 1976.

[3] D. H. Zhou, Y. G. Xi, and Z. J. Zhang, "Nonlinear adaptive fault detection filter," *Int. J. Syst. Sci.*, vol. 22, pp. 2563–2571, 1991.

[4] D. T. Magill, "Optimal adaptive estimation of sampled stochastic processes," *IEEE Transactions on Automatic Control*, vol. AC-10, pp. 434-439, 1965.

[5] A. S. Willsky and $et\ al.$, "Dynamic model-based techniques for the detection of incidents on freeways," in *IEEE Trans. Automat. Contr.*, vol. AC-25, June 1980.

[6] Y. Bar-Shalom and X. R. Li, *Estimation and Tracking: Principles, Techniques, and Software*. Boston, MA: Artech House, 1993.

[7] Y. M. Zhang and X. R. Li, "Detection and diagnosis of sensor and actuator failures using IMM estimator," *IEEE Tansaction on Aerospace and Electronic Systems*, vol. 34, no. 4, pp. 1293–1313, Oct 1998.

[8] R. Chen, X. Wang, and J. S. Liu, "Adaptive joint detection and decoding in flat-fading channels via mixture Kalman filtering," *IEEE Transactions on Information Theory*, vol. 46, no. 6, pp. 2079–2094, Sept. 2000.

[9] Y. Huang, J. Zhang, and P. M. Djurić, "Bayesian detection for BLAST," *IEEE Transactions on Signal Processing*, May 2004.

[10] R. Chen and J. Liu, "Mixture Kalman filters," *Journal of Royal Statist. Soc. B*, vol. 62, pp. 493–508, 2000.

[11] P. Fearnhead, "Sequential monte carlo method in filter theory," Ph.D. dissertation, Oxford University, England, 1998.