Supplementary material for:
Recursive parameter estimation for jump Markov linear systems

Mehmet Çetinkaya[a], Emre Özkan[b]

[a] *Department of Electrical and Electronics Engineering, Middle East Technical University, Turkey*
[b] *Institute of Industrial Automation and Software Engineering, University of Stuttgart, Germany*

Abstract

This supplementary material contains further examples to "Recursive parameter estimation for jump Markov linear systems" [1] as well as the derivation of the sufficient statistics for the state transition and measurement matrices of a jump Markov linear system.

Contents

1 Simulation Results **1**
1.1 Second Order System 2
1.2 Fourth Order System with Unknown Noise Characteristics 4
1.3 Data Requirements of EM Algorithms on the First-Order System 5
1.4 Data Requirements of EM Algorithms on the Higher-Order Systems 6
1.5 Impact of Parameter Estimates on State Estimation Accuracy 9

Appendix A Predictive Density and Merging Weights **11**

Appendix B Mode-Matched Backward Transition Density **11**

Appendix C Derivation of the Sufficient Statistics **12**

1. Simulation Results

In this section, the performance of the proposed online EM algorithm is evaluated on two simulation examples adapted from the literature [2] and [3]. We first investigate the transfer function identification problem for a 2nd order systems. The second simulation demonstrates identifying the measurement noise characteristics of a 4th order system. We then present the complete set of estimated parameters for the examples discussed in the main manuscript, where the data requirements of both methods were analyzed. To assess the quality of the parameter estimates obtained by the online EM algorithm, we report the state root-mean-square errors (RMSEs), comparing the performance of GPB2 when run with the true parameters versus the estimates produced by the online EM.

1.1. Second Order System

Consider the system given in [2], whose transfer functions and transition probability matrix are

$$H_1(z) = \frac{0.7406z^{-1} + 0.004861z^{-1}}{1 + 0.6178z^{-1} + 0.4385z^{-2}}, \tag{1a}$$

$$H_2(z) = \frac{-1.461z^{-1} + 1.98z^{-1}}{1 - 1.1814z^{-1} + 0.2715z^{-2}}, \tag{1b}$$

$$\Pi = \begin{bmatrix} 0.6 & 0.4 \\ 0.5 & 0.5 \end{bmatrix}. \tag{1c}$$

These transfer function representations can be converted to state-space models as follows. Given

$$H_k(z) = \frac{b_1^k z^{-1} + b_2^k z^{-2}}{1 + a_1^k z^{-1} + a_2^k z^{-2}} \tag{2}$$

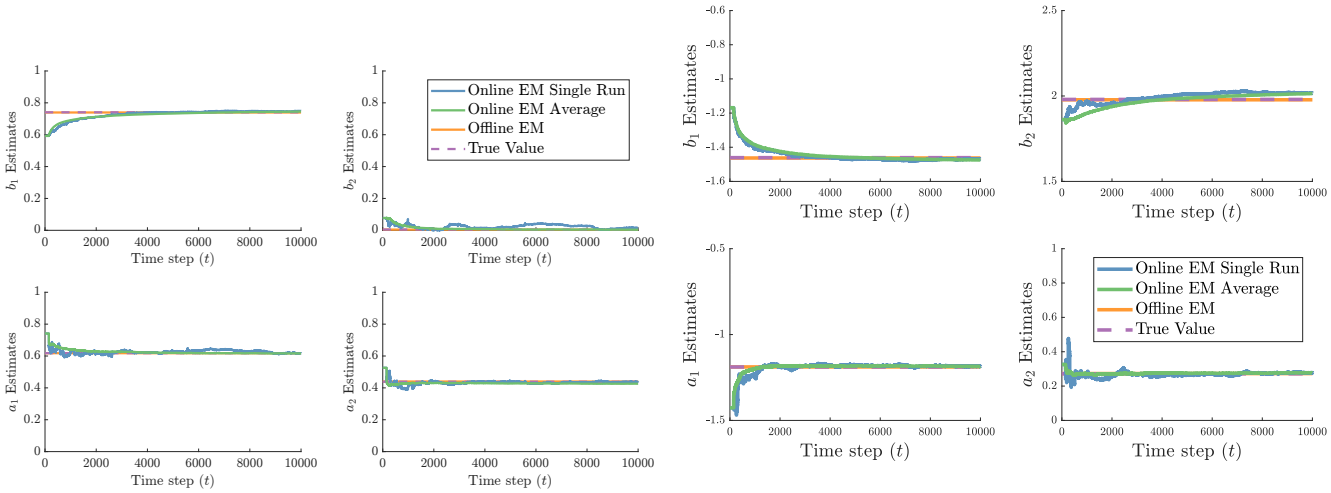
the state transition, measurement and input matrices of the corresponding model in controllable canonical form are

$$F^k = \begin{bmatrix} 0 & 1 \\ -a_2^k & -a_1^k \end{bmatrix}, H^k = \begin{bmatrix} b_2^k & b_1^k \end{bmatrix}, G^k = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \tag{3}$$

respectively. We run the online EM algorithm on the JMLS switching between these state-space models, where $k = 1, 2$. The measurement noise variances are $R^1 = 0.002$, $R^2 = 0.005$ and process noise covariances are $Q^1 = \text{Diag}(0.045, 0.03)$, $Q^2 = \text{Diag}(0.032, 0.02)$.

The system is excited with the input $u_t \sim \mathcal{N}(0, 1)$, and the total simulation time is $t = 10,000$. The unknown parameters of the models are initialized with a 20% error as $\hat{F}^i = 1.2F^i$ and $\hat{H}^i = 0.8H^i$. Step sizes used in the simulation are chosen as $\eta_t^1 = t^{-0.7}$ for the sufficient statistics related to the state transition matrix, and $\eta_t^2 = t^{-0.6}$ for those related to the measurement matrix. The burn-in period is set to $t_b = 150$.

The estimated parameters $a_1^k, a_2^k, b_1^k, b_2^k$ for $k = 1, 2$, averaged over 100 MC runs, are given in Figure 1 where the transfer function is parameterized as in (2). Table 1 summarizes the parameter estimates, percent errors, and simulation times. Both EM algorithms yield parameter estimates that are close to the true values. The online EM algorithm processed 10,000 measurements in 38 seconds, whereas the batch method required 150 iterations for parameter convergence, taking approximately $19,000$ seconds to complete the iterations.



Left: First mode

Right: Second mode

Figure 1: Single run parameter estimates of online and offline EM algorithms and average parameter estimates of the online EM algorithm over 100 independent runs for the second order system given in Section 1.1. Left: results for the first mode. Right: results for the second mode.

Table 1: Parameter estimates, absolute percent errors with respect to true values and simulation times of the second order system given in section 1.1

Parameter	b_1^1	b_2^1	a_1^1	a_2^1
True Values	0.741	0.005	0.618	0.439
Online EM Single Run Estimate	0.737	0.007	0.607	0.421
Percent Error	0.5%	40%	1.8%	4.1%
Online EM Average Estimate	0.745	0.004	0.616	0.426
Percent Error	0.6%	20%	0.3%	3.0%
Offline EM Estimate	0.739	0.004	0.617	0.440
Percent Error	0.3%	20%	0.2%	0.2%
Parameter	b_1^2	b_2^2	a_1^2	a_2^2
True Values	-1.461	1.980	-1.181	0.271
Online EM Single Run Estimate	-1.468	2.016	-1.185	0.278
Percent Error	0.5%	1.8%	0.3%	2.6%
Online EM Average Estimate	-1.475	2.014	-1.185	0.277
Percent Error	1.0%	1.7%	0.3%	2.2%
Offline EM Estimate	-1.463	1.977	-1.186	0.273
Percent Error	0.1%	0.2%	0.4%	0.7%
Online EM Computation Time	38 seconds			
Offline EM Computation Time	19000 seconds			

1.2. Fourth Order System with Unknown Noise Characteristics

Consider the nearly constant velocity model with position measurements [4]. The state-space model is

$$F^k = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \otimes I_2, H^k = \begin{bmatrix} I_2 & 0 \end{bmatrix}, \tag{4a}$$

$$Q^k = q \begin{bmatrix} \frac{T^3}{3} & \frac{T^2}{2} \\ \frac{T^2}{2} & T \end{bmatrix} \otimes I_2, \tag{4b}$$

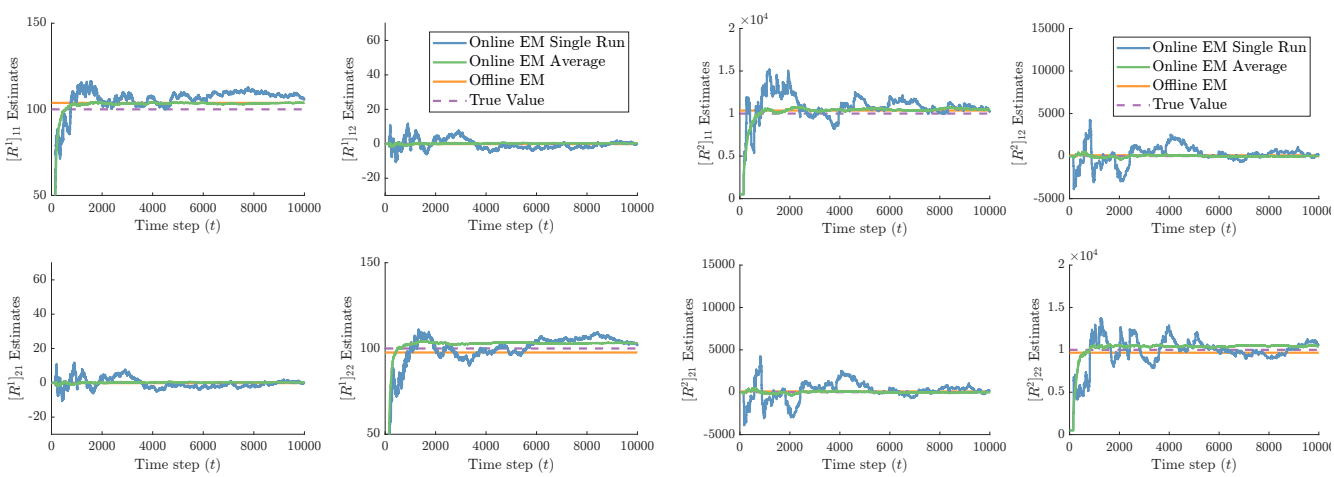
where $\otimes$ is the Kronecker product, $q = 1$, $T = 1s$, and the measurements are obtained either with nominal measurement noise $R^1 = 100I_2$, or outlier measurement noise $R^2 = 100R^1$. The TPM is adapted to

$$\Pi = \begin{bmatrix} 0.9 & 0.1 \\ 0.9 & 0.1 \end{bmatrix} \tag{4c}$$

so that measurements corrupted with Gaussians of severely increased covariances are obtained approximately 10 percent of the time [3].

The initial noise covariance estimates consist of a lower value for the nominal noise covariance and a slightly larger value for the outlier mode's covariance. The initial estimates are chosen as $\hat{R}^1 = 0.2R^1$ and $\hat{R}^2 = 5R^1$. To ensure an adequate exploration of the outlier mode before any maximization step, a burn-in period of $t_b = 150$ is chosen. The step size for the stochastic approximation is set to $\eta_t = t^{-0.8}$.

Figure 2 illustrates the estimated measurement noise covariances. Table 2 presents the final estimates and corresponding percent errors. Both the online EM and the offline EM algorithms have similar accuracy in noise parameter estimates. The online EM algorithm took 41 seconds to process the whole sequence, while the offline EM took 1268 seconds to perform 10 iterations.



Left: First (nominal) mode

Right: Second (outlier) mode

Figure 2: Single run noise covariance matrix estimates of online and offline EM algorithms and average parameter estimates of the online EM algorithm over 100 independent runs for the fourth order system given in Section 1.2. Left: results for the first (nominal) mode. Right: results for the second (outlier) mode.

Table 2: Parameter estimates, absolute percent errors with respect to true values and simulation times of the fourth order system given in section 1.2

Parameter	$[R^1]^1_{11}$	$[R^1]^1_{12}$	$[R^1]^1_{22}$
True Values	100	0	100
Online EM Single Run Estimate	105	0	102
Percent Error	5%	NA	2%
Online EM Average	104	0	103
Percent Error	4%	NA	3%
Offline EM Estimate	104	0	98
Percent Error	4%	NA	2%
Parameter	$[R^2]^1_{11}$	$[R^2]^1_{12}$	$[R^2]^1_{22}$
True Values	10000	0	10000
Online EM Single Run Estimate	10263	270	10640
Percent Error	2.6%	NA	6.4%
Online EM Average Estimate	10448	6	10410
Percent Error	4.5%	NA	4.1%
Offline EM Estimate	10355	96	9662
Percent Error	3.5%	NA	3.4%
Online EM Computation Time	41 seconds		
Offline EM Computation Time	1268 seconds		

1.3. Data Requirements of EM Algorithms on the First-Order System

We have simulated the first-order system from Section 5.1 to compare the data requirements of the online and offline EM algorithms. Figure 3 shows the average parameter estimates over 100 MC runs, with shaded regions denoting the two-sigma bounds. To reach 5% accuracy within the two-sigma bounds, the offline EM requires approximately 2,000–2,500 samples over 200 iterations to converge, while the online EM achieves comparable accuracy in a single pass after processing about 12,000 samples. Despite processing roughly six times more data, the recursive updates of the online variant run in approximately 24 seconds per run, compared to 1,550 seconds for the offline EM.

For the offline EM, although an ensemble average of 100 MC runs using only 200 samples per run falls within the same two-sigma interval, that scheme relies on $50 \times 200 = 10,000$ unique observations and incurs a total processing time of $8,350$ seconds. We therefore use the per-run convergence threshold ($2,000$–$2,500$ samples) as our fair comparison metric.

It is also worth noting that the convergence speed varies across parameters, as reflected in Figure 3. Some parameters (specifically those belonging to the second mode) converge earlier than those belonging to the first, a trend observed consistently across both EM variants.

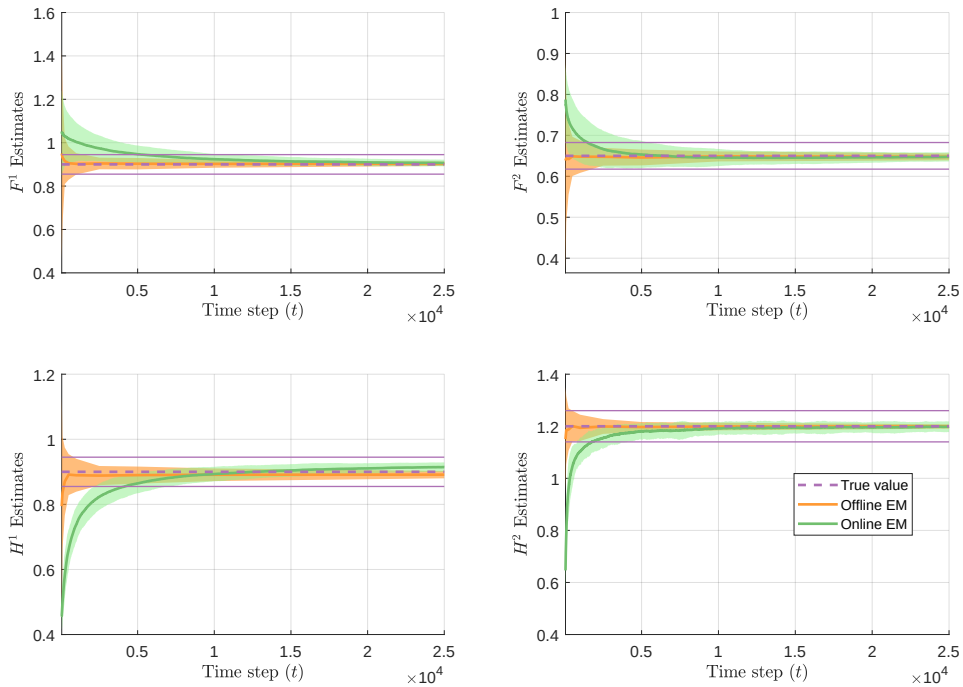


Figure 3: Parameter estimates produced by the online and offline EM algorithms for the unknown transfer function as a function of data-set size. Results independent runs, and the shaded regions denote the two-sigma bounds.

1.4. Data Requirements of EM Algorithms on the Higher-Order Systems

Similar to the previous subsection, we simulate the higher-order system from Section 5.2 to compare the data requirements of the online and offline EM algorithms. We consider systems with $n_x = 5$ and $n_x = 15$. Figures 4 (online EM with $\eta_t = t^{-0.7}$) and 5 (online EM with $\eta_t = t^{-0.8}$) show the estimation results for a representative diagonal entry of the covariance matrices (e.g., $\sqrt{[R^1]_{1,1}}$), the Frobenius norm error $\|\hat{R}^k - R^k\|_F$ of the noise covariance estimates, and selected entries of the estimated transition probability matrix. The top row corresponds to the $n_x = 5$ case, and the bottom row to $n_x = 15$.

To reach 10% relative accuracy in parameter estimates within the two-sigma bounds, the offline EM algorithm requires approximately 1,000 measurements over 10 iterations, whereas the online EM requires around 5,000 measurements in a single pass. Despite this difference in the amount of data needed, the wall times remain comparable: for $n_x = 5$, the offline EM takes approximately 140 seconds and the online EM 12 seconds; for $n_x = 15$, the offline EM takes approximately 290 seconds and the online EM 88 seconds.

We also use this simulation setup to further investigate the slow decrease in the error norm $\|\hat{R}^2 - R^2\|_F$ in the second example. Figures 4 (online EM with $\eta_t = t^{-0.7}$) and 5 (online EM with $\eta_t = t^{-0.8}$) illustrate this behavior. Employing a smaller step-size (i.e., $\eta_t = t^{-0.8}$) significantly reduces the variance of the parameter estimates. However, this reduction in variance comes at the cost of slower convergence in the mean, particularly evident in the nominal noise covariance entry $[\hat{R}^1]_{1,1}$ in Figure 5. As in the first-order case, the convergence behavior remains parameter

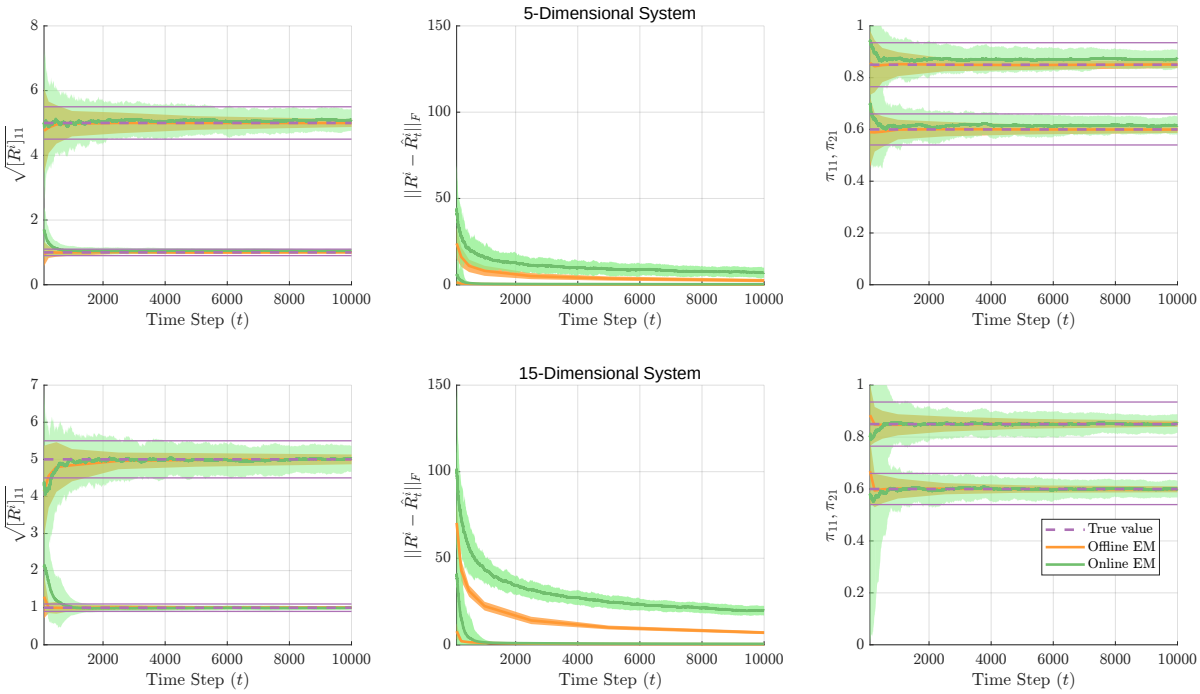


Figure 4: Parameter estimates produced by the online and offline EM algorithms for the unknown covariance matrix and transition probabilities as a function of data-set size. The top row shows the estimates for $n_x = 5$ while the bottom one shows the ones for $n_x = 15$. Results are averaged over 50 independent runs, and the shaded regions denote the two-sigma bounds. The step-size is set to $\eta_t = t^{-0.7}$.

dependent: certain parameters (e.g., $[R^1]_{11}$) converge more slowly than others (such as $[R^2]_{11}$ and the TPM entries). This trend is evident in Figure 4 for both EM methods. It is also important to note that the error norm $\|\hat{R}_t^i - R^i\|_F$ decreases steadily with increasing data size for both online and offline EM.

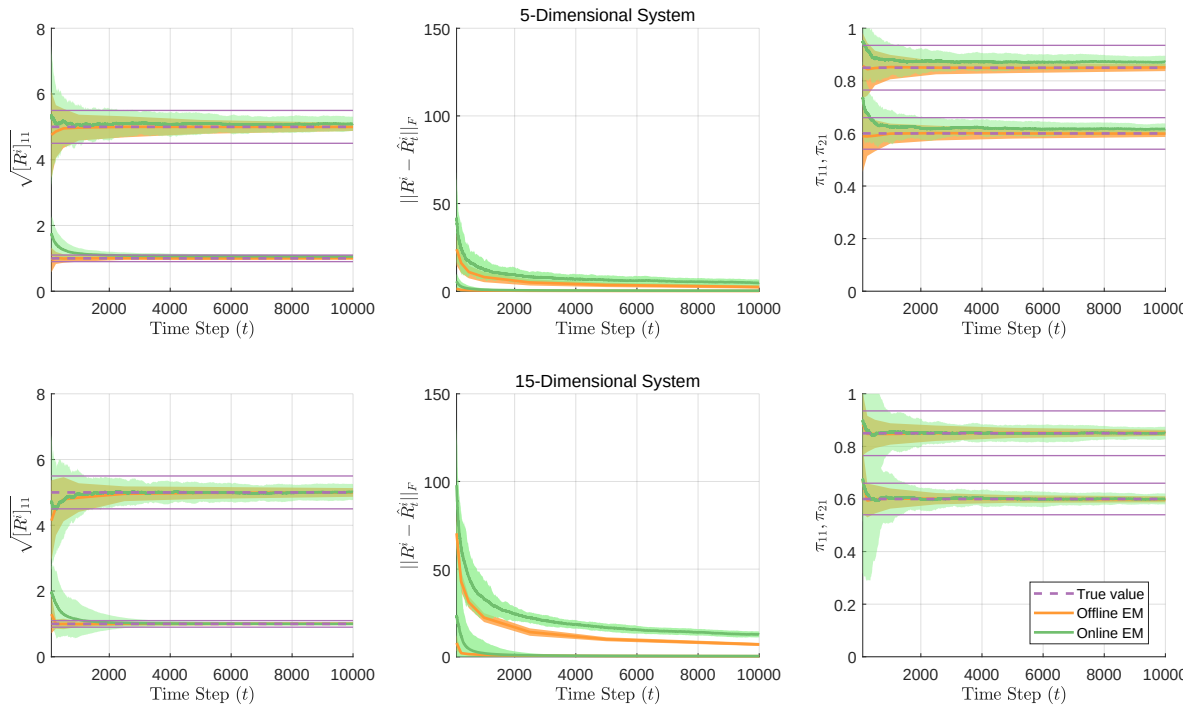


Figure 5: Parameter estimates produced by the online and offline EM algorithms for the unknown covariance matrix and transition probabilities as a function of data-set size. The top row shows the estimates for $n_x = 5$ while the bottom one shows the ones for $n_x = 15$. Results are averaged over 50 independent runs, and the shaded regions denote the two-sigma bounds. The step-size is set to $\eta_t = t^{-0.8}$.

1.5. Impact of Parameter Estimates on State Estimation Accuracy

To assess the reliability of the parameter estimates obtained by the proposed algorithm, we examine their impact on state estimation accuracy. In filtering problems, the root mean square error (RMSE) of the estimated states provides a natural and informative measure of performance. By comparing the RMSE against that obtained using the true parameters, we can directly quantify the influence of parameter estimation errors on filtering performance.

In our analysis, we perform 50 Monte Carlo (MC) simulations and report both a representative single-run and the MC average of the state RMSE. In a representative single run of the online EM algorithm, once the transient phase has passed (i.e., after the estimated parameters satisfy the accuracy criteria reported in Tables 1–3), the state RMSE shows only a slight increase relative to the GBP2 filter operating with true parameters. Specifically, the RMSE increases by approximately 0.34%, 0.22%, and 0.25% for the first-order system and for the higher-order systems with $n_x = 5$ and $n_x = 15$, respectively. The corresponding standard deviations of the MC runs are 0.15, 0.09, and 0.04. These results indicate that the proposed method maintains reasonably accurate state estimation performance despite relying on estimated parameters. The left-hand side of Figure 6 depicts the single-run RMSE of a GPB2 filter operating with the true parameters, together with the RMSE values from a single run of the online EM algorithm. The right-hand side depicts the average of the same entities over 50 MC runs. Figure 7 presents the results of the same experiment repeated for the higher order system with $n_x = 5$ and $n_x = 15$.

The state RMSE from a single run of the online EM algorithm closely aligns with the RMSE obtained using the true parameters, once the parameter estimates fall within the accuracy threshold discussed in the manuscript. A similar pattern is observed in the average over the MC runs.

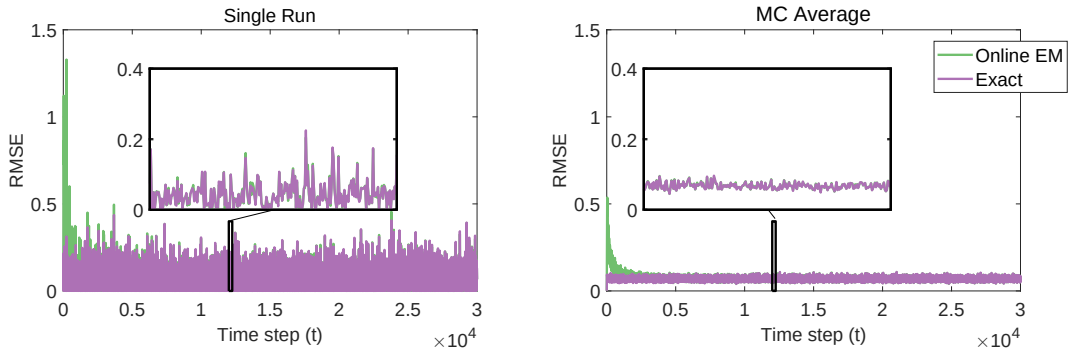


Figure 6: State RMSE of a GPB2 filter run with true parameters (referred to as exact) and run together with online EM. The left plot shows results from a single run, while the right one shows MC average over multiple runs. Note that after $t = 12{,}000$, where the online EM achieves less than 5% error in all parameter estimates, the RMSEs closely match.

The steady-state RMSE values, after the transient phase, are summarized in Table 3 for both a representative single run and the MC averages. Additionally, we report the percentage increase in RMSE for the proposed online EM relative to the GPB2 filters using true parameters, for both the single run and the MC averages (along with MC standard deviations).

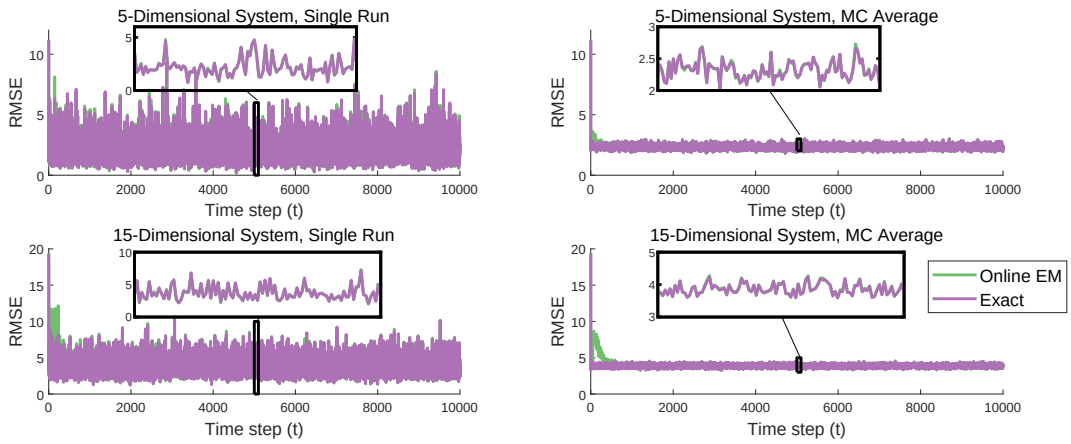


Figure 7: State RMSE of the GPB2 filter run with true parameters (referred to as exact) and run together with online EM. The top row corresponds to the system with $n_x = 5$, while the bottom row shows results for $n_x = 15$. The left column displays a single run, and the right column shows the average over 50 Monte Carlo simulations. Note that after $t = 5{,}000$, when the online EM achieves less than 10% error in all parameter estimates, the RMSEs closely match.

Table 3: RMSE comparison of the GPB2 filters running on the true parameters and the online EM algorithm

System	RMSE of GPB2 Filter Run with True Parameters		RMSE of Online EM		% RMSE Increase		
	Single Run	MC Average	Single Run	MC Average	Single Run	MC Average	MC Std. Dev.
First-Order	0.690	0.689	0.692	0.691	0.34	0.32	0.15
Higher-Order ($n_x = 5$)	2.333	2.339	2.338	2.344	0.22	0.20	0.09
Higher-Order ($n_x = 15$)	3.880	3.867	3.890	3.877	0.25	0.25	0.04

Appendix A. Predictive Density and Merging Weights

Plugging the posterior approximation of GPB2 filter in [1, (22)] into [1, (23)] and using the state-space model in [1, (1)] gives

$$\begin{aligned} p_{\theta'}(x_{t+1}|r_t = l, r_{t+1} = m, y_{0:t}) &= \int p_{\theta'}(x_{t+1}|r_{t+1} = m, x_t) p_{\theta'}(x_t|r_t = l, y_{0:t}) dx_t \\ &= \int \mathcal{N}(x_{t+1}; F^m x_t, Q^m) \mathcal{N}(x_t; x_t^l, Q^l) dx_t, \end{aligned} \tag{A.1}$$

which is a mode-matched prediction update using the mode m. Using the integral of two Gaussian densities [5, Appendix III.B]

$$\mathcal{N}(x; F\bar{x}, F\Sigma F^T + Q) = \int \mathcal{N}(x; Fz, Q)\mathcal{N}(z; \bar{x}, \Sigma) dz \tag{A.2}$$

and shifting the mean by the excitation input $G^m u_t$ yields the numerator of the merging coefficient as

$$p_{\theta'}(x_{t+1}|r_t = l, r_{t+1} = m, y_{0:t}) = \mathcal{N}(x_{t+1}; F^m x_{t|t}^l + G^m u_t; F^m \Sigma_{t|t}^l F^{m,T}). \tag{A.3}$$

The denominator follows similarly from

$$\begin{aligned} p_{\theta'}(x_{t+1}, r_{t+1} = m|y_{0:t}) &= \sum_{j=1}^{K} p(x_{t+1}, r_t = j, r_{t+1} = m|y_{0:t}) \\ &= \sum_{j=1}^{K} p_{\theta'}(x_{t+1}|r_{t+1} = m, r_t = j, y_{0:t}) p_{\theta'}(r_{t+1} = m|r_t = j) p_{\theta'}(r_t = j|y_{0:t}) \\ &= \sum_{j=1}^{K} \mathcal{N}(x_{t+1}; F^m x_{t|t}^j + G^m u_t; F^m \Sigma_{t|t}^j F^{m,T}) \pi_{jm} \alpha_t(j). \end{aligned} \tag{A.4}$$

Appendix B. Mode-Matched Backward Transition Density

When both of the densities in [1, (28)] are Gaussian, the multiplication corresponds to a Kalman filter measurement update treating x_{t+1} as a measurement. To find the parameters, the joint density can be expressed as

$$p\left(\begin{bmatrix} x_t \\ x_{t+1} \end{bmatrix} \middle| r_t = l, r_{t+1} = m, y_{0:t} \right) = \mathcal{N}\left(\begin{bmatrix} x_t \\ x_{t+1} \end{bmatrix}; \mu_{t,t+1}^{l,m}, \Sigma_{t,t+1}^{l,m} \right), \tag{B.1}$$

where

$$\mu_{t,t+1}^{l,m} = \begin{bmatrix} x_{t|t}^l \\ F^m x_{t|t}^l + G^m u_t \end{bmatrix} \tag{B.2}$$

and

$$\Sigma_{t,t+1}^{l,m} = \begin{bmatrix} \Sigma_{t|t}^l & \Sigma_{t|t}^l F^{m,T} \\ F^m \Sigma_{t|t}^l & F^m \Sigma_{t|t}^l F^{m,T} + Q^m \end{bmatrix}. \tag{B.3}$$

Using this joint density with the well-known Gaussian conditioning results [6, Eq. (353)] immediately gives [1, (29)].

Appendix C. Derivation of the Sufficient Statistics

In this section, we derive the sufficient statistics for the state transition and measurement matrices, F^k and H^k. These are obtained by differentiating the expected complete data log-likelihood with respect to the model parameters. The complete data log-likelihood is given by

$$\begin{aligned}
\log p(y_{0:t}, x_{0:t}, r_{0:t}; \theta) &\stackrel{+}{=} \sum_{i=0}^{t} \log \left(p_{r_i}(y_i|x_i;\theta_{r_i})\, p_{r_i}(x_i|x_{i-1};\theta_{r_i}) \right) \\
&\stackrel{+}{=} -0.5 \sum_{i=0}^{t} \sum_{k=1}^{K} \mathbb{1}(r_i = k)(y_i - H^k x_i)^T (R^k)^{-1} (y_i - H^k x_i) \\
&\quad - 0.5 \sum_{i=1}^{t} \sum_{k=1}^{K} \mathbb{1}(r_i = k)(x_i - F^k x_{i-1} - G^k u_{i-1})^T (Q^k)^{-1} (x_i - F^k x_{i-1} - G^k u_{i-1})
\end{aligned} \tag{C.1}$$

where $\stackrel{+}{=}$ denotes equality up to a constant independent of H^k and F^k. By taking the trace of this scalar expression and applying the cyclic property of the trace operator, it can be rewritten as

$$\begin{aligned}
\log p(y_{0:t}, x_{0:t}, r_{0:t}; \theta) \stackrel{+}{=} &-0.5 \sum_{i=0}^{t} \sum_{k=1}^{K} \mathbb{1}(r_i = k) \operatorname{Tr}([-2H^k x_i y_i^T + H^k x_i x_i^T (H^k)^T](R^k)^{-1}) \\
&-0.5 \sum_{i=1}^{t} \sum_{k=1}^{K} \mathbb{1}(r_i = k) \operatorname{Tr}([-2F^k x_{i-1}(x_i - G^k u_{i-1})^T + F^k x_{i-1} x_{i-1}^T (F^k)^T](Q^k)^{-1})
\end{aligned} \tag{C.2}$$

Taking the expectation of the above expression and employing results from [6, Eq.s (100), (118)]

$$\frac{\partial \operatorname{Tr}(XA)}{\partial X} = A^T, \tag{C.3}$$

$$\frac{\partial \operatorname{Tr}(XBX^TC)}{\partial X} = C^T X B^T + CXB = 2CXB \ \textbf{for} \ B = B^T, C = C^T, \tag{C.4}$$

we can then maximize the auxiliary function

$$\begin{aligned}
\frac{\partial Q(\theta, \theta')}{\partial H^k} &= \frac{\partial \mathbb{E}_{\theta'} \{\log p(y_{0:t}, x_{0:t}, r_{0:t}; \theta)\}}{\partial H^k} \\
&= -0.5 (R^k)^{-1} \mathbb{E}_{\theta'} \left\{ \sum_{i=0}^{t} \mathbb{1}(r_i = k)(-2y_i x_i^T + 2H^k x_i x_i^T) \right\} = 0.
\end{aligned} \tag{C.5}$$

Solving this equation yields the maximizer for H^k as

$$\hat{H}^k = \mathbb{E}_{\theta'} \left\{ \sum_{i=0}^{t} \mathbb{1}(r_i = k) y_i x_i^T \right\} \left(\mathbb{E}_{\theta'} \left\{ \sum_{i=0}^{t} \mathbb{1}(r_i = k) x_i x_i^T \right\} \right)^{-1} \tag{C.6}$$

$$= \mathbb{E}_{\theta'} \left\{ S_{k,t}^{4T}(z_{0:t}) \right\} \left(\mathbb{E}_{\theta'} \left\{ S_{k,t}^{0}(z_{0:t}) \right\} \right)^{-1}, \tag{C.7}$$

where the sufficient statistics for H^k are given by

$$S^4_{k,t}\left(z_{0:t}\right) = \sum_{i=0}^{t} \mathbb{1}\left(r_i = k\right) x_i y_i^T, \text{ and} \tag{C.8}$$

$$S^0_{k,t}\left(z_{0:t}\right) = \sum_{i=0}^{t} \mathbb{1}\left(r_i = k\right) x_i x_i^T. \tag{C.9}$$

We now differentiate the auxiliary function with respect to F^k

$$\begin{aligned}\frac{\partial Q(\theta,\theta')}{\partial F^k} &= \frac{\partial \mathbb{E}_{\theta'}\left\{\log p(y_{0:t}, x_{0:t}, r_{0:t};\theta)\right\}}{\partial F^k} \\ &= -0.5(Q^k)^{-1}\mathbb{E}_{\theta'}\left\{\sum_{i=1}^{t} \mathbb{1}(r_i = k)(-2(x_i - G^k u_{i-1})x_{i-1}^T + 2F^k x_{i-1}x_{i-1}^T)\right\} = 0. \end{aligned} \tag{C.10}$$

Solving this equation yields the maximizer for F^k as

$$\hat{F}^k = \mathbb{E}_{\theta'}\left\{\sum_{i=1}^{t} \mathbb{1}(r_i = k)(x_i - G^k u_{i-1})x_{i-1}^T\right\}\left(\mathbb{E}_{\theta'}\left\{\sum_{i=1}^{t} \mathbb{1}(r_i = k)x_{i-1}x_{i-1}^T\right\}\right)^{-1} \tag{C.11}$$

$$= \left(\mathbb{E}_{\theta'}\left\{S^{2T}_{k,t}(z_{0:t})\right\} - \mathbb{E}_{\theta'}\left\{S^{3T}_{k,t}(z_{0:t})\right\}\right)\left(\mathbb{E}_{\theta'}\left\{S^{1}_{k,t}(z_{0:t})\right\}\right)^{-1}, \tag{C.12}$$

where the sufficient statistics for F^k are given by

$$S^1_{k,t}\left(z_{0:t}\right) = \sum_{i=1}^{t} \mathbb{1}\left(r_i = k\right) x_{i-1} x_{i-1}^T, \tag{C.13}$$

$$S^2_{k,t}\left(z_{0:t}\right) = \sum_{i=1}^{t} \mathbb{1}\left(r_i = k\right) x_{i-1} x_i^T, \text{ and} \tag{C.14}$$

$$S^3_{k,t}\left(z_{0:t}\right) = \sum_{i=1}^{t} \mathbb{1}(r_i = k) x_{i-1}(G^k u_{i-1})^T. \tag{C.15}$$

References

[1] M. Çetinkaya, E. Özkan, Recursive parameter estimation for jump markov linear systems, manuscript under revision (2025).

[2] M. P. Balenzuela, A. G. Wills, C. Renton, B. Ninness, Parameter estimation for jump Markov linear systems, Automatica 135 (2022) 109949. doi:https://doi.org/10.1016/j.automatica.2021.109949.

[3] M. Roth, E. Özkan, F. Gustafsson, A student's t filter for heavy tailed process and measurement noise, in: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, 2013, pp. 5770–5774. doi:10.1109/ICASSP.2013.6638770.

[4] Y. Bar-Shalom, X. R. Li, T. Kirubarajan, Estimation with applications to tracking and navigation: theory algorithms and software, John Wiley & Sons, 2004.

[5] U. Orguner, M. Demirekler, Maximum likelihood estimation of transition probabilities of jump Markov linear systems, IEEE Transactions on Signal Processing 56 (10) (2008) 5093–5108.

[6] K. B. Petersen, M. S. Pedersen, et al., The matrix cookbook, Technical University of Denmark 7 (15) (2008) 510.