Clustered Information Filter for Markov Jump Linear Systems

Eduardo F Costa
Univ. São Paulo
efcosta@icmc.usp.br

Benoîte de Saporta
Univ. Montpellier
Benoite.de-Saporta@umontpellier.fr

- Support by Inria associate team CDSS, ANR grant Piece,
- FAPESP, FAPESP/FAPs/INRIA/INS2i-CNRS and CNPq
PROBLEM - BASIC FORMULATION
MARKOV CHAIN

\[ \{\theta(k), k \geq 0\} \quad (1) \]

- Finite state space \( \{1, 2, \ldots, N\} \)
- Initial distribution \( \pi_0 = [\Pr(\theta(0) = 1) \cdots \Pr(\theta(0) = N)] \).
- Partition \( S_1, \ldots, S_{N_C} \) for its state space.
- \( \rho(k) \) marks the partition visited \( \rho(k) = \sum_{m=1}^{N_C} m \times 1_{\{\theta(k) \in S_m\}} \).
DYNAMICAL SYSTEM

\[ x_{k+1} = A_{\theta(k)} x_k + G_{\theta(k)} w_k \]
\[ y_k = L_{\theta(k)} x_k + H_{\theta(k)} w_k, \quad k \geq 0 \]  

- \( x_0 : E[x_0] = \bar{x} \) and \( E[x_0 x'_0] = \Psi \).
- \( w \) noise, independent from \( x_0 \) and the Markov chain \( \theta \). \( E[w_k] = 0 \) and \( E[w_k w'_k] \) is the identity matrix for all \( k \).
- \( G_i H'_i = 0 \) and \( H_i H'_i > 0, 1 \leq i \leq N \).
- \( A_{\theta(k)} = A_i \) whenever \( \theta(k) = i \), \( A_i \) belonging to a given set of matrices, and similarly for the other matrices.
LUENBERGER OBSERVERS (AS USUAL!)

\[ \hat{x}_{k+1} = A_{\theta(k)}\hat{x}_k + M_k(y_k - L_{\theta(k)}\hat{x}_k) \]  

- \( M_k \) is referred to as the filter gain.
- The initial estimate is given by \( \hat{x}_0 = \bar{x} \).
- This produces an estimation error \( \tilde{x} = x - \hat{x} \) satisfying

\[ \tilde{x}_{k+1} = (A_{\theta(k)} - M_k L_{\theta(k)})\tilde{x}_k + (G_{\theta(k)} - M_k H_{\theta(k)})w_k, \]  

where \( w_k \) is a zero-mean Gaussian noise with covariance \( \Psi \).
$y(k), \theta(k)$ available at each time $k$.

- However, we do not take into account $\theta(0), \ldots, \theta(k - 1)$ when calculating the gain $M_k$ (to avoid an excessive number of branches):

$$M_k = h_k(\rho(0), \ldots, \rho(k - 1), \theta(k)),$$

for measurable functions $h_k$. (5)
OBJECTIVE FUNCTION:

\[
\min_{M_0, \ldots, M_s} E\{\|x_s - \hat{x}_s\|^2 | \mathcal{R}_s\}, \quad \text{s.t. (5)}, \quad (6)
\]

where \( \mathcal{R}_s = \{\rho(0), \ldots, \rho(s), y(0), \ldots, y(s), \theta(s)\} \).
MAIN POINT
THE USUAL ERROR COVARIANCE MATRIX OF THE FILTER FORMS A STOCHASTIC PROCESS:

It is easy to check that $X(k) := E\{\tilde{x}(k)\tilde{x}(k)\}$ satisfies:

$$X(k + 1) = (A_{\theta(k)} - M_k L_{\theta(k)})X(k)(A_{\theta(k)} - M_k L_{\theta(k)})'$$
$$+ (G_{\theta(k)} - M_k H_{\theta(k)})(G_{\theta(k)} - M_k H_{\theta(k)})',$$

• Thus $X(k)$ forms a stochastic process, with some nice properties:
THE USUAL ERROR COVARIANCE MATRIX OF THE FILTER FORMS A STOCHASTIC PROCESS:

\[ X(k + 1) = (A_{\theta(k)} - M_k L_{\theta(k)}) X(k) (A_{\theta(k)} - M_k L_{\theta(k)})' + (G_{\theta(k)} - M_k H_{\theta(k)}) (G_{\theta(k)} - M_k H_{\theta(k)}) , \]

- If the gains \( M_t \) are given \( 0 \leq t \leq s \), then \((X(k), \theta(k))\) is a Markov process
- More importantly, given \( X(k) \) and \( \theta(k) \), the variable \( X(k + 1) \) depends on \( M_k \) only
THE USUAL ERROR COVARIANCE MATRIX OF THE FILTER FORMS A STOCHASTIC PROCESS:

\[ X(k + 1) = (A_{\theta(k)} - M_k L_{\theta(k)}) X(k)(A_{\theta(k)} - M_k L_{\theta(k)})' + (G_{\theta(k)} - M_k H_{\theta(k)}) (G_{\theta(k)} - M_k H_{\theta(k)})', \]

- This allows us to “control the flux of information” via \( M_k \).
- For instance, if \( M_k \) depends on the whole realization \( \theta(0), \ldots, \theta(k) \), then \( X(k + 1) \) will do so - in this case yielding the classic Kalman filter (loosing Markovianity of \( (X, \theta) \)).
THE USUAL ERROR COVARIANCE MATRIX OF THE FILTER FORMS A STOCHASTIC PROCESS:

\[ X(k + 1) = (A_{\theta(k)} - M_k L_{\theta(k)}) X(k)(A_{\theta(k)} - M_k L_{\theta(k)})' + (G_{\theta(k)} - M_k H_{\theta(k)}) (G_{\theta(k)} - M_k H_{\theta(k)})', \]

- Now, if \( M_k \) depends on \( \pi_0, \theta(k) \), then \( X(k + 1) \) will do so, leading to the “classic” linear minimum mean square estimator (LMMSE) for Markov jump linear systems [2, 3]. Here, \((X, \theta)\) is Markov.
The usual error covariance matrix of the filter forms a stochastic process:

\[ X(k+1) = (A_{\theta(k)} - M_k L_{\theta(k)})X(k)(A_{\theta(k)} - M_k L_{\theta(k)})' + (G_{\theta(k)} - M_k H_{\theta(k)}) (G_{\theta(k)} - M_k H_{\theta(k)})', \]

- Generalizing both examples above: if \( M_k \) depends on \( \pi_0, \rho(0), \ldots, \rho(k-1), \rho(k) \), then \( X(k+1) \) will do so, leading to estimators lying in the between the Kalman and the LMMSE.
FORMULAS
SOME PROBABILITIES:

\[ p_{\ell_0, \ldots, \ell_{k-1}, i, k} := \Pr(\rho(0) = \ell_0, \ldots, \rho(k - 1) = \ell_{k-1}, \theta(k) = i) \]

can be computed recursively via:

\[ p_{\ell_0, \ldots, \ell_{k-1}, i, k} = \sum_{j \in \tilde{S}} p_{ji} p_{\ell_0, \ldots, \ell_{k-2}, j, k-1} \]

where we denote \( \tilde{S} = \{ j \in S_{\ell_k} : p_{\ell_0, \ldots, \ell_{k-2}, j, k-1} \neq 0 \} \),

The initial condition is

\[ p_{\ell_0, i, 1} = \sum_{j \in S_{\ell_0}} p_{ji} \pi_{0,j} \]
PRE-COMPUTING OPTIMAL “CONDITIONED” ERROR COVARIANCES:

\[ Y_{\ell_0, \ldots, \ell_{k-1}, i, k} := E(\tilde{x}_k \tilde{x}_k' \mathbb{1}_{\{\rho(0) = \ell_0, \ldots, \rho(k-1) = \ell_{k-1}, \theta(k) = i\}}). \]

can be computed recursively via Riccati-like equations:

\[
Y_{\ell_0, \ldots, \ell_{k-1}, i, k} = \begin{cases} 
0, & \text{if } p_{\ell_0, \ldots, \ell_{k-1}, i, k} = 0, \\
\sum_{j \in S} p_{ji} \left[ A_j Y_{\ell_0, \ldots, \ell_{k-2}, j, k-1} A_j' + p_{\ell_0, \ldots, \ell_{k-2}, j, k-1} G_j G_j' \\
- A_j Y_{\ell_0, \ldots, \ell_{k-2}, j, k-1} L_j' (L_j Y_{\ell_0, \ldots, \ell_{k-2}, j, k-1} L_j' \\
+ p_{\ell_0, \ldots, \ell_{k-2}, j, k-1} H_j H_j')^{-1} L_j Y_{\ell_0, \ldots, \ell_{k-2}, j, k-1} A_j' \right], & \text{otherwise.}
\end{cases}
\]

The initial condition is

\[ Y_{i, 0} = \pi_{0, i} \Psi. \]
PRE-COMPUTING OPTIMAL GAINS:

\[
M_k^* = \begin{cases} 
0, & \text{Pr}(\rho(0), \ldots, \rho(k-1), \theta(k)) = 0, \\
A_{\theta(k)} Y_{\rho(0), \ldots, \rho(k-1), \theta(k), kL'} & \text{otherwise,} \\
+ \Pr(\rho(0), \ldots, \rho(k-1), \theta(k)) H_{\theta(k)} H'_{\theta(k)} \cdot & \\
\cdot (L\theta(k) Y_{\rho(0), \ldots, \rho(k-1), \theta(k), kL'})^{-1}, & \\
\end{cases}
\]

Proof: see Theorem 3 in [1].
PRE-COMPUTING OPTIMAL CONDITIONED ERROR COVARIANCES:

\( X_{\ell_0,\ldots,\ell_{k-1},i,k} := E(\tilde{x}_k \tilde{x}_k' | \rho(0) = \ell_0, \ldots, \rho(k-1) = \ell_{k-1}, \theta(k) = i) \)

can be computed recursively via Riccati-like equations:

\[
X_{\ell_0,\ldots,\ell_{k-1},i,k} = \begin{cases} 
\text{arbitrary,} & \text{if } p_{\ell_0,\ldots,\ell_{k-1},i,k} = 0, \\
\sum_{j \in S} \frac{p_{j|i} p_{\ell_0,\ldots,\ell_{k-2},j,k-1} A_j X_{\ell_0,\ldots,\ell_{k-2},j,k-1} A_j' + G_j G_j'}{p_{\ell_0,\ldots,\ell_{k-1},i,k}} \\
- A_j X_{\ell_0,\ldots,\ell_{k-2},j,k-1} L_j' (L_j X_{\ell_0,\ldots,\ell_{k-2},j,k-1} L_j')^{-1} + H_j H_j' & \text{otherwise.}
\end{cases}
\]

(9)
PRE-COMPUTING OPTIMAL GAINS:

\[ M_k^* = \begin{cases} 
0, & \text{Pr}(\rho(0), \ldots, \rho(k-1), \theta(k)) = 0, \\
A_{\theta(k)} X_{\rho(0), \ldots, \rho(k-1), \theta(k), k L'_{\theta(k)}} \\
\quad \cdot \left(L_{\theta(k)} X_{\rho(0), \ldots, \rho(k-1), \theta(k), k L'_{\theta(k)}}ight)^{-1} + H_{\theta(k)} H'_{\theta(k)} & \text{otherwise},
\end{cases} \]

(10)

The above is very similar (in form) to the Kalman gain.
ONLINE COMPUTATION OF OPTIMAL CONDITIONED ERROR COVARIANCES:

The gains and error covariances can be computed during the system operation - we do not need to store all branches of $X_{\ell_0, \ldots, \ell_{k-1}, i, k}$ and respective gains.

Given a realization $\ell_0, \ldots, \ell_{k-1}, \ell_k$ and $\theta(k) = i$, if we denote $X(k, i) = X_{\ell_0, \ldots, \ell_{k-1}, i, k}$ then the formula above yields:

$$X(k, i) = \begin{cases} \text{arbitrary,} & \text{if } p_{\ell_0, \ldots, \ell_{k-1}, i, k} = 0, \\ \sum_{j \in S} \frac{p_{j i} p_{\ell_0, \ldots, \ell_{k-2}, j, k-1}}{p_{\ell_0, \ldots, \ell_{k-1}, i, k}} \left[ A_j X(k-1, j) A_j' + G_j G_j' \\
- A_j X(k-1, j) L_j' (L_j X(k-1, j) L_j' + H_j H_j')^{-1} L_j X(k-1, j) A_j' \right], & \text{otherwise}. \end{cases}$$
ONLINE COMPUTATION OF OPTIMAL CONDITIONED ERROR COVARIANCES:

\[
X(k, i) = \begin{cases} 
\text{arbitrary,} & \text{if } p_{\ell_0, \ldots, \ell_{k-1}, i, k} = 0, \\
\sum_{j \in S} \frac{p_{ji} p_{\ell_0, \ldots, \ell_{k-2}, j, k-1}}{p_{\ell_0, \ldots, \ell_{k-1}, i, k}} 
& \left[ A_j X(k - 1, j) A'_j + G_j G'_j 
- A_j X(k - 1, j) L'_j (L_j X(k - 1, j) L'_j + H_j H'_j)^{-1} L_j X(k - 1, j) A'_j \right], 
\end{cases}
\]

Note that the term inside the sum (where the gains come from) is a standard Riccati. In a sense, we are optimizing in the same way a Kalman filter does, however we take into account a “restricted information” error covariance process \( X(k, i) \).

We use this to prevent an excessive branching of \( X \).
PARTICULAR CASES - KALMAN AND LMMSE
KALMAN FILTER:

Consider each cluster contains a separate $\theta$, e.g. $S_1 = 1, S_2 = 2, \ldots, S_N = N$. Then

$$X_{\rho(0), \ldots, \rho_k-1, \theta(k), k} = X_{\theta(0), \ldots, \theta(k-1), \theta(k), k}$$

is the classic covariance matrix of a Kalman filter.

Getting back to the formula in the preceding slide, the term reduces to 1, moreover $\tilde{S}$ reduces to $\theta(k - 1)$ and we retrieve the classic Riccati of filtering.

$$X(k + 1, \theta(k)) = A_j X(k - 1, \theta(k - 1)) A_j' + G_j G_j'$$

$$- A_j X(k - 1, \theta(k - 1)) L_j' (L_j X(k - 1, \theta(k - 1)) L_j' + H_j H_j')^{-1} \cdot L_j X(k - 1, \theta(k - 1)) A_j'$$
LMMSE:

Consider only one cluster $S_1 = \{1, 2, \ldots, N\}$. Then

$$X_{\rho(0), \ldots, \rho_{k-1}, \theta(k), k} = X_{1, \ldots, 1, \theta(k), k}$$

does not branch at all - they can be stored as a set of matrices for each time $k$.

Note also that $p_{\ell_0, \ldots, \ell_{k-1}, i, k} = \Pr(\theta(k = i))$ and substituting this in (7) one obtains the formulas of the LMMSE given in [2, 3].
Table: Mean square error, CPU time to compute the gains and the number of gains for every cluster configuration.

| Clusters     | $E(||\tilde{\chi}_{10}||^2)$ | CPU time  | n. gains |
|--------------|-------------------------------|-----------|----------|
| \{1,2,3,4\} | 0.6699                        | 2.24 $\cdot 10^{-2}$ | 40       |
| \{1,2\},\{3,4\} | 0.6690                        | 3.32      | 4,092    |
| \{1,3\},\{2,4\} | 0.6680                        | 3.34      | 4,092    |
| \{1,4\},\{2,3\} | 0.6689                        | 3.37      | 4,092    |
| \{1\},\{2,3,4\} | 0.6696                        | 3.34      | 4,092    |
| \{2\},\{1,3,4\} | 0.6685                        | 3.35      | 4,092    |
| \{3\},\{1,2,4\} | 0.6678                        | 3.34      | 4,092    |
| \{4\},\{1,2,3\} | 0.6691                        | 3.33      | 4,092    |
| \{1,2\},\{3\},\{4\} | 0.6675                        | 1.14 $\cdot 10^3$ | 118,096  |
| \{1,3\},\{2\},\{4\} | 0.6675                        | 1.14 $\cdot 10^3$ | 118,096  |
| \{1,4\},\{2\},\{3\} | 0.6672                        | 1.14 $\cdot 10^3$ | 118,096  |
| \{1\},\{2,3\},\{4\} | 0.6687                        | 1.14 $\cdot 10^3$ | 118,096  |
| \{1\},\{2,4\},\{3\} | 0.6674                        | 1.14 $\cdot 10^3$ | 118,096  |
| \{1\},\{2\},\{3,4\} | 0.6682                        | 1.14 $\cdot 10^3$ | 118,096  |
| \{1\},\{2\},\{3\},\{4\} | 0.6618                        | 1.35 $\cdot 10^4$ | 1,398,100 |
MAGNETIC SUSPENSION SYSTEM

Figure: Schematics of the Magnetic levitator.

\[ \dot{z}_t = v_t \]

\[ m\ddot{v}_t = mg - \frac{L_0 i_t^2}{2a(1 + z_t / a)^2} \]

\[ L\dot{i}_t = -Ri_t + u_t \]
MAGNETIC SUSPENSION SYSTEM

- We take the parameters of a real-world maglev system, discretized with sampling period 0.1, linearized at around an operation point; the components of state $x$ are $x = [z \ z' \ i]$ (position, speed, current), leading to the following state space model (irrespective of the jump parameters):

$$
\begin{bmatrix}
3917 & 87.38 & -41.05 \\
175600 & 3917 & -1840 \\
0 & 0 & 4742 \times 10^{-4}
\end{bmatrix}
\begin{bmatrix}
x_k \\
0 \\
0
\end{bmatrix}
+ 
\begin{bmatrix}
0 & 0 & -9.152 \times 10^{-4} & 0 & 0 \\
0 & 0 & -4.106 \times 10^{-2} & 0 & 0 \\
0 & 0 & 2.612 \times 10^{-5} & 0 & 0
\end{bmatrix}
\begin{bmatrix}
w_k
\end{bmatrix}
$$
There are sensors for the position and coil current, with measurement noise. In “Normal” mode $\theta = 1$ we have

$$y_k = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} x_k + \begin{bmatrix} 0 & 0 & 0 & 0.01 & 0 \\ 0 & 0 & 0 & 0 & 0.01 \end{bmatrix} w_k$$
MAGNETIC SUSPENSION SYSTEM

- We assume that there are two possible types of failures - leading to complete loss of observation of the current and increasing the measurement noise;

- We have modeled the failures via a Markov chain, so that when $\theta = 1$ all sensors are operating normally; $\theta = 2$ describes loss of current readings; $\theta = 3$ describes both loss of current reading and higher measurement noise.

- We have considered the following transition probabilities:

$$P = \begin{bmatrix} 0.9 & 0.05 & 0.05 \\ 0.1 & 0.9 & 0 \\ 0.2 & 0 & 0.8 \end{bmatrix}$$
MAGNETIC SUSPENSION SYSTEM

- We start with a small horizon $s = 5$ so that (7) can be computed quickly. We have considered all possible partitions and obtained the following errors.

Table: Mean square error for every cluster configuration.

<table>
<thead>
<tr>
<th>Clusters</th>
<th>$E(|\tilde{x}_5|^2)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>${1,2,3}$</td>
<td>$2.276868 \times 10^3$</td>
</tr>
<tr>
<td>${1,2},{3}$</td>
<td>$2.276866 \times 10^3$</td>
</tr>
<tr>
<td>${1,3},{2}$</td>
<td>$2.276798 \times 10^3$</td>
</tr>
<tr>
<td>${2,3},{1}$</td>
<td>$2.276800 \times 10^3$</td>
</tr>
<tr>
<td>${1},{2},{3}$</td>
<td>$2.276798 \times 10^3$</td>
</tr>
</tbody>
</table>

- This suggests that the configuration $\{1,3\},\{2\}$ provides a good performance (very close to the KF) with a relatively small complexity.
Taking the results with small $s$ as a guideline, we select the cluster configuration $\{2, 3\}, \{1\}$ and simulate the filter for time horizon $s = 100$ (now using the formulas for online computation). Results are as follows, based on Monte Carlo simulation with 1000 repeats.

Table: Estimated mean square error for every cluster configuration, with horizon $s = 100$.

<table>
<thead>
<tr>
<th>Clusters</th>
<th>$E(|\tilde{x}_{100}|^2)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>${1,2,3}$</td>
<td>2930</td>
</tr>
<tr>
<td>${1,2}, {3}$</td>
<td>2450</td>
</tr>
<tr>
<td>${1,3}, {2}$</td>
<td>335</td>
</tr>
<tr>
<td>${2,3}, {1}$</td>
<td>2430</td>
</tr>
<tr>
<td>${1}, {2}, {3}$</td>
<td>178</td>
</tr>
</tbody>
</table>

Again, the configuration $\{1, 3\}, \{2\}$ is appealing for approximating the Kalman filter performance with a smaller complexity.
CONCLUDING REMARKS
• We have explored the stochastic process formed by the error covariance matrix $X(k) = E\{\tilde{x}(k)\tilde{x}(k)\}$ in the context of Markov jump linear systems.

• $X(k)$ is conditionally independent from $X(k - 2), X(k - 3), \ldots$ and past values of $\theta$ given $X(k - 1), \theta(k - 1)$, allowing us to choose what information is relevant to compute it via the gains $M_k$. 
• Using clustered gains in the form
  \[ M_k = h_k(\rho(0), \ldots, \rho(k-1), \theta(k)) \]
  and choosing the clusters, we can “control” the complexity of the filter in terms of number of gains to be pre-computed, and at same time the performance of the filter.

• Taking few clusters makes pre-computation/computation easy, and yields a low accuracy filter for \( x \).

• The task of finding a suitable choice of the clusters might be complex, taking into account that the CPU time and memory requirements are prohibitive for large horizons \( s >> 1 \) and large number of clusters; however, even a small \( s \) might help in this task, as illustrated in the maglev example.
• A direct extension/adaptation of our results allows to make the cluster configuration dependent on time $k$; e.g. one might use the LMMSE during a time interval (by choosing a single cluster) and shift to the Kalman filter later on, by adopting $N$ clusters. This gives extra flexibility to seek for the best computable filter.

• Future work will look into a continuous-time version. We believe that finding detectability-like conditions for keeping the process $X$ average-bounded is an interesting topic. One might also look for other types of indirect observation of $\theta$, possibly further generalizing the filter.
COSTA, E. F., AND DE SAPORTA, B.
Linear minimum mean square filters for Markov jump linear systems.
*IEEE Transactions on Automatic Control* 62, 7 (July 2017), 3567–3572.

COSTA, O. L. V.
Linear minimum mean square error estimation for discrete-time Markovian jump linear systems.

COSTA, O. L. V., AND TUESTA, E. F.
Finite horizon quadratic optimal control and a separation principle for Markovian jump linear systems.

ZHANG, L., AND BOUKAS, E.-K.
Mode-dependent $H_\infty$ filtering for discrete-time Markovian jump linear systems with partly unknown transition probabilities.
THANK YOU.