Contents

Part I

Literature, Theory and Background Material

Part II

Single Model Systems

Part III

Multiple Model Systems

Chapter 9

Inference using Linear Hybrid Models

In this chapter we generalise the Probabilistic Graphical Models of Chapters 6 and 7 as shown in Figure 9.1. We have added the discrete random variables $(s_0, s_1, s_2, ...)$, where each variable has M states, which we will call the switching variables. The goal of adding switching variables is to allow our graphical models to switch (or more precisely, choose based on the observation) between M different dynamical models. For the moment we restrict ourselves to linear transition functions i.e. we use linear state space models. The other variables retain their meaning as before. Models of this form are usually called Switching Kalman Filter Models [43].

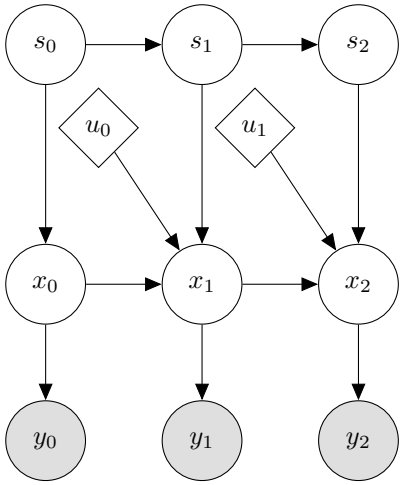

Figure 9.1: Graphical model of this section

One of the benefits of combining discrete switching variables with linear dynamical models is that it allows us to model nonlinear, even multi-modal, processes with linear models. Intuitively, we glue together linear models which each describe a non-linear model in some region and use the switch to determine which one (or combination of) to use. The switch assigns a weight to each model based on its ability to explain the evidence.

We model this system as follows. Let $s_t \in (1, 2, ..., M)$ denote a discrete, time homogeneous M state first order Markov chain with transition matrix P as discussed in Chapter 4. Let each state $s_t = i$ be associated with a parameter set $(A_i, B_i, W_i, C_i, V_i)$ used to evaluate the dynamical model shown in (9.1). If s_t were observed then (9.1) would simplify to a set of Latent Linear Dynamical Systems we could perform inference on using the methods investigated in Chapter 6. However, we assume that s_t is a hidden random variable.

$$\begin{aligned} x_{t+1} &= A_i x_t + B_i u_t + w_t \text{ with } \mathcal{N}(w_t|0, W_i) \\ y_{t+1} &= C_i x_{t+1} + v_{t+1} \text{ with } \mathcal{N}(v_{t+1}|0, V_i) \end{aligned} \tag{9.1}$$

To fully specify the system we also require the prior distributions $p(s_0)$ and $p(x_0|s_0)$ as well as the switch transition matrix P. For the purposes of this dissertation we assumed that the switch transition matrix is available. This matrix can be inferred using the Baum-Welch Algorithm [43] or it can be set using operator expertise as we will show later.

9.1 Exact Filtering

The switching variables $(s_0, s_1, s_2, ...)$ are discrete random variables exactly like the ones seen in Chapter 4. There we derived recursive analytic expressions for inference which were computationally inexpensive. The structure of the stochastic variables $(x_0, x_1, x_2, ...)$ and $(y_0, y_1, y_2, ...)$ are exactly the same as those found Chapter 6. There we derived the famous Kalman Filter equations which were also analytic, recursive and computationally inexpensive. Taking this into consideration, it seems plausible to believe that inference, specifically filtering, for hybrid systems like (9.1) can be formulated in a computationally feasible manner.

Unfortunately, it can be shown that this is not possible in general [35][42] because the memory requirements scale exponentially with time. Loosely speaking one can see this by noting that at the first time step the system is described by a weighted set of M Kalman Filter models (due to the linear assumption and the M switching indices). At time step two the system is described by a weighted set of M^2 Kalman Filter models. Continuing in this manner we see that at time step t the memory requirement is M^t. Clearly this is computationally infeasible and calls for approximate methods to be used.

In literature many approximate filtering algorithms exist and it is not clear which is best. Two of the more popular methods include Gaussian Sum Filtering [2] and Particle Filtering based methods (specifically the Rao-Blackwellisation approach, see [11][20]). Both of these methods take advantage of the Gaussian structure of the system and operate in a fixed memory space making them computationally attractive. We focus on Particle based methods because it can be extended to nonlinear systems with ease.

9.2 Rao-Blackwellised Particle Filter

It is our objective to find the joint posterior distribution $p(s_{0:t}, x_{0:t}|y_{0:t})$. This joint posterior admits filtering of Figure 9.1 if we discard the trajectory and focus only on s_t, x_t. By the chain rule (Definition 3.19) we immediately have that $p(s_{0:t}, x_{0:t}|y_{0:t}) = p(s_{0:t}|y_{0:t})p(x_{0:t}|y_{0:t}, s_{0:t})$. Given $s_{0:t}$ we see that $p(x_{0:t}|y_{0:t}, s_{0:t})$ can be evaluated using the Kalman Filter equations (see Chapter 6) and thus we are only concerned with finding some approximation for $p(s_{0:t}|y_{0:t})$. This is the essence of the Rao-Blackwellised Particle Filter - taking advantage of the conditionally linear Gaussian nature of the system to analytically evaluate a part of the posterior distribution [20].

Using the formulation of the adaptive Sequential Importance Sampling algorithm discussed in Chapter 7 we can apply it to find an approximation of $p(s_{0:t}|y_{0:t})$. We set $\gamma_t(s_{0:t}) = p(s_{0:t}, y_{0:t})$ and $Z_t = p(y_{0:t})$ and then have that $\frac{\gamma_t(s_{0:t})}{Z_t} = p(s_{0:t}|y_{0:t})$ as desired. We then choose our proposal distribution $q_t(s_{0:t}|y_{0:t})$ to be recursive and follow the same procedure as before, shown in (9.2).

$$\begin{aligned} w_t(s_{0:t}) &= \frac{\gamma_t(s_{0:t}, y_{0:t})}{q_t(s_{0:t}|y_{0:t})} \\ &= \frac{p(s_{0:t}, y_{0:t})}{q_t(s_{0:t}|y_{0:t})} \\ &\propto \frac{p(s_{0:t}|y_{0:t})}{q_t(s_{0:t}|y_{0:t})} \\ &\propto \frac{p(y_t|s_t)p(s_t|s_{t-1})}{q_t(s_t|s_{0:t-1}, y_{0:t})} \frac{p(s_{0:t-1}|y_{0:t-1})}{q_t(s_{0:t-1}|y_{0:t-1})} \\ &= \alpha_t(s_{0:t}) w_{t-1}(s_{0:t-1}) \end{aligned} \tag{9.2}$$

As before, we are not interested in the whole trajectory of the switching variable because we only need to perform filtering. Thus our proposal distribution can be chosen to be the prior i.e. $q_t(s_t|s_{0:t-1}, y_{0:t}) = p(s_t|s_{t-1})$. This is suboptimal but easy to sample from [20]. The incremental weight then simplifies to $\alpha_t(s_{0:t}) = p(y_t|s_t)$. We can evaluate this distribution by marginalising out x_t and using the properties of the Gaussian distributions as before (9.3).

$$\begin{aligned} \alpha_t(s_{0:t}) &= p(y_t|s_t) \\ &= \int_{x_t} p(y_t|x_t, s_t)p(x_t|s_{0:t}, y_{0:t-1}) \\ &= p(y_t|y_{0:t-1}, s_{0:t}) \\ &= \mathcal{N}\left(y_t|C_{s_t}A_{s_t}\mu_{t-1}, C_{s_t}\left(A_{s_t}\Sigma_{t-1}A_{s_t}^T + Q_{s_t}\right)C_{s_t} + R_{s_t}\right) \end{aligned} \tag{9.3}$$

Where the subscript s_t denotes the state of the switching variable at time t [43]. Upon inspection we see that (9.3) is just the one step ahead prediction likelihood as discussed in Chapter 6.2 [43]. Note that we will still need to resample the switching state from P periodically to prevent sample impoverishment.

We now have an efficient particle approximation of $p(s_t|y_t)$. To find the filtered posterior distribution as desired we note that $p(s_t, x_t|y_{0:t}) = \sum_i w_t(S_t^i)\delta(S_t^i, s_t)p(x_t|y_{0:t}, S_t^i)$ where S_t^i is

the i$^{\text{th}}$ particle within the framework of Chapter 7.1. Each particle thus consists of a weight, a switch sample and the sufficient statistics generated by the Kalman Filter for a Gaussian i.e. a mean and covariance. The complete algorithm is shown below.

Rao-Blackwellised Particle Filter Algorithm
For $t = 0$:

1. Sample $S_0^i \backsim p(s_0)$ and $\mu_{0|0}^i \backsim p(x_0|s_0)$.
2. Compute the weights $w_0(S_0^i) = p(y_0|S_0^i)$ where y_0 is the observation. Normalise $W_0^i \propto w_0(S_0^i)$.
3. Apply the update step of the Kalman Filter to each particle i and associated parameters to find μ_0^i and Σ_0^i.
4. If the number of effective particles is below some threshold apply resampling with roughening $(W_0^i, S_0^i, \mu_0^i, \Sigma_0^i)$ to obtain N equally weighted particles $(\frac{1}{N}, \bar{S}_0^i, \bar{\mu}_0^i, \bar{\Sigma}_0^i)$ and set $(\bar{W}_0^i, \bar{S}_0^i, \bar{\mu}_0^i, \bar{\Sigma}_0^i) \leftarrow (\frac{1}{N}, \bar{S}_0^i, \bar{\mu}_0^i, \bar{\Sigma}_0^i)$ otherwise set $(\bar{W}_0^i, \bar{S}_0^i, \bar{\mu}_0^i, \bar{\Sigma}_0^i) \leftarrow (W_0^i, S_0^i, \mu_0^i, \Sigma_0^i)$

For $t \geq 1$:

1. Sample $S_t^i \backsim p(S_t^i|\bar{S}_{t-1}^i)$.
2. Compute the weights $\alpha_t(S_t^i) = p(y_t|S_t^i)$ and normalise $W_t^i \propto \bar{W}_{t-1}^i \alpha_t(S_t^i)$.
3. Apply the Kalman Filter algorithm to μ_{t-1} and Σ_{t-1} for each particle i to find the sufficient statistics μ_t and Σ_t using the parameters corresponding to the state of S_t^i.
4. If the number of effective particles is below some threshold apply resampling with roughening $(W_t^i, S_t^i, \mu_t^i, \Sigma_t^i)$ to obtain N equally weighted particles $(\frac{1}{N}, \bar{S}_t^i, \bar{\mu}_t^i, \bar{\Sigma}_t^i)$ and set $(\bar{W}_t^i, \bar{S}_t^i, \bar{\mu}_t^i, \bar{\Sigma}_t^i) \leftarrow (\frac{1}{N}, \bar{S}_t^i, \bar{\mu}_t^i, \bar{\Sigma}_t^i)$ otherwise set $(\bar{W}_t^i, \bar{S}_t^i, \bar{\mu}_t^i, \bar{\Sigma}_t^i) \leftarrow (W_t^i, S_t^i, \mu_t^i, \Sigma_t^i)$

9.3 Rao-Blackwellised Particle Prediction

Like the Particle Predictor studied in the Chapter 7.3, performing prediction using Rao-Blackwellisation is straightforward because there is no weighting (updating the particles based on the observation) step. Each particle's switching state is merely propagated forward using the proposal distribution (the transition matrix P) and the Kalman prediction algorithm is used to evaluate the predicted mean and covariance. For the sake of brevity we do not supply an algorithm because it is a straightforward simplification of the Rao-Blackwellised Particle Filter Algorithm as shown above. The corresponding Probabilistic Graphical Model is shown in Figure 9.2.

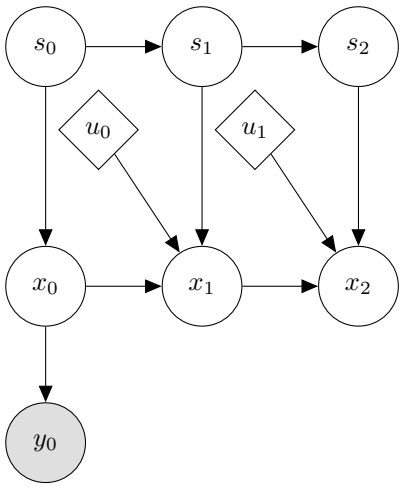

Figure 9.2: Rao-Blackwellised Particle Prediction Graphical Model

9.4 Smoothing and Viterbi Decoding

It is also possible to take advantage of the Gaussian structure in Figure 9.1 to derive a so-called Rao-Blackwellised Smoothing Algorithm. We do not include it here because it is not necessary for the purposes of this dissertation. We refer the reader to the relevant literature [11][20].

Viterbi decoding is likewise not within the scope of this dissertation and as such we refer the reader to [43] for more information. Suffice to say, by increasing the complexity of Figure 9.1 we increase the difficulty of inference in general.

9.5 Filtering the CSTR

We now apply the Rao-Blackwellised Particle Filter (RBPF) to the CSTR introduced in Chapter 5. The focus of this dissertation is on the application of Probabilistic Graphical Models to control, therefore our investigation into the various aspects which improve or degrade filtering performance will be relatively superficial and will target factors which are most relevant only. We will briefly investigate 3 aspects influencing the accuracy of the filter:

1. The effect the switch transition matrix P has on the filter.
2. The effect using more models has on the filter.
3. Using more state measurements.

Like in Chapter 7 we do not investigate the effect increasing the number of particles will have on inference. The same reasons apply and we use the same motivation in selecting the number of particles we use in this section.

It should be noted that in all the succeeding investigations only the most probable particle was used to estimate the current state. The reason for this will become clear in Chapter 10 and 12. It is possible that more accurate state estimates could be reached by using a weighted average of the particles - this approach should be investigated further.

Note that we use the same parameters (e.g. noise covariances etc.) unless otherwise noted as used in Chapter 6. Additionally we assume that the underlying model is the nonlinear CSTR model of Chapter 5.

We begin our investigation by only measuring temperature and using 3 linear models, derived by linearising the non-linear CSTR model at each nominal operating point. Since the CSTR has 3 nominal operating points we have 3 linear models. We compare the use of 2 different switch transition matrices P_1 and P_2 as shown in (9.4). The first index corresponds to the high temperature operating point ($M_1 = (A_1, B_1)$), the second index to the unstable operating point ($M_2 = (A_2, B_2)$) and the third index to the low temperature operating point ($M_3 = (A_3, B_3)$). Note that we assume $C_1 = C_2 = C_3$ because we do not change the way we measure the states between models; additionally we assume that the state and measurement noise is common to all models i.e. $W_1 = W_2 = W_3$ and $V_1 = V_2 = V_3$.

$$P_1 = \begin{pmatrix} 0.50 & 0.25 & 0.25 \\ 0.25 & 0.50 & 0.25 \\ 0.25 & 0.25 & 0.50 \end{pmatrix} \quad P_2 = \begin{pmatrix} 0.99 & 0.01 & 0.00 \\ 0.01 & 0.98 & 0.01 \\ 0.00 & 0.01 & 0.99 \end{pmatrix} \tag{9.4}$$

Intuitively P_1 indicates that we are less sure about the underlying dynamical transitions i.e. we believe it is possible for the system to jump from the dynamics of the low temperature operating point to the dynamics of the high temperature operating point. Conversely, P_2 indicates that we believe it is impossible for the system dynamics to jump from the low temperature operating point to the high temperature operating point without first transitioning through the unstable operating point.

Figure 9.3 shows the state space trajectory of the system we are attempting to perform filtering on. The operating points (points of linearisation) are superimposed on the state space.

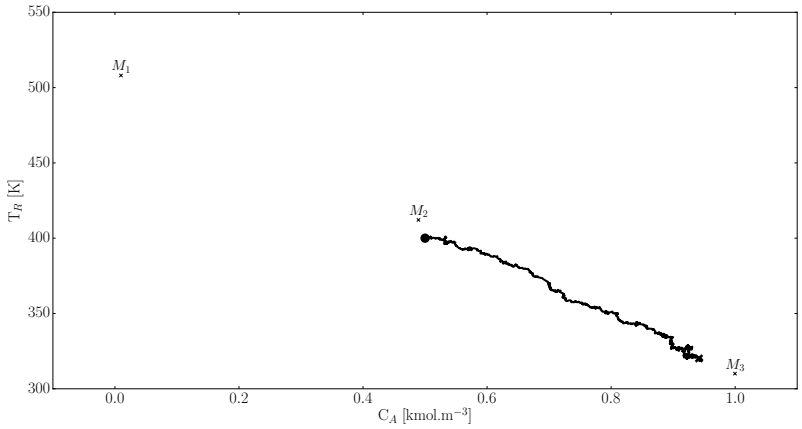

Figure 9.3: State space of the CSTR problem with the position of the 3 linear models superimposed thereupon. The trajectory followed by the system is also shown, the dot is the initial point and the cross the final point.

It is clear from Figure 9.3 that we expect the filter to use M_2 initially and then switch to M_3 as time progresses. Figure 9.4 shows how the RBPF filters the CSTR over a simulation window of 150 minutes. The average concentration error is 21.95% and the average temperature error is 0.42% for the state estimator.

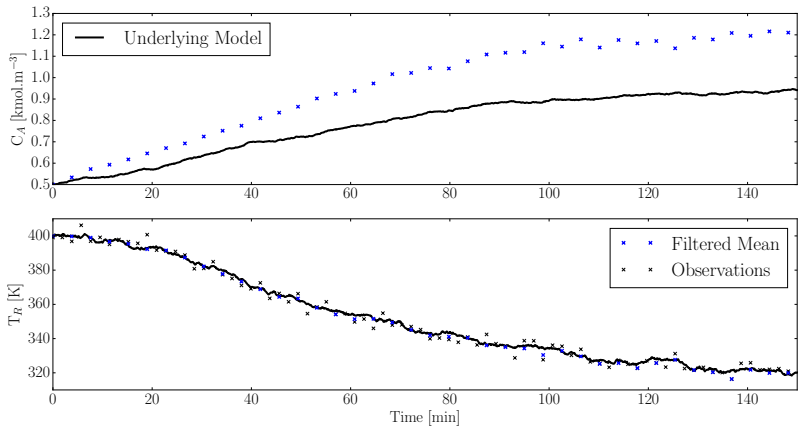

Figure 9.4: Filtering with the RBPF using 3 linear models and 500 particles. Switch transition matrix P_1 was used.

Figure 9.5 shows the state of the corresponding switching variable s_t over time. Since s_t is a discrete random variable we have that at each time slice $\sum_{i=1}^{M=3} s_t^i = 1$.

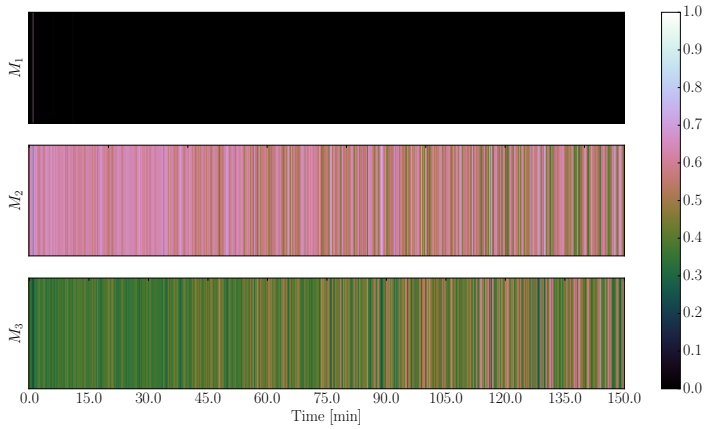

Figure 9.5: State of the switching variable s_t over time. The weight indicates the sum of the particle weights per model.

From Figure 9.4 we see that the filtering error is quite large in the unmeasured state. This has been the trend when performing inference on an unmeasured state, however the magnitude of the error does not justify the use of the more complicated Graphical Model. Additionally, we see that there is no clear switching point in Figure 9.5 - the filter relies on both M_2 an M_3 to estimate the state throughout the simulation. This is contrary to what we expected based on Figure 9.3.

Figure 9.6 shows how the RBPF filters the CSTR over a simulation window of 150 minutes using P_2. The average concentration error is 5.05% and the average temperature error is 0.33% for the state estimator. This is a vast improvement over the case where P_1 was used.

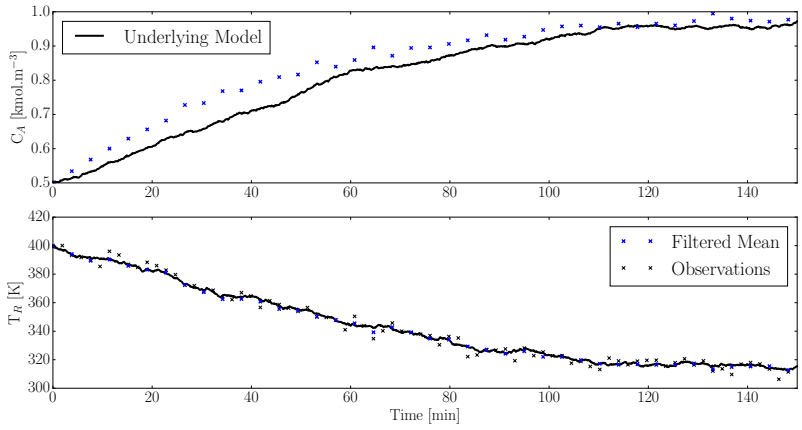

Figure 9.6: Filtering with the RBPF using 3 linear models and 500 particles. Switch transition matrix P_2 was used.

Figure 9.7 shows the state of the corresponding switching variable s_t over time.

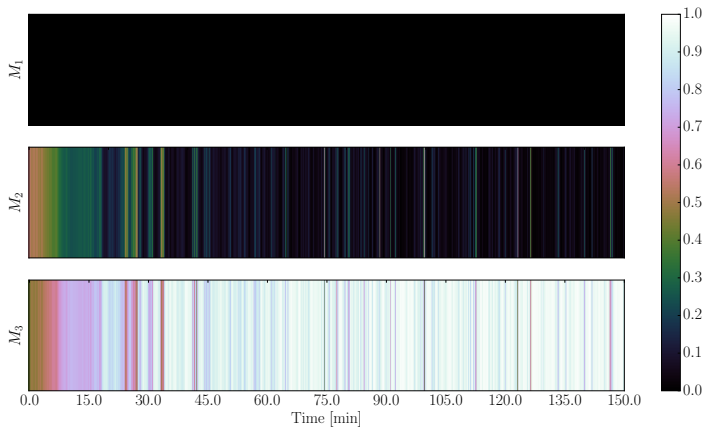

Figure 9.7: State of the switching variable s_t over time. The weight indicates the sum of the particle weights per model.

Unlike Figure 9.5 we do see a clear model transition around the 20 minute mark in Figure 9.7. This is the behaviour we expected - as the system moves away from the unstable operating point the corresponding Graphical Model becomes less important.

These results suggest that the switch transition matrix sets how "sticky" the model transitions are. The more vague they are, as in the case of P_1, the more unsure the filter is about which model is probably generating the observations. On the other hand, in the case of P_2, once the filter switched to the higher probability model it stayed there. The reason for this behaviour is simple: in the case of P_1 the approximate split of particles per model is equal. Since a relatively large number of particles represent M_2 it is reasonable that one of those particle's "guesses" will be near the observation. The filter then assigns a high weight to the inappropriate model and the cycle continues. In the case of P_2, once the model switches the transition matrix ensures that the number of particles representing M_2 is greatly reduced. This then improves filtering performance.

This behaviour is desirable because it is easier to base a control strategy off of one model than multiple models. However, the immediate drawback of the "sticky" approach is that the filter may be over confident or slow to switch models. Additionally if a machine learning approach is not used to infer the values of P it could become a tedious task to set P for a large system. Clearly the values used in P_2 were set by hand - more investigation is necessary to determine proper heuristics if this approach should be adopted in practice.

Next we investigate the effect using more models has on the filter. We use the same 3 model filter as before (using P_2) but compare it to a 7 model filter. The state transition matrix for the 7 model filter is shown in (9.5). We again assume that the state and measurement noise

is common to all models and that they all share the same observation matrix C.

$$P_3 = \begin{pmatrix} 0.98 & 0.00 & 0.00 & 0.00 & 0.00 & 0.01 & 0.00 \\ 0.00 & 0.98 & 0.00 & 0.01 & 0.01 & 0.01 & 0.00 \\ 0.01 & 0.00 & 0.98 & 0.00 & 0.00 & 0.01 & 0.01 \\ 0.00 & 0.00 & 0.00 & 0.98 & 0.00 & 0.01 & 0.00 \\ 0.00 & 0.01 & 0.00 & 0.00 & 0.99 & 0.00 & 0.00 \\ 0.01 & 0.01 & 0.01 & 0.01 & 0.00 & 0.96 & 0.00 \\ 0.00 & 0.00 & 0.01 & 0.00 & 0.00 & 0.00 & 0.99 \end{pmatrix} \tag{9.5}$$

The values of P_3 were set using the same reasoning as before. Figure 9.8 show state trajectory of the system (like Figure 9.3) but with the additional models superimposed thereupon. Clearly M_5, M_6 and M_7 correspond to high temperature, unstable and low temperature operating points.

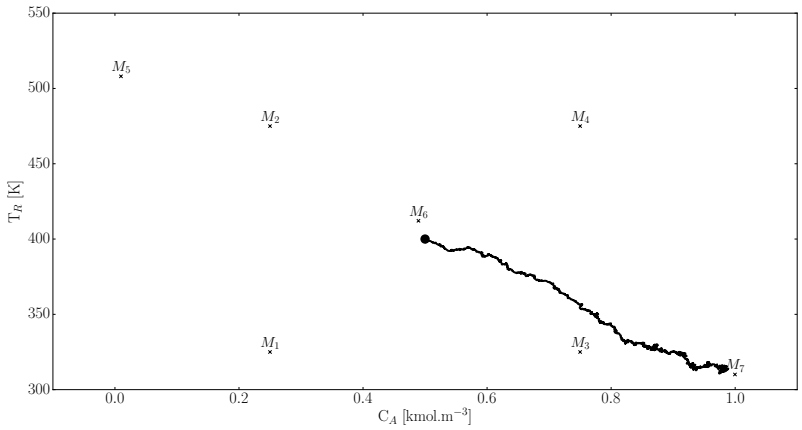

Figure 9.8: State space of the CSTR problem with the position of the 7 linear models superimposed thereupon. The trajectory followed by the system is also shown, the dot is the initial point and the cross the final point.

Figure 9.9 shows the effectiveness of the filter over the simulation window. The average concentration and temperature error is 5.36% and 0.36% respectively.

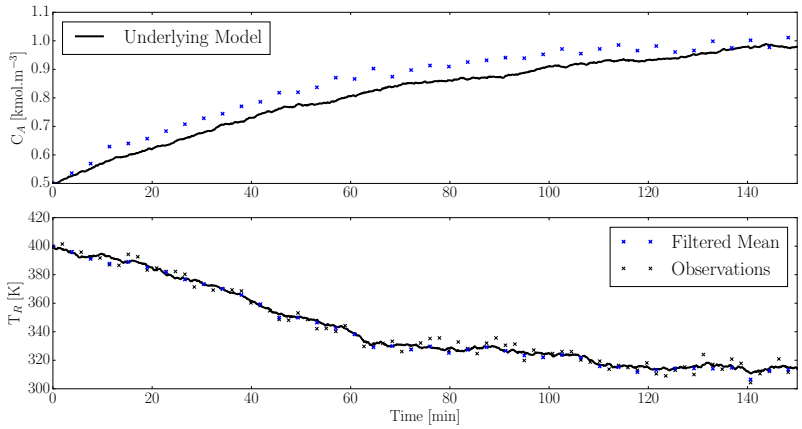

Figure 9.9: Filtering with the RBPF using 7 linear models and 500 particles. Switch transition matrix P_3 was used.

Interestingly enough we actually observe worse tracking performance when more models are used compared to the 3 model case with P_2. We expected the additional models to increase the effectiveness of the filter. Figure 9.10 shows the state of the corresponding switching variable s_t over time. A possible explanation for the performance degradation is evident here.

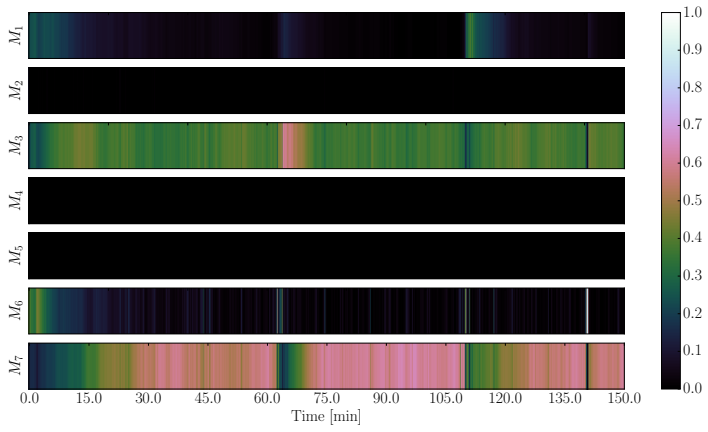

Figure 9.10: State of the switching variable s_t over time. The weight indicates the sum of the particle weights per model.

We certainly expected M_7 to be the dominant model near the end of the simulation; however M_3, while close to the low temperature operating point, played a significant role in the state estimate throughout the simulation. While we only used the most likely model (clearly a particle using M_7) to estimate the current state it is evident that there were a non-negligible number of particles which maintained the M_3 hypothesis. This implies that less particles were available to use the M_7 model and thus we see worse performance.

The crux of the problem is model overlap. While it is clear to a human that the system should only use M_7 near the end of the simulation the algorithm has no way of knowing this. It infers this based on the predictive ability of the models. Clearly M_7, M_3 and to a lesser extent M_1 and M_6 were all able to accurately predict future behaviour. For this reason they have non-negligible weights in Figure 9.10.

We have in fact already come across this problem in Figures 9.4 and 9.5. We saw that it is possible to attenuate this problem by making the switching transition matrix "stickier". Unfortunately this does not solve the underlying problem - the models are not different enough. Using more models would only make this problem worse.

Finally we investigate the effect measuring both states has on the accuracy of the filter. Due to the work in Chapters 6 and 7 we expect that by measuring concentration we will increase the filter accuracy. We use the 3 model filter with P_2 to demonstrate that this is the case.

In Figure 9.11 we see the filtering performance of the RBPF measuring both states. The average concentration and temperature error is 0.88% and 0.31%. This is a significant improvement over the tracking we saw in Figure 9.6.

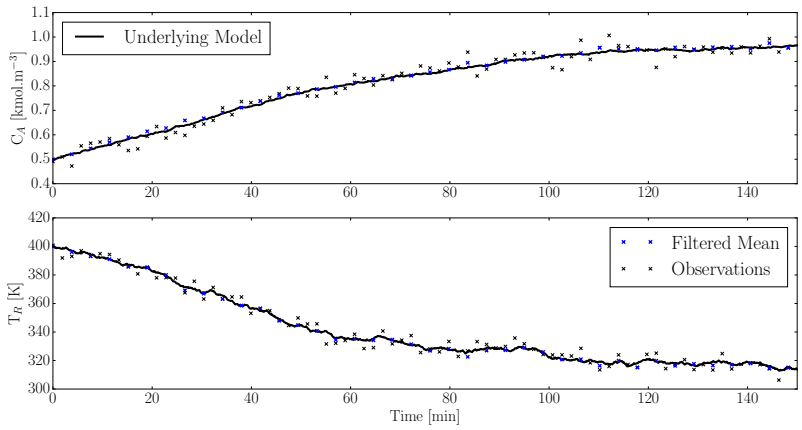

Figure 9.11: Filtering with the RBPF using 3 linear models and 500 particles. Switch transition matrix P_2 was used. Both states are measured.

In Figure 9.12 we see the state of the switching variable over the simulation run. Like Figure 9.7 we also see a clear switch occurring at approximately 15 minutes (actually at 20 minutes when measuring only one state). However, comparing Figures 9.7 and 9.12 closely we see less "switching noise" in the latter. The second measurement allows the filter to compare two state predictions to discern between models. This is clearly beneficial.

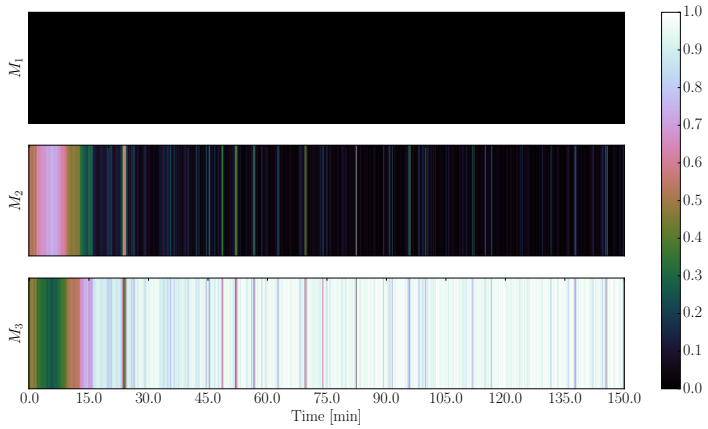

Figure 9.12: State of the switching variable s_t over time. The weight indicates the sum of the particle weights per model.

Finally, in Figure 9.13 we see only the most likely model at each time step. This is simply derived from Figure 9.12 by selecting the particle with the highest weighted switching index.

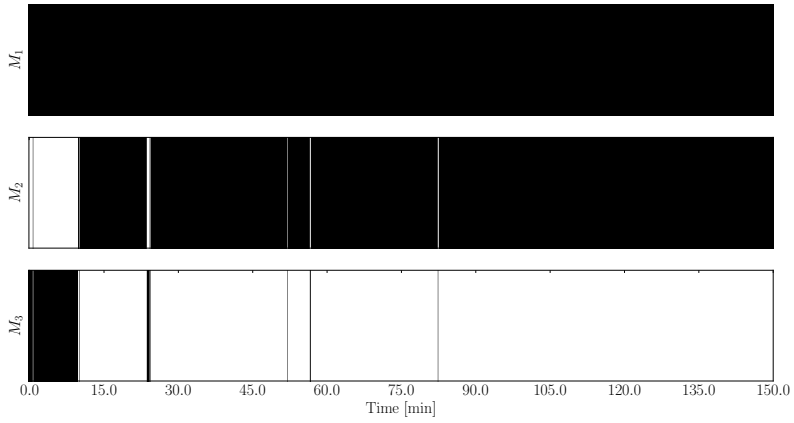

Figure 9.13: State of the switching variable s_t over time. The most likely model at each time step is highlighted in white. Black indicates the model is inactive.

Since we only base our state estimate on the most likely particle Figure 9.13 indicates from which model our state estimate is derived. Ideally one would like very little "switching noise" i.e. as soon as the system moves into territory where M_3 is accurate no particles should have high M_1 or M_2 weight. We see that this is mostly the case in Figure 9.13. In the next chapter we implement control using the most likely model.

Bibliography

[1] Y. Bar-Shalom, X.R. Li, and T. Kirubarajan. *Estimation with applications to tracking and navigation.* John Wiley and Sons, 2001.

[2] D. Barber. Expectation correction for smoothed inference in switching linear dynamical systems. *Journal of Machine Learning*, 7:2515–2540, 2006.

[3] D. Barber. *Bayesian Reasoning and Machine Learning.* Cambridge University Press, 2012.

[4] I. Batina, A.A. Stoorvogel, and S. Weiland. Optimal control of linear, stochastic systems with state and input constraints. In *Proceedings of the 41st IEEE Conference on Decision and Control*, 2002.

[5] A. Bemporad and M. Morari. Control of systems integrating logic, dynamics, and constraints. *Automatics*, 35:407–427, 1999.

[6] C.M. Bishop. *Pattern Recognition and Machine Learning.* Springer, 2006.

[7] L. Blackmore, Hui Li, and B. Williams. A probabilistic approach to optimal robust path planning with obstacles. In *American Control Conference*, June 2006.

[8] L. Blackmore, O. Masahiro, A. Bektassov, and B.C. Williams. A probabilistic particle-control approximation of chance-constrained stochastic predictive control. *IEEE Transactions on Robotics*, 26, 2010.

[9] M. Cannon, B. Kouvaritakis, and X. Wu. Probabilistic constrained mpc for multiplicative and additive stochastic uncertainty. *IEEE Transactions on Automatic Control*, 54(7), 2009.

[10] A.L. Cervantes, O.E. Agamennoni, and J.L Figueroa. A nonlinear model predictive control system based on weiner piecewise linear models. *Journal of Process Control*, 13:655–666, 2003.

[11] R. Chen and J.S. Liu. Mixture kalman filters. *Journal of Royal Statistical Society*, 62(3):493–508, 2000.

[12] J.J. Dabrowski and J.P. de Villiers. A method for classification and context based behavioural modelling of dynamical systems applied to maritime piracy. *Expert Systems with Applications*, 2014.

[13] B.N. Datta. *Numerical Methods for Linear Control Systems - Design and Analysis*. Elsevier, 2004.

[14] M. Davidian. Applied longitudinal data analysis. North Carolina State University, 2005.

[15] R. De Maesschalck, D. Jouan-Rimbaus, and D.L. Massart. Tutorial: The mahalanobis distance. *Chemometrics and Intelligent Laboratory Systems*, 50:1–18, 2000.

[16] J.P. de Villiers, S.J. Godsill, and S.S. Singh. Particle predictive control. *Journal of Statistical Planning and Inference*, 141:1753–1763, 2001.

[17] N. Deo. *Graph Theory with Applications to Engineering and Computer Science*. Prentice-Hall, 1974.

[18] M. Diehl, H.J. Ferreau, and N. Haverbeke. Efficient numerical methods for nonlinear mpc and moving horizon estimation. *Control and Information Sciences*, 384:391–417, 2009.

[19] A. Doucet and A.M. Johansen. A tutorial on particle filtering and smoothing: fifteen years later. Technical report, The Institute of Statistical Mathematics, 2008.

[20] A.D. Doucet, N.J. Gordon, and V. Krishnamurthy. Particle filters for state estimation of jump markov linear systems. *IEEE Transactions on Signal Processing*, 49(3):613–624, March 2001.

[21] J. Du, C. Song, and P. Li. Modeling and control of a continuous stirred tank reactor based on a mixed logical dynamical model. *Chinese Journal of Chemical Engineering*, 15(4):533–538, 2007.

[22] The Economist. In praise of bayes. Article in Magazine, September 2000.

[23] C. Edwards, S.K. Spurgeon, and R.J. Patton. Sliding mode observers for fault detection and isolation. *Automatica*, 36:541–553, 200.

[24] H.C. Edwards and D.E. Penny. *Elementary Differential Equations*. Pearson, 6th edition edition, 2009.

[25] W. Forst and D. Hoffmann. *Optimisation - Theory and Practice*. Springer, 2010.

[26] O.R. Gonzalez and A.G. Kelkar. *Electrical Engineering Handbook*. Academic Press, 2005.

[27] N.J. Gordon, D.J. Salmond, and A.F.M. Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. *IEE Proceedings-F*, 140(2):107–113, 1993.

[28] R. Isermann and P. Balle. Trends in the application of model based fault detection and diagnosis of technical processes. *Control Engineering Practice*, 5(5):709–719, 1997.

[29] K. Ito and K. Xiong. Gaussian filters for nonlinear filtering problems. *IEEE Transactions on Automatic Control*, 45(5):910–928, 2000.

[30] R. J. Jang and C.T. Sun. *Neuro-fuzzy and soft computing: a computational approach to learning and machine intelligence*. Prentice-Hall, 1996.

[31] D. Koller and N. Friedman. *Probabilistic Graphical Models*. MIT Press, 2009.

[32] K. B. Korb and A. E. Nicholson. *Bayesian Artificial Intelligence*. Series in Computer Science and Data Analysis. Chapman & Hall, first edition edition, 2004.

[33] M. Kvasnica, M. Herceg, L. Cirka, and M. Fikar. Model predictive control of a cstr: a hybrid modeling approach. *Chemical Papers*, 64(3):301–309, 2010.

[34] J.H. Lee, M. Morari, and C.E. Garcia. *Model Predictive Control*. Prentice Hall, 2004.

[35] U.N. Lerner. *Hybrid Bayesian Networks for Reasoning about Complex Systems*. PhD thesis, Stanford Univesity, 2002.

[36] P. Li, M. Wendt, H. Arellano-Garcia, and G. Wozny. Optimal operation of distrillation processes under uncertain inflows accumulated in a feed tank. *American Institute of Chemical Engineers*, 2002.

[37] P. Li, M. Wendt, and G. Wozny. A probabilistically constrained model predictive controller. *Automatica*, 38:1171–1176, 2002.

[38] W.L. Luyben. *Process Modeling, Simulation and Control for Chemical Engineers*. McGraw-Hill, 2nd edition edition, 1990.

[39] J.M. Maciejowski. *Predictive Control with constraints*. Prentice-Hall, 2002.

[40] O. Masahiro. Joint chance-constrained model predictive control with probabilistic resolvability. *American Control Conference*, 2012.

[41] P. Mhaskar, N.H. El-Farra, and P.D. Christofides. Stabilization of nonlinear systems with state and control constraints using lyapunov-based predictive control. *Systems and Control Letters*, 55:650–659, 2006.

[42] K.P. Murphy. Switching kalman filters. Technical report, Compaq Cambridge Research Lab, 1998.

[43] K.P. Murphy. *Dynamic Bayesian Networks: Representation, Inference and Learning*. PhD thesis, University of California, Berkeley, 2002.

[44] K.P. Murphy. *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.

[45] N. Nandola and S. Bhartiya. A multiple model approach for predictive control of nonlinear hybrid systems. *Journal of Process Control*, 18(2):131–148, 2008.

[46] L. Ozkan, M. V. Kothare, and C. Georgakis. Model predictive control of nonlinear systems using piecewise linear models. *Computers and Chemical Engineering*, 24:793–799, 2000.

[47] T. Pan, S. Li, and W.J. Cai. Lazy learning based online identification and adaptive pid control: a case study for cstr process. *Industrial Engineering Chemical Research*, 46:472–480, 2007.

[48] J.B. Rawlings and D.Q. Mayne. *Model Predictive Control*. Nob Hill Publishing, 2009.

[49] B. Reiser. Confidence intervals for the mahalanobis distance. *Communications in Statistics: Simulation and Computation*, 30(1):37–45, 2001.

[50] Y. Sakakura, M. Noda, H. Nishitani, Y. Yamashita, M. Yoshida, and S. Matsumoto. Application of a hybrid control approach to highly nonlinear chemical processes. *Computer Aided Chemical Engineering*, 21:1515–1520, 2006.

[51] A.T. Schwarm and Nikolaou. Chance constrained model predictive control. Technical report, University of Houston and Texas A&M University, 1999.

[52] C. Snyder, T. Bengtsson, P. Bickel, and J. Anderson. Obstacles to high-dimensional particle filtering. *Mathematical Advances in Data Assimilation*, 2008.

[53] S.J. Streicher, S.E. Wilken, and C. Sandrock. Eigenvector analysis for the ranking of control loop importance. *Computer Aided Chemical Engineering*, 33:835–840, 2014.

[54] D.H. van Hessem and O.H. Bosgra. Closed-loop stochastic dynamic process optimisation under input and state constraints. In *Proceedings of the American Control Conference*, 2002.

[55] D.H. van Hessem, C.W. Scherer, and O.H. Bosgra. Lmi-based closed-loop economic optimisation of stochastic process operation under state and input constraints. In *Proceedings of the 40th IEEE Conference on Decision and Control*, 2001.

[56] H. Veeraraghavan, P. Schrater, and N. Papanikolopoulos. Switching kalman filter based approach for tracking and event detection at traffic intersections. *Intelligent Control*, 2005.

[57] D. Wang, W. Wang, and P. Shi. Robust fault detection for switched linear systems with state delays. *Systems, Man and Cybernetics*, 39(3):800–805, 2009.

[58] R.S. Wills. Google's pagerank: the math behind the search engine. Technical report, North Carolina State University, 2006.

[59] J. Yan and R.R. Bitmead. Model predictive control and state estimation: a network example. In *15th Triennial World Conference of IFAC*, 2002.

[60] J. Yan and R.R. Bitmead. Incorporating state estimation into model predictive control and its application to network traffic control. *Automatica*, 41:595–604, 2005.

[61] M.B. Yazdi and M.R. Jahed-Motlagh. Stabilization of a cstr with two arbitrarily switching modes using model state feedback linearisation. *Chemical Engineering Journal*, 155(3):838–843, 2009.