Contents

Part I

Literature, theory and background material

Part II

Single model systems

Part III

Multiple model systems

Chapter 12

Stochastic switching linear control using nonlinear hybrid models

We continue our discussion of switching controller algorithms from Chapter 10 here. In this chapter we use the switching particle filter to identify the best model to use in the stochastic controllers we developed in Chapter 8. More precisely, let $M_i = (A_i, B_i)$ be the linearised model corresponding to the nonlinear models (f_i, g_i) as discussed in Chapter 11. By finding the most likely nonlinear model, using the switching particle filter, we aim to design a linear controller (based on the most likely nonlinear model) which is robust against system faults.

The fundamental difference between the controllers we develop in this chapter and those of Chapter 10 is that the linear model used for state prediction is, if the filter behaves as intended, an accurate approximation of the underlying dynamics. In Chapter 10 the controllers performed poorly because they were used to predict the system states into regions where they were not accurate. In this chapter we use the switching particle filter to identify when the underlying dynamics change. The more accurate model is then used for control; however, the crucial difference is that we do not attempt to traverse the state space as in Chapter 10. We rather solve the more modest goal of keeping the system at set point in the presence of system faults.

We assume the same scenario as introduced in Chapter 11.5 i.e. we assume we have 2 nonlinear plant models available. Model M_1 corresponds to the healthy CSTR and model M_2 corresponds to the CSTR with denatured catalyst (the faulty model). We will again avail ourselves of the switching controller algorithm repeated here for convenience.

Switching controller algorithm:

1. Use a switching filter algorithm, e.g. the switching particle filter, to update the state estimates of the particle population given the current observation. See Chapter 11 for more details.

2. Select the particle with the highest switching weight. Since each particle corresponds

to a certain model we implicitly have the most probable model M_i.

3. Use the mean and covariance information encoded by this particle within the context of the stochastic controller (LQG or MPC) formulation of Chapter 8. Use the most likely model, M_i from step 2, in this setting.

4. Repeat for the next observation.

A coincidental benefit of this approach is that the filter/controller combination will automatically detect the modelled fault. We do not consider the model averaging approach (in finding the best linear model to use for control) because it will not make physical sense: the plant is either healthy or broken but cannot be a mixture between the two.

Since the underlying graphical model in Chapter 9 and Chapter 11 is the same, we expect the filtering trends to be the same as those found in Chapter 9. For the sake of illustration we exclusively use both state measurements. There is no fundamental reason why one cannot use only one state measurement except that the filter performance will be worse.

For the remainder of this chapter we assume the control goal is to keep the the system at the unsteady concentration operating point of the healthy model, even in the presence of the denatured catalyst. In all the simulations the catalyst denatures at 100 minutes. This allows us to demonstrate that the switching controller is able to regulate both the healthy and faulty plant.

12.1 Unconstrained switching control

In this chapter we compare the LQG controller (discussed in Chapter 3.4.3 and 8.1) to the switching controller algorithm implemented within the context of the LQG controller as shown in (12.1). For the non-switching LQG controller we use a particle filter to estimate the current state. The same control parameters as those found in Chapter 8.4 are used. Note that the current state estimate x_0 is inferred from the respective observers.

$$\begin{aligned}
&\min_{\mathbf{u}} \mathbb{E}\left[\frac{1}{2}\sum_{k=0}^{N-1}\left(x_k^T Q x_k + u_k^T R u_k\right) + \frac{1}{2}x_N^T P_f x_N\right] \\
&\text{subject to } x_{t+1} = A_i x_t + B_i u_t + w_t
\end{aligned} \tag{12.1}$$

Note that we select the most likely model, $M_i = (A_i, B_i)$, based on the switch weight supplied by the switching particle filter at each time step. This model is then used in (12.1) and solved using the techniques of Chapter 8.1.

In Figure 12.1 we see the performance of the LQG controller applied to the CSTR system. At 100 minutes the catalyst denatures and the model used to design the controller becomes grossly inaccurate. The inappropriateness of the model also affects the particle filter's performance.

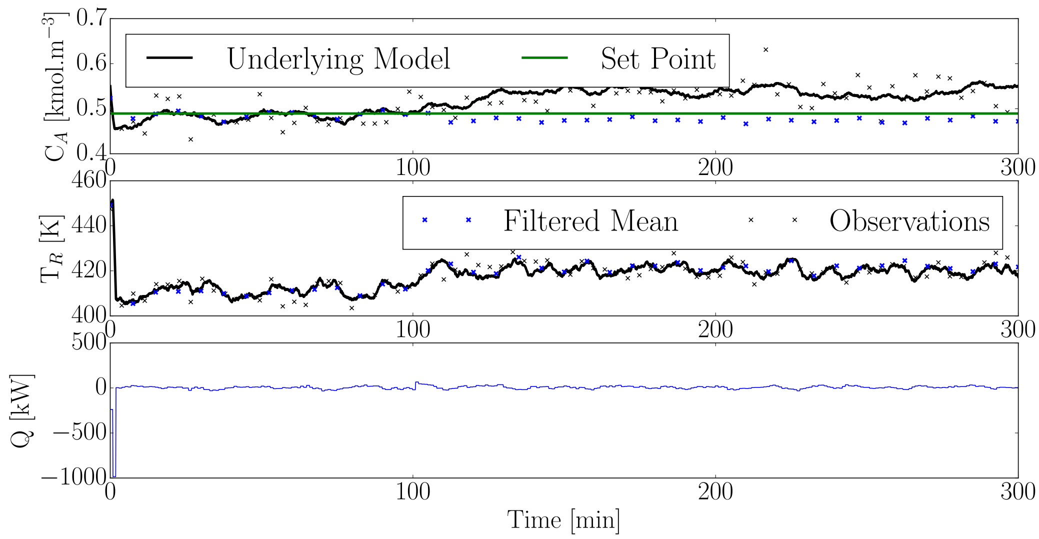

Figure 12.1: Standard LQG controller applied to the CSTR where the catalyst denatures at 100 minutes. The bootstrap particle filter was used for inference and the Gaussian approximation of the particles was used.

The average concentration error is 11.83% and the average controller input is 17.08 kW over the course of the simulation. We can clearly see that there is non-zero set point offset and control is bad in the sense of Definition 8.1. Clearly the LQG controller is ineffective in this scenario.

We used a constant disturbance model to infer the plant/model mismatch[1]. This was used to accordingly adjust the controller predictions as discussed in Chapter 2.2. It is interesting to note that the state estimator infers that the plant reaches set point but in reality there is non-zero offset. This is a consequence of using an inappropriate model in the controller.

This motivates the use of a controller which intelligently changes the model control is based upon, as discussed previously. In Figure 12.2 we see the set point tracking ability of the switching controller algorithm using the LQG controller. We have used the switching transition matrix $P_1 = \begin{pmatrix} 0.99 & 0.01 \\ 0.01 & 0.99 \end{pmatrix}$ as in Chapter 11.5.

[1]The results of this chapter implement the constant disturbance model to achieve zero set point offset. To keep notation the same we do not explicitly show it in (12.2) but mention it here.

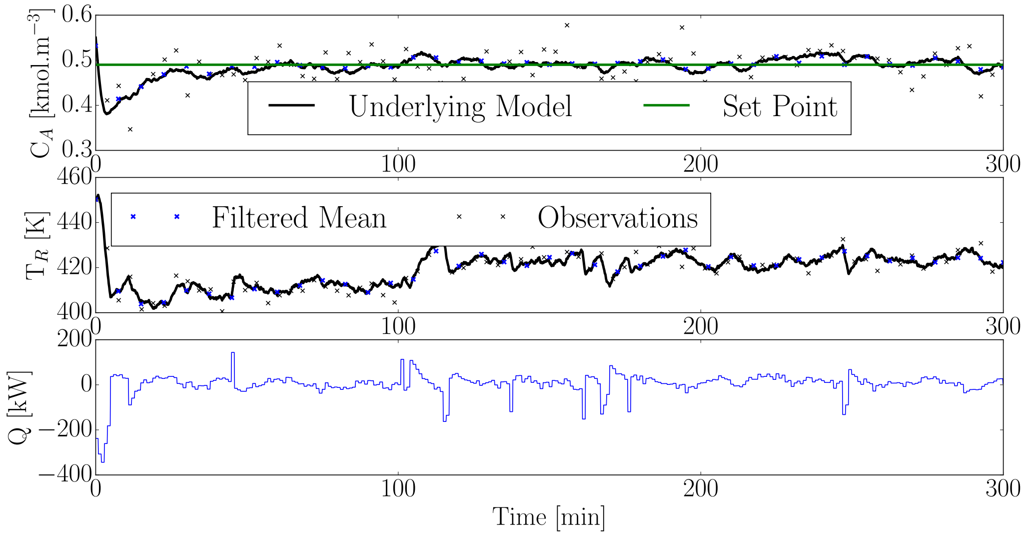

Figure 12.2: Switching LQG controller applied to the CSTR where the catalyst denatures at 100 minutes.

The average concentration error is 2.77% and the average controller input is 28.07 kW. It is clear that we have set point tracking even after the catalyst denatures. By inspecting Figure 12.3 we see that this is not surprising: the filter correctly (for the most part) identifies when the underlying model changes and then uses the better model for control.

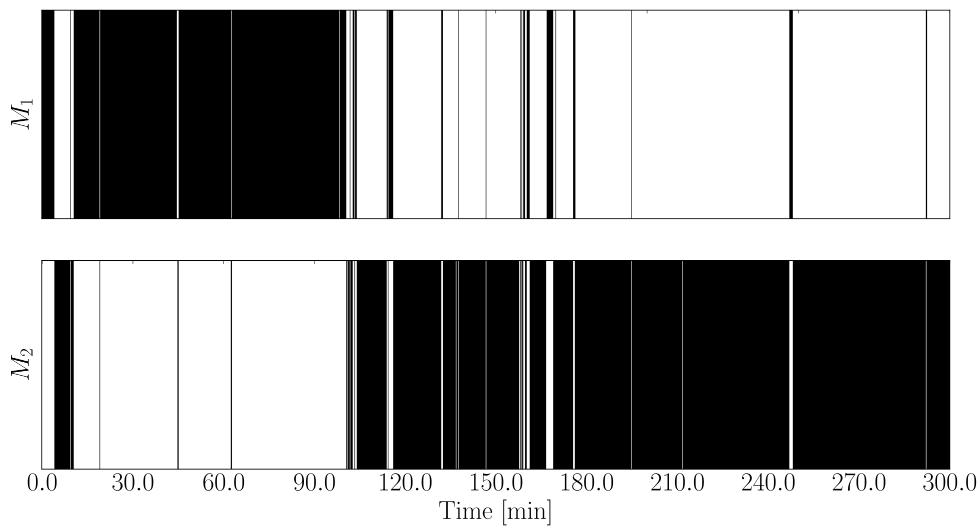

Figure 12.3: Most likely model identified using the particle filter within the context of the switching LQG controller algorithm.

However, like in Chapter 10.1.1 and 11.5 we see that there is some switching noise. The consequence of this noise is spikes in controller input. This happens because the controller uses the incorrect model to calculate the controller input. In Chapter 9 this problem was attenuated by making the switch transition matrix stickier.

In Figures 12.4 and 12.5 we use exactly the same algorithm except that we have modified the switch transition matrix: $P_2 = \begin{pmatrix} 0.999 & 0.001 \\ 0.001 & 0.999 \end{pmatrix}$. Based on Figure 12.4 it is clear that we have set point tracking even after the catalyst denatures.

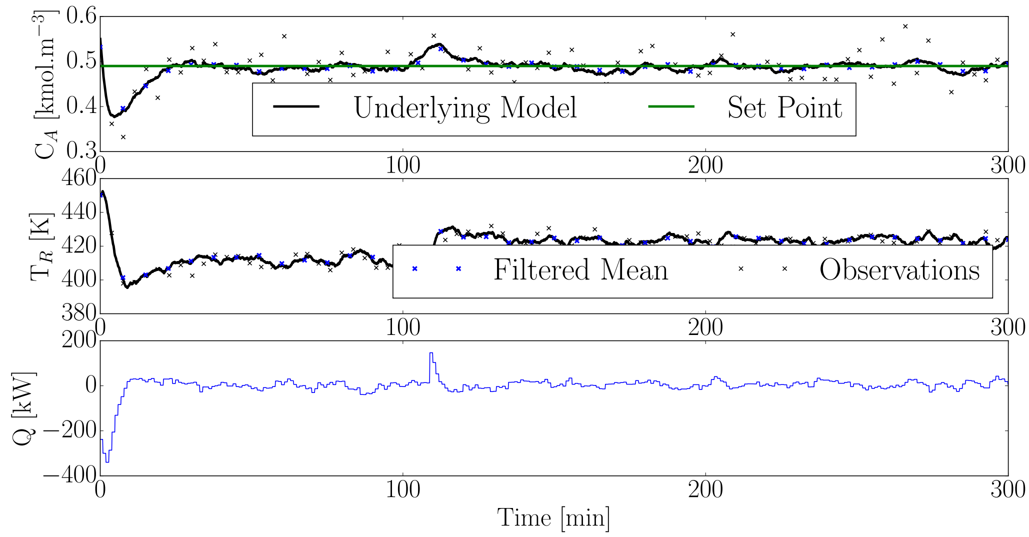

Figure 12.4: Switching LQG controller applied to the CSTR where the catalyst denatures at 100 minutes. Switch transition matrix P_2 was used.

The average concentration error is 2.38% and the average controller input is 19.06 kW. It is not surprising that the controller, using P_2 outperformed the controller using P_1. By inspecting Figure 12.5 we see that there is significantly less switcing noise.

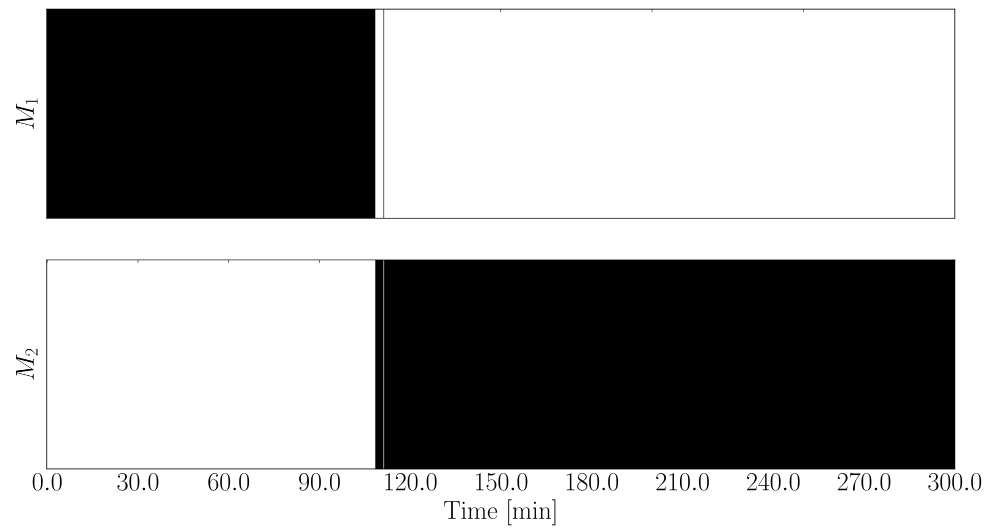

Figure 12.5: Most likely model identified using the particle filter within the context of the switching LQG controller algorithm. Switch transition matrix P_2 was used.

Since there is less switching noise there are less controller spikes in Figure 12.4 and thus less controller energy is wasted. Finally, a Monte Carlo simulation was performed (using 100 runs) to illustrate that the switching controller works as desired for not just the realisations shown in this chapter. The Monte Carlo average concentration error from set point was 2.58%. This indicates that the controller works as desired.

The results here lend further credibility to our claim that the instability seen in Chapter 10 is rooted in the inappropriateness of the model used for prediction rather than the switching noise. Figure 12.3 had significant switching noise yet the controller was not unstable. While reducing the amount of switching noise certainly improved control, as Figure 12.4 shows, fundamentally we are not using an inappropriate model for control prediction. This difference is was causes the significantly better stability properties seen here.

Motivated by the success of the switching LQG controller we incorporate the constraints in the sequel.

12.2 Constrained switching control

In this chapter we extend the switching controller algorithm of Chapter 12.1 to the deterministic and stochastic MPCs introduced in Chapter 8.2. We use the same parameters as before.

Like in Chapter 8.4 we first illustrate the performance of the stochastic MPC controller with expected value constraints shown in (12.2) (for some model M_i) and then incorporate chance constraints later.

$$
\begin{aligned}
&\min_{\mathbf{u}} \mathbb{E}\left[\frac{1}{2}\sum_{k=0}^{N-1}\left(x_k^T Q x_k + u_k^T R u_k\right) + \frac{1}{2}x_N^T P_f x_N\right] \\
&\text{subject to } x_{t+1} = A_i x_t + B_i u_t + w_t \\
&\text{and } \mathbb{E}[\begin{pmatrix}10\\1\end{pmatrix}^T x_t + 400] \geq 0 \ \forall\ t = 1, ..., N \\
&\text{and } |u_t| \leq 250 \ \forall\ t = 0, ..., N-1
\end{aligned}
\tag{12.2}
$$

Using the results of Chapter 8.2 we know that (12.2) can be reformulated as a deterministic problem given the (Gaussian) current state estimate x_0. The state estimate is derived from either the particle filter or switching particle filter using 200 and 500 particles respectively. For the switching particle filter we use the switch transition matrix P_2 due to the results in Chapter 12.1.

In Figure 12.6 we see the set point tracking performance of the (12.2) using the same particle filter as used in Chapter 12.1. Since the particle filter only uses the healthy plant model we only use M_1.

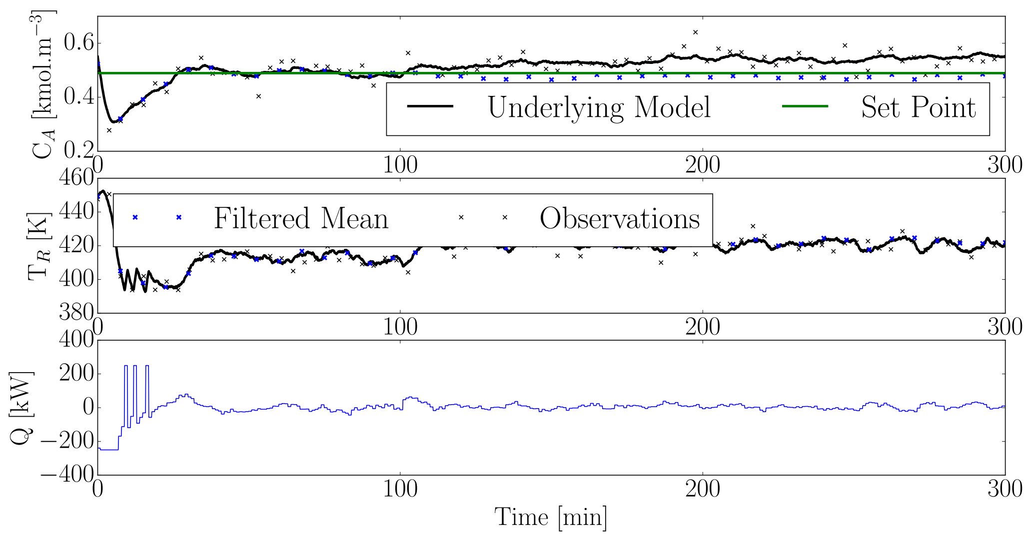

Figure 12.6: Deterministic MPC using a particle filter as the state estimator. The initial point is $(0.55, 450)$. An integrating disturbance model was used to estimate the mismatch between the underlying system and the controller model. The catalyst denatures at 100 minutes.

The average concentration error is 8.33% and the average controller input is 24.21 kW. Unfortunately we do not observe zero set point offset control but rather zero offset state estimates. Clearly the controller input generated by the MPC, which is based on the healthy plant, only drives the particle filter's predictions to the set point. We can see that the classic disturbance model approach [34] to ensure zero set point offset fails here because the underlying (faulty) model is too different from the controller model. Intuitively, we are attempting to control a tricycle (the faulty plant) using a model of a Ferrari.

In Figure 12.7 we see the switching controller algorithm applied within the context of (12.2). The model corresponding to the highest weighted switch at each time step was selected for control.

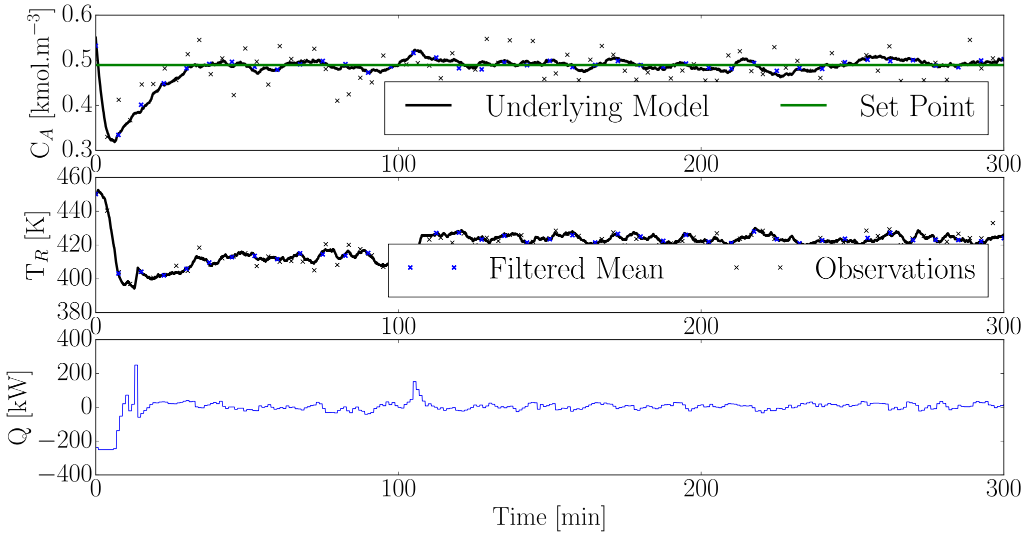

Figure 12.7: The switching MPC controller algorithm applied to the CSTR with catalyst which denatures at 100 minutes.

The average concentration error is 3.19% and the average controller input is 22.61 kW over the simulation time span. The performance of the switching controller is significantly better than the non-switching case. This is not surprising because, as Figure 12.8 shows, the filter correctly identifies when the plant breaks.

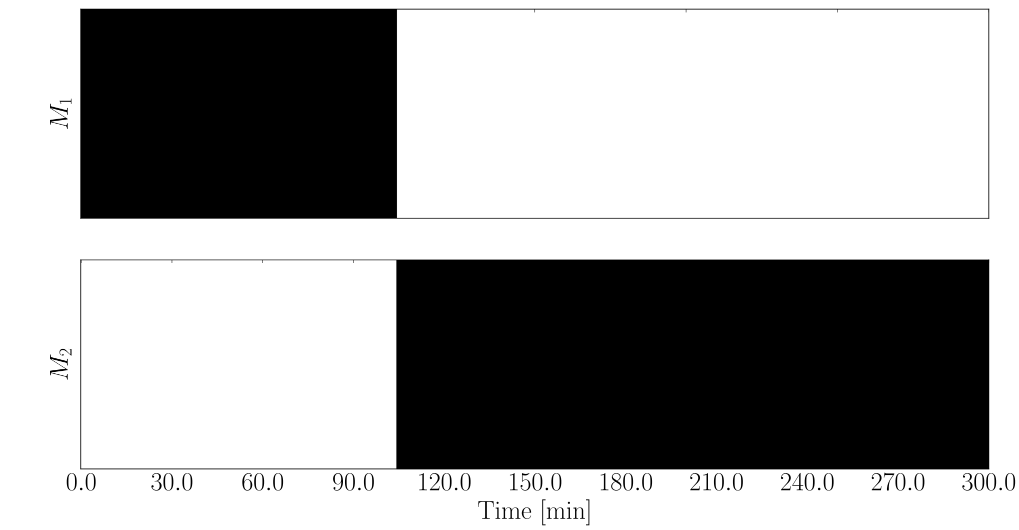

Figure 12.8: Most likely model identified using the particle filter within the context of the switching MPC controller algorithm.

There is almost no switching noise in Figure 12.3 due to the static nature of the switch transition matrix P_2. If P_1 were used we would expect more noise. As mentioned in Chapter 9 switching noise is not necessarily bad for inference. We do see that the filter does not immediately notice that the underlying model has changed. The stickier the switch transition matrix is the longer it will take for the controller to adjust the model. Depending on the application this delay could be problematic. In Figure 12.9 we see the state space trajectory of the expected value constraint MPC.

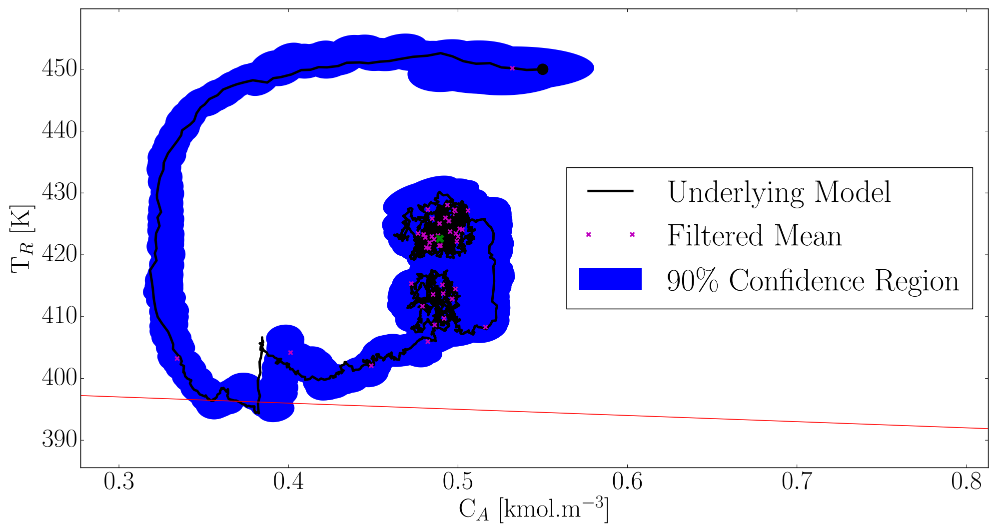

Figure 12.9: State space trajectory of the expected value constrained stochastic MPC using the switching controller algorithm.

Clearly there is a constraint violation - similar to that found in Chapter 8.4 and 8.5. By extending the MPC problem of (12.2) to the chance constrained (12.3) we attempt to ensure that the constraint is not violated in this way.

$$\begin{aligned}
&\min_{\mathbf{u}} \mathbb{E}\left[\frac{1}{2}\sum_{k=0}^{N-1}\left(x_k^T Q x_k + u_k^T R u_k\right) + \frac{1}{2}x_N^T P_f x_N\right] \\
&\text{subject to } x_{t+1} = A_i x_t + B_i u_t + w_t \\
&\text{and } \mathbb{E}[\begin{pmatrix}10\\1\end{pmatrix}^T x_t + 400] \geq 0 \ \forall\ t = 1, ..., N \\
&\text{and } \Pr(\begin{pmatrix}10\\1\end{pmatrix}^T x_t + 400 \geq 0) \geq 0.99 \ \forall\ t = 1, ..., N \\
&\text{and } |u_t| \leq 250 \ \forall\ t = 0, ..., N-1
\end{aligned} \tag{12.3}$$

The same switching controller algorithm, as discussed previously, is implemented. We have used the 99% chance constraint to highlight the effectiveness of the method compared to the expected value version.

In Figure 12.10 we see that the switching chance constrained MPC successfully tracks the set point.

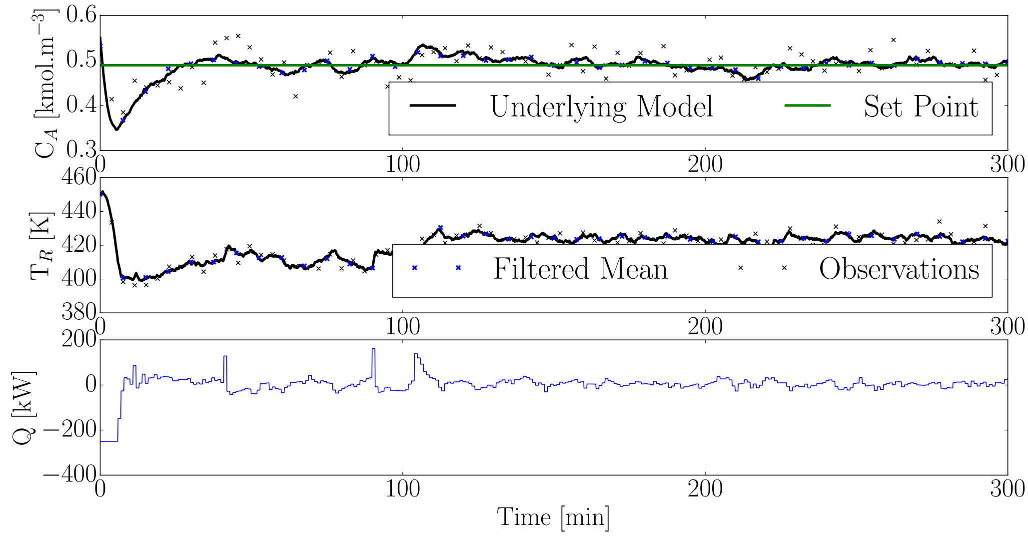

Figure 12.10: The switching MPC controller algorithm applied to the CSTR with catalyst which denatures at 100 minutes. The chance constrained MPC was used.

The average concentration error is 3.01% and the average controller input is 22.03 kW. In Figure 12.11 we see the familiar model switching diagram. Clearly the controller successfully isolates when the fault occurs.

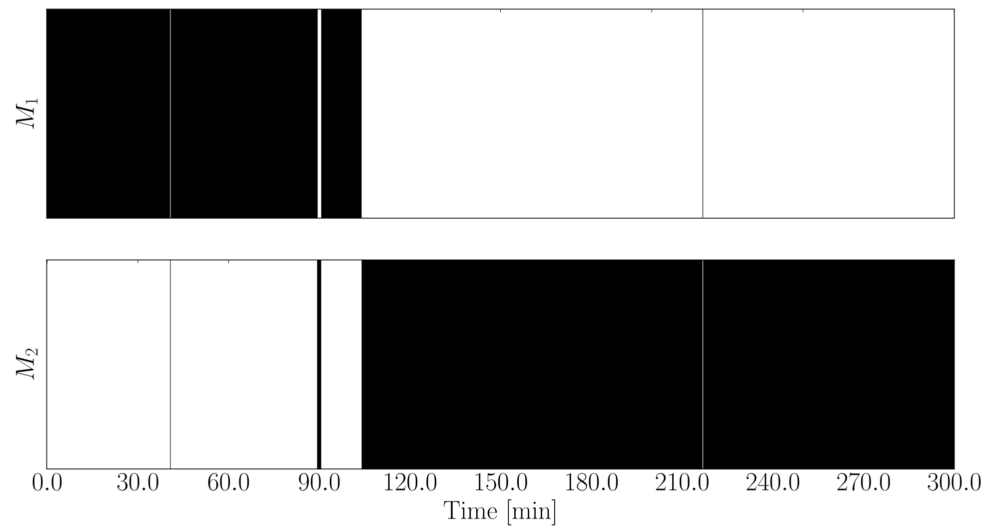

Figure 12.11: Most likely model identified using the particle filter within the context of the chance constrained switching MPC controller algorithm.

Finally, in Figure 12.12 we see that the constraint is not violated.

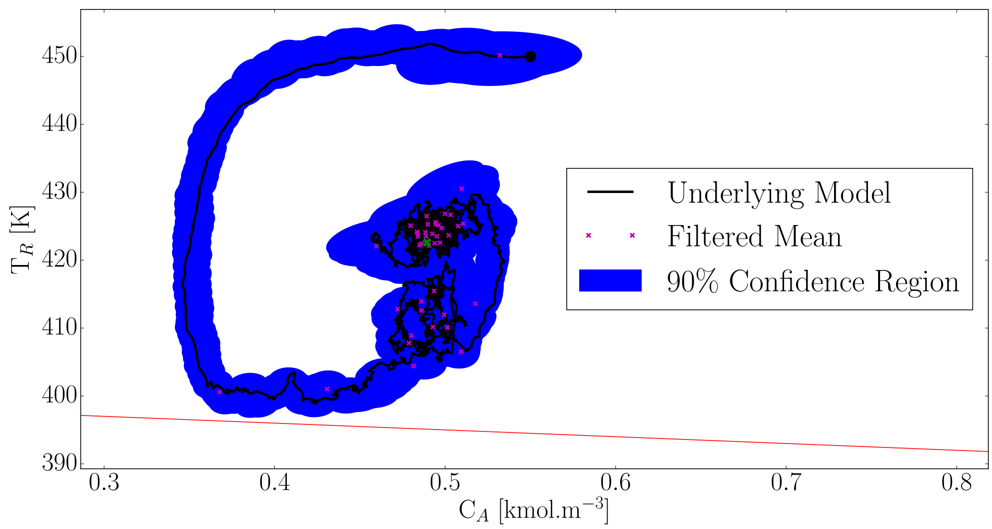

Figure 12.12: State space trajectory of the chance constrained stochastic MPC using the switching controller algorithm.

Since Figure 12.12 only illustrates that the constraint is not violated for a single run we again use a Monte-Carlo technique to justify the assertion that, for this example, the stochastic controller can successfully reduce the constraint violation probability. By simulating 100 runs it was found that the expected value stochastic MPC violated the constraint 156.9 times per run of 300 minutes. The chance constrained stochastic MPC violated the constraint

14.8 times per run of the same length. If the robustness of the switching controller can be improved there is significant upside to its implementation.

12.3 Conclusion

In this chapter we implemented the switching controller algorithm using the particle filter. Both the LQG and stochastic MPC were used in conjunction with the particle filter. While the controllers successfully regulated the system the controller/filter combination was not robust against switching noise. Since this noise has the potential to destabilise control more research needs to be done to investigate effective methods to assure stability. While stability issues plagued the implementation of the system, the fault detection and superior state estimation ability of the particle filter was found to be useful.

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.