Contents

Part I

Literature, Theory and Background Material

Part II

Single Model Systems

Chapter 6

Inference using linear models

In this section we consider probabilistic graphical models of the form shown in Figure 6.1. This model is a generalisation of the graphical model seen in Section 4. We now assume that the states $(x_0, x_1, x_2, ...)$ and observations $(y_0, y_1, y_2, ...)$ are continuous random variables but the inputs $(u_0, u_1, u_2, ...)$ are continuous deterministic variables. Models of this form are called latent dynamical systems - the famous Kalman filter model falls into this category if one assumes linearity and normality.
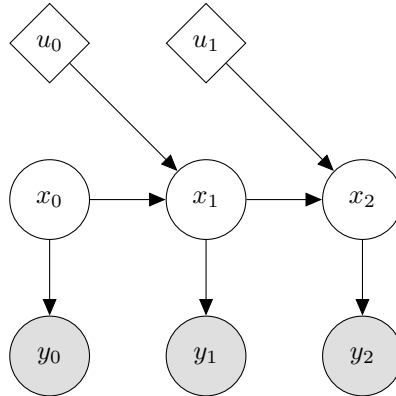
Figure 6.1: Graphical model considered in this section.

In the previous section we developed inference algorithms but assumed that the transition and observation functions were discrete. We also noted that this assumption is not appropriate for continuous systems. The reason is that one would invariably need to discretise the domain of the continuous random variable under consideration. This would result in intractably large discrete systems if one requires fine resolution. To address this issue we extend the previous model to include both continuous states and observations.

In this chapter we assume linearity and that all the random variables are Gaussian. While these are strong assumptions they form the building blocks of much more expressive models as we will discover in the next chapter. We also assume that the transition and observation functions are time invariant and that the state space model is of the form (6.1). From the

graphical model in Figure 6.1 we know that the latent variable x_t is observed through y_t.

$$x_{t+1} = Ax_t + Bu_t + w_{t+1} \text{ with } \mathcal{N}(w_{t+1}|0, W)$$
$$y_{t+1} = Cx_{t+1} + v_{t+1} \text{ with } \mathcal{N}(v_{t+1}|0, V)$$

(6.1)

Rewriting the state space model we see that the transition and observation probability density functions are given by (6.2). Note that we also assume that the system is first order Markov (see Definition 3.23).

$$p(x_{t+1}|x_t, u_t) = \mathcal{N}(x_{t+1}|Ax_t + Bu_t, W)$$
$$p(y_{t+1}|x_t + 1) = \mathcal{N}(y_{t+1}|Cx_{t+1}, V)$$

(6.2)

We have implicitly assumed that the noise is Gaussian and white[1]. Intuitively one can think of V as the noise associated with state measurements and W being a form of the uncertainty associated with the linear model of the plant. Additionally, W can also model any zero mean unmeasured disturbances which may influence the system[2]. Thus, larger V and W indicate more uncertainty in the system.

To fully specify the system we require the transition and observation probability density functions (these implicitly depend on the internal structure of the graphical model in Figure 6.1) as well as the prior (initial) distribution $p(x_0)$.

6.1 Kalman filter

The goal of filtering is to find the posterior distribution $p(x_t|y_{0:t}, u_{0:t-1})$. It is pleasing to note that this derivation will follow in an analogous manner to the filtering derivation in Chapter 4 albeit with continuous Gaussian distributions. The motivation for taking the joint of only the preceding hidden time step is the same as before. The graphical model corresponding to filtering is shown in Figure 6.2.
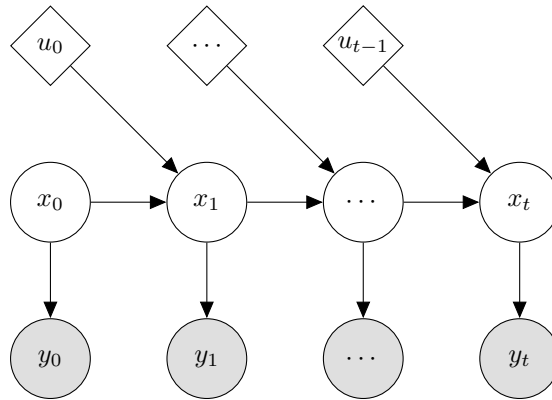

Figure 6.2: Extended - for illustration - graphical model used for filtering.

---

[1]The noise is temporally independent, has zero mean and finite variance.
[2]Note that for the purposes of this dissertation plant is a synonym for the system.

We start with the prediction expression in (6.3) and assume, due to the closure of linear conditional Gaussian distributions, that $\alpha(x_{t-1}) = p(x_{t-1}|y_{0:t-1}, u_{0:t-2}) = \mathcal{N}(x_{t-1}|\mu_{t-1}, \Sigma_{t-1})$ is available.

$$
\begin{aligned}
p(x_t|y_{0:t-1}, u_{0:t-1}) &= \int_{x_{t-1}} p(x_t, x_{t-1}|y_{0:t-1}, u_{0:t-1}) \\
&= \int_{x_{t-1}} p(x_{t-1}|y_{0:t-1}, u_{0:t-1})p(x_t|x_{t-1}, y_{0:t-1}, u_{0:t-1}) \\
&= \int_{x_{t-1}} p(x_{t-1}|y_{0:t-1}, u_{0:t-2})p(x_t|x_{t-1}, u_{t-1}) \\
&= \int_{x_{t-1}} \alpha(x_{t-1})\mathcal{N}(x_t|Ax_{t-1} + Bu_{t-1}, W) \\
&= \int_{x_{t-1}} \mathcal{N}(x_{t-1}|\mu_{t-1}, \Sigma_{t-1})\mathcal{N}(x_t|Ax_{t-1} + Bu_{t-1}, W)
\end{aligned}
\tag{6.3}
$$

Now we use Theorem 3.7 (Bayes' theorem for linear Gaussian models) to evaluate the marginal expression as shown in (6.4).

$$
\begin{aligned}
\int_{x_{t-1}} \mathcal{N}(x_{t-1}|\mu_{t-1}, \Sigma_{t-1})\mathcal{N}(x_t|Ax_{t-1} + Bu_{t-1}, W) &= \mathcal{N}(x_t|A\mu_{t-1} + Bu_{t-1}, W + A^T\Sigma_{t-1}A) \\
&= \mathcal{N}(x_t|\mu_{t|t-1}, \Sigma_{t|t-1})
\end{aligned}
\tag{6.4}
$$

Intuitively, (6.4) is the one step ahead prediction for the hidden state given all the past observations and the past and present inputs. Now we make use of Theorem 3.4 (Bayes' theorem) to update our view of x_t given the current observation as shown in (6.5).

$$
\begin{aligned}
p(x_t|y_{0:t}, u_{0:t-1}) &= p(x_t|y_t, y_{0:t-1}, u_{0:t-1}) \\
&= \frac{p(y_t|x_t, y_{0:t-1}, u_{0:t-1})p(x_t|y_{0:t-2}, u_{0:t-2}, u_{t-1})}{p(y_t|y_{0:t-1}, u_{0:t-1})} \\
&= \frac{p(y_t|x_t)p(x_t|y_{0:t-1}, u_{0:t-1})}{p(y_t|y_{0:t-1}, u_{0:t-1})} \\
&\propto p(y_t|x_t)p(x_t|y_{0:t-1}, u_{0:t-1}) \\
&= p(y_t|x_t)\mathcal{N}(x_t|A\mu_{t-1} + Bu_{t-1}, W + A^T\Sigma_{t-1}A) \\
&= \mathcal{N}(y_t|Cx_t, V)\mathcal{N}(x_t|\mu_{t|t-1}, \Sigma_{t|t-1})
\end{aligned}
\tag{6.5}
$$

Now we again make use of Theorem 3.7 to evaluate the conditional expression as shown in (6.6).

$$
\begin{aligned}
p(x_t|y_{0:t}, u_{0:t-1}) &= \mathcal{N}(y_t|Cx_t, V)\mathcal{N}(x_t|\mu_{t|t-1}, \Sigma_{t|t-1}) \\
&= \mathcal{N}(x_t|\Gamma(C^TV^{-1}y + \Sigma_{t|t-1}^{-1}\mu_{t|t-1}), \Gamma) \\
&\text{with } \Gamma = (\Sigma_{t|t-1}^{-1} + C^TV^{-1}C)^{-1}
\end{aligned}
\tag{6.6}
$$

By using the matrix identity $(A + BD^{-1}C)^{-1} = A^{-1} - A^{-1}B(D + CA^{-1}B)^{-1}CA^{-1}$ and defining $K_t = \Sigma_{t|t-1}C^T(C\Sigma_{t|t-1}C^T + V)^{-1}$ we can simplify Γ to the recursive posterior covariance estimate shown in (6.7). Similarly, using the same matrix identity together with $(P^{-1}B^TR^{-1}B)^{-1})^{-1}B^TR^{-1} = PB^T(BPB^T + R^{-1})$ and the definition of K_t we have the

posterior mean estimate as shown in (6.8). Together (6.7) and (6.8) are known as the Kalman filter equations [44].

$$\Sigma_t = (I - K_t C)\Sigma_{t|t-1} \tag{6.7}$$

$$\mu_t = \mu_{t|t-1} + K_t(y_t - C\mu_{t|t-1}) \tag{6.8}$$

Note that for the first time step only the update expression is evaluated as the prediction is the prior of x_0.

Intuitively, the Kalman filter equations use the state space model to predict the new state distribution and then adjust it by a correction factor $K_t(y_t - C\mu_{t|t-1})$. This factor depends on the difference between the actual observation and the predicted observation. The Kalman gain, K_t, represents the inferred confidence of the model. If the model is deemed accurate then the predictions make up most of μ_t but if the model is bad at predicting the observations then the observations play a bigger part in the next mean estimate [6].

6.2   Kalman prediction

The goal of prediction is to find an expression for the distributions $p(x_{t+h}|y_{0:t}, u_{0:t+h-1})$ and $p(y_{t+h}|y_{0:t}, u_{0:t+h-1})$ with $h \geq 1$. Note that these derivations follow in exactly the same way as the prediction derivations did in Chapter 4 given the current posterior state estimate of x_0. The reason for this is because the graphical models are the same (the deterministic inputs don't change the structure of the underlying random variable network). The graphical model corresponding to the two step ahead state prediction is shown in Figure 6.3.
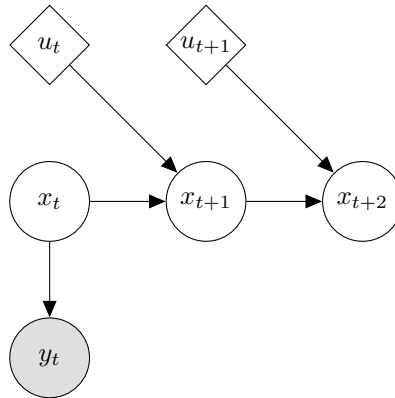

Figure 6.3: Graphical model used for state prediction.

We start the derivation by considering the one step ahead state prediction in (6.9).

$$
\begin{aligned}
p(x_{t+1}|y_{0:t}, u_{0:t}) &= \int_{x_t} p(x_{t+1}, x_t|y_{0:t}, u_{0:t}) \\
&= \int_{x_t} p(x_t|y_{0:t}, u_{0:t-1})p(x_{t+1}|x_t, y_{0:t}, u_{0:t}) \\
&= \int_{x_t} p(x_t|y_{0:t}, u_{0:t-1})p(x_{t+1}|x_t, u_t) \\
&= \int_{x_t} \alpha(x_t)p(x_{t+1}|x_t, u_t) \\
&= \int_{x_t} \mathcal{N}(x_t|\mu_t, \Sigma_t)\mathcal{N}(x_{t+1}|Ax_t + Bu_t, W) \\
&= \mathcal{N}(x_{t+1}|Ax_t + Bu_t, W + A\Sigma_t A^T) \\
&= \mathcal{N}(x_{t+1}|\mu_{t+1|t}, \Sigma_{t+1|t})
\end{aligned}
\tag{6.9}
$$

Note that μ_t and Σ_t is the filtered mean and covariance found by the Kalman filter. We have again relied upon Theorem 3.7 to evaluate the marginal integral. We now consider the two step ahead state prediction in (6.10).

$$
\begin{aligned}
p(x_{t+2}|y_{0:t}, u_{0:t+1}) &= \int_{x_{t+1}} p(x_{t+2}, x_{t+1}|y_{0:t}, u_{0:t+1}) \\
&= \int_{x_{t+1}} p(x_{t+1}|y_{0:t}, u_{0:t})p(x_{t+2}|x_{t+1}, y_{0:t}, u_{0:t+1}) \\
&= \int_{x_{t+1}} p(x_{t+1}|y_{0:t}, u_{0:t})p(x_{t+2}|x_{t+1}, u_{t+1}) \\
&= \int_{x_t} \mathcal{N}(x_{t+1}|\mu_{t+1|t}, \Sigma_{t+1|t})\mathcal{N}(x_{t+2}|Ax_{t+1} + Bu_{t+1}, W) \\
&= \mathcal{N}(x_{t+2}|A\mu_{t+1|t} + Bu_{t+1}, W + A\Sigma_{t+1|t}A^T) \\
&= \mathcal{N}(x_{t+2}|\mu_{t+2|t}, \Sigma_{t+2|t})
\end{aligned}
\tag{6.10}
$$

It is clear that we have derived a recursive algorithm to estimate the hth-step ahead state prediction as shown in (6.11).

$$
\begin{aligned}
&p(x_{t+h}|y_{0:t}, u_{0:t+h}) = \mathcal{N}(x_{t+h}|\mu_{t+h|t}, \Sigma_{t+h|t}) \\
&\text{with } \mu_{t+h|t} = A\mu_{t+h-1|t} + Bu_{t+h-1} \\
&\text{and } \Sigma_{t+h|t} = W + A\Sigma_{t+h-1|t}A^T \\
&\text{and } \mu_{t+1|t} = A\mu_t + Bu_t \\
&\text{and } \Sigma_{t+1|t} = W + A\Sigma_t A^T
\end{aligned}
\tag{6.11}
$$

Inspecting (6.11) we see that the predictive distribution is just the forward projection, using the transition function, of the filtered distribution. Note that it is possible for $\Sigma_{t+h|t}$ to become smaller, in some normed ($|\cdot|$) sense, for increasing h (obviously bounded by Q below). For, if the eigenvalues of A are less than unity we have that $|A\Sigma_{t+h|t}A^T| \leq |A\Sigma_{t+h-1|t}A^T|$.

Next we consider the observation prediction, $p(y_{t+h}|y_{0:t}, u_{0:t+h-1})$. Again consider the one

step ahead prediction as shown in (6.12).

$$
\begin{aligned}
p(y_{t+1}|y_{0:t}, u_{0:t}) &= \int_{x_t, x_{t+1}} p(y_{t+1}, x_{t+1}, x_t|y_{0:t}, u_{0:t}) \\
&= \int_{x_t, x_{t+1}} p(x_t|y_{0:t}, u_{0:t-1})p(y_{t+1}, x_{t+1}|x_t, y_{0:t}, u_{0:t}) \\
&= \int_{x_t, x_{t+1}} p(x_t|y_{0:t}, u_{0:t-1})p(x_{t+1}|x_t, y_{0:t}, u_{0:t})p(y_{t+1}|x_{t+1}, x_t, y_{0:t}, u_{0:t}) \\
&= \int_{x_t, x_{t+1}} \alpha(x_t)p(x_{t+1}|x_t, u_t)p(y_{t+1}|x_{t+1}) \\
&= \int_{x_t, x_{t+1}} \mathcal{N}(x_t|\mu_t, \Sigma_t)\mathcal{N}(x_{t+1}|Ax_t + Bu_t, W)\mathcal{N}(y_{t+1}|Cx_{t+1}, V) \\
&= \mathcal{N}(y_{t+1}|C\mu_{t+1|t}, V + C\Sigma_{t+1|t}C^T)
\end{aligned}
\tag{6.12}
$$

We have again used Theorem 3.7 and used the nomenclature of the one step ahead state prediction derivation. For the sake of brevity we trust that the reader will see the similarity between the two derivations and allow us to conclude, without proof, that the hth-step ahead observation prediction is given by (6.13).

$$
p(y_{t+h}|y_{0:t}, u_{0:t+h-1}) = \mathcal{N}(y_{t+h}|C\mu_{t+h|t}, R + C\Sigma_{t+h|t}C^T)
\tag{6.13}
$$

It is reassuring to note that the observation prediction is just the state prediction transformed by the observation function.

6.3   Smoothing and Viterbi decoding

For the sake of completeness we state the Kalman smoothing equations and briefly discuss Viterbi decoding within the context of conditional linear Gaussian systems.

The reason we do not go into detail with the smoothing algorithm is because it follows much the same structure as the hidden Markov model smoothing algorithm except that we make use of Theorem 3.7 to simplify the algebra. We are also primarily only interested in filtering and prediction because they are important for the purposes of control which is the focus of this dissertation.

The smoothing algorithm, also called the Rauch, Tung and Striebel (RTS) algorithm for $p(x_t|y_{0:T}, u_{0:T-1})$ is also a Gaussian distribution of the form $\mathcal{N}(\hat{\mu}_t, \hat{\Sigma}_t)$. The recursion expressions for the posterior mean and covariance are shown in (6.14).

$$
\begin{aligned}
&\hat{\mu}_t = \mu_t + J_t\left(\hat{\mu}_{t+1} - (A\mu_t + Bu_{t-1})\right) \\
&\hat{\Sigma}_t = \Sigma_t + J_t(\hat{\Sigma}_{t+1} - P_t)J_t^T \\
&\text{with } P_t = A\Sigma_t A^T + W \\
&\text{and } J_t = \Sigma_t A^T (P_t)^{-1} \\
&\text{and } \hat{\mu}_T = \mu_T \\
&\text{and } \hat{\Sigma}_T = \Sigma_T
\end{aligned}
\tag{6.14}
$$

Finally, we know from the Definition 3.19 (chain rule for Bayesian networks) and Figure 6.1 that the joint distribution for $p(x_{0:T}, y_{0:T}, u_{0:T-1}) = p(x_1)p(y_1|x_1)\Pi_{t=2}^{T}p(y_t|x_t)p(x_t|x_{t-1}, u_{t-1})$. Since Gaussian distributions are closed under multiplication this joint distribution is also a Gaussian distribution. It can be shown that maximising with respect to all latent variables jointly or maximising with respect to the marginal distributions of the latent variables is the same because the mean and the mode of a Gaussian distribution coincide [3]. This implies that Viterbi decoding is just the sequence of means found by the smoothing algorithm.

6.4 Filtering the CSTR

In this section we apply the Kalman filter to the CSTR introduced in Section 5. We use the linear model around the unstable operating point (C_A^2, T_R^2) as shown in (6.15). Note that the matrix A and vectors B, b depend on the step size and should be recalculated for different h. To make things concrete we have used $h = 0.1$ here. Note that V indicates that we only measure temperature for now.

$$
A = \begin{pmatrix} 0.9959 & -6.0308 \times 10^{-5} \\ 0.4186 & 1.0100 \end{pmatrix} \quad B = \begin{pmatrix} 0 \\ 8.4102 \times 10^{-5} \end{pmatrix} \quad C = \begin{pmatrix} 0 & 1 \end{pmatrix}
$$
$$
W = \begin{pmatrix} 1 \times 10^{-6} & 0 \\ 0 & 0.1 \end{pmatrix} \quad V = \begin{pmatrix} 10 \end{pmatrix}
$$
(6.15)

The system noise W indicates that the standard deviation of the concentration component of the model is 0.001 kmol/m^{-3} and the temperature component is 0.32 K. While these variances may seem small, bear in mind that noise is added at each time step which compounds its effect. The measurement noise implies that 68% of the measurements will fall between $\pm\sqrt{10}$ of the actual state. We use an initial state with mean at the initial condition and covariance W.

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. In this chapter we will primarily only investigate the benefit gained by including more state measurements.

In Figure 6.4 we illustrate the strengths and weaknesses of the Kalman filter. Since we derived the recursion equations analytically it is computationally efficient to use, the biggest cost is a matrix inversion which needs to be computed at each time step[3]. During the initial part of the simulation the filter very accurately estimates the current system states because the model is accurate in this region. Thus the filter is able to infer the true state in the presence of noisy measurements.

---

[3]It is even possible to avoid this step by noticing that the posterior covariance quickly converges to a constant covariance.

Unfortunately the recursion equations assumed the system can be described by a linear model. With time the trajectories move away from the linearisation point (because the linearisation point is unstable) and thus the linear model becomes less accurate. This has a detrimental effect on the quality of the Kalman filter estimate as the filter effectively starts to solely rely on the measurements to infer the states. This works reasonably well for the measured states (T_R), but since we do not measure concentration the filter is forced to incorporate the linear model prediction which is grossly inaccurate.

The average concentration estimation error[4] throughout the run is 22.73% while the average temperature estimation error is 0.47%. Clearly there is a significant benefit to measuring the state one wishes to infer.
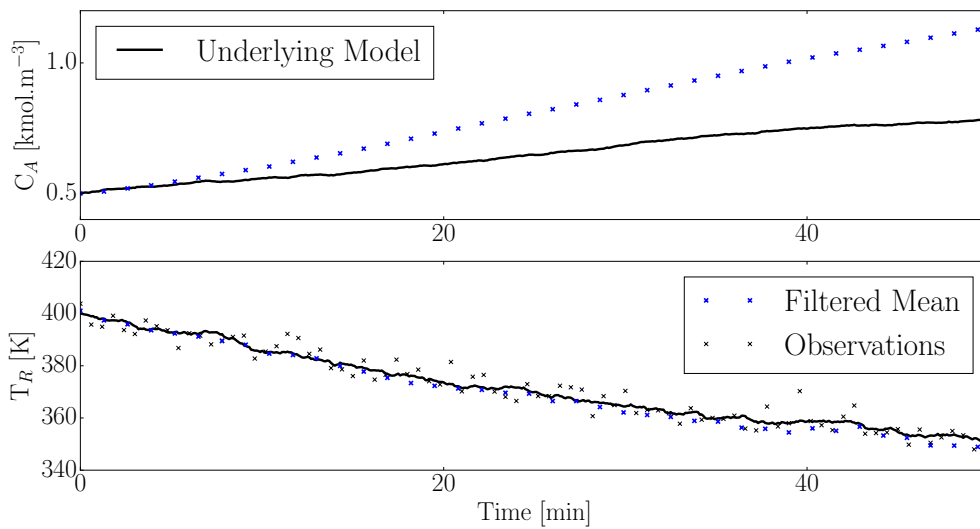

Figure 6.4: Kalman filter superimposed on the time series evolution of the CSTR with initial condition $(0.50, 400)$ and measuring only temperature.

In Figure 6.5 we see another interesting property of Kalman filters. The posterior covariance quickly converges to a constant value (the confidence region quickly stops changing shape) which is independent of the observations. This is a general property of linear Gaussian systems [3] and is evident from the recursion expression. The modelled system dynamics and noise are the only factors affecting the covariance. If the model is accurate this is not a problem but we see that as the model becomes less accurate the filter maintains the same level of confidence in its estimate. This is quite undesirable behaviour because the confidence in the estimate is not a function of the observations.

It is also interesting to consider the shape of the confidence region. Notice that it is short vertically - indicating less uncertainty in the temperature state dimension but wide horizontally -

---

[4]We define the average estimation error by $\frac{1}{N} \sum_{t=0}^{N} |\frac{\hat{x}_t - x_t}{x_t}|$ where $\hat{x}_t$ is the inferred state and and x_t the true underlying state at time t.

indicating more uncertainty in the concentration state dimension. Intuitively this is plausible because, since we do not measure concentration, we are less sure about the underlying state.

In Figure 6.5 we see that while the temperature estimate is still trustworthy (the temperature mean estimate lines up horizontally with the true underlying state) the concentration estimate diverges.
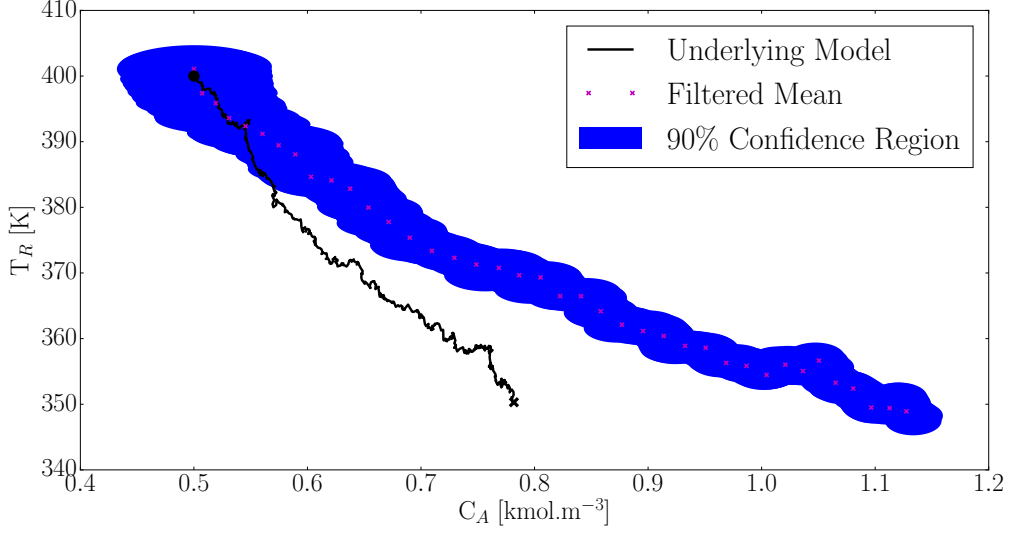

Figure 6.5: State space diagram of the CSTR with mean and 90% confidence region superimposed thereupon. Only temperature is measured.

The root of the problem lies in the unsuitability of the model rather than our inference technique. It can be shown that for linear systems with Gaussian noise the Kalman filter is the optimal state estimator [1].

Based on our discussion in Section 5 where the CSTR example was introduced we know that the linear models will not always be very accurate. We therefore modify (6.15) to also incorporate concentration measurements. In this case we have that $C = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ and $V = \begin{pmatrix} 1 \times 10^{-3} & 0 \\ 0 & 10 \end{pmatrix}$ with everything else the same. The time evolution of the states is shown in Figure 6.6 and the state space representation is shown in Figure 6.7.
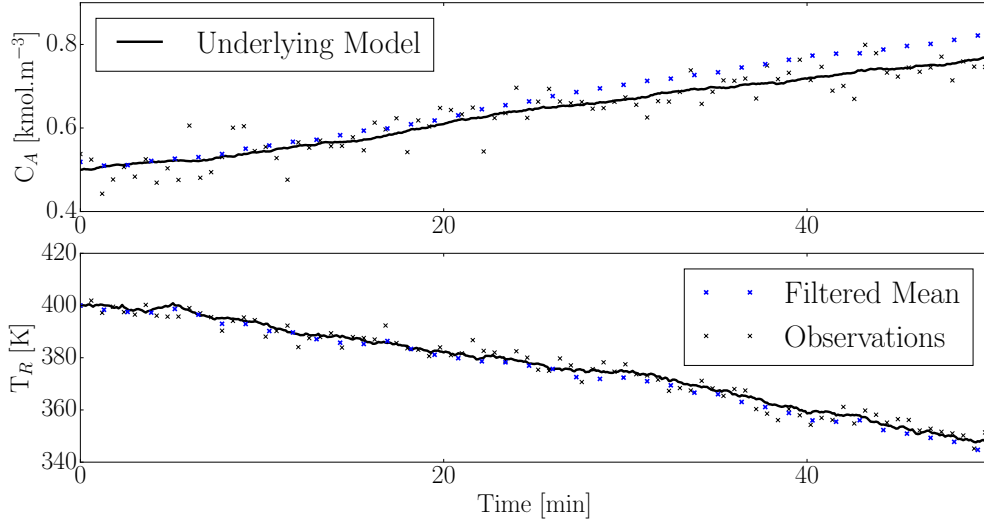
Figure 6.6: Kalman filter superimposed on the time series evolution of the CSTR with initial condition $(0.50, 400)$ and measuring both temperature and concentration.
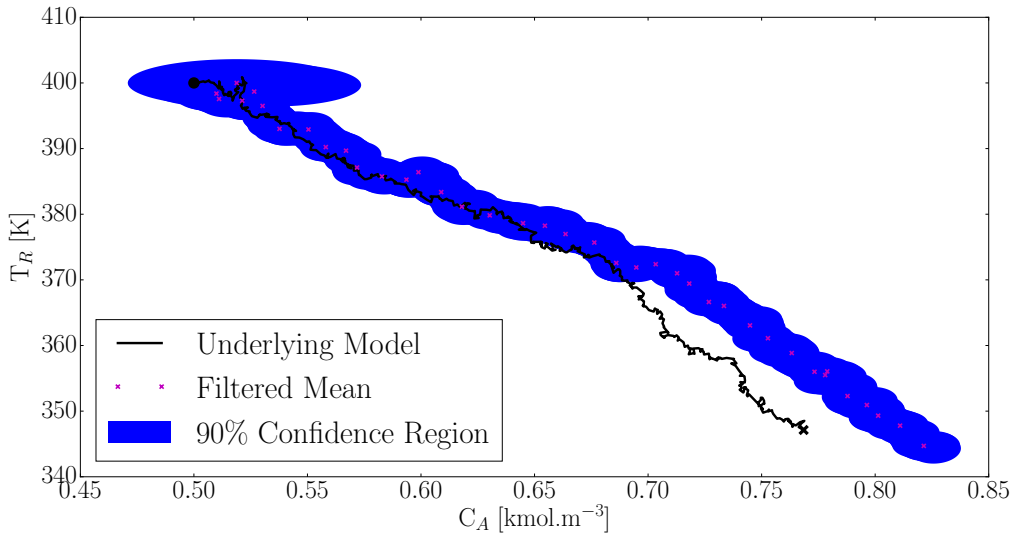
Figure 6.7: State space diagram of the CSTR with mean and 90% confidence region superimposed thereupon. Both concentration and temperature are measured.

Comparing Figures 6.5 and 6.7 we see that by incorporating the state measurement the state estimation is much more accurate. The average concentration and temperature estimation error is only 4.09% and 0.45% respectively. It is not necessary to directly measure concentration as we have done: any measurement which depends on C_A (or even both C_A and T_R) would suffice. The second measurement reduces our uncertainty in the concentration state estimate because we have more to base our inference on than just a bad model.

Part III

Multiple Model Systems

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 non-linear 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.