Contents

5 Inference using Linear Models

In this section we consider probabilistic graphical models of the form shown in Figure 23. This model is a generalisation of the graphical model seen in the Hidden Markov Model section. We now assume that the states $(x_1, x_2, ...)$ and observations $(y_1, y_2, ...)$ are continuous random variables but the inputs $(u_1, u_2, ...)$ are deterministic. Models of this form are called Latent Linear Dynamical Systems (the famous Kalman Filter model falls into this category).

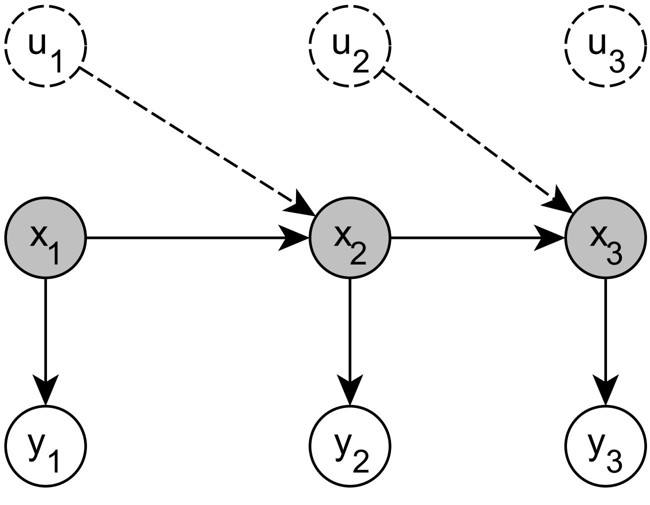

Figure 23: Graphical model of this section

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 data. 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.

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 section. We also assume that the transition and emission functions are time invariant and that the state space model is of the form (61).

$$\begin{aligned} x_{t+1} &= Ax_t + Bu_t + w_{t+1} \text{ with } \mathcal{N}(w_{t+1}|0, W) \\ y_{t+1} &= Cx_{t+1} + v_{t+1} \text{ with } \mathcal{N}(v_{t+1}|0, V) \end{aligned} \tag{61}$$

Rewriting the state space model we see that the transition and emission probability density functions are given by (62). Note that we also assume that the system is first order Markov.

$$\begin{aligned} p(x_{t+1}|x_t, u_t) &= \mathcal{N}(x_{t+1}|Ax_t + Bu_t, W) \\ p(y_{t+1}|x_t) &= \mathcal{N}(y_{t+1}|Cx_{t+1}, V) \end{aligned} \tag{62}$$

We have implicitly assumed that the noise is Gaussian and white[3]. 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 unmeasured disturbances which may influence the system[4]. Thus, larger V and W indicate more uncertainty in the system.

To fully specify the system we require the transition and emission probability density functions (these implicitly depend on the internal structure of the graphical model in Figure 23) as well as the prior (initial) distribution of x_0.

5.1 Filtering

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

We start with the prediction expression in (63) and assume, due to the closure of linear conditional Gaussian distributions, that $\alpha(x_{t-1}) = p(x_{t-1}|y_{1:t-1}, u_{1:t-2}) = \mathcal{N}(x_{t-1}|\mu_{t-1}, \Sigma_{t-1})$.

$$\begin{aligned}
p(x_t|y_{1:t-1}, u_{1:t-1}) &= \int_{x_{t-1}} p(x_t, x_{t-1}|y_{1:t-1}, u_{1:t-1}) \\
&= \int_{x_{t-1}} p(x_{t-1}|y_{1:t-1}, u_{1:t-1})p(x_t|x_{t-1}, y_{1:t-1}, u_{1:t-1}) \\
&= \int_{x_{t-1}} p(x_{t-1}|y_{1:t-1}, u_{1:t-2})p(x_t|x_{t-1}, u_{1:t-1}) \\
&= \int_{x_{t-1}} \alpha(x_{t-1})\mathcal{N}(x_t|Ax_{t-1} + Bu_{t-1}, W) \\
&= \int_{x_{t-1}} \mathcal{N}(x_{t-1}|\mu_{t-1}, \Sigma_{t-1})\mathcal{N}(x_t|Ax_{t-1} + Bu_{t-1}, W)
\end{aligned} \tag{63}$$

Now we use Bayes' Theorem for Linear Gaussian Models to evaluate the marginal expression as shown in (64).

$$\begin{aligned}
\int_{x_{t-1}} \mathcal{N}(x_{t-1}|\mu_{t-1}, \Sigma_{t-1})\mathcal{N}(x_t|Ax_{t-1} + Bu_{t-1}, W) &= \mathcal{N}(x_t|A\mu_{t-1} + Bu_{t-1}, W + A^T\Sigma_{t-1}A) \\
&= \mathcal{N}(x_t|\mu_{t|t-1}, \Sigma_{t|t-1})
\end{aligned} \tag{64}$$

Intuitively, (64) 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 Bayes' Theorem to

[3] The noise is temporally independent, has zero mean and finite variance.

[4] Note that for the purposes of this dissertation plant is a synonym for the system.

update our view of x_t given the current observation as shown in (65).

$$\begin{aligned}
p(x_t|y_{1:t}, u_{1:t-1}) &= p(x_t|y_t, y_{1:t-1}, u_{1:t-1}) \\
&= \frac{p(y_t|x_t, y_{1:t-1}, u_{1:t-1})p(x_t|y_{1:t-1}, u_{1:t-2}, u_{t-1})}{p(y_t|y_{1:t-1}, u_{1:t-1})} \\
&= \frac{p(y_t|x_t)p(x_t|y_{1:t-1}, u_{1:t-1})}{p(y_t|y_{1:t-1}, u_{1:t-1})} \\
&\propto p(y_t|x_t)p(x_t|y_{1:t-1}, u_{1:t-1}) \\
&= p(y_t|x_t)\mathcal{N}(x_t|A\mu_{t-1} + Bu_{t-1}, W + A^T\Sigma_{t-1}A) \\
&= \mathcal{N}(y_t|Cx_t, V)\mathcal{N}(x_t|\mu_{t|t-1}, \Sigma_{t|t-1})
\end{aligned} \tag{65}$$

Now we again make use of Bayes' Theorem for Linear Gaussian Models to evaluate the conditional expression as shown in (66).

$$\begin{aligned}
p(x_t|y_{1:t}, u_{1:t-1}) &= \mathcal{N}(y_t|Cx_t, V)\mathcal{N}(x_t|\mu_{t|t-1}, \Sigma_{t|t-1}) \\
&= \mathcal{N}(x_t|\Gamma(C^TV^{-1}y + \Sigma_{t|t-1}^{-1}\mu_{t|t-1}), \Gamma) \\
&\text{with } \Gamma = (\Sigma_{t|t-1}^{-1} + C^TV^{-1}C)^{-1}
\end{aligned} \tag{66}$$

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 (67). 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 (68). Together (67) and (68) are known as the Kalman Filter equations [36].

$$\Sigma_t = (I - K_tC)\Sigma_{t|t-1} \tag{67}$$

$$\mu_t = \mu_{t|t-1} + K_t(y_t - C\mu_{t|t-1}) \tag{68}$$

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 [5].

5.2 Prediction

The goal of prediction is to find an expression for the distributions $p(x_{t+h}|y_{1:t}, u_{1:t+h-1})$ and $p(y_{t+h}|y_{1:t}, u_{1:t+h-1})$ with $h \geq 1$. Note that these derivations follow in exactly the same way as the prediction derivations did for the Hidden Markov Models. 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).

We start the derivation by considering the one step ahead state prediction in (69).

$$\begin{aligned}
p(x_{t+1}|y_{1:t}, u_{1:t}) &= \int_{x_t} p(x_{t+1}, x_t|y_{1:t}, u_{1:t}) \\
&= \int_{x_t} p(x_t|y_{1:t}, u_{1:t-1})p(x_{t+1}|x_t, y_{1:t}, u_{1:t}) \\
&= \int_{x_t} p(x_t|y_{1:t}, u_{1:t-1})p(x_{t+1}|x_t, u_t) \\
&= \int_{x_t} \alpha(x_t)p(x_{t+1}|x_t, u_t) \\
&= \int_{x_t} \mathcal{N}(x_t|\mu_t, \Sigma_t)\mathcal{N}(x_{t+1}|Ax_t + Bu_t, W) \\
&= \mathcal{N}(x_{t+1}|Ax_t + Bu_t, W + A\Sigma_t A^T) \\
&= \mathcal{N}(x_{t+1}|\mu_{t+1|t}, \Sigma_{t+1|t})
\end{aligned} \tag{69}$$

Note that μ_t and Σ_t is the filtered mean and covariance. We have again relied upon Bayes' Theorem for Linear Gaussian Models to evaluate the marginal integral. We now consider the two step ahead state prediction in (70).

$$\begin{aligned}
p(x_{t+2}|y_{1:t}, u_{1:t+1}) &= \int_{x_{t+1}} p(x_{t+2}, x_{t+1}|y_{1:t}, u_{1:t+1}) \\
&= \int_{x_{t+1}} p(x_{t+1}|y_{1:t}, u_{1:t})p(x_{t+2}|x_{t+1}, y_{1:t}, u_{1:t+1}) \\
&= \int_{x_{t+1}} p(x_{t+1}|y_{1:t}, u_{1:t})p(x_{t+2}|x_{t+1}, u_{t+1}) \\
&= \int_{x_t} \mathcal{N}(x_{t+1}|\mu_{t+1|t}, \Sigma_{t+1|t})\mathcal{N}(x_{t+2}|Ax_{t+1} + Bu_{t+1}, W) \\
&= \mathcal{N}(x_{t+2}|A\mu_{t+1|t} + Bu_{t+1}, W + A\Sigma_{t+1|t}A^T) \\
&= \mathcal{N}(x_{t+2}|\mu_{t+2|t}, \Sigma_{t+2|t})
\end{aligned} \tag{70}$$

It is clear that we have derived a recursive algorithm to estimate the h^{th}-step ahead state prediction as shown in (71).

$$\begin{aligned}
&p(x_{t+h}|y_{1:t}, u_{1:t+h}) = \mathcal{N}(x_{t+h}|\mu_{t+h|t}, \Sigma_{t+h|t}) \\
&\text{with } \mu_{t+h|t} = A\mu_{t+h-1|t} + Bu_{t+h-1} \\
&\text{and } \Sigma_{t+h|t} = W + A\Sigma_{t+h-1|t}A^T \\
&\text{and } \mu_{t+1|t} = A\mu_t + Bu_t \\
&\text{and } \Sigma_{t+1|t} = W + A\Sigma_t A^T
\end{aligned} \tag{71}$$

Inspecting (71) 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 for increasing h (obviously bounded by Q below). For, if the eigenvalues of A are less than one 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, $p(y_{t+h}|y_{1:t}, u_{1:t+h-1})$. Again consider the one step ahead prediction as shown in (72).

$$
\begin{aligned}
p(y_{t+1}|y_{1:t},u_{1:t}) &= \int_{x_t,x_{t+1}} p(y_{t+1},x_{t+1},x_t|y_{1:t},u_{1:t}) \\
&= \int_{x_t,x_{t+1}} p(x_t|y_{1:t},u_{1:t-1})p(y_{t+1},x_{t+1}|x_t,y_{1:t},u_{1:t}) \\
&= \int_{x_t,x_{t+1}} p(x_t|y_{1:t},u_{1:t-1})p(x_{t+1}|x_t,y_{1:t},u_{1:t})p(y_{t+1}|x_{t+1},x_t,y_{1:t},u_{1:t}) \\
&= \int_{x_t,x_{t+1}} \alpha(x_t)p(x_{t+1}|x_t,u_t)p(y_{t+1}|x_{t+1}) \\
&= \int_{x_t,x_{t+1}} \mathcal{N}(x_t|\mu_t,\Sigma_t)\mathcal{N}(x_{t+1}|Ax_t+Bu_t,W)\mathcal{N}(y_{t+1}|Cx_{t+1},V) \\
&= \mathcal{N}(y_{t+1}|C\mu_{t+1|t},V+C\Sigma_{t+1|t}C^T)
\end{aligned}
\tag{72}
$$

We have again used Bayes' Theorem for Linear Gaussian Models 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 h^{th}-step ahead observation prediction is given by (73).

$$
p(y_{t+h}|y_{1:t},u_{1:t+h-1}) = \mathcal{N}(y_{t+h}|C\mu_{t+h|t},R+C\Sigma_{t+h|t}C^T) \tag{73}
$$

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

5.3 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 again make use of Bayes' Theorem for Linear Gaussian Models. 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 $p(x_t|y_{1:T},u_{1:T-1})$ 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 (74).

$$
\begin{aligned}
&\hat{\mu}_t = \mu_t + J_t\left(\hat{\mu}_{t+1} - (A\mu_t + Bu_{t-1})\right) \\
&\hat{\Sigma}_t = \Sigma_t + J_t(\hat{\Sigma}_{t+1} - P_t)J_t^T \\
&\text{with } P_t = A\Sigma_t A^T + W \\
&\text{and } J_t = \Sigma_t A^T(P_t)^{-1} \\
&\text{and } \hat{\mu}_T = \mu_T \\
&\text{and } \hat{\Sigma}_T = \Sigma_T
\end{aligned}
\tag{74}
$$

Finally, we know from the Chain Rule for Bayesian Networks and Figure 23 that the joint distribution for $p(x_{1:T}, y_{1:T}, u_{1:T-1}) = p(x_1)p(y_1|x_1)\Pi_{t=2}^{T} p(y_t|x_t)p(x_t|x_{t-1}, u_{t-1})$. 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].

5.4 Filtering the CSTR

In this section we apply the Kalman Filter to the CSTR introduced earlier. We use the linear model around the unstable operating point (C_A^2, T_R^2) as shown in (75). 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 we only measure temperature for now.

$$
\begin{aligned}
A &= \begin{pmatrix} 0.9959 & -6.0308 \times 10^{-5} \\ 0.4186 & 1.0100 \end{pmatrix} \\
B &= \begin{pmatrix} 0 \\ 8.4102 \times 10^{-5} \end{pmatrix} \\
C &= \begin{pmatrix} 0 & 1 \end{pmatrix} \\
W &= \begin{pmatrix} 1 \times 10^{-6} & 0 \\ 0 & 0.1 \end{pmatrix} \\
V &= \begin{pmatrix} 10 \end{pmatrix}
\end{aligned}
\tag{75}
$$

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.

In Figure 24 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. 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.

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.

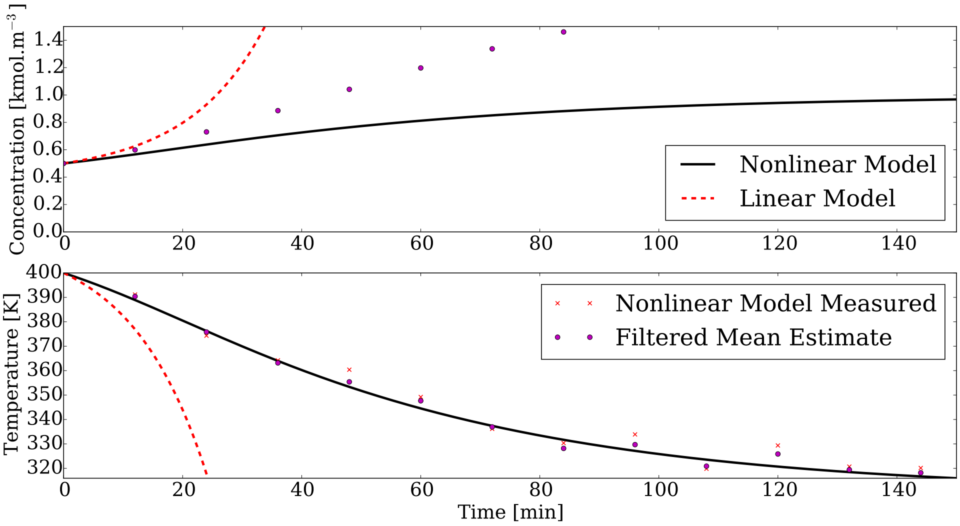

Figure 24: Kalman Filter superimposed on the time series evolution of the CSTR with initial condition $(0.50, 400)$ and measuring only temperature.

In Figure 25 we see another interesting property of Kalman Filters. The posterior covariance quickly converges to a constant value (the error ellipses quickly stop 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 error ellipses. Notice how they are short vertically - indicating less uncertainty in the temperature state dimension but wide horizontally - 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 25 we see that while the temperature estimate is still trustworthy (the black crosses line up horizontally with the red crosses) the concentration estimate diverges.

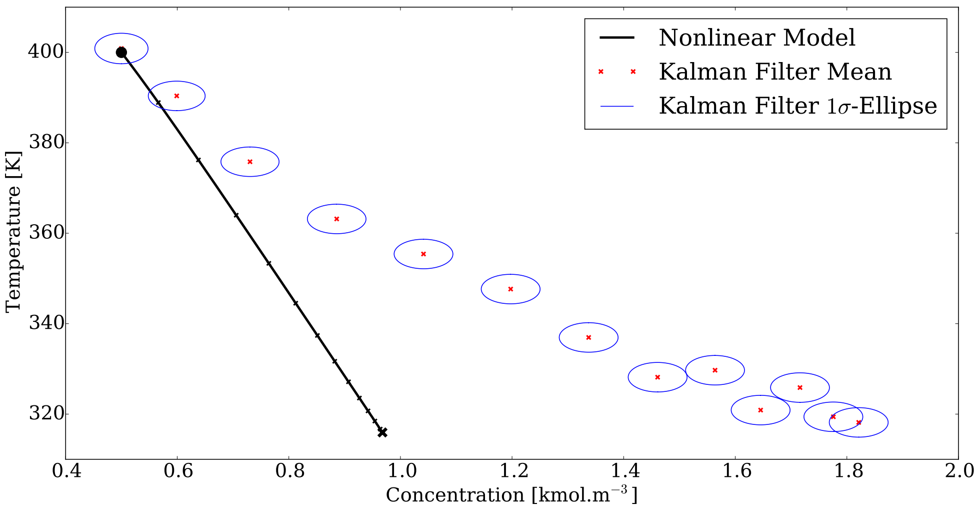

Figure 25: Phase plane of the hidden states of the CSTR with mean and one-sigma ellipses superimposed thereupon. Only temperature is measured.

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 where the CSTR example was introduced we know that the linear models will not always be very accurate. We therefore modify (75) 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 26 and the state space representation is shown in Figure 27.

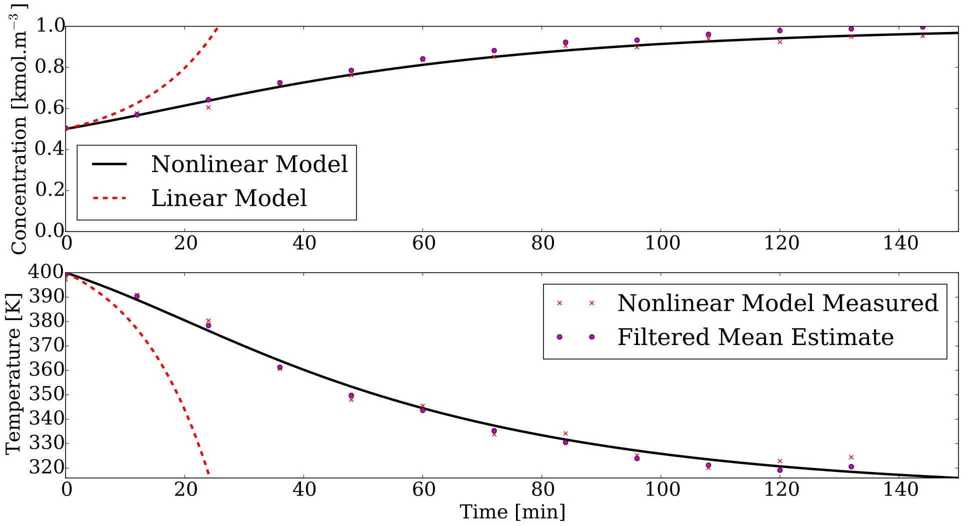

Figure 26: Kalman Filter superimposed on the time series evolution of the CSTR with initial condition $(0.50, 400)$ and measuring both temperature and concentration.

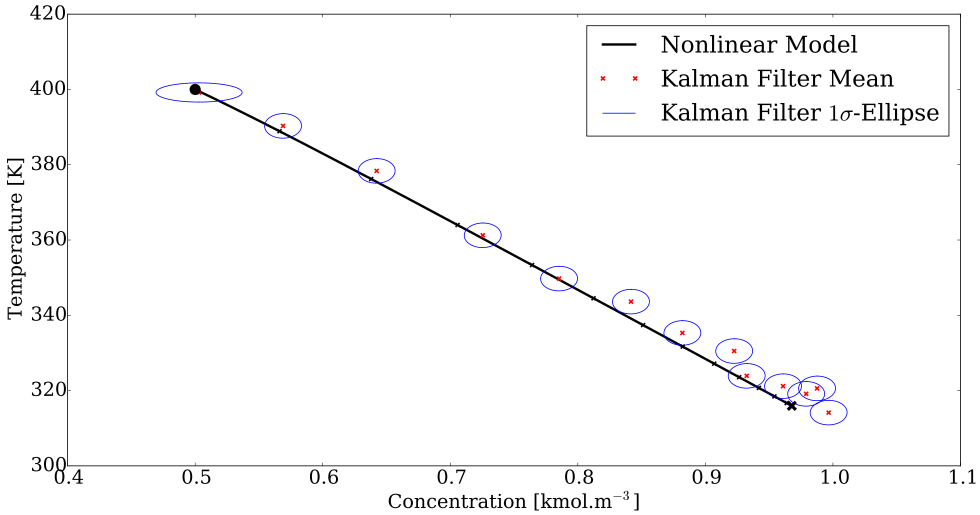

Figure 27: State space diagram of the CSTR with mean and one-sigma ellipses superimposed thereupon. Both concentration and temperature are measured.

Comparing Figures 25 and 27 we see that by incorporating the state measurement the state estimation is much more accurate. 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.

6 Inference using Nonlinear Models

In this section we still consider probabilistic graphical models of the form shown in Figure 28. The variables retain their meaning as before but we generalise the model by dropping the linearity assumption. Unfortunately, this generalisation, although allowing us to expand our investigation to a much more expressive class of models, makes closed form solutions to the inference problem intractable in general.

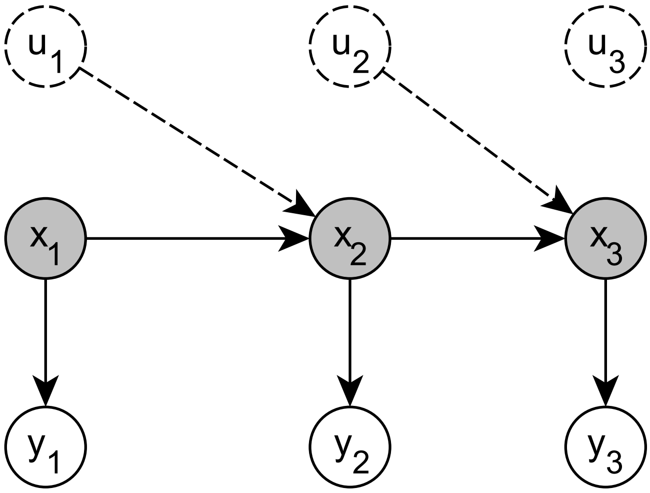

Figure 28: Graphical model of this section

We again assume that the transition and emission functions are time invariant. The state space model is now of the form (76).

$$\begin{aligned} x_{t+1} &= f(x_t, u_t, w_{t+1}) \\ y_{t+1} &= g(x_{t+1}, v_{t+1}) \end{aligned} \tag{76}$$

Note that we make no assumption about the functional form of the noise terms w_t, v_t. In practice it is customary to assume that they have zero mean but otherwise are not restricted. Additionally, to simplify notation we will omit the dependence on u of f and g and their associated distributions. Since u is a deterministic variable, by assumption, it is straightforward to incorporate it into later analysis.

6.1 Sequential Monte Carlo Methods

Many approximate inference techniques exist in literature, the most notable ones include Gaussian Sum Filters [22] and Particle based methods. We shall focus only on Sequential Monte Carlo (SMC) methods, of which Particle based methods are a subset, because it is simple to implement and generalises well (and easily) to more complex graphical models.

SMC methods are a general class of Monte Carlo methods which sample sequentially from the growing target distribution $\pi_t(x_{0:t})$. By only requiring that γ_t be known point-wise we have the framework of SMC methods as shown in (77). Note that Z_t is some normalisation constant [14].

$$\begin{aligned} \pi_t(x_{0:t}) &= \frac{\gamma_t(x_{0:t})}{Z_t} \\ Z_t &= \int_{x_{0:t}} \gamma_t(x_{0:t}) \end{aligned} \tag{77}$$

For example, in the context of filtering we have that $\gamma_t(x_{0:t}) = p(x_{0:t}, y_{0:t})$ and $Z_t = p(y_{0:t})$ so that $\pi_t(x_{0:t}) = p(x_{0:t}|y_{0:t})$.

It is possible to approximate the distribution $\pi_t(x_{0:t})$ by drawing N samples $X^i_{0:t} \backsim \pi_t(x_{0:t})$ and using the Monte Carlo method to find the approximation $\hat{\pi}_t(x_{0:t})$ as shown in (78).

$$\hat{\pi}_t(x_{0:t}) = \frac{1}{N}\sum_{i=1}^{N} \delta(X^i_{0:t}, x_{0:t}) \tag{78}$$

We denote the Dirac Delta function of x with mass located at x_0 by $\delta(x_0, x)$. It is easy to approximate the marginal $\pi_t(x_t)$ as shown in (79).

$$\hat{\pi}_t(x_t) = \frac{1}{N}\sum_{i=1}^{N} \delta(X^i_t, x_t) \tag{79}$$

It can be shown that the variance of the approximation error of π_t decreases at rate $\mathcal{O}(\frac{1}{N})$. Unfortunately there are two significant drawbacks to the Monte Carlo approximation. The first is that often we cannot sample from $\pi_t(x_{0:t})$ directly and the second is that even if we could it is often computationally prohibitive.

We use the Importance Sampling method to address the first problem. We do this by introducing an importance (sometimes called proposal) density $q_t(x_{0:t})$ such that $\pi_t(x_{0:t}) > 0 \implies q_t(x_{0:t}) > 0$. By substituting this into the SMC framework (77) we have (80).

$$\begin{aligned} \pi_t(x_{0:t}) &= \frac{w_t(x_{0:t})q_t(x_{0:t})}{Z_t} \\ Z_t &= \int_{x_{0:t}} w_t(x_{0:t})q_t(x_{0:t}) \end{aligned} \tag{80}$$

Where we have defined the unnormalised weight function $w_t(x_{0:t}) = \frac{\gamma_t(x_{0:t})}{q_t(x_{0:t})}$. It is possible, for example, to set q_t to a multivariate Gaussian which is easy to sample from. By drawing N samples $X^i_{0:t} \backsim q_t(x_{0:t})$ and using (80) we have (81).

$$\begin{aligned} \hat{\pi}_t(x_{0:t}) &= \frac{1}{N}\sum_{i=1}^{N} W^i_t \delta(X^i_{0:t}, x_{0:t}) \\ \hat{Z}_t &= \frac{1}{N}\sum_{i=1}^{N} w_t(X^i_{0:t}) \\ W^i_t &= \frac{w_t(X^i_{0:t})}{\sum_{i=1}^{N} w_t(X^i_{0:t})} \end{aligned} \tag{81}$$

Now we will attempt to modify the Importance Sampling method to address the second problem of computational cost incurred by the sampling routine.

We do this by selecting an importance/proposal distribution which factorises according to $q_t(x_{0:t}) = q_{t-1}(x_{0:t-1})q_t(x_t|x_{0:t-1}) = q_1(x_0)\Pi_{k=2}^{t} q_k(x_k|x_{0:k-1})$. In this way we only need to sample sequentially at each time step: at time $t = 1$ we sample $X_1^i \backsim q_1(x_1)$, at time $t = 2$ we sample $X_2^i \backsim q_2(x_2|x_1)$ and so we build up $X_{0:t}^i \backsim q_t(x_{0:t})$ factor by factor.

The weights can be written in the form (82).

$$\begin{aligned} w_t(x_{0:t}) &= \frac{\gamma_t(x_{0:t})}{q_t(x_{0:t})} \\ &= \frac{\gamma_{t-1}(x_{0:t-1})}{q_{t-1}(x_{0:t-1})} \frac{\gamma_t(x_{0:t})}{\gamma_{t-1}(x_{0:t-1})q_t(x_t|x_{0:t-1})} \\ &= w_{t-1}(x_{0:t-1})\alpha_t(x_{0:t-1}) \\ &= w_1(x_1)\Pi_{k=2}^{t}\alpha_k(x_{0:k}) \end{aligned} \tag{82}$$

Thus, at any time t we can obtain the estimates $\hat{\pi}_t(x_{0:t})$ and Z_t. The major limitation of this approach is that the variance of the resulting estimates typically increases exponentially with t [14].

We overcome this problem by resampling and thus introduce the Sequential Importance Resampling (SIR) method. So far we have a set of weighted samples generated from $q_t(x_{0:t})$ which builds the approximation $\hat{\pi}_t(x_{0:t})$. However, sampling directly from $\hat{\pi}_t(x_{0:t})$ does not approximate $\pi_t(x_{0:t})$. To obtain an approximate distribution of $\pi_t(x_{0:t})$ we need to sample from the weighted distribution $\hat{\pi}_t(x_{0:t})$. This is called resampling because we are sampling from a sampled distribution. Many techniques exist to perform this step efficiently. The crudest and most widely used one is to simply use the discrete multinomial distribution based on $W_{0:t}^i$ to draw samples from $\hat{\pi}_t(x_{0:t})$ [14].

The benefit of resampling is that it allows us to remove particles with low weight and thus keeps the variance of the estimate in check. We are finally ready to consider the general SIR algorithm:

SIR Algorithm
For $t = 1$:

1. Sample $X_1^i \backsim q_1(x_1)$.
2. Compute the weights $w_1(X_1^i)$ and $W_1^i \propto w_1(X_1^i)$.
3. Resample (W_1^i, X_1^i) to obtain N equally weighted particles $(\frac{1}{N}, \bar{X}_1^i)$.

For $t \geq 2$:

1. Sample $X_t^i \backsim q_t(x_t|\bar{X}_{0:t-1}^i)$ and set $X_{0:t}^i \leftarrow (\bar{X}_{0:t-1}^i, X_t^i)$.
2. Compute the weights $\alpha_t(X_{0:t}^i)$ and $W_t^i \propto \alpha_t(X_{0:t}^i)$.

3. Resample $(W_t^i, X_{0:t}^i)$ to obtain N equally weighted particles $(\frac{1}{N}, \bar{X}_{0:t}^i)$.

At any time t we have two approximations for $\pi(x_{0:t})$ as shown in (83).

$$\begin{aligned} \hat{\pi}(x_{0:t}) &= \sum_{i=1}^{N} W_t^i \delta(X_{0:t}^i, x_{0:t}) \\ \bar{\pi}(x_{0:t}) &= \frac{1}{N} \sum_{i=1}^{N} \delta(\bar{X}_{0:t}^i, x_{0:t}) \end{aligned} \tag{83}$$

The latter approximation represents the resampled estimate and the former represents the sampled estimate [14]. We prefer the former because in the limit as $N \to \infty$ it is a better approximation of π_t. However, as we have mentioned the variance of $\hat{\pi}(x_{0:t})$ tends to be unbounded and thus we often have that most of the particles in the particle population have very low weight. From a computational point of view this is wasteful. To ameliorate this we use the latter, resampled, estimate. However, the problem with the resampled estimate is that it effectively culls low weight particles and this reduces the diversity of the particle population [35].

We attempt to get the benefit of both worlds by only performing resampling when the weight variance of the particles becomes large. The Effective Sample Size (ESS) is a method whereby one determines when to perform resampling according to (84).

$$\text{ESS} = \frac{1}{\sum_{i=1}^{N} (W_n^i)^2} \tag{84}$$

If the ESS becomes smaller than some threshold (typically $\frac{N}{2}$) we resample to cull low weight particles and replace them with high weight particles. In this manner we have a computationally feasible method. This is called adaptive resampling and is a straightforward extension of the SMC algorithm as shown below.

Adaptive SIR Algorithm
For $t = 1$:

1. Sample $X_1^i \backsim q_1(x_1)$.
2. Compute the weights $w_1(X_1^i)$ and $W_1^i \propto w_1(X_1^i)$.
3. If resample criterion is satisfied then resample (W_1^i, X_1^i) to obtain N equally weighted particles $(\frac{1}{N}, \bar{X}_1^i)$ and set $(\bar{W}_1^i, \bar{X}_1^i) \leftarrow (\frac{1}{N}, \bar{X}_1^i)$ otherwise set $(\bar{W}_1^i, \bar{X}_1^i) \leftarrow (W_1^i, X_1^i)$.

For $t \geq 2$:

1. Sample $X_t^i \backsim q_t(x_t | \bar{X}_{0:t-1}^i)$ and set $X_{0:t}^i \leftarrow (\bar{X}_{0:t-1}^i, X_t^i)$.
2. Compute the weights $\alpha_t(X_{0:t}^i)$ and $W_t^i \propto \bar{W}_{t-1}^i \alpha_t(X_{0:t}^i)$.
3. If the resample criterion is satisfied then resample $(W_t^i, X_{0:t}^i)$ to obtain N equally weighted particles $(\frac{1}{N}, \bar{X}_{0:t}^i)$ and set $(\bar{W}_1^i, \bar{X}_t^i) \leftarrow (\frac{1}{N}, \bar{X}_t^i)$ otherwise set $(\bar{W}_t^i, \bar{X}_t^i) \leftarrow (W_t^i, X_t^i)$.

Numerous convergence results exist for the SMC methods we have discussed but the fundamental problem with this scheme is that of sample impoverishment. It is fundamentally impossibly to accurately represent a distribution on a space of arbitrarily high dimension with a finite set of samples [14]. We attempt to mitigate this problem by using resampling but degeneracy inevitably occurs for large enough t. Fortunately, for our purposes we will not be dealing with arbitrarily large dimensional problems because of the Markov assumption.

6.2 Particle Filter

We now apply the adaptive SIR algorithm in the setting of filtering. We set $\pi_t(x_{0:t}) = p(x_{0:t}|y_{0:t})$, $\gamma_t(x_{0:t}) = p(x_{0:t}, y_{0:t})$ and consequently $Z_t = p(y_{0:t})$. We use the recursive proposal distribution $q_t(x_{0:t}|y_{0:t}) = q(x_t|x_{0:t-1}, y_{0:t})q_{t-1}(x_{0:t-1}|y_{0:t-1})$. We then have the unnormalised weights as shown in (85).

$$\begin{aligned} w_t(x_{0:t}) &= \frac{\gamma_t(x_{0:t})}{q_t(x_{0:t}|y_{0:t})} \\ &= \frac{p(x_{0:t}, y_{0:t})}{q_t(x_{0:t}|y_{0:t})} \\ &\propto \frac{p(x_{0:t}|y_{0:t})}{q_t(x_{0:t}|y_{0:t})} \\ &\propto \frac{p(y_t|x_t)p(x_t|x_{t-1})}{q_t(x_t|x_{0:t-1}, y_{0:t})} \frac{p(x_{0:t-1}|y_{0:t-1})}{q_{t-1}(x_{0:t-1}|y_{0:t-1})} \\ &= \alpha_t(x_{0:t})w_{t-1}(x_{0:t-1}) \end{aligned} \tag{85}$$

For filtering we only care about $p(x_t|y_{0:t})$ and thus we do not need the entire trajectory $x_{0:t}$. This allows us to choose the proposal distribution $q_t(x_t|x_{0:t-1}, y_{0:t}) = q_t(x_t|x_{t-1}y_t)$. In this case the incremental weight α_t simplifies to (86).

$$\alpha_t(x_{0:t}) = \frac{p(y_t|x_t)p(x_t|x_{t-1})}{q_t(x_t|x_{t-1}, y_t)} \tag{86}$$

The most common proposal distribution is, the suboptimal, $q_t(x_t|x_{t-1}|y_t) = p(x_t|x_{t-1})$ because it is easy to sample from. This implies that the incremental weights simplify to $\alpha_t(x_{0:t}) = p(y_t|x_t)$. Using such a proposal distribution was initially proposed in [21] in the setting of the non-adaptive SIR method.

For general purpose filtering this is not very efficient because it amounts to "guessing until you hit". If the transitions are very stochastic inference can be improved by using the optimal proposal distribution $q_t(x_t|x_{t-1}, y_t) = p(x_t|x_{t-1}, y_t)$. While this is optimal it introduces some difficulty because, in general, it is more difficult to sample from. The focus of dissertation is not on optimal filtering and for the purposes of prediction the suggested proposal distribution is sufficiently good [35]. We thus restrict ourselves to the proposal distribution $p(x_t|x_{t-1})$ for simplicity.

Finally, we have mentioned that resampling kills off unlikely particles. An unfortunate consequence of this is that some particle diversity is lost. An empirical method used to attenuate this problem is to resample from a kernel around the particle selected by the resampling process. This is called roughening in [21]. We thus make a final modification to the adaptive SIR algorithm. We select a particle from the population in the standard way but resample from a Normal distribution centred around that particle and with a diagonal covariance matrix where the standard deviation of each diagonal is $KEN^{-\frac{1}{d}}$. We define E as the range of the particle's relevant component, N as the number of particles and d as the dimension of the problem. K is a tuning factor which specifies how broad the kernel we sample from should be.

For the sake of completeness we present the Particle Filter algorithm we used here. Recall that f and g are the transition and observation functions respectively. The algorithm is applied to each particle $i = 1, 2, ..., N$.

Particle Filter Algorithm

For $t = 1$:

1. Sample $X_1^i \backsim p(x_1)$.

2. Compute the weights $w_1(X_1^i) = p(Y_1^*|X_1^i) = \mathcal{N}(Y_1^*|g(X_1^i), \text{covar}[v_1])$ where Y_1^* is the observation. Normalise $W_1^i \propto w_1(X_1^i)$.

3. If the number of effective particles is below some threshold apply resampling with roughening (W_1^i, X_1^i) to obtain N equally weighted particles $(\frac{1}{N}, \bar{X}_1^i)$ and set $(\bar{W}_1^i, \bar{X}_1^i) \leftarrow (\frac{1}{N}, \bar{X}_1^i)$ otherwise set $(\bar{W}_1^i, \bar{X}_1^i) \leftarrow (W_1^i, X_1^i)$

For $t \geq 2$:

1. Sample $X_t^i = f(\bar{X}_{t-1}^i, w_t) \backsim p(x_t|\bar{X}_{t-1}^i)$.

2. Compute the weights $\alpha_t(X_t^i) = p(Y_t^*|X_t^i) = \mathcal{N}(Y_t^*|g(X_t^i), \text{covar}[v_t])$ and normalise $W_t^i \propto \bar{W}_{t-1}^i \alpha_t(X_t^i)$.

3. If the number of effective particles is below some threshold apply resampling with roughening (W_t^i, X_t^i) to obtain N equally weighted particles $(\frac{1}{N}, \bar{X}_t^i)$ and set $(\bar{W}_1^i, \bar{X}_t^i) \leftarrow (\frac{1}{N}, \bar{X}_t^i)$ otherwise set $(\bar{W}_t^i, \bar{X}_t^i) \leftarrow (W_t^i, X_t^i)$.

The algorithm presented above is a slight generalisation of the bootstrap Particle Filter as initially proposed by Gordon et. al. [21].

Intuitively the algorithm may be summarised like this. Particle Filters predict the next hidden state by projecting all the current particles forward using the transition function. For each particle the likelihood of the observation is calculated given the particle and measurement noise. This likelihood is related to the weight of each particle. Particles with a relatively high weight are then deemed to more accurately represent the posterior distribution and thus we infer the posterior state estimate based on the relative weights of each particle.

6.3 Particle Prediction

We are primarily interested in predicting the future hidden states but we also show how the future visible states may be predicted within the framework of Particle methods. Recalling the prediction derivations of the Hidden Markov Model section and the Linear Models section we expect the hidden state prediction to merely be an n step ahead projection of the current filtered particles. Likewise, we expect the visible state prediction to just be transformation of the predicted hidden states under the emission function.

Inspecting the bootstrap Particle Filter algorithm presented in the previous subsection we are relieved to find that this is the case. One just removes the observation update steps (steps 2 and 3) from the algorithm because we cannot observe the future. We illustrate the two step ahead predictions and trust that the reader understand what we mean.

Particle Prediction Algorithm

1. Sample $X^i_{t+1} = f(\bar{X}^i_t, w_{t+1}) \backsim p(x^i_{t+1}|y_t, \bar{X}^i_t)$
2. Project $X^i_{t+2} = f(X^i_{t+1}, w_{t+2}) \backsim p(x^i_{t+2}|y_t, X^i_{t:t+1})$
3. Project $Y^i_{t+2} = g(X^i_{t+2}, v_{t+2}) \backsim p(y^i_{t+2}|y_t, X^i_{t:t+1})$

6.4 Smoothing and Viterbi Decoding

In the context of nonlinear transition and emission functions smoothing and Viterbi decoding are much more difficult than before. For the purposes of this dissertation it is not important to consider inferences of that type and thus we merely refer the reader to literature where this is discussed [14][22][35][36][3].

6.5 Filtering the CSTR

In this section we apply the Particle Filter to the nonlinear CSTR problem. We first demonstrate the effectiveness of the Particle Filter by performing inference using the full nonlinear CSTR model measuring only temperature. Next we use the full nonlinear model again but measure both temperature and concentration. Finally, we apply the Particle Filter to the CSTR using the linear model and compare it to the Kalman Filter.

Although it is not necessary we assume that the process and measurement noise is Gaussian with the same distributions as in the previous section. Unless otherwise noted we use the same parameters (e.g. $h = 0.1$) as before. We have used 100 particles to represent the state posterior. In Figure 29 we see the state estimates as a function of time.

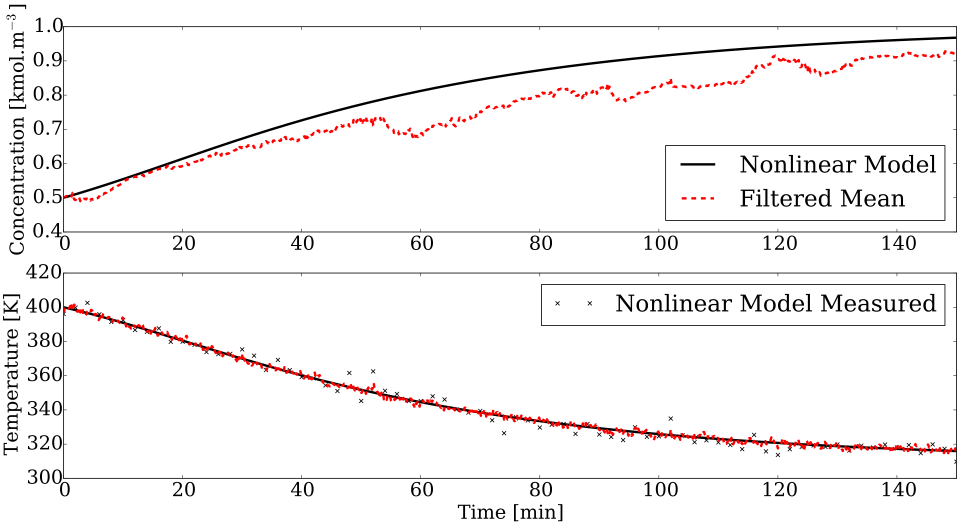

Figure 29: Time series state estimates using the Particle Filter on the nonlinear CSTR model with initial condition $(0.5, 400)$ and measuring only temperature.

The filter tracks both states reasonable well with a little more variance evident in the unmeasured state. The benefit of using the full nonlinear model is evident here - since the model is more accurate than the previously used linear model the filter infers the concentration more accurately. This is also reflected in the state space evolution curve in Figure 30.

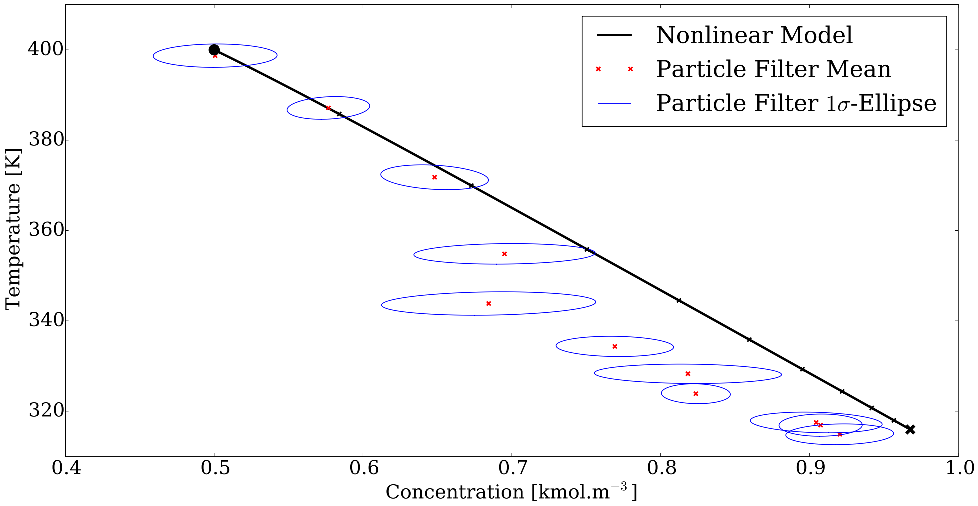

Figure 30: State space evolution of the Particle Filter on the nonlinear CSTR model with initial condition $(0.5, 450)$ and measuring only temperature.

We also see in Figure 30 that the variance of the estimates is quite high (the ellipses are quite big). We expect that by also measuring concentration this will decrease. In Figures 31 and 32 we incorporate a concentration measurement to aid inference.

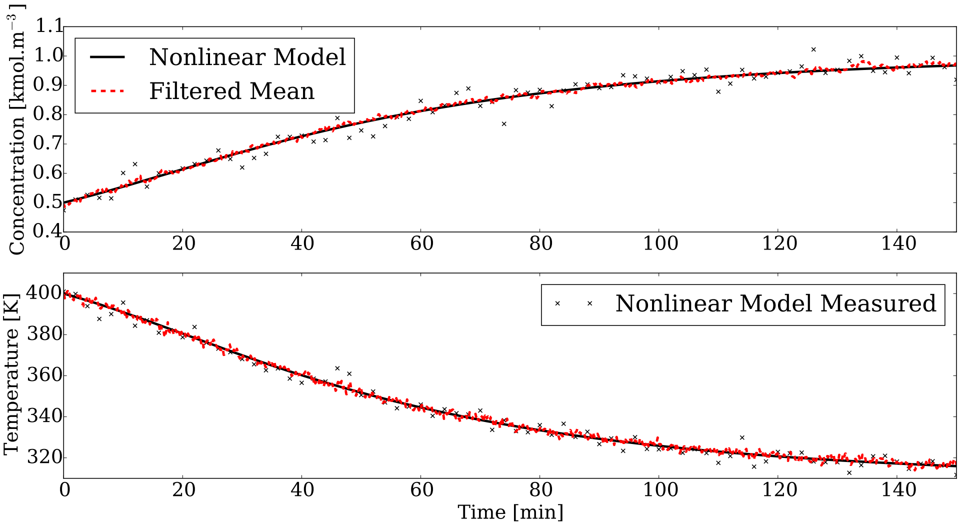

Figure 31: Time series state estimates using the Particle Filter on the nonlinear CSTR model with initial condition $(0.5, 450)$ and measuring both states.

It is clear that that the Particle Filter reliably tracks the state evolution in the presence of plant and measurement noise. We see that by also measuring the concentration the size of the error ellipses decrease in Figure 32.

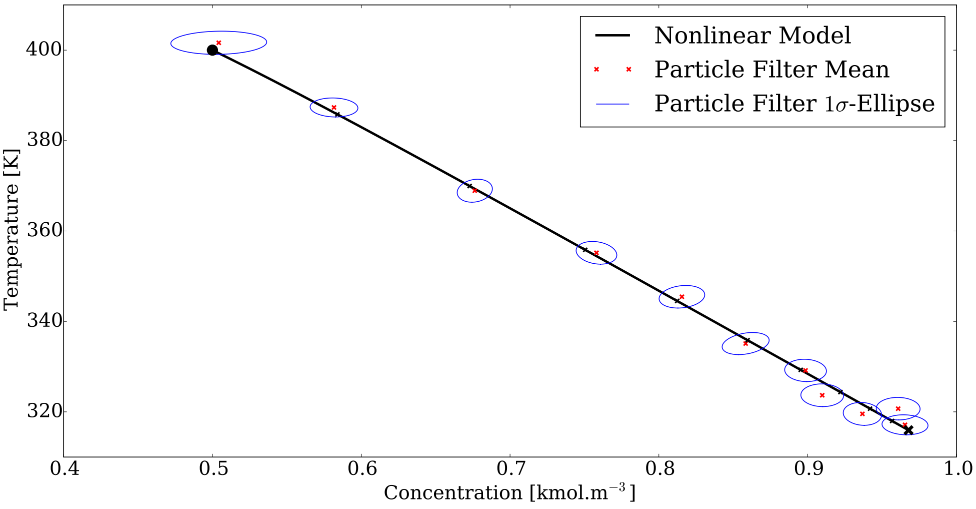

Figure 32: State space evolution of the Particle Filter on the nonlinear CSTR model with initial condition $(0.5, 450)$ and measuring both states.

Finally we compare the Particle Filter to the Kalman filter, using both temperature and concentration measurements and the linear model linearised around the unstable operating point. In Figure 33 we see that both the Particle Filter and the Kalman Filter are able to accurately estimate the posterior state distribution. Note that we have used 500 particles to meaningfully compare the distribution estimates (the Particle Filter should converge to the

Kalman Filter as the number of particles get big).

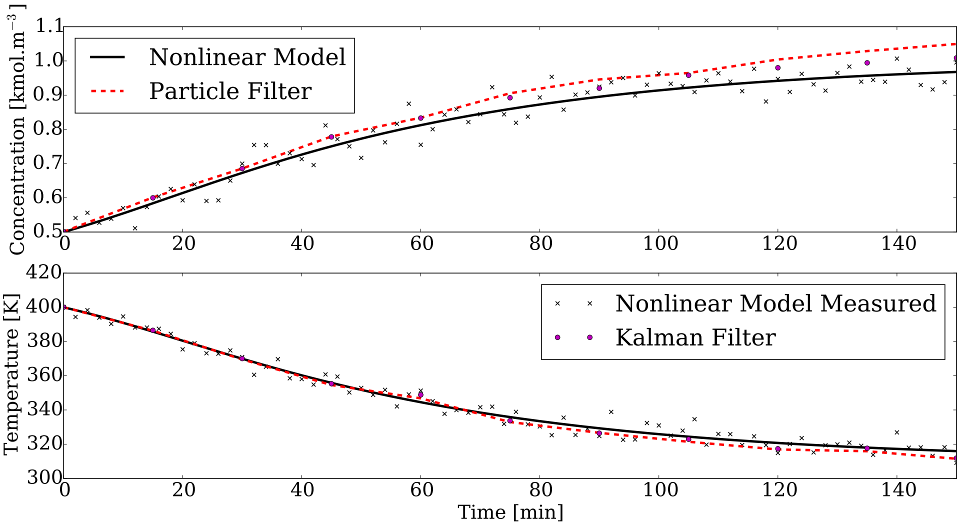

Figure 33: Time series state estimates using the Particle Filter and the Kalman Filter on the linear CSTR model (around the unstable operating point) with initial condition $(0.5, 400)$ and measuring both temperature and concentration.

In Figure 34 we see empirical proof of the assertion that the Kalman Filter is the optimum state estimator for conditionally linear Gaussian systems [1]: the 1σ ellipses for the Kalman Filter are smaller than the corresponding ellipses for the Particle Filter and they track the true states better than the Particle Filter.

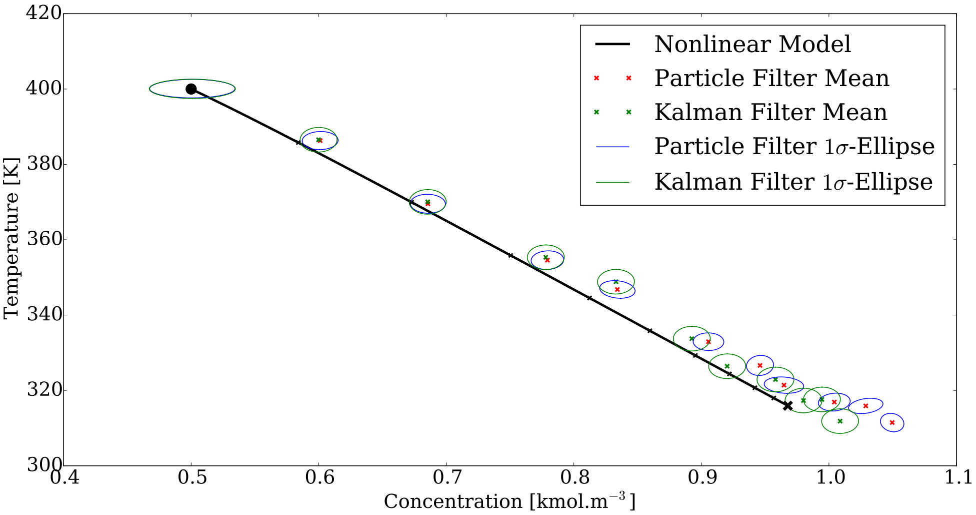

Figure 34: State space evolution of the Particle Filter and the Kalman Filter on the linear CSTR model (around the unstable operating point) with initial condition $(0.5, 400)$ and measuring both temperature and concentration.

It is clear that if one only has access to linear models it makes sense from both a computational and accuracy point of view to use the Kalman Filter. However, if one needs to perform inference on a system better descibed by a nonlinear model the Particle Filter becomes more useful.

7 Stochastic Linear Control

In this section we consider the stochastic reference tracking problem. It is required to move the states and manipulated variables of the system, shown in (87), to the set point (x_{sp}, u_{sp}) by manipulating the input variables u.

$$\begin{aligned} x_{t+1} &= f(x_t, u_t) + w_{t+1} \\ y_{t+1} &= g(x_{t+1}) + v_{t+1} \end{aligned} \tag{87}$$

We assume uncorrelated zero mean additive Gaussian noise in both the state function f and the observation function g with known covariances W and V respectively. Clearly it is not possible to achieve perfect control (zero offset at steady state) because of the noise terms, specifically w_t. For this reason we need to relax the set point goal a little bit. We will be content if our controller is able to achieve Definition 7.1.

Definition 7.1. Stochastic Reference Tracking Goal: Suppose we have designed a controller and set $\delta > 0$ as a controller benchmark. If there exists some positive number $t^* < \infty$ such that $\forall\ t > t^*$ the controller input causes $\mathbb{E}[(x_t - x_{sp})^T Q(x_t - x_{sp}) + (u_t - u_{sp})^T R(u_t - u_{sp})] < \delta$ we will have satisfied the Stochastic Reference Tracking Goal given δ.

In this section we limit ourselves by only considering controllers developed using a single linear model of the underlying, possibly nonlinear, system functions f and g. The linearised model control is based upon is shown in (88) and is subject to the same noise as (87).

$$\begin{aligned} x_{t+1} &= Ax_t + Bu_t + w_{t+1} \\ y_{t+1} &= Cx_{t+1} + v_{t+1} \end{aligned} \tag{88}$$

We will endeavour to develop predictive controllers using the Graphical Models of Section 5 and 6.

7.1 Current Literature

Linear unconstrained stochastic control subject to white additive Gaussian noise is well studied in literature. The Linear Quadratic Gaussian (LQG) controller introduced in Section 2.4 is one of the most fundamental results in Optimal Control Theory [10]. A significant drawback of the LQG controller, and by extension the LQR controller, is that it is inherently unconstrained.

Conventional deterministic MPC is very well studied in literature [38] and can be seen as the constrained generalisation of the LQR controller. A further generalisation of deterministic MPC is stochastic MPC whereby either the variables, the constraints or both have stochastic elements. In current literature the trend is to convert all the stochastic elements of the control problem into deterministic ones. This usually makes the problem somewhat more tractable from a computational point of view.

This conversion is usually achieved via two distinct approaches. In the first approach, which is also the one we employ, the probability distributions are assumed Gaussian and the systems linear. This allows one to greatly simplify the problem at the cost of those relatively strong assumptions. The second approach is to use a particle/sampling approach. Here the distributions are approximated by particles (much like the Particle Filter of Section 6) and no assumptions are made of form of the distributions. It is also not necessary to assume linear dynamics. The major practical drawback of this approach is that it can quickly become computationally intractable.

Indeed, this is the approach taken by [6]. They attempt to solve the stochastic MPC problem with stochastic (chance) constraints and variables by approximating the current and predicted distributions with particles. Referring back to Section 6.3 we see that this is indeed possible. By using the particle approach to model distributions it is possible to convert the resultant stochastic optimisation problem into a deterministic one. In the case where linear dynamics are used this becomes a Mixed Integer Linear or Quadratic Programming problem depending on the objective function. Their algorithm is appealing because it is not necessary to assume Gaussian distributions. However, it is possible that the algorithm could become computationally intractable due to the integer constraints which are used to approximate the chance constraints: it is necessary to include an integer variable for each particle at each time step in the prediction horizon. For large problems with long prediction horizons this can be problematic.

The approach taken by [29] is related to the sampling approach. They convert the stochastic chance constrained optimisation problem into a deterministic nonlinear optimisation problem. They then use a simulation approach to ensure that the chance constraints are satisfied. The approach taken by [4] uses a randomized optimisation algorithm in concert with the empirical mean of the variables. A penalty method is used to enforce constraints. The latter paper takes a step in the direction of simplifying the problem without sampling.

In the paper [28] the stochastic variables are assumed to be Gaussian and the stochastic optimisation problem is transformed into a nonlinear optimisation problem. Using the Gaussian assumption they are able to ensure feasibility and constraint satisfaction albeit conservatively. In [32] the feasibility of stochastically constrained predictive control is considered. An algorithm enforcing joint chance constraints and recursive feasibility is discussed using a risk allocation approach.

While [40] mainly concerns stochastic parameters in the optimisation problem it is shown that chance constraints can, in theory, be rewritten as deterministic constraints if the probability distributions are known and affine constraints are used. In [43] and [42] an ellipsoidal approximation is used to ensure constraint satisfaction. Using the ellipsoidal approximations the stochastic optimisation problem can transformed into a second order conic optimisation problem. The approach of using confidence ellipsoids is refined in [7].

Although [45] and [46] primarily deal with a univariate problem they show that if the underlying system is linear and Gaussian it is possible to manipulate the constrained stochastic problem into a deterministic problem without any approximations. Their analysis allows the stochastic objective function and constraints to be rewritten as deterministic expressions using the properties of conditionally linear Gaussian distributions.

In this work we show that by starting the MPC design from the graphical model shown in Figure 28 it is possible to easily and intuitively re-derive the LQG controller. During that derivation we naturally arrive at the conclusions reached by [45] and [46]. Next we generalise our method to incorporate stochastic constraints. We use a method related to the ellipsoidal approximation technique to handle the stochastic constraints. We show that our method replaces the chance constraints with linear constraints. This is highly desirable because it allows one to rewrite the full stochastic MPC problem as a conventional deterministic Quadratic Programming (QP) MPC problem. All of these results are underpinned by the realisation that Graphical Models, as used in the field of Machine Learning, are closely related to predictive control.

7.2 Unconstrained Stochastic Control

Our goal is to solve the problem in (89) given the current state estimate x_0. If the system is controllable and linear the Stochastic Reference Tracking Goal will be satisfied.

$$\begin{aligned}&\min_{\mathbf{u}} J_{LQG}(x_0, \mathbf{u}) = \mathbb{E}\left[\frac{1}{2}\sum_{k=0}^{N-1}\left(x_k^T Q x_k + u_k^T R u_k\right) + \frac{1}{2}x_N^T P_f x_N\right]\\ &\text{subject to } x_{t+1} = Ax_t + Bu_t + w_t\\ &\text{and } y_t = Cx_t + v_t\end{aligned} \tag{89}$$

Note that the future inputs $\mathbf{u} = (u_0, u_1, ..., u_{T-1})$ are denoted in boldface to emphasise that it could be a vector of vectors. Inspecting (89) we see that this is none other than the LQG control problem of Section 2.4.3. Therefore we know what the optimal solution should look like.

We start out analysis using the results of Section 5. We immediately realise that the optimal linear state estimator is the Kalman Filter. We assume that at every sequential time step we have the current state estimate, supplied by the Kalman Filter, and denote this by x_0. Since we are using the Kalman Filter the mean and covariance of the state estimate is well defined.

Inspecting **insert GM pic** we note that the state prediction equations derived in Subsection 5.2 are applicable. Thus we can predict the state distributions given the future inputs $\mathbf{u}$.

Before we proceed we prove a very intuitive result in Theorem 7.1. We will use this to link the predictive controller derived using the results of Section 5 to the LQR controller derived in Section 2.4.1. We provide two proofs, the second of which is more general than the first.

Theorem 7.1. Optimisation Equivalence Suppose we have two real valued objective functions $f(x_0, \mathbf{u})$ and $g(x_0, \mathbf{u})$ and we are required to minimise them with respect to $\mathbf{u}$ over the same space where they are both defined: $\mathbf{u} \in \mathcal{U}$ and $x_0 \in \mathcal{X}$. Furthermore, suppose there exists a real number $k \geq 0$ such that $\forall \mathbf{u} \in \mathcal{U}$ we have that $g(x_0, \mathbf{u}) + k = f(x_0, \mathbf{u})$. Finally, assume the existence and uniqueness of the global minimiser for each problem. Then the global minimiser $\mathbf{u}^*$ of $g(x_0, \mathbf{u})$ also minimises $f(x_0, \mathbf{u})$.

Proof. This proof only holds over functions which are at least twice differentiable and convex. By assumption we know that $\mathbf{u}^*$ is the minimiser of $g(x_0, \mathbf{u})$ given x_0. By the necessary conditions for optimality [19] we know that $\nabla g(x_0, \mathbf{u}^*) = 0$ and that $\nabla^2 g(x_0, \mathbf{u}^*)$ is positive semi-definite. Since f and g are both twice differentiable and $g(x_0, \mathbf{u}^*) + k = f(x_0, \mathbf{u}^*)$ it must hold that $\nabla g(x_0, \mathbf{u}^*) = \nabla f(x_0, \mathbf{u}^*)$ and $\nabla^2 g(x_0, \mathbf{u}^*) = \nabla^2 f(x_0, \mathbf{u}^*)$. Since $\nabla^2 g(x_0, \mathbf{u}^*)$ is positive semi-definite it must be that $\nabla^2 f(x_0, \mathbf{u}^*)$ is also positive semi-definite. Therefore $\mathbf{u}^*$ is necessarily a minimum of f. Since f is convex the minimum must also be a global minimum. □

Proof. This proof hold over differentiable and non-differentiable objective functions which are not necessarily convex. Suppose not i.e. there exists $\mathbf{u}_g \in \mathcal{U}$ such that $g(x_0, \mathbf{u}_g) < g(x_0, \mathbf{u})$ $\forall \mathbf{u} \in \mathcal{U}$ but $f(x_0, \mathbf{u}_g) \not< f(x_0, \mathbf{u})$ $\forall \mathbf{u} \in \mathcal{U}$. This implies that for $\mathbf{u}_f \in \mathcal{U}$ the global minimiser of f we have $f(x_0, \mathbf{u}_f) \leq f(x_0, \mathbf{u}_g)$.

Consider the case where $f(x_0, \mathbf{u}_f) = f(x_0, \mathbf{u}_g)$. This implies that both $\mathbf{u}_f$ and $\mathbf{u}_g$ are global minimisers of f and contradicts our assumption that the global minimiser is unique.

Consider the case where $f(x_0, \mathbf{u}_f) < f(x_0, \mathbf{u}_g)$. Since $g(x_0, \mathbf{u}) + k = f(x_0, \mathbf{u})$ $\forall \mathbf{u} \in \mathcal{U}$ this implies that $g(x_0, \mathbf{u}_f) < g(x_0, \mathbf{u}_g)$. But this contradicts our assumption that $\mathbf{u}_g$ is the global minimiser of g.

It must then hold that $f(x_0, \mathbf{u}_g) < f(x_0, \mathbf{u})$ $\forall \mathbf{u} \in \mathcal{U}$. Therefore the global minimiser $\mathbf{u}_g$ of $g(x_0, \mathbf{u})$ also minimises $f(x_0, \mathbf{u})$. □

Now we are in a position to show the equivalence between the LQR control problem and the LQG control problem using the results of Section 5. Theorem 7.2 shows how this is possible. It is quite reassuring to note that by starting within the framework of Graphical Models we arrive at the most important contribution of [45] and [46] in an intuitively simple manner.

Theorem 7.2. LQR and LQG Objective Function Difference Consider the LQR and LQG Objective Functions in (90) and (91) respectively.

$$J_{LQR}(x_0, \mathbf{u}) = \frac{1}{2}\sum_{k=0}^{N-1}\left(x_k^T Q x_k + u_k^T R u_k\right) + \frac{1}{2}x_N^T P_f x_N \tag{90}$$

with $x_{t+1} = Ax_t + Bu_t$

$$J_{LQG}(x_0, \mathbf{u}) = \mathbb{E}\left[\frac{1}{2}\sum_{k=0}^{N-1}\left(x_k^T Q x_k + u_k^T R u_k\right) + \frac{1}{2}x_N^T P_f x_N\right] \tag{91}$$

with $x_{t+1} = Ax_t + Bu_t + w_{t+1}$

Suppose x_0 is the state estimate supplied by the Kalman Filter given the latest observation in the stochastic case. In the deterministic case we have that $x_0 = \mathbb{E}[x_0] = \mu_0$ because we exactly observe the state. Given any input sequence $\mathbf{u} \in \mathcal{U}$, where $\mathcal{U}$ is the shared admissible input space, we have that $J_{LQR}(x_0, \mathbf{u}) + \frac{1}{2}\sum_{k=0}^{N} \text{tr}(Q\Sigma_k) = J_{LQG}(x_0, \mathbf{u})$ where $\Sigma_{t+1} = W + A\Sigma_t A^T$ and Σ_0 is the covariance matrix of the current state given by the Kalman Filter.

Proof. Expanding the LQG objective function and noting that $\mathbf{u}$ is deterministic we have (92). Note that the conditional expectations in the expansion originate from the graphical model in Figure 23 (due to the first order Markov assumption).

$$\begin{aligned} J_{LQG}(x_0, \mathbf{u}) &= \frac{1}{2}\mathbb{E}\left[x_0^T Q x_0 + u_0^T R u_0\right] + \frac{1}{2}\mathbb{E}\left[x_1^T Q x_1 + u_1^T R u_1 | x_0\right] + ... \\ &+ \frac{1}{2}\mathbb{E}\left[x_{N-1}^T Q x_{N-1} + u_{N-1}^T R u_{N-1} | x_{N-2}\right] + \frac{1}{2}\mathbb{E}\left[x_N^T P_f x_N | x_{N-1}\right] \\ &= \frac{1}{2}\mathbb{E}\left[x_0^T Q x_0\right] + \frac{1}{2}u_0^T R u_0 + \frac{1}{2}\mathbb{E}\left[x_1^T Q x_1 | x_0\right] + \frac{1}{2}u_1^T R u_1 + ... \\ &+ \frac{1}{2}\mathbb{E}\left[x_{N-1}^T Q x_{N-1} | x_{N-2}\right] + \frac{1}{2}u_{N-1}^T R u_{N-1} + \frac{1}{2}\mathbb{E}\left[x_N^T P_f x_N | x_{N-1}\right] \end{aligned} \tag{92}$$

We know that $x_0 \sim \mathcal{N}(\mu_0, \Sigma_0)$ because the current state estimate comes from the Kalman Filter. This means that we can evaluate the first expected value in (92) using Theorem 2.5 as shown in (93).

$$\mathbb{E}\left[x_0^T Q x_0\right] = \text{tr}(Q\Sigma_0) + \mu_0^T Q \mu_0 \tag{93}$$

Now we turn out attention to the second expected value in (92). First note that because we have x_0 and $\mathbf{u}$ we can use the result from Section 5.2 to predict (optimally) the distribution of x_1. Therefore we know that $x_1 \sim \mathcal{N}(A\mu_0 + Bu_0, W + A\Sigma_0 A^T)$. Now we let $\mu_1 = A\mu_0 + Bu_0$ and $\Sigma_0 = W + A\Sigma_0 A^T$. Then by using Theorem 2.5 as before we have (94).

$$\mathbb{E}\left[x_1^T Q x_1 | x_0\right] = \text{tr}(Q\Sigma_1) + \mu_1^T Q \mu_1 \tag{94}$$

Note that $\text{tr}(Q\Sigma_1)$ does not depend on u_0 but only on the initial state estimate x_0 which is independent of the future inputs $\mathbf{u}$. Notice that we can continue in this manner to simplify the LQG objective function to (95).

$$\begin{aligned} &J_{LQG}(x_0, \mathbf{u}) = \frac{1}{2}\sum_{k=0}^{N-1}\left(\mu_k^T Q \mu_k + u_k^T R u_k\right) + \frac{1}{2}\mu_N^T P_f \mu_N + \frac{1}{2}\sum_{k=0}^{N}\text{tr}(Q\Sigma_k) \\ &\text{with } \mu_{t+1} = A\mu_t + Bu_t \\ &\text{and } \Sigma_{t+1} = W + A\Sigma_t A^T \end{aligned} \tag{95}$$

Now note that except for the last term $J_{LQG}(x_0, \mathbf{u})$ is exactly the same as $J_{LQR}(x_0, \mathbf{u})$. The conclusion follows because $\frac{1}{2}\sum_{k=0}^{N} \text{tr}(Q\Sigma_k)$ is independent of $\mathbf{u}$. □

Finally we combine Theorem 7.1 and 7.2 to produce Theorem 7.3 which is the main result of this subsection.

Theorem 7.3. Solution of the Finite Horizon LQG control problem We wish to solve the LQG control problem within the framework of Graphical Models. The full problem is shown in (96).

$$\begin{aligned}
&\min_{\mathbf{u}} V(x_0, \mathbf{u}) = \mathbb{E}\left[\frac{1}{2}\sum_{k=0}^{N-1}\left(x_k^T Q x_k + u_k^T R u_k\right) + \frac{1}{2}x_N^T P_f x_N\right] \\
&\text{subject to } x_{t+1} = Ax_t + Bu_t + w_t \\
&\text{and } y_t = Cx_t + v_t
\end{aligned} \tag{96}$$

We assume that we have the Kalman Filter state estimate for x_0. We use Theorem 7.2 to prove that given x_0 and $\forall \mathbf{u} \in \mathcal{U}$ we have that $J_{LQR}(x_0, \mathbf{u}) + \frac{1}{2}\sum_{k=0}^{N} \operatorname{tr}(Q\Sigma_k) = J_{LQG}(x_0, \mathbf{u})$ with $\frac{1}{2}\sum_{k=0}^{N} \operatorname{tr}(Q\Sigma_k) \in \mathbb{R}$ a constant depending only on x_0. Thus we can use Theorem 7.1 to prove that we only need to solve for the optimal controller input $\mathbf{u}^0$ using the LQR objective function. Thus we can use Theorem 2.11 to find $\mathbf{u}$.

As we have mentioned before, the Separation Theorem implies that the solution of the LQG control problem is achieved by using the Kalman Filter to optimally estimate the current state and then using that state estimate in the optimal LQR controller. It is reassuring that Theorem 7.3 is confirmed by this result. The primary benefit of the Graphical Model approach is clear: we have solved the LQG problem without resorting to Stochastic Dynamical Programming.

Under some circumstances it is also possible to extend the result of Theorem 7.3 to the infinite horizon case as shown in Theorem 7.4.

Theorem 7.4. Solution of the Infinite Horizon LQG control problem If the linear model of (88) is stable then, using, with some minor adjustments, Theorems 7.1 and 7.2 it is possible to show that the infinite horizon LQG problem is solved in a similar manner: the Kalman Filter state estimate is used in conjunction with the infinite horizon LQR solution. This result can also be obtained by using the Separation Theorem.

To clarify why it is important that the linear system, i.e. the matrix A, is stable, consider the quantity $\frac{1}{2}\sum_{k=0}^{N} \operatorname{tr}(Q\Sigma_k)$. If it is unbounded the optimisation minimum will tend to infinity. Inspecting $\Sigma_{t+1} = W + A\Sigma_t A^T$ we see that $||\Sigma_\infty||$ will be unbounded if $||A\Sigma_t A^T||$ becomes unbounded (W is a constant). Note that $||\cdot||$ is some matrix norm. It can be shown that if the eigenvalues of A are less than unity i.e. the linear model is stable, then $||A\Sigma_{t+1}A^T|| \leq ||A\Sigma_{t+1}A^T||$ which implies that $\frac{1}{2}\sum_{k=0}^{N} \operatorname{tr}(Q\Sigma_k)$ is bounded and the optimisation is reasonable.

7.3 Constrained Stochastic Control

The goal of this section is to solve the stochastic constrained optimisation problem shown in (97). We assume that the underlying system is linear and the probability distributions are

Gaussian[5]. We also restrict our analysis to affine constraints.

$$\begin{aligned}
&\min_{\mathbf{u}} \mathbb{E}\left[\frac{1}{2}\sum_{k=0}^{N-1}\left(x_k^T Q x_k + u_k^T R u_k\right) + \frac{1}{2}x_N^T P_f x_N\right] \\
&\text{subject to } x_{t+1} = Ax_t + Bu_t + w_t \\
&\text{and } y_t = Cx_t + v_t \\
&\text{and } \mathbb{E}[d^T x_t + e] \geq 0 \ \forall\ t = 1, ..., N \\
&\text{and } \Pr(d^T x_t + e \geq 0) \geq p \ \forall\ t = 1, ..., N
\end{aligned} \tag{97}$$

It might seem that the last constraint is a duplicate of the preceding one. Closer inspection reveals their different character. The first inequality constraint requires that the predicted states satisfy the constraint "on average" while the second inequality constraint requires that the predicted states jointly satisfy the constraint with at least some probability p. The reason both constraints are required will become clear later.

Theorem 7.5 succinctly shows that it is simple to convert the first stochastic constraint in (97) to a linear deterministic constraint.

Theorem 7.5. Affine Expected Value Constraints Suppose we have a stochastic variable x with a known Gaussian distribution. Then the stochastic constraint $\mathbb{E}[d^T x + e] \geq 0$ simplifies to the deterministic constraint $d^T \mu + e \geq 0$ where $\mathbb{E}[x] = \mu$ is the mean of the stochastic variable.

Proof. We know that x is a Gaussian stochastic variable. By the results of Section 2.5 we know that $\mathbb{E}[d^T x + e] = d^T \mu + e$. This immediately implies the Theorem. □

Theorem 7.6 is a necessary step before we can convert the last stochastic inequality constraint of (97) into a nonlinear deterministic constraint.

Theorem 7.6. Shortest Squared Mahalanobis Distance between a Hyperplane and a Point Suppose we are given a symmetric positive semi-definite matrix S and a point y. The shortest squared Mahalanobis Distance between y and the hyperplane $b^T x + c = 0$ is given by $\frac{(b^T y + c)^2}{b^T S b}$.

Proof. It is natural to formulate Theorem 7.6 as an optimisation problem as shown in (98).

$$\begin{aligned}
&\min_{x} \ (x - y)^T S^{-1} (x - y) \\
&\text{subject to } b^T x + c = 0
\end{aligned} \tag{98}$$

Note that S and therefore also S^{-1} is symmetric. Using conventional calculus we have $\nabla f(x) = (S^{-1} + S^{-1^T})x - 2S^{-1}y = 2S^{-1}x - 2S^{-1}y$ and $\nabla g(x) = b^T$. Using the method of

[5] From the results of Section 5 the probability distributions will be Gaussian if the system dynamics are linear. However, it is well known [31] that MPC is not in general a linear controller. From an analytical point of view this is problematic. We assume that the nonlinearity introduced by the MPC is negligible.

Lagrangian Multipliers [19] we have the system of equations (99)

$$\begin{aligned} 2S^{-1}x - 2S^{-1}y + \lambda b &= 0 \\ b^T x + c &= 0 \end{aligned} \tag{99}$$

This can be rewritten in block matrix form as shown in (100).

$$\begin{pmatrix} 2S^{-1} & b \\ b^T & 0 \end{pmatrix} \begin{pmatrix} x \\ \lambda \end{pmatrix} = \begin{pmatrix} 2S^{-1}y \\ -c \end{pmatrix} \tag{100}$$

The special structure of the left hand side matrix in (100) allows us to analytically compute the inverse (see Theorem 2.13 in Section 2.5) as shown in (101).

$$\begin{pmatrix} 2S^{-1} & b \\ b^T & 0 \end{pmatrix}^{-1} = \begin{pmatrix} \frac{1}{2}S(I - \frac{bb^TS}{b^TSb}) & \frac{Sb}{b^TSb} \\ \frac{b^TS}{b^TSb} & -\frac{2}{b^TSb} \end{pmatrix} \tag{101}$$

To find the arguments which satisfy (99) we solve (102) which is equivalent to solving the system of linear equations in (100).

$$\begin{pmatrix} \frac{1}{2}S(I - \frac{bb^TS}{b^TSb}) & \frac{Sb}{b^TSb} \\ \frac{b^TS}{b^TSb} & -\frac{2}{b^TSb} \end{pmatrix} \begin{pmatrix} 2S^{-1}y \\ -c \end{pmatrix} = \begin{pmatrix} S(I - \frac{bb^TS}{b^TSb})S^{-1}y - c\frac{Sb}{b^TSb} \\ 2(\frac{b^TS}{b^TSb} + \frac{c}{b^TSb}) \end{pmatrix} \tag{102}$$

Therefore, the arguments which minimise (98) are $x^* = S(I - \frac{bb^TS}{b^TSb})S^{-1}y - c\frac{Sb}{b^TSb}$. Substituting this into the objective function we have (103).

$$\begin{aligned} & (x^* - y)^T S^{-1} (x^* - y) \\ &= \left(S\left(I - \frac{bb^TS}{b^TSb}\right)S^{-1}y - c\frac{Sb}{b^TSb} - y\right)^T S^{-1} \left(S\left(I - \frac{bb^TS}{b^TSb}\right)S^{-1}y - c\frac{Sb}{b^TSb} - y\right) \\ &= \left(\frac{Sbb^Ty}{b^TSb} + c\frac{Sb}{b^TSb}\right)^T S^{-1} \left(\frac{Sbb^Ty}{b^TSb} + c\frac{Sb}{b^TSb}\right) \\ &= \frac{(Sb)^T}{b^TSb}\left(b^Ty + c\right)^T S^{-1} \frac{Sb}{b^TSb}\left(b^Ty + c\right) \\ &= \frac{b^TS}{b^TSb}\left(b^Ty + c\right)^T \frac{b}{b^TSb}\left(b^Ty + c\right) \\ &= \left(b^Ty + c\right)^T \frac{b^TSb}{(b^TSb)^2}\left(b^Ty + c\right) \\ &= \frac{(b^Ty + c)^T(b^Ty + c)}{b^TSb} \\ &= \frac{(b^Ty + c)^2}{b^TSb} \end{aligned} \tag{103}$$

We can conclude that the shortest squared Mahalanobis Distance between a point y and the constraint of (98) is $\frac{(b^Ty+c)^2}{b^TSb}$. □

In Theorem 7.7 we apply Theorem 7.6 to convert the stochastic constraints into nonlinear deterministic constraints.

Theorem 7.7. Gaussian Affine Chance Constraints If the underlying distribution of a random variable x is Gaussian then the chance constraint $\Pr(d^Tx + e \geq 0) \geq p$ can be rewritten as the deterministic constraint $\frac{(d^T\mu+e)^2}{d^TSd} \geq k^2$ where k^2 is the (constant) critical value of the inverse cumulative Chi-Squared Distribution with the degrees of freedom equal to the dimensionality of x such that $\Pr(\mathcal{X} \leq k^2) = p$.

Proof. Intuitively Theorem 7.7 posits that if the shortest squared Mahalanobis distance is further away than some threshold k^2 the chance constraint $\Pr(dx + e \geq 0) \geq p$ will be satisfied. Since x is a Gaussian stochastic variable we have that $\mathbb{E}[x] = \mu$ and $\text{var}[x] = \Sigma$. Let $\Omega = \{x \in \mathbb{R}^n \mid (x-\mu)^T\Sigma^{-1}(x-\mu) \leq k^2\}$ and k^2 be the critical value such that for the Chi-Squared Distribution with degrees of freedom equal to the dimension of x we have that $\Pr(\mathcal{X} \leq k^2) = p$. Then it is well known [39] that $\int_\Omega p(x|\mu,\Sigma)dx = p$ where $p(\cdot|\mu,\Sigma)$ is the multivariate Gaussian probability distribution function of x. If the shortest squared Mahalanobis Distance from the mean of x is further away from the affine constraint $d^Tz + e = 0$ than k^2 it implies that the curve of the function $h(z) = (z-\mu)^T\Sigma^{-1}(z-\mu) = k^2$ does not intersect with the constraint. This implies, with at least a probability of p, that the chance constraint will not be violated because the "confidence ellipse" does not intersect the constraint. See Figure for a diagram of the principle behind Theorem 7.7 [12]. **insert picture** □

It is interesting to note the striking similarity between the work in [43] and [42] and Theorem 7.7 given that we started analysing the problem within the framework of Graphical Models. In Theorem 7.8 we combine and elaborate on the work of [45], [43] to yield a QP MPC which exactly satisfies the stochastic MPC in (97) given the assumptions. Note that the dimensionality of the problem is arbitrary.

Theorem 7.8. Conversion of the Stochastic MPC formulation to the standard deterministic QP MPC formulation Under the assumptions of linearity and Gaussian distributions we can reformulate the stochastic MPC problem shown in (97) as a standard deterministic QP MPC problem shown in (104).

$$\begin{aligned}
&\min_{\mathbf{u}} \frac{1}{2}\sum_{k=0}^{N-1}\left(\mu_k^TQ\mu_k + u_k^TRu_k\right) + \frac{1}{2}\mu_N^TP_f\mu_N + \frac{1}{2}\sum_{k=0}^{N}\text{tr}(Q\Sigma_k)\\
&\text{subject to } \mu_{t+1} = A\mu_t + Bu_t\\
&\text{and } \Sigma_{t+1} = W + A\Sigma_tA^T\\
&\text{and } d^T\mu_t + e \geq 0 \;\forall\; t = 1,...,N\\
&\text{and } d^T\mu_t + e \geq k\sqrt{d^T\Sigma_td} \;\forall\; t = 1,...,N
\end{aligned} \tag{104}$$

Other deterministic constraints, e.g. on the input, can be added as usual. Note that we have assumed that the initial state estimate x_0 is available in the form of its mean, μ_0, and covariance, Σ_0.

Proof. Let the admissible set of controller inputs $\mathcal{U}$ be the same for both the stochastic MPC and the deterministic MPC formulations. Furthermore, let the current state estimate x_0 be given. Then by Theorem 7.2 the objective function and equality constraints follow. By

Theorem 7.5 the first inequality constraint follows. The last inequality constraint follows from Theorem 7.7 as shown in (105).

$$
\begin{aligned}
\frac{(d^T\mu + e)^2}{d^T S d} \geq k^2 &\implies (d^T\mu + e)^2 \geq k^2 d^T S d \\
&\implies d^T\mu + e \geq k\sqrt{d^T S d}
\end{aligned}
\tag{105}
$$

The first line of (105) follows because S is positive semi-definite and $d \neq 0$ (otherwise it would not be a constraint), therefore it can be multiplied over the inequality sign. The second line follows because of the first inequality constraint in (104): we know that $d^T\mu + e \geq 0$ and therefore we can square root both sides of the inequality constraint. □

The beauty of Theorem 7.8 is that no new theory is necessary to analyse the stability and convergence results of the new MPC. This is highly desirable because it allows one to merely "add" the last inequality constraint to your existing MPC formulation. Most practical MPCs will have some form of state estimation and thus no new parameters are introduced either. Finally, since the problem is in standard QP form it is straightforward to implement and, even more importantly, it is computationally fast because the problem is trivially convex.

7.4 Reference Tracking

So far we have only dealt with controllers which would drive the system to the origin. The more general situation we are interested in is arbitrary reference point tracking. Fortunately, Section 2.4.2 applies without modification because we managed to cast the Stochastic MPC problem into a deterministic MPC problem.

7.5 Linear System

7.6 Nonlinear System

References

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

[6] 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.

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

[8] 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.

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

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

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

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

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

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

[15] 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.

[16] 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.

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

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

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

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

[21] 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.

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

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

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

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

[26] 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.

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

[28] 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.

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

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

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

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

[33] 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.

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

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

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

[37] 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.

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

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

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

[41] 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.

[42] 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.

[43] 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.

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

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

[46] 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.

[47] 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.