Contents

Part I

Literature, theory and background material

Chapter 3

Background theory

This chapter is composed of five sections which introduce the main concepts and results used throughout the rest of the dissertation. Section 3.1 introduces probability theory. Section 3.2 very briefly introduces some useful nomenclature from graph theory. These two sections serve as an entry point for Section 3.3 which deals with probabilistic graphical models. Section 3.4 deals with control theory and Section 3.5 introduces an important result from matrix linear algebra.

It might appear as though Sections 3.1 to 3.3 and Section 3.4 are not related to each other. However, the foundational theory introduced here is expanded upon later and the relationship then becomes clear.

3.1 Probability theory

The calculus of probability theory was developed by Fermat and Pascal in order to better understand the problems introduced by uncertainty in gambling. From this dubious genesis a rich and incredibly powerful field has developed. We start our brief introduction of probability theory by restating Kolmogorov's three probability axioms - these axioms underpin the entire theory of probability [31].

Let the set Ω be the universe of possible events, also called the event space; that is, if we are uncertain about which of a number of possibilities are true then we let Ω represent all of them collectively. Let P be some real valued function which satisfies the three axioms stated below.

Axiom 3.1. $P(\Omega) = 1$. The probability of any event in Ω occurring is 1.

Axiom 3.2. $\forall \alpha \in \Omega,\ P(\alpha) \geq 0$. The probability of any one (or set of) event(s) in Ω occurring is non-negative.

Axiom 3.3. $\forall \alpha, \beta \in \Omega,$ if $\alpha \cap \beta = \varnothing$ then $P(\alpha \cup \beta) = P(\alpha) + P(\beta)$. The probability of two

mutually disjoint sets of events in Ω occurring is equal to the sum of their probabilities.

A function P which satisfies these three axioms is known as a probability function. Based on these three axioms the theory can be extended to Theorem 3.1 [31].

Theorem 3.1. $\forall \alpha, \beta \in \Omega, P(\alpha \cup \beta) = P(\alpha) + P(\beta) + P(\alpha \cap \beta)$. The probability of two events occurring in Ω is equal to the sum of their probabilities less the probability of both occurring simultaneously.

3.1.1 Discrete random variables

We now make precise what we mean by random variables: a random variable is a non-deterministic variable which is characterised by some uncertainty in its measurement. Semantically we indicate a specific value taken on by the random variable X as $X = x$ or just denote it x. Thus, the function $P(X = x) = P(x) \in \mathbb{R}$ indicates the probability of event x occurring with respect to the random variable X. We denote $P(X)$ as the probability function of the random variable X. Thus, for the discrete random variable X we have that $P(X) = (P(x_1), P(x_2), ..., P(x_n))^T$ where $x_i \in X$ for $i = 1, 2, ..., n$ and $\sum_i P(x_i) = 1$. We defer the study of the continuous case until later.

Before we proceed let us briefly discuss how we can interpret the function P for any random variable X. If $P(X = x) = 1$ we are certain of event x occurring, i.e. X will only take on the value x. If $P(x) = 0$ we are certain that event x will not occur, i.e. X will never take on the value x. Thus our certainty of event x occurring is reflected by the magnitude of $P(x)$. Attempting to make the statement "our certainty of event x occurring" more precise leads us to two different physical interpretations of $P(x)$. The first is the frequentist interpretation: to the frequentist a probability is a long term average of the observations of event x occurring in the event space. While this interpretation is satisfying if one deals with something which is easily measured e.g. the probability of a fair die rolling a 6, it fails to explain statements like: "the probability of it raining tomorrow is 50%". The reason the last statement is problematic is because the time span is ill defined. If we rather understand probabilities to mean subjective degrees of belief in event x occurring this is no longer a problem. To ensure that these subjective beliefs are rational can be problematic. One way to ensure this is by requiring that if the probabilities were used in a betting game it is impossible to exploit them to one's advantage (or disadvantage). If this is possible then there is no difference between the interpretations described above [31].

We will deal extensively with joint and marginal probability distributions. Consider the random variables X and Y. The marginal probability distribution of X is the function $P(X)$ and describes the probabilities of events involving only the variable X. The joint probability distribution of X and Y is the function $P(X, Y) = P(X \cap Y)$ and describes the intersection (and) of the probability space of X and Y. We introduce, without proof, Theorem 3.2.

Theorem 3.2. Marginalisation By marginalising out X we mean we sum out X from the joint distribution $P(Y) = \sum_x P(x, Y)$. This extends to higher dimensions.

We can reduce any joint distribution to a marginal one by summing (or integrating in the case of continuous random variables) out the appropriate variable.

It is now necessary to define what we mean by conditional probability. Definition 3.1 makes precise how the knowledge that event y has occurred alters our view of event x occurring.

Definition 3.1. Conditional probability $P(X|Y) = \frac{P(X \cap Y)}{P(Y)}$

Note that if for some $y \in Y$ we have $P(Y = y) = 0$ then Definition 3.1 is undefined. Additionally, the function $P(\cdot|Y)$ is a probability function. We next define what we mean by a positive probability distribution in Definition 3.2.

Definition 3.2. A probability distribution is positive if $P(x) > 0 \; \forall \; x \in X$.

Clearly undefined conditional probabilities are not a problem in the setting of positive probability distributions. We also define the notion of independence, also sometimes called marginal independence, in Definition 3.3.

As before, let X, Y and Z be random variables. Intuitively X and Y are independent if the outcome of X does not influence the outcome of Y. It can be shown that independence is a symmetric property [31].

Definition 3.3. Independence $X \perp\!\!\!\perp Y \equiv P(X|Y) = P(X)$

Generalising the concept of independence we define conditional independence by Definition 3.4. Again this definition is symmetric [31].

Definition 3.4. Conditional independence $X \perp\!\!\!\perp Y|Z \equiv P(X|Y, Z) = P(X|Z)$

Intuitively, if X is conditionally independent of Y given Z then by observing Z one gains nothing by observing Y. Clearly if $Z = \emptyset$ we have (marginal) independence. We also introduce Theorem 3.3 which naturally leads us to the formulation of Bayes' theorem (using Definition 3.1) as shown in Theorem 3.4.

Theorem 3.3. Chain rule Given the random variables X_1 and X_2 we have $P(X_1, X_2) = P(X_1)P(X_2|X_1)$. The generalisation to an arbitrary number of random variables is straightforward.

Theorem 3.4. Bayes' theorem $P(X|Y) = \frac{P(Y|X)P(X)}{P(Y)}$

Under the Bayesian interpretation of Theorem 3.4 we see that the posterior probability of some hypothesis X given some evidence Y being true is just the likelihood $P(Y|X)$ of the hypothesis supporting the evidence multiplied by the prior probability of the hypothesis $P(X)$ normalised by the prior of the evidence $P(Y)$. It is also convenient to notice that $P(Y)$ is a normalising constant and thus $P(X|Y) \propto P(Y|X)P(X)$.

To fully describe a system of random variables it is only necessary to know the joint distribution $P(X_1, X_2, ..., X_n)$. Given the joint probability distribution inference (reasoning about the variables under uncertainty) may be performed. Common probabilistic queries involve computing posterior beliefs $P(X|Y = y)$ i.e. the probability function of X given we have some information about Y. Other queries involve find the most probable explanation (called a MAP query) of some evidence i.e. finding X which maximises $P(X, Y = y)$. More on this later.

Example of Bayes' theorem in action

This section will attempt to develop some intuition behind Theorem 3.4. We quote an excerpt from an article in the Economist [22] and illustrate the use of Bayes' theorem in a canonical medical example [32].

"The essence of the Bayesian approach is to provide a mathematical rule explaining how you should change your existing beliefs in the light of new evidence. In other words, it allows scientists to combine new data with their existing knowledge or expertise. The canonical example is to imagine that a precocious newborn observes his first sunset, and wonders whether the sun will rise again or not. He assigns equal prior probabilities to both possible outcomes, and represents this by placing one white and one black marble into a bag. The following day, when the sun rises, the child places another white marble in the bag. The probability that a marble plucked randomly from the bag will be white (ie, the child's degree of belief in future sunrises) has thus gone from a half to two-thirds. After sunrise the next day, the child adds another white marble, and the probability (and thus the degree of belief) goes from two-thirds to three-quarters. And so on. Gradually, the initial belief that the sun is just as likely as not to rise each morning is modified to become a near-certainty that the sun will always rise."

Now for the canonical medical example. Suppose you get tested for a certain disease. You know the disease affects 1 in 100 people. You also know that the false positive rate for the test is 20% and the false negative rate for the test is is 10%. Your test comes back positive. What are the chances of you having the disease given this information?

The information may be summarised as shown below. Let D be a binary random variable indicating the presence of the disease and $\neg D$ indicates the absence. Let T be a binary random variable indicating a positive test and $\neg T$ indicates a negative test.

1. The prior of the disease is $P(D) = 0.01$.
2. False positive rate $P(T|\neg D) = 0.2 \implies P(\neg T|\neg D) = 0.8$.
3. False negative rate $P(\neg T|D) = 0.1 \implies P(T|D) = 0.9$.

A naive approach would conclude that since $P(T|D) = 0.9$ you are 90% likely to have the

disease. However, using Bayesian inference/reasoning we have:

$$\begin{aligned} P(D|T) &= \frac{P(T|D)P(D)}{P(T)} \\ &= \frac{P(T|D)P(D)}{\sum_D P(D,T)} \\ &= \frac{P(T|D)P(D)}{\sum_D P(D)P(T|D)} \\ &= \frac{0.9 \times 0.01}{0.01 \times 0.99 + 0.99 \times 0.2} \\ &\approx 0.04 \end{aligned}$$

Clearly there is a big difference between the naive approach and the Bayesian (correct) approach. The power of Bayesian inference lies in the ability to reverse causal reasoning. That is, we know that the disease causes the test to be positive, $P(T|D)$, but we would like to reverse this reasoning to infer $P(D|T)$. This is immensely powerful as we shall soon discover.

3.1.2 Continuous random variables

So far in our discussion we have implicitly only used discrete random variables; that is, our probability space consisted out of a finite number of events or states. However, it is also necessary to make precise what we mean by a continuous random variable. A continuous random variable is characterised by a density function p which assigns a weight to each possible value of the variable. Intuitively this weight is *related* to the probability of that value occurring[1]. Although the density function is itself not a probability function, if it satisfies $p(x) \geq 0 \ \forall x \in X$ and $\int p(x)\mathrm{d}x = 1$, where we have implicitly integrated over the domain of p, then it can be used to generate one. The cumulative probability function $P(X \leq a) = \int_{-\infty}^{a} p(x)\mathrm{d}x$ is one such example[2].

Arguably the most well known continuous probability density distribution is the Gaussian or normal distribution. The Gaussian distribution arises naturally from a variety of different contexts and settings. For example, the central limit theorem, together with some mild assumptions, tells us that the sum of a set of N random variables is itself a random variable and in the limit can be described by a Gaussian distribution [6]. The Gaussian is regularly used because it has some very appealing analytical properties (and also often because it is physically meaningful) which we will investigate in some depth.

Since the probability of a specific value is not meaningful in the setting of continuous probability functions we abuse our notation and interchangeably denote the random variable X by x. We also do not indicate vector quantities in boldface; it can be assumed that all numbers

[1] Please note that strictly speaking $P(x) = 0$ for a specific point x in the domain of p. Technically it is correct to say that the probability of $P(x \in [a,b]) = \int_a^b p(y)\mathrm{d}y$; thus, if we want the probability of x occurring we could just make $[a,b]$ small to get some approximation.

[2] We have assumed that the domain of X is the entire real line.

are vectors unless otherwise noted. We will concern ourselves mostly with vector quantities and it will be obvious when we deal with scalar valued variables.

Definition 3.5. Gaussian distribution The univariate Gaussian or normal distribution of a random variable x is

$$\mathcal{N}(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right). \tag{3.1}$$

We call μ the mean and σ^2 the variance of the distribution. The multivariate Gaussian distribution is

$$\mathcal{N}(x|\mu,\Sigma) = \frac{1}{\sqrt{2\pi^{\frac{D}{2}}}} \frac{1}{|\Sigma|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right) \tag{3.2}$$

where μ is a D dimensional mean vector and Σ is a $D \times D$ dimensional covariance matrix. Note that we often use the inverse of the covariance matrix, called the precision matrix and define it $\Lambda \equiv \Sigma^{-1}$.

It is also appropriate to define some functions which apply equally well to the discrete case as to the continuous case (just replace the integration with summation in the setting of discrete random variables). We define the expectation (or mean or average) in Definition 3.6, the variance in Definition 3.7 and the covariance in Definition 3.8.

Definition 3.6. Expectation The average value of some integrable function f under the probability distribution p is denoted $\mathbb{E}[f] = \int p(x)f(x)dx$.

We have that $\mathbb{E}[x] = \mu$ if x is a Gaussian random variable.

Definition 3.7. Variance The variance of f is defined by $\text{var}[f] = \mathbb{E}[(f - \mathbb{E}[f])^2]$ and provides a measure of how much variability there is in f around its mean value $\mathbb{E}[f]$.

By expanding out the square we have the familiar formula $\text{var}[f] = \mathbb{E}[f^2] - \mathbb{E}[f]^2$. Also note that for a univariate Gaussian random variable x we have $\text{var}[x] = \sigma^2$.

Definition 3.8. Covariance For two random variables x, y (which may be column vectors) we define the covariance matrix $\text{cov}[x,y] = \mathbb{E}[xy^T] - \mathbb{E}[x]\mathbb{E}[y]$.

Note that $\text{cov}[x,x] = \text{cov}[x] = \text{var}[x]$. Covariance is a measure of how much two random variables vary together. If x is a D dimensional Gaussian random variable then $\text{cov}[x] = \Sigma$ as defined in Definition 3.5.

The identities in Theorem 3.5 will be useful in later sections. We refer the reader to [15] for justification.

Theorem 3.5. Gaussian expected value identities Suppose there exist constants $c \in \mathbb{R}^n$ and $C \in \mathbb{R}^{n\times n}$ and X is a normal random variable with statistics (μ, Σ). Then the following identities hold:

1. $\mathbb{E}[c^T X] = c^T\mu$

2. $\mathbb{E}[CX + c] = C\mu + c$

3. $\mathbb{E}[X^T CX] = \text{tr}(C\Sigma) + \mu^T C\mu$

Now we are in a position to perform some manipulations assuming we are using Gaussian random variables. We state without proof Theorem 3.6.

Theorem 3.6. Partitioned joint gaussians Given a Gaussian distribution $\mathcal{N}(x|\mu, \Sigma)$ with $\Lambda \equiv \Sigma^{-1}$ and $x = \begin{pmatrix} x_a \\ x_b \end{pmatrix}$, $\mu = \begin{pmatrix} \mu_a \\ \mu_b \end{pmatrix}$, $\Sigma = \begin{pmatrix} \Sigma_{aa} & \Sigma_{ab} \\ \Sigma_{ba} & \Sigma_{bb} \end{pmatrix}$ and $\Lambda = \begin{pmatrix} \Lambda_{aa} & \Lambda_{ab} \\ \Lambda_{ba} & \Lambda_{bb} \end{pmatrix}$ then we have the conditional distribution

$$\begin{aligned} &p(x_a|x_b) = \mathcal{N}(x_a|\mu_{a|b}, \Lambda_{aa}^{-1}) \\ &\text{with } \mu_{a|b} = \mu_a - \Lambda_{aa}^{-1}\Lambda_{ab}(x_b - \mu_b) \end{aligned} \tag{3.3}$$

and the marginal distribution

$$p(x_a) = \mathcal{N}(x_a|\mu_a, \Sigma_{aa}). \tag{3.4}$$

It is often easier to work with the precision matrix when dealing with conditional distributions.

Next we state and then prove Theorem 3.7 which we will use extensively. The proof for Theorem 3.6 uses the same techniques and can be found in [6].

Theorem 3.7. Bayes' theorem for linear gaussian models Suppose we have a marginal Gaussian distribution for x and a conditional Gaussian distribution for y:

$$\begin{aligned} p(x) &= \mathcal{N}(x|\mu, \Lambda^{-1}) \\ p(y|x) &= \mathcal{N}(y|Ax + b, L^{-1}). \end{aligned} \tag{3.5}$$

Then the marginal distribution for y is

$$p(y) = \mathcal{N}(y|A\mu + b, L^{-1} + A\Lambda^{-1}A^T) \tag{3.6}$$

and the conditional distribution for x given y is

$$\begin{aligned} &p(x|y) = \mathcal{N}(x|\Sigma(A^T L(y - b) + \Lambda\mu), \Sigma) \\ &\text{with } \Sigma = (\Lambda + A^T LA)^{-1}. \end{aligned} \tag{3.7}$$

where b is a known vector or function of some deterministic variable.

Proof. We begin our proof by noticing that for a general Gaussian $\mathcal{N}(\gamma|\alpha, \beta)$ we can write the exponent as

$$-\frac{1}{2}(\gamma - \alpha)^T \beta^{-1}(\gamma - \alpha) = -\frac{1}{2}\gamma^T\beta^{-1}\gamma + \gamma^T\beta^{-1}\alpha + \text{const} \tag{3.8}$$

where const is some real number which does not depend on γ. It is known that Gaussian distributions are closed under multiplication, i.e. if one multiplies two Gaussian distributions

the product is still a Gaussian distribution (of a higher dimension) [6]. To find the joint distribution we let $z = \begin{pmatrix} x \\ y \end{pmatrix}$ and consider the logarithm of the joint

$$\begin{aligned} \log(z) &= \log(p(x)) + \log(p(y|x)) \\ &= -\frac{1}{2}(x-\mu)^T\Lambda(x-\mu) - \frac{1}{2}(y-Ax-b)^T L(y-Ax-b) + const \end{aligned} \tag{3.9}$$

Here const denotes constant terms which are independent of x and y. Now we make use of (3.8) to find the mean and covariance of z. Continuing, we consider only the second order terms when (3.9) is expanded

$$\begin{aligned} &-\frac{1}{2}x^T(\Lambda + A^TLA)x - \frac{1}{2}y^TLy + \frac{1}{2}y^TLAx + \frac{1}{2}x^TA^TLy \\ &= -\frac{1}{2}\begin{pmatrix} x \\ y \end{pmatrix}^T \begin{pmatrix} \Lambda + A^TLA & -A^TL \\ -LA & L \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \\ &= -\frac{1}{2}z^TRz. \end{aligned} \tag{3.10}$$

From this we immediately have the precision of z: the matrix R; we also use a matrix inversion formula found in [6] to find the covariance

$$\text{cov}[z] = R^{-1} = \begin{pmatrix} \Lambda^{-1} & \Lambda^{-1}A^T \\ A\Lambda^{-1} & L^{-1} + A\Lambda^{-1}A^T \end{pmatrix}. \tag{3.11}$$

We now proceed in exactly the same way to find mean of z. By expanding 3.9 and only considering the first order terms in x and y we have

$$x^T\Lambda\mu - x^TA^TLb + y^TLb = \begin{pmatrix} x \\ y \end{pmatrix} \begin{pmatrix} \Lambda\mu - A^TLb \\ Lb \end{pmatrix}. \tag{3.12}$$

Again, by making use of (3.8) and the fact that the covariance of z is R^{-1} it is possible to show that $\mathbb{E}[z] = \begin{pmatrix} \mu \\ A\mu + b \end{pmatrix}$ as shown in [6]. By using Theorem 3.6 we immediately have the marginal and conditional distributions as required. □

We also introduce a useful metric for measuring the similarity between two distributions in Definition 3.9.

Definition 3.9. Kullback-Leibler divergence Consider some unknown distribution $p(x)$ and suppose we have modelled this distribution by $q(x)$. Kullback-Leibler divergence, also known as relative entropy, is defined $\text{KL}(p||q) = -\int p(x)\ln\left(\frac{q(x)}{p(x)}\right)dx$ and measures the additional amount of information, in *nats*, needed to specify the value of x [6].

Kullback-Leibler divergence can be used to measure the dissimilarity between two distributions. If the measure is zero the distributions are identical; care needs to be taken when using Kullback-Leibler divergence because the measure is not symmetric. We introduce Theorem 3.8 to measure the dissimilarity between a known distribution and a sampled approximation thereof. See [6] for the motivation.

Theorem 3.8. Kullback-Leibler sample divergence Suppose we observe a finite set of points x_n for $n = 1, 2, ..., N$ drawn from $p(x)$. Furthermore, suppose we would like to measure the information loss when p is approximated by q. We can measure this by $\mathrm{KL}(p||q) \approx \frac{1}{N}\sum_{n=1}^{N}(-\ln(q(x_n)) + \ln(p(x_n)))$. This measurement is bounded below by zero. If, as $N \to \infty$, the information loss is zero p and q are functionally equivalent.

We also briefly introduce the Mahalanobis distance in Definition 3.10.

Definition 3.10. Mahalanobis distance The Mahalanobis distance between x and a reference point $y \in \mathbb{R}^n$ given a covariance matrix $S \in \mathbb{R}^{n\times n}$, is defined by $D_M(x|y, S) = \sqrt{(x-y)^T S^{-1}(x-y)}$.

The Mahalanobis distance is a statistical distance metric which reduces to the Euclidean distance metric if $S = I$. It is found in the exponent of the Gaussian distribution density function and can be used to measure the "closeness" of points between distributions with a common covariance matrix. We will study it in more detail later.

3.2 Graph theory

A graph, G, is a data structure consisting of a set of nodes χ and edges ξ. A pair of nodes $X_i, X_j \in \chi$ can be connected by an edge. We will only consider directed graphs in this dissertation. This implies that every edge in ξ has a direction associated between the two nodes it connects i.e. $X_i \to X_j$ if there is an edge from X_i to X_j.

We now define some basic concepts which we will rely upon to further describe the types of graphs we will consider.

Definition 3.11. Directed path We say that the nodes $X_1, X_2, X_3, ..., X_n \in \chi$ form a directed path if $X_i \to X_{i+1}$ for $1 \leq i \leq n-1$.

Definition 3.12. Directed cycle A directed cycle is a non-singleton directed path which starts and ends at the same node.

Definition 3.13. Directed acyclic graph (DAG) A graph G is a DAG if it is directed and has no directed cycles.

In this dissertation we will only concern ourselves with DAGs. Figure 3.1 is an example of a DAG.

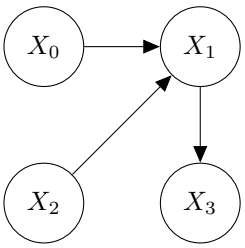

Figure 3.1: Example of a directed acyclic graph.

Next we define some nomenclature to further describe the nodes of a graph G.

Definition 3.14. Parents We say that the set of nodes $\kappa \subset \chi$ are the parents of node X_i if, for each node in κ, there exists an edge going to X_i.

Definition 3.15. Children We say that the set of nodes $\tau \subset \chi$ are the children of node X_i if, for each node in τ, there exists an edge going from X_i to that node.

Definition 3.16. Descendants We say that the set of nodes $\gamma \subset \chi$ are the descendants of node X_i if, for each node in γ, there exists a directed path from X_i to that node.

We also briefly define a structured approach to encoding a graph.

Definition 3.17. Adjacency matrix For a graph G with n nodes, the adjacency matrix A is an $n \times n$ matrix where $A_{ij} = 1$ if there is an edge from node i to node j and $A_{ij} = 0$ otherwise.

The adjacency matrix A for Figure 3.1 is shown below:

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

A detailed analysis of graph theory may be found in [17].

3.3 Probabilistic graphical models

Probabilistic graphical models are the union between probability theory and graph theory. Consider why, in general, it is infeasible to determine an arbitrary joint probability distribution. Suppose you have a set of n binary random variables and wish to determine their joint. This equates to finding $P(X_1, X_2, ..., X_n)$. To fully specify this model we would need to find and store $2^n - 1$ probabilities. For even moderately big n this is impractical, and this was for the simple case of a binary valued random variable. Clearly we require a more efficient way to represent the joint probability distribution.

3.3.1 Bayesian networks

A Bayesian network is a representation of the joint probability distribution of a set of random variables parametrised by:

1. A graph depicting local independence relationships.

2. Conditional probability distributions.

The fundamental assumption behind Bayesian networks, and more generally probabilistic graphical models, is that there is a useful underlying structure to the problem being modelled which can be captured by the Bayesian network. This underlying structure is available via conditional independence relationships between the variables.

Suppose P is the joint distribution of some set of random variables we require to do inference on.

Definition 3.18. I-Map The I-Map of P, denoted by $\mathcal{I}(P)$, is the set of independence assertions of the form $X \perp\!\!\!\perp Y|Z$ which hold over P.

Let G be a Bayesian network graph over the random variables $X_1, X_2, ..., X_n$ where each random variable is a node. We say that the distribution P factorises over the same space if P can be expressed as the product defined by the chain rule for Bayesian networks.

Definition 3.19. Chain Rule for Bayesian Networks The chain rule for Bayesian networks specifies that the joint factorises according to $P(X_1, ..., X_n) = \Pi_{i=1}^{n} P(X_i|\text{Parents}(X_i))$.

Each of the individual factors of P, as factorised by the chain rule for Bayesian networks, represents the conditional probability distributions required to parametrise the Bayesian network. It can be shown that a Bayesian network graph G over which P factorises is not unique. However, if the graph explicitly models the causality inherent in the system being modelled the representation is often much sparser [31]. A Bayesian network is then defined as the tuple (G, P) such that the joint P factorises over the graph G. We state without proof Theorem 3.9.

Theorem 3.9. Let G be a Bayesian network graph over a set of random variables χ and let P be a joint distribution over the same space. If P factorises according to G then G is an I-Map for P. Conversely, if G is an I-Map for P then P factorises according to G.

Thus, the conditional independences imply factorisation of P. Conversely, factorisation according to G implies the associated conditional independences.

To illustrate computational benefit of using Bayesian networks, consider again our simple system of n binary random variables $X_1, X_2, ..., X_n$. Suppose the Bayesian network in Figure 3.2 models the system.

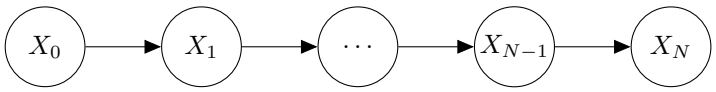

Figure 3.2: Example of a simple Bayesian network.

Without knowing any structure $2^n - 1$ parameters were needed to specify the joint. However, using the chain rule for Bayesian networks we can factorise the joint $P(X_1, ..., X_n) = P(X_1)P(X_2|X_1)...P(X_n|X_{n-1})$. This implies that we only require $2n - 1$ parameters. From a modelling perspective this is a significant gain.

The primary reason we would want to have a model of the joint distribution of a set of random variables is to reason with. To achieve this we invariably manipulate the joint distribution by either some form of marginalisation or optimisation. To make inference computationally tractable it is desirable to leverage the independence assertions implied by the network graph. To this end we expand on the independence assertions implied by the graph. Recall Theorem 3.9: since we have that the joint factorises over the graph we also have that any independence assertions implied by the graph's connectivity also apply to the joint.

We introduce the concept of d-separation as a method of determining whether a set of nodes X are conditionally independent of another set Y given the set E. Firstly we generalise the concept of a directed path to an undirected path between sets of variables.

Definition 3.20. Undirected path An undirected path between two sets of nodes X and Y is any sequence of nodes between a member of X and a member of Y such that every adjacent pair of nodes is connected by an edge regardless of direction and no node appears twice.

Definition 3.21. Blocked path A path is blocked, given a set of nodes **E**, if there is a node Z on the path for which at least one of the three conditions holds:

1. Z is in E and Z has one edge leading into it from the path and one edge leading out of it on the path.

2. Z is in E and Z has both edges leading out of it from the path.

3. Neither Z nor any descendant of Z is in E and both path edges lead into Z.

Definition 3.22. D-separation A set of nodes E d-separates two other sets of nodes X and Y if every path from a node in X to a node in Y is blocked given E.

To shed some more light on d-separation consider Figure 3.3. The first diagram depicts the first blocked condition, i.e. a causal chain. Node E blocks relevance of X to Y. The second diagram illustrates the second blocked condition, i.e. a common cause. Node E blocks X from being relevant to Y. Finally, the third diagram illustrates the third blocked condition or, more aptly, illustrates how lack of knowledge of the nodes in the path from X to Y implies that they are conditionally independent [32].

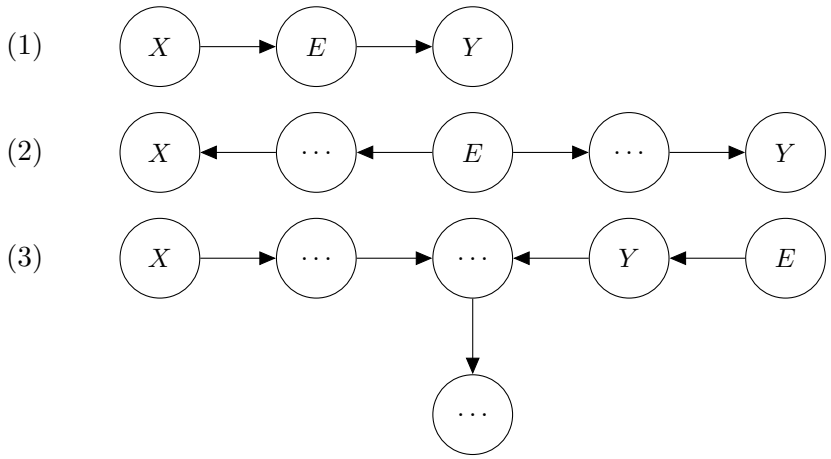

Figure 3.3: Examples of d-separation.

Using d-separation we can efficiently reason about the conditional independences implied by the graph and the observed variables (E). This becomes incredibly useful when one attempts to apply inference techniques because it can simplify the joint calculations significantly. More on this later.

Bayesian networks are commonly used to model situations which are not time dependent. We will primarily restrict ourselves to time series modelling in this dissertation. As such we will not delve deeper into static Bayesian network theory.

3.3.2 Dynamic Bayesian networks

Dynamical Bayesian networks generalise the conventional static Bayesian networks of the previous section. Dynamic, or temporal, Bayesian networks model systems which evolve with time. Since sequential, or temporal, data is abundant in most engineering applications we will primarily concern ourselves with such models. Notationally we denote a time dependent vector by $x_{1:t} = x_1, x_2, ..., x_t$, for example the joint $P(x_{1:3}) = P(x_1, x_2, x_3)$.

There are two important classes of analysis one may perform on sequential data using graphical models. On-line analysis, including prediction and filtering and off-line analysis, including smoothing and the most probable explanation (sometimes called Viterbi decoding). In both cases we are generally interested in learning something about a set of hidden state variables by performing inference on some set of observed variables.

A state space model assumes that there is some underlying hidden state (x_t) of the world which generates observations (y_t). These hidden states may evolve with time and may be functions of some inputs (u_t). The hidden states and observations are most generally assumed to be random variables. Any state space model is fully parametrised by the following information:

1. A prior probability distribution over the states: $P(x_0)$

2. A state transition function: $P(x_t|x_{0:t-1}, u_{0:t-1})$

3. An observation function: $P(y_t|x_{0:t}, u_{0:t-1})$

For the purposes of this dissertation we will assume that the state space model is known. If this model is not known machine learning techniques may be used find these models [43]. To simplify notation we will sometimes omit the dependence of the probability functions on the inputs $u_{0:t}$.

We will assume that all the systems we model satisfy the first order Markov assumption.

Definition 3.23. N$^{\text{th}}$-order Markov assumption A system satisfies the N$^{\text{th}}$ Markov assumption if $P(x_t|x_{0:t}) = P(x_t|x_{t-n:t-1})$. For example, a first order Markov system satisfies $P(x_t|x_{0:t}) = P(x_t|x_{t-1})$. Similarly with the observation function.

This is not as restrictive as it may seem at first. It is always possible to transform an N$^{\text{th}}$-order Markov system into a first order Markov system by modifying the state space [43]. We also assume that the state and observation functions remain the same for all time i.e. they are time invariant or homogeneous or stationary.

Intuitively, a state space model is a model of how x_t generates or causes y_t and x_{t+1}. The goal of inference is to invert this mapping. The four types of inference we will consider in this dissertation are:

1. Filtering: we attempt to infer $P(x_t|y_{0:t})$, i.e. we attempt to estimate the current state given all past observations.

2. Smoothing: we attempt to infer $P(x_{t-m}|y_{0:t})$ with $m > 0$, i.e. we attempt to estimate some past state given all the past and future observations. A more apt description of this process is applying hindsight to state estimation.

3. Prediction: we attempt to infer either $P(x_{t+m}|y_{0:t})$ or $P(y_{t+m}|y_{0:t})$ with $m > 0$, i.e. we attempt to estimate the future hidden states or observations given all the past observations.

4. Viterbi decoding: we attempt to perform $x^*_{1:t} = \arg\max_{x_{1:t}} P(x_{1:t}|y_{1:t})$, i.e. we attempt to infer the most likely sequence of states which best explain the observations.

It is customary to denote hidden (latent) variables by a clear node, observed (visible) variables by a shaded node and deterministic variables by a diamond shaped node. Additionally, it is also customary to separate the input, state and observation variables from each other: $z_t = (u_t, x_t, y_t)$.

To fully specify a dynamic Bayesian network we require the pair $(B_0, B_\rightarrow)$. The Bayesian network B_0 defines the prior over the random variables being modelled and $B_\rightarrow$ defines the transition and observation functions by means of a Bayesian network graph, typically over two time slices, see Figure 3.4 for an example. This Bayesian network graph may be factorised

according to the Bayesian network chain rule such that at each time slice

$$P(z_t|z_{t-1}) = \Pi_{i=1}^{N} P(z_t^i|\text{Parents}(z_t^i)). \tag{3.13}$$

A dynamic Bayesian network may be unrolled (temporally) into a (long) Bayesian network. If one views dynamic Bayesian networks as an extension of Bayesian networks all the previous theory applies. Using the chain rule for Bayesian networks again we can specify the full joint over time as

$$P(z_{0:T}) = \Pi_{t=1}^{T}\Pi_{i=1}^{N} P(z_t^i|\text{Parents}(z_t^i)). \tag{3.14}$$

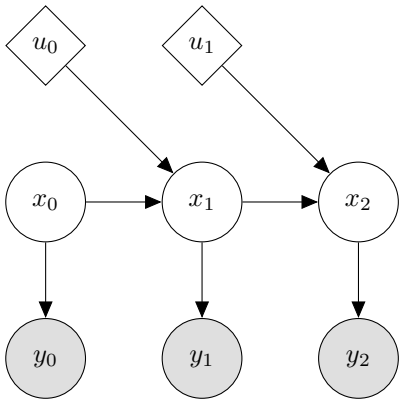

Figure 3.4: Example of a dynamic Bayesian network unrolled for 3 time slices.

3.4 Control

In this section we briefly introduce three fundamental control strategies. First, the linear quadratic regulator (LQR) which deals with the optimal control of linear discrete time invariant systems. Second, we deal with the stochastic generalisation of the LQR controller: the famous linear quadratic Gaussian (LQG) controller. Third and finally, we introduce deterministic model predictive control (MPC). The aim of this section is to introduce and illustrate the relationship between these controllers.

3.4.1 Linear quadratic regulator control

We start our analysis by assuming we have an accurate, linear, discrete, time invariant state space representation of a system

$$x_{t+1} = Ax_t + Bu_t. \tag{3.15}$$

The control sequence N steps into the future is denoted $\mathbf{u} = (u_0, u_1, ..., u_{N-1})$. It is our goal to derive a linear quadratic regulator (controller) given the system in (3.15) and the initial state x_0. Note that it is customary to assign $x_0 \leftarrow x_t$ at each time step to simplify the succeeding optimisation problem's notation.

Definition 3.24. Linear quadratic regulator (LQR) objective function The controller minimising the quadratic objective function

$$V(x_0, \mathbf{u}) = \frac{1}{2}\sum_{k=0}^{N-1}\left(x_k^T Q x_k + u_k^T R u_k\right) + \frac{1}{2}x_N^T P_f x_N \tag{3.16}$$

is called the LQR controller.

The optimisation of (3.16) is implicitly subject to the state dynamics (3.15). The matrices Q, R and Q_f are tuning parameters affecting the relative importance of the state and control inputs to the objective function respectively. We also assume that Q, P_f and R are real and symmetric matrices with the additional assumption that Q, P_f are positive semidefinite and R is positive definite.

We assume the reader is familiar with dynamic programming, and present Theorem 3.10 because it will be useful later. The proof may be found in [48] and follows from algebraic manipulations.

Theorem 3.10. Sum of quadratics Suppose two quadratic functions $V_1(x) = \frac{1}{2}(x-a)^T A(x-a)$ and $V_2(x) = \frac{1}{2}(x-b)^T B(x-b)$ are given. Then the sum $V_1(x) + V_2(x) = V(x)$ is also quadratic and $V(x) = \frac{1}{2}(x-v)^T H(x-v) + d$ with $H = A + B$, $v = H^{-1}(Aa + Bb)$ and $d = V_1(v) + V_2(v)$.

We now state the complete LQR problem for finite horizon linear systems

$$\begin{aligned}&\min_{\mathbf{u}} V(x_0, \mathbf{u}) = \frac{1}{2}\sum_{k=0}^{N-1}\left(x_k^T Q x_k + u_k^T R u_k\right) + \frac{1}{2}x_N^T P_f x_N \\ &\text{subject to } x_{t+1} = Ax_t + Bu_t\end{aligned} \tag{3.17}$$

and analytically solve it using backward dynamic programming. Expanding the objective function to examine its structure we have

$$\begin{aligned}\min_{\mathbf{u}} V(x_0, \mathbf{u}) &= \min_{\mathbf{u}} \frac{1}{2}\sum_{k=0}^{N-1}\left(x_k^T Q x_k + u_k^t R u_k\right) + \frac{1}{2}x_N^T P_f x_N \\ &= \min_{u_0, u_1, \ldots, u_{N-1}} \frac{1}{2}\left(x_0^T Q x_0 + u_0^T R u_0 + x_1^T Q x_1 + u_1^T R u_1 + \ldots + x_N^T P_f x_N\right) \\ &= \min_{u_0, u_1, \ldots, u_{N-2}} \frac{1}{2}\left(x_0^T Q x_0 + u_0^T R u_0 + \ldots + x_{N-2}^T Q x_{N-2} + u_{N-2}^T R u_{N-2}\right) \ldots \\ &\quad + \min_{u_{N-1}} \frac{1}{2}\left(x_{N-1}^T Q x_{N-1} + u_{N-1}^T R u_{N-1} + x_N^T P_f x_N\right).\end{aligned} \tag{3.18}$$

Note that given x_0 and the system dynamics all succeeding states are unknown only in the control input. The expansion of the objective function is recursive; this structure motivates the use of dynamic programming. By using Theorem 3.10 and the constraint $x_N = Ax_{N-1} + Bu_{N-1}$ we can simplify the last term in the separated minimisation problem of (3.16) by

writing

$$\begin{aligned}
\min_{u_{N-1}} V_{N-1}(x_N, u_{N-1}) &= \min_{u_{N-1}} \frac{1}{2}\left(x_{N-1}^T Q x_{N-1} + u_{N-1}^T R u_{N-1} + x_N^T P_f x_N\right) \\
&= \min_{u_{N-1}} \frac{1}{2}\left(x_{N-1}^T Q x_{N-1} + (u_{N-1} - v)^T H (u_{N-1} - v)\right) + d \\
&\text{with } H = R + B^T P_f B \\
&\text{and } v = K_{N-1} x_{N-1} \\
&\text{and } d = \frac{1}{2} x_{N-1}^T \left(K_{N-1}^T R K_{N-1} + (A + B K_{N-1})^T P_f (A + B K_{N-1})\right) x_{N-1} \\
&\text{and } K_{N-1} = -(B^T P_f B + R)^{-1} B^T P_f A.
\end{aligned} \tag{3.19}$$

This is the first step of backward dynamic programming used to solve the problem. Given the form of the objective function we see that the optimal input u_{N-1} is v and consequently that the optimal control law at time $N-1$ is a linear function, K_{N-1}, of x_{N-1}. We also see that the cost function of the last stage is quadratic. The optimal stage cost and controller action is

$$\begin{aligned}
u_{N-1}^0(x) &= K_{N-1} x \\
x_{N-1}^0(x) &= (A + B K_{N-1}) x \\
V_{N-1}^0(x) &= \frac{1}{2} x^T \Pi_{N-1} x \\
K_{N-1} &= -(B^T P_f B + R)^{-1} B^T P_f A \\
\Pi_{N-1} &= Q + A^T P_f A - A^T P_f B (B^T P_F B + R)^{-1} B^T P_f A.
\end{aligned} \tag{3.20}$$

The function $V_{N-1}^0(x)$ defines the optimal cost to go from state x for the last stage under the optimal control law $u_{N-1}^0(x)$. Now we proceed with the backward dynamic programming and solve

$$\min_{u_{N-2}} \frac{1}{2}\left(x_{N-2}^T Q x_{N-2} + u_{N-2}^T R u_{N-2}\right) + V_{N-1}^0(x_{N-1}). \tag{3.21}$$

But now we note the similarity between (3.19) and (3.21). Using $x_{N-1} = A x_{N-2} + B u_{N-2}$ and the same procedure as before we have

$$\begin{aligned}
u_{N-2}^0(x) &= K_{N-2} x \\
x_{N-2}^0(x) &= (A + B K_{N-2}) x \\
V_{N-2}^0(x) &= \frac{1}{2} x^T \Pi_{N-2} x \\
K_{N-2} &= -(B^T \Pi_{N-1} B + R)^{-1} B^T \Pi_{N-1} A \\
\Pi_{N-2} &= Q + A^T \Pi_{N-1} A - A^T \Pi_{N-1} B (B^T \Pi_{N-1} B + R)^{-1} B^T \Pi_{N-1} A.
\end{aligned} \tag{3.22}$$

The recursion to go from Π_{N-1} to Π_{N-2} is known as backward Ricatti iteration and is defined by

$$\Pi_{k-1} = Q + A^T \Pi_k A - A^T \Pi_k B (B^T \Pi_k B + R)^{-1} B^T \Pi_k A. \tag{3.23}$$

With terminal condition $\Pi_N = P_f$. We see that to find the optimal control policy we need to continue with the backward dynamic programming recursion relationships until $k = 1$. We summarise one of the most fundamental results in optimal control theory in Theorem 3.11.

Theorem 3.11. Solution of the finite horizon LQR control problem Given a finite horizon N and a discrete linear system as shown in (3.15) the optimal control policy which minimises the LQR objective function of defintion 3.24 is given by iterating

$$\begin{aligned} u_k^0(x) &= K_k x \\ K_k &= -(B^T \Pi_{k+1} B + R)^{-1} B^T \Pi_{k+1} A \end{aligned} \tag{3.24}$$

backwards for $k = N-1, N-2, ..., 1$ using backward Ricatti iteration as shown in (3.23). The optimal cost to go from time k to time N is $V_k^0(x) = \frac{1}{2} x^T \Pi_k x$.

After the optimal input $\mathbf{u}$ is found only u_0 is applied. For a treatment of the continuous case see [26].

Unfortunately optimal control in the setting described above does not guarantee stable control [48]. It can be shown that the finite horizon controller is not guaranteed to be stable i.e. there exist non-trivial systems for which the controller is unstable. This problem is fixed by considering the infinite horizon LQR problem.

Definition 3.25. Infinite horizon LQR problem Find the optimal control sequence $\mathbf{u}$ which solves

$$\begin{aligned} &\min_{\mathbf{u}} V(x, \mathbf{u}) = \frac{1}{2} \sum_{k=0}^{\infty} \left(x_k^T Q x_k + u_k^T R u_k \right) \\ &\text{subject to } x_{t+1} = A x_t + B u_t \\ &\text{and } x_0 = x. \end{aligned} \tag{3.25}$$

The same restrictions on the tuning parameters apply as before.

By assuming that the system under consideration is controllable it is possible to show that the infinite horizon LQR solution shown in Theorem 3.12 is convergent and stabilising [48].

Definition 3.26. Controllability A system is controllable is, for any pair of states x, z in the state space, z can be reached in finite time from x. That is, x can be controlled to z. It is possible to characterise a controllable system further. A system with n variables (which require control) is controllable if and only if rank $\begin{pmatrix} \lambda I - A & B \end{pmatrix} = n$ for all $\lambda \in \text{eig}(A)$.

Theorem 3.12. Solution of the infinite horizon LQR control problem Given the infinite horizon LQR problem of definition 3.25 it can be shown that the optimal control is given by

$$\begin{aligned} u_k^0(x) &= Kx \\ K &= -(B^T \Pi B + R)^{-1} B^T \Pi A \\ \Pi &= Q + A^T \Pi A - A^T \Pi B (B^T \Pi B + R)^{-1} B^T \Pi A. \end{aligned} \tag{3.26}$$

The optimal cost is given by $V^0(x) = \frac{1}{2} x^T K x$. The matrix Π can be found by iterating the Ricatti equation. This solution is stabilising if the system is controllable [48].

3.4.2 Reference tracking

The LQR control problem, as discussed in the previous section, applies to deterministic systems where the goal is to drive the controlled variables to the origin. It is straightforward to extend this approach to systems where it is desired to drive the states to a reference (set) point r_{sp}.

To achieve this we simply redefine the objective function in terms of deviation variables

$$\begin{aligned} \tilde{x}_t &= x_t - x_{sp} \\ \tilde{u}_t &= u_t - u_{sp}. \end{aligned} \tag{3.27}$$

The constants x_{sp} and u_{sp} are the state and corresponding controller set point one would like to drive the system to. The deviation variables are then used in the objective function

$$\begin{aligned} &\min_{\tilde{\mathbf{u}}} V(x_0, \tilde{\mathbf{u}}) = \frac{1}{2}\sum_{k=0}^{N-1}\left(\tilde{x}_k^T Q \tilde{x}_k + \tilde{u}_k^T R \tilde{u}_k\right) + \frac{1}{2}\tilde{x}_N^T P_f \tilde{x}_N \\ &\text{subject to } \tilde{x}_{t+1} = A\tilde{x}_t + B\tilde{u}_t \end{aligned} \tag{3.28}$$

as opposed to x_t and u_t. The system dynamics remain the same [48] and only $\tilde{u}_0$ is used as before. We apply $u_0 = \tilde{u}_0 + u_{sp}$ to the system. All that is required is that we specify x_{sp} and u_{sp}. This is done by solving

$$\begin{pmatrix} I - A & -B \\ HC & 0 \end{pmatrix}\begin{pmatrix} x_{sp} \\ u_{sp} \end{pmatrix} = \begin{pmatrix} 0 \\ r_{sp} \end{pmatrix}. \tag{3.29}$$

Note that H relates the observed variables to the controlled variables. If there are more measured outputs than manipulated variables (3.29) cannot be solved directly. It is possible to cast (3.29) into an optimisation problem. We refer the reader to [48] for a full treatise on the subject.

3.4.3 Linear quadratic Gaussian control

The LQR problem dealt with deterministic systems where the states were known exactly. However, this is problematic from a practical perspective because:

1. The system model is almost never known exactly.
2. The state measurements are almost always noisy.

The linear quadratic Gaussian (LQG) controller for stochastic systems of the form

$$\begin{aligned} x_{t+1} &= Ax_t + Bu_t + w_t \\ y_t &= Cx_t + v_t \end{aligned} \tag{3.30}$$

was developed to handle these problems. Additionally $w_t \sim \mathcal{N}(0, W)$ and $v_t \sim \mathcal{N}(0, V)$ which are independent white noise terms. Because the states and measurements are stochastic variables we cannot use the LQR objective function as before. Instead we use the LQG objective function which is a generalisation of the former as shown in definition 3.27.

Definition 3.27. Linear quadratic Gaussian (LQG) objective function The controller minimising the quadratic objective function

$$V(x_0, \mathbf{u}) = \mathbb{E}\left[\frac{1}{2}\sum_{k=0}^{N-1}\left(x_k^T Q x_k + u_k^T R u_k\right) + \frac{1}{2}x_N^T P_f x_N\right] \tag{3.31}$$

is called the LQG controller. The restrictions on the tuning parameters are the same as before. It is assumed that y_0 is used to infer x_0 which is then used in the optimisation problem.

The full LQG control problem is

$$\begin{aligned}&\min_{\mathbf{u}} V(x_0, \mathbf{u}) = \mathbb{E}\left[\frac{1}{2}\sum_{k=0}^{N-1}\left(x_k^T Q x_k + u_k^T R u_k\right) + \frac{1}{2}x_N^T P_f x_N\right] \\ &\text{subject to } x_{t+1} = Ax_t + Bu_t + w_t.\end{aligned} \tag{3.32}$$

It is indeed possible to solve this controller analytically using stochastic dynamic programming but the derivation is tedious. We rather employ the separation principle [13]. We do however re-derive the optimal controller in a later chapter.

Definition 3.28. Separation principle The solution of the LQG problem is obtained by combining the solution of the deterministic LQR problem and the optimal state estimation problem. The optimal current state estimate is used as the current deterministic state within the framework of the LQR controller. This is also sometimes called certainty equivalence.

The optimal state estimate of linear systems under Gaussian noise is known as the Kalman filter. In later a chapter we devote much time to its derivation but for now we merely introduce it loosely.

Definition 3.29. Kalman filter The optimal linear state estimator for Gaussian random variables is called the Kalman filter.

A schematic diagram of the solution of LQG control problem is shown in Figure 3.5.

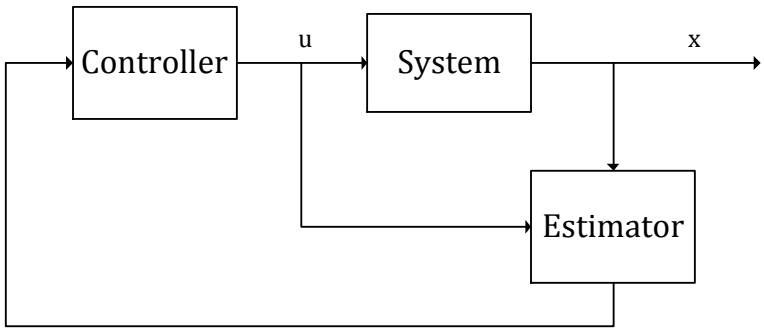

Figure 3.5: LQG control schematic. The estimator is the Kalman filter and the controller is the deterministic LQR.

The LQR and LQG controller solutions are two of the most fundamental results in optimal control theory [26]. In the next section we discuss model predictive control which is a generalisation of the controllers we have discussed so far.

3.4.4 Model predictive control

Model predictive control (MPC) is the constrained generalisation of the LQR controller. It is widely used industry and has been the subject of a significant amount of scholarly research. We introduce the classic deterministic linear MPC

$$\begin{aligned} &\min_{\mathbf{u}} V(x_0, \mathbf{u}) = \frac{1}{2}\sum_{k=0}^{N-1}\left(x_k^T Q x_k + u_k^T R u_k\right) + \frac{1}{2}x_N^T P_f x_N \\ &\text{subject to } x_{t+1} = Ax_t + Bu_t \\ &\text{and } d^T x_t + e \geq 0 \;\forall\; t = 1, 2, ..., N \end{aligned} \tag{3.33}$$

with coincidental control and prediction horizons N.

It is straightforward to incorporate more constraints of the form $d_i^T x_t + e_i \geq 0$ if required. The structure of (3.33) is important: the objective function is quadratic and the constraints are linear. There are two primary benefits of this structure. Firstly, the problem is provably convex which means that if a minimum is found it is the global minimum. Secondly, (and as a consequence of the first benefit) this allows for the use of specialised quadratic programming techniques which are fast and reliable.

Unfortunately there is no tractable way to analytically compute the control law offline, except in trivial cases, like the LQR controller. It is invariably necessary to use some optimisation algorithm to solve for $\mathbf{u}$ given x_0.

Recent advances in quadratic programming algorithms, like the interior point algorithm, have made it possible to solve quadratic programming problems almost as fast as linear programming problems. These solvers typically exploit the sparseness structure inherent in problems like (3.33) [39]. Thus, it is desirable to make use of these solvers wherever possible by ensuring that the underlying quadratic programming structure is not lost when modifications are made to the algorithm.

Finally, the stability and robustness (e.g. when there is plant/model mismatch) of deterministic linear MPC is well studied and understood in modern literature. See [48] and [39] for details. Thus, it is likewise desirable to exploit this knowledge rather than attempt to derive an MPC with completely different or new characteristics.

We will continue our study of linear MPC in later sections.

3.5 Matrix identities

It will be useful in a later section to have access to a block matrix inversion formula. We state the result without proof and refer the reader to [30] for more details.

Theorem 3.13. Block matrix inversion Suppose we have a square block matrix of the form $\begin{pmatrix} A & b \\ c^T & d \end{pmatrix}$ where A is an invertible matrix; b and c are conforming vectors; and d is a real number. Then the identity

$$\begin{pmatrix} A & b \\ c^T & d \end{pmatrix}^{-1} = \begin{pmatrix} A^{-1}(I + p^{-1}bc^TA^{-1}) & -p^{-1}A^{-1}b \\ -p^{-1}c^TA^{-1} & p^{-1} \end{pmatrix} \tag{3.34}$$

holds with $p = (d - c^TA^{-1}b)$.

Part II

Single model systems

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] F. Daum and J. Huang. Particle flow for nonlinear filters. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 5920–5923, May 2011.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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