Introduction to Estimation and the Kalman Filter

Hugh Durrant-Whyte
Australian Centre for Field Robotics
The University of Sydney NSW 2006
Australia
hugh@acfr.usyd.edu.au

January 2, 2001

Version 2.2

Contents

1 Introduction

Estimation is the problem of determining the value of an unknown quantity from one or more observations. The estimation problem arises whenever we use a 'sensor' to obtain measurements that relate to a quantity of interest. This may be, for example, the determination of the location of an aircraft on the basis of radar measurements of range and bearing, it may be the concentration of a particular chemical in a process plant on the basis of chemical, temperature and mass measurements, it may be the valuation of a bond on the basis of interest rate, yield and past demand. The estimation problem is central in linking the real world as observed by a sensor to the decisions we make about how to control or influence our environment.

At the heart of all estimation problems is the need to describe uncertainty. Clearly, if we could always take perfect measurements of a perfectly known process, then it would in principle be simple to decide what was happening in the real world. Of course this is never the case. The observations we make are always uncertain or noisy. They relate to process that is rarely known with confidence. Further, we very often do not observe the actual quantity of interest, rather the variables we observe only indirectly infer the value of this quantity. Throughout this course we use probabilistic methods to model uncertainty. A number of other uncertainty models could be used but the range of methods and applicability of the results that follow from using probabilistic models make this by far the most common modeling technique.

1.1 Course Outline

Broadly, this course has four parts. We shall first spend some time reviewing probabilistic modeling methods and see how these are used to describe uncertainty in both observations and in the processes of interest. We shall then look at a number of simple estimation methods and see how these are implemented in practice. The central part of this course is concerned with an estimation algorithm called the Kalman filter. The Kalman filter algorithm is the most widely used estimation algorithm in modern systems theory and finds application in almost every area of engineering. We will spend some time deriving and explaining the linear Kalman filter algorithm, and will detail how it is implemented and tested. The final part of this course will introduce the extended Kalman filter algorithm; a version of the linear Kalman filter which can be applied to problems in which the observation and process models are non-linear.

This course is intended as a practical introduction to estimation. As a consequence, there are a series of laboratories which require you to implement and test estimation, Kalman filter and extended Kalman filter algorithms. These laboratories are entirely based on the Matlab language. The laboratories are essential in understanding the estimation problem in general and in knowing how to apply the mathematics developed in this course to real problems of interest.

1.2 Bibliography

There are many good books on estimation theory. A number of the best ones and their contributions are listed here.

Maybeck [11] is an outstanding book who's value has not diminished with time. It is one of the few books that also deals with the practical issues of implementation, error budgets and filter tuning.

Barshalom and Fortman [4] focuses on issues in data association and tracking. There are subsequent books by Barshalom including [2, 3] which go further into these issues. The general development is unfortunately short of real implementation detail. The notation used in this course is based on that used in this book.

Brown and Hwang [5] is an excellent introductory book on estimation and Kalman filtering. It is a good starting point for a first introduction and considers many practical issues in implementation.

Gelb [8] is a very readable book with some interesting insights in to extended Kalman filtering. It also has some good summaries of the basic Kalman filtering algorithms.

Grewal and Andrews [9] covers a great deal of material not found elsewhere, particularly computational issues in both discrete and continuous time Kalman filtering.

Catlin [6] is interesting in addressing the geometric issues involved in estimation.

A mention should also be made of the special issue of the IEEE transactions on Automatic control on the applications of the Kalman filter [1]. This has a number of interesting and instructive examples of real Kalman filter implementations.

2 Models of Sensors, Processes and Uncertainty

To model sensors and processes we need a coherent method of describing uncertainty. Probability, and specifically probability distributions, provide this. We start with some revision of probability and stochastic processes, and briefly describe the use of Bayes theorem in estimation problems. With this we go on to describe models of processes and sensors using probability distributions.

2.1 Probability Distributions

For our purposes the central idea in probability is the concept of **likelihood**. In it's simplest terms a probability density function defined on the elements of a set $\mathcal{X}$ is simply a real valued function f_x which to every element $\mathbf{x} \in \mathcal{X}$ of this set assigns a real valued number: $f : \mathcal{X} \longrightarrow \Re$. The function f_x is a valid probability density function if;

1. f_x is positive

$$f_x(\mathbf{x}) \geq 0, \quad \forall \mathbf{x} \in \mathcal{X}. \tag{1}$$

2. The total probability mass assigned to the set $\mathcal{X}$ is 1; if $\mathbf{x}$ is a continuous-valued quantity then
$$\int_{-\infty}^{\infty} f_x(\mathbf{x})\mathrm{d}\mathbf{x} = 1, \tag{2}$$
or if $\mathbf{x}$ takes on discrete values then
$$\sum_{x\in\mathcal{X}} f_x(\mathbf{x}) = 1. \tag{3}$$

Strictly we should define a set of possible events, states or experimental outcomes, and associated with these a random variable on which the function f_x is defined. The probability density function f_x then models the relative likelihood of occurrence of different values of $\mathbf{x} \in \mathcal{X}$, so that if $f_x(\mathbf{x}_i) > f_x(\mathbf{x}_j)$ we will simply say that the value $\mathbf{x}_i$ is more likely than the value $\mathbf{x}_j$. This interpretation of a probability density function exposes a most useful property of probability models; we can compare two events, and decide which is most preferable, without having to directly compare their values.

It is of course possible to define a probability density function on several random variables simultaneously in the from of a **joint probability density function**. With $\mathbf{x} \in \mathcal{X}$ and $\mathbf{y} \in \mathcal{Y}$ the probability density function f_{xy} is a function mapping pairs of $\mathbf{x}$ and $\mathbf{y}$ to a real number; $f_{xy} : \mathcal{X} \times \mathcal{Y} \longrightarrow \Re$. f_{xy} is a valid probability density function if

1. $f_{x,y}$ is positive, i.e.
$$f_{xy}(\mathbf{xy}) \geq 0, \quad \forall \mathbf{x} \in \mathcal{X}, \mathbf{y} \in \mathcal{Y}. \tag{4}$$
2. The total probability mass assigned to the set $\mathcal{X} \times \mathcal{Y}$ is 1; if $\mathbf{x}$ and $\mathbf{y}$ are continuous-valued quantities then
$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f_{xy}(\mathbf{x},\mathbf{y})\mathrm{d}\mathbf{x}\mathrm{d}\mathbf{y} = 1, \tag{5}$$
or if $\mathbf{x}$ takes on discrete values then
$$\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} f_{xy}(\mathbf{x},\mathbf{y}) = 1. \tag{6}$$

We can now interpret f_{xy} as assigning a relative likelihood to each possible pair of outcomes $(\mathbf{x}, \mathbf{y})$. Density functions for three or more random variables follow analogously.

Given a joint probability density function f_{xy} defined on $\mathcal{X} \times \mathcal{Y}$ we can compute the **marginal probability density function** in either variable by simply integrating or summing over the remaining variable; if $\mathbf{x}$ and $\mathbf{y}$ are continuous variables then
$$f_x(\mathbf{x}) = \int_{-\infty}^{\infty} f_{xy}(\mathbf{x},\mathbf{y})\mathrm{d}\mathbf{y}, \qquad f_y(\mathbf{y}) = \int_{-\infty}^{\infty} f_{xy}(\mathbf{x},\mathbf{y})\mathrm{d}\mathbf{x}, \tag{7}$$
and if $\mathbf{x}$ and $\mathbf{y}$ are discrete variables then
$$f_x(\mathbf{x}) = \sum_{y\in\mathcal{Y}} f_{xy}(\mathbf{x},\mathbf{y}), \qquad f_y(\mathbf{y}) = \sum_{x\in\mathcal{X}} f_{xy}(\mathbf{x},\mathbf{y}). \tag{8}$$

It is well known however that the converse is not true; given the marginal densities in each random variable, it is not in general possible to construct the joint probability density function of all random variables together;

$$f_{xy}(\mathbf{x}, \mathbf{y}) \neq f_x(\mathbf{x}) f_y(\mathbf{y}) \tag{9}$$

This has serious implications in sensing and estimation problems because in principle given a set of random variables $\{\mathcal{X}_1, \cdots, \mathcal{X}_n\}$ it requires that we define a joint probability density function $f_{x_1,\cdots,x_n}$ over all possible combinations of outcomes $\{\mathbf{x}_1 \in \mathcal{X}_1, \cdots, \mathbf{x}_n \in \mathcal{X}_n\}$. In most cases there are too many combinations for this approach to modeling to be tractable. However, it is very often the case that the number of combinations on which the joint probability density function need be defined can be considerably reduced by employing the idea of independence or conditional independence.

2.1.1 Conditional Probability Distributions

The **conditional probability density function** $f_{x|y}$ of a random variable $\mathbf{x}$ conditioned on values of a second random variable $\mathbf{y}$ is defined by

$$f_{x|y}(\mathbf{x} \mid \mathbf{y}) \stackrel{\triangle}{=} \frac{f_{xy}(\mathbf{x}, \mathbf{y})}{f_y(\mathbf{y})}, \tag{10}$$

and should be interpreted as a standard probability density function satisfying the usual conditions which describes the probability or likelihood associated with each possible outcome $\mathbf{x} \in \mathcal{X}$ when the random variable $\mathbf{y} \in \mathcal{Y}$ takes on a fixed known value. We will often say that $f_{x|y}$ describes the probability or likelihood of $\mathbf{x}$ *given* $\mathbf{y}$. Equation 10 is often expressed in the from of the 'chain-rule' of conditional probabilities as

$$\begin{aligned} f_{xy}(\mathbf{x}, \mathbf{y}) &= f_{x|y}(\mathbf{x} \mid \mathbf{y}) f_y(\mathbf{y}) \\ &= f_{y|x}(\mathbf{y} \mid \mathbf{x}) f_x(\mathbf{x}). \end{aligned} \tag{11}$$

The expansion of a joint probability density function as a product of conditional distributions can be exploited directly in the calculation of marginal distributions. For continuous-valued random variables, substituting Equation 11 in to Equation 7 we obtain

$$f_x(\mathbf{x}) = \int_{-\infty}^{\infty} f_{x|y}(\mathbf{x} \mid \mathbf{y}) f_y(\mathbf{y}) \mathrm{d}\mathbf{y}. \tag{12}$$

For discrete-valued random variables, substituting Equation 11 in to Equation 8 we obtain

$$f_x(\mathbf{x}) = \sum_{\mathbf{y} \in \mathcal{Y}} f_{x|y}(\mathbf{x} \mid \mathbf{y}) f_y(\mathbf{y}). \tag{13}$$

This is sometimes called the **total probability theorem**. The decomposition also has an interesting and important interpretation as a rule for "assumption-based reasoning"; the likelihood of any event $\mathbf{x} \in \mathcal{X}$ is a weighted sum over the likelihoods in all the distinct ways that $\mathbf{x}$ might be realized where the weights are simply the likelihoods associated with each possible realization. This is a technique exploited in expert systems.

2.1.2 Independence and Conditional Independence

If it happens that knowledge of the value of $\mathbf{y}$ does not give us any more information about the value of $\mathbf{x}$ then $\mathbf{x}$ and $\mathbf{y}$ are said to be independent as

$$f_{x|y}(\mathbf{x} \mid \mathbf{y}) = f_x(\mathbf{x}). \tag{14}$$

With Equation 14 substituted into Equation 11 we obtain

$$f_{xy}(\mathbf{x}, \mathbf{y}) = f_x(\mathbf{x}) f_y(\mathbf{y}). \tag{15}$$

That is, the one exception to Equation 9 is when the two random variables are independent.

A weaker form of independence can be defined through the important idea of conditional independence. Given three random variables $\mathbf{x} \in \mathcal{X}$, $\mathbf{y} \in \mathcal{Y}$, and $\mathbf{z} \in \mathcal{Z}$ we can define the conditional distribution of $\mathbf{x}$ given both $\mathbf{y}$ and $\mathbf{z}$ as $f(\mathbf{x} \mid \mathbf{y}, \mathbf{z})$. If knowledge of the value of $\mathbf{z}$ makes the value of $\mathbf{x}$ independent of the value of $\mathbf{y}$ then we may write

$$f(\mathbf{x} \mid \mathbf{y}, \mathbf{z}) = f(\mathbf{x} \mid \mathbf{z}). \tag{16}$$

This may be the case for example if $\mathbf{z}$ indirectly contains all the information contributed by $\mathbf{y}$ to the value of $\mathbf{x}$. It is important to realize that the statement that $\mathbf{x}$ is conditionally independent on $\mathbf{y}$ given $\mathbf{z}$ is quite different from the statement that $\mathbf{x}$ is unconditionally independent on $\mathbf{y}$ as in Equation 14.

Conditional independence can be exploited in a number of different ways. In particular, applying the chain-rule to the joint probability density function on three random variables $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{z}$ we have

$$\begin{aligned} f(\mathbf{x}, \mathbf{y}, \mathbf{z}) &= f(\mathbf{x}, \mathbf{y} \mid \mathbf{z}) f(\mathbf{z}) \\ &= f(\mathbf{x} \mid \mathbf{y}, \mathbf{z}) f(\mathbf{y} \mid \mathbf{z}) f(\mathbf{z}), \end{aligned} \tag{17}$$

together with the conditional independence result of Equation 16 we obtain the intuitive result

$$f(\mathbf{x}, \mathbf{y} \mid \mathbf{z}) = f(\mathbf{x} \mid \mathbf{z}) f(\mathbf{y} \mid \mathbf{z}). \tag{18}$$

That is, if $\mathbf{x}$ is independent of $\mathbf{y}$ given knowledge of $\mathbf{z}$ then the joint probability density function of $\mathbf{x}$ and $\mathbf{y}$ conditioned on $\mathbf{z}$ is simply the product of the marginal distributions of $\mathbf{x}$ and $\mathbf{y}$ each conditioned on $\mathbf{z}$, analogously to Equation 15.

2.2 Bayes Theorem

We come now to arguably the most important result in the study of probabilistic models: Bayes theorem. Consider two random variables $\mathbf{x} \in \mathcal{X}$ and $\mathbf{z} \in \mathcal{Z}$ on which is defined a

joint probability density function $f(\mathbf{x}, \mathbf{z})$. The chain-rule of conditional probabilities can be used to expand this density function in two ways

$$\begin{aligned} f(\mathbf{x}, \mathbf{z}) &= f(\mathbf{x} \mid \mathbf{z}) f(\mathbf{z}) \\ &= f(\mathbf{z} \mid \mathbf{x}) f(\mathbf{x}). \end{aligned} \tag{19}$$

Rearranging in terms of one of the conditional densities, we obtain Bayes theorem

$$f(\mathbf{x} \mid \mathbf{z}) = \frac{f(\mathbf{z} \mid \mathbf{x}) f(\mathbf{x})}{f(\mathbf{z})}. \tag{20}$$

The value of this result lies in the interpretation of the probability density functions $f(\mathbf{x} \mid \mathbf{z})$, $f(\mathbf{z} \mid \mathbf{x})$, and $f(\mathbf{x})$. Suppose we start off by wanting to determine the various likelihoods of different values of an unknown state of nature $\mathbf{x} \in \mathcal{X}$. We may have prior beliefs about what values of $\mathbf{x}$ we might expect to see and these are encoded in the form of relative likelihoods in the **prior probability density function** $f(\mathbf{x})$. To obtain more information about the state $\mathbf{x}$ we gather some information or take an observation $\mathbf{z} \in \mathcal{Z}$. We model the observations made as a conditional probability density function $f(\mathbf{z} \mid \mathbf{x})$ which describes, for each fixed state of nature $\mathbf{x} \in \mathcal{X}$, the likelihood that we will make the observation $\mathbf{z} \in \mathcal{Z}$; the probability of $\mathbf{z}$ given $\mathbf{x}$. What we now want is to compute the new likelihoods associated with the state of nature $\mathbf{x}$ given both the original prior information and the information we gain by observation. This is encoded in the **posterior distribution** $f(\mathbf{x} \mid \mathbf{z})$ which describes the likelihoods associated with $\mathbf{x}$ *given* the observation $\mathbf{z}$. The marginal distribution $f(\mathbf{z})$ simply serves to normalize the posterior and may, if required, be computed from

$$f(\mathbf{z}) = \int_{-\infty}^{\infty} f(\mathbf{z} \mid \mathbf{x}) f(\mathbf{x}) \mathrm{d}\mathbf{x} \tag{21}$$

in the case when $\mathbf{x}$ is a continuous random variable, and from

$$f(\mathbf{z}) = \sum_{\mathbf{x} \in \mathcal{X}} f(\mathbf{z} \mid \mathbf{x}) f(\mathbf{x}) \tag{22}$$

in the case when $\mathbf{x}$ takes discrete values. The value of Bayes theorem is now clear, it provides a direct means of combing observed information with prior beliefs about the state of the world. Unsurprisingly, because of this, Bayes theorem lies at the heart of many estimation algorithms.

2.2.1 Estimation using Bayes Theorem

It is possible to apply Bayes theorem directly to the integration of observations emanating from several different sources. Suppose we now have a number of pieces of information

$\mathbf{Z}^n \triangleq \{\mathbf{z}_1 \in \mathcal{Z}_i, \cdots, \mathbf{z}_n \in \mathcal{Z}_n\}$, and we wish to use this information to construct a posterior distribution $f(\mathbf{x} \mid \mathbf{Z}^n)$ describing the relative likelihoods in the various values of the state of interest $\mathbf{x} \in \mathcal{X}$ given the information obtained. In principle we can employ Bayes theorem directly to compute this distribution function from

$$f(\mathbf{x} \mid \mathbf{Z}^n) = \frac{f(\mathbf{Z}^n \mid \mathbf{x}) f(\mathbf{x})}{f(\mathbf{Z}^n)} = \frac{f(\mathbf{z}_1, \cdots, \mathbf{z}_n \mid \mathbf{x}) f(\mathbf{x})}{f(\mathbf{z}_1, \cdots, \mathbf{z}_n)}. \tag{23}$$

In particular it would be difficult to do this because it requires that we know the distribution $f(\mathbf{z}_1, \cdots, \mathbf{z}_n \mid \mathbf{x})$ completely; that is, the joint distribution of all combinations of observations conditioned on the underlying state. However, it is usually quite reasonable to assume that *given* the true state $\mathbf{x} \in \mathcal{X}$ that the information obtained from the i^{th} information source is independent of the information obtained from other sources. The validity of this assumption is discussed below. With this assumption, Equation 16 implies that

$$f(\mathbf{z}_i \mid \mathbf{x}, \mathbf{z}_1, \cdots, \mathbf{z}_{i-1}, \mathbf{z}_{i+1}, \mathbf{z}_n) = f(\mathbf{z}_i \mid \mathbf{x}), \tag{24}$$

and from Equation 18 this gives

$$f(\mathbf{z}_1, \cdots, \mathbf{z}_n \mid \mathbf{x}) = f(\mathbf{z}_1 \mid \mathbf{x}) \cdots f(\mathbf{z}_n \mid \mathbf{x}) = \prod_{i=1}^{n} f(\mathbf{z}_i \mid \mathbf{x}). \tag{25}$$

Substituting this back into Equation 23 we obtain

$$f(\mathbf{x} \mid \mathbf{Z}^n) = [f(\mathbf{Z}^n)]^{-1} f(\mathbf{x}) \prod_{i=1}^{n} f(\mathbf{z}_i \mid \mathbf{x}). \tag{26}$$

Thus the updated likelihoods in the state, the posterior distribution on $\mathbf{x}$, is simply proportional to the product of prior likelihood and individual likelihoods from each information source. The marginal distribution $f(\mathbf{Z}^n)$ simply acts as a normalizing constant and may be computed, if required, in the case of continuous random variables from

$$f(\mathbf{Z}^n) = \int_{-\infty}^{\infty} f(\mathbf{Z}^n \mid \mathbf{x}) f(\mathbf{x}) \mathrm{d}\mathbf{x}, \tag{27}$$

and in the case of discrete random variables from

$$f(\mathbf{Z}^n) = \sum_{\mathbf{x} \in \mathcal{X}} f(\mathbf{Z}^n \mid \mathbf{x}) f(\mathbf{x}). \tag{28}$$

Equation 26 provides a simple and direct mechanism for computing the relative likelihood in different values of a state from any number of observations or other pieces of information. The following example demonstrates the basic principles in applying Bayes theorem to the problem of estimation.

Example 1

We will consider a simple example of the application of Bayes theorem to estimating a discrete parameter on the basis of one observation and some prior information. The environment of interest is modeled by a single state $\mathbf{x}$ *which can take on one of three values:*

$\mathbf{x}_1$*:* $\mathbf{x}$ *is a type 1 target.*

$\mathbf{x}_2$*:* $\mathbf{x}$ *is a type 2 target.*

$\mathbf{x}_3$*: No visible target.*

A single sensor observes $\mathbf{x}$ *and returns three possible values:*

$\mathbf{z}_1$*: Observation of a type 1 target.*

$\mathbf{z}_2$*: Observation of a type 2 target.*

$\mathbf{z}_3$*: No target observed.*

The sensor model is described by the likelihood matrix $f_1(\mathbf{z}_i \mid \mathbf{x}_j)$*:*

	$\mathbf{z}_1$	$\mathbf{z}_2$	$\mathbf{z}_3$
$\mathbf{x}_1$	0.45	0.45	0.1
$\mathbf{x}_2$	0.45	0.45	0.1
$\mathbf{x}_3$	0.1	0.1	0.8

The posterior distribution of the true state $\mathbf{x}$ *after making an observation* $\mathbf{z}_i$ *is given by*

$$f(\mathbf{x}_j \mid \mathbf{z}_i) = \alpha f_1(\mathbf{z}_i \mid \mathbf{x}_j) f(\mathbf{x}_j)$$

where α *is simply a normalizing constant set by requiring the sum, over* j*, of posterior probabilities to be equal to 1.*

In the first instance we will assume that we do not have any prior information about the possible likelihood of target types 1 and 2, and so we set the prior probability vector to $f(\mathbf{x}) = (0.333, 0.333, 0.333)$*. If we now observe* $\mathbf{z}_1$*, then clearly the posterior distribution will be given by* $f_1(\mathbf{x}i \mid \mathbf{z}_1) = (0.45, 0.45, 0.1)$ *(i.e. just the likelihood function given* $\mathbf{z}_1$ *has been observed).*

If now we subsequently use this posterior distribution as the prior for a new observation $f(\mathbf{x}_i) = (0.45, 0.45, 0.1)$*, and we again make the observation* $\mathbf{z}_1$*, then the new posterior distribution will be given by* $f_1(\mathbf{x} \mid \mathbf{z}_1) = (0.488, 0.488, 0.024)$ *(i.e. the element-wise product of the likelihood function given that* $\mathbf{z}_1$ *has been observed with the new prior probability).*

Notice that the result of this integration process is to increase the probability in both type 1 and type 2 targets at the expense of the no-target hypothesis. Clearly, although this sensor is good at detecting targets, it is not good at distinguishing between targets of different types.

We now consider a second sensor which makes the same three observations as the first

sensor, but whose likelihood matrix $f_2(\mathbf{z}_i \mid \mathbf{x}_j)$ *is described by*

	$\mathbf{z}_1$	$\mathbf{z}_2$	$\mathbf{z}_3$
$\mathbf{x}_1$	0.45	0.1	0.45
$\mathbf{x}_2$	0.1	0.45	0.45
$\mathbf{x}_3$	0.45	0.45	0.1

Whereas our first sensor was good at detecting targets but not at distinguishing between different target types, this second sensor has poor overall detection probabilities but good target discrimination capabilities. So for example, with a uniform prior, if we observe $\mathbf{z}_1$ *with this second sensor, the posterior distribution on possible true states will be given by* $f_2(\mathbf{x}_j \mid \mathbf{z}_1) = (0.45, 0.1, 0.45)$.

It clearly makes sense to combine the information from both sensors to provide a system with both good detection and good discrimination capabilities. From Equation 26, the product of the two likelihood functions gives us an overall likelihood function for the combined system as $f_{12}(\mathbf{z}_i, \mathbf{z}_j \mid \mathbf{x}_k) = f_1(\mathbf{z}_i \mid \mathbf{x}_k) f_2(\mathbf{z}_j \mid \mathbf{x}_k)$. *Thus if we observe* $\mathbf{z}_1$ *using the first sensor, then the posterior likelihood in event* $\mathbf{x}_1$ *is given by*

$$\begin{aligned} f_{12}(\mathbf{z}_1, \mathbf{z}_j \mid \mathbf{x}_1) &= f_2(\mathbf{z}_j \mid \mathbf{x}_1) f_1(\mathbf{z}_1 \mid \mathbf{x}_1) \\ &= (0.45, 0.1, 0.45) \times 0.45 \\ &= (0.2025, 0.045, 0.2025) \end{aligned}$$

Repeating this calculation for each $\mathbf{z}_i$, $\mathbf{z}_j$, $\mathbf{x}_k$ *triple we have the combined likelihood matrix*

$\mathbf{z}_j = \mathbf{z}_1$	$\mathbf{z}_1$	$\mathbf{z}_2$	$\mathbf{z}_3$
$\mathbf{x}_1$	0.2025	0.045	0.2025
$\mathbf{x}_2$	0.045	0.2025	0.2025
$\mathbf{x}_3$	0.045	0.045	0.01

$\mathbf{z}_j = \mathbf{z}_2$	$\mathbf{z}_1$	$\mathbf{z}_2$	$\mathbf{z}_3$
$\mathbf{x}_1$	0.2025	0.045	0.2025
$\mathbf{x}_2$	0.045	0.2025	0.2025
$\mathbf{x}_3$	0.045	0.045	0.01

$\mathbf{z}_j = \mathbf{z}_3$	$\mathbf{z}_1$	$\mathbf{z}_2$	$\mathbf{z}_3$
$\mathbf{x}_1$	0.045	0.01	0.045
$\mathbf{x}_2$	0.01	0.045	0.045
$\mathbf{x}_3$	0.36	0.36	0.08

The combined sensor provides substantial improvements in overall system performance[1] *. If for example we observe target 1 with the first sensor (the array block* $\mathbf{z}_j = \mathbf{z}_1$*) and again observe target 1 with the second sensor (the first column of this block), then the posterior distribution in the three hypotheses is*

$$f(\mathbf{x}_k \mid \mathbf{z}_1, \mathbf{z}_2) = \alpha \times (0.2025, 0.045, 0.045) = (0.692, 0.154, 0.154),$$

and so target 1 is clearly the most probable target. If however, we observe a type 2 target with the second sensor after having observed a type 1 target with the first sensor, a similar calculation gives the posterior as $(0.154, 0.692, 0.154)$*, that is target type 2 has high probability. This is because although sensor 1 observed a type 1 target, the likelihood function for sensor 1 tells us that it is poor at distinguishing between target types and so sensor 2 information is used for this purpose. If now we observe no target with sensor 2, having detected target type 1 with the first sensor, the posterior given both observations is given by* $(0.488, 0.488, 0.024)$*. That is we still believe that there is a target (because we know sensor 1 is much better at target detection than sensor 2), but we still have no idea which of target 1 or 2 it is as sensor 2 has been unable to make a valid detection. The analysis for sensor 1 detecting target 2 is identical to that for detection of target 1. Finally, if sensor 1 gets no detection, but sensor 2 detects target type 1, then the posterior likelihood is given by* $(0.108, 0.024, 0.868)$*. That is we still believe there is no target because we know sensor 1 is better at providing this information (and perversely, sensor 2 confirms this even though it has detected target type 1).*

The effectiveness of Equation 26 relies crucially on the assumption that the information obtained from different information sources are independent when conditioned on the true underlying state of the world; as defined in Equation 24. It would be right to question if this assumption is reasonable. It is clearly unreasonable to state that the information obtained is unconditionally independent;

$$f(\mathbf{z}_1, \cdots, \mathbf{z}_n) \neq f(\mathbf{z}_1) \cdots f(\mathbf{z}_n), \tag{29}$$

[1]Note that when summing over any two events or observations, the columns and rows still come to 1. In practical implementations, it is often sufficient to encode relative likelihoods of different events and to normalize only when computing the posterior distribution.

because each piece of information depends on a *common* underlying state $\mathbf{x} \in \mathcal{X}$. If it were the case that the information obtained where independent of this state, and therefore unconditionally independent of other information sources, there would be little value in using it to improve our knowledge of the state. It is precisely because the information obtained is unconditionally dependent on the underlying state that it has value as an information source. Conversely, it is quite reasonable to assume that the underlying state is the *only* thing in common between information sources and so once the state has been specified it is correspondingly reasonable to assume that the information gathered is conditionally independent given this state. There are two subtleties in this statement which should be clarified. If any residual aspects of the environment model or underlying state remain in the description of the information sources, the individual pieces of information obtained will not be conditionally independent because they will have this residual component in common. It is possible to conceive of cases in which information sources are not conditionally independent. This will be the case when the information gathered depends on a common effect which itself is conditionally independent on the state; if, for example, the information gathered derives from a single common source, or if the information gathering process itself physically affects the state of the environment (through irradiation or contact). The only recourse in such situations is to explicitly account for such effects and to incorporate them directly in any probabilistic model of the information source.

2.2.2 Recursive Estimation Using Bayes Theorem

The integration of information using Equation 26 would, in principle require that we remember all past information and, on arrival of new information in the form $f(\mathbf{z}_k \mid \mathbf{x})$, recompute the total likelihood based on all information gathered up to this time. However, Bayes theorem, in the form of Equation 26, also lends itself to the incremental or recursive addition of new information in determining a revised posterior distribution on the state. With $\mathbf{Z}^k \triangleq \{\mathbf{z}_k, \mathbf{Z}^{k-1}\}$ we have

$$\begin{aligned} f(\mathbf{x}, \mathbf{Z}^k) &= f(\mathbf{x} \mid \mathbf{Z}^k) f(\mathbf{Z}^k) \\ &= f(\mathbf{z}_k, \mathbf{Z}^{k-1} \mid \mathbf{x}) f(\mathbf{x}) = f(\mathbf{z}_k \mid \mathbf{x}) f(\mathbf{Z}^{k-1} \mid \mathbf{x}) f(\mathbf{x}), \end{aligned} \tag{30}$$

where we have assumed conditional independence of the observation sequence. Equating both sides of this expansion, we obtain

$$\begin{aligned} f(\mathbf{x} \mid \mathbf{Z}^k) f(\mathbf{Z}^k) &= f(\mathbf{z}_k \mid \mathbf{x}) f(\mathbf{Z}^{k-1} \mid \mathbf{x}) f(\mathbf{x}) \\ &= f(\mathbf{z}_k \mid \mathbf{x}) f(\mathbf{x} \mid \mathbf{Z}^{k-1}) f(\mathbf{Z}^{k-1}). \end{aligned} \tag{31}$$

Noting that $f(\mathbf{Z}^k)/f(\mathbf{Z}^{k-1}) = f(\mathbf{z}_k \mid \mathbf{Z}^{k-1})$ and rearranging gives

$$f(\mathbf{x} \mid \mathbf{Z}^k) = \frac{f(\mathbf{z}_k \mid \mathbf{x}) f(\mathbf{x} \mid \mathbf{Z}^{k-1})}{f(\mathbf{z}_k \mid \mathbf{Z}^{k-1})}. \tag{32}$$

The advantage of Equation 32 is that we need only compute and store the posterior likelihood $f(\mathbf{x} \mid \mathbf{Z}^{k-1})$ which contains a complete summary of all past information. When the next piece of information $f(\mathbf{z}_k \mid \mathbf{x})$ arrives, the previous posterior takes on the role of the current prior and the product of the two becomes, when normalized, the new posterior. Equation 32 thus provides a significant improvement in computational and memory requirements over Equation 26.

Example 2

An important example of the recursive application of Bayes theorem is in the calculation of the posterior distribution of a scalar $\mathbf{x}$ *under the assumption that the likelihood function for the observations given the true state is Gaussian with known variance* σ^2*;*

$$f(\mathbf{z}_k \mid \mathbf{x}) = \frac{1}{\sqrt{2\pi}\sigma^2} \exp\left(-\frac{1}{2}\frac{(\mathbf{z}_k - \mathbf{x})^2}{\sigma^2}\right).$$

If we assume that the posterior distribution in $\mathbf{x}$ *after taking the first* $k-1$ *observations is also Gaussian with mean* $\mathbf{x}_{k-1}$ *and variance* σ^2_{k-1}*,*

$$f(\mathbf{x} \mid \mathbf{Z}^{k-1}) = \frac{1}{\sqrt{2\pi}\sigma^2_{k-1}} \exp\left(-\frac{1}{2}\frac{(\mathbf{x}_{k-1} - \mathbf{x})^2}{\sigma^2_{k-1}}\right).$$

then the posterior distribution in $\mathbf{x}$ *after the first* k *observations is given by*

$$\begin{aligned} f(\mathbf{x} \mid \mathbf{Z}^k) &= K \tfrac{1}{\sqrt{2\pi}\sigma^2} \exp\left(-\tfrac{1}{2}\tfrac{(\mathbf{Z}_k - \mathbf{X})^2}{\sigma^2}\right) . \tfrac{1}{\sqrt{2\pi}\sigma^2_{k-1}} \exp\left(-\tfrac{1}{2}\tfrac{(\mathbf{X}_{k-1} - \mathbf{X})^2}{\sigma^2_{k-1}}\right) \\ &= \tfrac{1}{\sqrt{2\pi}\sigma^2_k} \exp\left(-\tfrac{1}{2}\tfrac{(\mathbf{X}_k - \mathbf{X})^2}{\sigma^2_k}\right) \end{aligned}$$

where K *is a constant independent of* $\mathbf{x}$ *chosen to ensure that the posterior is appropriately normalized, and* $\mathbf{x}_k$ *and* σ^2_k *are given by*

$$\mathbf{x}_k = \frac{\sigma^2_{k-1}}{\sigma^2_{k-1} + \sigma^2}\mathbf{z}_k + \frac{\sigma^2}{\sigma^2_{k-1} + \sigma^2}\mathbf{x}_{k-1}, \tag{33}$$

and

$$\sigma^2_k = \frac{\sigma^2\sigma^2_{k-1}}{\sigma^2 + \sigma^2_{k-1}} \tag{34}$$

The most important point to note about the posterior is that it too is a Gaussian; the product of two Gaussian distributions is itself Gaussian. Distributions that have this symmetry property are known as conjugate distributions. Given this property, it is clearly not necessary to go through the process of multiplying distributions together as it is sufficient to simply compute the new mean and variance of the posterior from Equations 33 and 34 as these completely characterize the associated Gaussian distribution.

2.3 The Expectation Operator

The expectation $\mathrm{E}[\mathbf{x}]$ or mean of a random variable $\mathbf{x}$ is simply defined as the first moment of the distribution $f(\mathbf{x})$. The expected value is often used as an estimate of the true value of $\mathbf{x}$, and under certain circumstances is the best value to choose as an estimate when no other information is made available.

For continuous-valued random variables, the expected value is defined as

$$\mathrm{E}[\mathbf{x}] \triangleq \int_{-\infty}^{\infty} \mathbf{x} f(\mathbf{x}) \mathrm{d}\mathbf{x} \tag{35}$$

and for discrete-valued random variables as

$$\mathrm{E}[\mathbf{x}] \triangleq \sum_{\mathbf{x} \in \mathcal{X}} \mathbf{x} f(\mathbf{x}). \tag{36}$$

The conditional expectation $\mathrm{E}[\mathbf{x} \mid \mathbf{y}]$ of a random variable $\mathbf{x}$ given that the related variable $\mathbf{y}$ takes a specific value, is defined in a similar manner. For continuous-valued random variables

$$\mathrm{E}[\mathbf{x} \mid \mathbf{y}] = \int_{-\infty}^{\infty} \mathbf{x} f(\mathbf{x} \mid \mathbf{y}) \mathrm{d}\mathbf{x} \tag{37}$$

and for discrete-valued random variables

$$\mathrm{E}[\mathbf{x} \mid \mathbf{y}] = \sum_{\mathbf{x} \in \mathcal{X}} \mathbf{x} f(\mathbf{x} \mid \mathbf{y}). \tag{38}$$

It is also possible to define the expected value of a function of a random variable. For continuous-valued random variables

$$\mathrm{E}[G(\mathbf{x})] = \int_{-\infty}^{\infty} G(\mathbf{x}) f(\mathbf{x}) \mathrm{d}\mathbf{x} \tag{39}$$

and for discrete-valued random variables

$$\mathrm{E}[G(\mathbf{x})] = \sum_{\mathbf{x} \in \mathcal{X}} G(\mathbf{x}) f(\mathbf{x}) \tag{40}$$

Of note are $\mathrm{E}[\mathbf{x}^n]$, the n^{th} moment of the distribution $f(\mathbf{x})$, and $\mathrm{E}[(\mathbf{x} - \mathrm{E}[\mathbf{x}])^n]$, the n^{th} central moment. In particular, when $\mathbf{x}$ is a scalar, the first moment $\overline{\mathbf{x}} = \mathrm{E}[\mathbf{x}]$ is the mean, and the second central moment $\sigma^2 = \mathrm{E}[(\mathbf{x} - \overline{\mathbf{x}})^2]$ is the variance of the distribution. When $\mathbf{x}$ is a vector, the variance is defined as $\mathbf{\Sigma} = \mathrm{E}[(\mathbf{x} - \overline{\mathbf{x}})(\mathbf{x} - \overline{\mathbf{x}})^T]$.

Expectation is a linear operator. For example with $G(\cdot)$ and $H(\cdot)$ two functions of $\mathbf{x}$, and A and B two constants, we have

$$\mathrm{E}[AG(\mathbf{x}) + BH(\mathbf{x})] = A\mathrm{E}[G(\mathbf{x})] + B\mathrm{E}[H(\mathbf{x})] . \tag{41}$$

The smoothing property of expectations states that the expected value of a conditional expected value is the (unconditional) expected value:

$$\begin{aligned} \mathrm{E}[\mathrm{E}[\mathbf{x} \mid \mathbf{y}]\ ] &= \int_{-\infty}^{\infty} \left[\int_{-\infty}^{\infty} \mathbf{x} f(\mathbf{x} \mid \mathbf{y}) \mathrm{d}\mathbf{x}\right] f(\mathbf{y}) \mathrm{d}\mathbf{y} \\ &= \int_{-\infty}^{\infty} \mathbf{x} \left[\int_{-\infty}^{\infty} f(\mathbf{x}, \mathbf{y}) \mathrm{d}\mathbf{y}\right] \mathrm{d}\mathbf{x} \\ &= \int_{-\infty}^{\infty} \mathbf{x} f(\mathbf{x}) \mathrm{d}\mathbf{x} = \mathrm{E}[\mathbf{x}]\ . \end{aligned} \tag{42}$$

Here, the inner expectation is taken as a function of $\mathbf{y}$, and this is averaged out by the outer expectation. One consequence of this is that the expected value of a state tends to be "smoother" than its true value.

2.4 Correlations and Power Spectra

In most situations of interest we will be interested not just in static random variables $\mathbf{x}$, but in stochastic processes $\mathbf{x}(t)$. In simple terms a stochastic process is simply a random variable that is a function of time. It can either be thought of as a random function of time, or alternatively as a random variable dependent both on a random event and on a time index. A stochastic process can be both discrete or continuous as both a random variable and a time function. We will most often be interested in continuous random variables which are continuous functions of time and so we use these exclusively in the following. Equivalent results may also be obtained for discrete random processes.

2.4.1 Autocorrelation and Autocovariance

For the purposes of modeling sensors and environments we are interested in being able to characterize a stochastic process in a compact and useful way. We can start by simply computing the mean and variance of a random process

$$\overline{\mathbf{x}}(t) \triangleq \mathrm{E}[\mathbf{x}(t)]\ = \int_{-\infty}^{\infty} \mathbf{x}(t) f(\mathbf{x}(t)) \mathrm{d}\mathbf{x}(t), \tag{43}$$

$$\mathbf{P}_{xx}(t) \triangleq \mathrm{E}[(\mathbf{x}(t) - \overline{\mathbf{x}}(t))(\mathbf{x}(t) - \overline{\mathbf{x}}(t))^T]\ = \int_{-\infty}^{\infty} (\mathbf{x}(t) - \overline{\mathbf{x}}(t))(\mathbf{x}(t) - \overline{\mathbf{x}}(t))^T f(\mathbf{x}(t)) \mathrm{d}\mathbf{x}(t), \tag{44}$$

so the mean, for example, becomes a simple (non-random) time function. However, there are a number of other properties of a stochastic process which it is important to characterize. In particular, it is useful to know how current values of the process $\mathbf{x}(t)$ depend on past values of the process $\mathbf{x}(t')$, $t' < t$. A measure of this dependence is the *autocorrelation kernel* $\mathbf{R}_{xx}(t, t')$:

$$\mathbf{R}_{xx}(t, t') \triangleq \mathrm{E}[\mathbf{x}(t)\mathbf{x}^T(t')]\ = \int_{-\infty}^{\infty} \mathbf{x}(t)\mathbf{x}^T(t') f(\mathbf{x}(t)) \mathrm{d}\mathbf{x}(t) \tag{45}$$

The autocorrelation of a stochastic process measures the correlation between two parts of the process at two different times. A related measure is the *autocovariance kernel* $\mathbf{P}_{xx}(t,t')$:

$$\begin{aligned}\mathbf{P}_{xx}(t,t') &\triangleq \mathrm{E}[(\mathbf{x}(t)-\overline{\mathbf{x}}(t))(\mathbf{x}(t')-\overline{\mathbf{x}}(t'))^T] \\ &= \mathbf{R}_{xx}(t,t')-\overline{\mathbf{x}}(t)\overline{\mathbf{x}}^T(t')\end{aligned} \tag{46}$$

Clearly the autocovariance at $t=t'$ is simply the covariance matrix; $\mathbf{P}_{xx}(t,t)=\mathbf{P}_{xx}(t)$.

The idea of autocorrelation and autocovariance can be extended to describe the relation between two different stochastic processes. The *cross-covariance kernel* between two processes $\mathbf{x}(t)$ and $\mathbf{z}(t)$ is defined as

$$\mathbf{P}_{xz}(t,t') \triangleq \mathrm{E}[(\mathbf{x}(t)-\overline{\mathbf{x}}(t))(\mathbf{z}(t')-\overline{\mathbf{z}}(t'))^T] \ , \tag{47}$$

with the cross-covariance matrix $\mathbf{P}_{xz}(t)=\mathbf{P}_{xz}(t,t)$ and the *cross correlation kernel* as

$$\mathbf{R}_{xz}(t,t') \triangleq \mathrm{E}[\mathbf{x}(t)\mathbf{z}^T(t')] \ = \mathbf{P}_{xz}(t,t')+\overline{\mathbf{x}}(t)\overline{\mathbf{z}}^T(t'). \tag{48}$$

A process is said to *independent* (or 'white') if the joint distribution on any set of time samples $\mathbf{x}(t_i)$ is equal to the product of their marginal distributions:

$$f(\mathbf{x}(t_1),\mathbf{x}(t_2),\cdots,\mathbf{x}(t_N)) = \prod_{i=1}^{N} f(\mathbf{x}(t_i)) \tag{49}$$

A process is said to be *uncorrelated* in time if

$$\mathbf{R}_{xx}(t,t') \triangleq \mathrm{E}[\mathbf{x}(t)\mathbf{x}^T(t')] \ = \mathrm{E}[\mathbf{x}(t)] \ \mathrm{E}[\mathbf{x}^T(t')] \ , \quad \forall t \neq t', \tag{50}$$

or equivalently $\mathbf{P}_{xx}(t,t')=0 \forall t \neq t'$. A process which is independent (white) is uncorrelated. However, a process that is uncorrelated is not generally independent except in the case where the process is Gaussian. This is because uncorrelatedness is concerned only with the properties of the second moments of a distribution whereas independence is concerned with all moments.

The idea of independence and uncorrelatedness can be extended to consider the relations between two stochastic processes. Two processes $\mathbf{x}(t)$ and $\mathbf{z}(t)$ are said to be independent if their joint distribution on any set of time samples $\{\mathbf{x}(t_i),\mathbf{z}(t_i)\}$ is equal to the product of their individual joint distributions on $\mathbf{x}(t_i)$ and $\mathbf{z}(t_i)$ respectively:

$$f(\mathbf{x}(t_1),\cdots,\mathbf{x}(t_N),\mathbf{z}(t_1),\cdots,\mathbf{z}(t_N)) = f(\mathbf{x}(t_1),\cdots,\mathbf{x}(t_N))f(\mathbf{z}(t_1),\cdots,\mathbf{z}(t_N)). \tag{51}$$

Two processes $\mathbf{x}(t)$ and $\mathbf{z}(t)$ are uncorrelated with each other if

$$\mathbf{R}_{xz}(t,t') \triangleq \mathrm{E}[\mathbf{x}(t)\mathbf{z}^T(t')] \ = \mathrm{E}[\mathbf{x}(t)] \ \mathrm{E}[\mathbf{z}^T(t')] \ , \quad \forall t \neq t', \tag{52}$$

or equivalently $\mathbf{P}_{xz}(t,t')=0$. As before, two processes which are independent are also uncorrelated, but the converse is only true for jointly Gaussian processes.

2.4.2 Stationarity

A process is termed strict-sense *stationary* if all its statistics are invariant to a time shift. This requires that the joint distribution of any set of time samples $\mathbf{x}(t_i)$ be invariant to a common shift in time τ:

$$f(\mathbf{x}(t_1+\tau), \mathbf{x}(t_2+\tau), \cdots, \mathbf{x}(t_N+\tau)) = f(\mathbf{x}(t_1), \mathbf{x}(t_2), \cdots, \mathbf{x}(t_N))$$

A process is wide-sense stationary if the first and second moments of the distribution are invariant to a time shift. Practically, this means that

1. $\mathrm{E}[\mathbf{x}(t)]$ is constant.
2. $\mathrm{E}[\mathbf{x}(t)\mathbf{x}^T(t+\tau)]$ depends only on the time difference τ, and so $\mathbf{R}_{xx}(t, t+\tau) = \mathbf{R}_{xx}(\tau)$ and $\mathbf{P}_{xx}(t, t+\tau) = \mathbf{P}_{xx}(\tau)$.
3. $\mathbf{R}_{xx}(0) = \mathrm{E}[\mathbf{x}(t)\mathbf{x}^T(t)]$ is finite and constant as also is $\mathbf{P}_{xx}(0)$

Although a process which is strict-sense stationary is also wide-sense stationary, the converse is not in general true. One exception to this is again a Gaussian process which is characterized completely by its mean and autocorrelation. Many useful stochastic processes are wide sense stationary, and many real phenomenon can be modeled using processes which have this property.

We will often have cause to estimate the mean and autocovariance of a process or signal from experimental information. It is usual to assume that the signal is indeed wide-sense stationary, or more generally can be considered as a system driven by noise which is wide-sense stationary. To compute the mean and autocovariance, we would normally obtain N samples of the process $\mathbf{x}_i(t), i = 1, \cdots, N,\ t = [0, T]$. each lasting for a time interval T which is much longer than the maximum time constant of interest. The expected value of $\mathbf{x}(t)$ may then be estimated from the ensemble average,

$$\overline{\mathbf{x}}(t) = \mathrm{E}[\mathbf{x}(t)] \approx \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i(t) \tag{53}$$

and the autocovariance from the sample autocovariance,

$$\begin{aligned} \mathbf{P}_{xx}(t_1, t_2) &= \mathrm{E}[(\mathbf{x}(t_1) - \overline{\mathbf{x}}(t_1))(\mathbf{x}(t_2) - \overline{\mathbf{x}}(t_2))^T] \\ &\approx \tfrac{1}{N}\textstyle\sum_{i=1}^{N}(\mathbf{x}_i(t_1) - \overline{\mathbf{x}}(t_1))(\mathbf{x}_i(t_2) - \overline{\mathbf{x}}(t_2))^T. \end{aligned} \tag{54}$$

If the process is assumed stationary, then the sample covariance should be averaged over all $t_1 - t_2 = \tau$ to yield $\mathbf{P}_{xx}(\tau)$.

2.4.3 Ergodicity

An important subclass of stationary processes is the set of ergodic processes. A process is termed *ergodic* if any statistic calculated by averaging all members of the ensemble of samples at a fixed time can be calculated equivalently by time-averaging over any single representative member of the ensemble. Not all stationary processes are ergodic although clearly every ergodic process is stationary. In general, there is no easy method of deciding if a process is ergodic. In practice, a series of time averages may be calculated on different realizations under the assumption of ergodicity and variations in these averages used to judge the validity of the assumption. If a process is known to be ergodic, then an estimate for the expected value of $\mathbf{x}(t)$ can be obtained from the average of N time samples of a single realization

$$\overline{\mathbf{x}} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}(t_i), \tag{55}$$

and the autocovariance from

$$\mathbf{P}_{xx}(\tau) = \frac{1}{N-\tau}\sum_{i=1}^{N-\tau}(\mathbf{x}(t_i)-\overline{\mathbf{x}})(\mathbf{x}(t_i+\tau)-\overline{\mathbf{x}})^T \tag{56}$$

In real situations it is important that long time sequences are used to estimate the mean and autocovariance to avoid any effects due to long characteristic time constants in the data. It is usually also advisable to use a flat-tailed window on the data to avoid propagation of edge effects caused by using a finite data set.

An uncorrelated stochastic process is simply a wide-sense stationary process in which the autocovariance is zero everywhere except at $\tau = 0$; $\mathbf{P}_{xx}(\tau) = \mathbf{P}_{xx}\delta(\tau)$. A slightly more interesting family of processes are those which are exponentially correlated with $\mathbf{P}_{xx}(\tau) = \mathbf{P}_{xx}e^{\tau/T}$. Two such processes are shown in Figure 1. In the first process, the time constant T is relatively short and so the process itself exhibits rapid oscillations while the autocovariance is highly peaked. In the second process the time constant T is relatively long and so the process oscillates only slowly and the autocovariance is spread. Physically, this is what one might expect, the autocovariance (or equivalent autocorrelation) describes the relation between different values of the process separated by the interval τ. When the autocovariance is highly peaked, values of the process separated by large time intervals have little relation to each other. As the autocorrelation spreads so the relation between samples of the process taken at a fixed time interval apart becomes closer. A second important point to note is that in both these example processes, the value of the covariance $\mathbf{P}_{xx} = \mathbf{P}_{xx}(0)$ is the same, and consequently tells us nothing about the time behaviour of the signal.

2.4.4 Power Spectral Density

For wide-sense stationary processes it is useful to also obtain a frequency-domain characterization of a process in the form of a *power spectral density*. The power spectral density

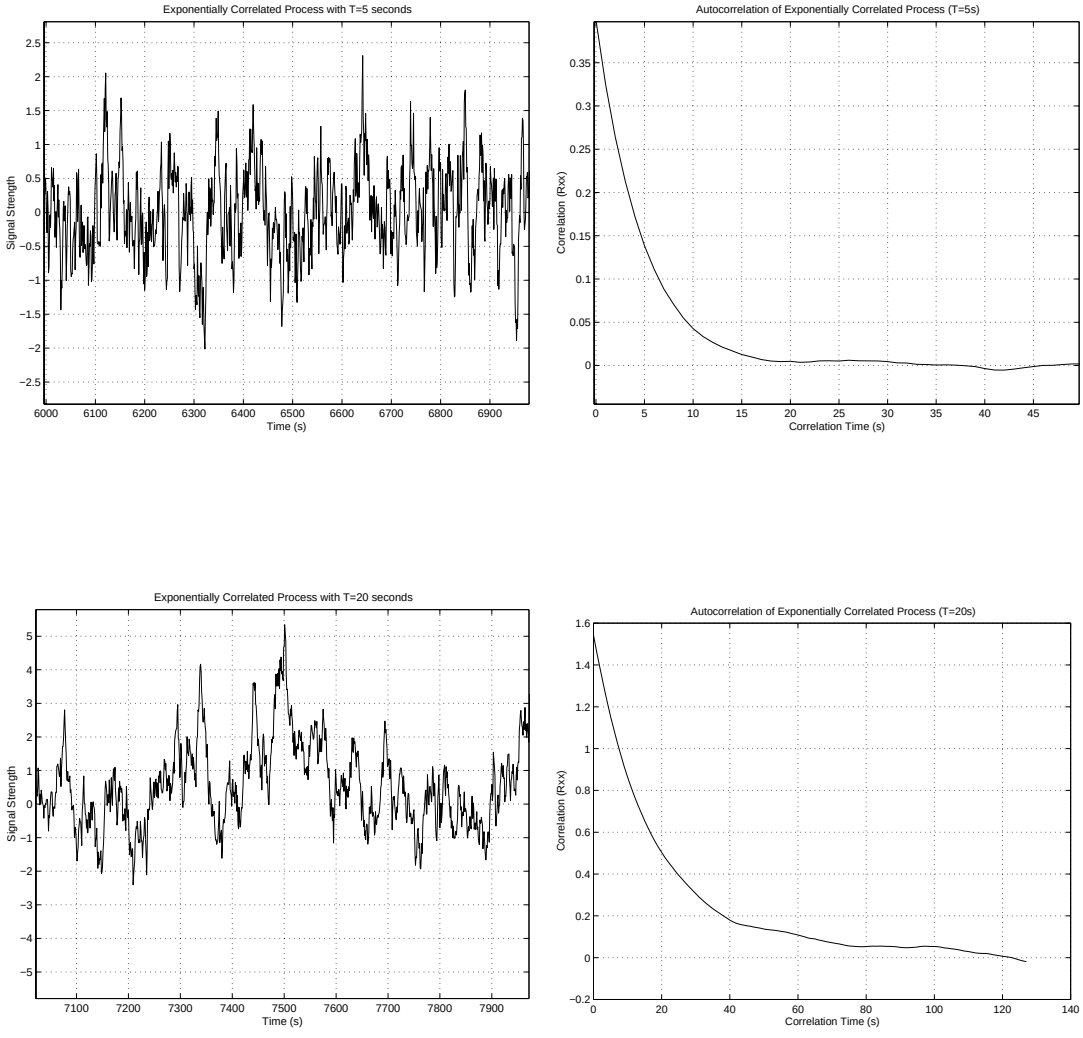


Figure 1: Two exponentially correlated scalar processes and their corresponding autocovariance. Note that the two processes have the same strength but different time constants

$\mathbf{S}_{xx}(\omega)$ of a wide-sense stationary process $\mathbf{x}(t)$ is defined as the Fourier transform of the correlation function $\mathbf{R}_{xx}(\tau)$:

$$\mathbf{S}_{xx}(\omega) = \int_{-\infty}^{\infty} \mathbf{R}_{xx}(\tau) e^{-j\omega\tau} \mathrm{d}\tau \tag{57}$$

The term 'power spectral density' is motivated by interpreting 'power' in the sense of mean squared value $\mathrm{E}[\mathbf{x}(t)\mathbf{x}^T(t)]$. Integration of $\mathbf{S}_{xx}(\omega)$ over an interval from ω_1 to ω_2 yields the mean squared value of the process consisting only of those harmonic components that lie between ω_1 and ω_2. Indeed, if we integrate over all frequencies we simply obtain the complete mean squared value (the inverse Fourier transform evaluated at $\tau = 0$)

$$\mathrm{E}[\mathbf{x}(t)\mathbf{x}^T(t)] \ = \mathbf{R}_{xx}(0) = \int_{-\infty}^{\infty} \mathbf{S}_{xx}(\omega) \mathrm{d}\omega \tag{58}$$

It is also possible to define the *cross-power spectral density* $\mathbf{S}_{xz}(\omega)$ of two wide-sense stationary processes $\mathbf{x}(t)$ and $\mathbf{z}(t)$in the obvious way

$$\mathbf{S}_{xz}(\omega) = \int_{-\infty}^{\infty} \mathbf{R}_{xz}(\tau) e^{-j\omega\tau} \mathrm{d}\tau \tag{59}$$

2.4.5 Systems Driven By White Noise

There is a fundamental theorem which we shall use extensively in developing estimation algorithms in various applications: **Any stationary noise process can be generated by an appropriate linear system being driven by white noise**. Practically, this means that if we have a noise model for a specific sensor or process, defined in terms of an autocorrelation function and power spectral density, we can *always* replace this by an equivalent linear system whose input is white noise and whose output is the required noise process. Most simply this is done in the following way (we will discuss this in more depth at a later stage in the course):

1. We take a number of samples from the sensor of interest and construct a power spectral density $S_{zz}(\omega)$ of the output signal $\mathbf{z}(k)$.

2. We note that if $S_{zz}(\omega)$ is actually the output from a system described by an impulse response function $h(t)$ then

   $$S_{zz}(\omega) = H(\omega)H^*(-\omega)S_{xx}(\omega)$$

   where $H(\omega)$ is the Fourier transform of $h(t)$ and $S_{xx}(\omega)$ is the power spectral density of the input.

3. If we assume that the input to the system is simply white noise with autocorrelation $Q\delta(\tau)$, then the input power spectral density will simply be a Q (a constant for all frequencies).

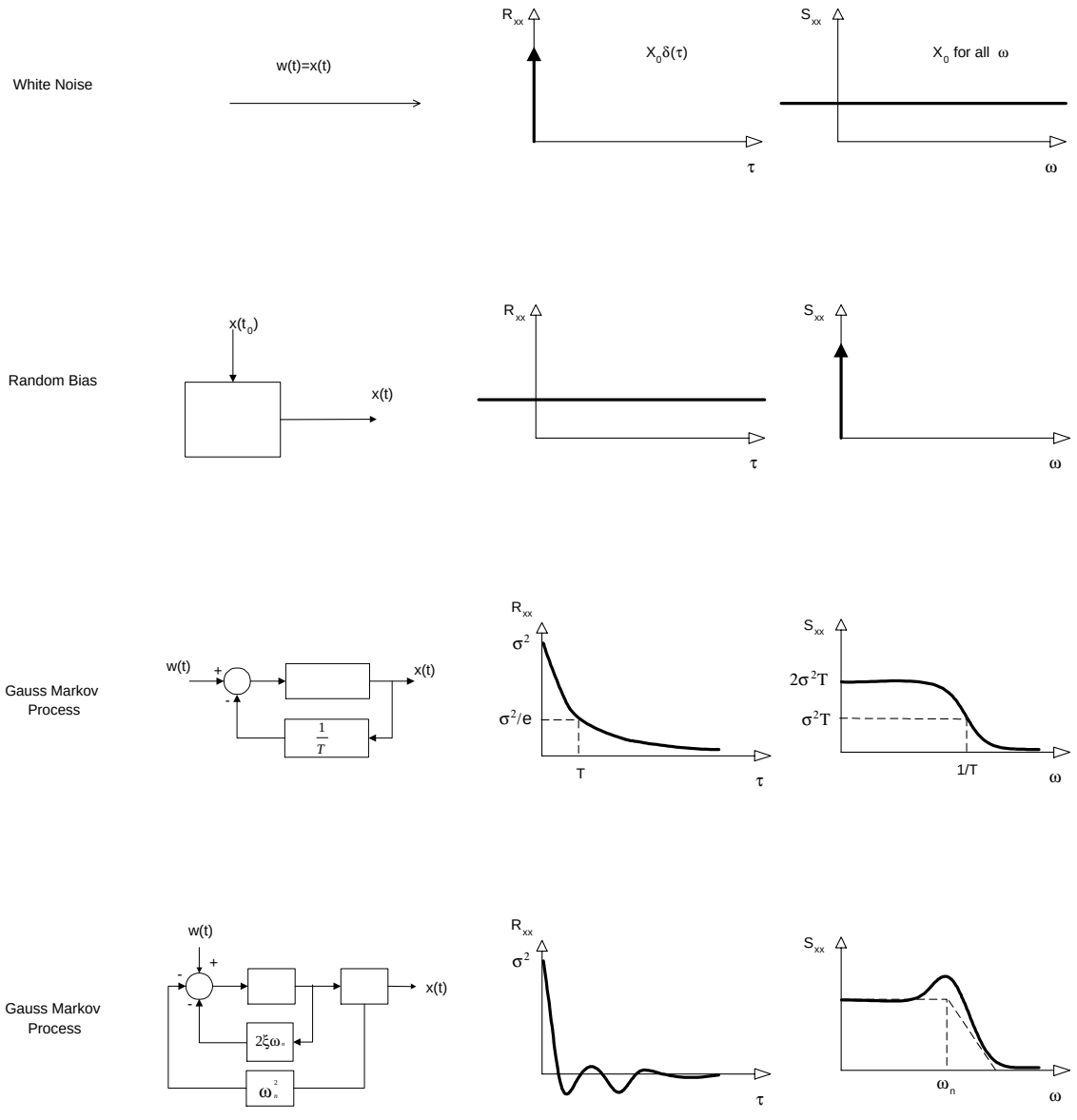


Figure 2: Four common noise models, their block diagrams, autocorrelations and power spectral densities

4. It follows then that $S_{yy}(\omega)$ can be modeled as the output of a system with impulse response function $h(t)$ (i.e. linear, time-invariant, etc), driven by white noise.

5. $H(\omega)$ can be found by curve-fitting to $S_{yy}(\omega)$, solving the equation $H(\omega)H^*(-\omega) = S_{yy}(\omega)/Q$, and taking the Inverse Fourier transform to find $h(t)$.

Figure 2 shows four example autocorrelations and their corresponding power spectral densities. Figure 2(a) describes an uncorrelated process with autocorrelation equal to the variance at $\tau = 0$ and zero elsewhere; $\mathbf{R}_{xx}(\tau) = \mathbf{P}_{xx}\delta\tau$. The power spectral density of this process is flat; $\mathbf{S}_{xx}(\omega) = \mathbf{P}_{xx}$, having constant strength at all frequencies (hence the term 'white'). Figure 2(b) describes a process which is exponentially correlated

$$\mathbf{R}_{xx}(\tau) = \mathbf{P}_{xx}e^{-|\tau|/T}.$$

An exponentially correlated process is sometimes called a *first order Gauss-Markov* process as it can be considered as the response of a first-order differential equation

$$\dot{x}(t) + \frac{1}{T}x(t) = w(t),$$

to uncorrelated white noise input $w(t)$. The power spectral density of such a process starts flat at low frequencies (corresponding to long time constants), then tails off at frequencies comparable to the correlation time constant as $\mathbf{S}_{xx}(\omega) = \frac{2\mathbf{P}_{xx}/T}{\omega^2+(1/T)^2}$ This type of frequency response is common of a wide variety of band-limited processes and is consequently a good model of many practical systems. Figure 2(c) describes a process which is completely correlated for all time, with autocorrelation equal to a constant; $\mathbf{R}_{xx}(\tau) = \mathbf{P}_{xx}$. The power spectral density of this process has all power concentrated at zero frequency; $\mathbf{S}_{xx}(\omega) = 2\pi\mathbf{P}_{xx}\delta(\omega)$. This type of model describes a process that consists entirely of a random bias (initial condition) but is otherwise constant. Figure 2(d) describes a process which is periodically correlated

$$\mathbf{R}_{xx}(\tau) = \frac{\mathbf{P}_{xx}}{\cos\nu}e^{-\zeta\omega_n|\tau|}\cos(\sqrt{1-\zeta^2}\omega_n|\tau| - \nu).$$

Such a process is usually called a *second order Gauss-Markov* process as it can be considered as the response x of a second-order differential equation of the form $\ddot{x}+2\zeta\omega_n\dot{x}+\omega_n^2 x = \omega_n^2 u$ to an input u of uncorrelated noise. In the example shown, $\zeta < 1$ and the autocorrelation is periodically negative indicating that if you know the value of $\mathbf{x}(t)$ at a time t, then you would expect $\mathbf{x}(t+\tau)$ to be of opposite sign for appropriate time intervals τ. The power spectral density of this process is given by

$$\mathbf{S}_{xx}(\omega) = \frac{2\mathbf{P}_{xx}}{\omega_n\cos\nu}\frac{\frac{\omega}{\omega_n}^2\sin(\alpha-\nu)+\sin(\alpha+\nu)}{\frac{\omega}{\omega_n}^4+2\frac{\omega}{\omega_n}^2(2\zeta-1)+1}$$

where $\alpha = \tan^{-1}(\zeta/\sqrt{1-\zeta^2})$ is known as the damping angle. This power spectral density is characteristic of a second-order resonant system, with approximately flat frequency response out to the natural frequency ω_n where there is a resonant frequency peak followed by a steep roll-off to zero. Second order process models with $\zeta > 1$ have similar autocorrelations and power spectral densities to first-order Gauss-Markov processes except, as would be expected from having two exponentials with real time-constants, the frequency roll-off is much faster. Such second-order processes are characteristic of "noise" due to wave motion of fuel slosh, for example.

2.5 Process Models

The process models of state we will considered may, in general, be described in state-space notation by a first order non-linear vector differential equation or state model of the form

$$\dot{\mathbf{x}}(t) = \mathbf{f}[\mathbf{x}(t), \mathbf{u}(t), t] + \mathbf{v}(t), \tag{60}$$

where $\mathbf{x}(t) \in \Re^n$ is the state of interest at time t, $\mathbf{u}(t) \in \Re^r$ a known control input, $\mathbf{f}[\cdot,\cdot,\cdot]$ a mapping (from $\Re^n \times \Re^r \times \Re^1$ to $\Re^n$) of state and control input to state 'velocity' at time t, and $\mathbf{v}(t)$ a random vector describing both dynamic driving noise and uncertainties in the state model itself. Implicit in this model is the understanding that the state $\mathbf{x}(t)$ at time t summarizes all past state information $\mathbf{x}(\tau), 0 \leq \tau < t$ and that together with subsequent control inputs $\mathbf{u}(\tau')$ is sufficient to describe all future state trajectories $\mathbf{x}(\tau'), \tau' > t$. For historical reasons the function $\mathbf{f}[\cdot,\cdot,t]$ will be termed the process model. The random vector $\mathbf{v}(t)$ will be termed process noise. In Equation 60 it is shown as being additive to the process model. In truth the assumption of additivity is unlikely to be valid as the process noise incorporates modeling uncertainty as well as driving noise.

Of particular interest in the estimation problems we will consider is the case in which the state model described by Equation 60 is linear in both state and control input in the form

$$\dot{\mathbf{x}}(t) = \mathbf{F}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t) + \mathbf{G}(t)\mathbf{v}(t), \tag{61}$$

where $\mathbf{F}(t)$ is an $n \times n$ time-varying matrix relating the state $\mathbf{x}(t)$ to the state velocity, $\mathbf{B}(t)$ is an $n \times r$ time-varying matrix relating control input to state velocity, and $\mathbf{G}(t)$ an $n \times q$ time-varying matrix relating the process noise vector to state velocity. Differential equations described by Equation 61 are well known to admit closed-form solutions as

$$\mathbf{x}(t) = \mathbf{\Phi}(t, t_0)\mathbf{x}(t_0) + \int_{t_0}^{t} \mathbf{\Phi}(t,\tau)\mathbf{B}(\tau)\mathbf{u}(\tau)\mathrm{d}\tau + \int_{t_0}^{t} \mathbf{\Phi}(t,\tau)\mathbf{G}(\tau)\mathbf{v}(\tau)\mathrm{d}\tau \tag{62}$$

where $\mathbf{\Phi}(\cdot,\cdot)$ is the state transition matrix satisfying the matrix differential equation

$$\dot{\mathbf{\Phi}}(t, t_0) = \mathbf{F}(t)\mathbf{\Phi}(t, t_0), \qquad \mathbf{\Phi}(t_0, t_0) = \mathbf{1}. \tag{63}$$

The state transition matrix has three important properties that should be noted:

1. It is uniquely defined for all t, t_0 in $[0, \infty]$.
2. (The semi-group property) $\mathbf{\Phi}(t_3, t_1) = \mathbf{\Phi}(t_3, t_2)\mathbf{\Phi}(t_2, t_1)$.
3. $\mathbf{\Phi}(t, t_0)$ is non singular and $\mathbf{\Phi}^{-1}(t, t_0)$=$\mathbf{\Phi}(t_0, t)$.

Specifically when $\mathbf{F}(t) = \mathbf{F}$ is a constant matrix, the state transition matrix is given by

$$\mathbf{\Phi}(t, t_0) = \mathbf{\Phi}(t - t_0) = \exp \mathbf{F}(t - t_0), \tag{64}$$

which clearly satisfies these three properties.

Although most real systems are not themselves linear, perturbation methods can be used to linearize true non-linear system models around some nominal state trajectory to yield a perturbation model which is itself linear. Such perturbation models are of great value in finding approximate solutions to otherwise intractable estimation problems. This also makes the study of linear-systems models of far more practical significance than simply their application to linear systems. Given a non-linear system model in the form

$$\dot{\mathbf{x}}(t) = \mathbf{f}[\mathbf{x}(t), \mathbf{u}(t), t] + \mathbf{G}(t)\mathbf{v}(t), \tag{65}$$

and a nominal trajectory described using the same process model

$$\dot{\mathbf{x}}_n(t) = \mathbf{f}[\mathbf{x}_n(t), \mathbf{u}_n(t), t], \tag{66}$$

we can expand Equation 65 about this nominal trajectory as a Taylor series;

$$\begin{aligned} \dot{\mathbf{x}}(t) \;\; = \;\; & \mathbf{f}[\mathbf{x}_n(t), \mathbf{u}_n(t), t] + \mathbf{G}(t)\mathbf{v}(t) \\ & + \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\substack{\mathbf{x}(t)=\mathbf{x}_n(t) \\ \mathbf{u}(t)=\mathbf{u}_n(t)}} (\mathbf{x}(t) - \mathbf{x}_n(t)) + O\left[(\mathbf{x}(t) - \mathbf{x}_n(t))^2\right] \\ & + \left.\frac{\partial \mathbf{f}}{\partial \mathbf{u}}\right|_{\substack{\mathbf{x}(t)=\mathbf{x}_n(t) \\ \mathbf{u}(t)=\mathbf{u}_n(t)}} (\mathbf{x}(t) - \mathbf{x}_n(t)) + O\left[(\mathbf{u}(t) - \mathbf{u}_n(t))^2\right]. \end{aligned} \tag{67}$$

Retaining only first order terms in this expansion and defining

$$\mathbf{F}(t) \stackrel{\triangle}{=} \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\substack{\mathbf{x}(t)=\mathbf{x}_n(t) \\ \mathbf{u}(t)=\mathbf{u}_n(t)}}, \qquad \mathbf{B}(t) \stackrel{\triangle}{=} \left.\frac{\partial \mathbf{f}}{\partial \mathbf{u}}\right|_{\substack{\mathbf{x}(t)=\mathbf{x}_n(t) \\ \mathbf{u}(t)=\mathbf{u}_n(t)}} \tag{68}$$

and

$$\delta\mathbf{x}(t) \stackrel{\triangle}{=} (\mathbf{x}(t) - \mathbf{x}_n(t)), \qquad \delta\mathbf{u}(t) \stackrel{\triangle}{=} (\mathbf{u}(t) - \mathbf{u}_n(t)) \tag{69}$$

we can subtract Equation 66 from Equation 67 to obtain a linear propagation model in the form

$$\delta\dot{\mathbf{x}}(t) = \mathbf{F}(t)\delta\mathbf{x}(t) + \mathbf{B}(t)\delta\mathbf{u}(t) + \mathbf{G}(t)\mathbf{v}(t). \tag{70}$$

This equation is now in the same form as Equation 61 and may be solved for the perturbed state vector $\delta\mathbf{x}(t)$ in closed form through Equation 62. Clearly this approximation is only valid when terms of second order and higher are small enough to be neglected; when the true and nominal trajectories are suitably close. With judicious design of the estimation algorithm this can be achieved surprisingly often. It is also possible to retain or approximate higher-order terms from Equation 67 and so improve the validity of the approximation.

In this course we will almost always be concerned with digital computer implementations of estimators and it is therefore appropriate that the models of state we consider should be described in a discrete time form. The discrete equivalent of Equation 60 is given by

$$\mathbf{x}(t_k) = \mathbf{f}[\mathbf{x}(t_{k-1}), \mathbf{u}(t_k), t_k] + \mathbf{G}(t_k)\mathbf{v}(t_k), \tag{71}$$

where the function $\mathbf{f}[\cdot,\cdot,\cdot]$ now maps the state $\mathbf{x}(t_{k-1})$ at time t_{k-1} and the control input $\mathbf{u}(t_k)$ at time t_k to the state $\mathbf{x}(t_k)$ at the next time step t_k Even when employing a discrete-time model, it should not be forgotten that the underlying process is often continuous. To build effective discrete-time models of state transition and driving noise we should always begin by describing the system in continuous form, to understand true state transition and identify true sources of uncertainty, and from this derive an equivalent discrete-time model. It is also sometimes useful to reconsider a discrete-time model in a continuous-time form to obtain steady-state information about estimator performance. For these reasons it is important to understand how an equivalent discrete-time model is obtained from an underlying continuous-time process.

In general the derivation of a non-linear discrete-time model in the form of Equation 71 from a non-linear continuous model of the form of Equation 60 can only be obtained through numerical integration and appropriate approximation on a case-by-case basis. However, if the continuous-time system model is linear in the form of Equation 61 then an equivalent discrete-time model can readily be obtained by evaluating Equation 62 over the interval (t_k, t_{k-1}) as

$$\mathbf{x}(t_k) = \mathbf{\Phi}(t_k, t_{k-1})\mathbf{x}(t_{k-1}) + \int_{t_{k-1}}^{t_k} \mathbf{\Phi}(t_k, \tau)\mathbf{B}(\tau)\mathbf{u}(\tau)\mathrm{d}\tau + \int_{t_{k-1}}^{t_k} \mathbf{\Phi}(t_k, \tau)\mathbf{G}(\tau)\mathbf{v}(\tau)\mathrm{d}\tau. \tag{72}$$

By assuming that $\mathbf{u}(t)$ and $\mathbf{v}(t)$ remain approximately constant over the interval (t_{k-1}, t_k) we can identify

$$\begin{aligned} \mathbf{F}(t_k) &\triangleq \mathbf{\Phi}(t_k, t_{k-1}) \\ \mathbf{B}(t_k) &\triangleq \textstyle\int_{t_{k-1}}^{t_k} \mathbf{\Phi}(t_k, \tau)\mathbf{B}(\tau)\mathrm{d}\tau \\ \mathbf{G}(t_k) &\triangleq \textstyle\int_{t_{k-1}}^{t_k} \mathbf{\Phi}(t_k, \tau)\mathbf{G}(\tau)\mathrm{d}\tau \end{aligned} \tag{73}$$

and obtain the equivalent discrete-time model in the form

$$\mathbf{x}(t_k) = \mathbf{F}(t_k)\mathbf{x}(t_{k-1}) + \mathbf{B}(t_k)\mathbf{u}(t_k) + \mathbf{G}(t_k)\mathbf{v}(t_k). \tag{74}$$

The accuracy of this model could be improved by taking mean values for both $\mathbf{u}(t)$ and $\mathbf{v}(t)$ over the sampling interval.

In almost all cases the time interval $\Delta t(k) \stackrel{\triangle}{=} t_k - t_{k-1}$ between successive samples of the state remains constant. In this case it is common particular to drop the time argument and simply index variables by the sample number. In this case Equation 71 is written as

$$\mathbf{x}(k) = \mathbf{f}[\mathbf{x}(k-1), \mathbf{u}(k), k] + \mathbf{G}(k)\mathbf{v}(k), \tag{75}$$

and Equation 74 as

$$\mathbf{x}(k) = \mathbf{F}(k)\mathbf{x}(k-1) + \mathbf{B}(k)\mathbf{u}(k) + \mathbf{G}(k)\mathbf{v}(k). \tag{76}$$

2.6 Sensor Models

The models of state observation considered in this course may, in general, be described in state-space notation by a non-linear vector function in the form

$$\mathbf{z}(t) = \mathbf{h}[\mathbf{x}(t), \mathbf{u}(t), t] + \mathbf{w}(t), \tag{77}$$

where $\mathbf{z}(t) \in \Re^m$ is the observation made at time t, $\mathbf{h}[\cdot, \cdot, \cdot]$ is a mapping (from $\Re^n \times \Re^r \times \Re$ to $\Re^m$) of state and control input to observations, and $\mathbf{w}(t)$ a random vector describing both measurement corruption noise and uncertainties in the measurement model itself. The function $\mathbf{h}[\cdot, \cdot, \cdot]$ is termed the observation model. It describes the observations $\mathbf{z}(t)$ that are made at a time t when the true state is given by $\mathbf{x}(t)$. The random vector $\mathbf{w}(t)$ will be termed the observation noise. As with process noise, the observation noise is assumed additive to the observation model in Equation 77, a situation unlikely to be met in particular but essential in making the estimation problem tractable.

We will often be interested in linear observation models of the form

$$\mathbf{z}(t) = \mathbf{H}(t)\mathbf{x}(t) + \mathbf{D}(t)\mathbf{w}(t), \tag{78}$$

where $\mathbf{H}(t)$ is an $m \times n$ time-varying matrix relating the state to the observation at time t and $\mathbf{D}(t)$ is a $m \times p$ time-varying matrix relating measurement and model noise to the observation made. Conventionally the dependence and control input is not explicit in the observation equation and is henceforth omitted unless required. Again, it is very often possible to use perturbation techniques to approximate non-linear observation models with an approximate model linear in the perturbations. Given a non-linear observation model in the form

$$\mathbf{z}(t) = \mathbf{h}[\mathbf{x}(t)] + \mathbf{D}(t)\mathbf{w}(t), \tag{79}$$

and a nominal observation described using the same observation model

$$\mathbf{z}_n(t) = \mathbf{h}[\mathbf{x}_n(t)], \tag{80}$$

we can expand Equation 79 about this nominal trajectory as a Taylor series;

$$\begin{aligned}\mathbf{z}(t) &= \mathbf{h}[\mathbf{x}(n)t] + \mathbf{D}(t)\mathbf{w}(t) \\ &\quad + \left.\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right|_{\mathbf{x}(t)=\mathbf{x}_n(t)} (\mathbf{x}(t) - \mathbf{x}_n(t)) + O\left[(\mathbf{x}(t) - \mathbf{x}_n(t))^2\right].\end{aligned} \tag{81}$$

Retaining only first order terms in this expansion and defining

$$\mathbf{H}(t) \triangleq \left.\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right|_{\mathbf{x}(t)=\mathbf{x}_n(t)} \tag{82}$$

and

$$\delta\mathbf{x}(t) \triangleq (\mathbf{x}(t) - \mathbf{x}_n(t)), \qquad \delta\mathbf{z}(t) \triangleq (\mathbf{z}(t) - \mathbf{z}_n(t)) \tag{83}$$

we can subtract Equation 80 from Equation 81 to obtain a linear propagation model for the observation in the form

$$\delta\mathbf{z}(t) = \mathbf{H}(t)\delta\mathbf{x}(t) + \mathbf{D}(t)\mathbf{w}(t) \tag{84}$$

This equation is now in the same form as Equation 78. Again the approximation is only valid when terms of second order and higher are small enough to be neglected; when true state and nominal state are suitably close. It is somewhat more difficult to ensure proximity of nominal and true model in the case of observation models than in the case of state models. The reason for this is that the observation vector is usually of lower dimension than the state vector and so large errors in components of a state vector may not show up in the corresponding observation. This means that although observation perturbations may be small, state perturbations may not be small so voiding the assumption that second-order and higher terms can be neglected in the expansion in Equation 81. A well known instance of this is the use of bearing-only sensor information (typically associated with passive sensors) to estimate point locations.

In the systems we will be concerned with, observations will normally be taken at discrete intervals in time. In this case the observation model becomes

$$\mathbf{z}(t_k) = \mathbf{h}[\mathbf{x}(t_k), \mathbf{u}(t_k), t_k] + \mathbf{w}(t_k), \qquad k = 1, 2, \cdots. \tag{85}$$

For synchronous observations we will normally drop the explicit time reference and write Equation 85 as

$$\mathbf{z}(k) = \mathbf{h}[\mathbf{x}(k), \mathbf{u}(k), k] + \mathbf{D}(k)\mathbf{w}(k), \tag{86}$$

the corresponding linear observation model of Equation 78 becomes

$$\mathbf{z}(k) = \mathbf{H}(k)\mathbf{x}(k) + \mathbf{D}(k)\mathbf{w}(k) \tag{87}$$

3 Introduction To Estimation

The subject of estimation has a long and illustrious history. For many centuries methods of estimation existed to solve problems in navigation, surveying and astronomy: many of which are still used. Probably the first formal estimation technique was the method of least-squares, developed by Gauss to estimate the trajectories of planets about the Sun. The use of probability as a formal method in estimation was a surprisingly long time in coming. Although probabilistic modeling techniques had been recognized since the late seventeenth century, it was not until Fisher's work on likelihoods in the first quarter of this century that probabilistic techniques were formally applied to the estimation problem. During and immediately after the second world war a considerable amount of work was done on the estimation problem directed primarily at military applications such as fire control and aircraft navigation. Most celebrated of the methods developed during this period is the Wiener filter. The early 1960's saw the development of the Kalman Filter as an efficient and versatile estimator. Kalman's original method provided a new means of computing least-squares estimates using state-space methods and employing a recursive algorithm for incorporating new observations The Kalman filter has found widespread use in many different application areas and has been employed in solving a huge range of estimation problems. The Kalman filter is arguably the single most important contribution made in estimation theory over the last forty years. Indeed, most of the techniques described in this course are based on the Kalman filter algorithm. The last fifteen years has seen an increasing use of decision-theoretic methods as a basis for the development of new estimation techniques. Much of this work is concerned with the use of estimator performance metrics other than quadratic loss to improve both overall estimation robustness and performance in the light of poorly known system and observation behaviours. An important example of such methods is the minimax or game-theoretic approach to estimation. Modern estimation theory encompasses a broad range of techniques applicable to a range of estimation problems.

3.1 Models

In the remainder of this course we will be concerned with estimation of continuous-valued quantities or states $\mathbf{x} \in \Re^n$. The states $\mathbf{x}$ incorporate all the parameters of interest in the estimation problem at hand: the location of objects, parametric descriptions of different features, the velocity and acceleration of points. Observations made of the different components of this state are also assumed to be continuous-valued quantities $\mathbf{z} \in \Re^m$: the range and bearing to an object, the orientation of a feature, the velocity of a point. The continuous-valued multi-sensor estimation problem concerns the use of the observations $\mathbf{z}$ obtained from a number of different sensors to compute an estimate $\hat{\mathbf{x}}$ of the true state $\mathbf{x}$.

3.2 Estimation Methods

The decision rules, or estimators, we shall consider may be written in the general form

$$\hat{\mathbf{x}}(t_i \mid t_j) = \hat{\mathbf{x}}(\mathbf{z}(\tau_k), \mathbf{u}(\tau_k), t_i : 0 \leq \tau_k \leq t_j,), \tag{88}$$

where $\hat{\mathbf{x}}(t_i \mid t_j)$ is the computed estimate[2] of the state $\mathbf{x}$ at time t_i based on observations made up to time t_j, and $\hat{\mathbf{x}}(\cdot, \cdot, \cdot)$ is a function that maps the observation sequence and control sequence to an estimate. The function $\hat{\mathbf{x}}(\cdot, \cdot, \cdot)$ is called the estimation rule (or estimator). It clearly depends on both the process and observation models. It must also be defined with respect to a performance criterion which allows comparison of different possible rules and estimates. This criterion is most often defined in terms of a loss function $L[\mathbf{x}(t), \hat{\mathbf{x}}(t \mid t_j)]$ (or equivalent utility function) which penalizes differences between the true state and the estimate. The observation information available to the estimator will now be in the form of a finite set or sequence of observations taken at discrete intervals in time $\mathbf{Z}^j \stackrel{\triangle}{=} \{\mathbf{z}(t_k) : 1 \leq k \leq j\}$, rather than a time history or function. The estimate at a time t_i must now be computed from

$$\begin{aligned} \hat{\mathbf{x}}(t_i \mid t_j) &= \hat{\mathbf{x}}(\mathbf{z}(t_1), \cdots, \mathbf{z}(t_k), \mathbf{u}(t_i), t_i : t_k \leq t_j) \\ &= \hat{\mathbf{x}}(\mathbf{Z}^j, \mathbf{u}(t_i), t_i). \end{aligned} \tag{89}$$

We shall now describe four of the most common decision rules or estimators.

3.3 Maximum Likelihood and Maximum *a posteriori* Rules

The probability density function $f(\mathbf{Z}^k \mid \mathbf{x})$ (the conditional probability of the observation sequence $\mathbf{Z}^k$ given the parameter $\mathbf{x}$), has so far been interpreted as the probability of a specific observation sequence $\mathbf{Z}^k$ given that the true state of nature is $\mathbf{x}$. It is also possible to consider the distribution as a function of $\mathbf{x}$;

$$\Lambda(\mathbf{x}) \stackrel{\triangle}{=} f(\mathbf{Z}^k \mid \mathbf{x}). \tag{90}$$

The function $\Lambda(\mathbf{x})$ is strictly termed the **Likelihood function**[3]. The likelihood function, interpreted as function of $\mathbf{x}$, is a measure of how "likely" a parameter value is, given the observations that have been made.

A widely used decision rule for estimation of *nonrandom* (constant) parameters is the **Maximum Likelihood** (ML) method. The ML method simply aims to find the value of

[2]The term $\hat{\mathbf{x}}(t_i \mid t_j)$ should be read as the estimate of the state $\mathbf{x}(\cdot)$ at time t_i given all information up to time t_j.

[3]This should not be confused with our non-technical use of the word likelihood throughout this section to describe the relative probability of different events.

the parameter $\mathbf{x}$ which maximizes the likelihood function $\Lambda(\mathbf{x})$. The maximum likelihood estimate $\hat{\mathbf{x}}^{ML}$ is thus given by

$$\begin{aligned} \hat{\mathbf{x}}^{ML} &\triangleq \arg\max_{\mathbf{X}\in\mathcal{X}} \Lambda(\mathbf{x}) \\ &= \arg\max_{\mathbf{X}\in\mathcal{X}} f(\mathbf{Z}^k \mid \mathbf{x}) \end{aligned} \tag{91}$$

It is a post-experimental, or 'frequentist', estimation philosophy based on the likelihood principle. The likelihood principle simply states that all we know about the unknown state is what we obtain through experimentation and so the likelihood function contains all the information we need to construct an estimate for $\mathbf{x}$.

Example 3

Consider the measurement

$$z = x + w$$

of the unknown parameter $\mathbf{x}$ *in the presence of additive measurement noise* w *assumed to be normally distributed with zero mean and variance* σ^2*. First assume that* $\mathbf{x}$ *is an unknown constant, then the likelihood function of* $\mathbf{x}$ *is:*

$$f(z \mid x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(z-x)^2}{2\sigma^2}\right)$$

Thus

$$\hat{x}^{ML}(k) = \arg\max_x f(z \mid x) = z,$$

that is, the ML estimate is simply the observation itself.

In the case where the parameter $\mathbf{x}$ is considered to be random, a realization of $\mathbf{x}$ according to a probability density $f(\mathbf{x})$ is assumed to have occurred; i.e., a sample was "drawn" from a "population" with the assumed probability density function. The value then stays constant during the measurement process. In this case, when $f(\mathbf{x})$ is taken to be the prior probability describing what we believe about $\mathbf{x}$ before obtaining any information, we can use Bayes rule to find a new *posterior* probability density function

$$f(\mathbf{x} \mid \mathbf{Z}^k) = \frac{f(\mathbf{Z}^k \mid \mathbf{x}) f(\mathbf{x})}{f(\mathbf{Z}^k)}. \tag{92}$$

This equation summarizes both the information obtained through observation $f(\mathbf{Z}^k \mid \mathbf{x})$ and any prior knowledge $f(\mathbf{x})$ we may have about the state of interest in terms of a *posterior* probability density function $f(\mathbf{x} \mid \mathbf{Z}^k)$. The posterior density function should be interpreted as a measure of how "likely" a parameter value is, given both the observations

$\mathbf{Z}^k$ and the prior information $f(\mathbf{x})$. Thus for a random parameter, the ML-equivalent estimator is the **maximum a posteriori** (MAP) estimate given by:

$$\begin{aligned} \hat{\mathbf{x}}^{MAP} &= \arg\max_{\mathbf{X}\in\mathcal{X}} f(\mathbf{x} \mid \mathbf{Z}^k) \\ &= \arg\max_{\mathbf{X}\in\mathcal{X}} \left[f(\mathbf{Z}^k \mid \mathbf{x}) f(\mathbf{x}) \right] \end{aligned} \tag{93}$$

the last equality follows from the fact that the denominator on the right hand side of 92 is independent of $\mathbf{x}$ and therefore irrelevant for its maximization.

Example 4

If the likelihood function is Gaussian as in Example 3, and the prior information given about the parameter is that $\mathbf{x}$ *is a Gaussian random variable with mean* $\overline{x}$ *and variance* σ_0^2*, then the posterior distribution of* $\mathbf{x}$ *conditioned on the observation* z *is*

$$f(x \mid z) = \frac{f(z \mid x) f(x)}{f(z)} = c(z) \exp\left(-\frac{(z-x)^2}{2\sigma^2} - \frac{(\overline{x}-x)^2}{2\sigma_0^2} \right) \tag{94}$$

where c *is a normalizing constant, independent of* $\mathbf{x}$*. The MAP estimate is thus*

$$\hat{x}^{MAP}(k) = \arg\max_x f(x \mid z) = \overline{x} + \frac{\sigma_0^2}{\sigma_0^2 + \sigma^2}(z - \overline{x}) \tag{95}$$

Notice that as $\sigma_0^2 \to \infty$*, so* $\hat{x}^{MAP}(k) \to \hat{x}^{ML}(k)$*, that is as confidence in the prior information decreases, so the MAP estimate becomes equivalent to the ML estimate.*

Example 5

If the likelihood function is Gaussian as in Example 3, but the prior probability density function of x *is given by the one-sided exponential* $f(x) = ae^{-ax}$*,* $x \geq 0$*, then the posterior density is given by*

$$f(x \mid z) = c(z) \exp\left(-\frac{(z-x)^2}{2\sigma^2} - ax \right), \qquad x \geq 0,$$

so the MAP estimate is

$$\hat{x}^{MAP} = \max(z - \sigma^2 a, 0).$$

The maximum a posteriori estimator finds the estimate $\hat{\mathbf{x}}$ which maximizes the posterior distribution $f(\mathbf{x} \mid \mathbf{Z}^k)$. Like the ML estimator, the MAP estimator computes the *mode* of the (posterior) distribution as the estimate. Indeed, according to Equation 92, if the prior distribution $f(\mathbf{x})$ is non-informative (constant) on all $\mathbf{x} \in \mathcal{X}$, then for a given observation sequence $\mathbf{Z}^k$, the maximum of the posterior distribution $f(\mathbf{x} \mid \mathbf{Z}^k)$ will be coincident with the maximum of the likelihood function $f(\mathbf{Z}^k \mid \mathbf{x})$ and so the MAP and ML estimates will be the same.

3.4 Least Squares and Minimum Mean Squared Estimation

Perhaps the most widely known estimation procedure for non-random parameters is the **least-squares** (LS) method. The least squares method is based directly on the use of the sum of squared errors as a loss function but without the explicit use of a probabilistic model of information. However, with some (reasonable) assumptions about the probabilistic nature of the observations, a least-squares estimator can be shown to be Bayes rule.

We begin with a fixed, but unknown, state $\mathbf{x} \in \mathcal{X}$. We take a sequence of measurements $\mathbf{Z}^k = \{\mathbf{z}(j) : j = 1, \cdots, k\}$ known to be related to the unknown state through the deterministic function $\mathbf{h}$ in the form

$$\mathbf{z}(j) = \mathbf{h}(j, \mathbf{x}) + \epsilon(j), \qquad j = 1, \cdots, k, \tag{96}$$

where $\epsilon(j)$ is a sequence of 'random disturbances' (sometimes called the residuals). No explicit assumption is made about the sequence $\epsilon(j)$, although it is implicit that they should have zero mean and constant strength. The least squares estimate of $\hat{\mathbf{x}}^{LS}(k)$ of the unknown state $\mathbf{x}$ after taking k measurements is the value of $\mathbf{x} \in \mathcal{X}$ which minimizes the sum of the squared errors:

$$\hat{\mathbf{x}}^{LS}(k) = \arg\min_{x \in \mathcal{X}} \sum_{j=1}^{k} \left[\mathbf{z}(j) - \mathbf{h}(j, \mathbf{x})\right]^T \left[\mathbf{z}(j) - \mathbf{h}(j, \mathbf{x})\right] \tag{97}$$

Example 6

A standard problem is when the model is linear

$$\mathbf{z}(j) = \mathbf{H}(j)\mathbf{x} + \epsilon(j), \qquad j = 1, \cdots, k, \tag{98}$$

and the minimization criteria becomes

$$L = \sum_{j=1}^{k} \left[\mathbf{z}(j) - \mathbf{H}(j)\mathbf{x}\right]^T \left[\mathbf{z}(j) - \mathbf{H}(j)\mathbf{x}\right]. \tag{99}$$

Differentiating this with respect to $\mathbf{x}$*, we have*

$$\frac{\mathrm{d}L}{\mathrm{d}\mathbf{x}} = \sum_{j=1}^{k} \left[\mathbf{H}^T(j)\mathbf{H}(j)\mathbf{x} - \mathbf{H}^T(j)\mathbf{z}(j)\right], \tag{100}$$

setting this equal to zero and rearranging in terms of $\mathbf{x}$ *we obtain*

$$\hat{\mathbf{x}}^{LS}(k) = \left(\sum_{j=1}^{k} \mathbf{H}^T(j)\mathbf{H}(j)\right)^{-1} \sum_{j=1}^{k} \mathbf{H}^T(j)\mathbf{z}(j), \tag{101}$$

when the inverse exists.

A related problem is that of finding an estimate that minimizes a weighted sum of squares;

$$L = \sum_{j=1}^{k} \left[\mathbf{z}(j) - \mathbf{H}(j)\mathbf{x}\right]^T \mathbf{W}(j) \left[\mathbf{z}(j) - \mathbf{H}(j)\mathbf{x}\right]. \tag{102}$$

Again, differentiating this with respect to $\mathbf{x}$*, we have*

$$\frac{\mathrm{d}L}{\mathrm{d}\mathbf{x}} = \sum_{j=1}^{k} \left[\mathbf{H}^T(j)\mathbf{W}(j)\mathbf{H}(j)\mathbf{x} - \mathbf{H}^T(j)\mathbf{z}(j)\right], \tag{103}$$

setting this equal to zero and rearranging in terms of $\mathbf{x}$ *we obtain*

$$\hat{\mathbf{x}}^{LS}(k) = \left(\sum_{j=1}^{k} \mathbf{H}^T(j)\mathbf{W}(j)\mathbf{H}(j)\right)^{-1} \sum_{j=1}^{k} \mathbf{H}^T(j)\mathbf{W}(j)\mathbf{z}(j) \tag{104}$$

The preceding example makes it clear that the least-squares estimator computes an estimate which is in some sense a mean. For random parameters $\mathbf{x}$, the counterpart of the least-squares estimate is obtained by minimizing the mean squared error, leading directly to the **minimum mean-squared error** MMSE estimate:

$$\hat{\mathbf{x}}^{MMSE}(k) = \arg\min_{\hat{\mathbf{x}} \in \mathcal{X}} \mathrm{E}[(\hat{\mathbf{x}} - \mathbf{x})^T(\hat{\mathbf{x}} - \mathbf{x}) \mid \mathbf{Z}^k]. \tag{105}$$

In this case, we take a sequence of measurements $\mathbf{Z}^k = \{\mathbf{z}(j); j = 1, \cdots, k\}$ related to the unknown state through the stochastic function $\mathbf{h}$ in the form

$$\mathbf{z}(j) = \mathbf{h}(j, \mathbf{x}) + \mathbf{w}(j), \qquad j = 1, \cdots, k, \tag{106}$$

where $\mathbf{w}(j)$ is a sequence of random variables with known probability density function. The estimation criteria that we choose to minimize is the expected value of the error given the observations

$$\begin{aligned} L &= \mathrm{E}[(\hat{\mathbf{x}} - \mathbf{x})^T(\hat{\mathbf{x}} - \mathbf{x}) \mid \mathbf{Z}^k] \\ &= \int_{-\infty}^{\infty} (\hat{\mathbf{x}} - \mathbf{x})^T(\hat{\mathbf{x}} - \mathbf{x}) f(\mathbf{x} \mid \mathbf{Z}^k) \mathrm{d}\mathbf{x}. \end{aligned} \tag{107}$$

Differentiating the objective L with respect to the estimate $\hat{\mathbf{x}}$ and setting equal to zero, we have

$$\frac{\partial L}{\partial \hat{\mathbf{x}}} = 2 \int_{-\infty}^{\infty} (\hat{\mathbf{x}} - \mathbf{x}) f(\mathbf{x} \mid \mathbf{Z}^k) \mathrm{d}\mathbf{x} = \mathbf{0}. \tag{108}$$

Rearranging, and using the fact that $\hat{\mathbf{x}}$ is a constant, we obtain

$$\begin{aligned} \hat{\mathbf{x}}^{MMSE}(k) &= \int_{-\infty}^{\infty} \mathbf{x} f(\mathbf{x} \mid \mathbf{Z}^k) \mathrm{d}\mathbf{x} \\ &= \mathrm{E}[\mathbf{x} \mid \mathbf{Z}^k] \end{aligned} \tag{109}$$

That is, in all cases, the minimum mean squared-error estimate is simply the conditional mean of the state $\mathbf{x}$ given all observations $\mathbf{Z}^k$.

Another useful result which we will make use of in later sections is the fact that the estimate which minimizes a *weighted* mean squared error is still just the conditional expected mean. With the estimation criterion now given by

$$\begin{aligned} L &= \mathrm{E}[(\hat{\mathbf{x}}-\mathbf{x})^T\mathbf{S}(\hat{\mathbf{x}}-\mathbf{x}) \mid \mathbf{Z}^k] \\ &= \int_{-\infty}^{\infty}[(\hat{\mathbf{x}}-\mathbf{x})]^T\mathbf{S}(\hat{\mathbf{x}}-\mathbf{x})f(\mathbf{x} \mid \mathbf{Z}^k)\mathrm{d}\mathbf{x}. \end{aligned} \tag{110}$$

the estimate $\hat{\mathbf{x}}$ is obtained from

$$\frac{\partial L}{\partial \hat{\mathbf{x}}} = 2\mathbf{S}\int_{-\infty}^{\infty}(\hat{\mathbf{x}}-\mathbf{x})\, f(\mathbf{x} \mid \mathbf{Z}^k)\mathrm{d}\mathbf{x} = \mathbf{0}, \tag{111}$$

which clearly has a solution given by Equation 109. That is, the weighting matrix $\mathbf{S}$ does not affect the value of the estimate.

Example 7

A case of particular interest is when the parameter x is known to be restricted to certain values (a restricted parameter space). In this case we can model our prior information about x as a uniform (improper distribution on those parts of the parameter space where x is known to lie; $I_X(x) = 1$, when $x \in X'$, and $I_X(x) = 0$, when $x \notin X'$.

For example if our prior information is that x takes on only positive values $f(x) = I_{(0,\infty)}$, and the likelihood function is again Gaussian, then the posterior distribution for x is given by

$$f(x \mid z) = \frac{\exp\{-(x-z)^2/2\sigma^2\}I_{(0,\infty)}}{\int_0^{\infty}\exp\{-(x-z)^2/2\sigma^2\}}\mathrm{d}x.$$

The MMSE estimate for x is then given by

$$\begin{aligned} \mathrm{E}[x \mid z] &= \frac{\int_0^{\infty} x\exp\{-(x-z)^2/2\sigma^2\}I_{(0,\infty)}\mathrm{d}x}{\int_0^{\infty}\exp\{-(x-z)^2/2\sigma^2\}\mathrm{d}x} \\ &= z + \frac{(2\pi)^{-1/2}\sigma\exp(-z^2/2\sigma^2)}{1-\Phi(-x/\sigma)}, \end{aligned}$$

where $\Phi(\cdot)$ is the standard Gaussian cumulative distribution function.

3.5 The Relation Between different Estimators

We have seen that both the maximum likelihood estimator and the maximum *a posteriori* estimator compute the estimate corresponding to the *mode* or maximum of a distribution, whereas the minimum mean squared-error estimator computes an estimate corresponding to the *mean* of a distribution. As might be expected, these estimators compute the same estimate when the mean and mode coincide.

Example 8

ML and LS estimators: Assume that k measurements are made:

$$z(j) = x + w(j)$$

Assume further that the noises $w(j)$ are all independent identically distributed (IID), Gaussian, with zero-mean and variance σ^2. The likelihood function is thus given by

$$\begin{aligned} f(x \mid \mathbf{Z}^k) &= \Pi_{j=1}^k f(x \mid z(j)) \\ &= \Pi_{j=1}^k \tfrac{1}{(2\pi)^{1/2}\sigma} \exp\left(-\tfrac{1}{2\sigma^2}[z(j)-x]^2\right) \\ &= \tfrac{1}{(2\pi)^{k/2}\sigma^k} \exp\left(-\tfrac{1}{2\sigma^2}\textstyle\sum_{j=1}^k [z(j)-x]^2\right) \end{aligned} \tag{112}$$

To find an ML estimate for $\mathbf{x}$, we maximize this likelihood function;

$$\hat{\mathbf{x}}^{ML}(k) = \arg\max_{\mathbf{X}\in\mathcal{X}} f(x \mid \mathbf{Z}^k) \tag{113}$$

Maximization in this case is equivalent to minimizing the exponent in Equation 112

$$\hat{\mathbf{x}}^{ML}(k) = \arg\min_{\mathbf{X}\in\mathcal{X}} \sum_{j=1}^{k} [z(j)-x]^2, \tag{114}$$

which is, of course, an LS estimate for $\mathbf{x}$. Thus, for the case of observations in IID Gaussian noise, ML and LS estimates coincide and are both given by the sample mean:

$$\hat{x}^{ML}(k) = \hat{x}^{LS}(k) = \frac{1}{k}\sum_{i=1}^{k} z_i \tag{115}$$

Example 9

MAP and MMSE estimators: In Example 1, the posterior probability density function of $\mathbf{x}$ was obtained in Equation 94. It is clear, by inspection, that the mean of this Gaussian probability density function is given by Equation 95, thus

$$\hat{x}^{MMSE}(k) = \mathrm{E}[x \mid z] = \hat{x}^{MAP}(k)$$

the MMSE estimate coincides with the MAP estimate. This happens because the mean and mode of a Gaussian probability density function coincide.

3.6 Linear Estimators

Linear estimators are the most simple and most widely used method in sensor estimation. In its most simple terms, a linear estimator computes an estimate $\hat{\mathbf{x}}(n)$ for a parameter $\mathbf{x}$ on the basis of a linear combination of observations $\mathbf{z}(i)$, $i = 1, \cdots, n$, in the following form

$$\hat{\mathbf{x}}(n) = \sum_{i=1}^{n} \mathbf{W}(i)\mathbf{z}(i), \tag{116}$$

where the $n \times m$ matrices $\mathbf{W}(i)$ are termed the observation weights or *gains*. The principle advantage of linear estimators lies in their simplicity; once the gains have been determined the parameter estimate may easily be computed by simply combining the observations taken using Equation 116.

It should be clear that in almost all real situations a linear estimate will not, in any sense, be optimal, particularly if the relation between the observations and the unknown parameter (the observation model) is non-linear or if the underlying noise processes are not of some standard form. However, in many practical estimation problems, a linear estimator can provide very good estimates for the states of interest *regardless* of whether the observation model is linear or not. Indeed it is rarely possible to find a non-linear estimator which is both tractable and which significantly improves estimator performance[4].

Example 10

Suppose we take n observations z_i, $i = 1, \cdots, n$ of a fixed parameter x according to an assumed observation model

$$z_i = x + w_i, \qquad i = 1, \cdots, n,$$

where w_i (although usually unspecified) is an uncorrelated white sequence of random vectors with equal variances. A common method of finding an estimate $\hat{x}_n$ of x based on the observation sequence $\{z_i : i = 1, \cdots, n\}$ is to take the average (or sample mean) of these n observations:

$$\hat{x}_n = \frac{1}{n} \sum_{i=1}^{n} z_i \tag{117}$$

This is clearly a **linear estimator** *for x because the estimate $\hat{x}_n$ is a linear combination of the observations z_i. The gains in this case are all constant and equal to $1/n$.*

[4] A note of caution is required here. There is a large literature on non-linear estimators and it may well be the case that a very good non-linear estimator suggests itself for a specific estimation problem; in which case it should be used. It may also be the case that a linear estimator produces results which are simply unreasonable under a desired optimality criterion; again recourse to a non-linear estimator is advised. However 'in general', for the systems of interest and under 'reasonable' optimality criteria, it is difficult to improve on a linear estimate on grounds of both performance and tractability.

An estimate for the variance in this estimate can be obtained from the variance of the sample mean (mean-squared-error):

$$\hat{\sigma}_n^2 = \frac{1}{n}\sum_{i=1}^{n}(z_i - \hat{x}_n)^2 \tag{118}$$

There are an infinite variety of possible linear estimators each characterized by the choice of gains $\mathbf{W}(i)$. The gains are most often determined by minimizing some associated optimality criterion such as minimum mean-squared error or maximum likelihood for example. It is quite possible that the gains chosen with respect to one optimality criterion may also be optimal with respect to a variety of other criteria. The gains may also be chosen in some non-linear manner or be related to both the parameter of interest and the observations made. It is important to realize that even if the gains in Equation 116 are chosen in some non-linear manner the estimator itself is still linear.

Example 11

The estimate given by Equation 117 is optimal with respect to a number of different criteria. In particular if the criterion L is given by

$$L = \frac{1}{n}\sum_{i=1}^{n}(z_i - \hat{x}_n)^2,$$

(i.e. the mean squared-error) then clearly

$$\hat{x}_n = \arg\min_{x\in\Re} L = \arg_{x\in\Re}\left[\frac{\partial L}{\partial \hat{x}_n} = \mathbf{0}\right] = \frac{1}{n}\sum_{i=1}^{n} z_i.$$

If the criterion L is given by a weighted sum of squared-errors

$$L = \sum_{i=1}^{n} S_i(z_i - \hat{x}_n)^2,$$

where the S_i represent a weight or confidence in each observation then

$$\hat{x}_n = \left(\sum_{i=1}^{n} S_i\right)^{-1}\sum_{i=1}^{n} S_i z_i,$$

which is of course a linear estimate, which reduces to Equation 117 when all the weights are equal $S_i = S$.

More generally if we seek a linear estimate $\hat{\mathbf{x}}(n)$ of a parameter vector $\mathbf{x}$ and if the observation model is given in linear form as

$$\mathbf{z}_i = \mathbf{H}_i\mathbf{x} + \mathbf{w}_i, \qquad i = 1, \cdots, n,$$

where $\mathbf{H}_i$ *is the observation model associated with the* i^th *observation and* $\mathbf{w}_i$ *is an uncorrelated white sequence with equal covariance, under a least-squares criteria*

$$\hat{\mathbf{x}}(n) = \arg\min_{x \in \Re^n} \sum_{i=1}^{n} \left[\mathbf{z}_i - \mathbf{H}_i \hat{\mathbf{x}}(n)\right]^T \mathbf{Q}_i \left[\mathbf{z}_i - \mathbf{H}_i \hat{\mathbf{x}}(n)\right],$$

we obtain the linear estimate

$$\hat{\mathbf{x}}(n) = \left(\sum_{i=1}^{n} \mathbf{H}_i^T \mathbf{Q}_i \mathbf{H}_i\right)^{-1} \sum_{i=1}^{n} \mathbf{H}_i^T \mathbf{Q}_i \mathbf{z}_i$$

when the inverse exists

It should be noted that the gains chosen with respect to some optimality criteria will generate the *best linear estimate* from Equation 116 and not necessarily the best of all estimates under this criteria.

A recursive linear estimator is a linear estimator which computes an estimate $\hat{\mathbf{x}}(n)$ for the parameter $\mathbf{x}$ on the basis of a linear combination of a previous estimate $\hat{\mathbf{x}}(n-1)$ and a new observation $\mathbf{z}(n)$ in the following form

$$\hat{\mathbf{x}}(n) = \mathbf{W}_x \hat{\mathbf{x}}(n-1) + \mathbf{W}(n)\mathbf{z}(n).$$

In a recursive estimator, a new estimate is computed each time an observation is made. All past observations are summarized in this estimate and are not saved for later use, the estimator is consequently *memory-less.*

Example 12

Returning to the example of the sample mean as an estimator, suppose observations continue to be made and we wish to continually update our estimate of x *with these new observations. It is clearly not sensible to use Equations 117 and 118 for this task, because we would have to remember all past observations and do a considerable amount of arithmetic to calculate the new estimate. Fortunately, a little bit of algebra shows us that we do not need to do this. Indeed, suppose we currently have an estimate* $\hat{x}_{n-1}$ *for the parameter* x *based on the first* $n-1$ *observations. Consider now the arrival of a new observation* z_n*, so that the new estimate* $\hat{x}_n$ *becomes:*

$$\begin{aligned} \hat{x}_n &= \tfrac{1}{n} \textstyle\sum_{i=1}^{n} z_i \\ &= \tfrac{1}{n} \left[z_n + \textstyle\sum_{i=1}^{n-1} \mathbf{z}(i)\right] \\ &= \tfrac{1}{n} \left[z_n + (n-1)\hat{x}_{n-1}\right] \\ &= \hat{x}_{n-1} + \tfrac{1}{n} \left[z_n - \hat{x}_{n-1}\right] \end{aligned} \tag{119}$$

This is a **recursive linear estimator** *for x. It provides a revised estimate $\hat{x}_n$ of x, as a linear combination of the previous estimate $\hat{x}_{n-1}$ and the new observation z_n.*

The revised sampled variance can be recursively calculated from

$$\hat{\sigma}_n^2 = \frac{n-1}{n}\left[\hat{\sigma}_{n-1}^2 + \frac{1}{n}(z_n - \hat{x}_{n-1})^2\right] \tag{120}$$

The Kalman Filter is a recursive linear estimator which successively calculates an estimate for a state, that evolves over time, on the basis of periodic observations that are linearly related to this state. The Kalman Filter employs an explicit statistical model of how the parameter of interest $\mathbf{x}$ evolves over time and an explicit statistical model of how the observations that are made are related to this parameter. The gains employed in a Kalman Filter are chosen to ensure that the resulting estimate minimizes mean-squared error. Under certain assumptions about the observation and process models used, the same gains can be shown to be optimal under a range of other criteria. Because the Kalman filter employs explicit models of observation and process and because it does not require storage of all past observations, the Kalman filter algorithm is ideally suited to dealing with complex estimation problems.

4 The Linear Discrete-Time Kalman Filter

We shall begin by deriving the linear, discrete-time, Kalman filter. The starting point is a discrete time state-transition equation which describes the evolution of a state vector over time. This is coupled with an observation equation which describes how observations are made of the state. We choose to derive the Kalman filter algorithm in terms of a recursive linear estimator; that is we will assume a solution for the estimator which is linear and subsequently find the gains which minimize mean-squared error. The resulting Kalman filter algorithm recursively calculates estimates of the state based on a linear combination of the previous estimates and the new observations that have been made. The derivation in terms of a linear estimator exposes the essential simplicity of the Kalman filter as simply a weighted sum of observations and estimates. We will see one other derivation of the algorithm in this course, directly from the definition of the conditional mean. This derivation shows the role of the Kalman filter as an **observer**, and estimator for states that are not directly observed.

4.1 The System and Observation Models

We start by assuming that the system that we are interested in can be described by a simple linear, discrete-time state transition equation of the form

$$\mathbf{x}(k) = \mathbf{F}(k)\mathbf{x}(k-1) + \mathbf{B}(k)\mathbf{u}(k) + \mathbf{G}(k)\mathbf{v}(k), \tag{121}$$

where $\mathbf{x}(k-1)$ is the state at time $k-1$, $\mathbf{u}(k)$ is an input (or control) vector, $\mathbf{v}(k)$ is some additive motion noise, $\mathbf{B}(k)$ and $\mathbf{G}(k)$ are input and noie transition matricies, $\mathbf{F}(k)$ is the state transition matrix, and $\mathbf{x}(k)$ is the state at the following timestep k.

We assume that observations of the state of this system are made according to a linear equation in the form

$$\mathbf{z}(k) = \mathbf{H}(k)\mathbf{x}(k) + \mathbf{w}(k), \tag{122}$$

where $\mathbf{z}(k)$ is the observation made at time k, $\mathbf{x}(k)$ is the state at time k, $\mathbf{H}(k)$ is the observation matrix (or model), and $\mathbf{w}(k)$ is some additive observation noise.

We initially assume that the noises $\mathbf{v}(k)$ and $\mathbf{w}(k)$ are all Gaussian, temporally uncorrelated and zero-mean

$$\mathrm{E}[\mathbf{v}(k)] = \mathrm{E}[\mathbf{w}(k)] = \mathbf{0}, \qquad \forall k,$$

with corresponding covariance

$$\mathrm{E}[\mathbf{v}(i)\mathbf{v}^T(j)] = \delta_{ij}\mathbf{Q}(i), \quad \mathrm{E}[\mathbf{w}(i)\mathbf{w}^T(j)] = \delta_{ij}\mathbf{R}(i).$$

We also assume that the process and observation noises are uncorrelated

$$\mathrm{E}[\mathbf{v}(i)\mathbf{w}^T(j)] = \mathbf{0}, \qquad \forall i, j.$$

This last assumption is not absolutely necessary, indeed in many cases, where the sensor itself is affected by the motion of the platform whose location is to be measured for example, it is important to take account of this correlation.

Example 13

Consider the linear continuous-time model for the motion of a particle moving with approximately constant velocity:

$$\begin{bmatrix} \dot{x}(t) \\ \ddot{x}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ \dot{x}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ v(t) \end{bmatrix}.$$

In this case the state-transition matrix from Equation 64 over a time interval ΔT is given by

$$\mathbf{\Phi}(\Delta T) = \begin{bmatrix} 1 & \Delta T \\ 0 & 1 \end{bmatrix}.$$

On the assumption that the process noise is white and uncorrelated with strength $q(t)$; with $\mathrm{E}[v(t)] = 0$ and $\mathrm{E}[v(t)v(\tau)] = q(t)\delta(t-\tau)$, then the equivalent discrete-time noise process is given by

$$\mathbf{v}(k) = \int_0^{\Delta T} \begin{bmatrix} \tau \\ 1 \end{bmatrix} v(k\Delta T + \tau)\mathrm{d}\tau$$

and

$$\mathbf{Q}(k) = \mathrm{E}[\mathbf{v}(k)\mathbf{v}^T(k)] = \int_0^{\Delta T} \begin{bmatrix} \tau \\ 1 \end{bmatrix} \begin{bmatrix} \tau & 1 \end{bmatrix} q \, \mathrm{d}\tau = \begin{bmatrix} \Delta T^3/3 & \Delta T^2/2 \\ \Delta T^2/2 & \Delta T \end{bmatrix} q.$$

With the definitions given in Equation 73, the equivalent discrete-time model is given by

$$\begin{bmatrix} x(k) \\ \dot{x}(k) \end{bmatrix} = \begin{bmatrix} 1 & \Delta T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x(k-1) \\ \dot{x}(k-1) \end{bmatrix} + \begin{bmatrix} \Delta T^2/2 \\ \Delta T \end{bmatrix} v(k).$$

If observations are made at each time step k of the location of the particle the observation model will be in the form

$$z_x = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ \dot{x}(k) \end{bmatrix} + w(k), \qquad \mathrm{E}[w^2(k)] \ = \sigma_r^2$$

4.2 Definitions

The Kalman filter algorithm produces estimates that minimize mean-squared estimation error conditioned on a given observation sequence

$$\hat{\mathbf{x}}(i \mid j) = \arg \min_{\hat{\mathbf{x}}(i \mid j) \in \Re^n} \mathrm{E}\left[(\mathbf{x}(i) - \hat{\mathbf{x}}(i \mid j))(\mathbf{x}(i) - \hat{\mathbf{x}}(i \mid j))^T \mid \mathbf{z}(1), \cdots, \mathbf{z}(j) \right]. \quad (123)$$

As we have seen previously the estimate obtained is simply the expected value of the state at time i conditioned on the observations up to time j. The estimate is thus defined as the conditional mean

$$\hat{\mathbf{x}}(i \mid j) \stackrel{\triangle}{=} \mathrm{E}[\mathbf{x}(i) \mid \mathbf{z}(1), \cdots, \mathbf{z}(j)] \stackrel{\triangle}{=} \mathrm{E}[\mathbf{x}(i) \mid \mathbf{Z}^j]. \quad (124)$$

The estimate variance is defined as the mean squared error in this estimate

$$\mathbf{P}(i \mid j) \stackrel{\triangle}{=} \mathrm{E}\left[(\mathbf{x}(i) - \hat{\mathbf{x}}(i \mid j))(\mathbf{x}(i) - \hat{\mathbf{x}}(i \mid j))^T \mid \mathbf{Z}^j \right]. \quad (125)$$

The estimate of the state at a time k given all information up to time k will be written as $\hat{\mathbf{x}}(k \mid k)$. The estimate of the state at a time k given only information up to time $k-1$ is called a one-step-ahead prediction (or just a prediction) and is written as $\hat{\mathbf{x}}(k \mid k-1)$. Indeed, taking expectations of Equation 121 conditioned on all the observations up to time $k-1$ we can obtain an equation for the prediction $\hat{\mathbf{x}}(k \mid k-1)$ in terms of the previous estimate $\hat{\mathbf{x}}(k-1 \mid k-1)$:

$$\begin{aligned} \hat{\mathbf{x}}(k \mid k-1) &= \mathrm{E}[\mathbf{x}(k) \mid \mathbf{Z}^{k-1}] \\ &= \mathrm{E}[\mathbf{F}(k)\mathbf{x}(k-1) + \mathbf{B}(k)\mathbf{u}(k) + \mathbf{G}(k)\mathbf{v}(k) \mid \mathbf{Z}^{k-1}] \\ &= \mathbf{F}(k)\mathrm{E}[\mathbf{x}(k-1) \mid \mathbf{Z}^{k-1}] + \mathbf{B}(k)\mathbf{u}(k) + \mathbf{G}(k)E[\mathbf{v}(k) \mid \mathbf{Z}^{k-1}] \\ &= \mathbf{F}(k)\hat{\mathbf{x}}(k-1 \mid k-1) + \mathbf{B}(k)\mathbf{u}(k), \end{aligned} \quad (126)$$

where we have used the fact that the process noise is assumed to have zero mean and that the control input $\mathbf{u}(k)$ is known perfectly.

Similarly, the prediction variance can be found in terms of the variance of the previous estimate by first subtracting Equation 126 from Equation 121, and taking expectations conditioned on all observations up to time $k-1$

$$\begin{aligned}
\mathbf{P}(k \mid k-1) &= \mathrm{E}[(\mathbf{x}(k) - \hat{\mathbf{x}}(k \mid k-1))(\mathbf{x}(k) - \hat{\mathbf{x}}(k \mid k-1))^T \mid \mathbf{Z}^{k-1}] \\
&= \mathbf{F}(k)\mathrm{E}[(\mathbf{x}(k-1) - \hat{\mathbf{x}}(k-1 \mid k-1)) \\
&\quad \times (\mathbf{x}(k-1) - \hat{\mathbf{x}}(k-1 \mid k-1))^T \mid \mathbf{Z}^{k-1}]\mathbf{F}^T(k) \\
&\quad + \mathbf{G}(k)\mathrm{E}[\mathbf{v}(k)\mathbf{v}^T(k)]\mathbf{G}^T(k) \\
&= \mathbf{F}(k)\mathbf{P}(k-1 \mid k-1)\mathbf{F}^T(k) + \mathbf{G}(k)\mathbf{Q}(k)\mathbf{G}^T(k),
\end{aligned} \tag{127}$$

where we have used the fact that previous estimates $\hat{\mathbf{x}}(k-1 \mid k-1)$ and current noises $\mathbf{v}(k)$ are uncorrelated. Note also the cancellation of the control term $\mathbf{u}(k)$ – it has no effect on the estimation accuracy because it is assumed perfectly known.

Example 14

Returning to the previous example of constant velocity one-dimensional particle motion, the one-step state prediction is simply given by

$$\begin{bmatrix} x(k|k-1) \\ \dot{x}(k|k-1) \end{bmatrix} = \begin{bmatrix} 1 & \Delta T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x(k-1|k-1) \\ \dot{x}(k-1|k-1) \end{bmatrix}.$$

The covariance in this prediction is computed from

$$\begin{bmatrix} P_{xx}(k|k-1) & P_{x\dot{x}}(k|k-1) \\ P_{x\dot{x}}(k|k-1) & P_{\dot{x}\dot{x}}(k|k-1) \end{bmatrix}$$

$$= \begin{bmatrix} 1 & \Delta T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} P_{xx}(k-1|k-1) & P_{x\dot{x}}(k-1|k-1) \\ P_{x\dot{x}}(k-1|k-1) & P_{\dot{x}\dot{x}}(k-1|k-1) \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \Delta T & 1 \end{bmatrix} + \mathbf{Q}(k)$$

$$= \begin{bmatrix} P_{xx}(k-1|k-1) + 2\Delta T P_{x\dot{x}}(k-1|k-1) + \Delta T^2 P_{\dot{x}\dot{x}}(k-1|k-1) + q\Delta T^3/3 \\ P_{x\dot{x}}(k-1|k-1) + \Delta T P_{\dot{x}\dot{x}}(k-1|k-1) + q\Delta T^2/2 \end{bmatrix}$$

$$\begin{bmatrix} P_{x\dot{x}}(k-1|k-1) + \Delta T P_{\dot{x}\dot{x}}(k-1|k-1) + q\Delta T^2/2 \\ P_{\dot{x}\dot{x}}(k-1|k-1) + q\Delta T \end{bmatrix}$$

The structure of the prediction stage is clear from the previous example. To predict the state we simply take our best estimate from the previous time-step and multiply this

by the state transition matrix on the assumption that the noise term may be ignored; that is we use our last estimate to generate a best-guess as to the state at the current time. The uncertainty in the prediction is clearly larger than the uncertainty in the previous estimate simply because the prediction is a 'guess'. This is reflected by the addition of the process-noise covariance. The second component of the prediction uncertainty is that position variance is increased by the addition of the term $\Delta T^2 P_{\dot{x}\dot{x}}(k-1|k-1)$ due to uncertainty in the previous estimate of velocity, whereas the velocity uncertainty simply increases due to the addition of process variance. This is to be expected; uncertainty in position prediction depends on uncertainty in the velocity but uncertainty in velocity does not depend on uncertainty in position. The off-diagonal terms, the cross-correlation between position and velocity, are increased to reflect the fact that the prediction position variance depends on the velocity variance. Indeed, because the state transition matrix $\mathbf{F}(k)$ is by definition inevitable the covariance in the previous estimate can be recovered from the prediction covariance; the cross-correlations reflect the coupling between state estimates and predictions caused by the dynamics modeled by the matrix $\mathbf{F}(k)$. We shall see that this coupling is crucial to the estimation of non-observed states and consequently argues that the development of a good dynamics model is essential to the estimation process.

4.3 Description of the Problem

The basic problem that we want to address is that of finding a *recursive linear estimator* for the state $\mathbf{x}(\cdot)$.

Recursive: We start by assuming that at a time $k-1$ we already have an estimate $\hat{\mathbf{x}}(k-1 \mid k-1)$ for the state which includes all information up to time $k-1$. At the next time step k we use this estimate to generate a prediction $\hat{\mathbf{x}}(k \mid k-1)$ for the state and we take another observation $\mathbf{z}(k)$. We want to find an estimate $\hat{\mathbf{x}}(k \mid k)$ for the state at time k which incorporates this new observation but which is based only on the prediction $\hat{\mathbf{x}}(k \mid k-1)$ and the new observation $\mathbf{z}(k)$.

Linear: We assume that the estimate that we are looking for will be a linear weighted sum of the prediction and the new observation in the following form:

$$\hat{\mathbf{x}}(k \mid k) = \mathbf{W}'(k)\hat{\mathbf{x}}(k \mid k-1) + \mathbf{W}(k)\mathbf{z}(k), \tag{128}$$

where $\mathbf{W}'(k)$ and $\mathbf{W}(k)$ are two (time varying) weighting (or gain) matrices.

The problem of finding an optimal linear estimator for $\hat{\mathbf{x}}(k \mid k)$ reduces to finding appropriate values for the two gain matrices $\mathbf{W}'(k)$ and $\mathbf{W}(k)$. In the Kalman filter algorithm these gains are chosen to minimize the conditional mean squared estimation error. The estimation error is defined by

$$\tilde{\mathbf{x}}(k \mid k) = \hat{\mathbf{x}}(k \mid k) - \mathbf{x}(k). \tag{129}$$

The conditional mean squared estimation error is given by

$$\begin{aligned} L(k) &= \mathrm{E}[\tilde{\mathbf{x}}^T(k \mid k)\tilde{\mathbf{x}}(k \mid k) \mid \mathbf{Z}^k] \\ &= \mathrm{trace}\left[\mathrm{E}[\tilde{\mathbf{x}}(k \mid k)\tilde{\mathbf{x}}^T(k \mid k) \mid \mathbf{Z}^k]\right] \\ &= \mathrm{trace}[\mathbf{P}(k \mid k)]. \end{aligned} \tag{130}$$

To find the values for the gain matrices which minimize $L(k)$ we use a two-step process. First, we use the fact that the estimate which minimizes mean-squared error is equal to the conditional mean and so must make the expected estimation error equal to zero. This 'unbiased estimate' condition allows us to eliminate the gain $\mathbf{W}'(k)$ from Equation 128. Second, we compute the estimate covariance $\mathbf{P}(k \mid k)$, which will be a function of the remaining gain $\mathbf{W}(k)$. We then substitute this into Equation 130 and minimize with respect to the gain matrix to determine the value which minimizes mean-squared estimation error.

4.4 The Unbiased Condition

An estimate is said to be unbiased if the conditional expected error between estimate and true state is zero. This will clearly be the case for an estimator $\hat{\mathbf{x}}(k \mid k)$ which is equal to the conditional mean $\mathrm{E}[\mathbf{x}(k) \mid \mathbf{Z}^k]$ as defined by Equation 124,

$$\begin{aligned} \mathrm{E}[\tilde{\mathbf{x}}(k \mid k) \mid \mathbf{Z}^k] &= \mathrm{E}[\hat{\mathbf{x}}(k \mid k) - \mathbf{x}(k) \mid \mathbf{Z}^k] \\ &= \hat{\mathbf{x}}(k \mid k) - \mathrm{E}[\mathbf{x}(k) \mid \mathbf{Z}^k] \\ &= \mathbf{0}. \end{aligned} \tag{131}$$

We assume that the estimate obtained at time $k-1$ is indeed the conditional mean and so is unbiased

$$\begin{aligned} \mathrm{E}[\tilde{\mathbf{x}}(k-1 \mid k-1) \mid \mathbf{Z}^{k-1}] &= \mathrm{E}[\hat{\mathbf{x}}(k-1 \mid k-1) - \mathbf{x}(k-1) \mid \mathbf{Z}^{k-1}] \\ &= \hat{\mathbf{x}}(k-1 \mid k-1) - \mathrm{E}[\mathbf{x}(k-1) \mid \mathbf{Z}^{k-1}] \\ &= \mathbf{0}. \end{aligned} \tag{132}$$

It follows that the expected value of the prediction error $\tilde{\mathbf{x}}(k \mid k-1)$ conditioned on the observation sequence $\mathbf{Z}^{k-1}$ is also zero

$$\begin{aligned} \mathrm{E}[\tilde{\mathbf{x}}(k \mid k-1) \mid \mathbf{Z}^{k-1}] &= \mathrm{E}[\hat{\mathbf{x}}(k \mid k-1) - \mathbf{x}(k) \mid \mathbf{Z}^{k-1}] \\ &= \mathbf{F}(k)\mathrm{E}[\hat{\mathbf{x}}(k-1 \mid k-1) - \mathbf{x}(k-1) \mid \mathbf{Z}^{k-1}] + \mathrm{E}[\mathbf{v}(k)] \\ &= \mathbf{0}. \end{aligned} \tag{133}$$

Furthermore, because the prediction error is independent of the current observation, we have

$$\mathrm{E}[\tilde{\mathbf{x}}(k \mid k-1) \mid \mathbf{Z}^k] = \mathrm{E}[\tilde{\mathbf{x}}(k \mid k-1) \mid \mathbf{Z}^{k-1}] = \mathbf{0}. \tag{134}$$

We can now use Equations 128 and 122 to find an expression for the estimation error $\tilde{\mathbf{x}}(k \mid k)$ in terms of the unknown gains

$$\begin{aligned} \tilde{\mathbf{x}}(k \mid k) &= \hat{\mathbf{x}}(k \mid k) - \mathbf{x}(k) \\ &= \mathbf{W}'(k)\hat{\mathbf{x}}(k \mid k-1) + \mathbf{W}(k)\mathbf{z}(k) - \mathbf{x}(k) \\ &= \mathbf{W}'(k)\hat{\mathbf{x}}(k \mid k-1) + \mathbf{W}(k)\mathbf{H}(k)\mathbf{x}(k) + \mathbf{W}(k)\mathbf{w}(k) - \mathbf{x}(k) \\ &= [\mathbf{W}'(k) + \mathbf{W}(k)\mathbf{H}(k) - \mathbf{1}]\,\mathbf{x}(k) + \mathbf{W}'(k)\tilde{\mathbf{x}}(k \mid k) + \mathbf{W}(k)\mathbf{w}(k). \end{aligned} \tag{135}$$

Using the fact that $\mathrm{E}[\mathbf{w}(k)] = \mathbf{0}$ and $\mathrm{E}[\tilde{\mathbf{x}}(k \mid k-1) \mid \mathbf{Z}^k] = \mathbf{0}$, we find that the conditional mean estimation error is given by

$$\mathrm{E}[\tilde{\mathbf{x}}(k \mid k) \mid \mathbf{Z}^k] = [\mathbf{W}'(k) + \mathbf{W}(k)\mathbf{H}(k) - \mathbf{1}]\,\mathrm{E}[\mathbf{x}(k) \mid \mathbf{Z}^k]. \tag{136}$$

For this to be zero and the estimate to be unbiased we are required to have $[\mathbf{W}'(k) + \mathbf{W}(k)\mathbf{H}(k) - \mathbf{1}] = \mathbf{0}$, or equivalently, $\mathbf{W}'(k) = \mathbf{1} - \mathbf{W}(k)\mathbf{H}(k)$. Substituting this into Equation 128 we find that the unbiased linear estimate can be found from

$$\begin{aligned} \hat{\mathbf{x}}(k \mid k) &= [\mathbf{1} - \mathbf{W}(k)\mathbf{H}(k)]\hat{\mathbf{x}}(k \mid k-1) + \mathbf{W}(k)\mathbf{z}(k) \\ &= \hat{\mathbf{x}}(k \mid k-1) + \mathbf{W}(k)\left[\mathbf{z}(k) - \mathbf{H}(k)\hat{\mathbf{x}}(k \mid k-1)\right]. \end{aligned} \tag{137}$$

Notice that $\mathbf{H}(k)\hat{\mathbf{x}}(k \mid k-1)$ is essentially a predicted observation, so that the latter form of this estimation equation can be interpreted as the sum of the prediction, plus some fraction $\mathbf{W}(k)$ of the difference between actual and predicted observations. That is, Equation 137 has a "predictor-corrector" structure.

4.5 Calculation of the Error Covariance

We are now interested in calculating the variance on the estimate $\hat{\mathbf{x}}(k \mid k)$ to be used in Equation 130. Substituting Equations 121 and 137 into Equation 129 gives an equation for the error in the estimate

$$\tilde{\mathbf{x}}(k \mid k) = [\mathbf{1} - \mathbf{W}(k)\mathbf{H}(k)]\tilde{\mathbf{x}}(k \mid k-1) + \mathbf{W}(k)\mathbf{w}(k). \tag{138}$$

The variance in the estimate is defined as the conditional mean squared error in the estimate, so squaring Equation 138 and taking expectations gives

$$\begin{aligned}\mathbf{P}(k \mid k) &= \mathrm{E}[\tilde{\mathbf{x}}(k \mid k)\tilde{\mathbf{x}}^T(k \mid k) \mid \mathbf{Z}^k] \\ &= (\mathbf{1} - \mathbf{W}(k)\mathbf{H}(k))\mathrm{E}[\tilde{\mathbf{x}}(k \mid k-1)\tilde{\mathbf{x}}^T(k \mid k-1) \mid \mathbf{Z}^{k-1}](\mathbf{1} - \mathbf{W}(k)\mathbf{H}(k))^T \\ &\quad +\mathbf{W}(k)\mathrm{E}[\mathbf{w}(k)\mathbf{w}^T(k)]\mathbf{W}^T(k) \\ &\quad +2(\mathbf{1} - \mathbf{W}(k)\mathbf{H}(k))\mathrm{E}[\tilde{\mathbf{x}}(k \mid k-1)\mathbf{w}^T(k)]\mathbf{W}^T(k)\end{aligned} \tag{139}$$

and with

$$\begin{aligned}\mathrm{E}[\mathbf{w}(k)\mathbf{w}^T(k)] &\triangleq \mathbf{R}(k) \\ \mathrm{E}[\tilde{\mathbf{x}}(k \mid k-1)\tilde{\mathbf{x}}^T(k \mid k-1) \mid \mathbf{Z}^k] &\triangleq \mathbf{P}(k \mid k-1) \\ \mathrm{E}[\tilde{\mathbf{x}}(k \mid k-1)\mathbf{w}^T(k)] &\triangleq \mathbf{0}\end{aligned} \tag{140}$$

we obtain

$$\mathbf{P}(k \mid k) = (\mathbf{1} - \mathbf{W}(k)\mathbf{H}(k))\mathbf{P}(k \mid k-1)(\mathbf{1} - \mathbf{W}(k)\mathbf{H}(k))^T + \mathbf{W}(k)\mathbf{R}(k)\mathbf{W}^T(k) \tag{141}$$

This is the variance in our new updated estimate in terms of the prediction variance $\mathbf{P}(k \mid k-1)$, the observation noise $\mathbf{R}(k)$, and the unknown gain matrix $\mathbf{W}(k)$.

4.6 The Choice of Gain

In the Kalman Filter, the gain matrix $\mathbf{W}(k)$ is theorem to minimize conditional mean-squared estimation error given by Equation 130 as

$$L(k) = \mathrm{E}[\tilde{\mathbf{x}}^T(k \mid k)\tilde{\mathbf{x}}(k \mid k)] = \mathrm{trace}[\mathbf{P}(k \mid k)]. \tag{142}$$

Using the fact that for any matrix $\mathbf{A}$ and a symmetric matrix $\mathbf{B}$,

$$\frac{\partial}{\partial \mathbf{A}}\left(\mathrm{trace}(\mathbf{A}\mathbf{B}\mathbf{A}^T)\right) = 2\mathbf{A}\mathbf{B}$$

and substituting Equation 141 in to Equation 142, differentiating with respect to the gain matrix $\mathbf{W}(k)$ and setting equal to zero gives:

$$\frac{\partial L}{\partial \mathbf{W}(k)} = -2(\mathbf{1} - \mathbf{W}(k)\mathbf{H}(k))\mathbf{P}(k \mid k-1)\mathbf{H}^T(k) + 2\mathbf{W}(k)\mathbf{R}(k) = \mathbf{0}.$$

Rearranging provides an expression for the gain matrix as

$$\mathbf{W}(k) = \mathbf{P}(k \mid k-1)\mathbf{H}^T(k)\Big[\mathbf{H}^T(k)\mathbf{P}(k \mid k-1)\mathbf{H}^T(k) + \mathbf{R}(k)\Big]^{-1}. \tag{143}$$

With this gain matrix, Equation 137 becomes the optimal (linear) minimum mean-squared error estimator under the stated conditions.

4.7 Summary

It is useful at this point to summarize the Kalman Filter algorithm. We start with an estimate $\hat{\mathbf{x}}(k-1 \mid k-1)$ for the state $\mathbf{x}(k-1)$ at a time $k-1$ based on the observations $\mathbf{Z}^{k-1}$ made up to this time which is equal to the conditional mean or minimum mean-squared error estimate. The algorithm has two stages:

Prediction of state and variance at time k:

$$\hat{\mathbf{x}}(k \mid k-1) = \mathbf{F}(k)\hat{\mathbf{x}}(k-1 \mid k-1) + \mathbf{B}(k)\mathbf{u}(k) \tag{144}$$

$$\mathbf{P}(k \mid k-1) = \mathbf{F}(k)\mathbf{P}(k-1 \mid k-1)\mathbf{F}^T(k) + \mathbf{G}(k)\mathbf{Q}(k)\mathbf{G}^T(k) \tag{145}$$

Update of state and variance at time k on the basis of the predicted state and variance and the new observation $\mathbf{z}(k)$

$$\hat{\mathbf{x}}(k \mid k) = \hat{\mathbf{x}}(k \mid k-1) + \mathbf{W}(k)\left(\mathbf{z}(k) - \mathbf{H}(k)\hat{\mathbf{x}}(k \mid k-1)\right) \tag{146}$$

$$\mathbf{P}(k \mid k) = (\mathbf{1} - \mathbf{W}(k)\mathbf{H}(k))\mathbf{P}(k \mid k-1)(\mathbf{1} - \mathbf{W}(k)\mathbf{H}(k))^T + \mathbf{W}(k)\mathbf{R}(k)\mathbf{W}^T(k) \tag{147}$$

where the gain matrix is given by

$$\mathbf{W}(k) = \mathbf{P}(k \mid k-1)\mathbf{H}^T(k)\Big[\mathbf{H}(k)\mathbf{P}(k \mid k-1)\mathbf{H}^T(k) + \mathbf{R}(k)\Big]^{-1} \tag{148}$$

Example 15

Returning to the example of constant velocity one-dimensional particle motion, with $\mathbf{H}(k) = [1\ 0]$ *we have*

$$\mathbf{H}(k)\mathbf{P}(k \mid k-1)\mathbf{H}^T(k) + \mathbf{R}(k) = P_{xx}(k|k-1) + r,$$

so from Equation 148 the gain matrix is given by

$$\begin{aligned}\mathbf{W}(k) &= \begin{bmatrix} P_{xx}(k|k-1) & P_{x\dot{x}}(k|k-1) \\ P_{x\dot{x}}(k|k-1) & P_{\dot{x}\dot{x}}(k|k-1)\end{bmatrix}\begin{bmatrix}1\\0\end{bmatrix}\frac{1}{P_{xx}(k|k-1)+r} \\ &= \frac{1}{P_{xx}(k|k-1)+r}\begin{bmatrix}P_{xx}(k|k-1)\\ P_{x\dot{x}}(k|k-1)\end{bmatrix}.\end{aligned}$$

In the Equation 146 this gives the updated estimates of position and velocity as

$$x(k|k) = \frac{r}{P_{xx}(k|k-1)+r}x(k|k-1) + \frac{P_{xx}(k|k-1)}{P_{xx}(k|k-1)+r}z_x(k)$$

$$\dot{x}(k|k) = \dot{x}(k|k-1) + \frac{P_{x\dot{x}}(k|k-1)}{P_{xx}(k|k-1)+r}\left(z_x(k) - x(k|k-1)\right)$$

The update equation for position shows clearly that the updated estimate is a linear weighted sum of previous estimate and observation. As we would expect from the unbiased condition the weights sum to one so that the weighted sum is normalized. Furthermore we see that the prediction is weighted by the variance in the observation so that the more uncertain we are in the observation the more we weight the prediction when computing the estimate. Likewise the observation is weighted by the variance or uncertainty in the prediction. The update equation for the velocity is even more interesting. The predicted velocity is updated by some fraction of how much we under or over estimate the predicted location observation. The weighting term is proportional to the cross-correlation between velocity and position predictions. That is, the cross-correlations resulting from the computation of the prediction covariance and consequently the formulation for the process model in the previous example has a direct impact on the estimation of unobserved quantities.

The updated variance may also be computed as

$$\begin{bmatrix} P_{xx}(k|k) & P_{x\dot{x}}(k|k) \\ P_{x\dot{x}}(k|k) & P_{\dot{x}\dot{x}}(k|k) \end{bmatrix} = \begin{bmatrix} \frac{rP_{xx}(k|k-1)}{P_{xx}(k|k-1)+r} & \frac{rP_{x\dot{x}}(k|k-1)}{P_{xx}(k|k-1)+r} \\ \frac{rP_{x\dot{x}}(k|k-1)}{P_{xx}(k|k-1)+r} & \frac{(r+P_{xx}(k|k-1))P_{\dot{x}\dot{x}}(k|k-1)-P_{x\dot{x}}^2(k|k-1)}{P_{xx}(k|k-1)+r} \end{bmatrix}$$

It can be seen that the position variance and the position-velocity cross-correlation are each reduced by a factor $\frac{r}{P_{xx}(k|k-1)+r}$ *after updating. The case of the velocity variance update is a little more complex. We first define the cross-correlation coefficient*

$$\rho = \frac{P_{x\dot{x}}(k|k-1)}{\sqrt{P_{xx}(k|k-1)P_{\dot{x}\dot{x}}(k|k-1)}}$$

whose magnitude must always be less than 1, then the velocity variance update may be written as

$$P_{\dot{x}\dot{x}}(k|k) = P_{\dot{x}\dot{x}}(k|k-1)\frac{r+[1-\rho^2]P_{xx}(k|k-1)}{P_{xx}(k|k-1)+r}.$$

Clearly when position and velocity are uncorrelated ($\rho = 0$) then the velocity variance after update is the same as the predicted velocity variance. In the limit as ρ increases toward 1, where position and velocity are completely correlated, the velocity variance update factor tends to $\frac{r}{P_{xx}(k|k-1)+r}$*; the same value as for the position variance. This agrees with what might be expected from the way we saw that the velocity estimate is itself updated by a factor proportional to the cross-correlation between position and velocity.*

4.8 The Innovation

A prediction can be made as to what observation will be made at a time k based on the observations that have been made up to time $k-1$ by simply taking expectations of the

observation Equation 122 conditioned on previous observations:

$$\begin{aligned}\hat{\mathbf{z}}(k \mid k-1) &\triangleq \mathrm{E}[\mathbf{z}(k) \mid \mathbf{Z}^{k-1}] \\ &= \mathrm{E}[\mathbf{H}(k)\mathbf{x}(k) + \mathbf{w}(k) \mid \mathbf{Z}^{k-1}] \\ &= \mathbf{H}(k)\hat{\mathbf{x}}(k \mid k-1)\end{aligned} \tag{149}$$

The difference between the observation $\mathbf{z}(k)$ and the predicted observation $\mathbf{H}(k)\hat{\mathbf{x}}(k \mid k-1)$ is termed the *innovation* (also sometimes called the residual) $\nu(k)$:

$$\nu(k) = \mathbf{z}(k) - \mathbf{H}(k)\hat{\mathbf{x}}(k \mid k-1) \tag{150}$$

The innovation is an important measure of the deviation between the filter estimates and the observation sequence. Indeed, because the 'true' states are not usually available for comparison with the estimated states, the innovation is often the only measure of how well the estimator is performing. In subsequent sections we will see that the innovation can be used to 'tune' the estimator and to build models of the observation and process noise sources. We will also see that the innovations can be used to 'validate' measurements before they are incorporated into the filtered estimates and can also be used to select between a finite number of different process models or identify different sensor failure modes. The innovation measure is of great practical value in estimator design.

The most important property of innovations is that they form an orthogonal, uncorrelated, white sequence;

$$\mathrm{E}[\nu(k) \mid \mathbf{Z}^{k-1}] = \mathbf{0}, \qquad \mathrm{E}[\nu(i)\nu^T(j)] = \mathbf{S}(i)\delta_{ij}. \tag{151}$$

The innovation clearly has zero mean

$$\begin{aligned}\mathrm{E}[\nu(k) \mid \mathbf{Z}^{k-1}] &= \mathrm{E}[\mathbf{z}(k) - \hat{\mathbf{z}}(k \mid k-1) \mid \mathbf{Z}^{k-1}] \\ &= \mathrm{E}[\mathbf{z}(k) \mid \mathbf{Z}^{k-1}] - \hat{\mathbf{z}}(k \mid k-1) \\ &= \mathbf{0}.\end{aligned} \tag{152}$$

The innovation variance can be obtained as follows

$$\begin{aligned}\mathbf{S}(k) &= \mathrm{E}\left[\nu(k)\nu^T(k)\right] \\ &= \mathrm{E}\left[(\mathbf{z}(k) - \mathbf{H}(k)\hat{\mathbf{x}}(k \mid k-1))(\mathbf{z}(k) - \mathbf{H}(k)\hat{\mathbf{x}}(k \mid k-1))^T\right] \\ &= \mathrm{E}\left[(\mathbf{H}(k)[\mathbf{x}(k) - \hat{\mathbf{x}}(k \mid k-1)] + \mathbf{w}(k))(\mathbf{H}(k)[\mathbf{x}(k) - \hat{\mathbf{x}}(k \mid k-1)] + \mathbf{w}(k))^T\right] \\ &= \mathbf{R}(k) + \mathbf{H}(k)\mathbf{P}(k \mid k-1)\mathbf{H}^T(k)\end{aligned} \tag{153}$$

and with $j < k$,

$$\begin{aligned} \mathrm{E}[\nu(k)\nu^T(j)] &= \mathrm{E}[\mathrm{E}[\nu(k)\nu^T(j) \mid \mathbf{Z}^{k-1}] \,] \\ &= \mathrm{E}[\mathrm{E}[\nu(k) \mid \mathbf{Z}^{k-1}] \; \nu^T(j)] \\ &= \mathbf{0}. \end{aligned} \tag{154}$$

As the innovations are also Gaussian, uncorrelatedness implies that the sequence is white.

It is also of interest to note the relation between the observation sequence and the innovation sequence. The joint probability density function associated with a sequence of observations $\mathbf{Z}^k$ may be expanded in terms of conditional probabilities as follows

$$\begin{aligned} f(\mathbf{Z}^k) &= f(\mathbf{z}(k), \mathbf{Z}^{k-1}) \\ &= f(\mathbf{z}(k) \mid \mathbf{Z}^{k-1}) f(\mathbf{Z}^{k-1}) \\ &= \textstyle\prod_{i=1}^{k} f(\mathbf{z}(i) \mid \mathbf{Z}^{i-1}) \end{aligned} \tag{155}$$

As each distribution $f(\mathbf{z}(i) \mid \mathbf{Z}^{i-1})$ is Gaussian, we have

$$\begin{aligned} f(\mathbf{z}(i) \mid \mathbf{Z}^{i-1}) &= \mathrm{N}\left[\mathbf{z}(i); \hat{\mathbf{z}}(i \mid i-1), \mathbf{S}(i)\right] \\ &= \mathrm{N}\left[\mathbf{z}(i) - \hat{\mathbf{z}}(i \mid i-1); \mathbf{0}, \mathbf{S}(i)\right] \\ &= \mathrm{N}\left[\nu(i); \mathbf{0}, \mathbf{S}(i)\right] \\ &= f(\nu(i)). \end{aligned} \tag{156}$$

That is, the joint probability density function of the measurement sequence $\mathbf{Z}^k$ is equal to the product of the marginal probability density functions of the corresponding innovations. The density $f(\mathbf{Z}^k)$ is called the filter likelihood function. Also of interest is the exponent of this density, called the modified log-likelihood function

$$\begin{aligned} \lambda(k) &= \textstyle\sum_{i=1}^{k} \nu^T(i)\mathbf{S}(i)\nu(i) \\ &= \lambda(k-1) + \nu^T(k)\mathbf{S}(k)\nu(k) \end{aligned} \tag{157}$$

The terms

$$q(i) = \nu^T(i)\mathbf{S}(i)\nu(i) \tag{158}$$

are known as the normalized innovations squared. It can be shown that the random variable $q(i)$ has a χ^2 distribution in $m = dim(\mathbf{z}(k))$ degrees of freedom, with mean m and variance $2m$. As we shall see, this can be used to provide confidence bounds for validating observations and filter performance.

The innovation and the innovation variance can be used to derive an alternative, simpler, form of the update equations:

$$\hat{\mathbf{x}}(k \mid k) = \hat{\mathbf{x}}(k \mid k-1) + \mathbf{W}(k)\nu(k) \tag{159}$$

$$\begin{aligned}
\mathbf{P}(k \mid k) &= \mathrm{E}[(\mathbf{x}(k) - \hat{\mathbf{x}}(k \mid k))(\mathbf{x}(k) - \hat{\mathbf{x}}(k \mid k))^T] \\
&= \mathrm{E}[(\mathbf{x}(k) - \hat{\mathbf{x}}(k \mid k-1) - \mathbf{W}(k)\nu(k)) \\
&\quad \times (\mathbf{x}(k) - \hat{\mathbf{x}}(k \mid k-1) - \mathbf{W}(k)\nu(k))^T] \\
&= \mathrm{E}[(\mathbf{x}(k) - \hat{\mathbf{x}}(k \mid k-1))(\mathbf{x}(k) - \hat{\mathbf{x}}(k \mid k-1))^T] \\
&\quad -\mathbf{W}(k)\mathrm{E}[\nu(k))\nu^T(k)]\mathbf{W}^T(k) \\
&= \mathbf{P}(k \mid k-1) - \mathbf{W}(k)\mathbf{S}(k)\mathbf{W}^T(k)
\end{aligned} \tag{160}$$

and from Equation 148

$$\mathbf{W}(k) = \mathbf{P}(k \mid k-1)\mathbf{H}^T(k)\mathbf{S}^{-1}(k) \tag{161}$$

5 Understanding the Kalman Filter

The Kalman filter mechanism is conceptually very simple. In essence it computes new estimates of the state of interest by simply taking a weighted average of a prediction, based on the previous estimate, and a new observation. Despite its essential simplicity the basic algorithm hides an elegance that makes it enormously powerful in a wide range of applications. In this section we will describe a number of important features of the Kalman filter and describe three different interpretations of the basic estimation process. The purpose of these interpretations is to show that the Kalman filter performs a number of different functions as an estimator which can be exploited in the solution of different estimation problems.

5.1 The Basic Filtering Cycle

The recursive nature of the Kalman filter gives the algorithm an essentially cyclic structure in which the same computations are performed at each time-step. Figure 3 makes this structure explicit. It shows three cycles; the first describing the evolution of the true state and the observations that are made of this state by the sensors, the second which describes the generation of estimates of the true state on the basis of these observations, and a third cycle which computes covariance information about the estimates and which also computes the gain matrix at each time step.

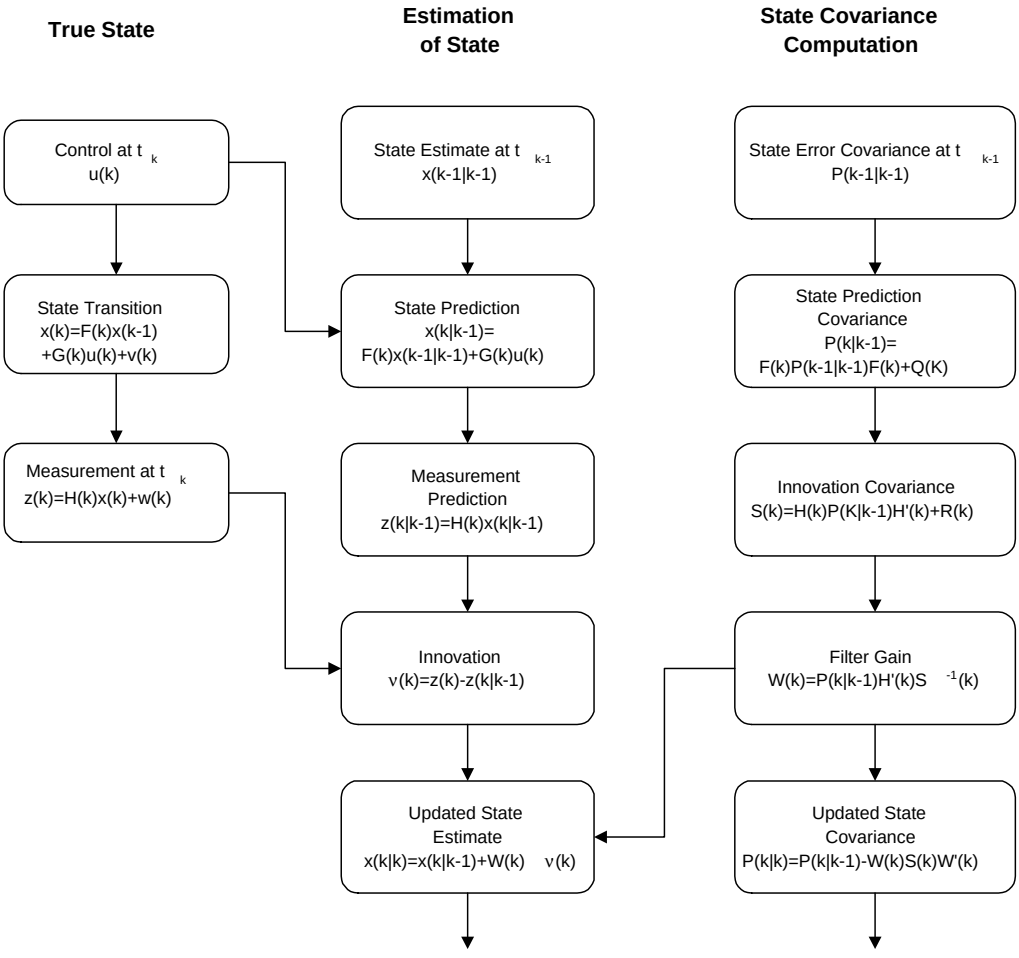


Figure 3: Block diagram of the Kalman filter cycle (after Barshalom and Fortmann 1988 [4])

The true state evolves over time according to a true state-space model given by Equation 61, with known input $\mathbf{u}(k)$ and subject to disturbances noise $\mathbf{v}(t)$. Observations are made of the true state according to a true observation model given by Equation 78. The observations are sampled at discrete points in time and are subject to observation noises $\mathbf{w}(k)$. The cycle associated with the evolution of the true state shown in Figure 3 makes it clear that the true state is never known; the only output from this stage are the sampled observations made of the true state.

The starting point for the estimation cycle is the generation of a state prediction from knowledge of the state estimate at the preceding time-step. This prediction is computed from Equation 144 and is based upon the known control input $\mathbf{u}(k)$ and the state transition model $\mathbf{F}(k)$. The state prediction is then used to compute a predicted observation from Equation 149 according to the model $\mathbf{H}(k)$. This predicted observation is then sub-

tracted from the true observation according to Equation 150 to give the innovation. The innovation is multiplied by the gain matrix (generated by the covariance loop) and added to the prediction to generate a state estimate according to Equation 159. The time index is then incremented and the cycle repeated. It is important to note that the only inputs to this cycle are the control input $\mathbf{u}(k)$, the observation $\mathbf{z}(k)$ and the gain matrix $\mathbf{W}(k)$. The state and observation models $\mathbf{F}(k)$ and $\mathbf{H}(k)$ must also be specified. The primary output is the state estimate $\hat{\mathbf{x}}(k \mid k)$.

The estimate covariance cycle also begins by generating a prediction covariance according to Equation 145 on the basis of the state model $\mathbf{F}(k)$ and the estimated process noise covariance $\mathbf{Q}(k)$. The innovation covariance is then computed according to Equation 153 on the basis of the observation model $\mathbf{H}(k)$ and the estimated observation noise $\mathbf{R}(k)$. The innovation covariance, together with the prediction covariance, is then used to compute the gain matrix $\mathbf{W}(k)$ according to Equation 161. The gain matrix is passed to the state estimation loop and is also used to compute the updated state covariance $\mathbf{P}(k \mid k)$ according to Equation 160. Finally the time index is incremented and the cycle is repeated. The only inputs to this covariance loop are the estimated process noise covariance $\mathbf{Q}(k)$ and the estimated observation noise covariance $\mathbf{R}(k)$. As in the estimation loop, the process model $\mathbf{F}(k)$ and the observation model must also be specified. The primary output from this loop is the estimate covariance $\mathbf{P}(k \mid k)$. It is important to note that the covariance loop is independent of the observations that are made and so is also independent of the evolution of the true state. This is simply because the information required by the covariance loop is the covariance of the process and observation noises and these are only available in terms of the estimated covariances $\mathbf{Q}(k)$ and $\mathbf{R}(k)$ respectively. One consequence of this is that all state covariance information can be computed off-line prior to any observations being made. It follows that all the gain matrices required by the state estimation cycle can also be computed off-line. This is significant because the majority of the computational requirements of the estimation process are dedicated to this task. As will be shown in a subsequent section, the state covariance tends to a steady-state value over time as, consequently, does the gain matrix. This fact can be used to construct estimators in which the gain matrix is time-invariant (and usually equal to its steady-state value) eliminating the need for the covariance loop and leading to significantly reduced computational requirements.

5.2 The Kalman Filter as a Weighted Average

With the definition of the Kalman filter as a linear weighted sum of prediction and observation, the most obvious interpretation of the estimation algorithm is simply as an averaging process. The weights in this averaging process are $\mathbf{1} - \mathbf{W}(k)\mathbf{H}(k)$ associated with the prediction and $\mathbf{W}(k)$ associated with the observation. It is useful to determine

these weights explicitly. Rewriting Equation 148 for the gain matrix we have

$$\begin{aligned}\mathbf{W}(k) &= \mathbf{P}(k \mid k-1)\mathbf{H}^T(k)\Big[\mathbf{H}(k)\mathbf{P}(k \mid k-1)\mathbf{H}^T(k) + \mathbf{R}(k)\Big]^{-1} \\ &= \mathbf{P}(k \mid k-1)\mathbf{H}^T(k)\Big(\mathbf{H}(k)\mathbf{P}(k \mid k-1)\mathbf{H}^T(k)\Big)^{-1} \times \\ &\quad \left[\Big(\mathbf{H}(k)\mathbf{P}(k \mid k-1)\mathbf{H}^T(k)\Big)^{-1} + \mathbf{R}^{-1}(k)\right]^{-1}\mathbf{R}^{-1}(k).\end{aligned} \tag{162}$$

Substituting this into Equation 146 and premultiplying by $\mathbf{H}(k)$ gives

$$\begin{aligned}\mathbf{H}(k)\hat{\mathbf{x}}(k \mid k) &= \mathbf{H}(k)\hat{\mathbf{x}}(k \mid k) + \Big[(\mathbf{H}(k)\mathbf{P}(k \mid k-1)\mathbf{H}^T(k))^{-1} + \mathbf{R}^{-1}(k)\Big]^{-1} \times \\ &\quad \mathbf{R}^{-1}(k)\left[\mathbf{z}(k) - \mathbf{H}(k)\hat{\mathbf{x}}(k \mid k-1)\right],\end{aligned} \tag{163}$$

and rearranging

$$\begin{aligned}\mathbf{H}(k)\hat{\mathbf{x}}(k \mid k) &= \Big[(\mathbf{H}(k)\mathbf{P}(k \mid k-1)\mathbf{H}^T(k))^{-1} + \mathbf{R}^{-1}(k)\Big]^{-1} \\ &\quad \times \Big[(\mathbf{H}(k)\mathbf{P}(k \mid k-1)\mathbf{H}^T(k))^{-1}\mathbf{H}(k)\hat{\mathbf{x}}(k \mid k-1) + \mathbf{R}^{-1}(k)\mathbf{z}(k)\Big].\end{aligned} \tag{164}$$

This is just a weighted sum of the predicted observation $\mathbf{H}(k)\hat{\mathbf{x}}(k \mid k-1)$ and the actual observation $\mathbf{z}(k)$ in which $(\mathbf{H}(k)\mathbf{P}(k+1 \mid k)\mathbf{H}^T(k))^{-1}$ is the inverse of the prediction covariance projected in to observation space and is our confidence in the predicted observation, and $\mathbf{R}^{-1}(k)$ is the inverse observation noise covariance and is our confidence in the observation itself. The sum is normalized by the total confidence $[(\mathbf{H}(k)\mathbf{P}(k \mid k-1)\mathbf{H}^T(k))^{-1} + \mathbf{R}^{-1}(k)]^{-1}$ and the result is a new estimate for the state projected into the observation space as $\mathbf{H}(k)\hat{\mathbf{x}}(k \mid k)$. The covariance in this projected estimate is simply given by the normalization factor of the weighted sum

$$\mathbf{H}(k)\mathbf{P}(k \mid k)\mathbf{H}^T(k) = \Big[(\mathbf{H}(k)\mathbf{P}(k \mid k-1)\mathbf{H}^T(k))^{-1} + \mathbf{R}^{-1}(k)\Big]^{-1} \tag{165}$$

This averaging process works by first projecting the state vector into observation space as $\mathbf{H}(k)\hat{\mathbf{x}}(k \mid k-1)$ and corresponding covariance as $\mathbf{H}(k)\mathbf{P}(k \mid k-1)\mathbf{H}^T(k)$, where the observation $\mathbf{z}(k)$ and its corresponding covariance $\mathbf{R}(k)$ are directly available, and then computes an updated estimate $\mathbf{H}(k)\hat{\mathbf{x}}(k \mid k)$ as a weighted sum of these, again in observation space. Clearly, this interpretation of the Kalman filter only holds for those states which can be written as a linear combination of the observation vector.

Example 16

If the state and observation are one-dimensional with $\hat{\mathbf{x}}(k \mid k-1) = \overline{x}$ *and* $\mathbf{z}(k) = z$*, the observation model is simply the identity* $\mathbf{H}(k) = 1$*, and the corresponding covariances are given by* $\mathbf{P}(k \mid k-1) = \sigma_0^2$ *and* $\mathbf{R}(k) = \sigma^2$*, then the updated estimate is given by*

$$\begin{aligned} \hat{x} &= \frac{\left(\overline{x}/\sigma_0^2 + z/\sigma^2\right)}{\left(1/\sigma_0^2 + 1/\sigma^2\right)} \\ &= \frac{\sigma^2\overline{x} + \sigma_0^2 z}{\sigma^2 + \sigma_0^2} \\ &= \overline{x} + \frac{\sigma_0^2}{\sigma_0^2 + \sigma^2}(z - \overline{x}) \end{aligned} \tag{166}$$

which is simply the same weighted average obtained in Equation 95.

5.3 The Kalman Filter as an Observer

As we have seen, the Kalman filter is also able to provide estimates of states which are not a linear combination of the observation vector; estimates of velocity on the basis of observations of position, for example. It does this by employing the cross-correlation between observed and unobserved states to find that part of the observation information which is correlated with the unobserved states. In this sense, the Kalman filter is sometimes referred to as a **linear observer**.

This interpretation of the Kalman filter algorithm is most easily appreciated by considering the estimator directly as the conditional mean

$$\begin{aligned} \hat{\mathbf{x}}(k \mid k) &= \mathrm{E}[\mathbf{x}(k) \mid \mathbf{Z}^k] \\ &= \int_{-\infty}^{\infty} \mathbf{x}(k) f(\mathbf{x}(k) \mid \mathbf{Z}^k) \mathrm{d}\mathbf{x}. \end{aligned} \tag{167}$$

The conditional distribution for $\mathbf{x}(k)$ given the observation sequence $\mathbf{Z}^k$ can be rewritten as

$$f(\mathbf{x}(k) \mid \mathbf{Z}^k) = \frac{f(\mathbf{x}(k), \mathbf{z}(k) \mid \mathbf{Z}^{k-1})}{f(\mathbf{z}(k) \mid \mathbf{Z}^{k-1})}. \tag{168}$$

These two distributions are both considered to be Gaussian as follows:

$$\begin{aligned} f(\mathbf{x}(k)\mathbf{z}(k) \mid \mathbf{Z}^{k-1}) \propto \exp\Bigg(& -\frac{1}{2}\begin{bmatrix} (\mathbf{x}(k) - \hat{\mathbf{x}}(k \mid k-1))^T \\ (\mathbf{z}(k) - \hat{\mathbf{z}}(k \mid k-1))^T \end{bmatrix}^T \begin{bmatrix} \mathbf{P}_{xx} & \mathbf{P}_{xz} \\ \mathbf{P}_{xz}^T & \mathbf{P}_{zz} \end{bmatrix}^{-1} \\ & \times \begin{bmatrix} (\mathbf{x}(k) - \hat{\mathbf{x}}(k \mid k-1)) \\ (\mathbf{z}(k) - \hat{\mathbf{z}}(k \mid k-1)) \end{bmatrix} \Bigg) \end{aligned} \tag{169}$$

$$f(\mathbf{z}(k)) \propto \exp\left(-\frac{1}{2}(\mathbf{z}(k) - \hat{\mathbf{z}}(k \mid k-1))^T \mathbf{P}_{zz}^{-1} (\mathbf{z}(k) - \hat{\mathbf{z}}(k \mid k-1))\right). \tag{170}$$

We employ the matrix inversion lemma to rewrite the inverse covariance in Equation 169 as

$$\begin{bmatrix} \mathbf{P}_{xx} & \mathbf{P}_{xz} \\ \mathbf{P}_{xz}^T & \mathbf{P}_{zz} \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{I}_{xx} & -\mathbf{P}_{xx}\mathbf{P}_{xz}\mathbf{P}_{zz}^{-1} \\ -\mathbf{P}_{zz}^{-1}\mathbf{P}_{xz}^T\mathbf{I}_{xx} & \mathbf{P}_{zz}^{-1} + \mathbf{P}_{zz}^{-1}\mathbf{P}_{xz}^T\mathbf{I}_{xx}\mathbf{P}_{xz}\mathbf{P}_{zz}^{-1} \end{bmatrix} \tag{171}$$

where

$$\mathbf{I}_{xx} = \left(\mathbf{P}_{xx} - \mathbf{P}_{xz}\mathbf{P}_{zz}^{-1}\mathbf{P}_{xz}^T\right)^{-1}. \tag{172}$$

Substituting Equation 169, 170 and 171 into Equation 168, the exponent of the Gaussian distribution $f(\mathbf{x}(k) \mid \mathbf{Z}^k)$ is found to be proportional to

$$\begin{aligned} \lambda \;=\; & (\mathbf{x}(k) - \hat{\mathbf{x}}(k \mid k-1))^T \mathbf{I}_{xx} (\mathbf{x}(k) - \hat{\mathbf{x}}(k \mid k-1)) \\ & -2(\mathbf{x}(k) - \hat{\mathbf{x}}(k \mid k-1))^T \mathbf{I}_{xx}\mathbf{P}_{xz}\mathbf{P}_{zz}^{-1}(\mathbf{z}(k) - \hat{\mathbf{z}}(k \mid k-1)) \\ & +(\mathbf{z}(k) - \hat{\mathbf{z}}(k \mid k-1))^T \mathbf{P}_{zz}^{-1}\mathbf{P}_{xz}^T\mathbf{I}_{xx}\mathbf{P}_{xz}\mathbf{P}_{zz}^{-1}(\mathbf{z}(k) - \hat{\mathbf{z}}(k \mid k-1)) \end{aligned} \tag{173}$$

When $\mathbf{x}(k)$ is equal to the conditional mean $\hat{\mathbf{x}}(k \mid k)$ associated with this distribution, then Equation 173 becomes zero (corresponding to the central peak of the Gaussian). Completing the squares in Equation 173 gives

$$\lambda = (\mathbf{x}(k) - \hat{\mathbf{x}}(k \mid k))^T \mathbf{P}^{-1}(k \mid k)(\mathbf{x}(k) - \hat{\mathbf{x}}(k \mid k)), \tag{174}$$

with

$$\hat{\mathbf{x}}(k \mid k) = \hat{\mathbf{x}}(k \mid k-1) + \mathbf{P}_{xz}\mathbf{P}_{zz}^{-1}\left(\mathbf{z}(k) - \hat{\mathbf{z}}(k \mid k-1)\right) \tag{175}$$

and

$$\mathbf{P}(k \mid k) = \mathbf{I}_{xx}^{-1} = \mathbf{P}_{xx} - \mathbf{P}_{xz}\mathbf{P}_{zz}^{-1}\mathbf{P}_{xz}^T \tag{176}$$

Equations 175 makes it clear how new observations $\mathbf{z}(k)$ contribute to the updated estimate $\hat{\mathbf{x}}(k \mid k)$. First, the difference between observation and predicted observation is computed. Then the term $\mathbf{P}_{xz}$, by definition, tells us how the states are correlated with the observations made, and finally $\mathbf{P}_{zz}^{-1}$ is used to 'normalize' this correlation by the confidence we have in the new observation. This shows that provided we have knowledge of how the states are correlated to the observations made, we may use the observation information to estimate the state even in cases where the state is not a linear combination of the observation vector. The example developed in previous sections shows how this works to compute velocity as well as position from measurements of position alone.

An important question arising from this analysis is "when can a state be observed ?", that is under what conditions can we estimate a value for a state that is not directly measured. Observability simply asks if an initial state $\mathbf{x}(0)$ can be *uniquely* deduced from knowledge of an observation sequence $\mathbf{Z}^k$. This requires that the Observability Gramian

$$\mathcal{O}(0, N) \triangleq \sum_{i=1}^{N} \mathbf{F}^T(i)\mathbf{H}^T(i)\mathbf{H}(i)\mathbf{F}(i) \tag{177}$$

is non singular, or equivalently that the null space of $\mathcal{O}(0,N)$ is $\mathbf{0} \in \Re^n$. If this condition is met, then all states will be observable. Note that the rank of each term in the sum is at most m, so there is in general some minimal number $N \geq n/m$ of measurements that must be taken before the model can be completely observable. For time-invariant systems, the observability condition can be reduced to checking if the $n \times nm$ matrix

$$\left[\mathbf{H}^T; \mathbf{F}^T\mathbf{H}^T; \cdots; (\mathbf{F}^T)^{n-1}\right] \tag{178}$$

has rank n.

Example 17

There are two interesting cases of unobservability that occur frequently in particular. The first case occurs when the measurements that are made relate to a fixed linear combination of states. We will call this **measurement unobservability***. For example, if we have two states x_1 and x_2 and our observation model is described by*

$$z(k) = \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + w(k),$$

with a state transition model

$$\begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \mathbf{v}(k),$$

the observability criteria of Equation 178 becomes

$$\left[\begin{pmatrix} 1 \\ 1 \end{pmatrix}; \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} 1 \\ 1 \end{pmatrix}\right] = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix},$$

which is clearly rank deficient. This unobservability manifests itself in the Kalman filter algorithm as a singular state covariance matrix $\mathbf{P}(k \mid k)$; in which the two states are completely correlated. The algorithm will still function as normal, but will be extremely sensitive to changes in initial conditions. It should be carefully noted that it is quite possible to construct situations in which the observability condition is satisfied even when the measurements made are linear combinations of states.

A second unobservability situation occurs when the measured state is actually given by the integral or time-sum of the state of interest. For example, if we have a constant velocity model in the form

$$\begin{bmatrix} x(k) \\ \dot{x}(k) \end{bmatrix} = \begin{bmatrix} 1 & \Delta T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x(k-1) \\ \dot{x}(k-1) \end{bmatrix} + \begin{bmatrix} \Delta T^2 \\ \Delta T \end{bmatrix} v(k).$$

and we observe only velocity

$$z_{\dot{x}} = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} x(k) \\ \dot{x}(k) \end{bmatrix} + w(k),$$

then the observability criteria of Equation 178 becomes

$$\left[\begin{pmatrix}0\\1\end{pmatrix};\begin{pmatrix}1 & 0\\ \Delta T & 1\end{pmatrix}\begin{pmatrix}0\\1\end{pmatrix}\right] = \begin{bmatrix}0 & 0\\ 1 & 1\end{bmatrix}.$$

This type of unobservability manifests itself in the Kalman filter algorithm as a continual growth in the variance associated with the state x (although the state $\dot{x}$ will continue to be updated normally). Unobservability of this form is common in navigation problems when estimating position on the basis of input from a gyroscope or accelerometer.

Of these two types of unobservability, the first is potentially more problematic as it can often go unnoticed with potentially catastrophic consequences. Very similar types of effect (in which the state estimate covariance is singular) can occur if the system is also uncontrollable or when the states themselves are not linearly independent (theoretically impossible from the definition of the state transition matrix, but all too easy to do in particular !).

5.4 The Kalman Filter as a Projection Operation

A third method of interpreting the Kalman filter is as geometric operation that projects the true state $\mathbf{x}(k)$ on to the subspace spanned by the observation sequence $\mathbf{Z}^k$. This interpretation is propagation valuable in understanding the contributions made by different observations to the state estimation problem, particularly in situations where the state vector is constrained in some way.

Recall that fundamentally the estimates we are considering $\hat{\mathbf{x}}(k \mid k)$ are simply linear combinations of the observations made $\mathbf{Z}^k$ in the form

$$\hat{\mathbf{x}}(k \mid k) = \sum_{j=1}^{k} \alpha_j \mathbf{z}(j), \tag{179}$$

where α_j are a set of constants to be determined. Defining the subspace spanned by the first k observations as

$$\mathcal{M}(k) = \text{span}\{\mathbf{z}(1), \cdots, \mathbf{z}(k)\}\ , \tag{180}$$

it is clear from Equation 179 that the estimate $\hat{\mathbf{x}}(k \mid k)$ lies in the sub-space $\mathcal{M}(k)$. Furthermore, our objective in computing the estimate $\hat{\mathbf{x}}(k \mid k)$ is to minimize the conditional mean-squared error

$$\text{E}[(\mathbf{x}(k) - \hat{\mathbf{x}}(k \mid k))^T (\mathbf{x}(k) - \hat{\mathbf{x}}(k \mid k)) \mid \mathbf{Z}^k] \tag{181}$$

This is clearly minimized when the 'distance' between the estimate $\hat{\mathbf{x}}(k \mid k)$ and the true state $\mathbf{x}(k)$ is minimized. Of all $\hat{\mathbf{x}}(k \mid k) \in \mathcal{M}(k)$ the one that is 'closest' to $\mathbf{x}(k)$ is the perpendicular projection of $\mathbf{x}(k)$ into $\mathcal{M}(k)$. It follows that the estimation error

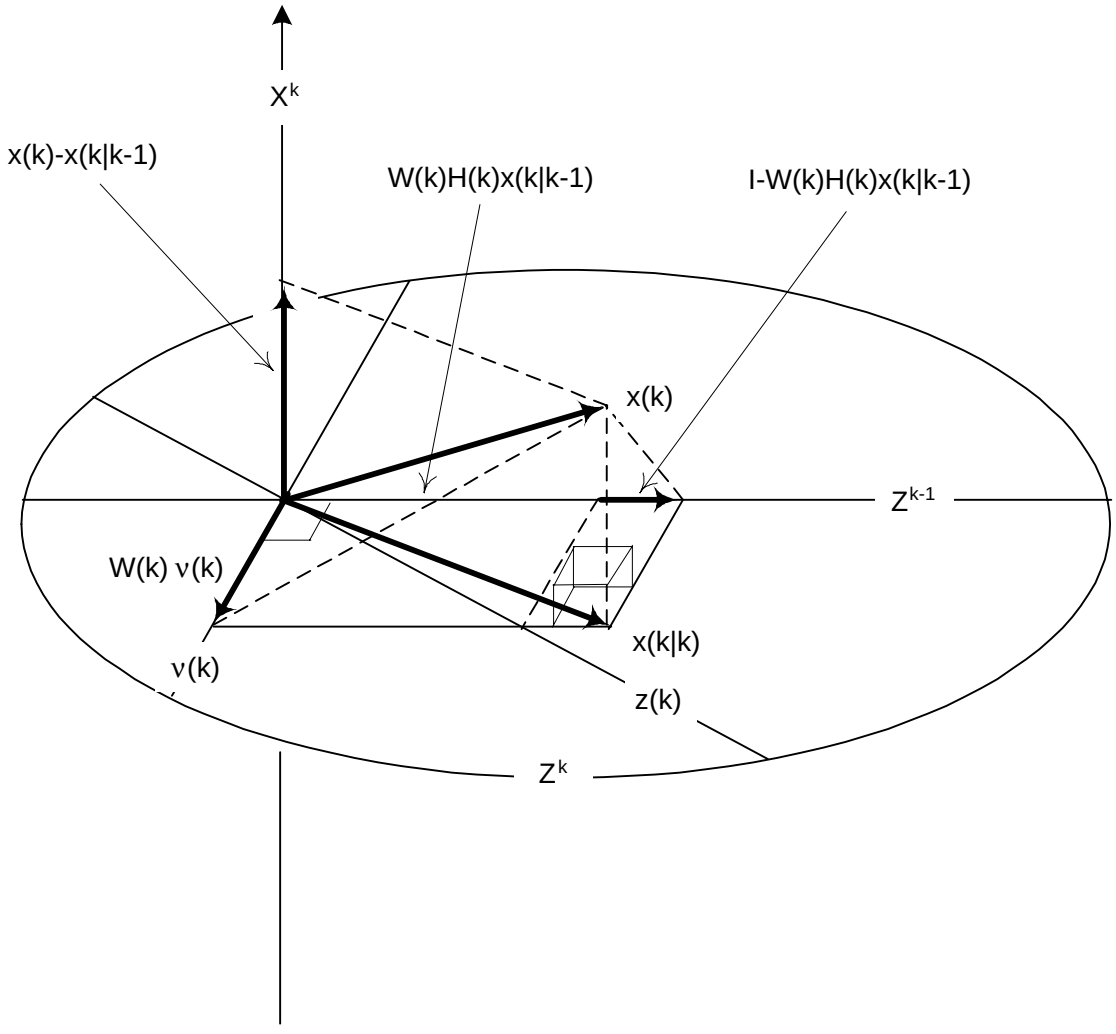


Figure 4: A projection diagram for the Kalman filter. The complete space is the true state space $\mathbf{X}^k$. The sub-space spanned by the first k observations is shown as the ellipse $\mathbf{Z}^k$. The space $\mathbf{Z}^{k-1}$ spanned by the first $k-1$ observations is shown as a line sub-space of the space $\mathbf{Z}^k$. The true state $\mathbf{x}(k)$ lies in the space $\mathbf{X}^k$, the estimate $\hat{\mathbf{x}}(k \mid k)$ lies in the space $\mathbf{Z}^k$ (because $\hat{\mathbf{x}}(k \mid k)$ is a linear combination of observations up to time k), prediction $\hat{\mathbf{x}}(k \mid k-1)$ lies in the space $\mathbf{Z}^{k-1}$. The estimate is a perpendicular projection of the true state on to the space $\mathbf{Z}^k$. By construction, the innovation space $\nu(k)$ is perpendicular to the space $\mathbf{Z}^{k-1}$. The new estimate $\hat{\mathbf{x}}(k \mid k)$ is the sum of: 1) the projection of $\mathbf{x}(k)$ onto $\mathbf{Z}^{k-1}$ along $\mathbf{z}(k)$, given by $[\mathbf{1} - \mathbf{W}(k)\mathbf{H}(k)]\hat{\mathbf{x}}(k \mid k-1)$; and 2) the projection of $\mathbf{x}(k)$ onto $\mathbf{z}(k)$ along $\mathbf{Z}^{k-1}$, given by $\mathbf{W}(k)\mathbf{z}(k)$. For comparison, the estimate that would result based only on the observation $\mathbf{z}(k)$ is $\hat{\mathbf{x}}(k \mid \mathbf{z}(k))$, the projection of $\mathbf{x}(k)$ onto $\mathbf{z}(k)$. Diagramatic method from [12] (see also [6])

$\tilde{\mathbf{x}}(k \mid k) = \mathbf{x}(k) - \hat{\mathbf{x}}(k \mid k)$ is perpendicular to $\mathcal{M}(k)$; $\tilde{\mathbf{x}}(k \mid k) \perp \mathcal{M}(k)$. This is a special case of the more general projection theorem for minimum mean-squared estimators.

The recursive Kalman filter may also be interpreted in terms of projection operations. First, we note that the innovation $\nu(k)$ at any one timestep is, by construction, orthogonal to the prediction $\hat{\mathbf{x}}(k \mid k-1)$:

$$\begin{aligned} &\mathrm{E}[(\mathbf{H}(k)\hat{\mathbf{x}}(k \mid k-1))^T(\mathbf{z}(k) - \mathbf{H}(k)\hat{\mathbf{x}}(k \mid k-1) \mid \mathbf{Z}^{k-1}] \\ &= (\mathbf{H}(k)\hat{\mathbf{x}}(k \mid k-1))^T\mathrm{E}[(\mathbf{z}(k) - \mathbf{H}(k)\hat{\mathbf{x}}(k \mid k-1) \mid \mathbf{Z}^{k-1}] \\ &= \mathbf{0}. \end{aligned} \tag{182}$$

The estimate of the state $\hat{\mathbf{x}}(k \mid k)$ is, of course simply the sum of the two orthogonal vectors $\hat{\mathbf{x}}(k \mid k-1)$ and $\mathbf{W}(k)\nu(k)$. Figure 4 shows this graphically using multi-dimensional space representation. The space spanned by the set $\mathbf{Z}^k$ is shown as a plane embedded in the overall state space, $\mathbf{Z}^{k-1}$ as a line embedded in the $\mathbf{Z}^k$ plane, the new observation $\mathbf{z}(k)$ as a line so arranged that $\mathbf{Z}^{k-1}$ and $\mathbf{z}(k)$ together generate $\mathbf{Z}^k$. The innovation $\nu(k)$ is described as a line orthogonal to $\mathbf{Z}^{k-1}$ also embedded in the $\mathbf{Z}^k$. The prediction $\hat{\mathbf{x}}(k \mid k-1)$ lies in the space $\mathbf{Z}^{k-1}$, $\mathbf{W}(k)\nu(k)$ lies in the space orthogonal to $\mathbf{Z}^{k-1}$, and the estimate $\hat{\mathbf{x}}(k \mid k)$ lies in the space $\mathbf{Z}^k$ and is again the projection of $\mathbf{x}(k)$ in to this space. Figure 4 makes clear the role of $\mathbf{W}(k)$ and $[\mathbf{1} - \mathbf{W}(k)\mathbf{H}(k)]$ as complimentary projections. The estimate can be thought of either as the orthogonal sum of prediction $\hat{\mathbf{x}}(k \mid k-1)$ and weighted innovation $\mathbf{W}(k)\nu(k)$, or as the sum of the parallel vectors $[\mathbf{1} - \mathbf{W}(k)\mathbf{H}(k)]\hat{\mathbf{x}}(k \mid k-1)$ and $\mathbf{W}(k)\mathbf{z}(k)$.

6 Implementing The Kalman Filter

Although the Kalman filter algorithm itself appears to be quite general and straightforward, its successful implementation tends to be very problem-specific, relying heavily on engineering judgment to adjust and tune process and sensor models. As a rule of thumb, it takes an order of magnitude more time and effort to tune and adjust a filter to work well, over the time it takes to implement the basic algorithm. This rule increases to two orders of magnitude in the case of non-linear process and observation models. There are two main reasons for this: First, in dealing with real data from a real process it is very rarely the cases that the true state of nature is known. That is, there is never any absolute truth by which to judge the performance of the algorithm, all that is available for this judgment are the observations which are themselves used by the filter. This leads to a problem of introspection in which the only way to judge the performance of a filter is with respect to other possible filter performances. The second reason for the difficulty of implementation is simply not knowing when the filter performance is limited by the adequacy of the sensor and process models rather than an inability to fine tune the algorithm: can

I do any better by improving my models or am I simply limited by the performance of the algorithm itself ? Again this is often a question of engineering judgment which can only be based on the comparison of different possible filters with respect to the observations that are made.

As in all matters of engineering judgment there are some well understood procedures and rules which provide a systematic means of approaching the problem of design and implementation. In the case of the Kalman filter these rules are as follows:

1. Understand your sensor: The first step is simply to be familiar with the physics of the device: propagation medium, wave-length or emission characteristics, maximum and minimum ranges, etc. The second step is to acquire as much data as possible, in a variety of situations, from the sensor to be employed. A surprising amount can be learnt by simply looking at this data and appreciating what kind of information is likely to be available to the filter. It is quite pointless designing a filter without knowing what information will be available for use.

2. Understand your process: Again familiarity with the kinematics, physics or geometry of the process under consideration is essential: parameters of importance should be identified, constraints and physical limits made precise, key time-constants should be measured, etc. Observation of the process in a variety of operating modes, with additional instrumentation if required, can yield surprising insights into filter design and implementation. It (almost) goes without saying that it is simply not sensible to begin designing a filter without a clear understanding of the problem to be solved.

3. Model your sensor: Having obtained as much information as possible from the device, an accurate kinematic and statistical model of the sensor must be developed. In the Kalman filter algorithm this simply reduces to the construction of an appropriate observation model $\mathbf{H}(k)$ and noise source $\mathbf{v}(k)$. The performance of the filter will be directly dependent on the adequacy of this model. As we have and will continue to stress, there is simply no substitute for developing precise and detailed models of the sensing process.

4. Model your process: The first step is to build as accurate a 'truth model' as possible, describing all aspects of the process to be estimated. This model will undoubtedly be too large and intractable to be employed directly in the filter algorithm but is still an essential step in understanding which states are significant and which marginal in obtaining desirable filter performance. The second step is to reduce this model to those states which have a direct and significant impact on filter performance and to construct an appropriate process model $\mathbf{F}(k)$ and process noise $\mathbf{w}(k)$. This has to be done on a case-by-case basis with respect to an overall 'state-budget'.

5. Filter Coding: The easiest part of the implementation is simply coding the Kalman filter algorithm and models generated in the preceding analysis. Some clearly defined

rules exist for sequencing the various components of the algorithm and computing of state estimates and their associated covariances.

6. Initialization: The recursive formulation of the Kalman filter algorithm means that we must provide some reasonable 'guess' for the initial conditions $\hat{\mathbf{x}}(0 \mid 0)$ and $\mathbf{P}(0 \mid 0)$. Although not critical to long-term performance in the linear algorithm (we shall see that the effects of initial conditions diminish rapidly with time) initialization is still of importance in the filtering of data from real systems.

7. Analysis of Innovation Sequence: The first and most important method of analysing filter performance is using the innovation or residual sequence. Recall that the innovation is simply the difference between the true observation and the predicted observation and that under the assumptions made in deriving the Kalman filter, the innovation sequence will be white and uncorrelated with known covariance $\mathbf{S}(k)$. We will see that testing the innovation sequence for these properties tells us a great deal about how the filter is working and can be used directly to tune filter performance.

8. Analysis of Steady-State Performance: The innovation is a single measure of filter performance which is affected by both the observation and process models. The second step in analysing filter performance is to separate out these two factors by looking simultaneously at both the observation sequence and at the steady-state properties of the filter; the state estimates, state predictions, and their respective covariances.

9. Analysis of Error Conditions: In real systems a final step is required to identify and eliminate erroneous or spurious data from consideration in the filtering process; to ensure that the filter is sufficiently robust for proper use.

The object of this section is to provide as much practical advice as possible on how to design and build a linear Kalman filter. The previous section described the problem of building and validating sensor and process models. Here we will concentrate on the implementation of the algorithm and the tuning of the filter to achieve desired performance. Although a great deal of what follows must be example specific, the techniques and procedures employed can and should be generalized in all applications of the Kalman filter Algorithm.

We begin be describing the overall estimation architecture and the way in which it should be implemented. Taking the example of one-dimensional target motion developed throughout this section we show first how the filter is initialized and then describe how the performance of the filter may be analyzed using the innovation sequence and steady-state conditions.

6.1 Overall Architecture

We shall begin by taking as an example the simple one-dimensional constant-velocity particle motion problem used throughout this section. As we have derived explicit equations for each stage of the estimation cycle using this model, the results of the following implementation may easily be identify. The state model we have employed is given by

$$\begin{bmatrix} x(k) \\ \dot{x}(k) \end{bmatrix} = \begin{bmatrix} 1 & \Delta T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x(k-1) \\ \dot{x}(k-1) \end{bmatrix} + \begin{bmatrix} \Delta T^2/2 \\ \Delta T \end{bmatrix} v(k),$$

where ΔT is the time interval between time steps $k-1$ and k, and where the noise process vk is assumed to be an uncorrelated Gaussian sequence with covariance

$$\mathbf{Q}(k) = \begin{bmatrix} \Delta T^3/3 & \Delta T^2/2 \\ \Delta T^2/2 & \Delta T \end{bmatrix} \sigma_q^2.$$

Observations are assumed to be made at a time step k of the location of the particle according to

$$z_x = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ \dot{x}(k) \end{bmatrix} + w(k),$$

where $w(k)$ is assumed to a white Gaussian sequence with covariance σ_r^2. The input parameters to the filter are thus:

$$\mathbf{F}(k) = \begin{bmatrix} 1 & \Delta T \\ 0 & 1 \end{bmatrix}, \qquad \mathbf{H}(k) = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

with σ_q^2 and σ_r^2 the two noise covariances.

Simulated data for the following analysis was generated using these same models, with $v(k)$ and $w(k)$ computed using a pseudo-random number generator to provide random numbers distributed as a Gaussian with zero mean and covariances σ_q^2 and σ_r^2 respectively. With $\sigma_q = 0.01$m and $\sigma_r = 0.1$m, the true position and observations of position, and true velocity are shown in Figures 5, and 6 respectively. Note that although we are using a 'constant velocity' model, the velocity is not constant ! Indeed, as is clear from the state dynamics equation, the velocity is actually the sum of the disturbances injected into the system and so is a Brownian process.

6.2 Initialization

The first step is to initialize the filter by choosing a starting estimate for the state $\hat{\mathbf{x}}(0 \mid 0)$ and it's variance $\mathbf{P}(0 \mid 0)$. In general, the effect of these initial estimates diminishes with time and they do not affect the steady state performance of a filter. In a simple linear filter, the filter will converge to the same steady-state track regardless of the initial values of state and state covariance. However, as we will see, the choice of a reasonably good

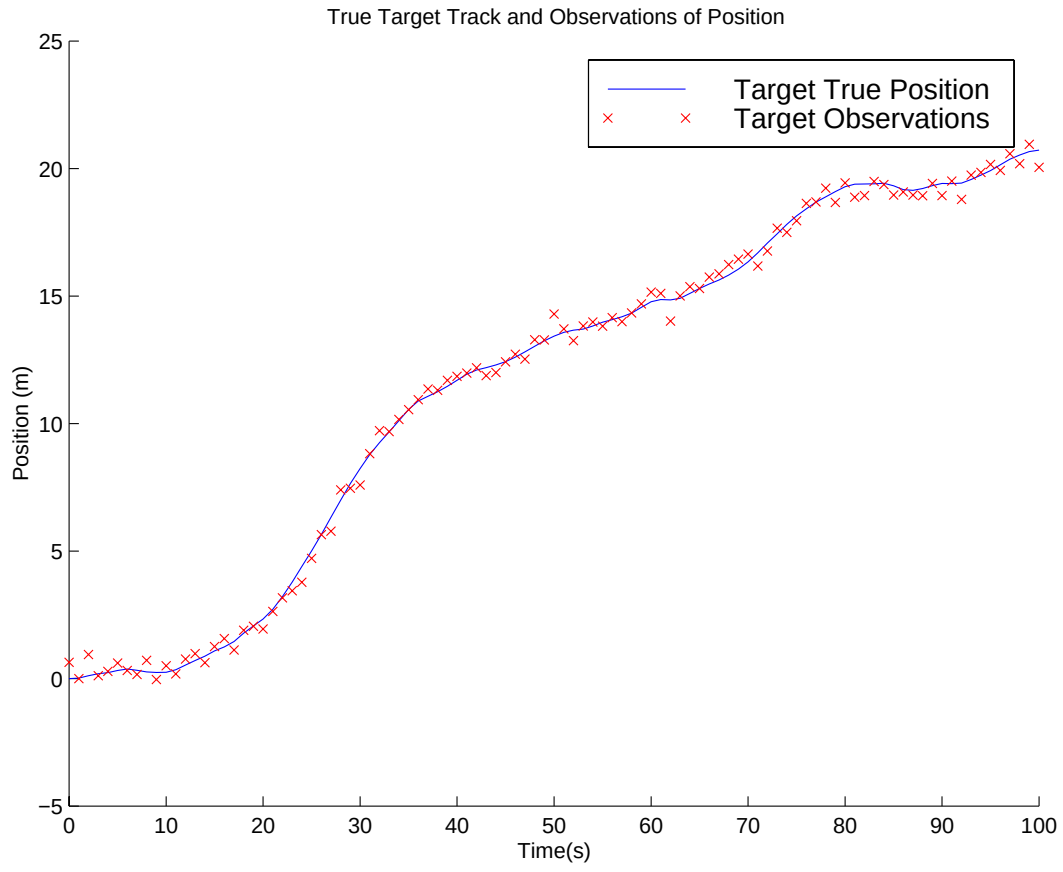


Figure 5: Plot of true target position and observations made of the target; used in the remainder of this example

initial estimate improves convergence and is essential in the convergence of the extended Kalman filter.

We will initialize our filter with estimated standard deviation in process and observation noise set equal to the same values used to generate the track and observation data ($\sigma_q = 0.01$, $\sigma_r = 0.1$). We will see in the next section how these may be determined from experimental observation data.

Figure 7 shows the position track computed by the Kalman filter algorithm on the basis of the observations made in Figure 5 for a number of different initial conditions. It is clear from this that the Kalman filter converges to the same track regardless of what initial conditions are chosen for the state. A simple method of obtaining a good initial estimate of state is to batch-process the first few observations to estimate initial position and velocity values provide initial values;

$$\hat{x}(1 \mid 1) = z(1), \qquad \hat{\dot{x}}(1 \mid 1) = \frac{z(2) - z(1)}{\Delta T}.$$

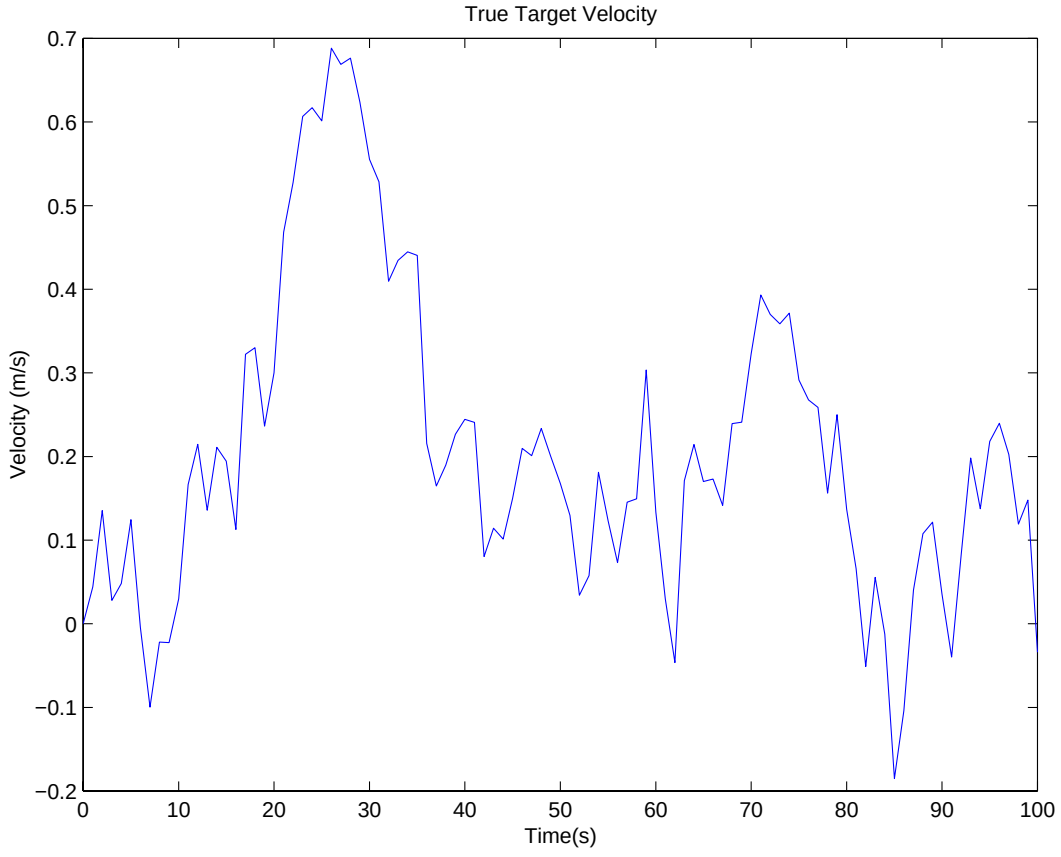


Figure 6: Plot of true velocity of target

This can be improved by using a simple least-squares fit if required. This will be considered in more detail when it is required for the extended Kalman filter.

The simplest method of initialize the state covariance is by choosing a constant α and setting $\mathbf{P}(0 \mid 0) = \alpha^2 \mathbf{Q}(k)$, (typically $\alpha = 10$ is a reasonable starting estimate). Again however, the effect of this initial estimate diminishes with time. Figure 8 shows that the magnitude of the estimated position error ($\sqrt{P_{xx}(k|k)}$) converges in time to the same steady state value for a variety of values of α. As a rule-of-thumb, the initial covariance information should be large enough to encompass any likely uncertainty in the initial guess for $\hat{\mathbf{x}}(0 \mid 0)$. One means of estimating this uncertainty is to use the mean-squared error computed by a least-squared fit to the first few observations.

6.3 Steady-State Performance

The steady-state performance of the filter is uniquely determined by the values of process and observation noise covariances, $\mathbf{Q}(k)$ and $\mathbf{R}(k)$ respectively which are chosen as filter inputs. Figure 9 shows a complete state track for the filter initialized with estimated

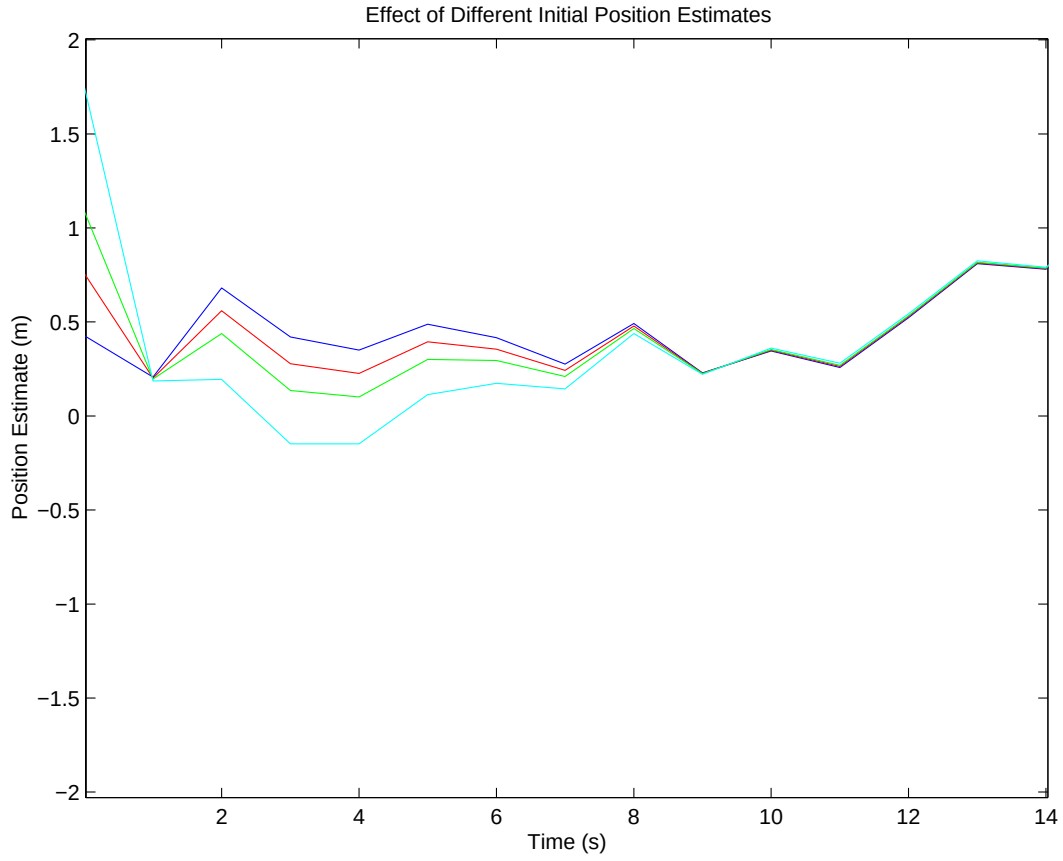


Figure 7: Estimated Position With Several Different Initial Conditions.

process and observation noise standard-deviation equal to the actual values employed to generate the true track and observations ($\sigma_q = 0.01$, $\sigma_r = 0.1$). Figure 9(a) shows the complete track history. We have made a point of *not* showing the true value of the position $x(k)$ as in any real situation this will not be available to us to study filter performance. It is difficult to see from this graph how the filter works because the changes in position of the particle due to velocity and acceleration changes dominate the effects of estimation uncertainty. This is typical of this type of kinematic model; small changes in velocity and acceleration at the beginning of an estimation cycle cause large changes in position over time. Figure 9(b) shows the filter operation in close-up when it has reached steady state. The key thing to note is that the updated estimate of position $\hat{x}(k|k)$ always lies between the predicted position $\hat{x}(k|k-1)$ and the observed value of position $z(k)$. This is to be expected from our understanding of the Kalman filter as a weighted average; the updated estimate is simply a weighted sum of observation and prediction. Indeed, we know from

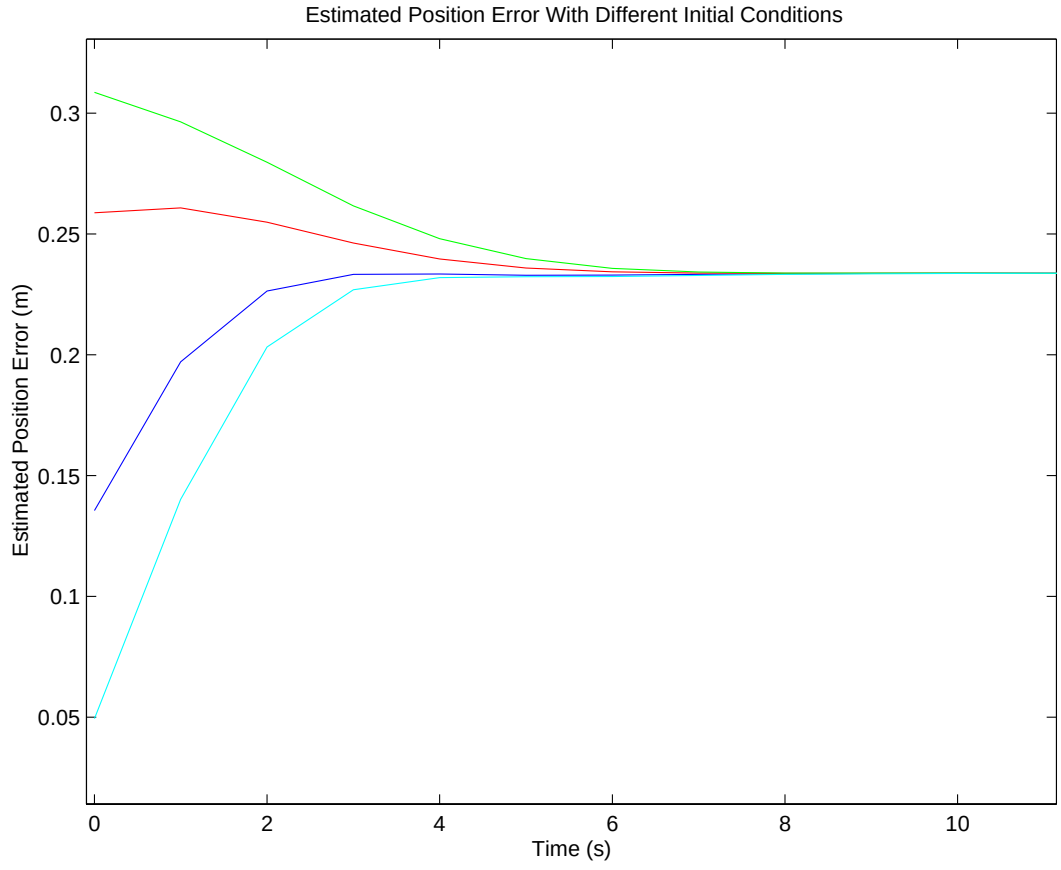


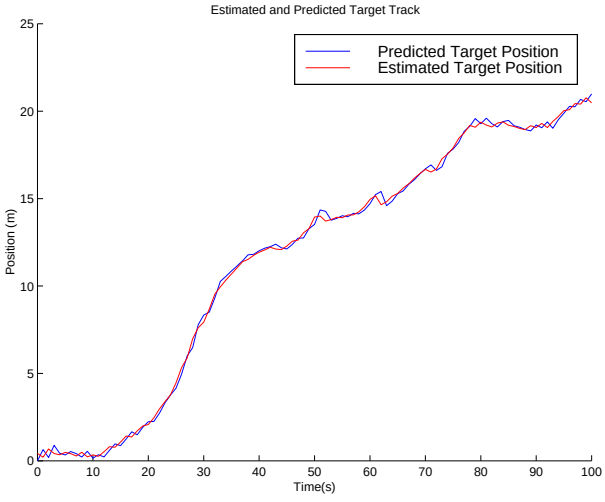
Figure 8: Estimated Position Standard Deviation With Several Different Values of α.

Equation 164 that the weights are given by

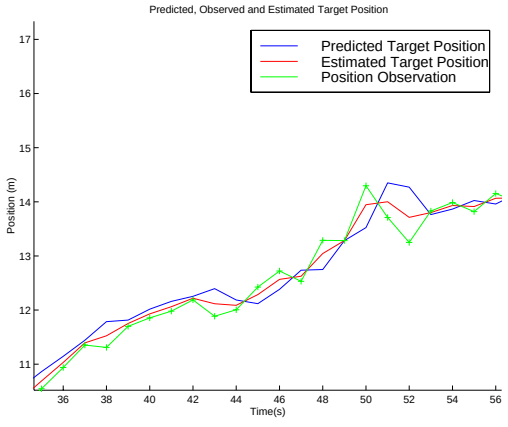
$$\hat{x}(k|k) = \frac{\sigma_r^2}{P_{xx}(k|k-1) + \sigma_r^2}\hat{x}(k|k-1) + \frac{P_{xx}(k|k-1)}{P_{xx}(k|k-1) + \sigma_r^2}z(k).$$

This is shown graphically in Figure 9(b). It is also clear from this figure that the innovation is just the distance between the prediction and the observation, and that the subsequent predicted position is computed from the updated position estimate and the updated estimate of velocity. It should be recalled that for a synchronous linear Kalman filter the gains reach a steady-state value so that the weights associated with the observations and predictions during an update also tend to constant values. This means that when a steady-state has been achieved, the estimate will lie between observation and prediction in constant proportion.

The velocity predictions $\hat{\dot{x}}(k|k-1)$ and velocity estimates $\hat{\dot{x}}(k|k)$ are shown in Figure 10. No direct observations are made of the velocity state and so estimates are produced through the cross-correlation between position and velocity, according to our interpretation of the Kalman filter as an observer in Equation 175. This is made explicit in



(*a*)



(*b*)

Figure 9: Position Track: (a) Complete Track; (b) Detail of track showing update mechanism.

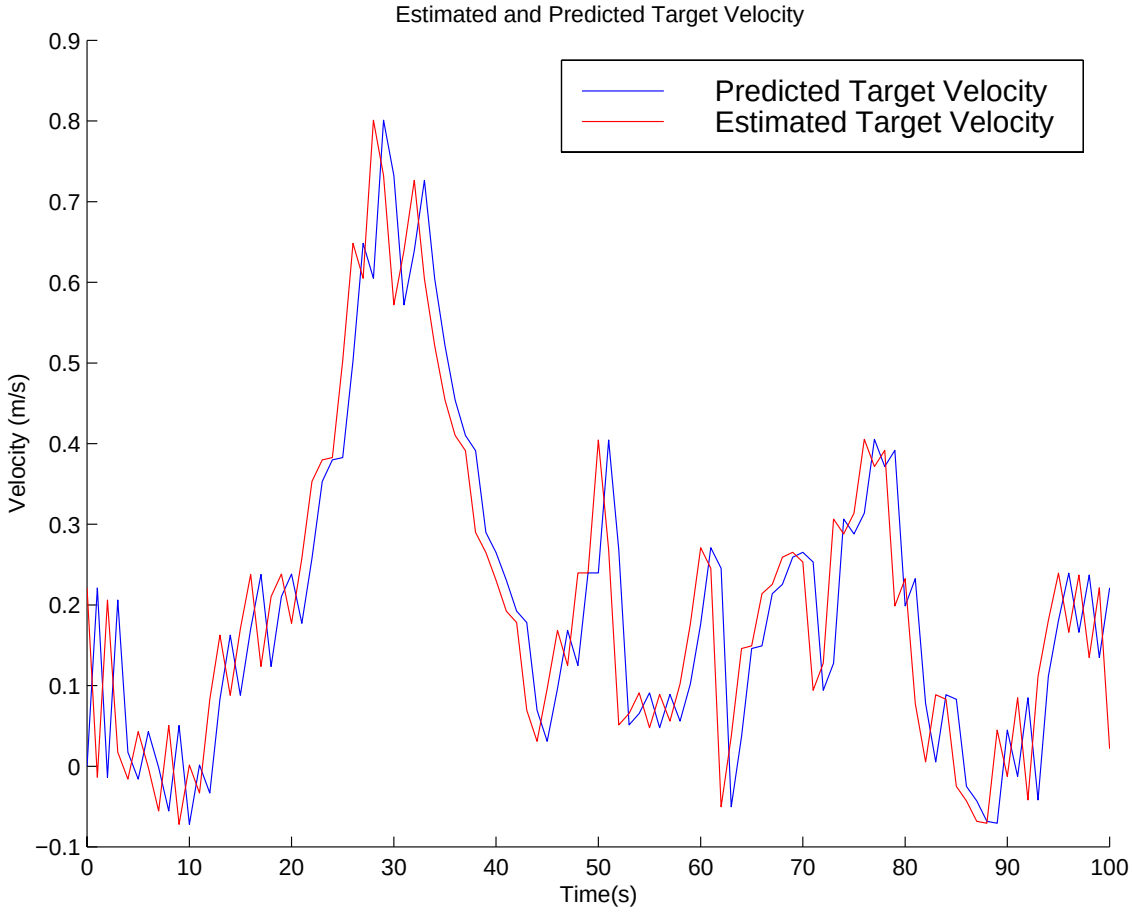


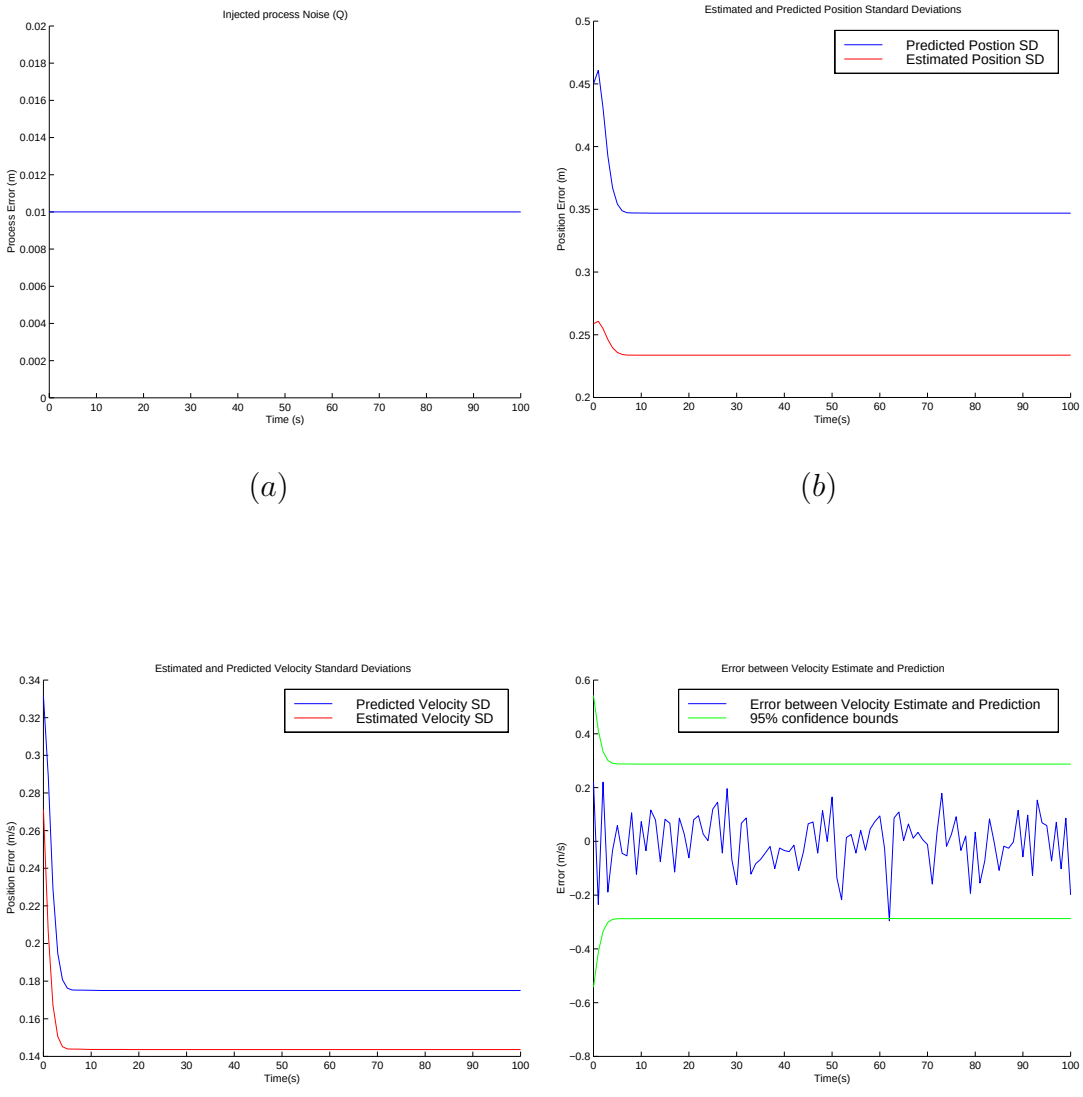
Figure 10: Estimated track velocity.

Example 15 which shows that the updated velocity estimate is proportional to both the cross-correlation between position and velocity and the position innovation as

$$\hat{\dot{x}}(k|k) = \hat{\dot{x}}(k|k-1) + \frac{P_{x\dot{x}}(k|k-1)}{P_{xx}(k|k-1) + \sigma_r^2} \left(z(k) - \hat{x}(k|k-1)\right).$$

Figure 10 shows the complete velocity-track history. The apparent lag between prediction and estimate shown in the Figure is due to the fact that the particle is actually accelerating at this time while the model assumes that the velocity is constant. As with the position update, the fact that the gain matrix is constant in steady-state means that the same fraction of the innovation will be used to update velocity at every time-step. Note that the velocity estimates are smoother than the true values, this is due to the smoothing properties of the estimator (Equation 42).

Figure 11 shows the complete estimation and prediction error track for position and velocity standard deviations. Figure 11(a) shows the relative level of input process noise,



(a) (b)

(c) (d)

Figure 11: Track of Estimated Standard Deviations: (a) Process Noise Standard Deviation; (b) Position Prediction and Position Estimate Standard Deviation; (c) Velocity Prediction and Velocity Estimate Standard Deviations; (d) Position-Velocity Cross-Correlations.

Figure 11(b) shows the standard deviation in position prediction and position estimate, Figure 11(c) shows the cross-correlation between position and velocity estimates, and Figure 11(d) shows the standard deviation in velocity prediction and velocity estimate. The most obvious thing to note about these tracks is that the error reaches a steady state in only a very small number of iterations. A second point to note is that the estimate error is always less than or equal to the prediction error.

6.4 Filter Consistency

The most important aspect of filter implementation is to decide if the estimates produced are a 'reasonably good' reflection of the value of the true state. An estimator is said to be *consistent* if it is unbiased;

$$\hat{\mathbf{x}}(k \mid k) = \mathrm{E}[\mathbf{x}(k) \mid \mathbf{Z}^k] \ ,$$

and its state estimation errors satisfy

$$\mathrm{E}[(\mathbf{x}(k) - \hat{\mathbf{x}}(k \mid k)) (\mathbf{x}(k) - \hat{\mathbf{x}}(k \mid k))^T \mid \mathbf{Z}^k] \ = \mathbf{P}(k \mid k).$$

In principle, filter consistency may be checked in the following way. We first construct the normalized state estimation error squared

$$p(k) = (\mathbf{x}(k) - \hat{\mathbf{x}}(k \mid k))^T \mathbf{P}^{-1}(k \mid k) (\mathbf{x}(k) - \hat{\mathbf{x}}(k \mid k)) \, .$$

Under the hypothesis that the filter is consistent, $p(k)$ is a χ^2 distribution in $n = dim(\mathbf{x}(k))$ degrees of freedom and

$$\mathrm{E}[p(k)] \ = n. \tag{183}$$

The test is whether Equation 183 can be accepted. In any real situation however, we will never have access to the true values of the parameters to be estimated (except through simulation of an 'exact' truth model). This means that measures of performance, such as state-error, which require us to have knowledge of the true state are not practically very useful. Typically, the only information we have about a process is the observation sequence, and so the only practical way of studying filter performance is to compare this observation sequence with the observation sequence predicted by the filter. The innovation, or difference between actual and predicted observation as defined by Equation 150, is the most direct measure of filter performance. We know that if the assumptions employed in the filter are correct, then the innovation sequence $\nu(k)$ will be zero-mean and white with covariance $\mathbf{S}(k)$. To check that a filter is indeed consistent, we need to observe that the innovations have this property.

Figure 12 shows the innovation sequence for this example together with corresponding '$\pm\sigma$ and $\pm 2\sigma$' bounds. A first look at this plot would seem to indicate that indeed the innovation sequence has zero-mean and is white with approximately 95% of the innovations fall within a 2-σ gate. Often it is sufficient to simply look at the innovation

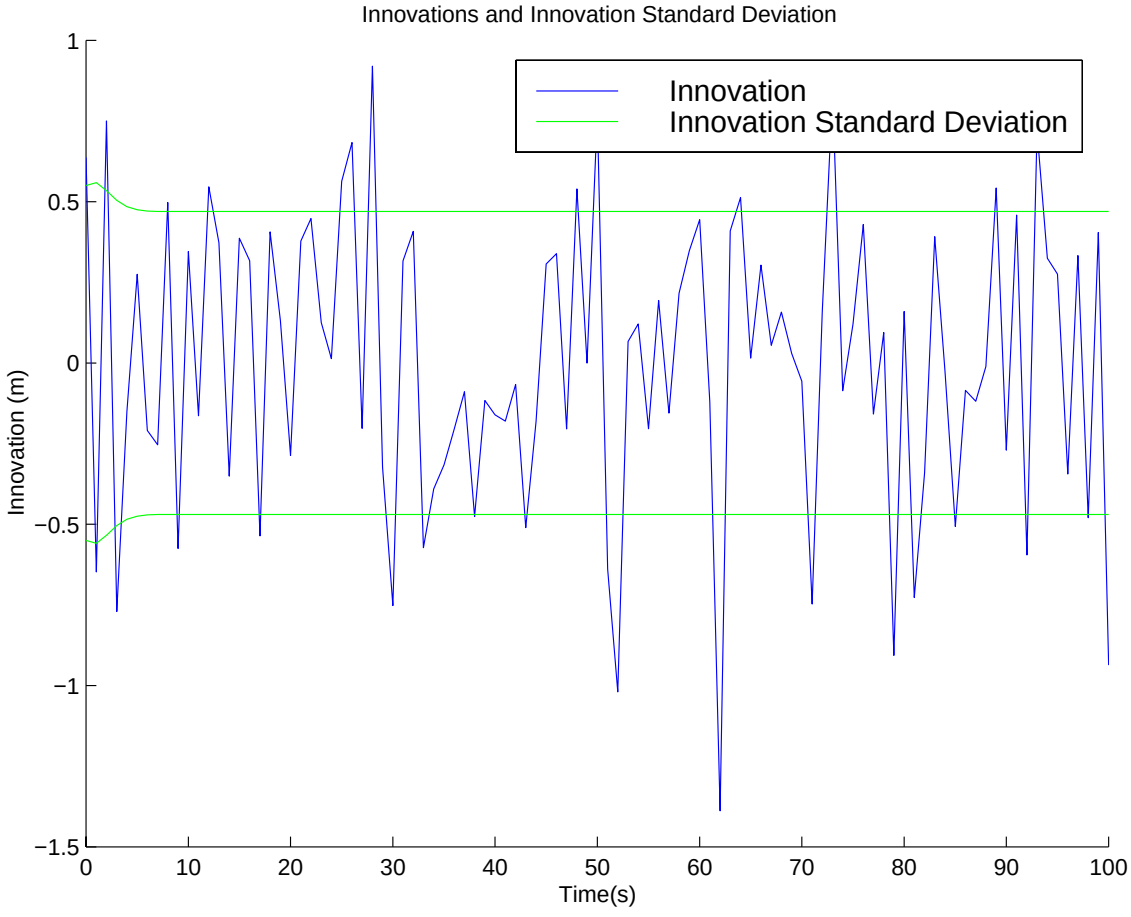


Figure 12: Innovations and Innovation Standard Deviations

sequence and check that indeed it appears reasonable. However in some cases, particularly when constructing a model for the first time, it is sensible to use a more rigorous test of consistency.

To be precise about the consistency of a filter we must test that the innovation sequence is both unbiased and white. To test for unbiasedness, we first compute the sequence of normalized innovations squared according to

$$q_i(k) = \nu_i^T(k)\mathbf{S}_i^{-1}(k)\nu_i(k)$$

as shown in Figure 13. We know that if the filter assumptions are indeed correct then the normalized innovations squared are each χ^2 in m degrees of freedom and so

$$\mathrm{E}[q(k)] \ = m. \tag{184}$$

Equation 184 provides a test for unbiasedness. To approximate this mean we should, in principle, make N independent runs of the filter, providing N independent samples of the

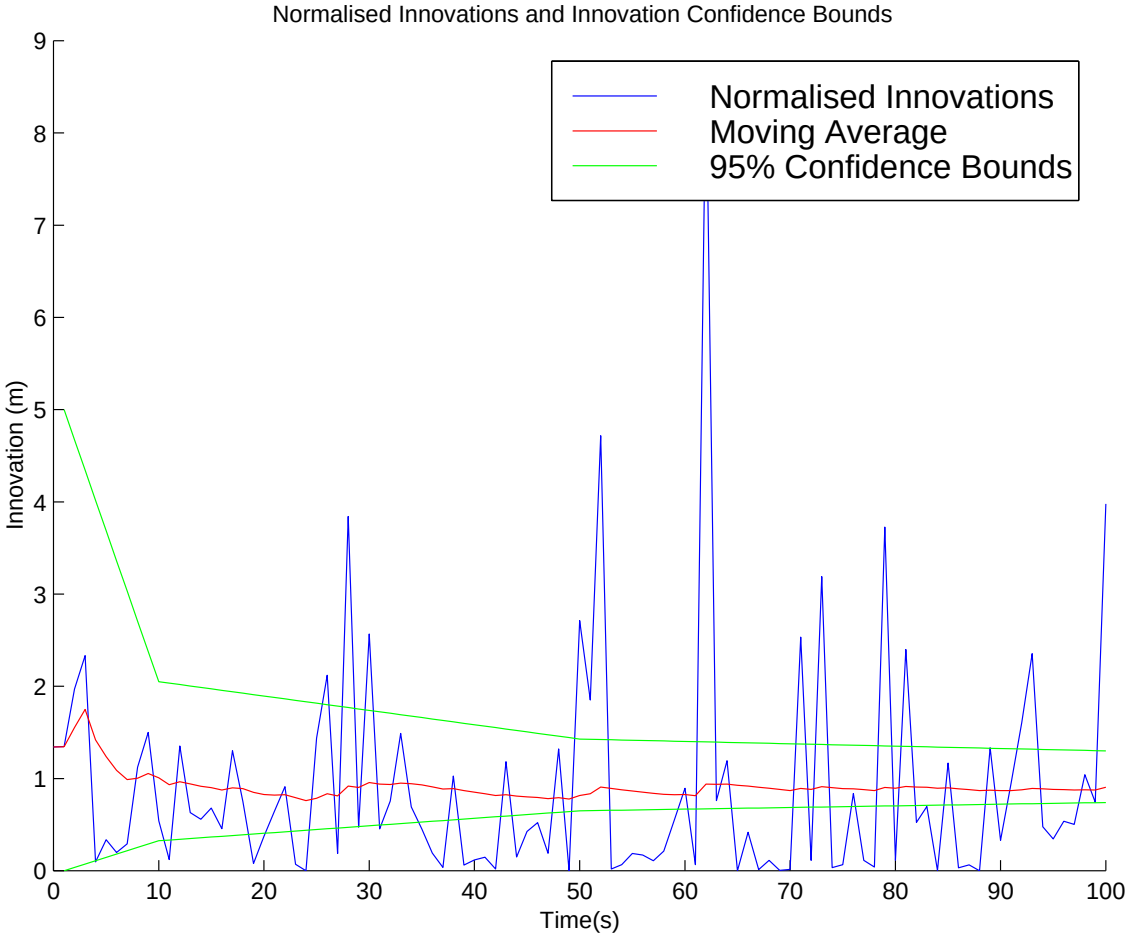


Figure 13: Normalized Innovation, moving average, and 95% Confidence Bounds.

sequence $q^i(k), i = 1, \cdots, N$, and compute the sample mean as a test statistic;

$$\overline{q}(k) = \frac{1}{N} \sum_{i=1}^{N} q^i(k).$$

We then know that $N\overline{q}(k)$ is χ^2 in Nm degrees of freedom and so we can provide, from χ^2 tables, a confidence interval to test the hypothesis in Equation 184. For the equivalent hypothesis test employing normalized state errors, we would indeed have to acquire a set of independent samples and construct the sample mean. However, because the innovation sequence is assumed to be uncorrelated and white, we may in this case exploit the fact that the sequence is ergodic and so the sample mean can be approximated by a time average from a suitably long time sequence. Thus the test in Equation 184 may be implemented

by computing the average

$$\overline{q} = \frac{1}{N}\sum_{i=1}^{N} q(i), \tag{185}$$

from a single sample run as the test statistic and testing that $N\overline{q}$ is a χ^2 random variable in Nm degrees of freedom. Figure 13 shows the value of the average $\overline{q}$ for 100 samples. Clearly, this average converges to 1.0. We wish now to construct a confidence interval or bounding set $[b_1, b_2]$ between which the average $\overline{q}$ should lie to test if we should accept the hypothesis H_0 that $N\overline{q}$ is distributed as χ^2_{100} with a specified probability $1-\alpha$;

$$P(\overline{q} \in [b_1, b_2] \mid H_0) = 1 - \alpha. \tag{186}$$

With $1 - \alpha = 0.95$ we have from tables

$$\begin{aligned} [b_1, b_2] &= [\tfrac{\chi^2_{100}(0.025)}{100}, \tfrac{\chi^2_{100}(0.975)}{100}] \\ &= [\tfrac{74}{100}, \tfrac{130}{100}] \\ &= [0.74, 1.3]. \end{aligned} \tag{187}$$

Clearly $\overline{q} \in [0.74, 1.30]$ and so the hypothesis is accepted.

It remains to test the whiteness of the innovation sequence; that is to test that indeed

$$\mathrm{E}[\nu^T(i)\nu(j)] = \mathbf{S}(i)\delta_{ij}. \tag{188}$$

To do this we construct a test statistic for Equation 188 and check that except when $i = j$, the statistic is zero within a given tolerance. Again, exploiting the fact that the innovation is both stationary and ergodic if it is indeed white, a test statistic for Equation 188 is the time-averaged (biased) autocorrelation

$$\mathbf{r}(\tau) = \frac{1}{N-\tau}\sum_{i=0}^{N-\tau-1} \nu^T(i)\nu(i+\tau)$$

It is usual to normalize so that $\mathbf{r}(0) = 1$. The resulting normalized autocorrelation for our current example is shown in Figure 14. As expected, it is highly peaked at $\tau = 0$ with a smaller random component for $\tau > 10$. We need to check that this random component is, within certain acceptable bounds, zero. For large enough sample size the test statistic is approximately normally distributed with standard deviation $\frac{1}{\sqrt{N}}$, thus 95% confidence bounds in the normalized autocorrelation may be approximated by $\pm\frac{2}{\sqrt{N}}$ as shown in Figure 14. The 'random' component of the autocorrelation clearly falls within these bounds on at least 95% of samples and so we may accept the hypothesis that the innovation sequence is white.

There is an immense literature on statistical testing, particularly associated with tests for zero-means and whiteness properties. For most of the problems we are interested

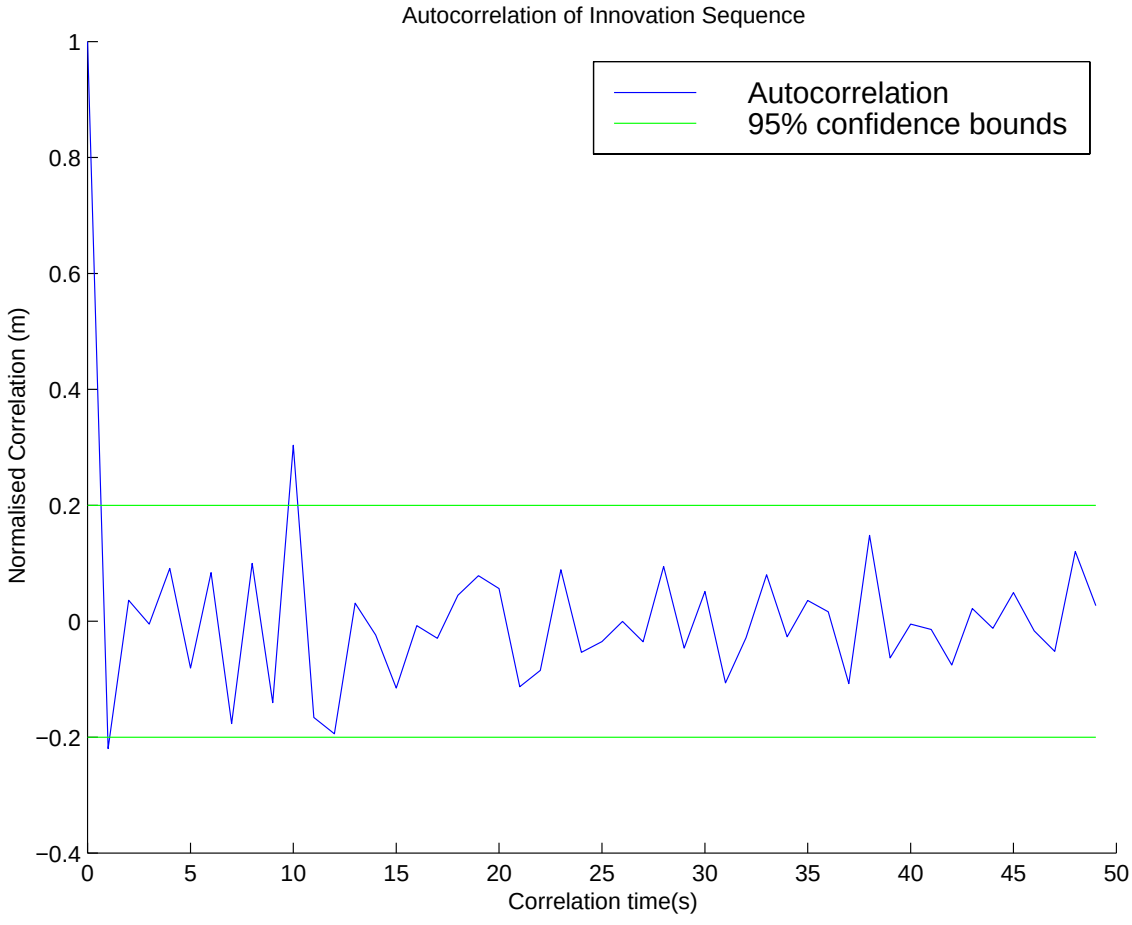


Figure 14: Time-Averaged (Biased) Normalized Autocorrelation of the Innovation Sequence.

in it is very often sufficient to simply look directly at the innovation sequence itself to check that the filter is performing as desired. However the innovation sequence can be misleading, sometimes hiding correlations which are consequences of significant modeling errors. When designing a filter from start, it may also be advisable to employ the simple tests described above to check that the innovation sequence is indeed zero mean and white. These tests should be treated with caution. In particular too small a sample size will not give meaningful results. More sophisticated tests can be applied to investigate statistical properties of the filter but these are beyond the scope of this course.

6.5 Validating Sensor and Process Models

Having decided on a sensor and process model together with values for the associated process and observation noise variances, it is essential that these be validated to ensure

that the filter is consistent and operates in the manner expected. In the previous section, we considered only the output of a filter which exactly modeled the true observation and process and which had process and observation noise variances exactly matching the true parameters. In this section we consider what happens when the models and noise parameters are incorrectly specified, how this can be detected, and how the filter can subsequently be 'tuned' to generate 'correct' models and so provide the required performance.

We will again use the example of a constant-velocity process model. Every different filter will have a different type of behaviour and this example can only show some general principles which should be applied to the problem of model validation. As before, a considerable amount of engineering judgement must be employed to ensure that sensible decisions about the appropriate model and appropriate noise levels are made at an early stage in the design. We emphasis again that the only way to make sensible decisions about process and observation models is to acquire a substantial amount of real data and to undertake a careful and exact modeling of both the measurement process and the state description: Remember that the filter is only as good as the models it uses. A second point worth repeating is that the process of tuning and validating the models and filter takes an order of magnitude more time than writing the original filter.

6.5.1 Validation Tests

In any real situation, the true value of the state is unknown, and the only information available to the designer is the actual measurements made of the state which are themselves used in the filter. As we have seen, this means that the innovation sequence is the principle means of analysing and validating filter performance. In the following, we will use four methods to help in the validation and tuning procedure:

- **Magnitude of the innovation:** The simplest indication of filter correctness is to look directly at the innovation sequence to ensure that approximately 95% of innovations fall within a 2-σ gate. If too many fall outside of this gate, this indicates that the noise levels set in the filter are generally lower than the true noise levels. Conversely, if the innovations are all well within the 2-*sigma* gate, this indicates that the noise levels have been set too high. Additionally, if non-random periodic fluctuations or non-random drifts are observed, this also indicates that something is wrong.

- **χ^2 test of the normalized innovation squared:** If the overall noise level is suspect, a χ^2 test of the normalized innovation squared can be performed using Equations 185 and 186. If the average normalized innovation falls outside the bounding set defined by this test, then the innovation can not be assumed to be Gaussian with zero mean and computed variance $\mathbf{S}(k)$. Generally, if the average normalized innovation falls below these bounds, then the assumed noise levels in the filter are too

high, and conversely if the average normalized innovation falls above these bounds, then the assumed noise levels are too low.

- **The autocorrelation of the innovation sequence:** If the innovation exhibits non-random periodic variations or non-random drift, it is likely that the innovation sequence is correlated (non-white). To test this, the autocorrelation of the innovation sequence needs to be computed. As described in the previous section, if the autocorrelation consistently exceeds its variance bounds away from zero, then the whiteness test will have failed. The autocorrelation generally fails the whiteness test for two reasons; either the process has significant higher-order unmodeled dynamics, or the ratio of observation noise to process noise is too high. In addition, the shape of the computed autocorrelation can provide useful information about why the innovation sequence is correlated.

- **The error between state estimate and prediction:** One essential difficulty in using the innovation sequence alone as a measure of filter performance is that it is often difficult to distinguish the relative contribution of process errors and measurement errors. The error between state estimate and prediction provides a broad means of isolating only the process error. In particular, the error in the highest order states should be approximately uncorrelated and bounded by their associated covariance.

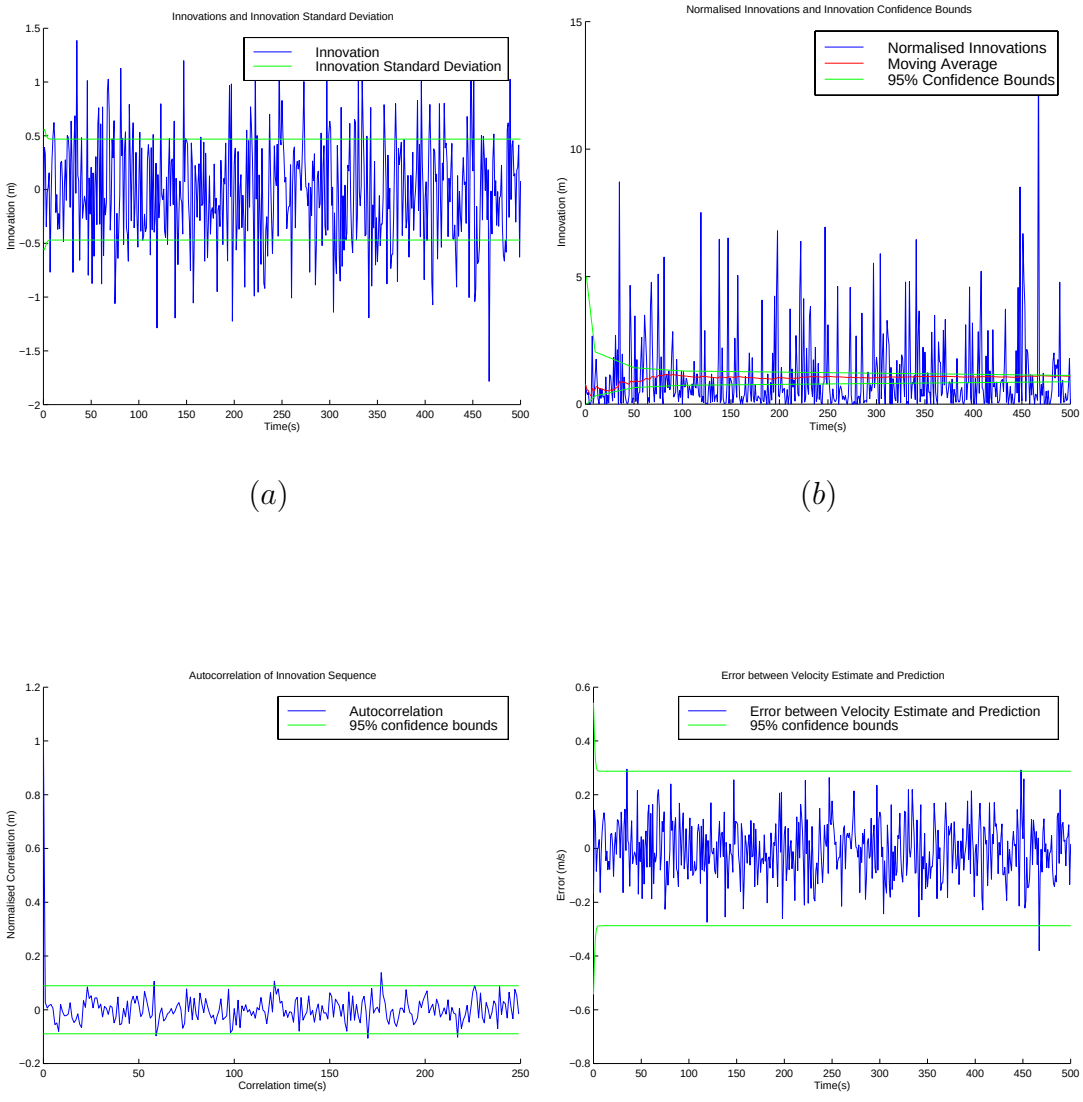
We shall see that errors in process and observation do produce different types of innovation behaviour. In the three tests described, these behaviours can be used to isolate the identity and cause of error. It should be noted however that the only way of completely separating the effects of process and observation error is through knowledge of the true state.

We shall now consider two types of error. The first is an error in the specification of process and observation noise, the second is an error in the specification of process model. Errors in the observation model are not considered here as the structural model itself is either quite simple (and does not usually involve a kinematic or dynamic model), or involves intrinsic correlations which must be modeled explicitly. This latter case is dealt with in the next section.

6.5.2 Noise Model Errors

We consider first errors in the specification of process and observation noise. This is the most common problem in designing and validating a Kalman filter algorithm. We deal with four cases; under-estimate of process noise, under-estimate of observation noise, over-estimate of process noise, and over-estimate of observation noise. From these we draw some general principles for detecting and correcting errors in the specification of estimated noise levels.

For reference, Figure 15 shows the results for an exactly matched constant-velocity filter with $\hat{\sigma}_q = \sigma_q = 0.01$ and $\hat{\sigma}_r = \sigma_r = 0.1$. Figure 15(a) shows the innovation sequence



(a) (b)

(c) (d)

Figure 15: Exactly Matched Constant-Velocity Filter with $\hat{\sigma}_q = \sigma_q = 0.01$ and $\hat{\sigma}_r = \sigma_r = 0.1$. (a) Innovation Sequence and Estimated Innovation Standard Deviation; (b) Normalized Innovations Squared, Moving Average and 95% Confidence Bounds; (c) Normalized, Biased Autocorrelation of the Innovation Sequence and 95% Confidence Bounds; (d) Error between Velocity Estimate and Velocity Prediction, With 95% Confidence Bounds.

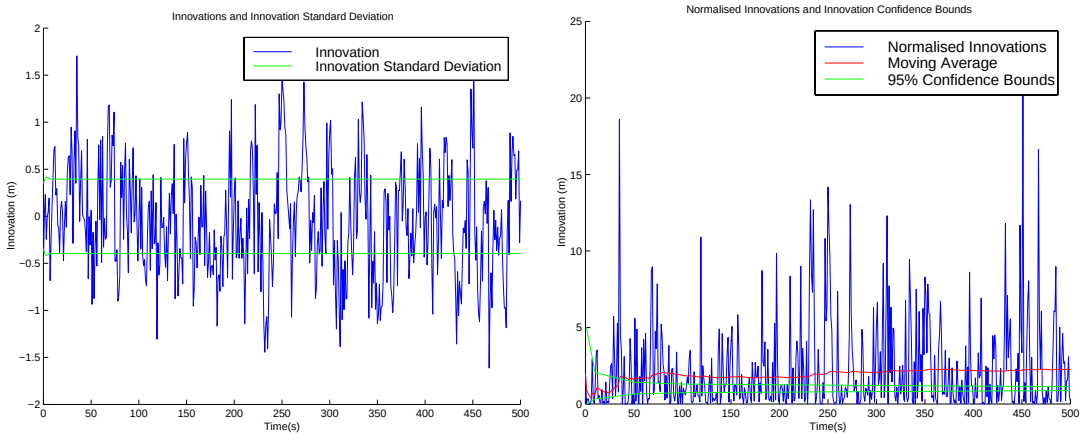
together with estimated innovation standard deviation. Figure 15(b) shows the normalized innovations squared, together with its moving average and 95% confidence bounds. Figure 15(c) shows the normalized, biased, autocorrelation of the innovation sequence, together with 95% confidence bounds. Figure 15(d) shows the error between velocity estimate and velocity prediction, together with estimated 95% confidence bounds[5].

Figure 16 shows the results of underestimating the process noise level by a factor of 10 ($\hat{\sigma}_q = 0.001$). The innovations shown in Figure 16(a) appear to remain within the bounds of the computed standard deviation, but show substantial non-random periodic oscillations. The normalized innovations squared shown in Figure 16(b) are larger than would normally be expected, and the sample mean falls above the computed 95% confidence bounds for the associated χ^2 test. This tells us that the overall (combined observation and process) noise level is too low. Figure 16(c) shows the time-averaged biased autocorrelation of the innovation sequence. An exponential, periodic, time-correlation with relatively short time-constant is evident. This indicates that the position estimate is successively over-estimated then under-estimated with frequency inversely proportional to the correlation time-constant. This itself is because too much weight is placed on past observations (the position prediction) rather than the current observations, resulting in the position track appearing to have too much 'inertia'. Figure 16(d) shows the error between estimated and predicted velocity together with its associated 95% confidence bounds. Although this error is within bounds, it is not white as it shows marked periodic correlation.

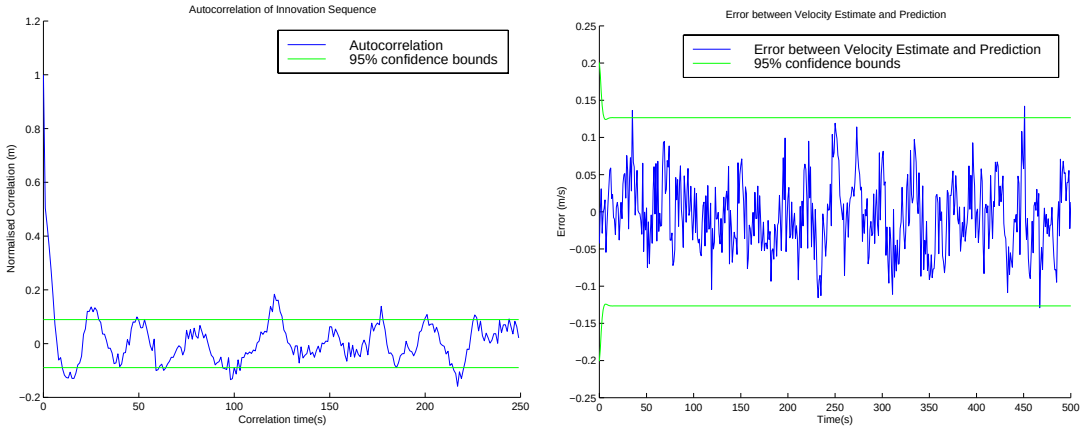
Figure 17 shows the results of underestimating the observation noise level by a factor of 10 ($\hat{\sigma}_r = 0.01$). The innovations shown in Figure 17(a) clearly exceed the bounds of the computed standard deviation, but do not show any non-random periodic oscillations. The normalized innovations squared shown in Figure 17(b) are substantially larger than would normally be expected, and the sample mean falls well above the computed 95% confidence bounds. This tells us that the overall noise level is much too low. Figure 17(c) shows the time-averaged biased autocorrelation of the innovation sequence. This confirms that there are no substantial time correlations in the innovation sequence. Figure 17(d) shows the error between estimated and predicted velocity together with its associated 95% confidence bounds. Although, in this case, the error looks uncorrelated in time, it is clearly not bounded by the confidence interval.

Figure 18 shows the results of over-estimating the process noise level by a factor of 10 ($\hat{\sigma}_q = 0.1$). The innovations shown in Figure 18(a) are well within the bounds of the computed standard deviation, and show no obvious signs of time-correlated behaviour. The normalized innovations squared shown in Figure 18(b) are smaller than would normally

[5]Note that the difference between position estimate and predicted is proportional to velocity which itself is the result of summing (integrating) the noise source (by definition in the constant velocity model). The position error is thus a first-order Gauss-Markov process, and is not suitable for simple analysis. In general, the error of interest is that which is assumed driven by white noise alone. In effect it is this particular assumption which is being verified.

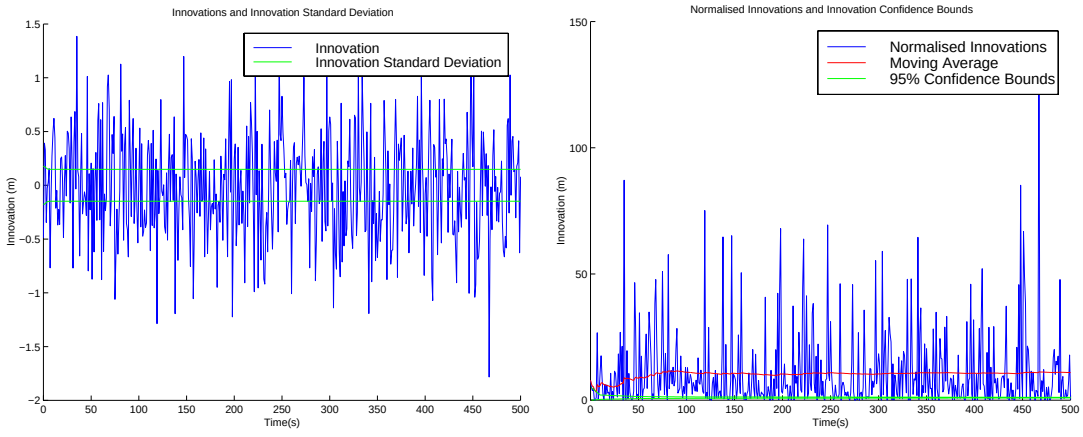


(a) (b)

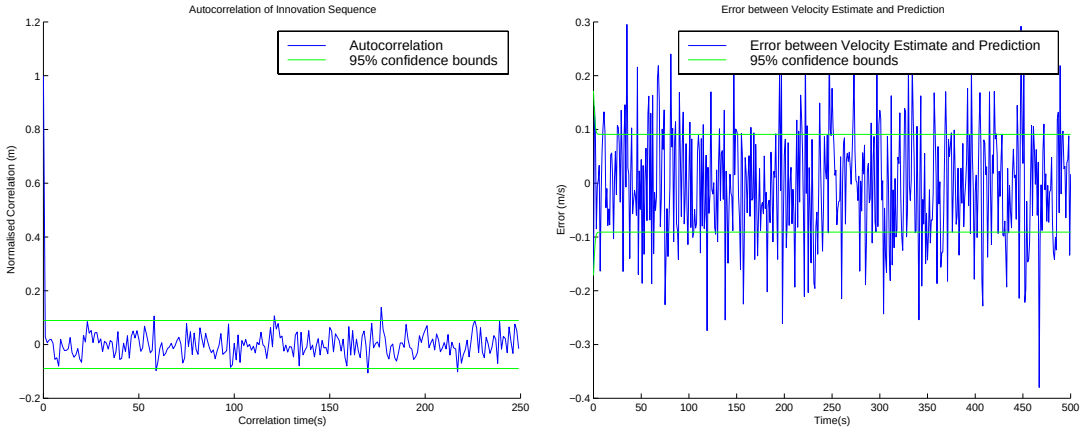


(c) (d)

Figure 16: The Effect of Underestimating Process Noise Levels: $\hat{\sigma}_q = \sigma_q/10 = 0.001$, $\hat{\sigma}_r = \sigma_r = 0.1$. (a) Innovation Sequence and Estimated Innovation Standard Deviation; (b) Normalized Innovations Squared, Moving Average and 95% Confidence Bounds; (c) Normalized, Biased Autocorrelation of the Innovation Sequence and 95% Confidence Bounds; (d) Error between Velocity Estimate and Velocity Prediction, With 95% Confidence Bounds.

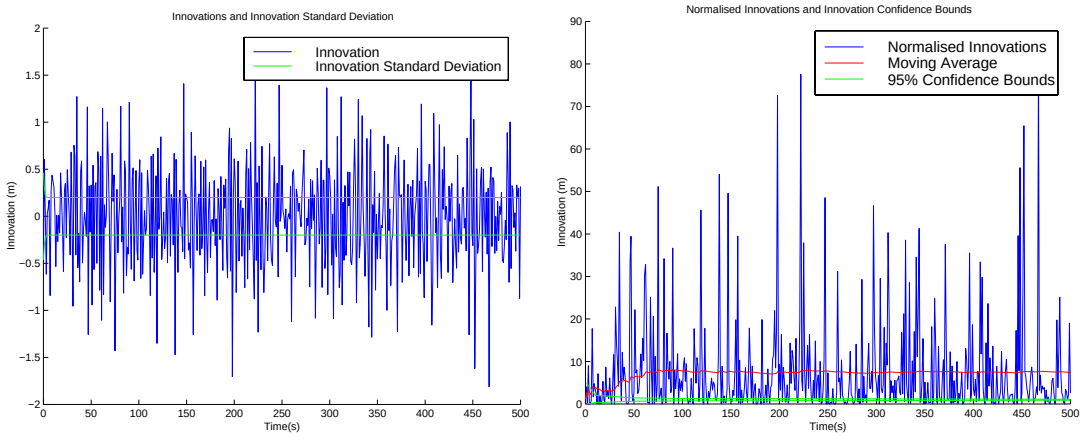


(a) (b)

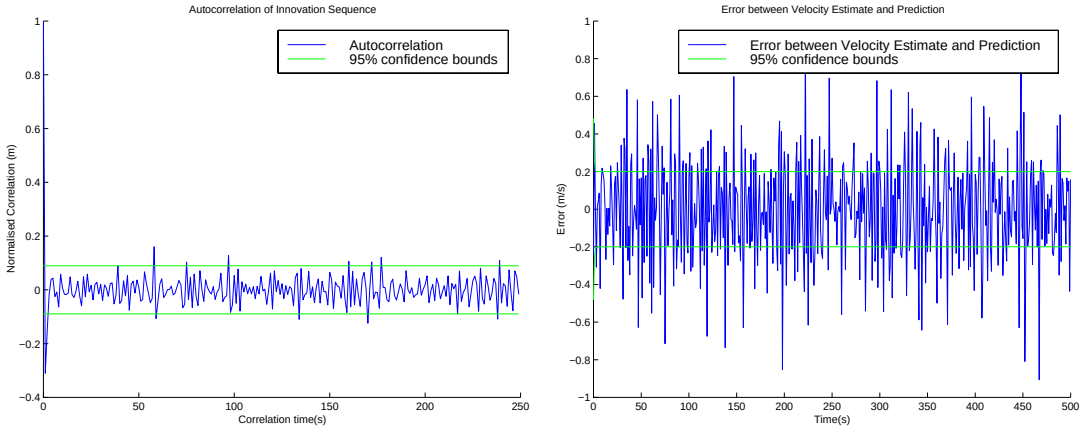


(c) (d)

Figure 17: The Effect of Underestimating Observation Noise Levels, $\hat{\sigma}_q = \sigma_q = 0.01$ and $\hat{\sigma}_r = \sigma_r/10 = 0.01$. (a) Innovation Sequence and Estimated Innovation Standard Deviation; (b) Normalized Innovations Squared, Moving Average and 95% Confidence Bounds; (c) Normalized, Biased Autocorrelation of the Innovation Sequence and 95% Confidence Bounds; (d) Error between Velocity Estimate and Velocity Prediction, With 95% Confidence Bounds.



(a) (b)



(c) (d)

Figure 18: The Effect of Over-Estimating Process Noise Levels, $\hat{\sigma}_q = 10\sigma_q = 0.1$ and $\hat{\sigma}_r = \sigma_r = 0.1$. (a) Innovation Sequence and Estimated Innovation Standard Deviation; (b) Normalized Innovations Squared, Moving Average and 95% Confidence Bounds; (c) Normalized, Biased Autocorrelation of the Innovation Sequence and 95% Confidence Bounds; (d) Error between Velocity Estimate and Velocity Prediction, With 95% Confidence Bounds.

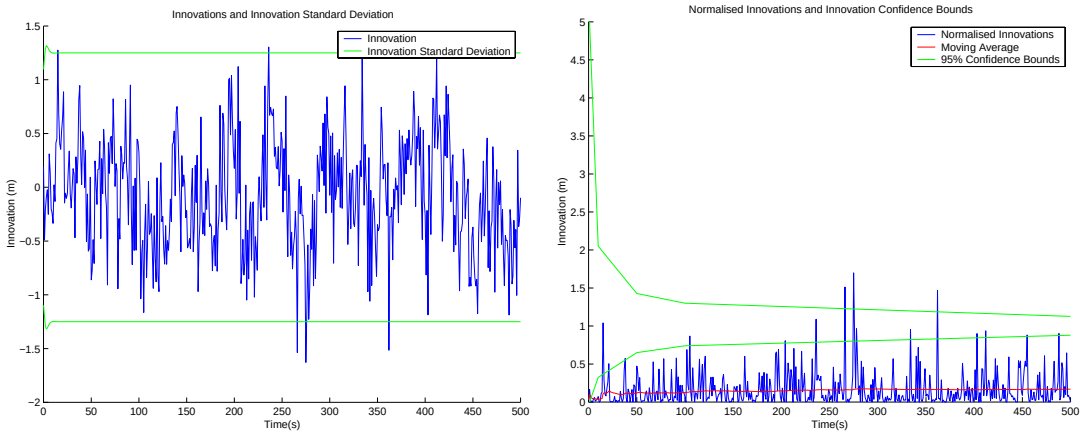
be expected, and the sample mean falls below the computed 95% confidence bounds. This tells us that the overall noise level is too high. Figure 18(c) shows the time-averaged biased autocorrelation of the innovation sequence. It confirms that there are no significant time-correlations. Figure 18(d) shows the error between estimated and predicted velocity together with its associated 95% confidence bounds. The error is lower than the computed bounds would lead us to expect, but is otherwise uncorrelated.

Figure 19 shows the results of over-estimating the observation noise level by a factor of 10 ($\hat{\sigma}_r = 1.0$). The innovations shown in Figure 19(a) are all well below the computed standard deviation, and also show substantial non-random periodic oscillations. The normalized innovations squared shown in Figure 19(b) are much smaller than would normally be expected, and the sample mean falls well below the computed 95% confidence bounds. This tells us that the overall noise level is much too high. Figure 19(c) shows the time-averaged biased autocorrelation of the innovation sequence. An exponential time-correlation with relatively long time-constant is evident. As before, this indicates that the position estimate is successively over-estimated then under-estimated. This is because the *relative* weight placed on past observations (the position prediction) is too large with respect to the weight placed on the current observations. Figure 19(d) shows the error between estimated and predicted velocity together with its associated 95% confidence bounds. This is clearly well within bounds but shows substantial time correlations.

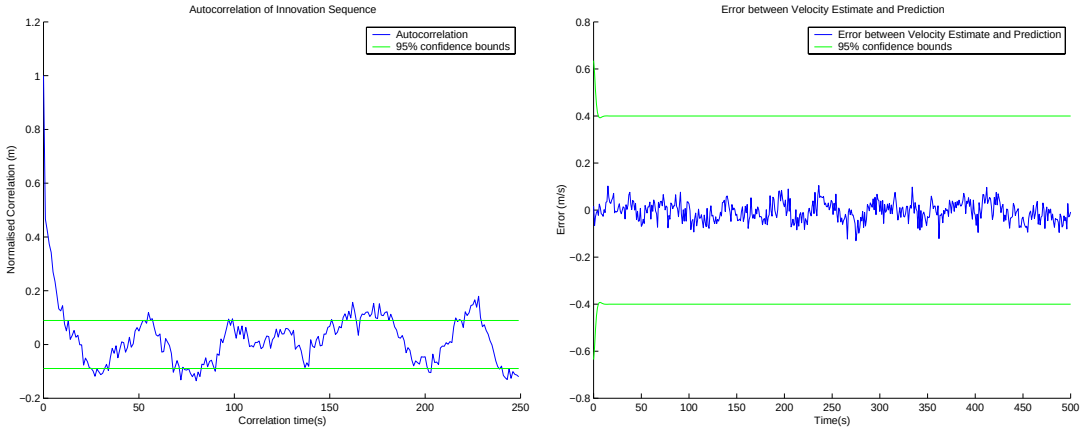
From these four cases, we can deduce a procedure for the detection and correction of errors in estimated process and observation noise. The first step in the procedure involves testing that the ratio of process to observation noise is correct. We can see from Figures 16 and 19 that if this ratio is too small (process noise is relatively too small with respect to observation noise), then the innovation sequence will be correlated. The most sensitive test of for this is that the autocorrelation of the innovation sequence should be white; being zero everywhere (except at zero correlation time)within specified confidence bounds. These correlations appear regardless of the absolute noise levels, and so this procedure allows adjustment of the noise ratio without needing to be precise about absolute noise levels. The second step in the procedure involves testing and setting the overall level of the combined process and observation noise models. A first indication that the levels are correct can be obtained from the innovation sequence itself. However, the χ^2 test provides the most sensitive indication of overall noise levels, and is the test that should be used to set absolute noise values. An important point to note is that the tuning process is much more sensitive to changes in observation noise than to changes in process noise. This is because, in this example, the process noise drives velocity and not position, whereas observation noise affects position directly.

6.5.3 Process Model Errors

We now consider errors in the specification of process model. The most common errors in this class occur when significant states in the true process are omitted from the model used



(a) (b)



(c) (d)

Figure 19: The Effect of Over-Estimating Observation Noise Levels, $\hat{\sigma}_q = \sigma_q = 0.01$ and $\hat{\sigma}_r = 10\sigma_r = 1.0$. (a) Innovation Sequence and Estimated Innovation Standard Deviation; (b) Normalized Innovations Squared, Moving Average and 95% Confidence Bounds; (c) Normalized, Biased Autocorrelation of the Innovation Sequence and 95% Confidence Bounds; (d) Error between Velocity Estimate and Velocity Prediction, With 95% Confidence Bounds.

by the filter. The emphasis here is on the word 'significant'. In all real cases, the model will only be an approximation to the true process, however, it is essential that the model describe those states which have an important impact on the validity and performance of the filter. A complete discussion of reduced order modeling is beyond the scope of this course. Here we will concentrate on identifying the absence of a significant unmodeled state, and techniques for tuning filter performance to compensate for this omission.

We consider specifically the case of using a constant-velocity filter to track a particle which has substantial accelerations and whose true process model is described by the linear discrete-time state transition equation

$$\begin{bmatrix} x(k) \\ \dot{x}(k) \\ \ddot{x}(k) \end{bmatrix} = \begin{bmatrix} 1 & \Delta T & \Delta T^2/2 \\ 0 & 1 & \Delta T \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x(k-1) \\ \dot{x}(k-1) \\ \ddot{x}(k-1) \end{bmatrix} + \begin{bmatrix} \Delta T^3/3 \\ \Delta T^2/2 \\ \Delta T \end{bmatrix} v(k),$$

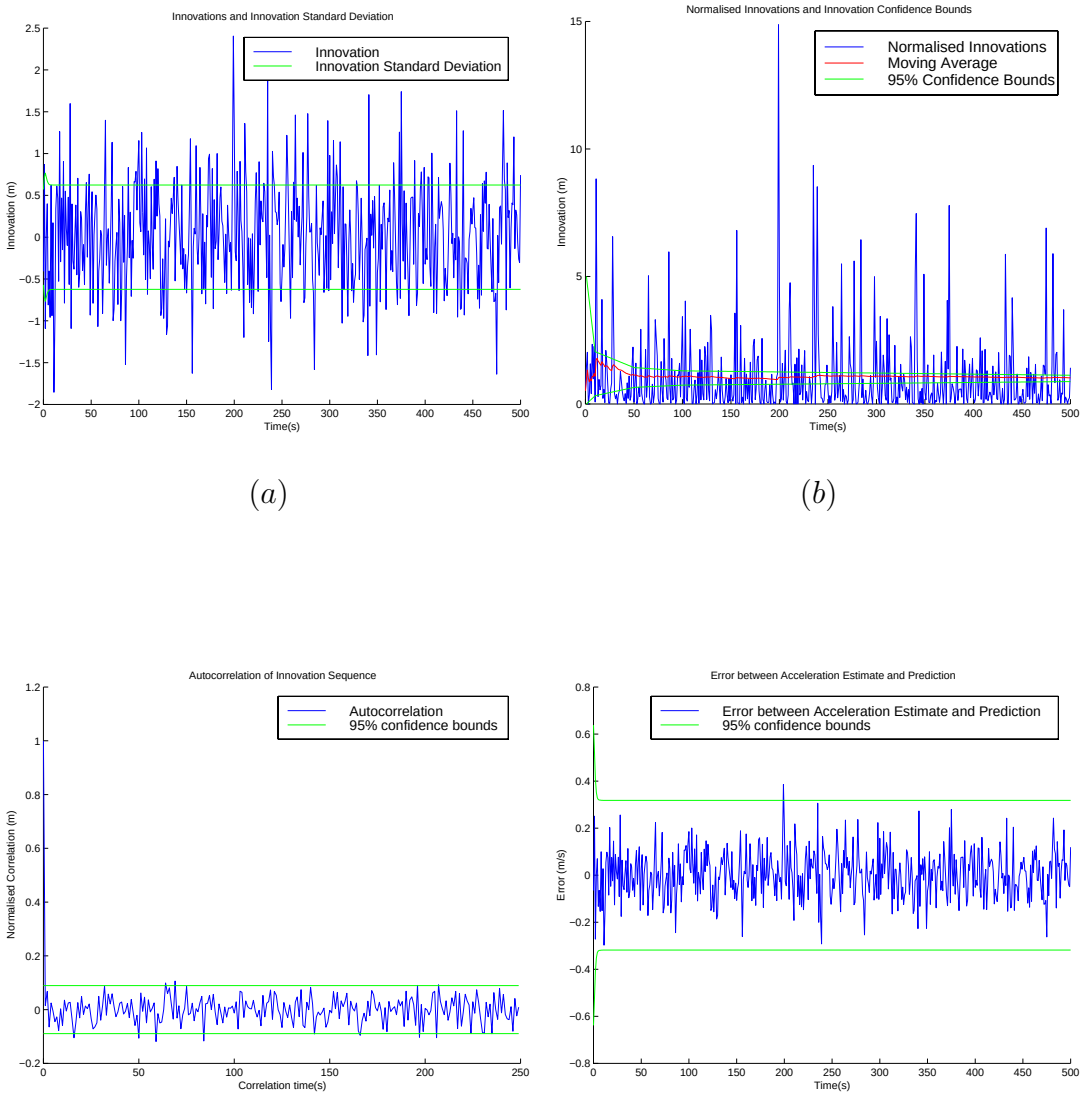
with

$$\mathbf{Q}(k) = \mathrm{E}[\mathbf{v}(k)\mathbf{v}^T(k)] = \begin{bmatrix} \Delta T^5/20 & \Delta T^4/8 & \Delta T^3/6 \\ \Delta T^4/8 & \Delta T^3/3 & \Delta T^2/2 \\ \Delta T^3/6 & \Delta T^2/2 & \Delta T \end{bmatrix} q.$$

For reference, Figure 20 shows the results for an exactly matched constant-acceleration filter with $\hat{\sigma}_q = \sigma_q = 0.01$ and $\hat{\sigma}_r = \sigma_r = 0.1$. Figure 20(a) shows the innovation sequence together with estimated innovation standard deviation. Figure 20(b) shows the normalized innovations squared, together with its moving average and 95% confidence bounds. Figure 20(c) shows the normalized, biased, autocorrelation of the innovation sequence, together with 95% confidence bounds. Figure 20(d) shows the error between acceleration estimate and acceleration prediction, together with estimated 95% confidence bounds.

Figure 21 shows the results obtained for a constant-velocity filter with $\hat{\sigma}_q = 0.01$ and $\hat{\sigma}_r = 0.1$. Figure 21(a) shows the innovation sequence together with estimated innovation standard deviation. This is clearly not white nor even periodically correlated, rather the innovation sequence looks like a first-order Gauss Markov process. Figure 21(b) shows the normalized innovations squared, together with its moving average and 95% confidence bounds. The process is clearly not even stationary and so the mean drifts way beyond the set confidence bounds. Figure 21(c) shows the normalized, biased, autocorrelation of the innovation sequence, together with 95% confidence bounds. As expected, the autocorrelation shows a pure exponential time correlation characteristic of first-order Gauss-Markov processes. Figure 21(d) shows the error between velocity estimate and velocity prediction, together with estimated 95% confidence bounds. This too resembles a first order Gauss-Markov process. This might be expected as the velocity error is driven by the process acceleration which itself is the result of integrating a white-noise process.

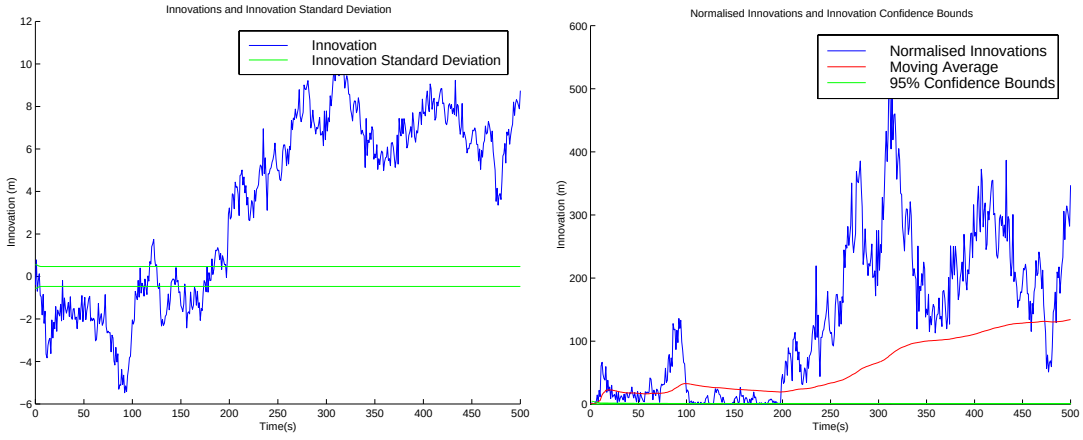
Figure 22 shows the results obtained for a constant-velocity filter with process noise increased by a factor of ten $\hat{\sigma}_q = 0.1$ and the observation noise the same $\sigma_r = 0.1$. Figure 22(a) shows the innovation sequence together with estimated innovation standard



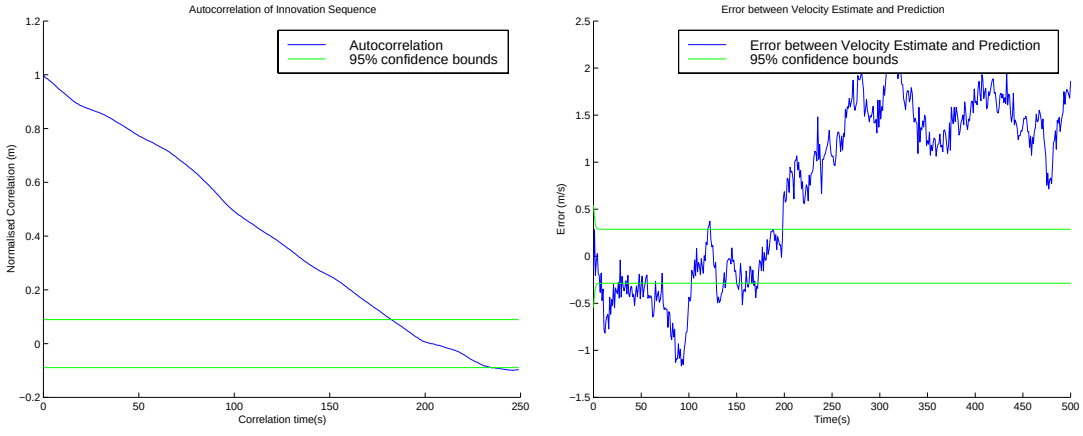
(a) (b)

(c) (d)

Figure 20: Exactly Matched Constant-Acceleration Filter with $\hat{\sigma}_q = \sigma_q = 0.01$ and $\hat{\sigma}_r = \sigma_r = 0.1$. (a) Innovation Sequence and Estimated Innovation Standard Deviation; (b) Normalized Innovations Squared, Moving Average and 95% Confidence Bounds; (c) Normalized, Biased Autocorrelation of the Innovation Sequence and 95% Confidence Bounds; (d) Error between Acceleration Estimate and Acceleration Prediction, With 95% Confidence Bounds.

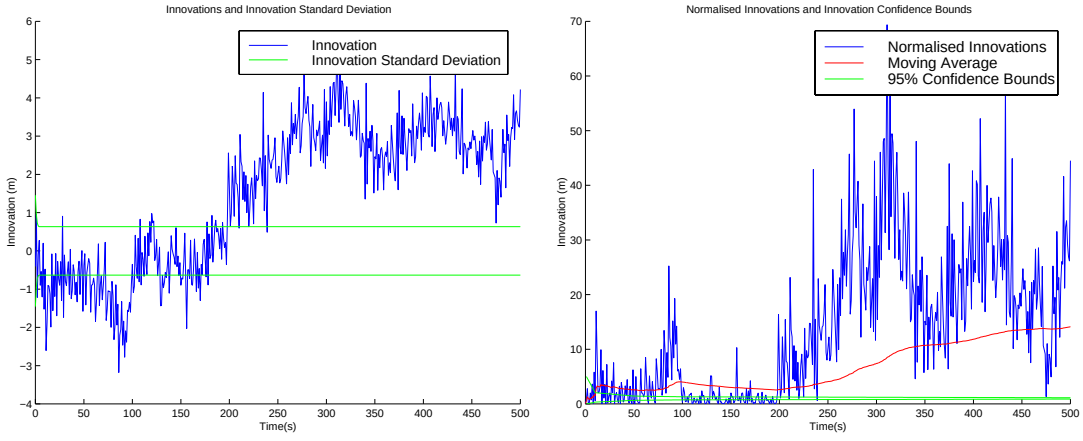


(a) (b)

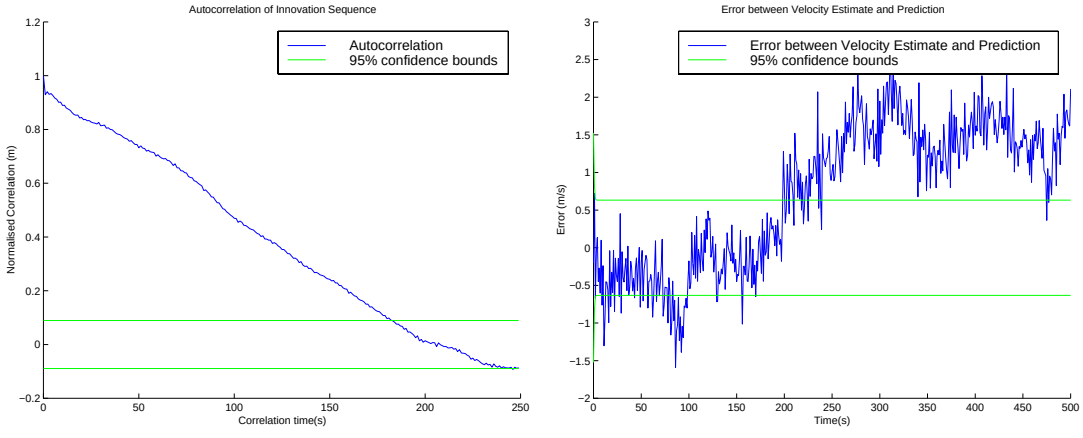


(c) (d)

Figure 21: Constant-Velocity Filter tracking a Constant-Acceleration Particle, $\hat{\sigma}_q = 0.01$ and $\hat{\sigma}_r = \sigma_r = 0.1$. (a) Innovation Sequence and Estimated Innovation Standard Deviation; (b) Normalized Innovations Squared, Moving Average and 95% Confidence Bounds; (c) Normalized, Biased Autocorrelation of the Innovation Sequence and 95% Confidence Bounds; (d) Error between Velocity Estimate and Velocity Prediction, With 95% Confidence Bounds.



(a) (b)



(c) (d)

Figure 22: Constant-Velocity Filter tracking a Constant-Acceleration Particle: The Effect of Increasing Process Noise by a factor of 10, $\hat{\sigma}_q = 0.1$ and $\hat{\sigma}_r = \sigma_r = 0.1$. (a) Innovation Sequence and Estimated Innovation Standard Deviation; (b) Normalized Innovations Squared, Moving Average and 95% Confidence Bounds; (c) Normalized, Biased Autocorrelation of the Innovation Sequence and 95% Confidence Bounds; (d) Error between Velocity Estimate and Velocity Prediction, With 95% Confidence Bounds.

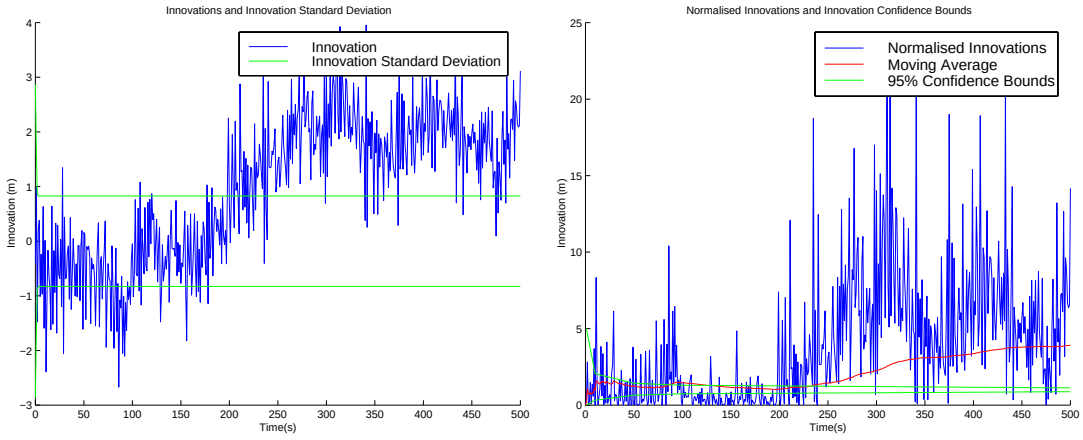
deviation. This is an improvement on Figure 21, showing a mainly white character superimposed on a long-term (Gauss-Markov) drift. Figure 22(b) shows the normalized innovations squared, together with its moving average and 95% confidence bounds. The mean starts out within bounds but gradually drifts away from the whiteness hypothesis bounds. Figure 22(c) shows the normalized, biased, autocorrelation of the innovation sequence, together with 95% confidence bounds. The autocorrelation still shows substantial exponential time correlation whose level has been reduced by the addition of a characteristic whiteness 'spike' at zero time lag. Figure 22(d) shows the error between velocity estimate and velocity prediction, together with estimated 95% confidence bounds. Again, this shows substantial different characteristics superimposed on a gradual drift.

Figure 22 indicates that by increasing process noise still further we might be able to meet the required whiteness tests. In doing so, we need also to reduce observation noise so that the total noise level satisfies the χ^2 test. Figure 23 shows the results obtained for a constant-velocity filter with process noise increased by a factor of forty $\hat{\sigma}_q = 0.4$ and the observation noise reduced by half $\sigma_r = 0.05$. Figure 23(a) shows the innovation sequence together with estimated innovation standard deviation. This is only a marginal improvement on Figure 22 which although well bounded, still shows a marked drift over time. Figure 23(b) shows the normalized innovations squared, together with its moving average and 95% confidence bounds. The mean is just about in bounds over the whole time sequence which indicates that the total noise levels are approximately correct. Figure 23(c) shows the normalized, biased, autocorrelation of the innovation sequence, together with 95% confidence bounds. The autocorrelation still shows substantial exponential time correlation although the level has again been reduced by the further addition of a larger white' component. Figure 23(d) shows the error between velocity estimate and velocity prediction, together with estimated 95% confidence bounds. Again, although there is a significant white component, there is still a gradual drift.

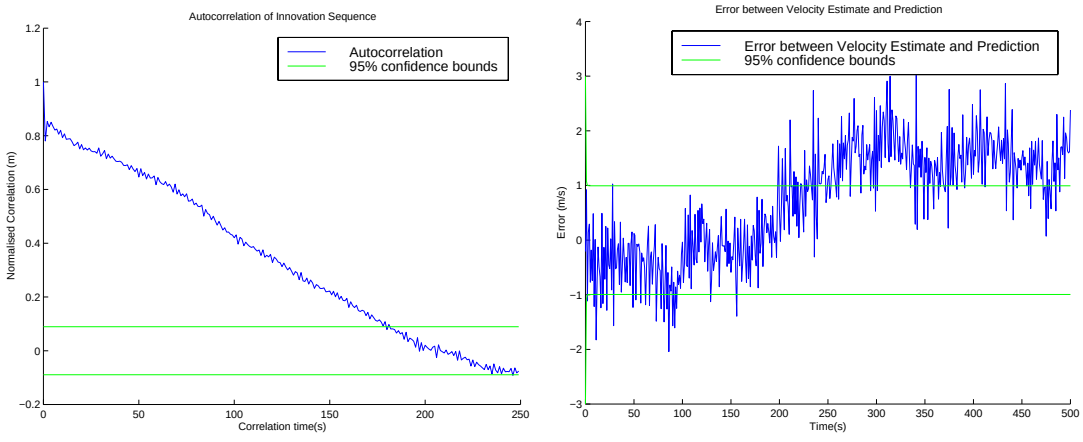
Figure 23 indicates that no amount of increased process noise will entirely compensate for the long-term drift caused by accelerations in the observed innovation sequence. One way of overcoming this problem is to introduce some 'stabilizing' noise $s(k)$ directly into the model of the observed state (position in this case). The new filter process model now looks like the following

$$\begin{bmatrix} x(k) \\ \dot{x}(k) \end{bmatrix} = \begin{bmatrix} 1 & \Delta T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x(k-1) \\ \dot{x}(k-1) \end{bmatrix} + \begin{bmatrix} \Delta T^2/2 \\ \Delta T \end{bmatrix} v(k) + \begin{bmatrix} \Delta T \\ 0 \end{bmatrix} s(k)$$

The stabilizing noise is assumed white with variance σ_s^2. Figure 24 shows the results obtained for a stabilized constant-velocity filter with process noise strength $\hat{\sigma}_q = 0.025$, observation noise strength $\hat{\sigma}_r = 0.05$, and stabilization noise strength $\sigma_s = 0.015$ Figure 24(a) shows the innovation sequence together with estimated innovation standard deviation. The sequence is well bounded and looks white with no noticeable drift. Figure 24(b) shows the normalized innovations squared, together with its moving average and 95% confidence bounds. The mean is just about in bounds over the whole time sequence

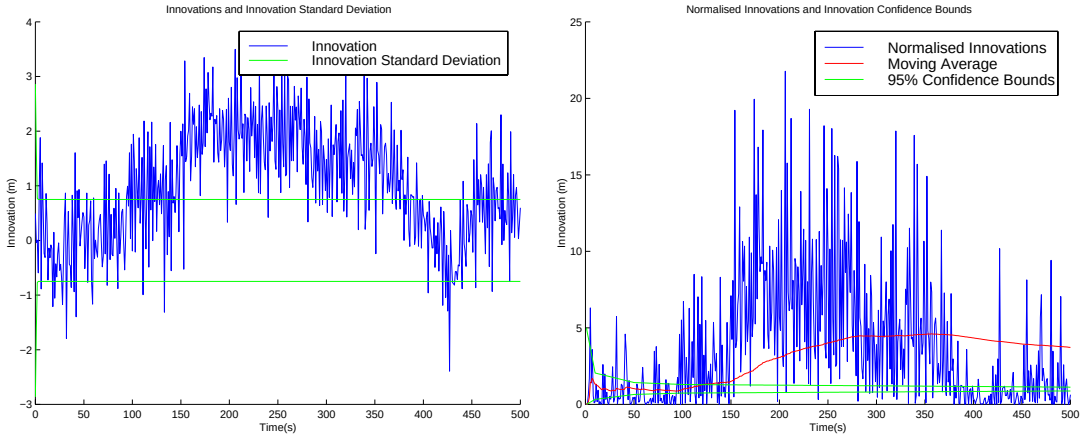


(a) (b)

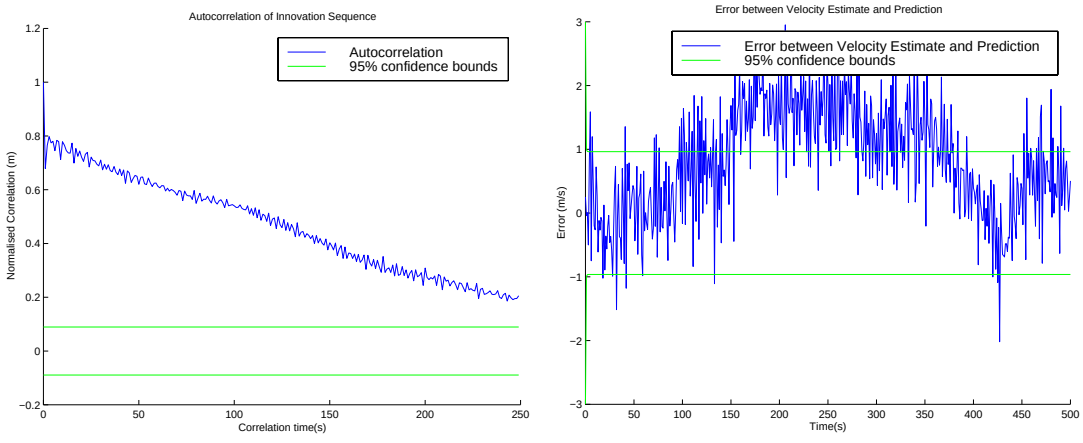


(c) (d)

Figure 23: Constant-Velocity Filter tracking a Constant-Acceleration Particle: The Effect of Increasing Process Noise Levels by a factor of 40 and decreasing Observation Noise Levels by a factor of 2, with $\hat{\sigma}_q = 0.4$ and $\hat{\sigma}_r = 0.05$. (a) Innovation Sequence and Estimated Innovation Standard Deviation; (b) Normalized Innovations Squared, Moving Average and 95% Confidence Bounds; (c) Normalized, Biased Autocorrelation of the Innovation Sequence and 95% Confidence Bounds; (d) Error between Velocity Estimate and Velocity Prediction, With 95% Confidence Bounds.



(a) (b)



(c) (d)

Figure 24: Constant-Velocity Filter tracking a Constant-Acceleration Particle: The Effect of Adding Stabilizing Noise, $\sigma_s = 0.015$, with $\hat{\sigma}_q = 0.025$ and $\hat{\sigma}_r = 0.05$. (a) Innovation Sequence and Estimated Innovation Standard Deviation; (b) Normalized Innovations Squared, Moving Average and 95% Confidence Bounds; (c) Normalized, Biased Autocorrelation of the Innovation Sequence and 95% Confidence Bounds; (d) Error between Velocity Estimate and Velocity Prediction, With 95% Confidence Bounds.

which indicates that the total noise levels are correct. Figure 24(c) shows the normalized, biased, autocorrelation of the innovation sequence, together with 95% confidence bounds. The autocorrelation indicates the innovation sequence is indeed white as required. Figure 24(d) shows the error between velocity estimate and velocity prediction, together with estimated 95% confidence bounds. This error now looks approximately white.

The use of approximately noise is quite common, particularly in non-linear filtering problems, to account for both modeling and linearisation errors. Its effect is to directly associate these errors with corresponding errors in observed states. The approximately noise is dissociated from the process kinematics so that one result of this is that the correlations developed between different states will no longer reflect the original designed model. For this reason, the approximately noise level should never be made too large. A second hidden consequence of adding approximately noise is that the state predictions and state estimates have larger covariances than the filter which uses a correct process model.

6.6 Steady-State Filters

We have seen that for systems described by linear time-invariant state and observation models of the form

$$\mathbf{x}(k) = \mathbf{F}\mathbf{x}(k-1) + \mathbf{G}\mathbf{u}(k) + \mathbf{v}(k), \tag{189}$$

$$\mathbf{z}(k) = \mathbf{H}\mathbf{x}(k) + \mathbf{w}(k), \tag{190}$$

driven by constant uncorrelated noise of the form

$$\mathrm{E}[\mathbf{v}(i)\mathbf{v}^T(j)] = \mathbf{Q}\delta_{ij}, \qquad \mathrm{E}[\mathbf{w}(i)\mathbf{w}^T(j)] = \mathbf{R}\delta_{ij}, \tag{191}$$

the estimate covariance $\mathbf{P}(k \mid k)$, prediction covariance $\mathbf{P}(k \mid k-1)$, and innovation covariance $\mathbf{S}(k)$ all tend to constant steady-state values after only a few time-steps. As the gain matrix $\mathbf{W}(k)$ is a function only of these covariances and the constant system models $\mathbf{H}$ and $\mathbf{F}$, it too tends to a constant steady-state value. Primarily, the reason for this is simply that the covariance calculations do not depend on the observations that are actually made, but only on the constant values of the therefore $\mathbf{F}$, $\mathbf{H}$, $\mathbf{Q}$, and $\mathbf{R}$. Specifically

$$\begin{aligned}
\mathbf{P}(k \mid k-1) &= \mathbf{F}\mathbf{P}(k-1 \mid k-1)\mathbf{F}^T + \mathbf{Q}, \\
\mathbf{S}(k) &= \mathbf{H}\mathbf{P}(k \mid k-1)\mathbf{H}^T + \mathbf{R}, \\
\mathbf{W}(k) &= \mathbf{P}(k \mid k-1)\mathbf{H}^T\mathbf{S}^{-1}(k), \\
\mathbf{P}(k \mid k) &= \mathbf{P}(k \mid k-1) - \mathbf{W}(k)\mathbf{S}(k)\mathbf{W}^T(k)
\end{aligned} \tag{192}$$

That the covariance and gain-matrix tend to a constant steady-state value is not obvious from these equations other than simply from observation of a simulated calculation. It

can however be shown that this is the case by considering the equivalent continuous time variance propagation equations (the Riccati Equation). In practice, the steady-state values achieved can be computed in closed form for only the simplest of problems.

If the covariances and gain matrix all tend to a constant steady-state value after only a few time-steps, it seems sensible that the computationally intensive process of computing these values on-line be avoided by simply inserting these constant values in to the filter from the start. In effect, this means that a constant gain matrix is used in the computation of the estimates

$$\hat{\mathbf{x}}(k \mid k) = \hat{\mathbf{x}}(k \mid k-1) + \mathbf{W}\left[\mathbf{z}(k) - \mathbf{H}\hat{\mathbf{x}}(k \mid k-1)\right], \tag{193}$$

where $\mathbf{W} = \lim_{k\to\infty} \mathbf{W}(k)$. This steady-state value of the gain matrix is most easily computed by simply simulating the covariance calculation off-line (this of course does not require that any observations are made).

In the case of a "constant velocity" model, the gain matrix can be described by two dimensionless coefficients α and β,

$$\hat{\mathbf{x}}(k \mid k) = \hat{\mathbf{x}}(k \mid k-1) + \begin{bmatrix} \alpha \\ \beta/T \end{bmatrix}\left[\mathbf{z}(k) - \mathbf{H}\hat{\mathbf{x}}(k \mid k-1)\right]. \tag{194}$$

This is known as the α–β filter. In the case of a "constant acceleration" model, the gain matrix can be described by three dimensionless coefficients α, β and γ,

$$\hat{\mathbf{x}}(k \mid k) = \hat{\mathbf{x}}(k \mid k-1) + \begin{bmatrix} \alpha \\ \beta/T \\ \gamma/T^2 \end{bmatrix}\left[\mathbf{z}(k) - \mathbf{H}\hat{\mathbf{x}}(k \mid k-1)\right]. \tag{195}$$

This is known as the α–β–γ filter. In practice, these constant parameter values are theorem through a combination of guess and judgement.

The steady-state filter provides a substantial reduction in the amount of on-line computation that must be performed by the filter which in many situations may be invaluable, particularly in multi-target applications when many filters must be employed simultaneously. Providing that the assumption of constant system and noise models applies, the error incurred by using a steady-state filter is significant only during initialization of the filter.

In practice, steady-state filters are used even in situations where the basic assumptions of linear time-invariance and constant noise injection are violated. This is particularly true in cases where computational resources are strictly limited (an increasingly rare occurrence).

7 The Extended Kalman Filter

In almost all real situations the state or environment of interest in a data-fusion problem will not evolve in a linear manner and so simple linear models such as Equation 121 will not

be adequate to describe the environment models and quantities of interest. Furthermore, the observations made by the sensors of the system will in many cases not be linearly related to the states that describe the environment. Consequently, the methods that we have developed for estimation with linear models can not be directly applied to these problems.

In the development of estimators for linear systems, we have employed a number of criteria by which to judge the relative worth of different state estimates. In particular, the single-sensor Kalman filter algorithm will compute an estimate which minimizes mean-squared state error. We have seen that in general this estimate is given by the conditional mean of the state given all observations up to the current time.

$$\hat{\mathbf{x}}(k \mid k) = \mathrm{E}[\mathbf{x}(k) \mid \mathbf{Z}^k] \ . \tag{196}$$

When all stated assumptions about linearity of process and whiteness of noise are valid, the conditional mean can be calculated recursively from a linear sum of state prediction and observation according to

$$\hat{\mathbf{x}}(k \mid k) = \hat{\mathbf{x}}(k \mid k-1) + \mathbf{W}(k) \left[\mathbf{z}(k) - \mathbf{H}(k)\hat{\mathbf{x}}(k \mid k-1)\right]. \tag{197}$$

Clearly, when the state evolves in a non-linear way or the observations made are only related to the states in some non-linear manner then the assumptions required for Equation 197 to produce the conditional mean as an estimate will be violated. In such cases, we should in principle return to the definition of the estimate given in Equation 196 and make explicit the fact that the estimate be computed as the first moment of the conditional distribution of state given the observation sequence

$$\hat{\mathbf{x}}(k \mid k) = \int_{-\infty}^{+\infty} \mathbf{x}(k) f(\mathbf{x}(k) \mid \mathbf{Z}^k) \mathrm{d}\mathbf{x}. \tag{198}$$

In practice, if the state and observation model are non-linear, it is all but impossible to form the required probability density functions and to compute the conditional mean except through extensive numeric integration.

The question arises as to whether it is possible to find an estimation algorithm, in cases where the state and observation models are non-linear, which has the same linear recursive form as Equation 197 and which provides an estimate which is at least approximately equal to the conditional mean in Equation 196. The answer is almost always 'yes'. By suitable linearisation of the non-linear state and observation equations it is often possible to find a linear estimator for a non-linear system which has the same basic form as the linear Kalman filter. The estimator is called the **Extended Kalman Filter** or EKF. The extended Kalman filter will not in general be the best estimator under any reasonable criteria; there will almost certainly be, at least in theory, a non-linear estimator which will perform better. However, the extended Kalman filter will be the best *linear estimator* with respect to minimum-mean-squared error. In practice, the extended Kalman filter can

be made to produce extremely good estimates of the states of a non-linear system even when the assumptions required for its derivation clearly do not hold. As a consequence the extended Kalman filter has found wide application in a variety of estimation problems(see for example [1]).

A few words of caution are needed. As we will see, the extended Kalman filter is conceptually very simple and its derivation follows from simple arguments of linearisation and the results already obtained for the linear Kalman filter algorithm. However, the difficulty arises in implementation; the extended Kalman filter can be made to work well but it may also perform very badly indeed and may even become unstable with the estimates diverging from the true state. This is most often due to lack of careful modeling of sensors and environment, and on a failure to understand the limitations of the algorithm, rather than any inherent fault with the algorithm itself. Unlike the linear Kalman filter, which will almost always work without concern, a considerable amount of engineering judgment needs to be employed if the extended Kalman filter is to be made to work well.

7.1 System and Observation Models

We start by assuming that the system of interest can be described by a non-linear discrete-time state transition equation in the form

$$\mathbf{x}(k) = \mathbf{f}\left(\mathbf{x}(k-1), \mathbf{u}(k), k\right) + \mathbf{v}(k), \tag{199}$$

where $\mathbf{x}(k-1)$ is the state at time $k-1$, $\mathbf{u}(k)$ is a known input vector, $\mathbf{v}(k)$ is some additive process noise, $\mathbf{x}(k)$ is the state at time step k, and $\mathbf{f}(\cdot, \cdot, k)$ is a non-linear state transition function mapping previous state and current control input to the current state.

We assume that observations of the state of this system are made according to a non-linear observation equation in the form

$$\mathbf{z}(k) = \mathbf{h}\left(\mathbf{x}(k)\right) + \mathbf{w}(k) \tag{200}$$

where $\mathbf{z}(k)$ is the observation made at time k, $\mathbf{x}(k)$ is the state at time k, $\mathbf{w}(k)$ is some additive observation noise, and $\mathbf{h}(\cdot, k)$ is a non-linear observation model mapping current state to observations.

As before, we assume that the noises $\mathbf{v}(k)$ and $\mathbf{w}(k)$ are all Gaussian, temporally uncorrelated and zero-mean

$$\mathrm{E}[\mathbf{v}(k)] = \mathrm{E}[\mathbf{w}(k)] = \mathbf{0}, \qquad \forall k,$$

with corresponding covariance

$$\mathrm{E}[\mathbf{v}(i)\mathbf{v}^T(j)] = \delta_{ij}\mathbf{Q}(i), \quad \mathrm{E}[\mathbf{w}(i)\mathbf{w}(j)] = \delta_{ij}\mathbf{R}(i).$$

We also assume that the process and observation noises are uncorrelated

$$\mathrm{E}[\mathbf{v}(i)\mathbf{w}^T(j)] = \mathbf{0}, \qquad \forall i, j.$$

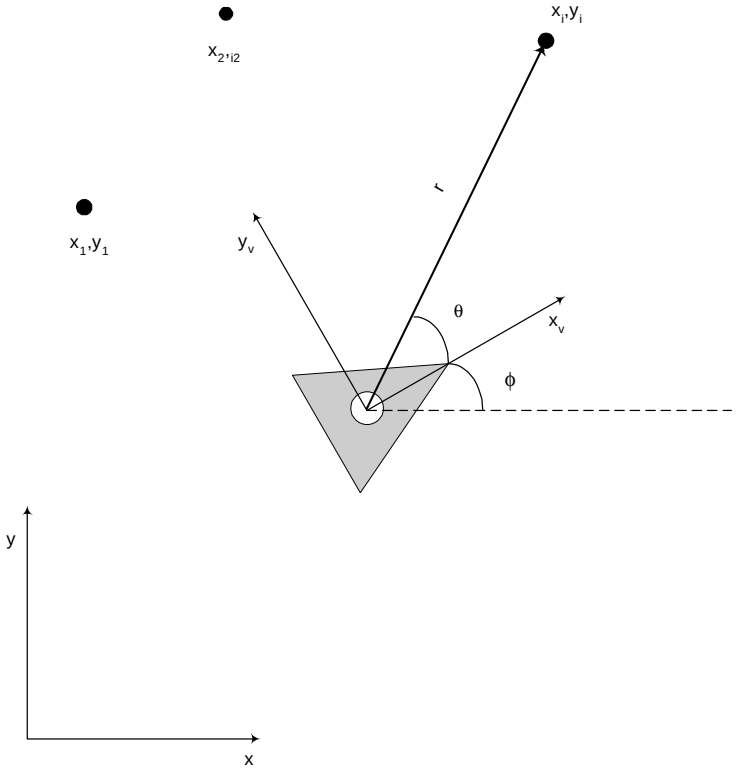


Figure 25: The Geometry of a Mobile Vehicle

Example 18

Consider the motion of an object a vehicle in the plane as shown in Figure 25. The state of the vehicle at any time instant k is determined by its location and orientation $\mathbf{x}(k) = [x(k), y(k), \phi(k)]^T$. *The control is exerted over the vehicle motion through a demanded velocity and direction of travel* $\mathbf{u}(k) = [V(k), \psi(k)]^T$. *The motion of the vehicle can now be described in terms of the simple non-linear state transition equation*

$$\begin{bmatrix} x(k) \\ y(k) \\ \phi(k) \end{bmatrix} = \begin{bmatrix} x(k-1) + TV(k)\cos(\phi(k-1) + \psi(k)) \\ y(k-1) + TV(k)\sin(\phi(k-1) + \psi(k)) \\ \phi(k-1) + T\frac{V(k)}{B}\sin(\psi(k)) \end{bmatrix} + \begin{bmatrix} q_x(k) \\ q_y(k) \\ q_\phi(k) \end{bmatrix} \tag{201}$$

where B *is the wheel base-line,* T *the time interval between time steps, and* $\mathbf{q}(k) = [q_x(k), q_y(k), q_\phi(k)]^T$ *the random vector describing the noise in the process due both to modeling errors and uncertainty in control.*

We will assume that the vehicle is equipped with a sensor that can measure the range and bearing to a number of beacons sited at fixed locations $\mathbf{B}_i = [X_i, Y_i]^T$, $i = 1, \cdots, N$, *in the environment. The measurement equation for each beacon is simply given by the*

non-linear measurement model

$$\begin{bmatrix} z_r^i(k) \\ z_\theta^i(k) \end{bmatrix} = \begin{bmatrix} \sqrt{(X_i - x(k))^2 + (Y_i - y(k))^2} \\ \arctan\left(\frac{Y_i - y(k)}{X_i - x(k)}\right) - \phi(k) \end{bmatrix} + \begin{bmatrix} r_r^i(k) \\ r_\theta^i(k) \end{bmatrix}, \tag{202}$$

where the random vector $\mathbf{r}_k(i) = [r_r^i(k), r_\theta^i(k)]^T$ *describes the noise in the observation process due to both modeling errors and uncertainty in observation.*

7.2 State Prediction

We begin by assuming that we already have an estimate at time $k-1$ which is approximately equal to the conditional mean,

$$\hat{\mathbf{x}}(k-1 \mid k-1) \approx \mathrm{E}[\mathbf{x}(k-1) \mid Z^{k-1}]. \tag{203}$$

We would now like to find a prediction $\hat{\mathbf{x}}(k \mid k-1)$ for the state at the next time k based only on the information available up to time $k-1$. Following Equation 67 we can expand Equation 199 as a Taylor series about the estimate $\hat{\mathbf{x}}(k-1 \mid k-1)$ to obtain

$$\begin{aligned} \mathbf{x}(k) &= \mathbf{f}\left(\hat{\mathbf{x}}(k-1 \mid k-1), \mathbf{u}(k), k\right) + \nabla \mathbf{f}_x(k)\left[\mathbf{x}(k-1) - \hat{\mathbf{x}}(k-1 \mid k-1)\right] \\ &\quad + O\left(\left[\mathbf{x}(k-1) - \hat{\mathbf{x}}(k-1 \mid k-1)\right]^2\right) + \mathbf{v}(k) \end{aligned} \tag{204}$$

where $\nabla \mathbf{f}_x(k)$ is the Jacobian of $\mathbf{f}$ evaluated at $\mathbf{x}(k-1) = \hat{\mathbf{x}}(k-1 \mid k-1)$. Truncating Equation 204 at first order, and taking expectations conditioned on the first $k-1$ observations gives an equation for the state prediction as

$$\begin{aligned} \hat{\mathbf{x}}(k \mid k-1) &= \mathrm{E}[\mathbf{x}(k) \mid \mathbf{Z}^{k-1}] \\ &\approx \mathrm{E}\left[\mathbf{f}\left(\hat{\mathbf{x}}(k-1 \mid k-1), \mathbf{u}(k), k\right)\right. \\ &\quad \left. + \nabla \mathbf{f}_x(k)\left[\mathbf{x}(k-1) - \hat{\mathbf{x}}(k-1 \mid k-1)\right] + \mathbf{v}(k) \mid \mathbf{Z}^{k-1}\right] \\ &= \mathbf{f}\left(\hat{\mathbf{x}}(k-1 \mid k-1), \mathbf{u}(k), k\right), \end{aligned} \tag{205}$$

where the last line follows from the assumption that the estimate $\hat{\mathbf{x}}(k-1 \mid k-1)$ is approximately equal to the conditional mean (Equation 203) and that the process noise $\mathbf{v}(k)$ has zero mean.

The state estimate error at a time i given all observations up to time j is simply defined as

$$\tilde{\mathbf{x}}(i \mid j) \triangleq \mathbf{x}(i) - \hat{\mathbf{x}}(i \mid j), \tag{206}$$

and the state covariance is defined as the outer product of this error with itself conditioned on the observations made

$$\mathbf{P}(i \mid j) \triangleq \mathrm{E}[\tilde{\mathbf{x}}(i \mid j)\tilde{\mathbf{x}}^T(i \mid j) \mid \mathbf{Z}^j] \ . \tag{207}$$

Specifically, the prediction error $\tilde{\mathbf{x}}(k \mid k-1)$ can be found by subtracting the true state $\mathbf{x}(k)$ given in Equation 204 from the prediction given in Equation 205

$$\begin{aligned}
\tilde{\mathbf{x}}(k \mid k-1) &= \mathbf{x}(k) - \hat{\mathbf{x}}(k \mid k-1) \\
&= \mathbf{f}\left(\hat{\mathbf{x}}(k-1 \mid k-1), \mathbf{u}(k), k\right) + \nabla\mathbf{f}_x(k)\left[\mathbf{x}(k-1) - \hat{\mathbf{x}}(k-1 \mid k-1)\right] \\
&\quad + O\left(\left[\mathbf{x}(k-1) - \hat{\mathbf{x}}(k-1 \mid k-1)\right]^2\right) + \mathbf{v}(k) - \mathbf{f}\left(\hat{\mathbf{x}}(k-1 \mid k-1), \mathbf{u}(k), k\right) \\
&\approx \nabla\mathbf{f}_x(k)\left[\mathbf{x}(k-1) - \hat{\mathbf{x}}(k-1 \mid k-1)\right] + \mathbf{v}(k) \\
&= \nabla\mathbf{f}_x(k)\tilde{\mathbf{x}}(k-1 \mid k-1) + \mathbf{v}(k)
\end{aligned} \tag{208}$$

Clearly the prediction is unbiased when the previous estimate is unbiased and the condition that the noise sequences are zero mean and white hold. Taking the expectation, condition on the observations made up to time $k-1$ of the outer product of the prediction error gives an expression for the prediction covariance in terms of the covariance of the previous estimate

$$\begin{aligned}
\mathbf{P}(k \mid k-1) &\triangleq \mathrm{E}[\tilde{\mathbf{x}}(k \mid k-1)\tilde{\mathbf{x}}^T(k \mid k-1) \mid \mathbf{Z}^{k-1}] \\
&\approx \mathrm{E}[(\nabla\mathbf{f}_x(k)\tilde{\mathbf{x}}(k-1 \mid k-1) + \mathbf{v}(k))\left(\nabla\mathbf{f}_x(k)\tilde{\mathbf{x}}(k-1 \mid k-1) + \mathbf{v}(k)\right)^T \mid \mathbf{Z}^{k-1}] \\
&= \nabla\mathbf{f}_x(k)\mathrm{E}[\tilde{\mathbf{x}}(k-1 \mid k-1)\tilde{\mathbf{x}}^T(k-1 \mid k-1) \mid \mathbf{Z}^{k-1}]\ \nabla\mathbf{f}_x^T(k) + \mathrm{E}[\mathbf{v}(k)\mathbf{v}^T(k)] \\
&= \nabla\mathbf{f}_x(k)\mathbf{P}(k-1 \mid k-1)\nabla\mathbf{f}_x^T(k) + \mathbf{Q}(k),
\end{aligned} \tag{209}$$

where the last two lines follow from the fact that the estimate and true state at time $k-1$ are statistically dependent only on the noise terms $\mathbf{w}(j)$ and $\mathbf{v}(j)$, $j \leq k-1$, and so, by assumption, are uncorrelated with the process noise $\mathbf{v}(k)$ at time k.

Example 19

Continuing our mobile vehicle example, we assume that we have an estimate $\hat{\mathbf{x}}(k-1 \mid k-1) = [\hat{x}(k-1 \mid k-1), \hat{y}(k-1 \mid k-1), \hat{\phi}(k-1 \mid k-1)]^T$ *of vehicle location at time* $k-1$ *based on all observations made up to time* $k-1$*, and that the control input* $\mathbf{u}(k) = [V(k), \psi(k)]^T$ *is known perfectly. According to Equation 205, we can obtain a prediction of the vehicle location at the next time step by simply substituting this previous estimate into the state*

transition equation, Equation 201, and assuming zero process noise,

$$\begin{bmatrix} \hat{x}(k \mid k-1) \\ \hat{y}(k \mid k-1) \\ \hat{\phi}(k \mid k-1) \end{bmatrix} = \begin{bmatrix} \hat{x}(k-1 \mid k-1) + TV(k)\cos(\hat{\phi}(k-1 \mid k-1) + \psi(k)) \\ \hat{y}(k-1 \mid k-1) + TV(k)\sin(\hat{\phi}(k-1 \mid k-1) + \psi(k)) \\ \hat{\phi}(k-1 \mid k-1) + T\frac{V(k)}{B}\sin(\psi(k)) \end{bmatrix}. \quad (210)$$

To find the prediction covariance, we first must find the Jacobian of the state transition equation. Differentiating each element of the state vector in Equation 201 with respect to each state in turn and evaluating the result at the previous estimate $\mathbf{x}(k) = \hat{\mathbf{x}}(k-1 \mid k-1)$*, we obtain*

$$\nabla \mathbf{f}_x(k) = \begin{bmatrix} 1 & 0 & -TV(k)\sin(\hat{\phi}(k-1 \mid k-1) + \psi(k)) \\ 0 & 1 & +TV(k)\cos(\hat{\phi}(k-1 \mid k-1) + \psi(k)) \\ 0 & 0 & 1 \end{bmatrix} \quad (211)$$

For the sake of illustration, we will assume that the covariance in the previous estimate $\mathbf{P}(k-1 \mid k-1)$ *is diagonal, with* $\mathbf{P}(k-1 \mid k-1) = diag\{\sigma_x^2, \sigma_y^2, \sigma_\phi^2, \}$*, and that the process noise covariance is also diagonal* $\mathbf{Q}(k) = diag\{q_x^2, q_y^2, q_\phi^2, \}$*. The prediction covariance can now be computed from Equation 209 as*

$$\mathbf{P}(k \mid k-1) = \begin{bmatrix} \sigma_x^2 + T^2V^2(k)\sin^2(\hat{\phi}(k-1 \mid k-1) + \psi(k))\sigma_\phi^2 + q_x^2 \\ -T^2V^2(k)\sin(\hat{\phi}(k-1 \mid k-1) + \psi(k))\cos(\hat{\phi}(k-1 \mid k-1) + \psi(k))\sigma_\phi^2 \\ -TV(k)\sin(\hat{\phi}(k-1 \mid k-1) + \psi(k))\sigma_\phi^2 \end{bmatrix}$$

$$\begin{bmatrix} -T^2V^2(k)\sin(\hat{\phi}(k-1 \mid k-1) + \psi(k))\cos(\hat{\phi}(k-1 \mid k-1) + \psi(k))\sigma_\phi^2 \\ \sigma_y^2 + T^2V^2(k)\cos^2(\hat{\phi}(k-1 \mid k-1) + \psi(k))\sigma_\phi^2 + q_y^2 \\ TV(k)\cos(\hat{\phi}(k-1 \mid k-1) + \psi(k))\sigma_\phi^2 \end{bmatrix}$$

$$\begin{bmatrix} -TV(k)\sin(\hat{\phi}(k-1 \mid k-1) + \psi(k))\sigma_\phi^2 \\ TV(k)\cos(\hat{\phi}(k-1 \mid k-1) + \psi(k))\sigma_\phi^2 \\ \sigma_\phi^2 + q_\phi^2 \end{bmatrix} \quad (212)$$

This shows that along-path error is dependent only on position uncertainty, whereas cross-path error is dependent on orientation uncertainty, and increases in proportion to distance traveled.

7.3 Observation Prediction and Innovation

We are now interested in computing a predicted observation and a corresponding innovation to be used in updating the predicted state. As with the state equation, we may expand Equation 200, describing the observations made, as a Taylor series about the state

prediction $\hat{\mathbf{x}}(k \mid k-1)$

$$\begin{aligned} \mathbf{z}(k) &= \mathbf{h}\left(\mathbf{x}(k)\right) + \mathbf{w}(k) \\ &= \mathbf{h}\left(\hat{\mathbf{x}}(k \mid k-1)\right) + \nabla \mathbf{h}_x(k)\left[\hat{\mathbf{x}}(k \mid k-1) - \mathbf{x}(k)\right] \\ &\quad + O\left(\left[\hat{\mathbf{x}}(k \mid k-1) - \mathbf{x}(k)\right]^2\right) + \mathbf{w}(k), \end{aligned} \tag{213}$$

where $\nabla \mathbf{h}_x(k)$ is the Jacobian of $\mathbf{h}$ evaluated at $\mathbf{x}(k) = \hat{\mathbf{x}}(k \mid k-1)$. Truncating Equation 213 at first order, and taking expectations conditioned on the first $k-1$ observations gives an equation for the predicted observation as

$$\begin{aligned} \hat{\mathbf{z}}(k \mid k-1) &\triangleq \mathrm{E}[\mathbf{z}(k) \mid \mathbf{Z}^{k-1}] \\ &\approx \mathrm{E}[\mathbf{h}\left(\hat{\mathbf{x}}(k \mid k-1)\right) + \nabla \mathbf{h}_x(k)\left[\hat{\mathbf{x}}(k \mid k-1) - \mathbf{x}(k)\right] + \mathbf{w}(k) \mid \mathbf{Z}^{k-1}] \\ &= \mathbf{h}\left(\hat{\mathbf{x}}(k \mid k-1)\right), \end{aligned} \tag{214}$$

where the last two lines follow from the fact that the state prediction error and the observation noise both have zero mean.

After taking an observation $\mathbf{z}(k)$, the innovation can be found by subtracting the predicted observation as

$$\nu(k) = \mathbf{z}(k) - \mathbf{h}\left(\hat{\mathbf{x}}(k \mid k-1)\right) \tag{215}$$

The innovation covariance can now be found from the mean squared error in the predicted observation as follows: The error in the predicted observation can be approximated by subtracting this prediction from the initial series expansion of the observation in Equation 213 as

$$\begin{aligned} \tilde{\mathbf{z}}(k \mid k-1) &\triangleq \mathbf{z}(k) - \hat{\mathbf{z}}(k \mid k-1) \\ &= \mathbf{h}\left(\hat{\mathbf{x}}(k \mid k-1)\right) + \nabla \mathbf{h}_x(k)\left[\hat{\mathbf{x}}(k \mid k-1) - \mathbf{x}(k)\right] \\ &\quad + O\left(\left[\hat{\mathbf{x}}(k \mid k-1) - \mathbf{x}(k)\right]^2\right) + \mathbf{w}(k) \\ &\quad - \mathbf{h}\left(\hat{\mathbf{x}}(k \mid k-1)\right) \\ &\approx \nabla \mathbf{h}_x(k)\left[\hat{\mathbf{x}}(k \mid k-1) - \mathbf{x}(k)\right] + \mathbf{w}(k) \end{aligned} \tag{216}$$

Note that we have made a distinction between the 'estimated' observation error $\tilde{\mathbf{z}}(k \mid k-1)$ and the actual or measured observation error, the innovation, $\nu(k)$. Squaring the expression for the estimated observation error and taking expectation conditions on the first

$k-1$ measurements gives an equation for the innovation covariance as:

$$\begin{aligned}\mathbf{S}(k) &= \mathrm{E}[\tilde{\mathbf{z}}(k \mid k-1)\tilde{\mathbf{z}}^T(k \mid k-1)] \\ &= \mathrm{E}[(\nabla \mathbf{h}_x(k)\left[\hat{\mathbf{x}}(k \mid k-1) - \mathbf{x}(k)\right] + \mathbf{w}(k))\,(\nabla \mathbf{h}_x(k)\left[\hat{\mathbf{x}}(k \mid k-1) - \mathbf{x}(k)\right] + \mathbf{w}(k))^T] \\ &= \nabla \mathbf{h}_x(k)\mathbf{P}(k \mid k-1)\nabla \mathbf{h}_x^T(k) + \mathbf{R}(k)\end{aligned} \tag{217}$$

where the last line follows from the fact that the state prediction is dependent only on the noise terms $\mathbf{w}(j)$ and $\mathbf{v}(j)$, $j \leq k-1$, and so by assumption is statistically uncorrelated with the current observation noise $\mathbf{w}(k)$.

Example 20

Continuing our mobile robot example, we assume that we have computed a predicted vehicle location, $\hat{\mathbf{x}}(k \mid k-1) = [\hat{x}(k \mid k-1), \hat{y}(k \mid k-1), \hat{\phi}(k \mid k-1)]^T$. *From this location we want to predict the observations we will make. From Equations 214 and 202, we simply have*

$$\begin{bmatrix} \hat{z}_r^i(k \mid k-1) \\ \hat{z}_\theta^i(k \mid k-1) \end{bmatrix} = \begin{bmatrix} \sqrt{(X_i - \hat{x}(k \mid k-1))^2 + (Y_i - \hat{y}(k \mid k-1))^2} \\ \arctan\left(\frac{Y_i - \hat{y}(k|k-1)}{X_i - \hat{x}(k|k-1)}\right) - \hat{\phi}(k \mid k-1) \end{bmatrix}$$

To find the innovation, or observation prediction error covariance, we first must find the Jacobian $\nabla \mathbf{h}_x(k)$. *Differentiating each element of the observation vector in Equation 202 with respect to each state in turn and evaluating the result at the prediction* $\hat{\mathbf{x}}(k \mid k-1)$, *we obtain*

$$\nabla \mathbf{h}_x(k) = \begin{bmatrix} \frac{\hat{x}(k|k-1) - X_i}{d} & \frac{\hat{y}(k|k-1) - Y_i}{d} & 0 \\ -\frac{\hat{y}(k|k-1) - Y_i}{d^2} & \frac{\hat{x}(k|k-1) - X_i}{d^2} & -1 \end{bmatrix}, \tag{218}$$

where $d = \sqrt{(X_i - \hat{x}(k \mid k-1))^2 + (Y_i - \hat{y}(k \mid k-1))^2}$ *is the predicted distance between vehicle and beacon. For the sake of illustration we will assume that the prediction covariance matrix* $\mathbf{P}(k \mid k-1)$ *is diagonal (this will never be the case in practice), with* $\mathbf{P}(k \mid k-1) = diag\{\sigma_x^2, \sigma_y^2, \sigma_\phi^2, \}$, *and that the observation noise covariance is also diagonal* $\mathbf{R}(k) = diag\{r_r^2, r_\theta^2\}$. *The innovation covariance can now be computed from Equation 217 as*

$$\mathbf{S}(k) = \frac{1}{d^2}\begin{bmatrix} (X_i - \hat{x}(k \mid k-1))^2\sigma_x^2 + (Y_i - \hat{y}(k \mid k-1))^2\sigma_y^2 + r_r^2 & \\ (X_i - \hat{x}(k \mid k-1))(Y_i - \hat{y}(k \mid k-1))\left(\sigma_y^2 - \sigma_x^2\right)/d & \\ (X_i - \hat{x}(k \mid k-1))(Y_i - \hat{y}(k \mid k-1))\left(\sigma_y^2 - \sigma_x^2\right)/d & \\ (Y_i - \hat{y}(k \mid k-1))^2\sigma_x^2/d^2 + (X_i - \hat{x}(k \mid k-1))^2\sigma_y^2/d^2 + d^2(\sigma_\phi^2 + r_\theta^2) & \end{bmatrix} \tag{219}$$

It can be seen that the orientation error does not affect range innovation, only bearing innovation. It should be noted that this is only the case when the location to be estimated and the sensor location coincide.

7.4 Update Equations

We are interested in finding a recursive linear estimator $\hat{\mathbf{x}}(k \mid k)$ for the state $\mathbf{x}(k)$ which is described by the non-linear state transition of Equation 199 and which is being observed according to the non-linear observation model of Equation 200. We assume that a prediction $\hat{\mathbf{x}}(k \mid k-1)$ for the state at time k has been made on the basis of the first $k-1$ observations $\mathbf{Z}^{k-1}$ according to Equation 205 and that an observation $\mathbf{z}(k)$ has been made at this time. The estimator will be assumed to be in the form of an unbiased average of the prediction and innovation in the form

$$\hat{\mathbf{x}}(k \mid k) = \hat{\mathbf{x}}(k \mid k-1) + \mathbf{W}(k)\left[\mathbf{z}(k) - \mathbf{h}(\hat{\mathbf{x}}(k \mid k-1))\right]. \tag{220}$$

With the estimate given by Equation 220, it simply remains to select an appropriate gain matrix $\mathbf{W}(k)$. As in the case of linear system and observation models, we will choose the gain which minimizes conditional mean-squared estimation error.

The conditional mean-squared estimation error given in Equation 130 is simply equal to the trace of the estimate covariance $\mathbf{P}(k \mid k)$ which itself is simply the expected value of the state error $\tilde{\mathbf{x}}(k \mid k)$ squared. From Equation 220 and the approximate observation error given in Equation 216, the state error is given by

$$\begin{aligned}
\tilde{\mathbf{x}}(k \mid k) &= \hat{\mathbf{x}}(k \mid k) - \mathbf{x}(k) \\
&= [\hat{\mathbf{x}}(k \mid k-1) - \mathbf{x}(k)] + \mathbf{W}(k)\left[\mathbf{h}(\mathbf{x}(k)) - \mathbf{h}(\hat{\mathbf{x}}(k \mid k-1))\right] + \mathbf{W}(k)\mathbf{w}(k) \\
&\approx [\hat{\mathbf{x}}(k \mid k-1) - \mathbf{x}(k)] - \mathbf{W}(k)\nabla\mathbf{h}_x(k)\left[\hat{\mathbf{x}}(k \mid k-1) - \mathbf{x}(k)\right] + \mathbf{W}(k)\mathbf{w}(k) \\
&= \left[\mathbf{I} - \mathbf{W}(k)\nabla\mathbf{h}_x(k)\right]\tilde{\mathbf{x}}(k \mid k-1) + \mathbf{W}(k)\mathbf{w}(k)
\end{aligned} \tag{221}$$

Clearly the estimate is unbiased when the prediction is unbiased and the condition that the noise sequences are zero mean and white hold. Taking the expectation, condition on the observations made up to time k of the outer product of the state error gives an expression for the state covariance in terms of the prediction covariance

$$\begin{aligned}
\mathbf{P}(k \mid k) &\triangleq \mathrm{E}[\tilde{\mathbf{x}}(k \mid k)\tilde{\mathbf{x}}^T(k \mid k) \mid \mathbf{Z}^k] \\
&\approx \left[\mathbf{I} - \mathbf{W}(k)\nabla\mathbf{h}_x(k)\right]\mathrm{E}[\tilde{\mathbf{x}}(k \mid k-1)\tilde{\mathbf{x}}^T(k \mid k-1) \mid \mathbf{Z}^{k-1}]\left[\mathbf{I} - \mathbf{W}(k)\nabla\mathbf{h}_x(k)\right]^T \\
&\quad + \mathbf{W}(k)\mathrm{E}[\mathbf{w}(k)\mathbf{w}^T(k)]\,\mathbf{W}^T(k) \\
&\approx \left[\mathbf{I} - \mathbf{W}(k)\nabla\mathbf{h}_x(k)\right]\mathbf{P}(k \mid k-1)\left[\mathbf{I} - \mathbf{W}(k)\nabla\mathbf{h}_x(k)\right]^T + \mathbf{W}(k)\mathbf{R}(k)\mathbf{W}^T(k)
\end{aligned} \tag{222}$$

The gain matrix $\mathbf{W}(k)$ is now theorem to minimize mean squared estimation error

$$L(k) = \mathrm{E}[\tilde{\mathbf{x}}^T(k \mid k)\tilde{\mathbf{x}}(k \mid k)] = \mathrm{trace}[\mathbf{P}(k \mid k)]. \tag{223}$$

Following Equation 142 we have

$$\frac{\partial L}{\partial \mathbf{W}(k)} = -2(\mathbf{I} - \mathbf{W}(k)\nabla\mathbf{h}_x(k))\mathbf{P}(k \mid k-1)\nabla\mathbf{h}_x^T(k) + 2\mathbf{W}(k)\mathbf{R}(k) = \mathbf{0}. \tag{224}$$

Rearranging provides an expression for the gain matrix as

$$\begin{aligned} \mathbf{W}(k) &= \mathbf{P}(k \mid k-1)\nabla\mathbf{h}_x^T(k)\Big[\nabla\mathbf{h}_x(k)\mathbf{P}(k \mid k-1)\nabla\mathbf{h}_x^T(k) + \mathbf{R}(k)\Big]^{-1} \\ &= \mathbf{P}(k \mid k-1)\nabla\mathbf{h}_x^T(k)\mathbf{S}^{-1}(k), \end{aligned} \tag{225}$$

where the last line follows from Equation 217. With this gain matrix, Equation 220 becomes the best (minimum mean-squared error) linear unbiased estimator for the state $\mathbf{x}(k)$ under the stated conditions.

7.5 Summary

It is useful at this point to summarize the Extended Kalman Filter algorithm. We start with an estimate $\hat{\mathbf{x}}(k-1 \mid k-1)$ for the state $\mathbf{x}(k-1)$ at a time $k-1$ based on the observations $\mathbf{Z}^{k-1}$ made up to this time which is assumed approximately equal to the conditional mean or minimum mean-squared error estimate. The algorithm has two stages:

Prediction of State and Variance:

$$\hat{\mathbf{x}}(k \mid k-1) = \mathbf{f}\left(\hat{\mathbf{x}}(k-1 \mid k-1), \mathbf{u}(k)\right) \tag{226}$$

$$\mathbf{P}(k \mid k-1) = \nabla\mathbf{f}_x(k)\mathbf{P}(k-1 \mid k-1)\nabla\mathbf{f}_x^T(k) + \mathbf{Q}(k) \tag{227}$$

Update of State and Variance:

$$\hat{\mathbf{x}}(k \mid k) = \hat{\mathbf{x}}(k \mid k-1) + \mathbf{W}(k)\left[\mathbf{z}(k) - \mathbf{h}(\hat{\mathbf{x}}(k \mid k-1))\right] \tag{228}$$

$$\mathbf{P}(k \mid k) = \mathbf{P}(k \mid k-1) - \mathbf{W}(k)\mathbf{S}(k)\mathbf{W}^T(k) \tag{229}$$

where

$$\mathbf{W}(k) = \mathbf{P}(k \mid k-1)\nabla\mathbf{h}_x^T(k)\mathbf{S}^{-1}(k) \tag{230}$$

and

$$\mathbf{S}(k) = \nabla\mathbf{h}_x(k)\mathbf{P}(k \mid k-1)\nabla\mathbf{h}_x^T(k) + \mathbf{R}(k). \tag{231}$$

7.6 Understanding the Extended Kalman Filter

A comparison of Equations 144–148 with Equations 226–231 makes it clear that the extended Kalman filter algorithm is very similar to the linear Kalman filter algorithm, with the substitutions $\mathbf{F}(k) \rightarrow \nabla \mathbf{f}_x(k)$ and $\mathbf{H}(k) \rightarrow \nabla \mathbf{h}_x(k)$ being made in the equations for the variance and gain propagation. This similarity might have been expected on the grounds that the extended Kalman filter equations were derived by linearizing process and observation models at each time-step around the predicted state at that time. We know from Equation 70 and Equation 84 that this results in a linear stabilizing model for both the state error and observation error. Thus, the extended Kalman filter is, in effect, a linear estimator for a state error which is described by a *linear* equation of the form of Equation 70 and which is being observed according to a *linear* equation of the form of Equation 84.

The extended Kalman filter works in much the same way as the linear Kalman filter with some notable caveats:

- The Jacobians $\nabla \mathbf{f}_x(k)$ and $\nabla \mathbf{h}_x(k)$ are typically not constant, being functions of both state and timestep. This means that unlike the linear filter, the covariances and gain matrix must be computed on-line as estimates and predictions are made available, and will not in general tend to constant values. This significantly increase the amount of computation which must be performed on-line by the algorithm.

- As the linearised model is derived by perturbing the true state and observation models around a predicted or nominal trajectory, great care must be taken to ensure that these predictions are always 'close enough' to the true state that second order terms in the are indeed insignificant. If the nominal trajectory is too far away from the true trajectory then the true covariance will be much larger than the estimated covariance and the filter will become poorly matched. In extreme cases the filter may also become unstable.

- The extended Kalman filter employs a separating model which must be computed from an approximate knowledge of the state. Unlike the linear algorithm, this means that the filter must be accurately initialized at the start of operation to ensure that the separating models obtained are valid. If this is not done, the estimates computed by the filter will simply be meaningless.

The additional computation required by the extended Kalman filter was in the past seen as a significant problem and many ingenious schemes were devised to limit the calculations required in specific applications. However, increases in available real-time computing power are such that the computing resources required are now usually not a problem.

7.7 Implementation of Extended Kalman Filter

Linear approximations for non-linear functions should be treated with care. In particular, the application of the extended Kalman filter is fraught with danger and should only be attempted after a few stiff drinks. However, the extended Kalman filter has seen a huge variety of successful applications ranging from missile guidance to process plant control – so it can be made to work.

All the guidelines described for the linear Kalman filter apply, with twice the importance, to the extended Kalman Filter: In particular understand your sensor and understand your process; you will not get anywhere with a non-linear problem without doing this. Furthermore, the tests described for checking that the filter is performing correctly apply equally well to the extended Kalman filter. Tests based on the deficient of the innovation provide the main means of analysing filter performance. These tests however are complicated by the need to consider the changes in covariance, observation and state prediction caused by the dependence of the state model on the states themselves. In general, the testing of an extended Kalman filter requires the consideration of rather more cases than is required in the linear filter. A second important problem in non-linear problems is dealing with modeling errors in both the process model and the observation model. These are far more difficult to detect and to account for than in the linear case. Typically, the design of an extended Kalman filter relies far more on the development of a 'truth model'. Such a model is often derived from a detailed numerical analysis of a complete system (an air-frame for example), developed from system design codes. From this a model is built up and an 'error budget' determined in which the cost, in terms of estimation accuracy, is determined for various approximations to the 'true' model (see Maybeck [11] Chapter 6 for an excellent introduction to the problem of error budgeting) . Determining the pay-off between the detailed contained in the extended Kalman filter model and the resulting estimation accuracy is a difficult process and is necessarily application specific.

8 Example Implementation of the Extended Kalman Filter

This section provides a complete example implementation of a moderately complex extended Kalman Filter. The example is based on a land-mark based navigation system developed for an Autonomous Guided Vehicle (AGV) system able to transport ISO standard cargo containers in a port environment. A photograph of the complete vehicle with load is shown in Figure 26. The fully laden weight of the vehicle is 80 tonnes, and it travels at speeds up to 6m/s. The AGV navigation system is based on the use of a millimeter wave radar sensor detecting the range and bearing to a number of fixed known beacons located in the environment. A detailed description of this vehicle and the implementation of the system as a whole may be found in [7].

Figure 26: Photograph of completed AGV with 40 foot container load. One of the two radars can be seen at the front of the vehicle below the main deck.

This example builds on the extended Kalman filter examples developed so far in this course. It introduces two additional elements; the estimation of unmeasured states (specifically a wheel radius), and the use of non-linear error models. These features are of considerable value in many EKF problems.

The example begins by establishing and justifying process and observation models for this system. A simulation is then developed and the nominal performance analysed. The effects of initial conditions, variations in estimated errors and the insertion of fault conditions are then considered.

8.1 The Process Model

The process model consists of a description of the nominal motion of the vehicle and a model of the way in which uncertainties arise in this prediction. It provides essential information to predict vehicle location and for observing unmeasured vehicle parameters. The process model described here is appropriate for vehicles traveling at relatively high speeds on pneumatic tyres over rough terrain.

The error model developed for the vehicle makes explicit the fact that vehicle location uncertainty is predominantly caused by errors in drive velocity (slipping), steer traction (skidding) and changes in wheel radius. Further, the model explicitly accounts for the fact that these errors vary with vehicle state. As a consequence, the process error model does not follow the normal assumption used in the Kalman filter literature of being linearly additive. For this reason, the derivation of the process model and corresponding error model is accomplished in three stages. First a "nominal" process model is developed which describes vehicle motion in the absence of control, modeling and state errors. Second, this

model is used to see how errors in velocity, steer, wheel radius and previous state values are propagated through time. Finally, a linearised model of the error propagation equations are used to provide equations for state estimate covariance propagation.

8.1.1 Nominal Process Model

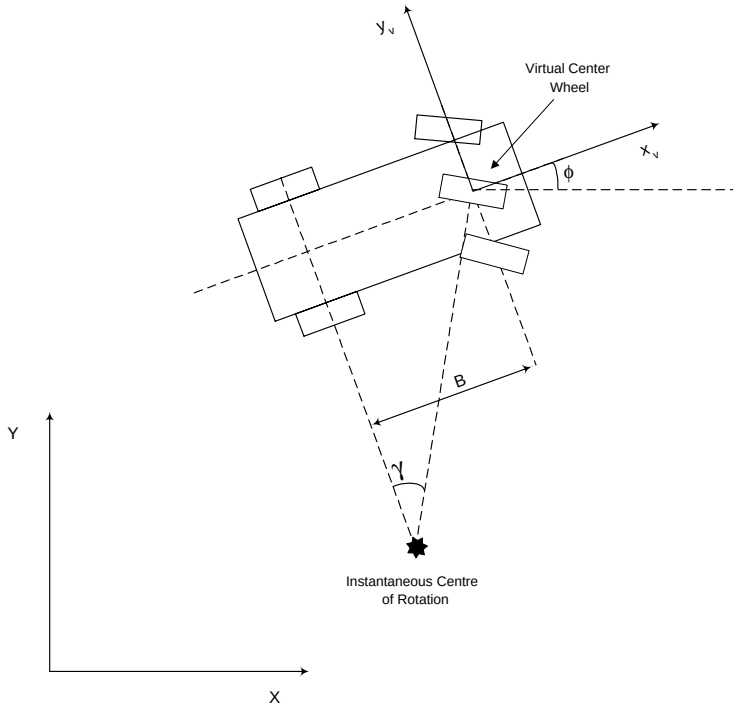


Figure 27: Kinematic arrangement of the example vehicle.

With reference to Figure 27, the nominal vehicle motion may be described by a simple "two-wheel bicycle" model, with a single steerable front wheel and a passive rear wheel. In this case, the vehicle model is given by

$$\begin{aligned} \dot{x}(t) &= R(t)\omega(t)\cos(\phi(t)+\gamma(t)) \\ \dot{y}(t) &= R(t)\omega(t)\sin(\phi(t)+\gamma(t)) \\ \dot{\phi}(t) &= \frac{R(t)\omega(t)}{B}(\sin\gamma(t)) \\ \dot{R}(t) &= 0, \end{aligned} \tag{232}$$

where the vehicle location $[x(t), y(t), \phi(t)]$ is referenced to the centre of the front axle. The control inputs are the steer angle γ, and the ground speed $V(t) = R(t)\omega(t)$ of the front wheel. The ground speed of the wheel is set equal to the rotational speed of the

wheel $\omega(t)$ (a measured quantity) multiplied by the wheel radius $R(t)$. The last equation makes explicit wheel radius variations.

The continuous-time model of Equation 232 is converted in to a discrete-time state transition equation. A synchronous sampling interval ΔT for both drive and steer encoders is assumed, all derivatives are approximated by first-order forward differences, all control signals ($\omega(t)$, $\gamma_f(t)$ and $\gamma_r(t)$) are assumed approximately constant over the sample period, and all continuous times are replaced with a discrete time index as $t = k\Delta T \triangleq k$. Equation 232 then becomes

$$\begin{aligned} x(k+1) &= x(k) + \Delta T R(k)\omega(k)\cos\left[\phi(k)+\gamma(k)\right] \\ y(k+1) &= y(k) + \Delta T R(k)\omega(k)\sin\left[\phi(k)+\gamma(k)\right] \\ \phi(k+1) &= \phi(k) + \Delta T \frac{R(k)\omega(k)}{B}\sin\gamma(k) \\ R(k+1) &= R(k). \end{aligned} \tag{233}$$

The state vector at a time k is then defined as

$$\mathbf{x}(k) = [x(k), y(k), \phi(k), R(k)]^T,$$

the control vector as

$$\mathbf{u}(k) = [\omega(k), \gamma(k)]^T$$

and the nominal (error-free) state transition as

$$\mathbf{x}(k+1) = \mathbf{f}\left(\mathbf{x}(k), \mathbf{u}(k)\right), \tag{234}$$

where the transition function $\mathbf{f}(\cdot)$ is defined in Equation 233. This defines the nominal process model.

During operation, the true vehicle state $\mathbf{x}(k)$ will never be known. Instead, an estimate of the state is computed through beacon observation and knowledge of the drive and encoder signals. We follow the notation developed in [8] for the remainder of this example to minimise the use of bracketed time scripts. Let

$$\hat{\mathbf{x}}^+(k) = [\hat{x}^+(k), \hat{y}^+(k), \hat{\phi}^+(k), \hat{R}^+(k)]^T$$

denote the estimate made of the state $\mathbf{x}(k)$ at time k based on all observations and control knowledge up to time k. Further, let

$$\overline{\mathbf{u}}(k) = [\overline{\omega}(k), \overline{\gamma}(k),]^T$$

be the mean measured (from encoders) value of the true control vector $\mathbf{u}(k)$.

Equation 234 may be used to generate a prediction $\hat{\mathbf{x}}^-(k+1)$,

$$\hat{\mathbf{x}}^-(k+1) = [\hat{x}^-(k+1), \hat{y}^-(k+1), \hat{\phi}^-(k+1), \hat{R}^-(k+1)]^T,$$

of the true state $\mathbf{x}(k+1)$ at time $k+1$ as

$$\hat{\mathbf{x}}^-(k+1) = \mathbf{f}\left(\hat{\mathbf{x}}^+(k), \overline{\mathbf{u}}(k)\right). \tag{235}$$

8.1.2 Process Error Model

The prediction stage of the extended Kalman filter also requires that the manner in which errors are injected into the system and the way in which these errors are propagated in time be accurately modeled. Errors are injected into the AGV system by three primary sources; forward drive signals, steer angle signals and changes in wheel radius. The forward drive error is modelled as a combination of additive disturbance error $\delta\omega(k)$ and multiplicative slip error $\delta q(k)$:

$$\omega(k) = \overline{\omega}(k)\,[1 + \delta q(k)] + \delta\omega(k),$$

where $\overline{\omega}(k)$is taken to be the mean measured wheel rotation rate as recorded by the vehicle encoders, and $\omega(k)$ is defined to be the true mean wheel rotation rate defined through Equation 233. The steer drive error is similarly modelled as a combination of an additive disturbance error $\delta\gamma(k)$ and a multiplicative skid error $\delta s(k)$:

$$\gamma(k) = \overline{\gamma}(k)\,[1 + \delta s(k)] + \delta\gamma(k),$$

where $\overline{\gamma}(k)$is taken to be the mean measured axle steer angle as recorded by the steer encoders, with $\gamma(k)$ defined to be the true mean axle steer angle defined through Equation 233. The error in wheel radius is modelled as a discrete additive disturbance rate error (a random walk) so that

$$R(k) = \hat{R}^{+}(k) + \Delta T \delta R(k).$$

The source errors $\delta q(k)$, $\delta\omega(k)$, $\delta s(k)$, $\delta\gamma(k)$, and $\delta R(k)$ are modeled as constant, zero mean, uncorrelated white sequences, with variances σ_q^2, σ_ω^2, σ_s^2, σ_γ^2 and σ_R^2 respectively.

The error models for forward drive and steer signals are designed to reflect two important features. First, the multiplicative component of the error models reflect the increased uncertainty in vehicle motion as speed and steer angles increase (slipping and skidding)[6]. Second, the additive component of the error models is designed to reflect both stationary uncertainty and motion model errors such as axle offsets. The additive error is also important to stabilize the estimator algorithm. The random walk model for wheel radius is intended to allow adaptation of the estimator to wheel radius changes caused by uneven terrain and by changes in vehicle load (the chassis and wheels deform substantially at loads of 30-60 tonnes). In theory, such changes are best modeled as abrupt variations in parameter values. In practice such models turned out to be very difficult to implement. The random-walk model was found to accommodate typical variations in a sensible manner.

[6]Fundamentally, these errors are actually due to linear and rotational inertial forces acting at the interface between tyre and road. A model that explicitly incorporate these effects is described in [10].

8.1.3 Error Propagation Equations

To develop the error model, it is necessary to consider how errors in the state estimate and knowledge of control input feed through Equation 233 to generate prediction errors. The error between the true state and estimated state, and between the true state and the prediction are given by

$$\delta\mathbf{x}^+(k) = \mathbf{x}(k) - \hat{\mathbf{x}}^+(k), \text{ and} \qquad \delta\mathbf{x}^-(k+1) = \mathbf{x}(k+1) - \hat{\mathbf{x}}^-(k+1), \tag{236}$$

respectively, and the difference between true and measured control input is denoted by

$$\delta\mathbf{u}(k) = \mathbf{u}(k) - \overline{\mathbf{u}}(k). \tag{237}$$

With these definitions, together with Equations 234 and 235, we have

$$\begin{aligned} \delta\hat{\mathbf{x}}^-(k+1) &= \mathbf{x}(k+1) - \hat{\mathbf{x}}^-(k+1) \\ &= \mathbf{f}\left(\mathbf{x}(k), \mathbf{u}(k)\right) - \mathbf{f}\left(\hat{\mathbf{x}}^+(k), \overline{\mathbf{u}}(k)\right) \\ &= \mathbf{f}\left(\hat{\mathbf{x}}^+(k) + \delta\mathbf{x}^+(k), \overline{\mathbf{u}}(k) + \delta\mathbf{u}(k)\right) - \mathbf{f}\left(\hat{\mathbf{x}}^+(k), \overline{\mathbf{u}}(k)\right). \end{aligned} \tag{238}$$

Evaluating this expression using Equation 233, and neglecting all second-order error products, we obtain an error transfer model as

$$\begin{aligned} \delta\hat{x}^-(k+1) = \delta\hat{x}^+(k) &+ \Delta T \cos\left(\hat{\phi}^+(k) + \overline{\gamma}(k)\right)\left[\delta\Omega(k) + \overline{\omega}(k)\delta R(k)\right] \\ &- \Delta T \sin\left(\hat{\phi}^+(k) + \overline{\gamma}(k)\right)\left[\delta\Gamma(k) + \hat{R}^+(k)\overline{\omega}(k)\delta\phi(k)\right] \\ \delta\hat{y}^-(k+1) = \delta\hat{y}^+(k) &+ \Delta T \sin\left(\hat{\phi}^+(k) + \overline{\gamma}(k)\right)\left[\delta\Omega(k) + \overline{\omega}(k)\delta R(k)\right] \\ &+ \Delta T \cos\left(\hat{\phi}^+(k) + \overline{\gamma}(k)\right)\left[\delta\Gamma(k) + \hat{R}^+(k)\overline{\omega}(k)\delta\phi(k)\right] \\ \delta\hat{\phi}^-(k+1) = \delta\hat{\phi}^+(k) &+ \Delta T \frac{\sin\overline{\gamma}(k)}{B}\left[\delta\Omega(k) + \overline{\omega}(k)\delta R(k)\right] \\ &+ \Delta T \frac{\cos\overline{\gamma}(k)}{B}\delta\Gamma(k) \\ \delta\hat{R}^-(k+1) = \delta\hat{R}^+(k) &+ \Delta T\delta R(k) \end{aligned} \tag{239}$$

where

$$\delta\Omega(k) = \hat{R}^+(k)\overline{\omega}(k)\delta q(k) + \hat{R}^+(k)\delta\omega(k) \tag{240}$$

is the composite along-track rate error describing control induced error propagation along the direction of travel, and

$$\delta\Gamma(k) = \hat{R}^+(k)\overline{\omega}(k)\overline{\gamma}(k)\delta s(k) + \hat{R}^+(k)\overline{\omega}(k)\delta\gamma(k) \tag{241}$$

is the composite cross-track rate error describing control induced error propagation perpendicular to the direction of travel.

These error transfer equations may be written in state-vector notation as follows; define the state error transfer matrix as

$$\mathbf{F}(k) = \begin{bmatrix} 1 & 0 & -\Delta T \hat{R}^+(k)\overline{\omega}(k)\sin(\hat{\phi}^+(k)+\overline{\gamma}(k)) & \Delta T\overline{\omega}(k)\cos(\hat{\phi}^+(k)+\overline{\gamma}(k)) \\ 0 & 1 & \Delta T \hat{R}^+(k)\overline{\omega}(k)\cos(\hat{\phi}^+(k)+\overline{\gamma}(k)) & \Delta T\overline{\omega}(k)\sin(\hat{\phi}^+(k)+\overline{\gamma}(k)) \\ 0 & 0 & 1 & \Delta T\overline{\omega}(k)\frac{\sin\overline{\gamma}(k)}{B} \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

and the source error transfer matrix as

$$\mathbf{G}(k) = \begin{bmatrix} \cos(\hat{\phi}^+(k)+\overline{\gamma}(k)) & -\sin(\hat{\phi}^+(k)+\overline{\gamma}(k)) & 0 \\ \sin(\hat{\phi}^+(k)+\overline{\gamma}(k)) & \cos(\hat{\phi}^+(k)+\overline{\gamma}(k)) & 0 \\ \frac{\sin\overline{\gamma}(k)}{B} & \frac{\cos\overline{\gamma}(k)}{B} & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

With $\delta\mathbf{w}(k) = [\delta\Omega(k), \delta\Gamma(k), \delta R(k)]^T$, the error transfer Equations now become

$$\delta\mathbf{x}^-(k+1) = \mathbf{F}(k)\delta\mathbf{x}^+(k) + \Delta T\mathbf{G}(k)\delta\mathbf{w}(k). \tag{242}$$

Define

$$\begin{aligned} \mathbf{P}^-(k+1) &= \mathrm{E}\left[\delta\mathbf{x}^-(k+1)\delta\mathbf{x}^-(k+1)^T\right], \\ \mathbf{P}^+(k) &= \mathrm{E}\left[\delta\mathbf{x}^+(k)\delta\mathbf{x}^+(k)^T\right], \\ \mathbf{\Sigma}(k) &= \mathrm{E}\left[\delta\mathbf{w}(k)\delta\mathbf{w}(k)^T\right], \end{aligned}$$

and assume that $\mathrm{E}[\delta\mathbf{x}^+(k)\delta\mathbf{w}(k)^T] = \mathbf{0}$. Squaring Equation 242, and taking expectations then gives an equation for the propagation of covariance information as

$$\mathbf{P}^-(k+1) = \mathbf{F}(k)\mathbf{P}^+(k)\mathbf{F}^T(k) + \Delta T^2\mathbf{G}(k)\mathbf{\Sigma}(k)\mathbf{G}^T(k), \tag{243}$$

where, on the assumption that the source errors are uncorrelated, $\mathbf{\Sigma}(k)$ is given by

$$\mathbf{\Sigma}(k) = \begin{bmatrix} \left[\hat{R}^+(k)\right]^2\left([\overline{\omega}(k)]^2\sigma_q^2 + \sigma_\omega^2\right) & 0 & 0 \\ 0 & \left[\hat{R}^+(k)\overline{\omega}(k)\right]^2\left(\overline{\gamma}(k)^2\sigma_s^2 + \sigma_\gamma^2\right) & 0 \\ 0 & 0 & \sigma_R^2 \end{bmatrix}. \tag{244}$$

8.2 Observation Model

After predicting the vehicle location, the next step in the navigation process is to take an observation and to combine this information together with the prediction to produce an updated estimate of the vehicle location. The essential observation information used by the radar consists of measurements of range and bearing made by the radar units to a number of beacons placed at fixed and known locations in the environment. The basic layout of the observation process is shown in Figure 28.

Processing of observations occurs in four stages.

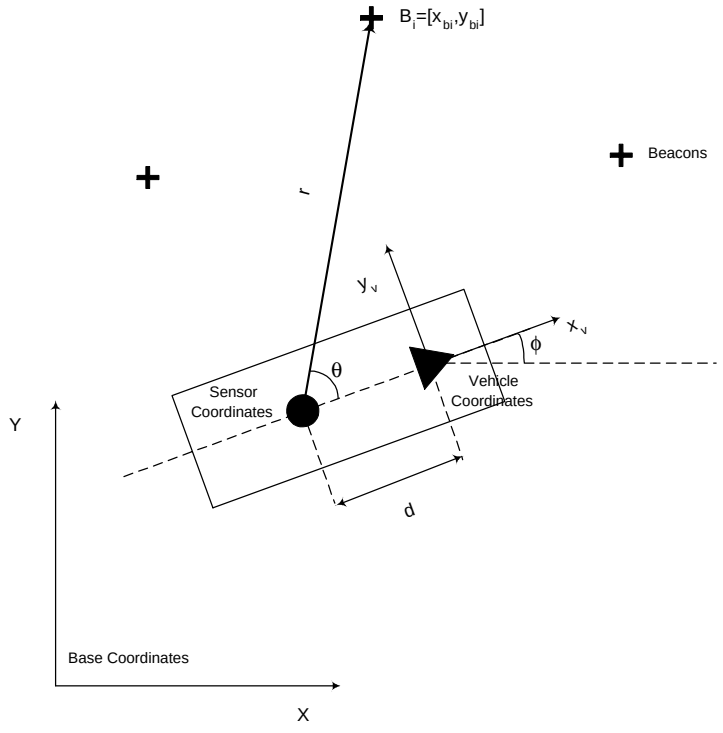


Figure 28: Kinematic arrangement of the example vehicle.

1. The measurement is converted into a Cartesian observation referenced to the vehicle coordinate system.

2. The vehicle-centered observation is transformed into base-coordinates using knowledge of the predicted vehicle location at the time the observation was obtained.

3. The observation is then matched to a map of beacons maintained by the AGV in base-coordinates.

4. The matched beacon is transformed back into a vehicle centered coordinate system where it is used to update vehicle location according to the standard extended Kalman filter equations.

In the following, it is assumed that the measurements are taken at a discrete time instant k when a prediction of vehicle location is already available. The true asynchronicity of the measurement and prediction cycle is considered elsewhere. Further, extensive use of the following standard result is made: If two random variables $\mathbf{a}$ and $\mathbf{b}$ are related by the non-linear equation $\mathbf{a} = \mathbf{g}(\mathbf{b})$, then the mean $\overline{\mathbf{a}}$ of $\mathbf{a}$ may be approximated in terms of the mean $\overline{\mathbf{b}}$ of $\mathbf{b}$ by

$$\overline{\mathbf{a}} = \mathbf{g}\left(\overline{\mathbf{b}}\right) \tag{245}$$

and that the variance $\mathbf{\Sigma}_a$ of $\mathbf{a}$ may be approximated in terms of the variance $\mathbf{\Sigma}_b$ of $\mathbf{b}$ by

$$\mathbf{\Sigma}_a = \nabla \mathbf{g}_b \mathbf{\Sigma}_b \nabla \mathbf{g}_b^T, \tag{246}$$

where $\nabla \mathbf{g}_b$ is the Jacobian of $\mathbf{g}(\cdot)$ taken with respect to $\mathbf{b}$, evaluated at the mean $\overline{\mathbf{b}}$.

8.2.1 Observation Processing and Matching

The four processing stages are now detailed:

Stage 1: The radar provides observations of range $r(k)$ and bearing $\theta(k)$ to a fixed target in the environment as shown in Figure 28. The radar itself is located on the centerline of the vehicle with a longitudinal offset d from the vehicle centered coordinate system. The observations $\mathbf{z}_v(k)$, in Cartesian coordinates, referred to the vehicle frame are given by

$$\mathbf{z}_v(k) = \begin{bmatrix} z_{xv}(k) \\ z_{yv}(k) \end{bmatrix} = \begin{bmatrix} d + r(k)\cos\theta(k) \\ r(k)\sin\theta(k) \end{bmatrix} \tag{247}$$

We assume that the errors in range and bearing may be modeled as a Gaussian uncorrelated white sequence with constant variances σ_r^2 and σ_θ^2 respectively. The observation variance $\mathbf{\Sigma}_z(k)$ in vehicle coordinates may be obtained by applying Equation 246 to Equation 247 to obtain

$$\mathbf{\Sigma}_z(k) = \begin{bmatrix} \cos\theta(k) & -\sin\theta(k) \\ \sin\theta(k) & \cos\theta(k) \end{bmatrix} \begin{bmatrix} \sigma_r^2 & 0 \\ 0 & r^2\sigma_\theta^2 \end{bmatrix} \begin{bmatrix} \cos\theta(k) & \sin\theta(k) \\ -\sin\theta(k) & \cos\theta(k) \end{bmatrix} \tag{248}$$

Stage 2: The observation, defined in vehicle coordinates, now needs to be transformed into absolute world coordinates so that it can be matched to the map of beacon locations. The map of beacon locations must be maintained in a fixed coordinate system otherwise every element of the map would have to be transformed into vehicle coordinates before a decision on a best match can be obtained. If the current predicted vehicle location in base coordinates is $[\hat{x}^-(k), \hat{y}^-(k), \hat{\phi}^-(k)]^T$, then from Figure 28, the observation $\mathbf{z}_b(k)$ in cartesian base coordinates is given by

$$\mathbf{z}_b(k) = \begin{bmatrix} z_{xb}(k) \\ z_{yb}(k) \end{bmatrix} = \begin{bmatrix} \hat{x}^-(k) + z_{xv}(k)\cos\hat{\phi}^-(k) - z_{yv}(k)\sin\hat{\phi}^-(k) \\ \hat{y}^-(k) + z_{xv}(k)\sin\hat{\phi}^-(k) + z_{yv}(k)\cos\hat{\phi}^-(k) \end{bmatrix} \tag{249}$$

The observation variance may also be transformed into base coordinates by applying Equation 246 to Equation 249 (with respect to both $\mathbf{x}(k)$ and $\mathbf{z}_v(k)$), to give

$$\mathbf{\Sigma}_b(k) = \mathbf{T}_x(k)\mathbf{P}^-(k)\mathbf{T}_x^T(k) + \mathbf{T}_z(k)\mathbf{\Sigma}_z(k)\mathbf{T}_z^T(k) \tag{250}$$

where

$$\mathbf{T}_x(k) = \begin{bmatrix} 1 & 0 & -z_{xv}(k)\sin\hat{\phi}^-(k) - z_{yv}(k)\cos\hat{\phi}^-(k) & 0 \\ 0 & 1 & -z_{xv}(k)\cos\hat{\phi}^-(k) - z_{yv}(k)\sin\hat{\phi}^-(k) & 0 \end{bmatrix}, \tag{251}$$

and

$$\mathbf{T}_z(k) = \begin{bmatrix} \cos\hat{\phi}^-(k) & -\sin\hat{\phi}^-(k) \\ \sin\hat{\phi}^-(k) & \cos\hat{\phi}^-(k) \end{bmatrix}, \tag{252}$$

and where $\mathbf{P}^-(k)$ is the predicted vehicle state covariance.

Stage 3: With the observation and observation covariance now in base coordinates, it is possible to match the measurement with the beacon map. In typical AGV applications, the beacons are surveyed into the site using a theodolite, and so it is assumed that no error is associated with their location. The beacon locations $\mathbf{b}_i = [x_{bi}, y_{bi}], i = 1, \cdots, N$ are loaded into the AGV at run time. The matching process (more generally data association) is a complex problem. Issues such as clutter rejection, false detections, matching complexity all need to be addressed. However, given *a priori* knowledge of beacon locations and a high signal to noise ratio, a simple validation gate (normalised innovation) may be used to associate observations to beacons in the following form:

$$(\mathbf{b}_i - \mathbf{z}_b(k))^T \, \mathbf{\Sigma}_b^{-1}(k) \, (\mathbf{b}_i - \mathbf{z}_b(k)) < \alpha. \tag{253}$$

The gate size α is normally taken to be quite small (0.5) to ensure low false alarm rates.

Stage 4: Once a correct association with a single beacon $\mathbf{b} = [x_b, y_b]^T$ has been made, a vehicle centered observation prediction needs to be generated so that the vehicle location may be updated. The reason why this update should not done in base coordinates is because the location estimates become very sensitive to approximations in orientation uncertainty in this coordinate system. Vehicle centered observations can be generated by transforming the matched beacon from base coordinates back through the predicted vehicle location to the estimated vehicle centered coordinate system according to

$$\hat{\mathbf{z}}_v = \begin{bmatrix} \hat{z}_{vx} \\ \hat{z}_{vy} \end{bmatrix} = \begin{bmatrix} \cos\hat{\phi}^-(k) & \sin\hat{\phi}^-(k) \\ -\sin\hat{\phi}^-(k) & \cos\hat{\phi}^-(k) \end{bmatrix} \begin{bmatrix} x_b - \hat{x}^-(k) \\ y_b - \hat{y}^-(k) \end{bmatrix} \tag{254}$$

8.2.2 Vehicle Update

The vehicle estimate may now be updated using the usual Kalman filter equations. The updated state estimate is computed from

$$\hat{x}^+(k) = \hat{x}^-(k) + \mathbf{W}(k) \left[\mathbf{z}_v(k) - \hat{\mathbf{z}}_v \right], \tag{255}$$

and the covariance from

$$\mathbf{P}^+(k) = \mathbf{P}^-(k) - \mathbf{W}(k)\mathbf{S}(k)\mathbf{W}^T(k), \tag{256}$$

where the gain matrix is given by

$$\mathbf{W}(k) = \mathbf{P}^-(k)\mathbf{H}^T(k)\mathbf{S}^{-1}(k), \tag{257}$$

the innovation covariance by

$$\mathbf{S}(k) = \mathbf{H}(k)\mathbf{P}^{-}(k)\mathbf{H}^{T}(k) + \mathbf{\Sigma}_z(k), \tag{258}$$

and where

$$\mathbf{H}(k) = \begin{bmatrix} -\cos\hat{\phi}^{-}(k) & -\sin\hat{\phi}^{-}(k) & -(x_b - \hat{x}^{-}(k))\sin\hat{\phi}^{-}(k) + (y_b - \hat{y}^{-}(k))\cos\hat{\phi}^{-}(k) & 0 \\ \sin\hat{\phi}^{-}(k) & -\cos\hat{\phi}^{-}(k) & -(x_b - \hat{x}^{-}(k))\cos\hat{\phi}^{-}(k) - (y_b - \hat{y}^{-}(k))\sin\hat{\phi}^{-}(k) & 0 \end{bmatrix} \tag{259}$$

is the Jacobian of the transform matrix given in Equation 254, taken with respect to the state and evaluated at the prediction as required in Equation 246.

8.3 System Analysis

In this section we describe the implementation of this navigation system as a detailed example of an extended Kalman filter.

The code is implemented in Matlab and consists of two main parts;

1. The specification of a vehicle path, followed by the generation of true vehicle trajectory, true vehicle control inputs and simulated observations of a number of beacons.

2. The filtering of observations with control inputs in an extended Kalman filter to provide estimates of vehicle position, heading and mean wheel radius, together with associated estimation errors.

This structure allows a single vehicle run to be generated and subsequently to evaluate the effect of different filter parameters, initial conditions and injection errors.

8.3.1 Trajectory Generation

The vehicle trajectory is generated in the following stages:

1. A number of spline points are defined. At the same time the beacon locations are defined.

2. A smooth spline curve is fitted to these points.

3. A velocity profile for the vehicle is defined.

4. A proportional control algorithm is used to control the steering angle to maintain the vehicle on-path.

5. The resulting true vehicle location is recorded as it follows the path, together with the steer and speed control inputs.

A typical trajectory (and the one used in the remainder of this section) together with beacon locations is shown in Figure 29. The true steer angle, as a function of time, generated in following this path is shown in Figure 30. The true velocity of the vehicle was set at 3m/s. The plot of steer angle is a useful time history of the trajectory. It shows the vehicle follows a relatively smooth path except at a few sharp steer points around the locations (150,100) (at 90 seconds), and (200,250) (at 160 seconds) in the path.

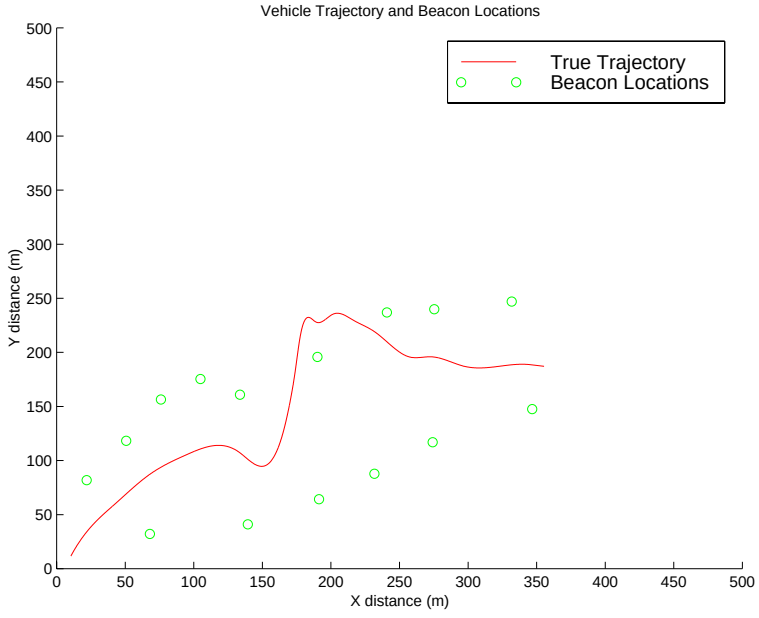


Figure 29: True vehicle trajectory and beacon locations for the example navigation system.

8.3.2 Observation Generation

Once the true path of the vehicle is known, simulated observations of beacons taken by a sensor on the vehicle can be easily generated. In this example, the simulated sensor is mechanically scanned through 360^o at a fixed scan rate of 2Hz. A simple intersection algorithm is then used to determine if a beacon is detected in any given time period. The corresponding relative range and bearing to the beacon from the vehicle is returned and a simulated additive observation error is incorporated. Figure 31 shows an example set of simulated observations referenced to the true vehicle trajectory. The beacon detection rate is approximately 8Hz.

8.3.3 Filter Structure and Initialisation

The navigation structure follows the theoretical development described in previous sections. The filter has three main stages: Prediction, Observation Matching, and Update.

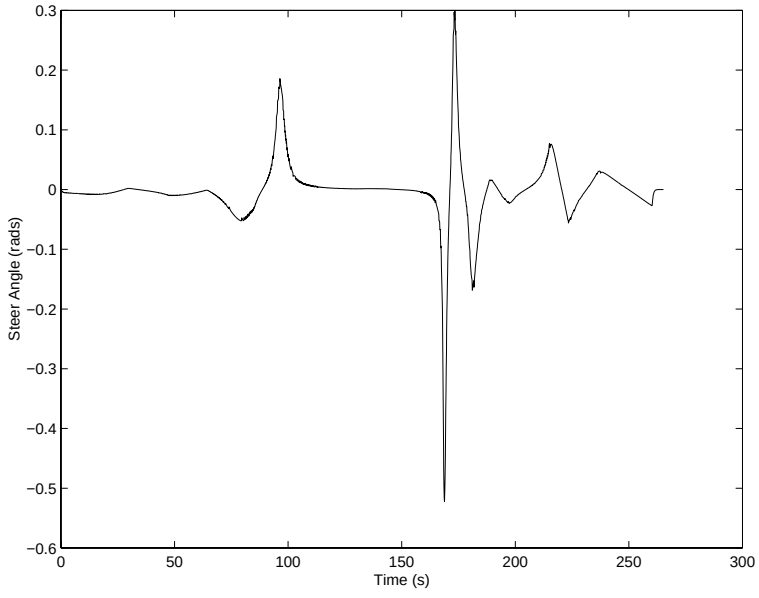


Figure 30: True steer angle of the vehicle as a function of time.

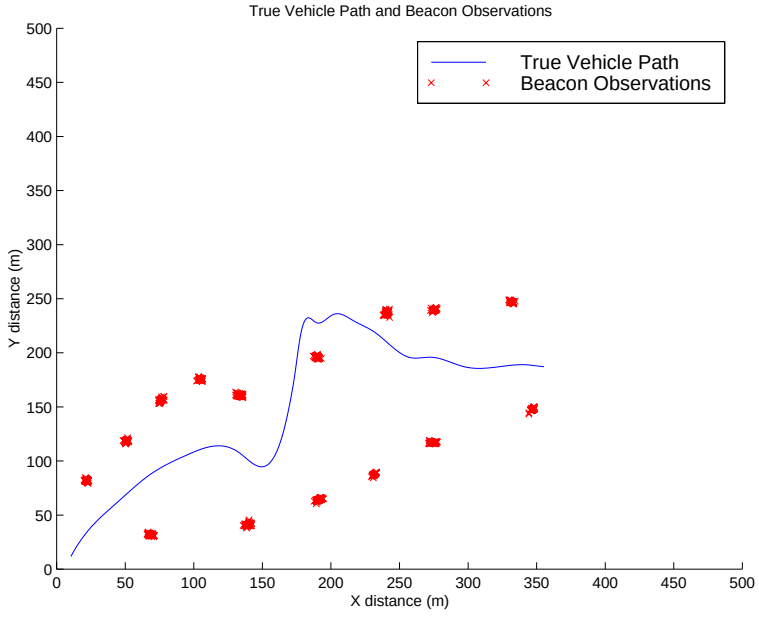


Figure 31: Generated beacon observations, referenced to true vehicle trajectory.

Predictions are made synchronously at 10Hz on the basis of true (measured) steer and drive control inputs. If there is no beacon observation made in the sample time, then the prediction becomes the estimate at that time. If an observation is made, it is matched to one of the beacons in the initial map and a filter update is performed at that time.

The nominal estimated values for the error source terms were defined as follows:

Multiplicative Slip Error σ_q (%/100):	0.02
Additive Slip Error σ_ω (rads/s):	0.1
Multiplicative Skid Error σ_s (%/100):	0.01
Additive Skid Error σ_γ (rads):	0.035
Wheel Radius Error Rate σ_R (m/s):	0.001
Radar Range Error σ_r (m):	0.3
Radar Bearing Error σ_θ (rads):	0.035

The performance of the vehicle navigation system is relatively insensitive to specific values of process model errors up to a factor of 2-4, while being quite sensitive to estimated observation errors.

The navigation is initialised with the initial true vehicle location. The initial position errors are taken as

$$\sqrt{\mathbf{P}(0 \mid 0)} = \begin{bmatrix} 0.3 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.3 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.05 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.01 \end{bmatrix}$$

8.3.4 Nominal Filter Performance

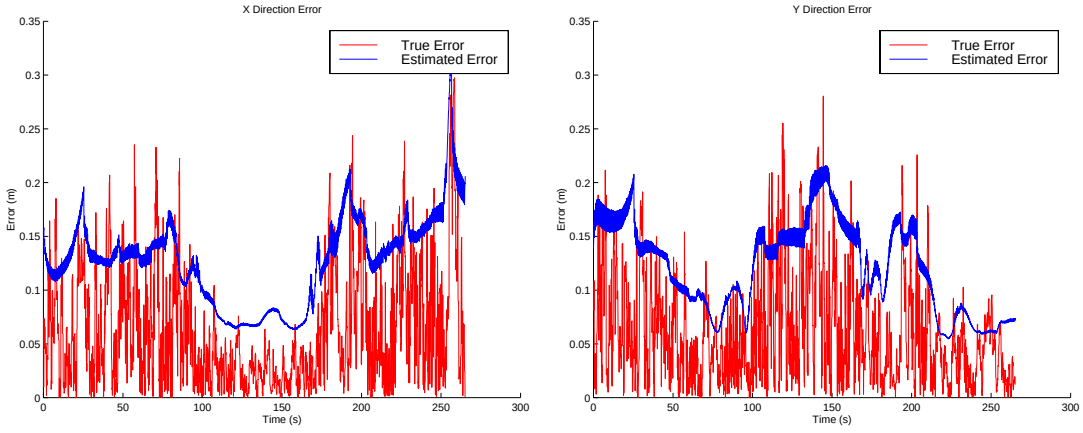
Once initialised, the filter is run with the observations and control signals generated by the simulator. The resulting estimates are best described in terms of a plot of the true state errors[7] and estimated state errors. The true state error is given by

$$\delta\mathbf{x}^+(k) = \mathbf{x}(k) - \hat{\mathbf{x}}^+(k)$$

and the estimated state error by

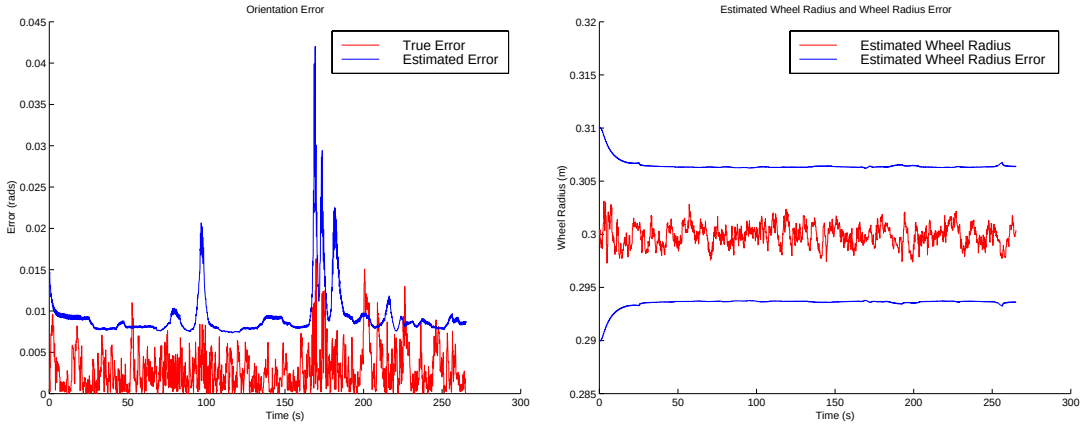
$$\delta\hat{\mathbf{x}^+}(k) = \sqrt{\text{diag}(\mathbf{P}(k \mid k))}.$$

Figure 32 shows the true and estimated errors in vehicle location (x and y) estimates as a function of time. Figure 33 shows the true and estimated errors in vehicle orientation and wheel radius estimates as a function of time.



(a) (b)

Figure 32: Actual and estimated vehicle position errors in (a) x-direction and (b) y-direction. Actual error (the error between true vehicle path and estimated vehicle path) is not normally available in a real system. The estimated error (standard deviation or square-root of covariance) in vehicle path is generated by the filter.



(a) (b)

Figure 33: (a) Actual and estimated vehicle orientation error. (b) Estimated vehicle mean wheel radius with 1-σ estimated error bounds.

The first clear difference between these plots and those corresponding to a linear system is that the estimated errors are not constant and do not reach a steady-state value. This is because the state transition matrix, control input matrix and observation model matrix are all Jacobians whose specific value depends on the state at the time of evaluation.

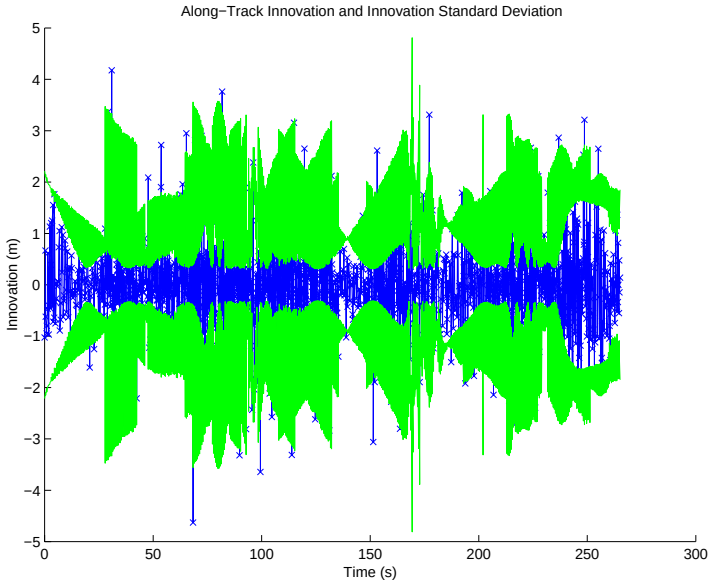
The position error plots show that the filter is well matched with at least 60% of actual position errors falling within the estimated (1-σ) error bound. The position errors vary substantially over the run. The main variation occurs between 100 and 200 seconds. This is caused by the vehicle changing orientation: Recall that the errors injected in to the system consist of along-track and cross-track errors, with long-travel (the direction of travel) errors logically being larger than cross-track (side slip) errors. In the position error plots, the vehicle changes orientation from approximately 45^o to 90^o during the period 100-200 seconds. Consequently, the errors in x and y are approximately equal before 100 seconds, but between 100-200 seconds, the x error is predominantly due to lateral errors, whereas the y errors are predominantly due to (larger) longitudinal errors. Secondary variations in positional errors (at 260 seconds for example) are caused by changes in the locally observable geometry of the beacons. When a large number of near beacons are observable, the position error decreases. If few beacons, or only distant beacons, are observable, the position error grows.

The plot of orientation error also shows a non-constant behaviour. The obvious feature in these plots are the sudden spikes in estimated orientation error at 100 and 160-180 seconds. These correspond to the times at which the vehicle is steering sharply. At these points, the multiplicative steering noise model makes a significant contribution to estimated state error. As the steer angle increases, so the estimated noise, due to steering, injected into the filter also increases. This is exactly the behaviour required of the filter as the increase in estimated error accommodates actual error experienced by the vehicle state estimator. Such multiplicative models are thus an effective way of capturing errors due to slipping and skidding of high-speed vehicles.

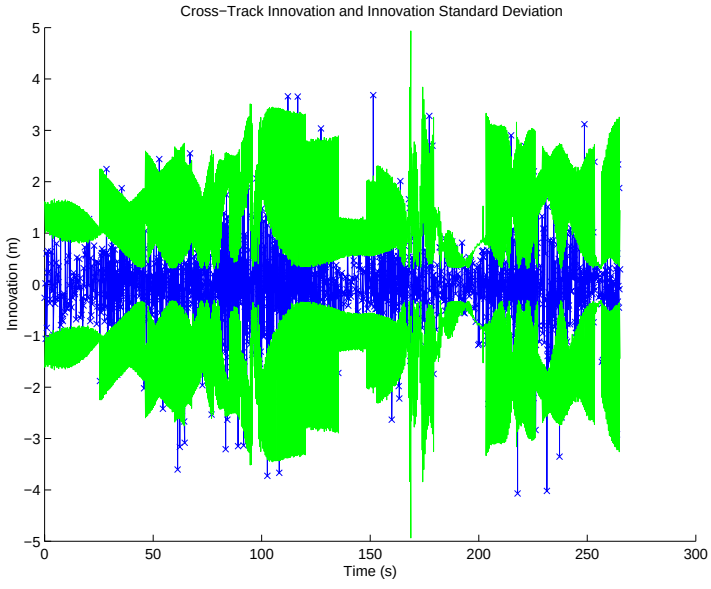
The plot of wheel radius error shows a much more constant behaviour. As errors in wheel radius feed through into vehicle position error, so observation of position while the vehicle is in motion allows estimation (observability) of the wheel radius. Changes in wheel radius model long-travel slip as well as variations in terrain. In the example shown, for constant velocity motion, there is little slip and so the estimated wheel radius closely follows true wheel radius.

Figure 34 shows the innovations and innovation standard deviations for the estimator in along-track and cross-track coordinates. The innovations are the difference between predicted observations and the observations actually made. In real applications, they are the primary means of determining the correct operation of the filter. In this example, observations are made of range and bearing to fixed beacons. These can be converted in to x and y coordinates in any convenient reference frame. For navigation problems,

[7]It should be noted that in any real application, the true state errors are never made available as the true state is never known. Filter tuning must, as always, use only the filter innovations.



(*a*)



(*b*)

Figure 34: Filter innovations and innovation standard deviations in (a) Along-track direction and (b) Cross-Track direction.

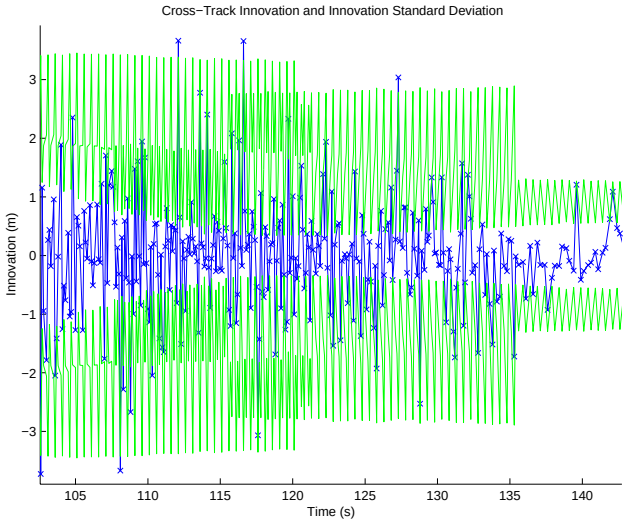


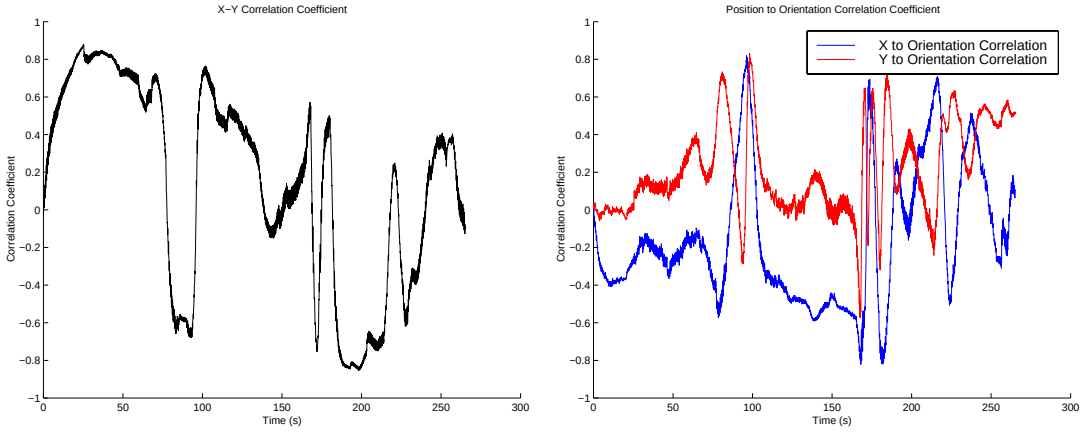
Figure 35: Detail of Cross-track innovations.

the most useful frame is one that is aligned with the direction of motion of the vehicle, in which innovations are aligned along-track and cross-track. The reason for this is that the estimated error rates for the vehicle are also best described in this coordinate system. Consequently, to tune estimate vehicle errors, the along-track and cross-track innovations are the most helpful.

The innovations show a filter that is clearly very well tuned with most innovations falling within the 1-*sigma* standard deviation. The innovations and innovation standard deviations show substantial variation over time. The periodic (triangular) rise and fall of the innovation standard deviations is due to specific beacons coming into view, the vehicle approaching these beacons (standard deviations reducing), and then moving away from the beacon (standard deviations increasing). Periods from many different beacons are seen together in the time history of the two innovation plots. This behaviour is made even clearer in Figure 35 which shows a detailed view of the cross-track innovation. This clearly shows a number of different beacons being observed; the vehicle is receding from some of these (standard deviation increasing), and approaching others (standard deviation decreasing). The composite effect of seeing many different beacons is an apparent high frequency variation in all estimated state errors. The innovation plots also give a picture of the relative information contribution of each beacon: Beacons with low information value result in large estimated innovation standard deviation.

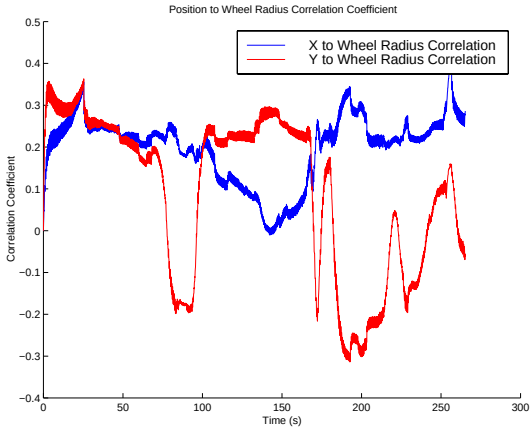
Figure 36 shows a plot of the correlation coefficients between all state variables. The correlation coefficients are defined by

$$\rho_{ij} = \frac{P^+_{ij}(k)}{\sqrt{P^+_{ii}(k)P^+_{jj}(k)}}.$$

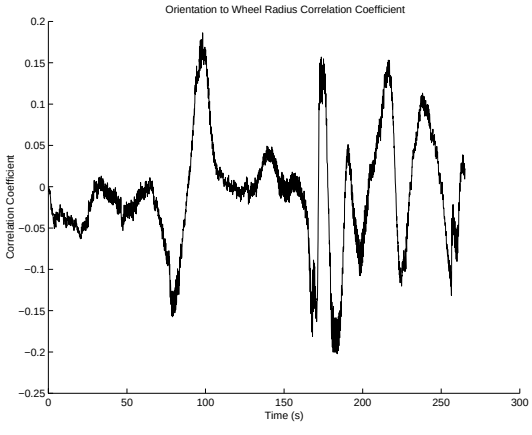


(a)

(b)



(c)



(d)

Figure 36: Filter correlation coefficients (a) x to y, (b) x and y to orientation, (c) x and y to wheel radius, (d) orientation to wheel radius.

When $\rho_{ij} = 0$, states are uncorrelated, when $|\rho_{ij}| \to 1$, the states are highly correlated. Correlation between states is desirable to aid estimation performances, but can cause quite complex and counter-intuitive behaviour in the filter. Figure 36(a) shows that, in the example presented, x and y vehicle state estimates are usually highly correlated. The reason for this is that the error is injected along-track and cross-track, and the vehicle is primarily traveling along a line parallel to $x = y$. Thus the same injected along and cross track errors appear in both x and y state estimates and this is reflected in the fact that these two estimates are highly correlated. It will be noted that the points of switch to negative correlation correspond exactly to the points of large steering angle when the vehicle changes direction. It should be noted that if the vehicle were traveling with the track aligned with the global coordinate frame, then the vehicle x and y estimates would be almost uncorrelated.

Figure 36(b) shows that the vehicle location estimates are moderately and consistently correlated with vehicle orientation estimates. This is logically due to the fact that errors in orientation cause errors in position. At high steering rates, when orientation error is at it's highest, the position becomes highly correlated with orientation. This is a consequence of the large steering uncertainty being injected into the system, feeding through to cross-track error and then subsequently to orientation and position errors (thus increasing their correlation).

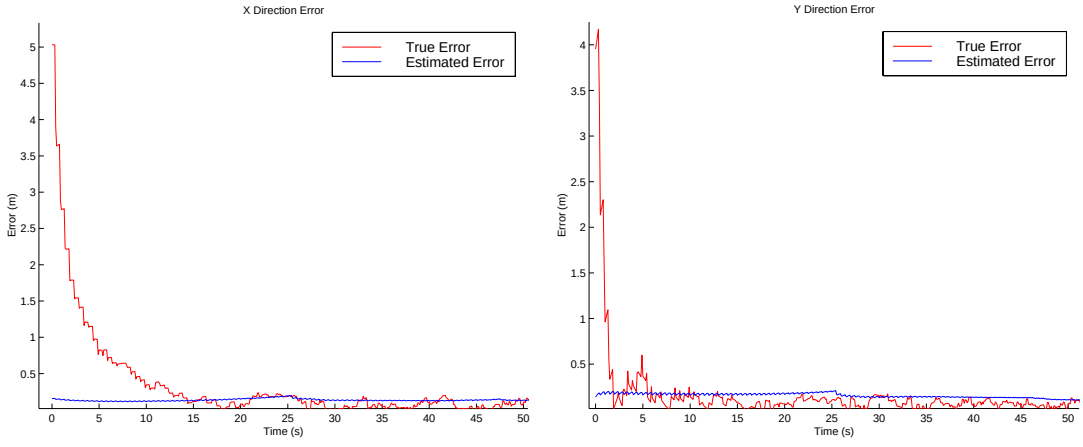
Figures 36(c) and (d) show that the wheel radius estimate is only weakly correlated with position and orientation. The correlation between wheel radius and other, measured, states, is essential in allowing the wheel radius to be observed and estimated. However, error in vehicle motion is due to both track error and wheel radius error and the correlation between states and wheel radius reflects only the error due to the wheel radius itself. The observed variations in position to wheel radius correlation are simply due to the changing direction of travel of the vehicle: wheel radius errors only couple in to along-track motion error.

8.3.5 Errors and Faults

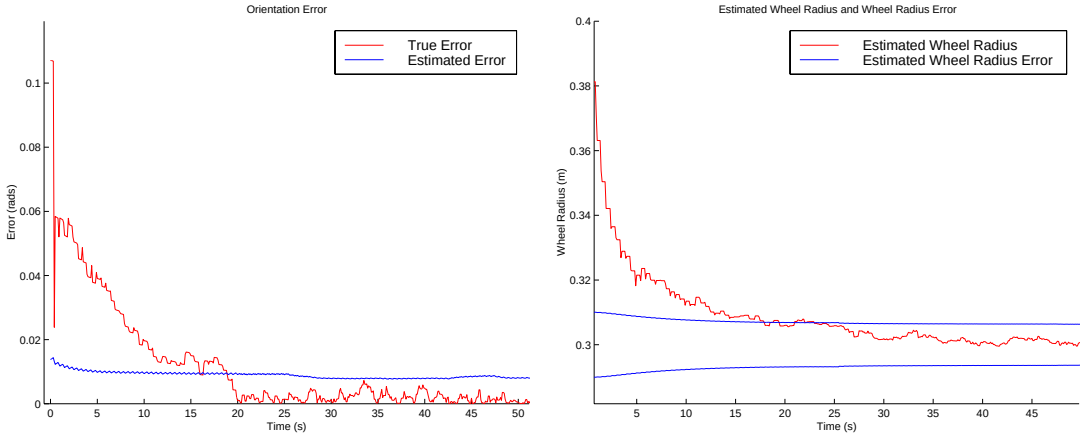
The linear Kalman filter is stable for any initial conditions and for any perturbation around true state values. No matter what the error between actual and estimated state, the filter will always converge to the correct estimate. No such guarantees are possible for the extended Kalman filter.

With a non-linear process model, and the various linearisation processes involved in the EKF, there is no guarantee that the filter will converge given an error in state estimate. Initialisation error in particular is often one of the most difficult practical problems to address in any real-world application of the EKF. Unfortunately, there are no general solutions to the initialisation problem as it depends heavily on the process model employed, sampling intervals and the specific domain in question.

A practical solution to the problem of initialisation is to use a batch process estimate



(a) (b)



(c) (d)

Figure 37: Initial transients in filter state estimates following an initialisation error $[\delta x, \delta y, \delta\phi, \delta R] = [5.0, 5.0, 0.2, 0.1]$: estimates in (a) x, (b) y, (c) ϕ and (d) R.

to provide initial values for state variables. Of interest then is what initial error can be tolerated before the filter becomes unstable. For the example presented here, it turns out that very large errors can be tolerated with relative ease. Figure 37 shows the transients in the filter with an initial error of $[\delta x, \delta y, \delta\phi, \delta R] = [5.0, 5.0, 0.2, 0.1]$. It can be seen that the state estimates converge rapidly to the correct values. In this navigation example, the process model employed, while non-linear, is relatively "smooth" so even when there are large errors, linearisation is still a good approximation of the direction and magnitude of the error.

Practically, in this example, very large errors in state can be tolerated and the filter will still converge, however what kills the algorithm is the errors that result in data association between observed and true beacons. Many other process models are also similarly "smooth" and can accommodate substantial initialisation errors. However, there are also many non-linear process models that are not smooth and for which only moderate state errors can cause catastrophe. Common amongst these are systems that involve products of states to be estimated.

A degree of care is always necessary to ensure an extended Kalman filter estimate is sufficiently close to the true state that the filter will converge.

8.3.6 Tuning Filter Parameters

In this section estimated filter noise parameters are varied and the effects of these changes on filter performance are analysed. The purpose of this is to demonstrate what filter mismatch effects can occur, how to identify these, and how to consequently tune a filter to achieve acceptable performance levels.

Unlike the case for linear filters, there is no general methodology for estimating filter noise parameters for extended Kalman filters in general. However, there are some general principles that should be observed during tuning. In particular, it is often not hard to start with "reasonable" values or at least reasonable ranges of values for the different filter parameters. Many of these can be deduced from observation of the system, simple experiments or simulation. In complex problems, it is good practice to establish an "error budget" and to systematically analyse contributions to overall estimation performance (see [11] Chapter 6 for an excellent discussion of this process). It is also good practice (and generally produces best results) if the *true source* of error is identified and analysed. For example, in the navigation system described in this section, there are no direct sources of position error; position error is simply a consequence of steering, drive and wheel radius errors (the true sources of error).

References

[1] Special issue on the applications of the kalman filter. *IEEE Trans. Automatic Control*, 28(3), 1983.

[2] Y. Bar-Shalom. *Multi-Target Multi-Sensor Tracking.* Artec House, 1990.

[3] Y. Bar-Shalom. *Multi-Target Multi-Sensor Tracking II.* Artec House, 1990.

[4] Y. Bar-Shalom and T.E. Fortmann. *Tracking and Data Association.* Academic Press, 1988.

[5] R.G. Brown and P.Y.C. Hwang. *Introduction to Applied Kalman Filtering.* John Wiley, 1992.

[6] D.E. Catlin. *Estimation, Control and the Discrete Kalman Filter.* Springer Verlag, 1984.

[7] H.F. Durrant-Whyte. An autonomous guided vehicle for cargo handling applications. *Int. J. Robotics Research*, 15(5):407–440, 1996.

[8] M. Gelb. *Applied Optimal Estimation.* MIT press, 1974.

[9] M.S. Grewal and A.P. Andrews. *Kalman Filtering Theory and Practise.* Prentice Hall, 1993.

[10] S. Julier and H. Durrant-Whyte. Navigation and parameter estimation of high-speed road vehicles. In *Proc. IEEE Int. Conf. Robotics and Automation*, 1995.

[11] P.S. Maybeck. *Stochastic Models, Estimaton and Control, Vol. I.* Academic Press, 1979.

[12] M. Stone. *Coordinate Free Multivaribale Statistics.* Oxford University Press, 1989.