# Estimation II

Ian Reid
Hilary Term, 2001

## 1 Discrete-time Kalman filter

We ended the first part of this course deriving the Discrete-Time Kalman Filter as a recursive Bayes' estimator. In this lecture we will go into the filter in more detail, and provide a new derivation for the Kalman filter, this time based on the idea of **Linear Minimum Variance (LMV) estimation of discrete-time systems**.

### 1.1 Background

The problem we are seeking to solve is the continual estimation of a set of parameters whose values change over time. Updating is achieved by combining a set of observations or measurements $\mathbf{z}(t)$ which contain information about the signal of interest $\mathbf{x}(t)$. The role of the estimator is to provide an estimate $\hat{\mathbf{x}}(t+\tau)$ at some time $t+\tau$. If $\tau > 0$ we have a **prediction** filter, if $\tau < 0$ a **smoothing** filter and if $\tau = 0$ the operation is simply called **filtering**.

Recall that an estimator is said to be **unbiased** if the expectation of its output is the expectation of the quantity being estimated, $E[\hat{\mathbf{x}}] = E[\mathbf{x}]$.

Also recall that a **minimum variance unbiased estimator (MVUE)** is an estimator which is unbiased and minimises the mean square error:

$$\hat{\mathbf{x}} = \arg\min_{\hat{\mathbf{x}}} E[||\hat{\mathbf{x}} - \mathbf{x}||^2|\mathbf{z}] = E[\mathbf{x}|\mathbf{z}]$$

The term $E[||\mathbf{x} - \hat{\mathbf{x}}||^2]$, the so-called **variance of error**, is closely related to the **error covariance matrix**, $E[(\mathbf{x} - \hat{\mathbf{x}})(\mathbf{x} - \hat{\mathbf{x}})^T]$. Specifically, the variance of error of an estimator is equal to the trace of the error covariance matrix,

$$E[||\mathbf{x} - \hat{\mathbf{x}}||^2] = \text{trace} E[(\mathbf{x} - \hat{\mathbf{x}})(\mathbf{x} - \hat{\mathbf{x}})^T].$$

The Kalman filter is a **linear** minimum variance of error filter (i.e. it is the best linear filter over the class of all linear filters) over time-varying and time-invariant filters. In the case of the state vector $\mathbf{x}$ and the observations $\mathbf{z}$ being jointly Gaussian distributed, the MVUE estimator is a linear function of the measurement set $\mathbf{z}$ and thus the MVUE (sometimes written MVE for Minimum Variance of Error estimator) is also a LMV estimator, as we saw in the first part of the course.

**Notation**

The following notation will be used.

| | |
|---|---|
| $\mathbf{z}_k$ | observation vector at time $k$. |
| $\mathbf{Z}^k$ | the set of all observations up to (and including) time $k$. |
| $\mathbf{x}_k$ | system state vector at time $k$. |
| $\hat{\mathbf{x}}_{k\|i}$ | estimation of $\mathbf{x}$ at time $k$ based on time $i$, $k \geq i$. |
| $\tilde{\mathbf{x}}_{k\|k}$ | estimation error, $\hat{\mathbf{x}}_{k\|k} - \mathbf{x}_k$, (tilde notation) |
| $\mathbf{P}_k$ | Covariance matrix. |
| $\mathbf{F}_k$ | State transition matrix. |
| $\mathbf{G}_k$ | Input (control) transition matrix. |
| $\mathbf{H}_k$ | Output transition matrix. |
| $\mathbf{w}_k$ | process (or system, or plant) noise vector. |
| $\mathbf{v}_k$ | measurement noise vector. |
| $\mathbf{Q}_k$ | process (or system, or plant) noise covariance matrix. |
| $\mathbf{R}_k$ | measurement noise covariance matrix. |
| $\mathbf{K}_k$ | Kalman gain matrix. |
| $\boldsymbol{\nu}_k$ | innovation at time $k$. |
| $\mathbf{S}_k$ | innovation covariance matrix at time $k$. |

## 1.2 System and observation model

We now begin the analysis of the Kalman filter. Refer to figure 1. We assume that the system can be modelled by the state transition equation,

$$\mathbf{x}_{k+1} = \mathbf{F}_k \mathbf{x}_k + \mathbf{G}_k \mathbf{u}_k + \mathbf{w}_k \tag{1}$$

where $\mathbf{x}_k$ is the state at time $k$, $\mathbf{u}_k$ is an input control vector, $\mathbf{w}_k$ is additive system or process noise, $\mathbf{G}_k$ is the input transition matrix and $\mathbf{F}_k$ is the state transition matrix.

We further assume that the observations of the state are made through a measurement system which can be represented by a linear equation of the form,

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k, \tag{2}$$

where $\mathbf{z}_k$ is the observation or measurement made at time $k$, $\mathbf{x}_k$ is the state at time $k$, $\mathbf{H}_k$ is the observation matrix and $\mathbf{v}_k$ is additive measurement noise.

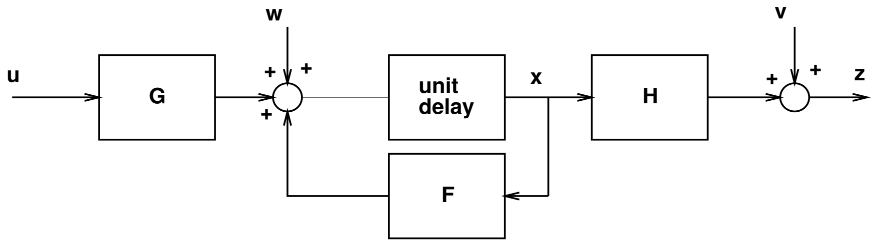



Figure 1: *State-space model.*

### 1.3 Assumptions

We make the following assumptions;

- The process and measurement noise random processes $\mathbf{w}_k$ and $\mathbf{v}_k$ are uncorrelated, zero-mean white-noise processes with known covariance matrices. Then,

$$
\begin{aligned}
E[\mathbf{w}_k \mathbf{w}_l^T] &= \begin{cases} \mathbf{Q}_k & k = l, \\ \mathbf{0} & \text{otherwise,} \end{cases} && (3) \\
E[\mathbf{v}_k \mathbf{v}_l^T] &= \begin{cases} \mathbf{R}_k & k = l, \\ \mathbf{0} & \text{otherwise,} \end{cases} && (4) \\
E[\mathbf{w}_k \mathbf{v}_l^T] &= 0 \quad \text{for all } k, l && (5)
\end{aligned}
$$

  where $\mathbf{Q}_k$ and $\mathbf{R}_k$ are symmetric positive semi-definite matrices.

- The initial system state, $\mathbf{x}_0$ is a random vector that is uncorrelated to both the system and measurement noise processes.

- The initial system state has a known mean and covariance matrix

$$\hat{\mathbf{x}}_{0|0} = E[\mathbf{x}_0] \quad \text{and} \quad \mathbf{P}_{0|0} = E[(\hat{\mathbf{x}}_{0|0} - \mathbf{x}_0)(\hat{\mathbf{x}}_{0|0} - \mathbf{x}_0)^T] \tag{6}$$

Given the above assumptions the task is to determine, given a set of observations $\mathbf{z}_1, \ldots, \mathbf{z}_{k+1}$, the estimation filter that at the $k+1$th instance in time generates an optimal estimate of the state $\mathbf{x}_{k+1}$, which we denote by $\hat{\mathbf{x}}_{k+1}$, that minimises the expectation of the squared-error loss function,

$$E[||\mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1}||^2] = E[(\mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1})^T(\mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1})] \tag{7}$$

### 1.4 Derivation

Consider the estimation of state $\hat{\mathbf{x}}_{k+1}$ based on the observations up to time $k$, $\mathbf{z}_1 \ldots, \mathbf{z}_k$, namely $\hat{\mathbf{x}}_{k+1|\mathbf{Z}^k}$. This is called a one-step-ahead prediction or simply a **prediction**. Now, the solution to the minimisation of Equation 7 is the expectation of the state at time $k+1$ conditioned on the observations up to time $k$. Thus,

$$\hat{\mathbf{x}}_{k+1|k} = E[\mathbf{x}_{k+1}|\mathbf{z}_1, \dots, \mathbf{z}_k] = E[\mathbf{x}_{k+1}|\mathbf{Z}^k] \tag{8}$$

Then the predicted state is given by

$$\begin{aligned}
\hat{\mathbf{x}}_{k+1|k} &= E[\mathbf{x}_{k+1}|\mathbf{Z}^k] \\
&= E[\mathbf{F}_k\mathbf{x}_k + \mathbf{G}_k\mathbf{u}_k + \mathbf{w}_k|\mathbf{Z}^k] \\
&= \mathbf{F}_k E[\mathbf{x}_k|\mathbf{Z}^k] + \mathbf{G}_k\mathbf{u}_k + E[\mathbf{w}_k|\mathbf{Z}^k] \\
&= \mathbf{F}_k\hat{\mathbf{x}}_{k|k} + \mathbf{G}_k\mathbf{u}_k
\end{aligned} \tag{9}$$

where we have used the fact that the process noise has zero mean value and $\mathbf{u}_k$ is known precisely.

The estimate variance $\mathbf{P}_{k+1|k}$ is the mean squared error in the estimate $\hat{\mathbf{x}}_{k+1|k}$.

Thus, using the facts that $\mathbf{w}_k$ and $\hat{\mathbf{x}}_{k|k}$ are uncorrelated:

$$\begin{aligned}
\mathbf{P}_{k+1|k} &= E[(\mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1|k})(\mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1|k})^T|\mathbf{Z}^k] \\
&= \mathbf{F}_k E[(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k})(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k})^T|\mathbf{Z}^k]\mathbf{F}_k^T + E[\mathbf{w}_k\mathbf{w}_k^T] \\
&= \mathbf{F}_k\mathbf{P}_{k|k}\mathbf{F}_k^T + \mathbf{Q}_k
\end{aligned} \tag{10}$$

Having obtained a predictive estimate $\hat{\mathbf{x}}_{k+1|k}$ suppose that we now take another observation $\mathbf{z}_{k+1}$. How can we use this information to update the prediction, ie. find $\hat{\mathbf{x}}_{k+1|k+1}$? We assume that the estimate is a linear weighted sum of the prediction and the new observation and can be described by the equation,

$$\hat{\mathbf{x}}_{k+1|k+1} = \mathbf{K}'_{k+1}\hat{\mathbf{x}}_{k+1|k} + \mathbf{K}_{k+1}\mathbf{z}_{k+1} \tag{11}$$

where $\mathbf{K}'_{k+1}$ and $\mathbf{K}_{k+1}$ are weighting or **gain** matrices (of different sizes). Our problem now is to reduced to finding the $\mathbf{K}_{k+1}$ and $\mathbf{K}'_{k+1}$ that minimise the conditional mean squared estimation error where of course the estimation error is given by:

$$\tilde{\mathbf{x}}_{k+1|k+1} = \hat{\mathbf{x}}_{k+1|k+1} - \mathbf{x}_{k+1} \tag{12}$$

## 1.5 The Unbiased Condition

For our filter to be unbiased, we require that $E[\hat{\mathbf{x}}_{k+1|k+1}] = E[\mathbf{x}_{k+1}]$. Let's assume (and argue by induction) that $\hat{\mathbf{x}}_{k|k}$ is an unbiased estimate. Then combining equations (11) and (2) and taking expectations yields

$$\begin{aligned}
E[\hat{\mathbf{x}}_{k+1|k+1}] &= E[\mathbf{K}'_{k+1}\hat{\mathbf{x}}_{k+1|k} + \mathbf{K}_{k+1}\mathbf{H}_{k+1}\mathbf{x}_{k+1} + \mathbf{K}_{k+1}\mathbf{v}_{k+1}] \\
&= \mathbf{K}'_{k+1}E[\hat{\mathbf{x}}_{k+1|k}] + \mathbf{K}_{k+1}\mathbf{H}_{k+1}E[\mathbf{x}_{k+1}] + \mathbf{K}_{k+1}E[\mathbf{v}_{k+1}]
\end{aligned} \tag{13}$$

Note that the last term on the right hand side of the equation is zero, and further note that the prediction is unbiased:

$$\begin{aligned}
E[\hat{\mathbf{x}}_{k+1|k}] &= E[\mathbf{F}_k\hat{\mathbf{x}}_{k|k} + \mathbf{G}_k\mathbf{u}_k] \\
&= \mathbf{F}_k E[\hat{\mathbf{x}}_{k|k}] + \mathbf{G}_k\mathbf{u}_k \\
&= E[\mathbf{x}_{k+1}]
\end{aligned} \tag{14}$$

Hence by combining equations (13) and (14))

$$E[\hat{\mathbf{x}}_{k+1|k+1}] = (\mathbf{K}'_{k+1} + \mathbf{K}_{k+1}\mathbf{H}_{k+1})E[\mathbf{x}_{k+1}]$$

and the condition that $\hat{\mathbf{x}}_{k+1|k+1}$ be unbiased reduces the requirement to

$$\begin{aligned} \mathbf{K}'_{k+1} + \mathbf{K}_{k+1}\mathbf{H}_{k+1} &= \mathbf{I} \\ \text{or} \quad \mathbf{K}'_{k+1} &= \mathbf{I} - \mathbf{K}_{k+1}\mathbf{H}_{k+1} \end{aligned} \tag{15}$$

We now have that for our estimator to be unbiased is must satisfy

$$\begin{aligned} \hat{\mathbf{x}}_{k+1|k+1} &= (\mathbf{I} - \mathbf{K}_{k+1}\mathbf{H}_{k+1})\hat{\mathbf{x}}_{k+1|k} + \mathbf{K}_{k+1}\mathbf{z}_{k+1} \\ &= \hat{\mathbf{x}}_{k+1|k} + \mathbf{K}_{k+1}[\mathbf{z}_{k+1} - \mathbf{H}_{k+1}\hat{\mathbf{x}}_{k+1|k}] \end{aligned} \tag{16}$$

where $\mathbf{K}$ is known as the **Kalman gain**.

Note that since $\mathbf{H}_{k+1}\hat{\mathbf{x}}_{k+1|k}$ can be interpreted as a predicted observation $\hat{\mathbf{z}}_{k+1|k}$, equation 16 can be interpreted as the sum of a prediction and a fraction of the difference between the predicted and actual observation.

## 1.6 Finding the Error Covariance

We determined the prediction error covariance in equation (10). We now turn to the updated error covariance

$$\begin{aligned} \mathbf{P}_{k+1|k+1} &= E[\tilde{\mathbf{x}}_{k+1|k+1}\tilde{\mathbf{x}}_{k+1|k+1})^T|\mathbf{Z}^k] \\ &= E[(\mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1|k+1})(\mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1|k+1})^T] \\ &= (\mathbf{I} - \mathbf{K}_{k+1}\mathbf{H}_{k+1})E[\tilde{\mathbf{x}}_{k+1|k}\tilde{\mathbf{x}}^T_{k+1|k}](\mathbf{I} - \mathbf{K}_{k+1}\mathbf{H}_{k+1})^T \\ &\quad +\mathbf{K}_{k+1}E[\mathbf{v}_{k+1}\mathbf{v}^T_{k+1}]\mathbf{K}^T_{k+1} + 2(\mathbf{I} - \mathbf{K}_{k+1}\mathbf{H}_{k+1})E[\tilde{\mathbf{x}}_{k+1|k}\mathbf{v}^T_{k+1}]\mathbf{K}^T_{k+1} \end{aligned}$$

and with

$$\begin{aligned} E[\mathbf{v}_{k+1}\mathbf{v}^T_{k+1}] &= \mathbf{R}_{k+1} \\ E[\tilde{\mathbf{x}}_{k+1|k}\tilde{\mathbf{x}}^T_{k+1|k}] &= \mathbf{P}_{k+1|k} \\ E[\tilde{\mathbf{x}}_{k+1|k}\mathbf{v}^T_{k+1}] &= \mathbf{0} \end{aligned}$$

we obtain

$$\mathbf{P}_{k+1|k+1} = (\mathbf{I} - \mathbf{K}_{k+1}\mathbf{H}_{k+1})\mathbf{P}_{k+1|k}(\mathbf{I} - \mathbf{K}_{k+1}\mathbf{H}_{k+1})^T + \mathbf{K}_{k+1}\mathbf{R}_{k+1}\mathbf{K}^T_{k+1} \tag{17}$$

Thus the covariance of the updated estimate is expressed in terms of the prediction covariance $\mathbf{P}_{k+1|k}$, the observation noise $\mathbf{R}_{k+1}$ and the Kalman gain matrix $\mathbf{K}_{k+1}$.

## 1.7 Choosing the Kalman Gain

Our goal is now to minimise the conditional mean-squared estimation error with respect to the Kalman gain, $\mathbf{K}$.

$$
\begin{aligned}
L &= \min_{\mathbf{K}_{k+1}} E[\tilde{\mathbf{x}}_{k+1|k+1}^T \tilde{\mathbf{x}}_{k+1|k+1} | \mathbf{Z}^{k+1}] \\
&= \min_{\mathbf{K}_{k+1}} \text{trace} \left( E[\tilde{\mathbf{x}}_{k+1|k+1} \tilde{\mathbf{x}}_{k+1|k+1}^T | \mathbf{Z}^{k+1}] \right) \\
&= \min_{\mathbf{K}_{k+1}} \text{trace} \left( \mathbf{P}_{k+1|k+1} \right) \qquad (18)
\end{aligned}
$$

For any matrix $\mathbf{A}$ and a symmetric matrix $\mathbf{B}$

$$
\frac{\partial}{\partial \mathbf{A}} \left( \text{trace}(\mathbf{ABA}^T) \right) = 2\mathbf{AB}
$$

(to see this, consider writing the trace as $\sum_i \mathbf{a}_i^T \mathbf{B} \mathbf{a}_i$ where $\mathbf{a}_i$ are the columns of $\mathbf{A}^T$, and then differentiating w.r.t. the $\mathbf{a}_i$).

Combining equations (17) and (18) and differentiating with respect to the gain matrix (using the relation above) and setting equal to zero yields

$$
\frac{\partial L}{\partial \mathbf{K}_{k+1}} = -2(\mathbf{I} - \mathbf{K}_{k+1}\mathbf{H}_{k+1})\mathbf{P}_{k+1|k}\mathbf{H}_{k+1}^T + 2\mathbf{K}_{k+1}\mathbf{R}_{k+1} = \mathbf{0}
$$

Re-arranging gives an equation for the gain matrix

$$
\mathbf{K}_{k+1} = \mathbf{P}_{k+1|k}\mathbf{H}_{k+1}^T [\mathbf{H}_{k+1}\mathbf{P}_{k+1|k}\mathbf{H}_{k+1}^T + \mathbf{R}_{k+1}]^{-1} \qquad (19)
$$

Together with Equation 16 this defines the optimal linear mean-squared error estimator.

### 1.8 Summary of key equations

At this point it is worth summarising the key equations which underly the Kalman filter algorithm. The algorithm consists of two steps; a prediction step and an update step.

**Prediction:** also known as the time-update. This predicts the state and variance at time $k+1$ dependent on information at time $k$.

$$
\begin{aligned}
\hat{\mathbf{x}}_{k+1|k} &= \mathbf{F}_k \hat{\mathbf{x}}_{k|k} + \mathbf{G}_k \mathbf{u}_k \qquad (20) \\
\mathbf{P}_{k+1|k} &= \mathbf{F}_k \hat{\mathbf{P}}_{k|k} \mathbf{F}_k^T + \mathbf{Q}_k \qquad (21)
\end{aligned}
$$

**Update:** also known as the measurement update. This updates the state and variance using a combination of the predicted state and the observation $\mathbf{z}_{k+1}$.

$$
\begin{aligned}
\hat{\mathbf{x}}_{k+1|k+1} &= \hat{\mathbf{x}}_{k+1|k} + \mathbf{K}_{k+1} \left[ \mathbf{z}_{k+1} - \mathbf{H}_{k+1} \hat{\mathbf{x}}_{k+1|k} \right] \qquad (22) \\
\mathbf{P}_{k+1|k+1} &= (\mathbf{I} - \mathbf{K}_{k+1}\mathbf{H}_{k+1})\mathbf{P}_{k+1|k}(\mathbf{I} - \mathbf{K}_{k+1}\mathbf{H}_{k+1})^T + \mathbf{K}_{k+1}\mathbf{R}_{k+1}\mathbf{K}_{k+1}^T \qquad (23)
\end{aligned}
$$

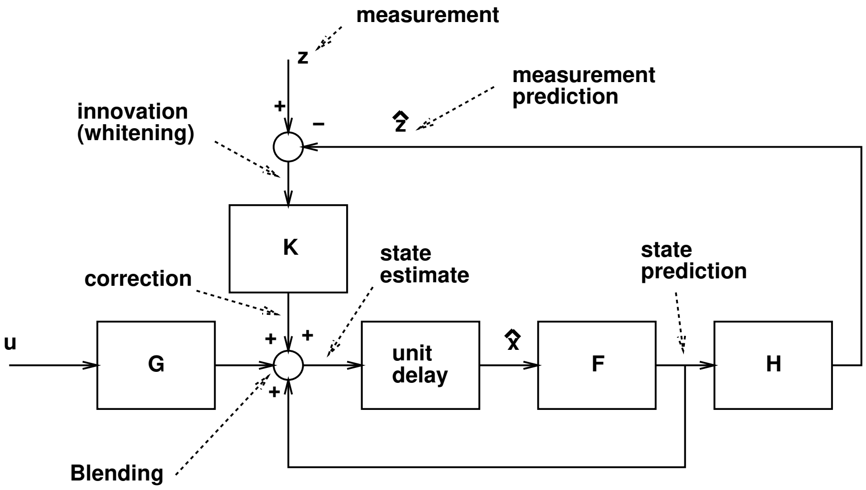



Figure 2: *Discrete-time Kalman filter block diagram.*

where the gain matrix is given by

$$\mathbf{K}_{k+1} = \mathbf{P}_{k+1|k}\mathbf{H}_{k+1}^T[\mathbf{H}_k\mathbf{P}_{k+1|k}\mathbf{H}_{k+1}^T + \mathbf{R}_{k+1}]^{-1} \tag{24}$$

Together with the initial conditions on the estimate and its error covariance matrix (equation 6) this defines the discrete-time sequential, recursive algorithm for determining the linear minimum variance estimate known as the Kalman filter.

## 1.9 Interpreting the Kalman Filter

We now take a look at the overall Kalman filter algorithm in more detail. Figure 2 summarises the stages in the algorithm in block diagram form.

The **innovation**, $\boldsymbol{\nu}_{k+1}$, is defined as the difference between the observation (measurement) $\mathbf{z}_{k+1}$ and its prediction $\hat{\mathbf{z}}_{k+1|k}$ made using the information available at time $k$. It is a measure of the new information provided by adding another measurement in the estimation process.

Given that

$$\begin{aligned}\hat{\mathbf{z}}_{k+1|k} &= E[\mathbf{z}_{k+1}|\mathbf{Z}^k] \\ &= E[\mathbf{H}_{k+1}\mathbf{x}_{k+1} + \mathbf{v}_{k+1}|\mathbf{Z}^k] \\ &= \mathbf{H}_{k+1}\hat{\mathbf{x}}_{k+1|k} \end{aligned} \tag{25}$$

the innovation $\boldsymbol{\nu}_{k+1}$ can be expressed by

$$\boldsymbol{\nu}_{k+1} = \mathbf{z}_{k+1} - \mathbf{H}_{k+1}\hat{\mathbf{x}}_{k+1|k} \tag{26}$$

The innovation or residual is an important measure of how well an estimator is performing. For example it can be used to validate a measurement prior to it being included as a member of the observation sequence (more on this later).

The process of transforming $\mathbf{z}_{k+1}$ into $\boldsymbol{\nu}_{k+1}$ is sometimes said to be achieved through the **Kalman whitening filter**. This is because the innovations form an uncorrelated orthogonal white-noise process sequence $\mathbf{V}^{k+1}$ which is statistically equivalent to the observations $\mathbf{Z}^{k+1}$. This is important because where as $\mathbf{z}_{k+1}$ is in generally statistically correlated, the innovation $\boldsymbol{\nu}_{k+1}$ is uncorrelated so effectively provides new information or "innovation".

The innovation has zero mean, since,

$$\begin{aligned} E[\boldsymbol{\nu}_{k+1}|\mathbf{Z}^k] &= E[\mathbf{z}_{k+1} - \hat{\mathbf{z}}_{k+1|k}|\mathbf{Z}^k] \\ &= E[\mathbf{z}_{k+1}|\mathbf{Z}^k] - \hat{\mathbf{z}}_{k+1|k}, \\ &= \mathbf{0} \end{aligned} \tag{27}$$

and the innovation variance $\mathbf{S}_{k+1}$ is given by

$$\begin{aligned} \mathbf{S}_{k+1} &= E[\boldsymbol{\nu}_{k+1}\boldsymbol{\nu}_{k+1}^T], \\ &= E[(\mathbf{z}_{k+1} - \mathbf{H}_{k+1}\hat{\mathbf{x}}_{k+1|k})(\mathbf{z}_{k+1} - \mathbf{H}_{k+1}\hat{\mathbf{x}}_{k+1|k})^T] \\ \mathbf{S}_{k+1} &= \mathbf{R}_{k+1} + \mathbf{H}_{k+1}\mathbf{P}_{k+1|k}\mathbf{H}_{k+1}^T \end{aligned} \tag{28}$$

Using Equation 26 and 28 we can re-write the Kalman updates in terms of the innovation and its variance as follows.

$$\hat{\mathbf{x}}_{k+1|k+1} = \hat{\mathbf{x}}_{k+1|k} + \mathbf{K}_{k+1}\boldsymbol{\nu}_{k+1} \tag{29}$$

$$\begin{aligned} \mathbf{P}_{k+1|k+1} &= E[(\mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1|k} - \mathbf{K}_{k+1}\boldsymbol{\nu}_{k+1})(\mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1|k} - \mathbf{K}_{k+1}\boldsymbol{\nu}_{k+1})^T] \\ &= E[(\mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1|k})(\mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1|k})^T] - \mathbf{K}_{k+1}E[\boldsymbol{\nu}_{k+1}\boldsymbol{\nu}_{k+1}^T] \\ \mathbf{P}_{k+1|k+1} &= \mathbf{P}_{k+1|k} - \mathbf{K}_{k+1}\mathbf{S}_{k+1}\mathbf{K}_{k+1}^T \end{aligned} \tag{30}$$

where, from Equation 19

$$\mathbf{K}_{k+1} = \mathbf{P}_{k+1|k}\mathbf{H}_{k+1}^T\mathbf{S}_{k+1}^{-1} \tag{31}$$

and

$$\mathbf{S}_{k+1} = \mathbf{H}_{k+1}\mathbf{P}_{k+1|k}\mathbf{H}_{k+1}^T + \mathbf{R}_{k+1}. \tag{32}$$

This is a convenient form of the Kalman filter often used in analysis.

Although primarily used as a state estimator the Kalman filter algorithm can be used to estimate parameters other than the state vector. These are illustrated in Figure 2.

1. If applied to estimate $\hat{\mathbf{z}}_{k+1|k}$ it is called a **measurement** filter.
2. If applied to estimate $\hat{\mathbf{x}}_{k+1|k}$ it is called a **prediction** filter.
3. If applied to estimate $\boldsymbol{\nu}_{k+1}$ it is called a **whitening** filter.
4. If applied to estimate $\hat{\mathbf{x}}_{k+1|k+1}$ it is called a **Kalman** filter.

## 1.10 Example

Consider a vehicle tracking problem where a vehicle is constrained to move in a straight line with a constant velocity. Let $p(t)$ and $\dot{p}(t)$ represent the vehicle position and velocity. We assume that observations of position can be made where the measurement noise is $v(t)$. Since the vehicle is moving at constant speed, $\ddot{p}(t) = 0$.

**System model:** The system state can be described by $\mathbf{x}_(t) = [p(t), \dot{p}(t)]^T$. The system state and output equations can be expressed by

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{w}(t), \\ \mathbf{z}(t) &= \mathbf{H}\mathbf{x}(t) + \mathbf{v}(t), \end{aligned} \tag{33}$$

where

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad \mathbf{H} = \begin{bmatrix} 1 & 0 \end{bmatrix}.$$

Suppose we have sampled observations of the system at discrete time intervals $\triangle T$, then the discrete equivalent is given by (see later this lecture for a derivation)

$$\mathbf{x}_{k+1} = \mathbf{F}_k \mathbf{x}_k + \mathbf{w}_k \tag{34}$$

where

$$\mathbf{F}_k = e^{\mathbf{A}\triangle T} = \begin{bmatrix} 1 & \triangle T \\ 0 & 1 \end{bmatrix} \tag{35}$$

$$\mathbf{z}_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_k \tag{36}$$

**Kalman filter:** Suppose that the known mean and covariance of $\mathbf{x}_0 = \mathbf{x}(0)$ are,

$$\hat{\mathbf{x}}_{0|0} = E(\mathbf{x}_0) = \begin{bmatrix} 0 \\ 10 \end{bmatrix} \qquad \mathbf{P}_{0|0} = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix} \tag{37}$$

Assume also that

$$\mathbf{Q}_k = E(\mathbf{w}_k \mathbf{w}_k^T) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad \mathbf{R}_k = E(\mathbf{v}_k^2) = 1 \tag{38}$$

The Kalman filter involves sequential application of the recursive equations as given above for $k = 0, 1, \ldots$. Here is some Matlab code to implement them, and an example program

```
function [xpred, Ppred] = predict(x, P, F, Q)

     xpred = F*x;
     Ppred = F*P*F' + Q;


function [nu, S] = innovation(xpred, Ppred, z, H, R)

     nu = z - H*xpred;                    %% innovation
     S = R + H*Ppred*H';                  %% innovation covariance


function [xnew, Pnew] = innovation_update(xpred, Ppred, nu, S, H)

     K = Ppred*H'*inv(S);           %% Kalman gain
     xnew = xpred + K*nu;           %% new state
     Pnew = Ppred - K*S*K';         %% new covariance
```

```
%%% Matlab script to simulate data and process usiung Kalman filter

delT = 1;
F = [ 1 delT
      0   1  ];
H = [ 1 0 ];
x = [ 0
     10];
P = [ 10  0
       0 10 ];
Q = [  1  1
       1  1 ];
R = [ 1 ];
z = [2.5 1 4 2.5 5.5 ];

for i=1:5
   [xpred, Ppred] = predict(x, P, F, Q);
   [nu, S] = innovation(xpred, Ppred, z(i), H, R);
   [x, P] = innovation_update(xpred, Ppred, nu, S, H);
end
```

**Results:** The plots in Figure 3a-c illustrate the result of running the Kalman filter using $\Delta t = 1$. Some interesting observations can be made as follows.

1. Both $\mathbf{K}_{k+1}$ and $\mathbf{P}_{k+1|k+1}$ tend to constant (steady-state) values as $k \to \infty$.
2. The estimates $\hat{\mathbf{x}}_{k+1|k+1}$ tend to follow the measurement values quite closely. Indeed since $\mathbf{K}_{k+1}$ is a weighting function acting on the measurement it is clear that this effect is more prominent when $\mathbf{K}_{k+1}$ is high.
3. From Equation 19 $\mathbf{K}_{k+1}$ is decreases with $\mathbf{R}_{k+1}$ and increases with $\mathbf{Q}_{k+1}$. Thus the convergence properties are dependent on the relative magnitudes of the process and measurement noise. Figure 4a illustrates this effect clearly. Here we have re-run the Kalman filter but decreased the elements of $\mathbf{Q}_{k+1}$ by a factor of 10 and 100 ($\mathbf{R}_{k+1}$ was kept at the original value). It is clear from the figure that the net effect is that the estimates follow the measurements less closely. Similar effects can be observed if the relative magnitude of $\mathbf{R}_{k+1}$ is increased (Figure 4b). How do you explain what is observed in the latter case?

This example illustrates the fact that the performance of a Kalman filter is dependent on initialisation conditions. In the next lecture we examine this observation in more detail when we consider the performance of the discrete-time Kalman filter.

## 1.11 Deriving the System Model

Up to this point we have somewhat glossed over the derivation of the discrete system evolution model, but since there are one or two points which are not entirely obvious, we illustrate it here with some examples.

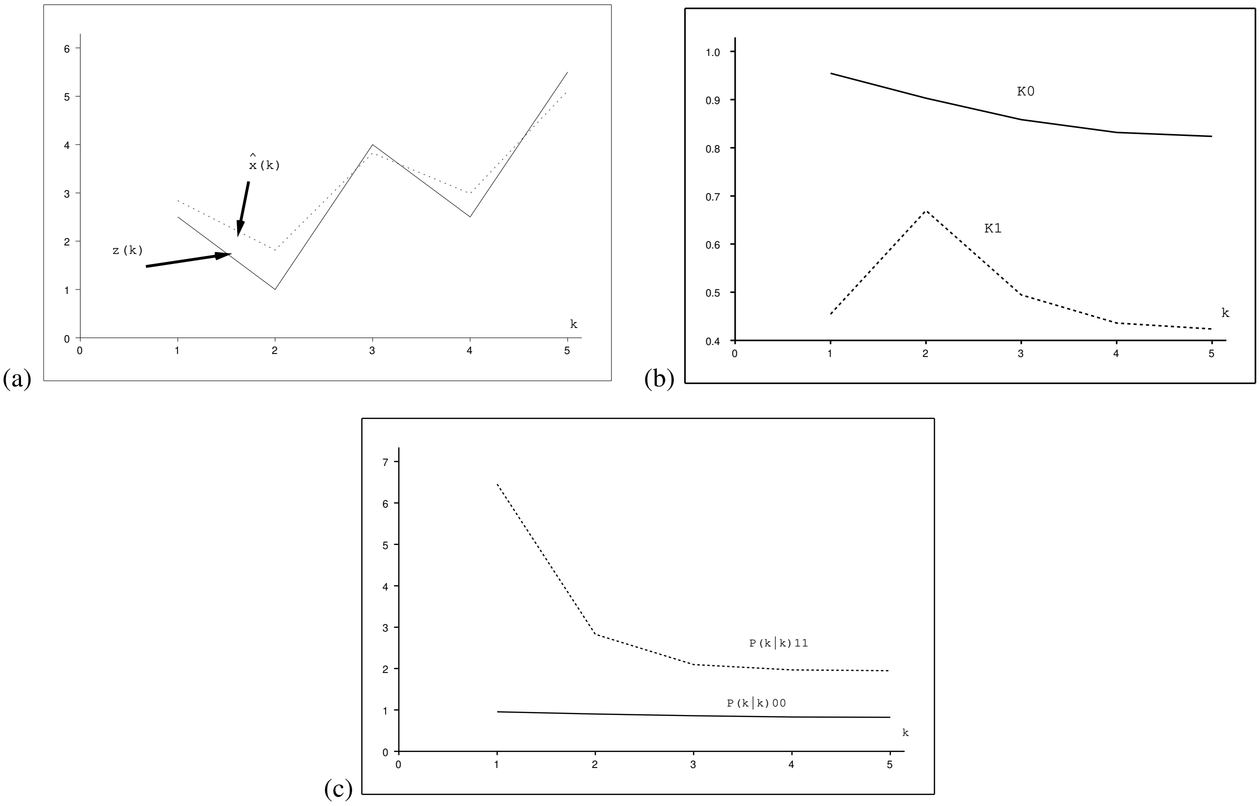



Figure 3: *Example 1: (a) measurement and estimated state trajectories; (b) Kalman gain; (c) diagonal elements of error covariance matrix. Note how in the latter diagrams the curves tend to constant values as $k$ gets large.*

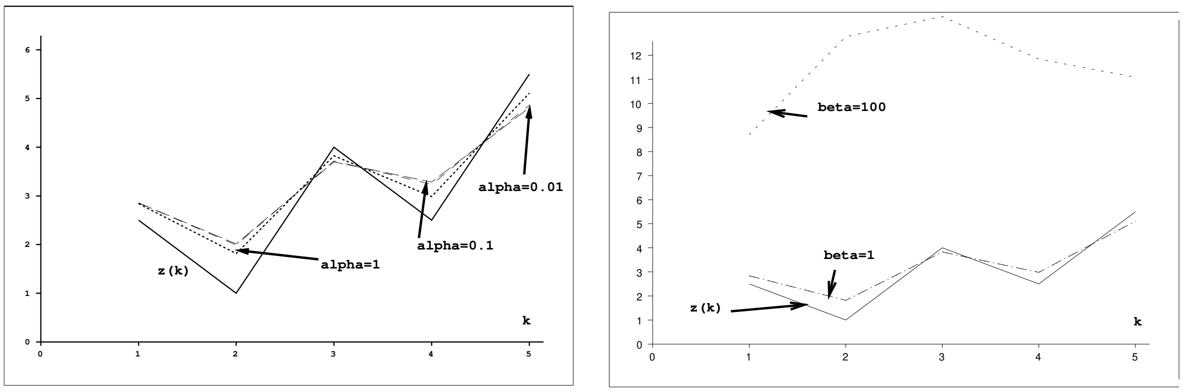



Figure 4: *Example 1: Effect of changing the process and measurement noise covariance matrices on the estimated position trajectory. (a) decreasing $\mathbf{Q}_{k+1}$ by a factor of 10 and 100; (b) increasing $\mathbf{R}_{k+1}$ by a factor of 100.*

### 1.11.1 Constant-velocity particle

Consider a particle moving at constant velocity. It's ideal motion is described by $\ddot{x}(t) = 0$. In the real world, the velocity will undergo some perturbation $w(t)$ which we will assume to be randomly distributed. Therefore the real motion equation is given by,

$$\ddot{x}(t) = w(t)$$

where

$$\begin{aligned} E[w(t)] &= 0 \\ E[w(t)w(\tau)] &= q(t)\delta(t-\tau) \end{aligned}$$

Recall from your 2nd year maths that this latter expectation is the **auto-correlation function** and that an autocorrelation of this form corresponds to a **constant power spectral density**, more commonly referred to as **white noise**.

The state vector is,

$$\mathbf{x} = \begin{bmatrix} x \\ \dot{x} \end{bmatrix} \tag{39}$$

and the continuous state-space system equation is

$$\frac{\partial \mathbf{x}}{\partial t} = \mathbf{A}\mathbf{x}(t) + \mathbf{B}w(t) \tag{40}$$

where

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{B} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{41}$$

We obtain the discrete time equivalent for equation 40 for a sample interval of $\Delta t$ by integating up over one sampling period. Taking Laplace transforms we obtain:

$$\begin{aligned} & s\mathbf{X}(s) - \mathbf{x}(0) = \mathbf{A}\mathbf{X}(s) + \mathbf{B}\mathbf{W}(s) \\ \Rightarrow \quad & (s\mathbf{I} - \mathbf{A})\mathbf{X}(s) = \mathbf{x}(0) + \mathbf{B}\mathbf{W}(s) \\ \Rightarrow \quad & \mathbf{X}(s) = (s\mathbf{I} - \mathbf{A})^{-1}\mathbf{x}(0) + (s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}\mathbf{W}(s) \end{aligned}$$

Now taking inverse Laplace transforms:

$$\mathbf{x}(\Delta t) = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \mathbf{x}(0) + \int_0^{\Delta t} \begin{bmatrix} 1 & \Delta t - \tau \\ 0 & 1 \end{bmatrix} \mathbf{B}w(\tau)d\tau$$

Note that the integration on the right hand side is a consequence of the convolution rule for Laplace transforms.

Shifting this forward in time to match our initial conditions we have

$$\begin{aligned} \mathbf{x}(t_k + \Delta t) &= \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \mathbf{x}(t_k) + \int_0^{\Delta t} \begin{bmatrix} 1 & \Delta t - \tau \\ 0 & 1 \end{bmatrix} \mathbf{B}w(t_k + \tau)d\tau \\ &= \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \mathbf{x}(t_k) + \int_0^{\Delta t} \begin{bmatrix} \Delta t - \tau \\ 1 \end{bmatrix} w(t_k + \tau)d\tau \\ \text{or} \quad \mathbf{x}_{k+1} &= \quad \mathbf{F}_k \quad \mathbf{x}_k \quad + \quad \mathbf{w}_k \end{aligned} \tag{42}$$

It remains to determine the process noise covariance, $\mathbf{Q}_k = E[\mathbf{w}_k\mathbf{w}_k^T]$:

$$\begin{aligned}
\mathbf{Q}_k &= E[\mathbf{w}_k\mathbf{w}_k^T] \\
&= E[(\int_0^{\Delta t}\begin{bmatrix}\Delta t - u \\ 1\end{bmatrix} w(t_k+u)du)(\int_0^{\Delta t}\begin{bmatrix}\Delta t - v \\ 1\end{bmatrix} w(t_k+v)dv)^T] \\
&= \int_0^{\Delta t}\int_0^{\Delta t}\begin{bmatrix}\Delta t - u \\ 1\end{bmatrix}\begin{bmatrix}\Delta t - v & 1\end{bmatrix} E[w(t_k+u)w(t_k+v)^T]dudv \\
&= \int_0^{\Delta t}\begin{bmatrix}(\Delta t - u)^2 & (\Delta t - u) \\ (\Delta t - u) & 1\end{bmatrix} q(t_k+u)du \\
&= q\begin{bmatrix}\frac{1}{3}\Delta t^3 & \frac{1}{2}\Delta t^2 \\ \frac{1}{2}\Delta t^2 & \Delta t\end{bmatrix}
\end{aligned} \tag{43}$$

since (by our initial assumptions) $q$ is constant.

In summary:

$$\mathbf{F}_k = \begin{bmatrix}1 & \Delta t \\ 0 & 1\end{bmatrix} \quad \text{and} \quad \mathbf{Q}_k = q\begin{bmatrix}\frac{1}{3}\Delta t^3 & \frac{1}{2}\Delta t^2 \\ \frac{1}{2}\Delta t^2 & \Delta t\end{bmatrix} \tag{44}$$

Note that we have derived the process noise covariance using a **continuous-time white noise** assumption. If, on the other hand we assume a **piece-wise constant white noise** model, then the target undergoes a constant acceleration $w_k$, with the accelerations independent from period to period, and the process noise is slightly different. Bar-Shalom (p86) discusses (and derives the piecewise version) this and you should familiarise yourselves with both, but always keeeping in mind that both are approximations.

### 1.11.2 Constant-acceleration particle

The analysis is similar to the previous case. In this case the particle moves at constant acceleration. The ideal particle motion is described by $\partial\ddot{x}(t)/\partial t = 0$. In the real world, the acceleration will not be perfectly constant, thus,

$$\frac{\partial\ddot{x}(t)}{\partial t} = w(t),$$

where as before,

$$\begin{aligned}
E[w(t)] &= 0, \\
E[w(t)w(\tau)] &= q(t)\delta(t-\tau)
\end{aligned}$$

We can express the state-space model of the system as follows.

The state vector is

$$\mathbf{x} = \begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix} \tag{45}$$

The continuous state-space system equation is,

$$\frac{\partial\mathbf{x}}{\partial t} = \mathbf{A}\mathbf{x}(t) + \mathbf{B}w(t), \tag{46}$$

where

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{47}$$

The discrete-time equivalent to Equation 46 for a sample interval of $\Delta t$ is given by,

$$\mathbf{x}_{k+1} = \mathbf{F}_k \mathbf{x}_k + \mathbf{w}_k \tag{48}$$

Once again taking Laplace transforms (etc) and using the assumption that $q$ is constant, we can determine the state transition matrix $\mathbf{F}k$ and the process noise covariance matrix $\mathbf{Q}_k$:

$$\begin{aligned} \mathbf{F}_k = e^{\mathbf{A}\Delta t} \qquad &= \begin{bmatrix} 1 & \Delta t & \Delta t^2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix} \\ \text{and} \quad \mathbf{Q}_k = E[\mathbf{w}_k \mathbf{w}_k^T] &= q \begin{bmatrix} \Delta t^5/20 & \Delta t^4/8 & \Delta t^3/6 \\ \Delta t^4/8 & \Delta t^3/3 & \Delta t^2/2 \\ \Delta t^3/6 & \Delta t^2/2 & \Delta t \end{bmatrix} \end{aligned} \tag{49}$$

# 2 Kalman Filter Performance

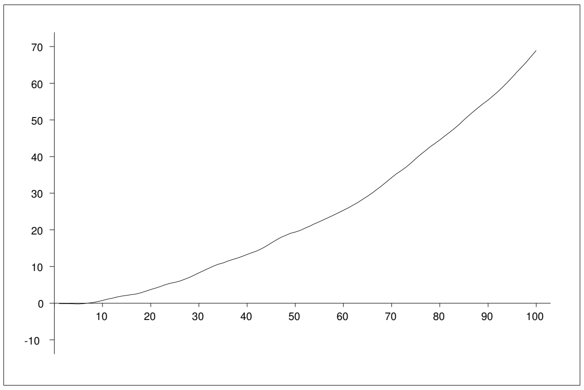
In this lecture we consider how to evaluate the performance of a Kalman filter. We will focus on understanding the following problems

1. how a Kalman filter of a perfectly modelled system with perfectly estimated noise behaves.
2. the effect that changes in the values of the noise sources have on the overall performance of a Kalman filter.
3. how to recognise if the filter assumptions are met in practice. This is particularly important in practical situations since many real systems are not well represented by a linear model and measurement and system noise is non-Gaussian.

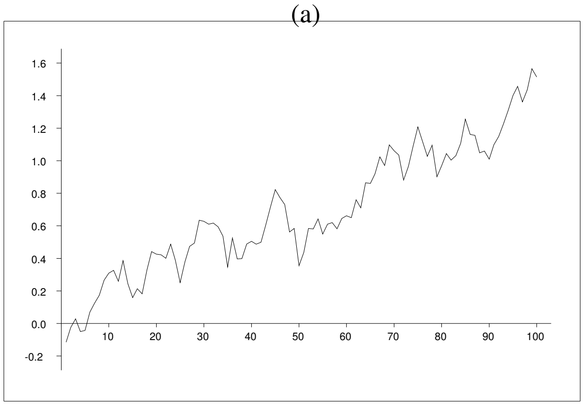
In the case of (3), here, we will only consider how to detect a problem with a Kalman filter. We consider how to modify the Kalman filter to accommodate nonlinear process and measurement models and non-Gaussian noise in the final lecture.

## 2.1 Example

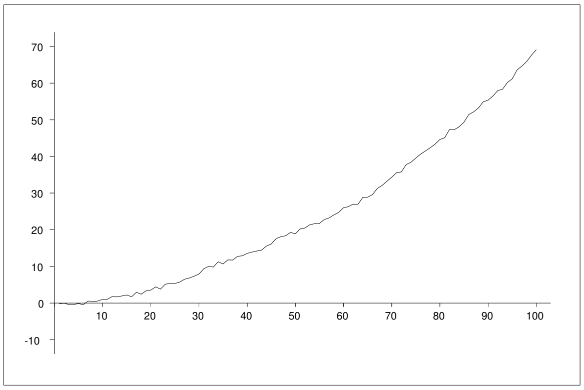
We will illustrate Kalman filter performance using as an example a variant of the constant velocity model we considered in section 1.10, in which we use the Kalman filter to track a vehicle moving in a straight line.



(a)



(b)



(c)

Figure 5: *Input to Example 2: (a) true position; (b) true velocity; (c) observations.*

**System model**

We assume that sampled observations are acquired at discrete time intervals $\Delta t$ and the system state and output equations are of the form

$$\mathbf{x}_{k+1} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \mathbf{x}_k + \mathbf{w}_k \tag{50}$$

$$\mathbf{z}_k = \begin{bmatrix} 1 & 0 \end{bmatrix} \mathbf{x}_k + \mathbf{v}_k. \tag{51}$$

Further, we assume that the process and observation noise are given by

$$\mathbf{Q}_k = \begin{bmatrix} \Delta T^3/3 & \Delta T^2/2 \\ \Delta T^2/2 & \Delta T \end{bmatrix} \sigma_q^2 \qquad \mathbf{R}_k = \sigma_r^2. \tag{52}$$

We will take $\sigma_q^2 = 0.01$, $\sigma_r^2 = 0.1$ and assume that the vehicle starts from rest so that $\mathbf{x}_{0|0} = [0, 0]^T$. Figure 5 shows the true position and velocity and observations for a run of 100 samples computed from the system equations using a pseudo-random number generator to generate normally distributed random numbers for the variances $\sigma_r^2$ and $\sigma_q^2$.

## 2.2 Performance under ideal modelling conditions

We first consider the performance of the Kalman filter under ideal modelling conditions meaning that the system model is known precisely as are the process and noise models.

Figure 6 shows the position track achieved by applying the Kalman filter. Figure 6a shows the predicted and estimated positions together with the measurements for the complete track. A close-up of the initial track is shown in Figure 6b. The key thing to note here is that the updated estimate $\hat{\mathbf{x}}_{k+1|k+1}$ always lies between the prediction $\hat{\mathbf{x}}_{k+1|k}$ and the measurement $\mathbf{z}_{k+1}$. This follows from the fact that the update is a weighted sum of the prediction and the measurement (see lecture 1). Figure 6c shows a close-up of the estimator when it is in the steady-state. In this case, the weights (i.e. the Kalman gain) used in the update are approximately constant.

Figure 7 shows the velocity track achieved by applying the Kalman filter. No measurement is made of the velocity state so estimates are produced via the cross-correlation between the velocity and position (ie through $\mathbf{P}$).

### 2.2.1 Steady-state performance

Figure 8 shows the predicted and estimated error covariances for position and velocity. In particular, note that they tend to constant values as $k$ gets large.

Performance in the steady-state turns out to be dependent on the values chosen for the process and measurement noise covariance matrices, $\mathbf{Q}$ and $\mathbf{R}$.

Given that

$$\mathbf{P}_{k+1|k} = \mathbf{F}_k \mathbf{P}_{k|k} \mathbf{F}_k^T + \mathbf{Q}_k$$

and

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{P}_{k|k-1}\mathbf{H}_k^T[\mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^T + \mathbf{R}_k]^{-1}\mathbf{H}_k\mathbf{P}_{k|k-1}$$

we have

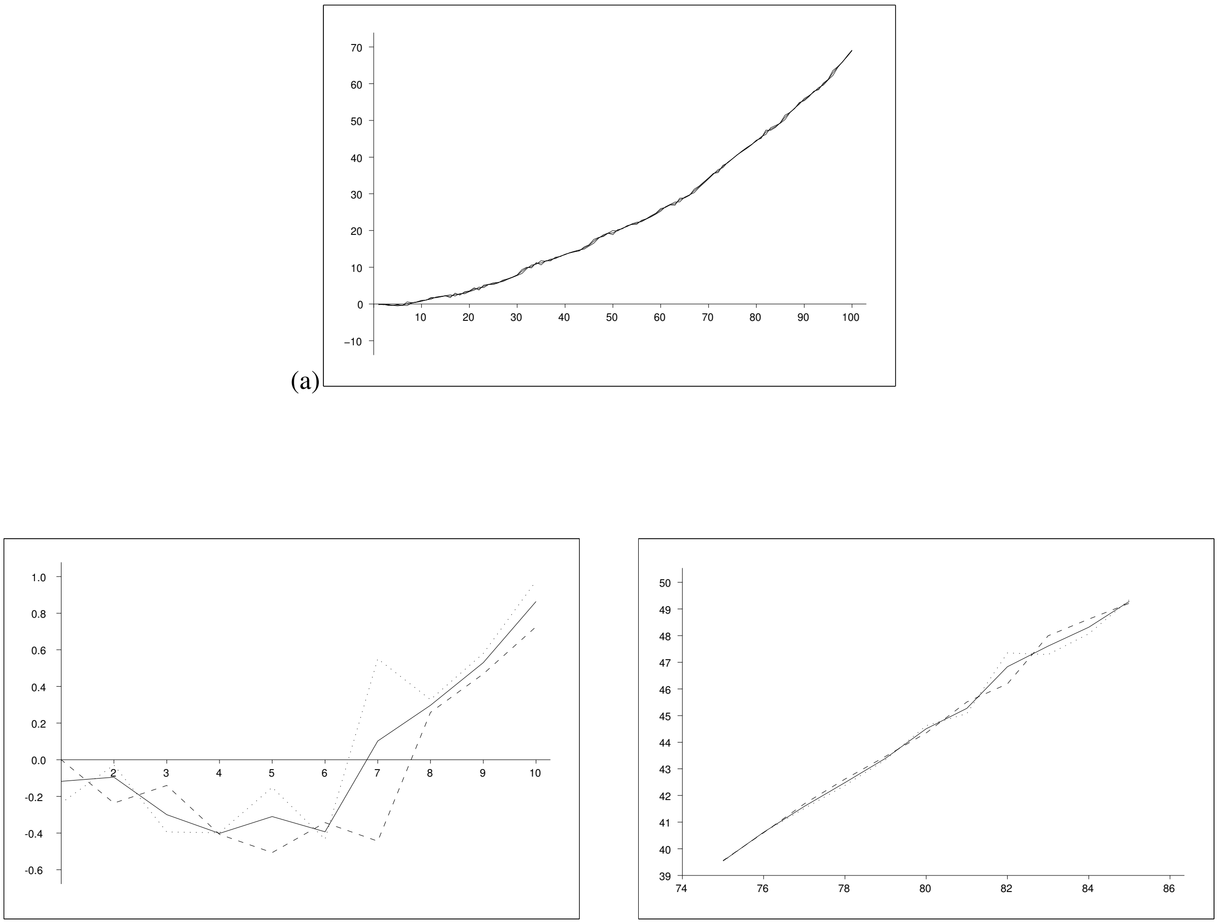
$$\mathbf{P}_{k+1|k} = \mathbf{F}_k[\mathbf{P}_{k|k-1} - \mathbf{P}_{k|k-1}\mathbf{H}_k^T[\mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^T + \mathbf{R}_k]^{-1}\mathbf{H}_k\mathbf{P}_{k|k-1}]\mathbf{F}_k^T + \mathbf{Q}_k \quad (53)$$

Equation (53) is known as the discrete-time **matrix Ricatti equation**. It turns out that if the system is time-invariant (i.e. $\mathbf{F}$, $\mathbf{G}$, and $\mathbf{H}$ are constant), and the measurement and process noise are stationary ($\mathbf{Q}$ and $\mathbf{R}$ are constant) then as $k \to \infty$ the solution to equation (53) converges to a positive definite matrix $\bar{\mathbf{P}}$ provided that the system model is completely **observable** and completely **controllable** (for precise definitions of these see Jacobs). The corresponding gain matrix $\bar{\mathbf{K}} = \bar{\mathbf{P}}\mathbf{H}^T\mathbf{S}^{-1}$ will also be constant and called the **steady-state gain**.

The importance of this result is that in some applications you can assume that the Kalman filter works under steady-state conditions. In this case you fix the value of $\bar{\mathbf{P}}$ and hence $\bar{\mathbf{K}}$ from the start and initial conditions do not need to be specified. Since $\mathbf{K}$ is now fixed it means that considerable computational saving can be made since $\mathbf{K}$ does not have to be recomputed at each time step.

### 2.2.2 Initialisation

Recall that part of the requirements for a Kalman filter is specification of initial conditions. Therefore, when considering implementation of a Kalman filter an important concern is how to set (guess!)



(a)

(b)

(c)

Figure 6: *Position track: (a) predicted, estimated positions and observation ; (b) initial track (zoom of (a)); (c) steady-state track (zoom of (a).*

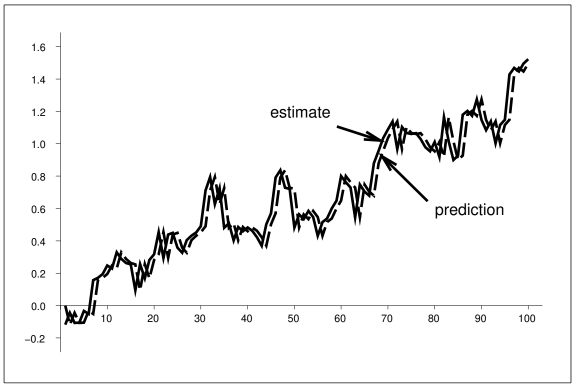



Figure 7: *Velocity track.*

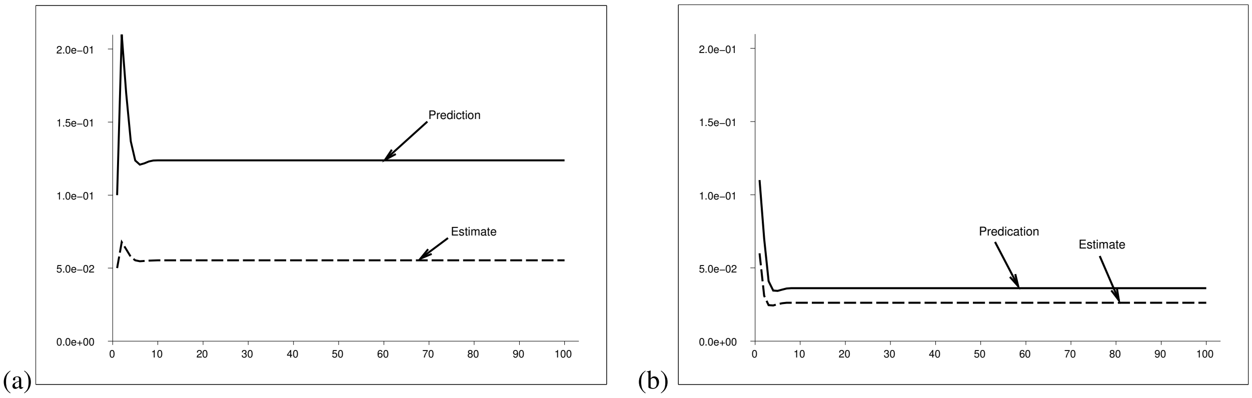



Figure 8: *(a) Position error covariance; (b) Velocity error covariance.*

values for $\mathbf{x}_{0|0}$ and $\mathbf{P}_{0|0}$ as they are not known. The obvious question to ask then, is does it matter how good (or bad) your guess is?

One possibility is to initialise the state vector from the measurements

$$\hat{\mathbf{x}}_{0|0} = \begin{bmatrix} \mathbf{z}_0 \\ \frac{\mathbf{z}_0 - \mathbf{z}_{-1}}{\Delta t} \end{bmatrix}$$

and a simple way to initialise the state covariance matrix is to set it to be a multiple $R$ of the process noise matrix

$$\mathbf{P}_{0|0} = R\mathbf{Q}_k,$$

where $R$ is a constant (typically $R = 10$).

Figure 9 illustrates the effect that changing initialisation parameters has on long term Kalman filter performance. Note that regardless of the initial values both $\hat{\mathbf{x}}$ and $\mathbf{P}$ tend to constant values in a few iterations.

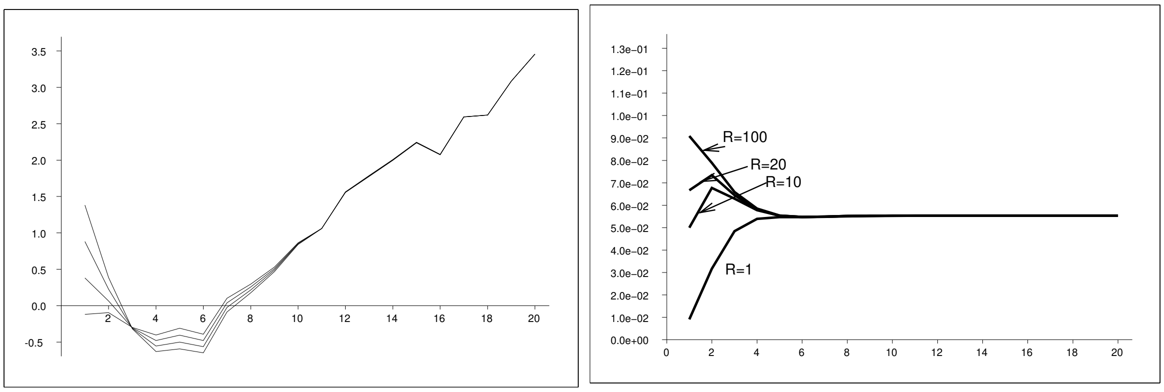
More formally, it can be shown that provided that the system is observable and controllable the error due to poor initialisation tends to zero as $k \to \infty$. Finally note that although good initialisation is desirable for a linear Kalman filter it is not essential (the estimator merely takes longer to settle down). However, good initialisation is critical in the implementation of Kalman filters for nonlinear system models (see final lecture).

### 2.2.3 Checking consistency

Since in practice we can not measure performance with respect to the state error measures (since we don't know the true state values) how do we check that the filter is performing correctly? The answer is that we can define filter performance measures in terms of the **innovation**

We know that if the filter is working correctly then $\boldsymbol{\nu}_k$ is zero mean and white with a covariance $\mathbf{S}_k$ (see previous lecture). So we can verify that the filter is consistent by applying the following two procedures.

1. check that the innovation is consistent with its covariance by verifying that the magnitude of the innovation is bounded by $\pm 2\sqrt{\mathbf{S}_k}$.




(a) (b)

Figure 9: *(a) Effect of using different position initialisation values; (b) Effect of changing $R$ to initialise $\mathbf{P}_{0|0}$.*

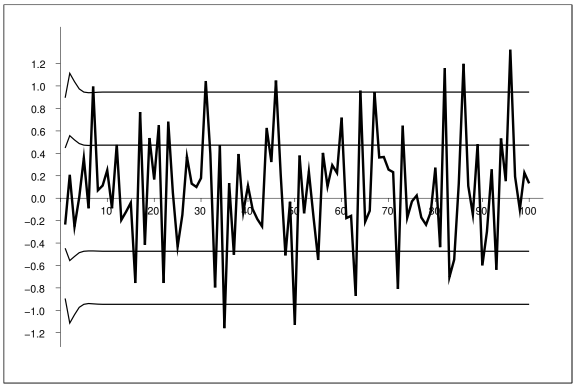


Figure 10: *Innovation and innovation standard deviation bounds.*

2. verify that the innovation is unbiased and white. This can be verified by using hypothesis testing ($\chi^2$ test).

**Test 1 - Innovation magnitude bound test:** Figure 10 shows the innovation sequence together with the $\pm\sigma$ and $\pm 2\sigma$ bounds on its magnitude. The figure would seem to indicate that the innovation is unbiased and approximately $95\%$ of the values lie within the $\pm 2\sigma$ bound as required. This simple test is sometimes sufficient to check filter consistency. However, in practice it is more usual to also apply the test discussed next.

**Test 2 - Normalised innovations squared $\chi^2$ test:** To test for unbiasedness we compute the normalised innovations squared

$$q_{k+1}(i) = \boldsymbol{\nu}_{k+1}(i)\mathbf{S}_{k+1}^{-1}(i)\boldsymbol{\nu}_{k+1}(i) \tag{54}$$

for a sequence of $i$ trials of a Kalman filter. If the filter assumptions are met then the $q_{k+1}(i)$ are each $\chi^2$ in $m$ degrees of freedom, where $m = 1$ in our case (the dimension of the measurement

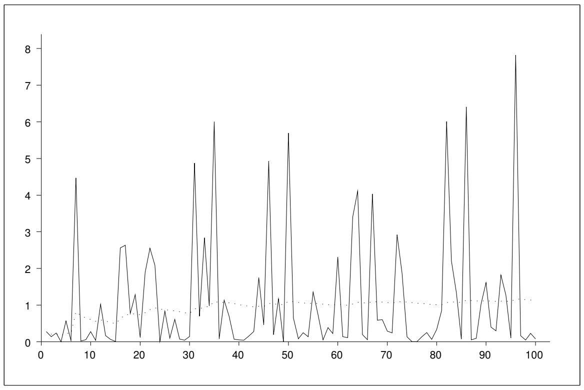



Figure 11: *Normalised innovation and moving average.*

vector). Thus

$$E[q_{k+1}] = m \tag{55}$$

This provides the test for unbiasedness. To estimate the mean we need to have $N$ independent samples of $q_{k+1}(i), i = 1, \ldots N$. The mean of this sequence,

$$\bar{q}_{k+1} = \frac{1}{N} \sum_{i=1}^{N} q_{k+1}(i)$$

can be used as a test statistic since $N\bar{q}_{k+1}$ is $\chi^2$ on $Nm$ degrees of freedom.

In our case, however, we can exploit the fact that the innovations are **ergodic** to estimate the sample mean from the time average for a long sequence (ie. the moving average) rather than an ensemble average. Thus we can estimate the mean as,

$$\bar{q} = \frac{1}{N} \sum_{k=1}^{N} q_k \tag{56}$$

from a single run of a Kalman filter. Figure 11 shows the normalised innovation and the moving average of the innovation. The latter tends to $1.0$ as $k$ gets large. To test unbiasedness we need to verify that $\bar{q}$ lies in the confidence interval $[r_1, r_2]$ defined by the hypothesis $H_0$ that $N\bar{q}$ is $\chi^2_{Nm}$ distributed with probability $1 - \alpha$. Thus we need to find $[r_1, r_2]$ such that

$$P(N\bar{q} \in [r_1, r_2] | H_0) = 1 - \alpha, \tag{57}$$

For the example we are considering, $N = 100$, $\bar{q} = 1.11$, and let $\alpha = 0.05$ (ie. define the two-sided $95\%$ confidence region). Using statistical tables we find that,

$$\begin{aligned} [r_1, r_2] &= \left[\chi^2_{100}(0.025), \chi^2_{100}(0.975)\right], \\ &= [74.22, 129.6] \end{aligned}$$

The hypothesis is indeed acceptable for this example.

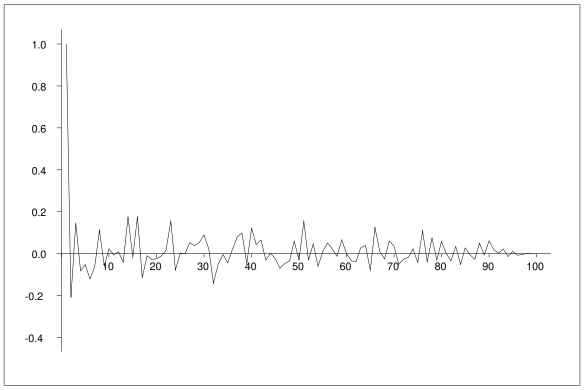



Figure 12: *Autocorrelation of the innovation.*

**Test 3 - Innovation whiteness (autocorrelation) test:** To test for whiteness we need to prove that

$$E[\boldsymbol{\nu}_i^T \boldsymbol{\nu}_j] = \mathbf{S}_i \delta_{ij} \tag{58}$$

We can test this by checking that everywhere except where $i = j$, the statistic defined by Equation (58) is zero within allowable statistical error. Again, we can exploit ergodicity to redefine the test statistic as a time-averaged correlation

$$r(\tau) = \frac{1}{N} \sum_{k=0}^{N-\tau-1} \boldsymbol{\nu}_k^T \boldsymbol{\nu}_{k+\tau} \tag{59}$$

The autocorrelation is usually normalised by $r(0)$. Figure 12 shows the normalised auto-correlation of the innovation for the example we are considering. Note that it peaks at $\tau = 0$ and everywhere else is distributed randomly about zero. We can test that the oscillations about zero are random by estimating the variance of the test statistic. For large enough $N$ we can assume that $r(\tau)$ is normally distributed with mean zero and variance $1/N$. Then we can compute the $2\sigma$-gate as $\pm 2/\sqrt{N}$ and check that at least $95\%$ of the values fall within this confidence region. Again in our example the autocorrelation satisfies the hypothesis.

## 2.3 Model validation

So far we have only considered the performance of a Kalman filter when both the system model and noise processes are known precisely. A Kalman filter may not perform correctly if there is either modelling or noise estimation error or both. Here we discuss the causes and identify most of the important techniques used to control a Kalman filter from diverging. We consider two types of error and their characteristics;

1. Error in the process and observation noise specification.
2. Error in the modelling of system dynamics (process model).

The three tests that we introduced in the last section will be used as a basis for trying to tell when something has gone wrong with the filter.

### 2.3.1 Detecting process and observation noise errors

The example used in the previous section will be used to illustrate general characteristics that are observed when the process and observation noise are under- and over-estimated.

Some **matlab** code to generate a simulation sequence.

```
%%% Matlab script to generate some "true" data for later assessment
%%% Generates:
%%%     x: the state history which evolves according to
%%%              x(k+1) = Fx(k) + w(k)
%%%     w: the process noise history (randomly generated)
%%%     z: a set of observations on the state corrupted by noise
%%%     v: the noise on each observation (randomly generated)

N = 100;

delT = 1;
F = [ 1 delT
      0   1  ];
H = [ 1 0 ];
sigma2Q = 0.01;
sigma2R = 0.1;
Q = sigma2Q *  [ delT^3/3  delT^2/2
                 delT^2/2  delT ];
P = 10*Q;
R = sigma2R * [ 1 ];

x = zeros(2,N);
w = zeros(2,N);
z = zeros(1,N);
v = zeros(1,N);
for i=2:N
    w(:,i) = gennormal([0;0], Q);          % generate process noise
    x(:,i) = F*x(:,i-1) + w(:,i);          % update state
    v(:,i) = gennormal([0], R);            % generate measurement noise
    z(:,i) = H * x(:,i) + v(:,i);           % get "true" measurement
end

plot(x(1,:));
```

The **matlab** code to process the sequence and generate the various graphs is given below.

```
%%% Matlab script to assess Kalman filter performance
%%% The script assumes the existence of a vector z of
%%% noise corrupted observations

N = length(z);          % number of Klamn filter iterations

Qfactor = 1;            % process noise mult factor
Rfactor = 10;          % measurement noise mult factor

delT = 1;               % time step
F = [ 1 delT            % update matrix
```

```
      0   1  ];
H = [ 1 0 ];            % measurement matrix

sigmaQ = Qfactor*sqrt(0.01);
sigmaR = Rfactor*sqrt(0.1);
Q = sigmaQ^2 * [ 1/3  1/2     % process noise covariance matrix
                 1/2  1 ];
P = 10*Q;
R = sigmaR^2 * [ 1 ];         % measurement noise covariance

xhat = zeros(2,N);        % state estimate
nu = zeros(1,N);          % innovation
S = zeros(1,N);           % innovation (co)variance
q = zeros(1,N);           % normalised innovation squared

for i=2:N
    [xpred, Ppred] = predict(xhat(:,i-1), P, F, Q);
    [nu(:,i), S(:,i)] = innovation(xpred, Ppred, z(i), H, R);
    [xhat(:,i), P] = innovation_update(xpred, Ppred, nu, S, R);
    q(:,i) = nu(:,i)'*inv(S(:,i))*nu(:,i);
end

sumQ = sum(q)       % determine Sum q which is Chi^2 on N d.o.f.
r = xcorr(nu);      % get autocorrealtion of innovation

plot(xhat(1,:));    % plot state estimate
pause;

plot(nu)            % plot innovation and 2sigma confidence interval
hold on;
plot(2*sqrt(S),'r');
plot(-2*sqrt(S),'r');
hold off;
pause;

plot(q);            % plot normalised innovation squared
pause;

plot(r(N:2*N-1)/r(N));  % plot autocorr of innovation (normalised)
```

**Under-estimating $\sigma_q$:** Refer to Figure 13. This illustrates the performance tests for the case when the process noise is under-estimated by a factor of 10.

A greater quantity of innovations than expected (i.e. $> 5\%$) fall outside the $2\sigma$ gate (obvious even from visual inspection).

The normalised innovations squared are larger than expected and the sample mean falls outside the confidence bound defined by the $\chi^2$ test (for my trial the value came to 492.34/100 which is clearly above the 95% confidence region [74.22/100,129.6/100] computed above). This tells us that the combined process and measurement noise levels are too low, i.e. too little weight is placed on current measurements in the update process.

The autocorrelation sequence shows time correlations.

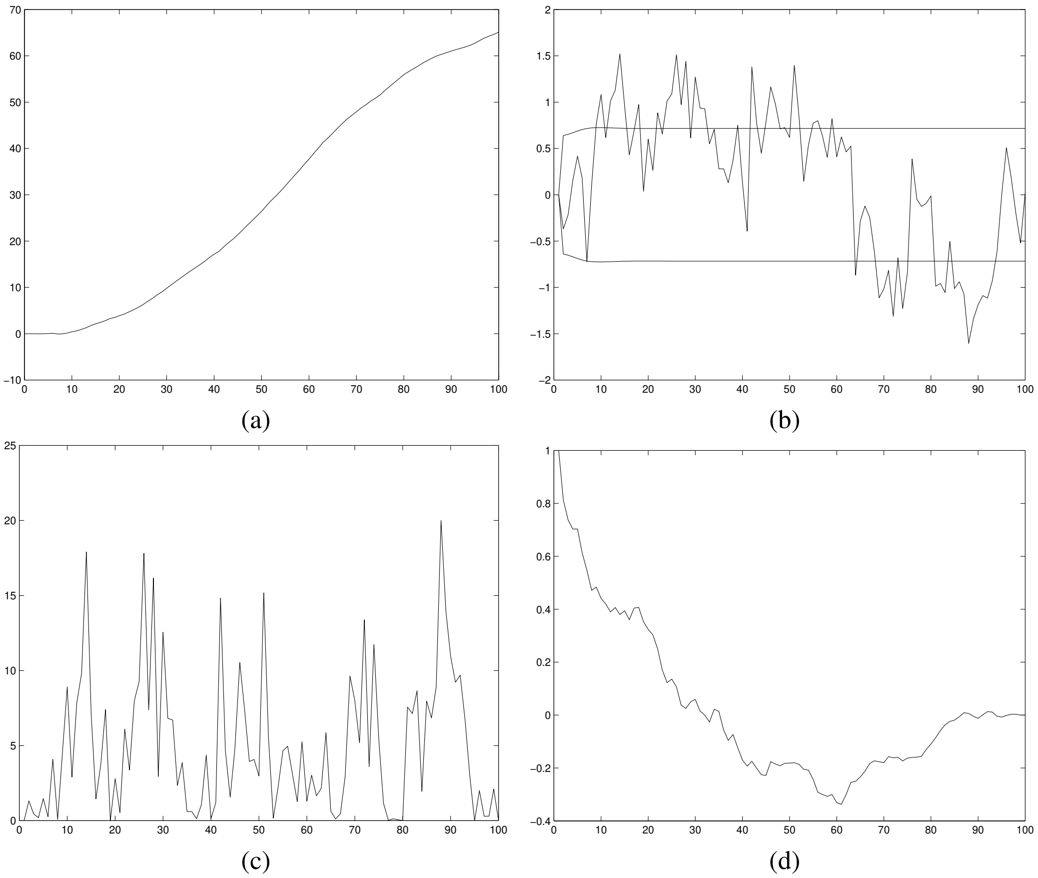


Figure 13: *Effect of underestimating $\sigma_q$ by a factor of* 10*. (a) state estimates; (b) innovation sequence; (c) normalised innovations squared; (d) normalised autocorrelation of the innovation sequence.*

**Over-estimating $\sigma_q$:** Refer to Figure 14. This illustrates the performance tests for the case when the process noise is over-estimated by a factor of 10. The innovations are well within the required bounds.

The normalised innovations squared are smaller than expected and the sum (32.81, or eqivalently the average) falls below the confidence bound defined by the $\chi^2$ test. This tells us that the combined process and measurement noise levels is too high.

The autocorrelation sequence shows no obvious time correlations.

**Under-estimating $\sigma_r$:** Refer to Figure 15. This illustrates the performance tests for the case when the measurement noise is under-estimated by a factor of 10.

The innovations exceed the $2\sigma$ bounds more often than allowable.

The normalised innovations squared are larger than expected and the sample mean (3280/100) falls outside the confidence bound [0.74,1.3] defined by the $\chi^2$ test. This tells us that the combined process and measurement noise levels is too low.

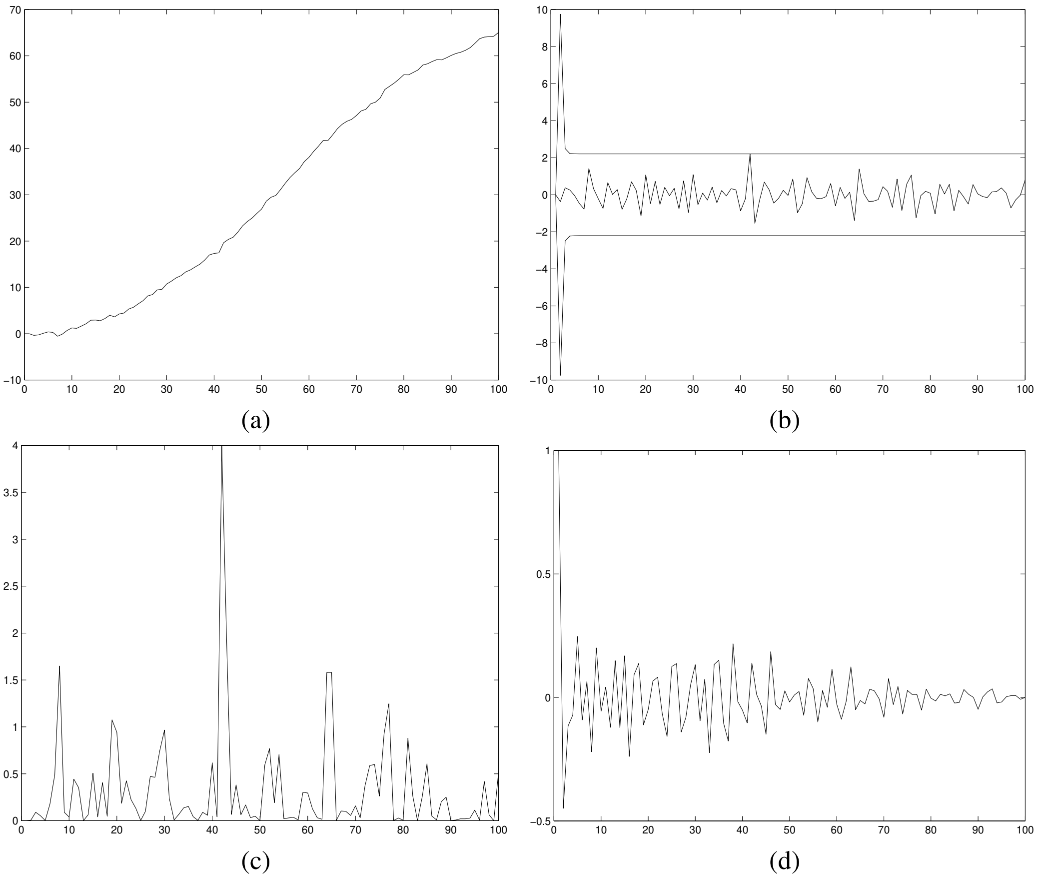



Figure 14: *Effect of overestimating $\sigma_q$ by a factor of* $10$*. (a) state estimates; (b) innovation sequence; (c) normalised innovations squared; (d) normalised autocorrelation of the innovation sequence.*

The autocorrelation sequence shows no obvious time correlations.

**Over-estimating $\sigma_r$:** Refer to Figure 16. This illustrates the performance tests for the case when the measurement noise is over-estimated by a factor of 10.

The innovations are below the $2\sigma$ bounds.

The normalised innovations squared are smaller than expected and the sample mean (4.95/100) falls outside the confidence bound defined by the $\chi^2$ test. This tells us that the combined process and measurement noise levels is too high.

The autocorrelation sequence shows time correlations.

**General observations:**

1. If the ratio of process to measurement noise is too low the innovation sequence becomes correlated.

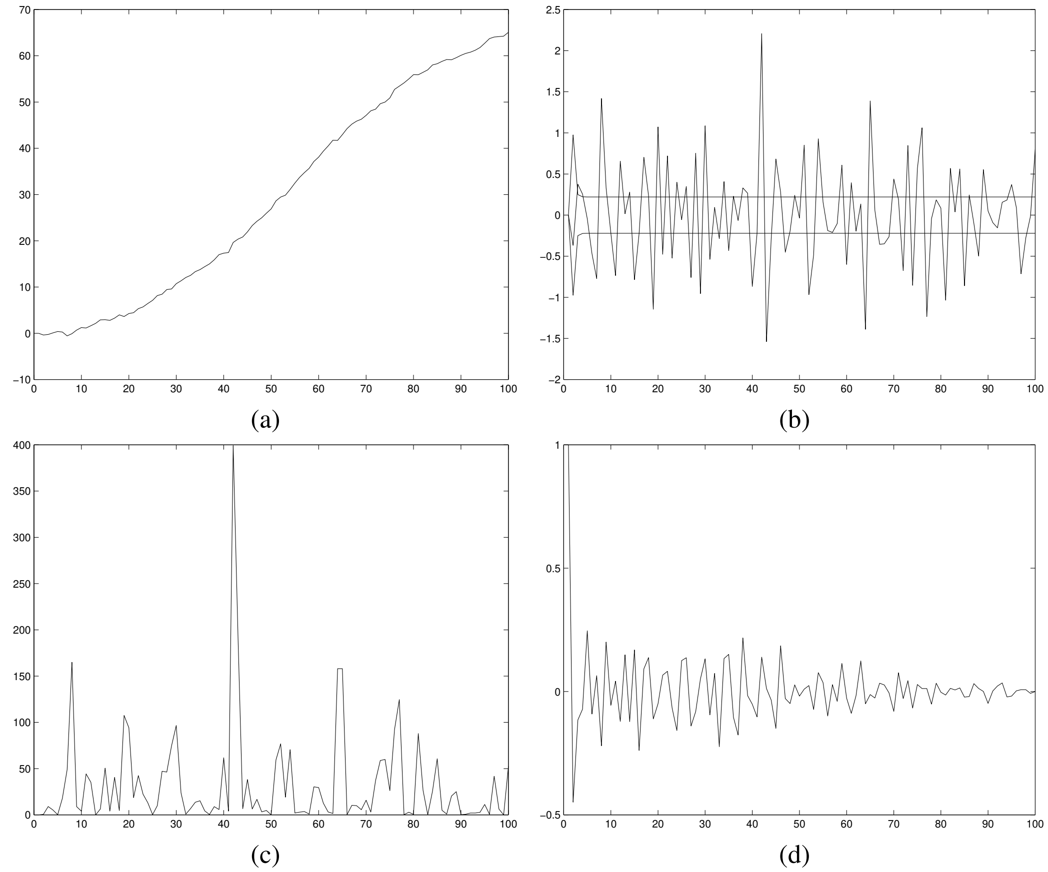



Figure 15: *Effect of underestimating $\sigma_r$ by a factor of* 10*. (a) state estimates; (b) innovation sequence; (c) normalised innovations squared; (d) normalised autocorrelation of the innovation sequence.*

2. The absolute values of the process and measurement noise can be set by adjusting their values so that the $\chi^2$ innovation test is satisfied.

3. In the example shown here, tuning is much more sensitive to changes in the measurement noise rather than the process noise. In this example, this is because measurement noise affects position, process noise only affects velocity (refer to the continuous system model in Lecture 1).

#### 2.3.2 Detecting process modelling errors

We now consider what happens if we try to apply an estimator to measurement data that doesn't fit the model - the so-called *mis-matched filter problem*.

Specifically, we consider the case of using a constant-velocity Kalman filter to track a particle which has a true motion trajectory defined by a constant-acceleration model. Thus, the true motion is

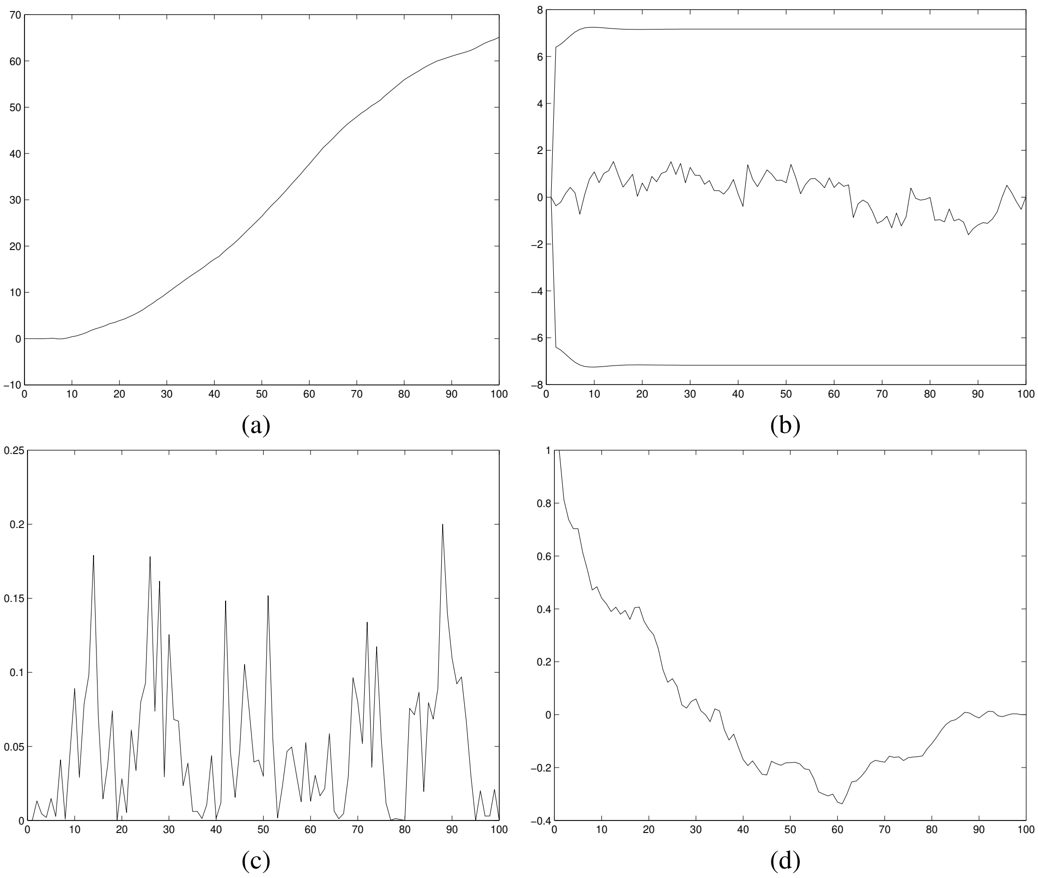


Figure 16: *Effect of overestimating $\sigma_r$ by a factor of* $10$*. (a) state estimates; (b) innovation sequence; (c) normalised innovations squared; (d) normalised autocorrelation of the innovation sequence.*

described by the transition equation

$$\mathbf{x}_{k+1} = \mathbf{F}\mathbf{x}_k + \mathbf{w}_k \tag{60}$$

where the state transition matrix is

$$\mathbf{F} = \begin{bmatrix} 1 & \Delta T & \Delta T^2/2 \\ 0 & 1 & \Delta T \\ 0 & 0 & 1 \end{bmatrix} \tag{61}$$

with

$$\mathbf{Q} = E[\mathbf{w}_k \mathbf{w}_k^T] = \begin{bmatrix} \Delta T^5/20 & \Delta T^4/8 & \Delta T^3/6 \\ \Delta T^4/8 & \Delta T^3/3 & \Delta T^2/2 \\ \Delta T^3/6 & \Delta T^2/2 & \Delta T \end{bmatrix} \sigma_q^2 \tag{62}$$

Figure 17 shows the result of applying the constant-velocity filter to the constant-acceleration model where the filter noise parameters were $\sigma_q = 0.01$ and $\sigma_r = 0.1$.

Observe that the innovation behaves like a first order Gauss-Markov process (recall this implies that in continuous-time $dx/dt + A(t)x = w$, where $w$ is white noise). The normalised squared values

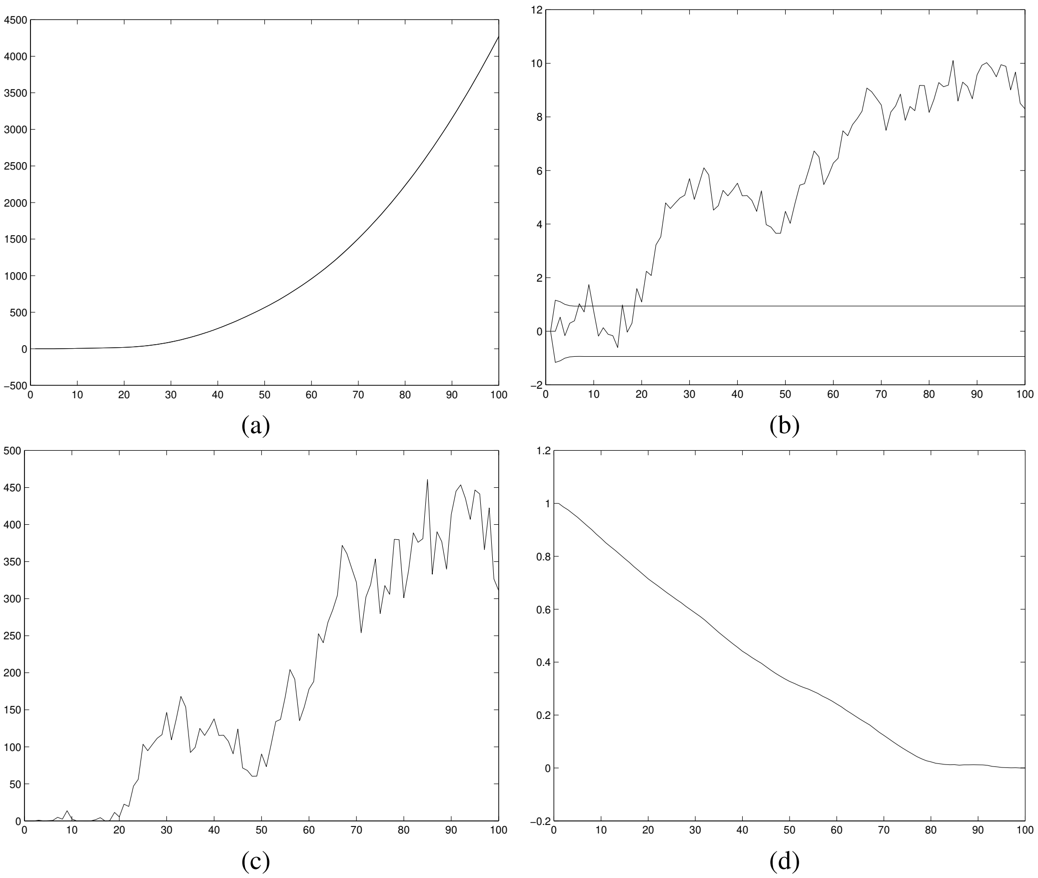



Figure 17: *Performance tests for an unmatched filter. (a) state estimates; (b) innovation sequence; (c) normalised innovations squared; (d) normalised autocorrelation of the innovation sequence.*

show a substantial drift in the mean and is not stationary. The autocorrelation reduces exponentially in time - again typical of a first-order Gauss-Markov process.

**Boosting Q to reduce effects of modelling errors:** one obvious thing to try in order to reduce the effects of modelling errors is to boost the magnitude of the process noise **Q** artificially to take into account unmodelled errors. Recall that this should boost the value of the Kalman gain and hence let the estimate follow the measurements more closely. The result of doing this where the process noise was increased by a factor of 10 is shown in Figure 18. Some improvement is seen but this has not totally compensated for the process model error.

# 3 The Continuous-Time Kalman Filter

So far we have considered the discrete-time formulation of the Kalman filter. This is the version which finds the most wide-spread application in practice. The Kalman filter estimation approach can also be derived for continuous-time. This is what we look at in this section. It is interesting to

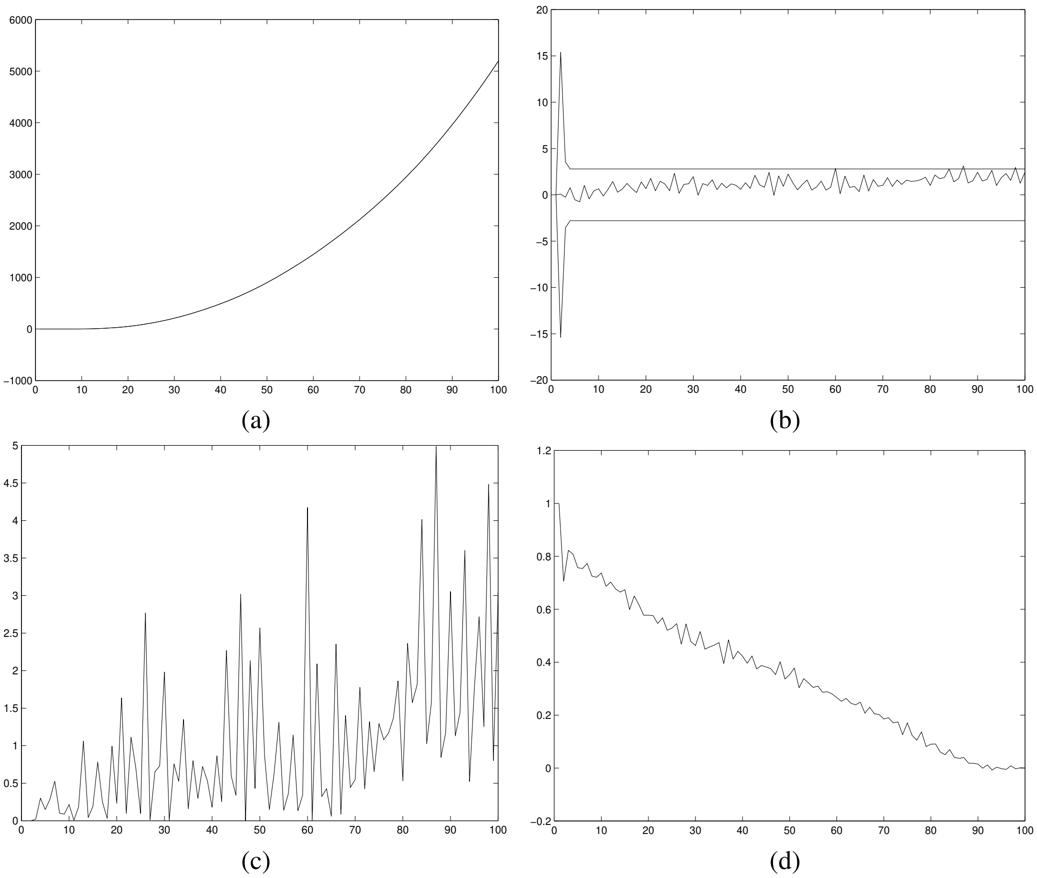



Figure 18: *Performance tests for an unmatched filter with process noise boosted to compensate. (a) state estimates; (b) innovation sequence; (c) normalised innovations squared; (d) normalised autocorrelation of the innovation sequence.*

study the continuous-time Kalman filter for two principal reasons;

1. to understand the asymptotic behaviour of discrete-time Kalman filters and,
2. to provide insight into the relationship between Kalman filtering and Wiener filtering.

We consider both of these factors in this section. In reading what follows you may find it useful to refer back to the equivalent results that we have previously derived for the discrete-time case in Sections 1 and 2.

## 3.1 Models and assumptions

The continuous-time Kalman filter is the limiting case of the discrete-time Kalman filter as the sample time becomes infinitely small. We will not derive the continuous equations here merely state the key equations (for example, see Gelb p119 onwards, or Brown Chapter 7 for a derivation).

In the continuous case the **system model** is given by

$$\frac{\partial \mathbf{x}(t)}{\partial t} = \mathbf{F}(t)\mathbf{x}(t) + \mathbf{w}(t) \tag{63}$$

where the process noise has covariance $\mathbf{Q}(t)$.

The **measurement model** is given by

$$\mathbf{z}(t) = \mathbf{H}(t)\mathbf{x}(t) + \mathbf{v}(t) \tag{64}$$

where the measurement noise has covariance $\mathbf{R}(t)$. We assume that the inverse $\mathbf{R}^{-1}(t)$ exists.

Recall from the derivation of the discrete-time estimator that we also need to specify initial conditions for the state and its error covariance. Thus we assume that the initial conditions

$$E[\mathbf{x}(0)] = \hat{\mathbf{x}}(0), \qquad E[(\mathbf{x}(0) - \hat{\mathbf{x}}(0))(\mathbf{x}(0) - \hat{\mathbf{x}}(0))^T] = \mathbf{P}(0) \tag{65}$$

are given.

## 3.2 Kalman filter equations

State estimation is governed by the equation

$$\dot{\hat{\mathbf{x}}}(t) = \mathbf{F}(t)\hat{\mathbf{x}}(t) + \mathbf{K}(t)[\mathbf{z}(t) - \mathbf{H}(t)\hat{\mathbf{x}}(t)] \tag{66}$$

Error covariance propagation is determined by the differential equation

$$\dot{\mathbf{P}}(t) = \mathbf{F}(t)\mathbf{P}(t) + \mathbf{P}(t)\mathbf{F}^T(t) + \mathbf{Q}(t) - \mathbf{K}(t)\mathbf{R}(t)\mathbf{K}^T(t) \tag{67}$$

which is known as the **matrix Riccati equation**. This matrix differential equation has been studied extensively and an analytic solution exists for the constant parameter case.

Here the Kalman gain matrix is defined by

$$\mathbf{K}(t) = \mathbf{P}(t)\mathbf{H}^T(t)\mathbf{R}^{-1}(t) \tag{68}$$

In summary Equations 66, 67 and 68 together with the initial conditions specified by Equation 65 describe the **continuous-time Kalman filter** algorithm. You should compare these equations with the equivalent results for the discrete-time case.

## 3.3 Example

Consider the problem of estimating the value of a constant signal $x(t)$ given measurements corrupted by Gaussian white noise which is zero-mean and has constant spectral density $\alpha$. Derive (1) the continuous-time Kalman filter; and (2) the discrete-time Kalman filter assuming a sampling time interval of $\Delta T$.

**Continuous-time solution:** The state-space model equations of the problem are

$$\dot{x}(t) = 0 \tag{69}$$

$$z(t) = x(t) + v(t), \quad v \sim N(0, \alpha) \tag{70}$$

The scalar Riccati equation (Equation 67) governs error covariance propagation and is given by

$$\dot{p} = fp + pf + q - krk \tag{71}$$

where $k = ph/r$. In this problem $f = q = 0, h = 1, r = \alpha$. Therefore

$$\dot{p} = -k^2 r \quad \text{and} \quad k = p/\alpha$$

Substituting for $k$ and integrating Equation 71 we can solve for $p$ as follows

$$\begin{aligned}
\dot{p} &= \frac{-p^2}{\alpha} \\
\int_{p_0}^{p} \frac{dp}{p^2} &= -\frac{1}{\alpha} \int_0^t dt, \\
p &= p_0 (1 + (p_0/\alpha)\, t)^{-1}
\end{aligned} \tag{72}$$

Hence the Kalman gain is given by

$$k = \frac{p}{\alpha} = (p_0/\alpha)[1 + (p_0/\alpha)t]^{-1}$$

and the state estimation by

$$\hat{\dot{x}}(t) = (p_0/\alpha)[1 + (p_0/\alpha)t]^{-1}(z(t) - \hat{x}(t))$$

Finally, note that as $t \to \infty$, $k \to 0$ and the estimate reaches a constant value.

**Discrete-time solution:** Let us consider what the result would have been if rather than analyse the continuous-time measurements we had sampled the signals at instants in time $t = k\Delta T, k = 0, 1, 2, \ldots$ (satisfying the Nyquist criterion of course).

In this case, the discrete-time space-model is

$$x(k+1) = x(k) \tag{73}$$

$$z(k) = x(k) + v(k), \quad v(k) \sim N(0, \alpha) \tag{74}$$

We have $\mathbf{F}(k) = \mathbf{H}(k) = 1$, $\mathbf{Q}(k) = 0$ and $\mathbf{R}(k) = \alpha$.

The predicted state and error covariance are given by (see Section 1)

$$\hat{x}(k+1|k) = \hat{x}(k|k) \quad \text{and} P(k+1|k) = P(k|k)$$

Using this result the update equation for the error covariance is

$$P(k+1|k+1) = P(k|k) - K(k+1)[\alpha + P(k|k)]K(k+1) \tag{75}$$

where $K(k+1) = P(k|k)[P(k|k) + \alpha]^{-1}$. Making this substitution for $K(k+1)$ into Equation 75 gives

$$P(k+1|k+1) = P(k|k)\left[\frac{1}{1 + P(k|k)/\alpha}\right]. \tag{76}$$

If our initial error covariance was $P_0$ then it follows from Equation 76 that, at time $k+1$

$$P(k+1|k+1) = P_0 \left[\frac{1}{1 + kP_0/\alpha}\right] \tag{77}$$

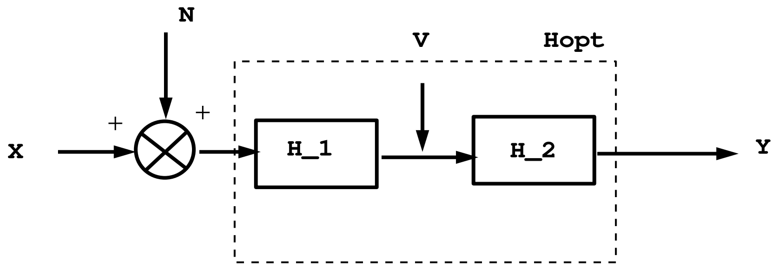



Figure 19: *Wiener filter.*

Hence the state update equation is

$$\hat{x}(k+1|k+1) = \hat{x}(k|k) + K(k+1)\left[z(k+1) - \hat{x}(k|k)\right] \tag{78}$$

where

$$K(k+1) = \frac{(P_0/\alpha)}{1 + (kP_0/\alpha)}$$

Compare this with the result for the continuous-time case. Note again that as $k \to \infty$, $\hat{x}(k+1|k+1)$ tends to a constant value.

One final point: unfortunately only simple types of continuous-time problems such as the example given above can be solved analytically using the covariance equations. For more complicated problems, numerical methods are required. This is the main reason why the continuous-time Kalman filter has not found wide-spread use as an estimation method.

## 3.4 Relation to the Wiener Filter

In this section we consider the connection of the Kalman filter to the Wiener filter. We will not derive the optimal filter from first principles. Here our interest is in the relation to the Kalman filter.

**Problem statement:** The Wiener filter is the linear minimum variance of error estimation filter from among all time-invariant filters. Briefly, in one-dimension, we consider the problem of how to find the optimal impulse function $h(t)$ which gives the best estimate of a signal $\hat{s}(t)$ where the information available is a corrupted version of the original,

$$z(t) = s(t) + n(t). \tag{79}$$

Here $n(t)$ is additive noise. If (1) we know the power spectral densities of the signal and noise, $S_X(s)$ and $S_N(s)$ respectively; and (2) the observations have been acquired for sufficient time length so that the spectrum of $z(t)$ reflects both $s(t)$ and $n(t)$ then the goal is to find the optimal filter response $h(t)$ to recover the underlying signal.

**Wiener filter solution:** The Wiener filter solution to this problem is to find the transfer function corresponding to $h(t)$ using frequency response methods. It can be shown (see for example Brown chapter 4) that if the signal and noise are uncorrelated then the Wiener optimal filter takes on the form

$$H^W_{opt}(s) = \frac{S_X(s)}{S_X(s) + S_N(s)} \tag{80}$$

where $S_X(s)$ and $S_N(s)$ are the power spectral densities of the signal and noise. However, the filter defined by Equation 80 defines a non-causal filter, meaning that the output depends on future values of the input $z(t)$ as well as the past, or

$$h(\tau) \neq 0 \quad \text{for some } \tau \leq 0$$

For "real-time" operation this filter is not physically realisable (however it can be worth studying for off-line processing applications). To generate a **causal** filter we need to define the filter such that it depends only on past and current values of the input $z(t)$, thus,

$$h(\tau) = 0 \quad \text{for all } \tau \leq 0$$

Thus we want our optimal filter $H^W_{opt}(s)$ to satisfy this. The key is that if a transfer function $F(s)$ has only poles in the left half plane then its inverse $f(t)$ is a positive time function (proof omitted). We can make $H^W_{opt}(s)$ have this property by doing the following.

Consider the denominator of Equation 80. It can be separated into a part containing poles and zeros in the left hand plane $[S_X + S_N]_-$ and a part containing only poles and zeros in the $[S_X + S_N]_+$. This process is called **spectral decomposition**.

Let

$$H_1(s) = \frac{1}{[S_X + S_N]_-}, \quad H_2(s) = \frac{S_X}{[S_X + S_N]_+} \tag{81}$$

Then we can consider $H_{opt}$ as being the cascade of two filters, $H_1(s)$ (causal) and $H_2(s)$ (non-causal) as illustrated in Figure 19. Let $v(t)$ be the intermediate signal as shown. Then the power spectral density of $v(t)$, $S_V(s)$ is given by

$$\begin{aligned} S_V(s) &= |H_1|^2(S_N + S_X) \\ &= 1 \end{aligned} \tag{82}$$

In other words it is white noise and hence uncorrelated (independent of time). To make the whole filter causal we can ignore the negative tail of the impulse response corresponding to $H_2(s)$. We do this by taking partial fractions of $H_2(s)$ and discarding the terms with poles in the right hand plane. We denote this by $\{H_2(s)\}_-$. Hence the causal Wiener optimal filter is given by

$$H^W_{opt}(s) = \frac{1}{[S_X(s) + S_N(s)]_-} \left\{ \frac{S_X(s)}{[S_N(s) + S_X(s)]_+} \right\}_- \tag{83}$$

**Example:** As an illustration of this approach consider as an example

$$S_X(s) = \frac{1}{-s^2 + 1}, \quad S_N(s) = 1 \tag{84}$$

Then,

$$\begin{aligned} S_N(s) + S_X(s) &= \frac{-s^2 + 2}{-s^2 + 1} \\ &= \left[ \frac{(s + \sqrt{2})}{(s + 1)} \right] \left[ \frac{(-s + \sqrt{2})}{(-s + 1)} \right] \end{aligned} \tag{85}$$

It follows that,

$$H_1(s) = \frac{(s+1)}{(s+\sqrt{2})} \tag{86}$$

$H_2(s)$ is given by

$$\begin{aligned} H_2(s) &= \frac{\frac{1}{(-s^2+1)}(-s+1)}{(-s+\sqrt{2})} \\ &= \frac{1}{(s+1)(-s+\sqrt{2})} \end{aligned} \tag{87}$$

Re-writing this equation in partial fractions gives

$$H_2(s) = \frac{\sqrt{2}-1}{s+1} + \frac{\sqrt{2}-1}{-s+\sqrt{2}}$$

Hence

$$\{H_2(s)\}_- = \frac{\sqrt{2}-1}{(s+1)} \tag{88}$$

Combining Equations 86 and 88 gives the optimal Wiener filter as

$$H^W_{opt}(s) = H_1(s)\,\{H_2(s)\}_- = \frac{(\sqrt{2}-1)}{(s+\sqrt{2})} \tag{89}$$

or in the time domain

$$h^W_{opt}(t) = (\sqrt{2}-1)e^{-\sqrt{2}t}, \quad t \geq 0 \tag{90}$$

**The Kalman filter equivalent:** We can solve the same LMV problem by using an alternative approach based on state-space techniques that leads to a Kalman filter solution.

Specifically, we can re-write Equation 79 in terms of a state-space equation as

$$\mathbf{z}(t) = \mathbf{H}\mathbf{x}(t) + \mathbf{v}(t)$$

We can find $\mathbf{x}$ as the solution to the steady-state continuous-time Kalman filtering problem and hence find the (optimal) transfer function between the signal and measurement. More details of this approach are given next followed by a re-working of the example described earlier using this alternative approach.

We have to make some assumptions about the Kalman filtering problem before we start. Let us assume that the system and measurement equations have linear constant coefficients (i.e. $\mathbf{F}$ and $\mathbf{H}$ are time-invariant). Let us also assume that the noise processes are time-invariant ($\mathbf{Q}$ and $\mathbf{R}$ are constant). Further we assume that $\mathbf{F}$ is controllable, and $\mathbf{F}$ and $\mathbf{H}$ are observable. Under these conditions the Kalman filter will reach a steady-state condition where the error covariance matrix $\mathbf{P} \to \mathbf{P}_\infty$. This means that the matrix Riccati equation (Equation 67) now becomes

$$\begin{aligned} 0 &= \mathbf{F}\mathbf{P}_\infty + \mathbf{P}_\infty\mathbf{F}^T + \mathbf{Q} - \mathbf{K}_\infty\mathbf{R}\mathbf{K}^T_\infty, \\ &= \mathbf{F}\mathbf{P}_\infty + \mathbf{P}_\infty\mathbf{F}^T + \mathbf{Q} - \mathbf{P}_\infty\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\mathbf{P}_\infty, \end{aligned} \tag{91}$$

and state estimation is given by

$$\dot{\hat{\mathbf{x}}}(t) = \mathbf{F}\hat{\mathbf{x}}(t) + \mathbf{K}_\infty[\mathbf{z}(t) - \mathbf{H}\hat{\mathbf{x}}(t)] \tag{92}$$

where the steady-state gain is $\mathbf{K}_\infty = \mathbf{P}_\infty \mathbf{H}^T \mathbf{R}^{-1}$. Equations 91 and 92 define the **steady-state (stationary) continuous-time Kalman filter**.

Now let us re-arrange Equation 92

$$\dot{\hat{\mathbf{x}}}(t) - (\mathbf{F} - \mathbf{K}_\infty \mathbf{H})\hat{\mathbf{x}}(t) = \mathbf{K}_\infty \mathbf{z}(t)$$

Taking the Laplace transform and ignoring initial conditions (since we are in the steady state) we have

$$(s\mathbf{I} - \mathbf{F} - \mathbf{K}_\infty \mathbf{H})\hat{\mathbf{X}}(s) = \mathbf{K}_\infty \mathbf{Z}(s)$$

or

$$\frac{\hat{\mathbf{X}}(s)}{\mathbf{Z}(s)} = \mathbf{H}^K_{opt}(s) = \mathbf{K}_\infty \left[(s\mathbf{I} - \mathbf{F} - \mathbf{K}_\infty \mathbf{H})\right]^{-1} \tag{93}$$

This last equation defines a transfer function between the state estimate and the measurement which when multiplied by $\mathbf{H}$ gives the **Wiener optimal filter**,

$$\mathbf{H}^W_{opt}(s) = \mathbf{H}\mathbf{H}^K_{opt}(s). \tag{94}$$

Note that the *causal* Wiener filter and the continuous-time Kalman filter are equivalent under the assumptions of time-invariance and one can be determined from the other. The key difference is that one approach is based on state-space models and the other on frequency domain concepts (auto and cross correlations).

**Example:** Let us re-work the example we considered before using the Kalman filter approach.

The power spectral density of $x(t)$ is decomposed as

$$S_X(s) = \frac{1}{-s^2+1} = \frac{1}{s+1}\frac{1}{-s+1}$$

Thus the state-space model is

$$\begin{aligned} \dot{x}(t) &= -x(t) + w(t) && (95) \\ z(t) &= x(t) + v(t) && (96) \end{aligned}$$

Then this implies that $\mathbf{F} = -1, \mathbf{Q} = 1, \mathbf{H} = 1, \mathbf{R} = 1$.

Substituting these values into the steady-state Riccati equation gives

$$-2\mathbf{P}_\infty - \mathbf{P}^2_\infty + 1 = 0$$

or $\mathbf{P}_\infty = \sqrt{2} - 1$. Hence $\mathbf{K}_\infty = \mathbf{P}_\infty \mathbf{H}^T \mathbf{R}^{-1} = \sqrt{2} - 1$.

The optimal transfer function between the state and measurement is therefore

$$\mathbf{H}^K_{opt}(s) = \mathbf{K}_\infty \left[(s\mathbf{I} - \mathbf{F} - \mathbf{K}_\infty \mathbf{H})\right]^{-1} = \frac{\sqrt{2}-1}{s+\sqrt{2}}$$

giving

$$\mathbf{H}^{W}_{opt}(s) = \mathbf{H}\mathbf{H}^{K}_{opt}(s) = \frac{\sqrt{2}-1}{s+\sqrt{2}}$$

which agrees with Equation 89.

# 4 Further Topics in Kalman Filtering

This section deals with some variants of the discrete Kalman filter which prove useful when some of the assumptions of the conventional Kalman filter break down. Recall that three of the key problem areas for a Kalman filter are,

1. **Initialisation:** we assume that the initial state vector and its error covariance matrix are known.
2. **Modelling:** we assume that we have an accurate linear model of the process and measurement system.
3. **Noise:** we assume that the process and sensor noise processes are Gaussian.

In this section we look at how to deal with each of these problems.

We begin by considering the **information filter** which is a variant on the conventional Kalman filter which gives more accurate results when there is no information about the initial state.

Next we consider how to cope with modelling error. In most practical cases the linear equations describing the system and observation models of Equations 14 and 15 in Section 1 are not a good approximation to reality. Although, as we have seen, it is possible to detect that our assumptions about modelling and noise are invalid it is clear that what we need to do is extend the estimation approach to accommodate nonlinear models. Recall that for the Kalman filter algorithm, the estimate is the conditional mean of the state given all the measurements up to time $k+1$. We showed in Section 1 that under the assumption that the process was linear and the noise processes white this led to linear, recursive solution of the form,

$$\hat{\mathbf{x}}_{k+1|k+1} = \hat{\mathbf{x}}_{k+1|k} + \mathbf{K}_{k+1}[\mathbf{z}_{k+1} - \mathbf{H}_{k+1}\hat{\mathbf{x}}_{k+1|k}]. \tag{97}$$

We show that there exists an equivalent version of Equation 97 that can be used in the case of a nonlinear system model. We do this by linearising the (non-linear) state and observation matrices about the estimated trajectory. This leads to the so-called **Extended Kalman Filter (EKF)** which is the best linear estimator with respect to the minimum-mean-square error.

Finally we take a brief look at validating measurements where due to non-Gaussian sensor noise some measurements could be confused with background clutter or outliers.

## 4.1 The Information Filter

The information filter (or inverse covariance filter) is an alternative form of the Kalman filter algorithm which is mathematically equivalent to the conventional Kalman filter but used in preference to it when either,

1. the measurement dimension is large compared to that of the process noise; or,
2. the initial system state is unknown.

As with the conventional algorithm, the information filter is a recursive linear estimator that repeatedly estimates the state of a variable called the **information vector** and its error covariance matrix called the **information matrix**. This idea should be familiar from the first part of the course where we discussed recursive least squares.

The **information vector** at time $k+1$ given a set of observations up to this time is defined as

$$\hat{\boldsymbol{\theta}}_{k+1|k+1} = \mathbf{P}^{-1}_{k+1|k+1}\hat{\mathbf{x}}_{k+1|k+1} \tag{98}$$

It is then straightforward to prove that its covariance is the inverse of the error covariance matrix, $\mathbf{P}^{-1}_{k+1|k+1}$, or **information matrix**.

**Update equations:** Recall that the conventional Kalman filter update equation for the error covariance matrix is given by

$$\begin{aligned} \mathbf{P}_{k+1|k+1} &= \mathbf{P}_{k+1|k} - \mathbf{K}_{k+1}\mathbf{H}_{k+1}\mathbf{P}_{k+1|k} & (99)\\ \mathbf{K}_{k+1} &= \mathbf{P}_{k+1|k}\mathbf{H}^T_{k+1}(\mathbf{H}_{k+1}\mathbf{P}_{k+1|k}\mathbf{H}^T_{k+1} + \mathbf{R}_{k+1})^{-1} & (100) \end{aligned}$$

The inverse of the (posterior) covariance matrix is given by

$$\mathbf{P}^{-1}_{k+1|k+1} = \mathbf{P}^{-1}_{k+1|k} + \mathbf{H}^T_{k+1}\mathbf{R}^{-1}_{k+1}\mathbf{H}_{k+1} \tag{101}$$

**Proof:**

$$\begin{aligned} & [\mathbf{P} - \mathbf{P}\mathbf{H}^T(\mathbf{H}^T\mathbf{P}\mathbf{H}^T + \mathbf{R})^{-1}\mathbf{H}\mathbf{P}][\mathbf{P}^{-1} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}] \\ = \; & \mathbf{I} + \mathbf{P}\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H} - \mathbf{P}\mathbf{H}^T(\mathbf{H}^T\mathbf{P}\mathbf{H}^T + \mathbf{R})^{-1}\mathbf{H} - \mathbf{P}\mathbf{H}^T(\mathbf{H}^T\mathbf{P}\mathbf{H}^T + \mathbf{R})^{-1}\mathbf{H}\mathbf{P}\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H} \\ = \; & \mathbf{I} + \mathbf{P}\mathbf{H}^T[\mathbf{R}^{-1} - (\mathbf{H}^T\mathbf{P}\mathbf{H}^T + \mathbf{R})^{-1} - (\mathbf{H}^T\mathbf{P}\mathbf{H}^T + \mathbf{R})^{-1}\mathbf{H}\mathbf{P}\mathbf{H}^T\mathbf{R}^{-1}]\mathbf{H} \\ = \; & \mathbf{I} + \mathbf{P}\mathbf{H}^T[\mathbf{R}^{-1} - (\mathbf{H}^T\mathbf{P}\mathbf{H}^T + \mathbf{R})^{-1}(\mathbf{I} + \mathbf{H}^T\mathbf{P}\mathbf{H}^T\mathbf{R}^{-1})]\mathbf{H} \\ = \; & \mathbf{I} + \mathbf{P}\mathbf{H}^T[\mathbf{R}^{-1} - (\mathbf{H}^T\mathbf{P}\mathbf{H}^T + \mathbf{R})^{-1}(\mathbf{R} + \mathbf{H}^T\mathbf{P}\mathbf{H}^T)\mathbf{R}^{-1}]\mathbf{H} \\ = \; & \mathbf{I} + \mathbf{P}\mathbf{H}^T[\mathbf{I} - \mathbf{I}]\mathbf{R}^{-1}\mathbf{H} \\ = \; & \mathbf{I} \end{aligned}$$

The gain can be written

$$\mathbf{K}_{k+1} = \mathbf{P}_{k+1|k+1}\mathbf{H}^T_{k+1}\mathbf{R}^{-1}_{k+1} \tag{102}$$

**Proof:**

$$\begin{aligned} \mathbf{K} &= \mathbf{P}_{k+1|k}\mathbf{H}^T[\mathbf{H}\mathbf{P}_{k+1|k}\mathbf{H}^T + \mathbf{R}]^{-1} \\ &= \mathbf{P}_{k+1|k+1}\mathbf{P}^{-1}_{k+1|k+1}\mathbf{P}_{k+1|k}\mathbf{H}^T\mathbf{R}^{-1}\mathbf{R}[\mathbf{H}\mathbf{P}_{k+1|k}\mathbf{H}^T + \mathbf{R}]^{-1} \\ &= \mathbf{P}_{k+1|k+1}\mathbf{P}^{-1}_{k+1|k+1}\mathbf{P}_{k+1|k}\mathbf{H}^T\mathbf{R}^{-1}[\mathbf{H}\mathbf{P}_{k+1|k}\mathbf{H}^T\mathbf{R}^{-1} + \mathbf{I}]^{-1} \\ &= \mathbf{P}_{k+1|k+1}[\mathbf{P}^{-1}_{k+1|k} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}]\mathbf{P}_{k+1|k}\mathbf{H}^T\mathbf{R}^{-1}[\mathbf{H}\mathbf{P}_{k+1|k}\mathbf{H}^T\mathbf{R}^{-1} + \mathbf{I}]^{-1} \\ &= \mathbf{P}_{k+1|k+1}[\mathbf{I} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\mathbf{P}_{k+1|k}]\mathbf{H}^T\mathbf{R}^{-1}[\mathbf{H}\mathbf{P}_{k+1|k}\mathbf{H}^T\mathbf{R}^{-1} + \mathbf{I}]^{-1} \\ &= \mathbf{P}_{k+1|k+1}\mathbf{H}^T\mathbf{R}^{-1}[\mathbf{I} + \mathbf{H}\mathbf{P}_{k+1|k}\mathbf{H}^T\mathbf{R}^{-1}][\mathbf{H}\mathbf{P}_{k+1|k}\mathbf{H}^T\mathbf{R}^{-1} + \mathbf{I}]^{-1} \\ &= \mathbf{P}_{k+1|k+1}\mathbf{H}^T\mathbf{R}^{-1} \end{aligned}$$

Next consider how to update the information vector:

$$\begin{aligned}\hat{\boldsymbol{\theta}}_{k+1|k+1} &= \mathbf{P}^{-1}_{k+1|k+1}\hat{\mathbf{x}}_{k+1|k+1} \\ &= \mathbf{P}^{-1}_{k+1|k+1}(\mathbf{I}-\mathbf{KH})\hat{\mathbf{x}}_{k+1|k} + \mathbf{P}^{-1}_{k+1|k+1}\mathbf{Kz} \\ &= (\mathbf{P}^{-1}_{k+1|k+1} - \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})\hat{\mathbf{x}}_{k+1|k} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{z} \\ &= \mathbf{P}^{-1}_{k+1|k}\hat{\mathbf{x}}_{k+1|k} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{z} \end{aligned} \tag{103}$$

Note that this is exactly the form we derived from Bayes' Rule in the first part of the course: it is an **information weighted sum of prediction and measurement**

The update can be written succinctly as

$$\hat{\boldsymbol{\theta}}_{k+1|k+1} = \hat{\boldsymbol{\theta}}_{k+1|k} + \mathbf{H}^T_{k+1}\mathbf{R}^{-1}_{k+1}\mathbf{z}_{k+1}$$

**Prediction equations:** Recall that the state prediction is defined by

$$\begin{aligned}\hat{\mathbf{x}}_{k+1|k} &= \mathbf{F}\hat{\mathbf{x}}_{k|k} + \mathbf{Gu} \\ \mathbf{P}_{k+1|k} &= \mathbf{F}\mathbf{P}_{k|k}\mathbf{F}^T + \mathbf{Q}\end{aligned}$$

It follows that the corresponding prediction equations for the information filter are

$$\hat{\boldsymbol{\theta}}_{k+1|k} = \mathbf{P}^{-1}_{k+1|k}\mathbf{F}\mathbf{P}_{k|k}\hat{\boldsymbol{\theta}}_{k|k} + \mathbf{P}^{-1}_{k+1|k}\mathbf{Gu} \tag{104}$$

$$\mathbf{P}^{-1}_{k+1|k} = \left[\mathbf{F}\mathbf{P}_{k|k}\mathbf{F}^T + \mathbf{Q}\right]^{-1} \tag{105}$$

### 4.1.1 Summary of key equations

**Prediction:**

$$\begin{aligned}\hat{\boldsymbol{\theta}}_{k+1|k} &= \mathbf{P}^{-1}_{k+1|k}\mathbf{F}\mathbf{P}_{k|k}\hat{\boldsymbol{\theta}}_{k|k} + \mathbf{P}^{-1}_{k+1|k}\mathbf{Gu} \\ \mathbf{P}^{-1}_{k+1|k} &= \left[\mathbf{F}\mathbf{P}_{k|k}\mathbf{F}^T + \mathbf{Q}\right]^{-1}\end{aligned}$$

**Update:**

$$\begin{aligned}\mathbf{P}^{-1}_{k+1|k+1} &= \mathbf{P}^{-1}_{k+1|k} + \mathbf{H}^T_{k+1}\mathbf{R}^{-1}_{k+1}\mathbf{H}_{k+1} \\ \hat{\boldsymbol{\theta}}_{k+1|k+1} &= \boldsymbol{\theta}_{k+1|k} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{z}\end{aligned}$$

**Comments:** As noted at the beginning of this section certain problems are better solved using the information filter rather than the conventional Kalman filter.

In the case that there is no information about the initial state then the magnitude of $\mathbf{P}_{0|0}$ should be set very large in the conventional Kalman filter. This may lead to significant loss in accuracy. The information filter does not have this problem as it uses the inverse of the initial state covariance error matrix.

When the dimension of the measurement vector $m$ is significantly larger than that of the process noise $p$ the information filter is computationally more efficient that the conventional Kalman filter. This is because one of the computationally expense steps in either case is matrix inversion. In the case of the conventional Kalman filter the inversion of the $m \times m$ matrix $(\mathbf{H}_{k+1}\mathbf{P}^T_{k+1|k}\mathbf{H}^T_{k+1} +$

$\mathbf{R}_{k+1}$) is required. In the case of the inverse covariance filter we compute the inverse of the $p \times p$ matrix defined by Equation 105.

Finally, note that given the output from one filter it is easy to find the equivalent output of the other filter using Equation 98 and the imformation matrix (inverse covariance).

## 4.2 Extended Kalman Filter

In this section we consider the extension of Kalman filtering ideas to the case of non-linear system models.

We assume that the system can be represented by a nonlinear discrete-time state-space model of the form

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k, k) + \mathbf{w}_k, \tag{106}$$

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k, k) + \mathbf{v}_k, \tag{107}$$

where $\mathbf{f}(.,.,k)$ is a nonlinear state transition matrix and $\mathbf{h}(.,.,k)$ is the nonlinear observation matrix.

We assume that the process and measurement noise are Gaussian, uncorrelated and zero-mean, and have no cross-correlation. Thus

$$\begin{aligned} E[\mathbf{w}_k] &= 0 \\ E[\mathbf{v}_k] &= 0 \\ E[\mathbf{w}_i \mathbf{w}_j^T] &= \delta_{ij} \mathbf{Q}_i \\ E[\mathbf{v}_i \mathbf{v}_j^T] &= \delta_{ij} \mathbf{R}_i \\ E[\mathbf{w}_i \mathbf{v}_j^T] &= \mathbf{0} \end{aligned}$$

### 4.2.1 Prediction

As in the linear case, we assume that we have at time $k$

$$\hat{\mathbf{x}}_{k|k} = E[\mathbf{x}_k | \mathbf{Z}^k], \qquad \mathbf{P}_{k|k}$$

To generate the prediction we expand Equation 106 in a Taylor's series about the prediction $\hat{\mathbf{x}}_{k|k}$ up to the first-order terms.

$$\mathbf{x}_{k+1} = \mathbf{f}(\hat{\mathbf{x}}_{k|k}, \mathbf{u}_k, k) + \left[\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right] [\mathbf{x}_k - \hat{\mathbf{x}}_{k|k}] + O([\mathbf{x}_k - \hat{\mathbf{x}}_{k|k}]^2) + \mathbf{w}_k \tag{108}$$

where the Jacobian of $\mathbf{f}$ is evaluated at $\hat{\mathbf{x}}_{k|k}$. Taking the Expectation of Equation 108, ignoring higher than first order terms, and assuming that $\hat{\mathbf{x}}_{k|k}$ is approximately equal to the conditional mean and that the process noise has zero mean, yields

$$\begin{aligned} \hat{\mathbf{x}}_{k+1|k} &= E[\mathbf{x}_{k+1} | \mathbf{Z}^k] \\ &= \mathbf{f}(\hat{\mathbf{x}}_{k|k}, \mathbf{u}_k, k) \end{aligned} \tag{109}$$

The state covariance can be found as follows. First the prediction error is given by

$$\begin{aligned}
\tilde{\mathbf{x}}_{k+1|k} &= \mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1|k} \\
&= \mathbf{f}(\hat{\mathbf{x}}_{k|k}, \mathbf{u}_k, k) + \left[\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right][\mathbf{x}_k - \hat{\mathbf{x}}_{k|k}] \\
&\quad + O([\mathbf{x}_k - \hat{\mathbf{x}}_{k|k}]^2) + \mathbf{w}_k - \mathbf{f}(\hat{\mathbf{x}}_{k|k}, \mathbf{u}_k, k) \\
&\approx \left[\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right][\mathbf{x}_k - \hat{\mathbf{x}}_{k|k}] + \mathbf{w}_k \\
&= \left[\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right][\tilde{\mathbf{x}}_{k|k}] + \mathbf{w}_k \qquad (110)
\end{aligned}$$

The prediction covariance is then found by taking the Expectation of the product of the prediction error with it's transpose:

$$\begin{aligned}
\mathbf{P}_{k+1|k} &= E[\tilde{\mathbf{x}}_{k+1|k}\tilde{\mathbf{x}}_{k+1|k}^T|\mathbf{Z}^k] \\
&\approx E[(\left[\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right][\tilde{\mathbf{x}}_{k|k}] + \mathbf{w}_k)(\left[\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right][\tilde{\mathbf{x}}_{k|k}] + \mathbf{w}_k)^T|\mathbf{Z}^k] \\
&= \left[\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right] E[\tilde{\mathbf{x}}_{k|k}\tilde{\mathbf{x}}_{k|k}^T|\mathbf{Z}^k] \left[\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right]^T + E[\mathbf{w}_k\mathbf{w}_k^T] \\
&= \left[\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right] \mathbf{P}_{k|k} \left[\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right]^T + \mathbf{Q}_k \qquad (111)
\end{aligned}$$

Note that the prediction covariance has the same form as its linear equivalent with the Jacobian $\left[\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right]$ playing the role of transition matrix $\mathbf{F}_k$.

#### 4.2.2 Observation Prediction and Innovation

The observation $\mathbf{z}$ can be written as a Taylor series expanded about the prediction $\hat{\mathbf{x}}_{k+1|k}$:

$$\mathbf{z}_{k+1} = \mathbf{h}(\hat{\mathbf{x}}_{k+1|k}) + \left[\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right][\hat{\mathbf{x}}_{k+1|k} - \mathbf{x}_{k+1}] + O([\hat{\mathbf{x}}_{k+1|k} - \mathbf{x}_{k+1}]^2) + \mathbf{w}_{k+1}$$

Truncating to first order and taking expectations yields the predicted observation

$$\hat{\mathbf{z}}_{k+1|k} \approx \mathbf{h}(\hat{\mathbf{x}}_{k+1|k}) \qquad (112)$$

The innovation is then found as

$$\boldsymbol{\nu}_{k+1} = \mathbf{z}_{k+1} - \mathbf{h}(\hat{\mathbf{x}}_{k+1|k}) \qquad (113)$$

and the innovation covariance is found as follows:

$$\begin{aligned}
\mathbf{S}_{k+1} &= E[\boldsymbol{\nu}_{k+1}\boldsymbol{\nu}_{k+1}^T] \\
&= E[(\mathbf{z}_{k+1} - \mathbf{h}(\hat{\mathbf{x}}_{k+1|k})(\mathbf{z}_{k+1} - \mathbf{h}(\hat{\mathbf{x}}_{k+1|k})^T] \\
&\approx E\left[\left(\left[\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right](\mathbf{x}_{k+1|k} - \mathbf{x}_k) + \mathbf{w}_{k+1}\right)\left((\mathbf{x}_{k+1|k} - \mathbf{x}_k)^T\left[\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right]^T + \mathbf{w}_{k+1}^T\right)\right] \\
&= \left[\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right] \mathbf{P}_{k+1|k} \left[\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right]^T + \mathbf{R}_{k+1} \qquad (114)
\end{aligned}$$

where the Jacobian of $\mathbf{h}$ is evaluated at $\hat{\mathbf{x}}_{k+1|k}$. Again note that Equation 114 is in the same form as its linear counterpart except that $\left[\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right]$ has replaced $\mathbf{H}_{k+1}$.

#### 4.2.3 Update

By similar reasoning to that made for the linear case it is possible to derive from first principles the equations for the filter gain, state update and covariance update. The forms turn out to be the same as for the linear Kalman filter with $\mathbf{H}_{k+1}$ replaced by $\left[\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right]$. Thus, the Kalman gain is given by

$$\mathbf{K}_{k+1} = \mathbf{P}_{k+1|k} \left[\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right]^T \mathbf{S}_{k+1}^{-1} \tag{115}$$

The state update is given by

$$\hat{\mathbf{x}}_{k+1|k+1} = \hat{\mathbf{x}}_{k+1|k} + \mathbf{K}_{k+1}[\mathbf{z}_{k+1} - \mathbf{h}(\hat{\mathbf{x}}_{k+1|k})] \tag{116}$$

and the covariance update is given by

$$\mathbf{P}_{k+1|k+1} = \mathbf{P}_{k+1|k} - \mathbf{K}_{k+1}\mathbf{S}_{k+1}\mathbf{K}_{k+1}^T \tag{117}$$

#### 4.2.4 Summary of key equations

**Prediction:**

$$\begin{aligned} \hat{\mathbf{x}}_{k+1|k} &= \mathbf{f}(\hat{\mathbf{x}}_{k|k}, \mathbf{u}_k, k) && (118) \\ \mathbf{P}_{k+1|k} &= \left[\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right] \mathbf{P}_{k|k} \left[\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right]^T + \mathbf{Q}_k && (119) \end{aligned}$$

**Update:**

$$\begin{aligned} \hat{\mathbf{x}}_{k+1|k+1} &= \hat{\mathbf{x}}_{k+1|k} + \mathbf{K}_{k+1}[\mathbf{z}_{k+1} - \mathbf{h}(\hat{\mathbf{x}}_{k+1|k})] && (120) \\ \mathbf{P}_{k+1|k+1} &= \mathbf{P}_{k+1|k} - \mathbf{K}_{k+1}\mathbf{S}_{k+1}\mathbf{K}_{k+1}^T && (121) \end{aligned}$$

where

$$\mathbf{K}_{k+1} = \mathbf{P}_{k+1|k} \left[\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right]^T \mathbf{S}_{k+1}^{-1} \tag{122}$$

and

$$\mathbf{S}_{k+1} = \left[\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right] \mathbf{P}_{k+1|k} \left[\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right]^T + \mathbf{R}_{k+1} \tag{123}$$

### 4.3 Some general comments

1. The Jacobians $\left[\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right]$ and $\left[\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right]$ are functions of both the state and timestep; they are not constant.

2. **Stability:** Since we are dealing with perturbation models of the state and observation matrices about the predicted trajectory, it is important that predictions are close enough to the true states otherwise the filter will be poorly matched and possibly diverge.

3. **Initialisation:** Unlike in the linear case, special care has to be taken when initialising the Extended Kalman filter.

4. **Computational cost:** The Extended Kalman filter is computationally significantly more expensive than it's linear counterpart. This limited its early use in applications. However, today real-time computing implementations of the EKF can be achieved using moderate computing resources.

## 4.4 Implementation

Implementation issues are similar to those of the linear Kalman filter and you can test the performance of the filter using all the techniques introduced in Section 2.

In particular, special care has to be taken to check whether the system and noise process modelling assumptions are met. There are obviously some errors introduced by using a linearised model.

A further important point to note is that the state covariance matrix is only an approximation to the mean square error and not a true covariance. Recall that $\mathbf{P}_{k+1|k+1}$ determines the weight given to new measurements in the updating procedure. Thus, if $\mathbf{P}_{k+1|k+1}$ is erroneous and becomes small, the measurements have little affect on the estimation and it is quite possible that the EKF will diverge.

## 4.5 Example - a simple robot vehicle

Consider the simplified description of a robot vehicle in motion illustrated in Figure 20. The state is described by $\mathbf{x}_k = [x_k, y_k, \phi_k]^T$. Control is provided by an input control vector which determines the velocity of travel, $\mathbf{u}_k = [V_k, \varphi_k]^T$.

The motion of the vehicle can be described by the nonlinear state transition equation

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \phi_{k+1} \end{bmatrix} = \begin{bmatrix} x_k + \Delta t V_k \cos(\phi_k + \varphi_k) \\ y_k + \Delta t V_k \sin(\phi_k + \varphi_k) \\ \phi_k + \frac{V_k}{B} \Delta t \sin(\varphi_k) \end{bmatrix} + \mathbf{q}_k \tag{124}$$

Here, $\Delta t$ is the time interval between time steps, $B$ is the wheel baseline and $\mathbf{q}_k$ is the noise vector which combines errors in modelling the process and control.

We assume that measurements of range (depth) and bearing are made to a set of beacons at fixed locations $\mathbf{B}_i = [X_i, Y_i]^T, i = 1, \ldots N$. The nonlinear observation model is therefore

$$\mathbf{z}_k = \begin{bmatrix} r_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} \sqrt{(X - x_k)^2 + (Y - y_k)^2} \\ \tan^{-1}\left(\frac{Y - y_k}{X - x_k}\right) - \phi_k \end{bmatrix} + \mathbf{r}_k \tag{125}$$

where $\mathbf{r}_k$ is the observation noise.

**Prediction:**

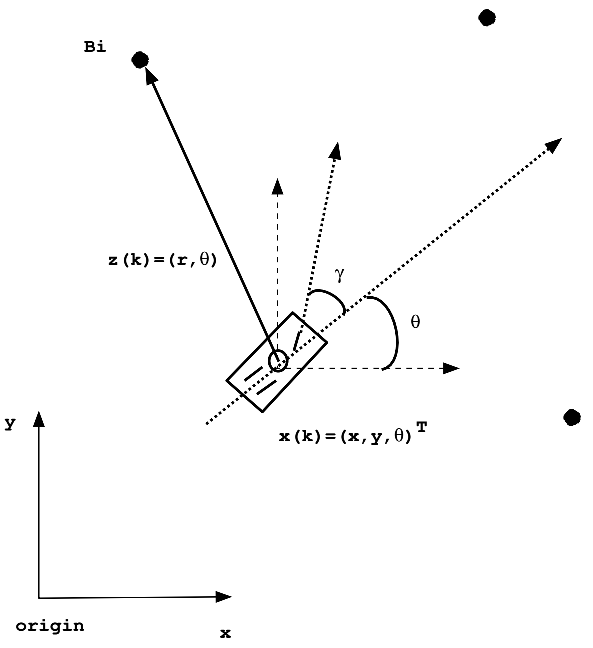



Figure 20: *Example: geometry of a simple robot vehicle.*

From equation 109 the predicted state $\hat{\mathbf{x}}_{k+1|k}$ is given by

$$\begin{bmatrix} \hat{x}_{k+1|k} \\ \hat{y}_{k+1|k} \\ \hat{\phi}_{k+1|k} \end{bmatrix} = \begin{bmatrix} \hat{x}_{k|k} + \Delta t V_k \cos(\hat{\phi}_{k|k} + \varphi_k) \\ \hat{y}_{k|k} + \Delta t V_k \sin(\hat{\phi}_{k|k} + \varphi_k) \\ \hat{\phi}_{k|k} + \frac{V_k}{B} \Delta t \sin(\varphi_k) \end{bmatrix} + \mathbf{q}_k \tag{126}$$

The prediction covariance matrix is

$$\mathbf{P}_{k+1|k} = \left[\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right] \mathbf{P}_{k|k} \left[\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right]^T + \mathbf{Q}_k \tag{127}$$

where

$$\left[\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right] = \begin{bmatrix} 1 & 0 & -\Delta t V_k \sin(\hat{\phi}_{k|k} + \varphi_k) \\ 0 & 1 & +\Delta t V_k \cos(\hat{\phi}_{k|k} + \varphi_k) \\ 0 & 0 & 1 \end{bmatrix} \tag{128}$$

**Update:**

The equations for the updating of the state and its covariance are:

$$\begin{aligned} \hat{\mathbf{x}}_{k+1|k+1} &= \hat{\mathbf{x}}_{k+1|k} + \mathbf{K}_{k+1}[\mathbf{z}_k - \mathbf{h}(\hat{\mathbf{x}}_{k+1|k})] \\ \mathbf{P}_{k+1|k+1} &= \mathbf{P}_{k+1|k} - \mathbf{K}_{k+1}\mathbf{S}_{k+1}\mathbf{K}_{k+1}^T \end{aligned} \tag{129}$$

where

$$\mathbf{K}_{k+1} = \mathbf{P}_{k+1|k} \left[\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right]^T \mathbf{S}_{k+1}^{-1} \tag{130}$$

and

$$\mathbf{S}_{k+1} = \left[\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right] \mathbf{P}_{k+1|k} \left[\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right]^T + \mathbf{R}_{k+1} \tag{131}$$

and

$$\left[\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right] = \begin{bmatrix} \frac{\hat{x}_{k+1|k}-X}{d} & \frac{\hat{y}_{k+1|k}-Y}{d} & 0 \\ -\frac{\hat{y}_{k+1|k}-Y}{d^2} & \frac{\hat{x}_{k+1|k}-X}{d^2} & -1 \end{bmatrix} \tag{132}$$

Here $d = \sqrt{(X - \hat{x}_{k+1|k})^2 + (Y - \hat{y}_{k+1|k})^2}$.

## 4.6 Measurement Validation - coping with Non-Gaussian Noise

Recall that in the first part of the course we considered the problem of **validating** new measurements. Such considerations are particularly important in applications such as sonar tracking where measurement 'outliers' are common; i.e. the sensor distribution (conditioned on the true value) is no longer Gaussian.

The solution to this problem we developed was to set up a **validation gate** or **innovation gate**; any measurement lying in the "region" of space defined by the gate is considered to be associated with the target. Note that since the measurement vector typically has a number of components (say N) the "region" will be a region in N-dimensional space (typically an ellipsoid). We use the same approach with the Kalman Filter and recap it here.

Assume that we already have a predicted measurement $\hat{\mathbf{z}}_{k+1|k}$, hence the **innovation**, $\boldsymbol{\nu}_{k+1}$ and its associated covariance $\mathbf{S}_{k+1}$. Under the assumption that the innovation is normally distributed, the normalised innovation $\boldsymbol{\nu}_{k+1}\mathbf{S}_{k+1}^{-1}\boldsymbol{\nu}_{k+1}^T$ is distributed as a $\chi^2$ distribution on $n$ degrees of freedom, where $n$ is the dimension of the vector $\boldsymbol{\nu}$.

Hence, as we saw in the first part of the course, we can define a confidence region (which we call here the validation gate) such that

$$\begin{aligned} R(\gamma) &= \{\mathbf{z}_{|}(\mathbf{z}_{k+1} - \hat{\mathbf{z}}_{k+1|k})^T \mathbf{S}_{k+1}^{-1} (\mathbf{z}_{k+1} - \hat{\mathbf{z}}_{k+1|k}) \leq \gamma\} \\ &= \{\mathbf{z}_{|}\boldsymbol{\nu}_{k+1}^T \mathbf{S}_{k+1}^{-1} \boldsymbol{\nu}_{k+1} \leq \gamma\} \end{aligned} \tag{133}$$

where $\gamma$ can be obtained from standard $\chi^2$ distribution tables.

In the context of the Kalman Filter algorithm, Equation 133 constrains the region of space where we look for a measurement. We assume that the correct measurement will be detected in this region. It is possible, however, that more than one measurement will be in the valid region. The problem of distinguishing between the correct measurement and measurements arising from background clutter and other targets (false alarms) is called **data association**. This topic falls outside of the scope of the course (see, for example, Bar-Shalom and Fortmann).