ECEN426
Advanced Mechatronic Systems
Kalman Filters

Christopher Hollitt

September 7, 2021

Introduction

During the course we are going to capture all of the information we are trying to track within a *state vector* $\boldsymbol{x}$. For example, the position and velocity of our robot will be elements withing that vector, as will the positions of various beacons.

Our job when performing slam will be to find an estimate of $\boldsymbol{x}$, which we will denote as $\hat{\boldsymbol{x}}$.

In this section we will look at how we can express our uncertainty about our estimate, which we will denote $\boldsymbol{P}$.

Finally, we will assume that the state vector evolves according to a set of linear equations $\boldsymbol{x}(t+1) = \boldsymbol{A}\boldsymbol{x}(t)$.

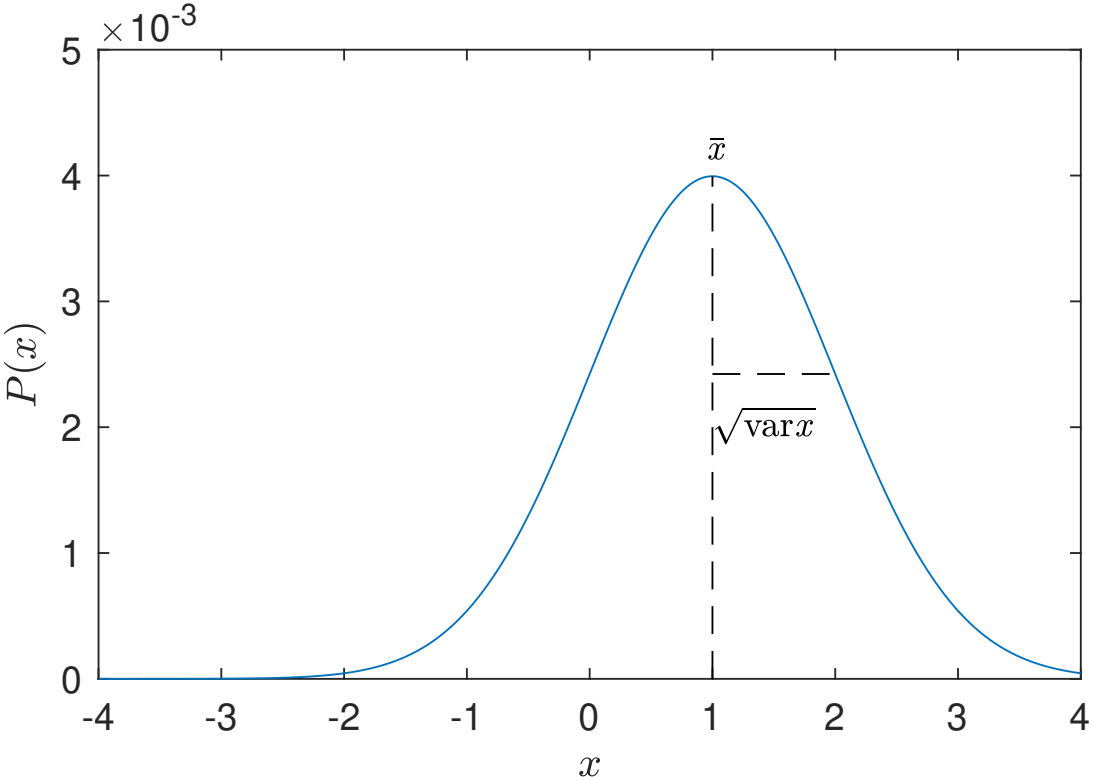
Some of this will be familiar if you are taking ECEN415, but in any case we will discuss all of this in more detail later.

Part I

Uncertainty

Uncertainty of Scalar Quantities

When discussing a scalar random variable x, we often assume that x is normally distributed and characterise x by its mean and its variance.



These are determined via expectation values. That is, we ask what we would expect to find on average if we were to take many measurements of x.

$$\bar{x} := \mathrm{E}\{x\}$$

$$\mathrm{var}\, x := \mathrm{E}\{(x - \bar{x})^2\}$$

Covariance of two scalar quantities

We similarly define the covariance of two signals x_i and x_j to be

$$\operatorname{cov}(x_i, x_j) := \mathrm{E}\big\{(x_i - \bar{x}_i)(x_j - \bar{x}_j)\big\}\,.$$

If two variables that are perfectly uncorrelated then they have a covariance of zero. Conversely, if two variables tend to vary together then we will obtain a non-zero covariance. Notice that if we consider the covariance of a signal x_i with itself we recover the familiar variance.

$$\begin{aligned}\operatorname{cov}(x_i, x_i) &= \mathrm{E}\{(x_i - \bar{x}_i)(x_i - \bar{x}_i)\} \\ &= \mathrm{E}\big\{(x_i - \bar{x}_i)^2\big\} \\ \operatorname{cov}(x_i, x_i) &= \operatorname{var}(x_i)\end{aligned}$$

Note also that covariance is commutative so that $\operatorname{cov}(x_i, x_j) = \operatorname{cov}(x_j, x_i)$.

Vector covariance

We wish to extend our treatment to allow for discussion of vector quantities, such as state vectors. This is so that we can think about the covariance between the state variables x_i and x_j.
To do this we define the *covariance matrix* for $\boldsymbol{x} \in \mathbb{R}^n$, which has the covariance of its i-th and j-th components as its (i, j)-th element.

$$\boldsymbol{\Sigma}_{\boldsymbol{x}} = \begin{bmatrix} \operatorname{cov}(x_1, x_1) & \operatorname{cov}(x_1, x_2) & \cdots & \operatorname{cov}(x_1, x_n) \\ \operatorname{cov}(x_2, x_1) & \operatorname{cov}(x_2, x_2) & \cdots & \operatorname{cov}(x_2, x_n) \\ \vdots & \vdots & \ddots & \vdots \\ \operatorname{cov}(x_n, x_1) & \operatorname{cov}(x_n, x_2) & \cdots & \operatorname{cov}(x_n, x_n) \end{bmatrix}$$

$$= \begin{bmatrix} \operatorname{var}(x_1) & \operatorname{cov}(x_1, x_2) & \cdots & \operatorname{cov}(x_1, x_n) \\ \operatorname{cov}(x_2, x_1) & \operatorname{var}(x_2) & \cdots & \operatorname{cov}(x_2, x_n) \\ \vdots & \vdots & \ddots & \vdots \\ \operatorname{cov}(x_n, x_1) & \operatorname{cov}(x_n, x_2) & \cdots & \operatorname{var}(x_n) \end{bmatrix}$$

Note that if $\boldsymbol{x} \in \mathbb{R}^n$, then $\boldsymbol{\Sigma}_{\boldsymbol{x}} \in \mathbb{R}^{n \times n}$.

Vector covariance

Notice that the covariance matrix is often denoted $\boldsymbol{\Sigma}$ in allusion to the σ^2 used to describe the variance of a scalar signal.

Because the covariance is commutative ($\mathrm{cov}(x_i, x_j) = \mathrm{cov}(x_j, x_i)$) the covariance matrix is always symmetric. This symmetry, along with the fact that every real quantity must have a non-negative variance implies that all covariance matrices must be positive semi-definite ($\boldsymbol{z}^{\mathsf{T}} \boldsymbol{\Sigma} \boldsymbol{z} \geq 0 \quad \forall \boldsymbol{z}$).

In terms of the underlying vector, the covariance matrix can be calculated as

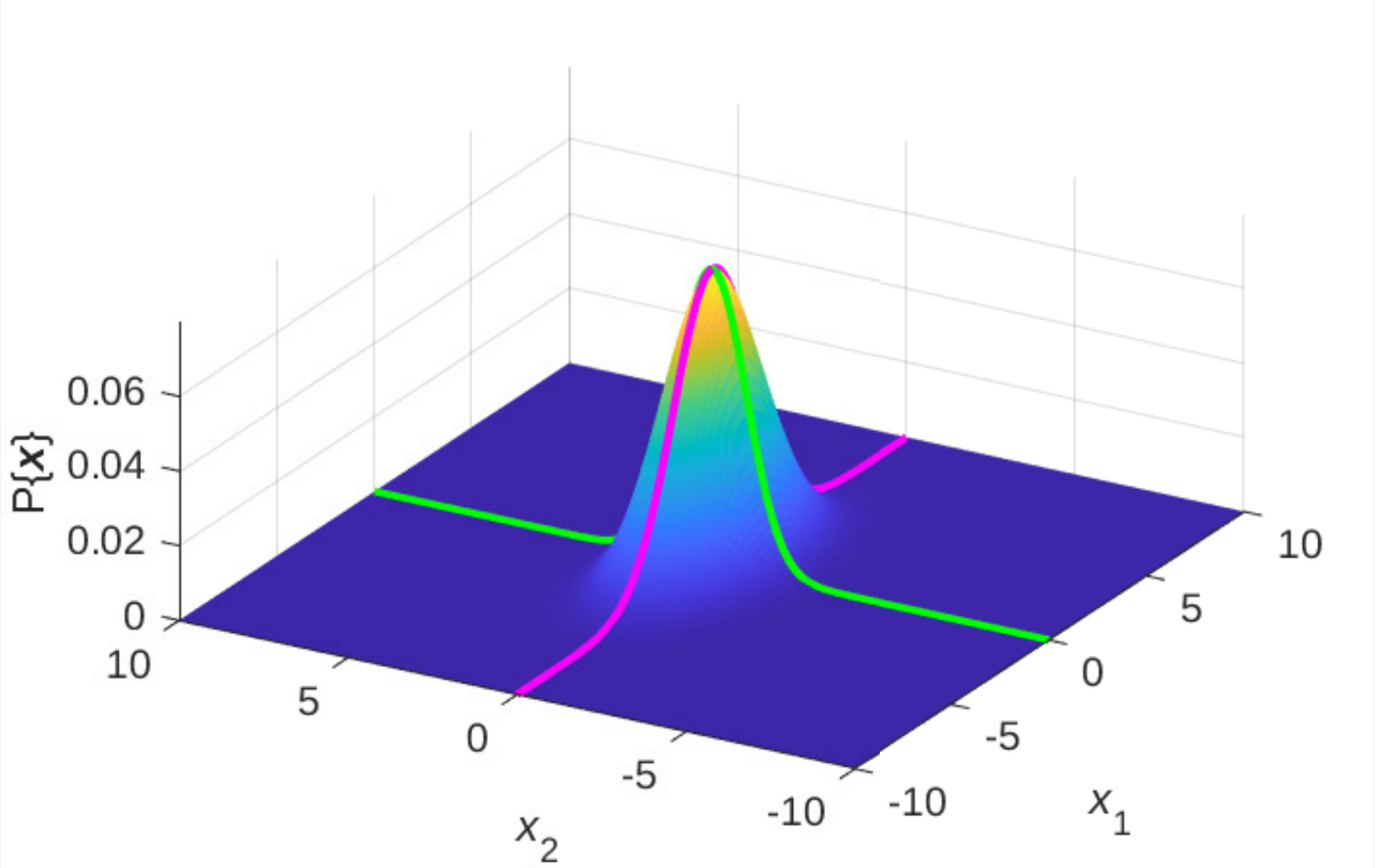
$$\boldsymbol{\Sigma}_{\boldsymbol{x}} = \mathrm{E}\left\{ (\boldsymbol{x} - \bar{\boldsymbol{x}})(\boldsymbol{x} - \bar{\boldsymbol{x}})^{\mathsf{T}} \right\}.$$

In the case of zero mean quantities, for which $\mathrm{E}\{\boldsymbol{x}\} = \bar{\boldsymbol{x}} = 0$ we have the simpler relation

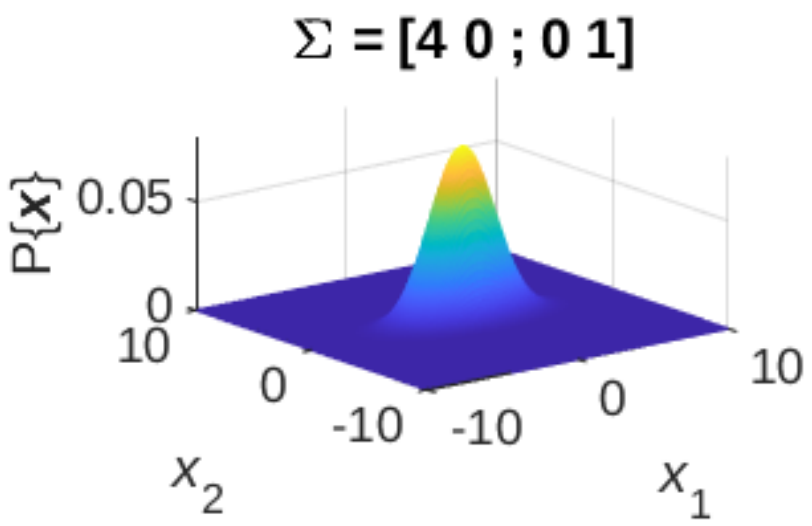
$$\boldsymbol{\Sigma}_{\boldsymbol{x}} \equiv \mathrm{E}\left\{ \boldsymbol{x}\boldsymbol{x}^{\mathsf{T}} \right\}.$$

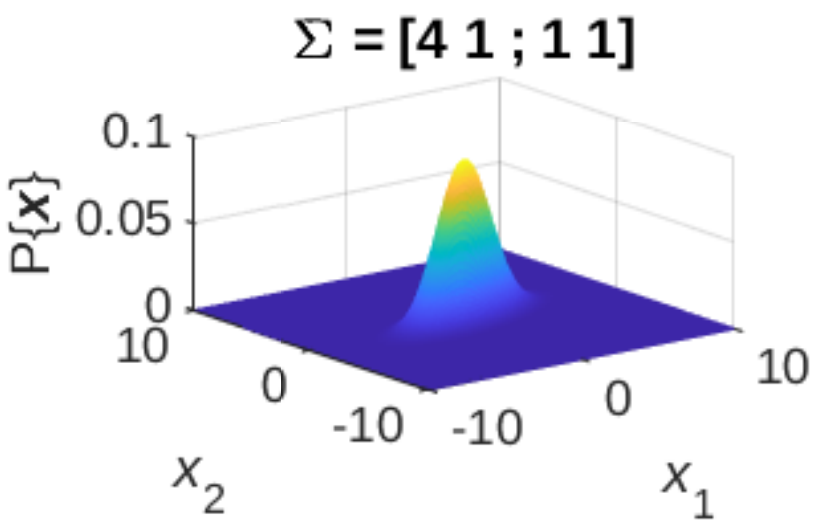
Interpretation of Covariance

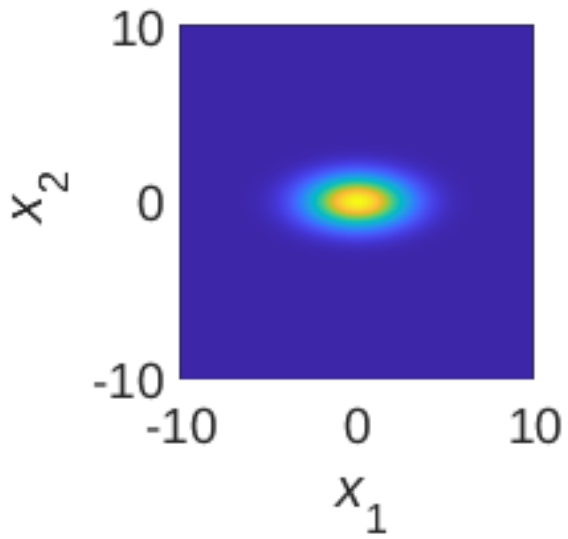
For $\boldsymbol{\Sigma} \in \mathbb{R}^{2\times 2}$ we can plot the bivariate normal distribution. In this case we have used $\boldsymbol{\Sigma} = \begin{bmatrix} 4 & 0 \\ 0 & 1 \end{bmatrix}$.

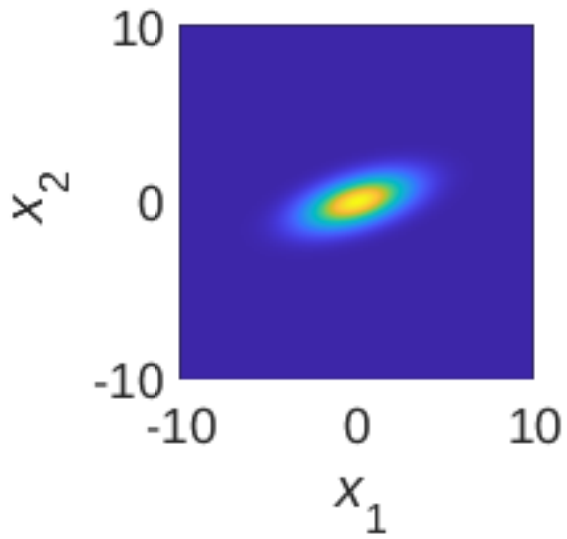


Interpretation of Covariance









Uncertainty Ellipsoids

The shapes of the probability contours around the mean of the probability density function are elliptical and have size and orientation determined by the covariance matrix.

- The ellipse is large in directions having large variance.
- If $\boldsymbol{\Sigma}$ is diagonal then the elements of $\boldsymbol{x}$ are uncorrelated, so the ellipse is aligned with x_i.
- Non-diagonal $\boldsymbol{\Sigma}$ leads to apparent rotation of the uncertainty ellipse. That is, the major and minor axes of the ellipse are not aligned with the coordinate system.

The shape of the ellipsoids can be described by $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{x} = k$ for k is a measure of deviation from the mean. Different k yield different sized ellipses.

- The axes of the ellipse are aligned with the eigenvectors of $\boldsymbol{\Sigma}$.
- The equatorial radii are given by the square roots of the eigenvalues of $\boldsymbol{\Sigma}$.

Uncertainty Ellipsoids

If you wish to find an ellipse within which you expect to find a certain proportion of samples drawn from a population, then we can find k using a chi-squared distribution. That is, in n dimensions, the ellipse containing a proportion α of samples would be

$$\boldsymbol{x}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{x} = \chi_n^2(\alpha)$$

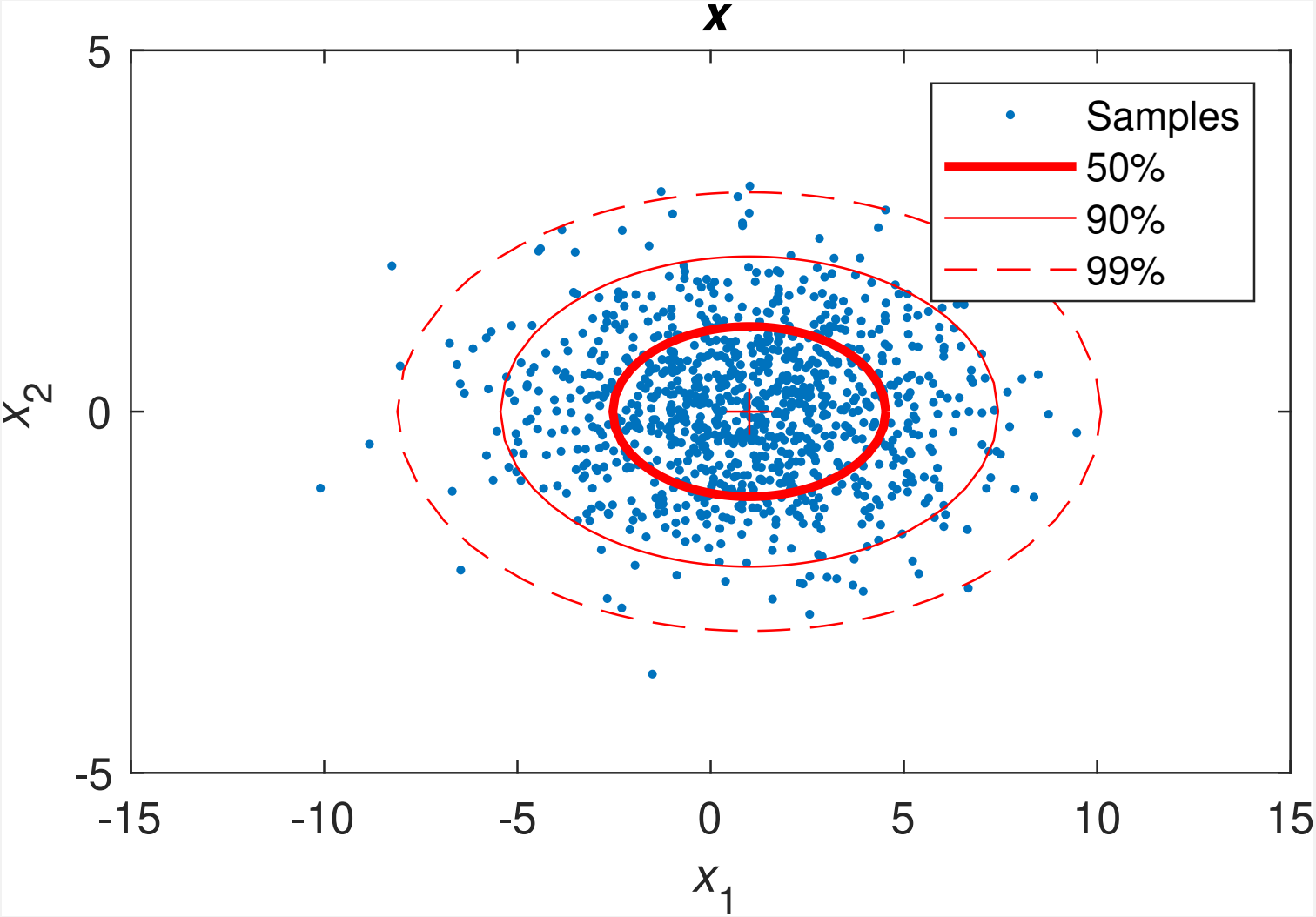
For example, if $n = 2$, then we have

- For 50% confidence ($\alpha = 0.5$), $\chi_2^2 = 1.39$
- For 90% confidence ($\alpha = 0.90$), $\chi_2^2 = 4.60$
- For 99% confidence ($\alpha = 0.95$), $\chi_2^2 = 9.21$

These confidence levels will be used in the illustrative figures to follow.

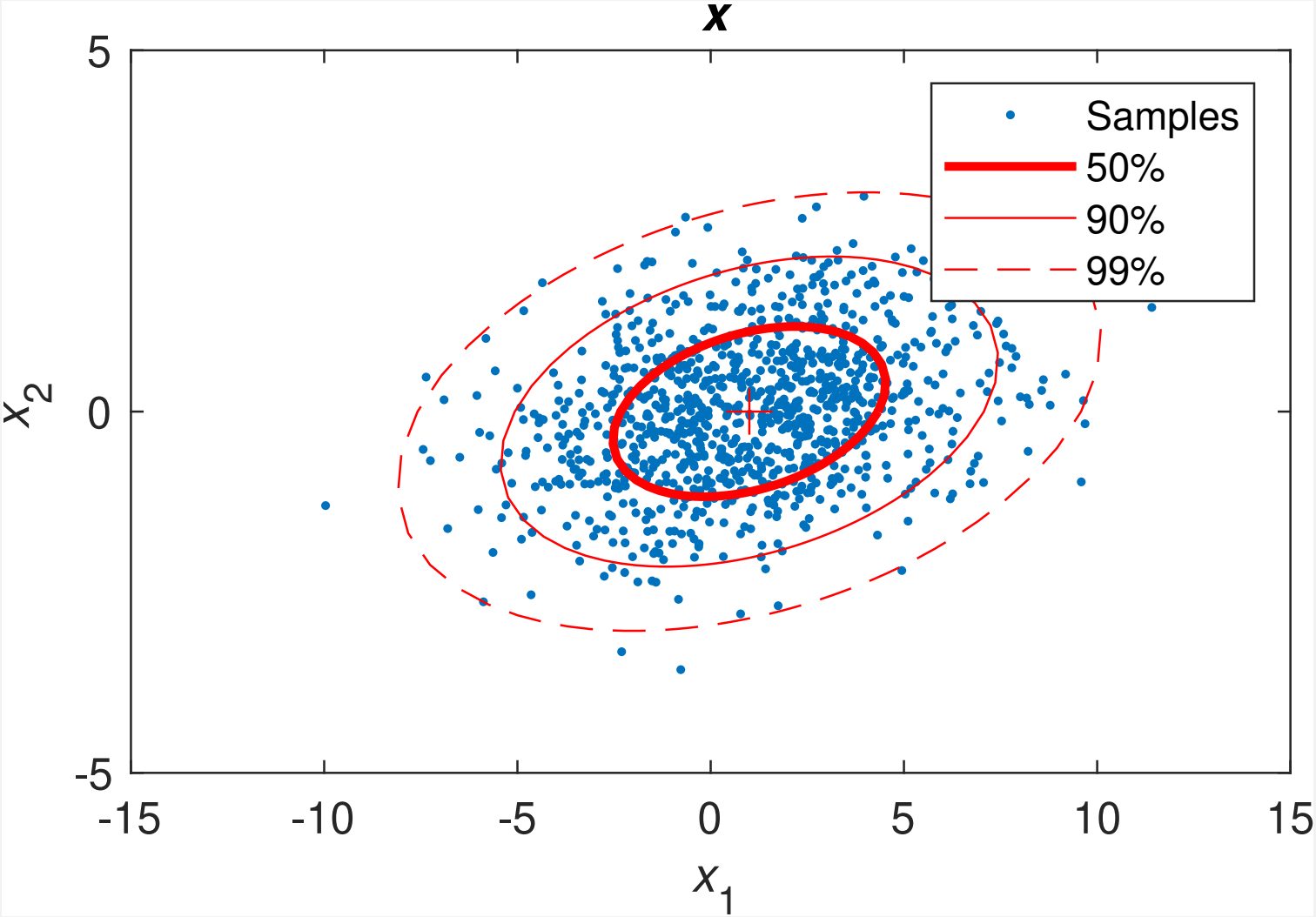
Uncertainty Ellipsoids

1000 samples drawn from a distribution having $\bar{x} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, $\Sigma_x = \begin{bmatrix} 9 & 0 \\ 0 & 1 \end{bmatrix}$.



Uncertainty Ellipsoids

1000 samples from a distribution having $\bar{x} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, $\Sigma_x = \begin{bmatrix} 9 & 1 \\ 1 & 1 \end{bmatrix}$



Quadratic Forms

Notice that we are now using matrices for a new job. When used in a so-called *quadratic form*, $\boldsymbol{x}^{\mathsf{T}}\boldsymbol{M}\boldsymbol{x} = k$, the matrix $\boldsymbol{M}$ is used to describe a shape.

We will only make use of this in describing ellipses (which requires $\boldsymbol{M}$ to have positive eigenvalues). However, the same machinery can describe hyperboloids, which have at least one positive and one negative eigenvalue and paraboloids can arise when there is at least one zero eigenvalue.

Note that it doesn't make sense to treat these 'shape' matrices in the same way as we have used matrices that describe mappings. Matrices with the two different uses will appear together, so try to keep the meanings distinct.

Part II

Propagation of uncertainty

Propagation of an uncertain vector

Imagine that we have a (true) state vector $\boldsymbol{x}$ that we have estimated as having mean $\hat{\boldsymbol{x}}$ and covariance $\boldsymbol{P}$. As an example, we may have found this by taking many measurements and then calculating the empirical mean and covariance.

We would like to know what happens if we subject $\boldsymbol{x}$ to a mapping described by a matrix $\boldsymbol{A}$. How should we modify $\hat{\boldsymbol{x}}$ and $\boldsymbol{P}$ to describe the distribution after the mapping? That is, what will be the mean and covariance of $\boldsymbol{Ax}$?

We know that $\boldsymbol{x}$ maps to $\boldsymbol{Ax}$, so it makes sense to map our estimate of the mean in the same way; $\hat{\boldsymbol{x}} \mapsto \boldsymbol{A}\hat{\boldsymbol{x}}$.

Propagation of covariance

To determine the mapped form of the covariance matrix we need to substitute our new vectors into the covariance equation. Let us denote the covariance after the mapping as $\boldsymbol{P}_{\text{post}}$.

$$
\begin{aligned}
\boldsymbol{P}_{\text{post}} &= \mathrm{E}\left\{(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{A}\hat{\boldsymbol{x}})(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{A}\hat{\boldsymbol{x}})^{\mathsf{T}}\right\} \\
&= \mathrm{E}\left\{(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{A}\hat{\boldsymbol{x}})\left((\boldsymbol{A}\boldsymbol{x})^{\mathsf{T}} - (\boldsymbol{A}\hat{\boldsymbol{x}})^{\mathsf{T}}\right)\right\} && (1) \\
&= \mathrm{E}\left\{(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{A}\hat{\boldsymbol{x}})\left(\boldsymbol{x}^{\mathsf{T}}\boldsymbol{A}^{\mathsf{T}} - \hat{\boldsymbol{x}}^{\mathsf{T}}\boldsymbol{A}^{\mathsf{T}}\right)\right\} && (2) \\
&= \mathrm{E}\left\{\boldsymbol{A}\boldsymbol{x}\boldsymbol{x}^{\mathsf{T}}\boldsymbol{A}^{\mathsf{T}} - \boldsymbol{A}\boldsymbol{x}\hat{\boldsymbol{x}}^{\mathsf{T}}\boldsymbol{A}^{\mathsf{T}} - \boldsymbol{A}\hat{\boldsymbol{x}}\boldsymbol{x}^{\mathsf{T}}\boldsymbol{A}^{\mathsf{T}} + \boldsymbol{A}\hat{\boldsymbol{x}}\hat{\boldsymbol{x}}^{\mathsf{T}}\boldsymbol{A}^{\mathsf{T}}\right\} \\
&= \boldsymbol{A}\,\mathrm{E}\left\{\boldsymbol{x}\boldsymbol{x}^{\mathsf{T}} - \boldsymbol{x}\hat{\boldsymbol{x}}^{\mathsf{T}} - \hat{\boldsymbol{x}}\boldsymbol{x}^{\mathsf{T}} + \hat{\boldsymbol{x}}\hat{\boldsymbol{x}}^{\mathsf{T}}\right\}\boldsymbol{A}^{\mathsf{T}} \\
&= \boldsymbol{A}\,\mathrm{E}\left\{\boldsymbol{x}\left(\boldsymbol{x}^{\mathsf{T}} - \hat{\boldsymbol{x}}^{\mathsf{T}}\right) - \hat{\boldsymbol{x}}\left(\boldsymbol{x}^{\mathsf{T}} - \hat{\boldsymbol{x}}^{\mathsf{T}}\right)\right\}\boldsymbol{A}^{\mathsf{T}} \\
&= \boldsymbol{A}\,\mathrm{E}\left\{(\boldsymbol{x} - \hat{\boldsymbol{x}})\left(\boldsymbol{x}^{\mathsf{T}} - \hat{\boldsymbol{x}}^{\mathsf{T}}\right)\right\}\boldsymbol{A}^{\mathsf{T}} && (3) \\
&= \boldsymbol{A}\,\mathrm{E}\left\{(\boldsymbol{x} - \hat{\boldsymbol{x}})(\boldsymbol{x} - \hat{\boldsymbol{x}})^{\mathsf{T}}\right\}\boldsymbol{A}^{\mathsf{T}} \implies \boldsymbol{P}_{\text{post}} = \boldsymbol{A}\boldsymbol{P}\boldsymbol{A}^{\mathsf{T}}
\end{aligned}
$$

Propagation of an uncertain vector

In (1) and (3) we have used $(\boldsymbol{Y}+\boldsymbol{Z})^{\mathsf{T}} = \boldsymbol{Y}^{\mathsf{T}} + \boldsymbol{Z}^{\mathsf{T}}$.
In (2) we have used $(\boldsymbol{YZ})^{\mathsf{T}} = \boldsymbol{Z}^{\mathsf{T}}\boldsymbol{Y}^{\mathsf{T}}$.

Thus if we have a vector $\boldsymbol{x} \sim \mathcal{N}(\hat{\boldsymbol{x}}, \boldsymbol{P})$ that we act on with the matrix $\boldsymbol{A}$, then the output of the operation will be $\boldsymbol{Ax} \sim \mathcal{N}\left(\boldsymbol{A}\hat{\boldsymbol{x}}, \boldsymbol{APA}^{\mathsf{T}}\right)$

In other words, mapping by $\boldsymbol{A}$ produces

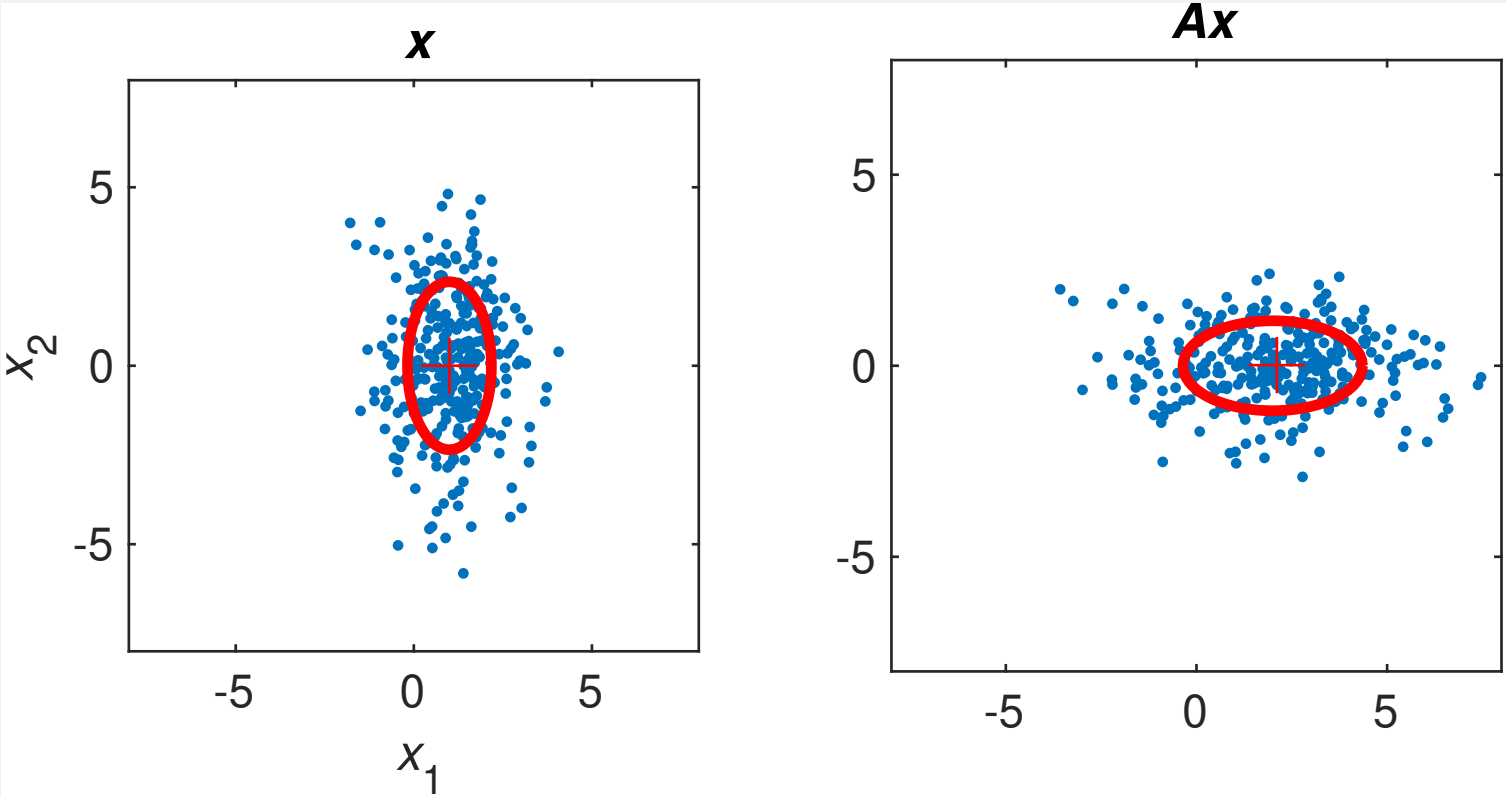
$$\hat{\boldsymbol{x}} \mapsto \boldsymbol{A}\hat{\boldsymbol{x}}$$

$$\boldsymbol{P} \mapsto \boldsymbol{APA}^{\mathsf{T}}$$

This should not be too startling. You may recall that if there is a scalar variable $x \sim \mathcal{N}\left(\mu, \sigma^2\right)$, then $ax \sim \mathcal{N}\left(a\mu, a^2\sigma^2\right)$.

Example

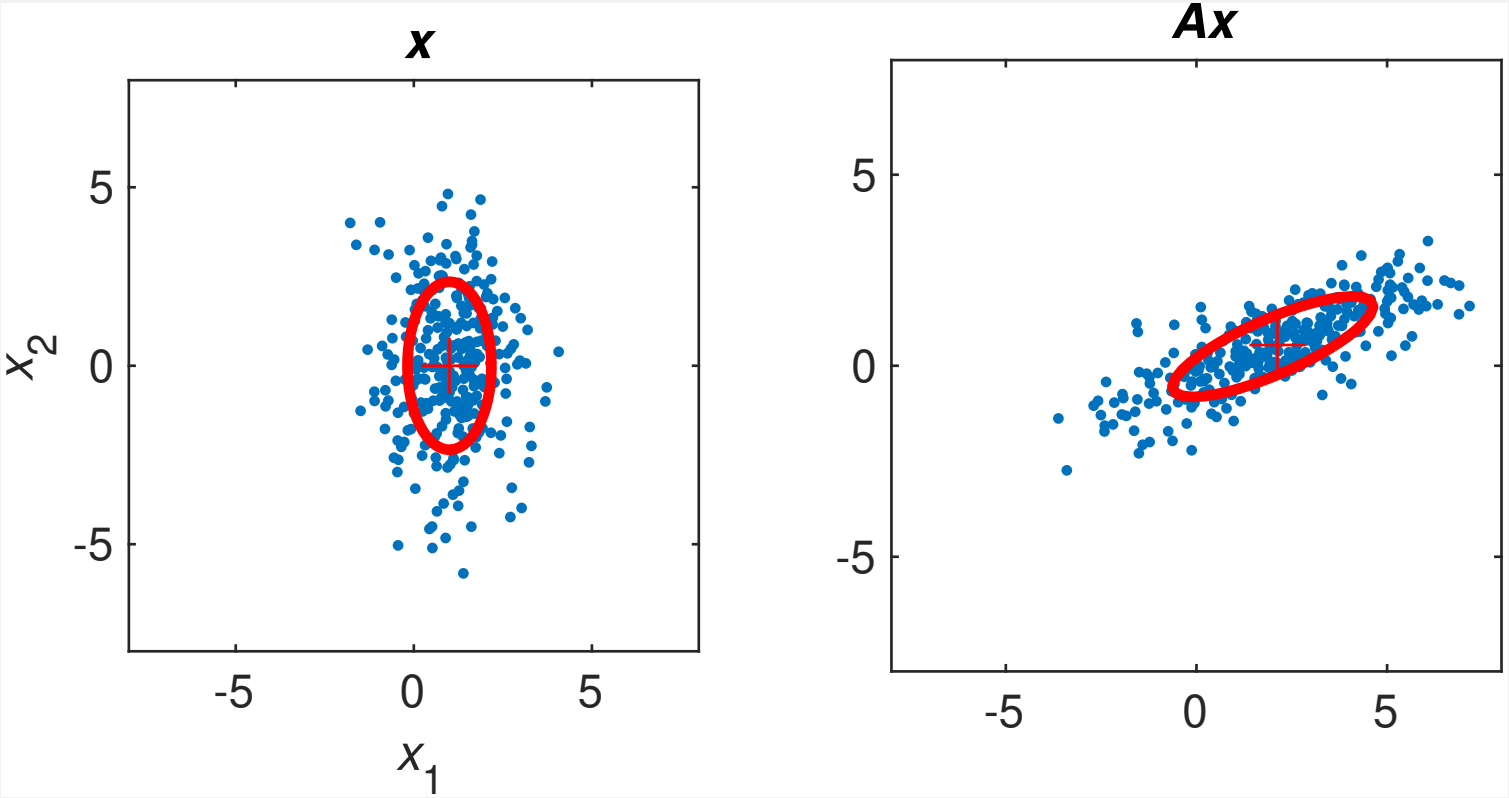
$$\boldsymbol{x} \sim \mathcal{N}\left(\begin{bmatrix}1\\0\end{bmatrix}, \begin{bmatrix}1 & 0\\0 & 4\end{bmatrix}\right), \boldsymbol{A} = \begin{bmatrix}2 & 0\\0 & 0.5\end{bmatrix} \mapsto \boldsymbol{Ax} \sim \mathcal{N}\left(\begin{bmatrix}2\\0\end{bmatrix}, \begin{bmatrix}4 & 0\\0 & 1\end{bmatrix}\right)$$



The red cross and ellipse indicate the mean and the 50% confidence region.

Example

$$\boldsymbol{x} \sim \mathcal{N}\left(\begin{bmatrix}1\\0\end{bmatrix}, \begin{bmatrix}1 & 0\\0 & 4\end{bmatrix}\right), \boldsymbol{A} = \begin{bmatrix}2 & 0.5\\0.5 & 0.5\end{bmatrix} \mapsto \boldsymbol{Ax} \sim \mathcal{N}\left(\begin{bmatrix}2\\0.5\end{bmatrix}, \begin{bmatrix}5 & 2\\2 & \frac{5}{4}\end{bmatrix}\right)$$



The red cross and ellipse indicate the mean and the 50% confidence region.

Generating noise in Matlab

When simulating perturbed systems we often need to generate noise signals with a specified covariance. The equations above tell us how we can form any desired covariance by beginning with a white noise signal and mapping it with an appropriate matrix. Say we want to create a noise signal having covariance $\boldsymbol{Q}$.

We begin with a white noise signal having covariance $\boldsymbol{I}$; uncorrelated noise in each state variable with unit variance. We then map with some appropriate matrix $\boldsymbol{M}$ to get the desired $\boldsymbol{Q}$.

$$\begin{aligned}\boldsymbol{Q} &= \boldsymbol{M}\boldsymbol{I}\boldsymbol{M}^{\mathrm{T}} \\ &= \boldsymbol{M}\boldsymbol{M}^{\mathrm{T}}\end{aligned}$$

That is, we must find some matrix $\boldsymbol{M}$ that satisfies the above expression. It transpires that this is always possible if $\boldsymbol{Q}$ is positive definite and can be performed using the Cholesky factorisation.

Generating noise in Matlab

The following Matlab commands will generate a sample of zero mean noise with covariance $\boldsymbol{Q}$.

Vector noise ($\boldsymbol{w} \in \mathbb{R}^n$): `w = chol(Q)*randn(n,1);`

The Cholesky factorization command `chol` returns one example of a matrix square root. We can see that the equation is simply a generalisation of the familiar method for generating scalar noise with specified variance.

Scalar noise ($\boldsymbol{w} \in \mathbb{R}$): `w = sqrt(Q)*randn(1);`

The Cholesky factorisation will fail if $\boldsymbol{Q}$ is singular. In such cases we can generate noise using singular value decomposition.

Continuous to discrete time noise conversion

Sometimes we are given a noise covariance matrix for a system in continuous time, but wish to use a discrete time model (or Kalman filter...).

Noise Q_c and R_c in a continuous time model can be approximated by a discrete time noise covariance

$$Q_d(t_s) = \int_0^{t_s} e^{\tau A} Q_c e^{\tau A^T} \, d\tau$$

$$R_d(t_s) = \frac{R_c}{t_s}$$

This looks a little messy (and it is), however for small systems it is reasonably straightforward to calculate Q_d via series expansion of the matrix exponential. You should expect the various entries of Q_d to include different powers of t_s.

Part III

The Kalman Filter

State Space Models – State

We are going to presume that everything we need to know about the world can be captured in a set of numbers, known as the "state". Here the state is made up of real numbers, because we care about things like position and velocity.

Each piece of information is called a *state variable*, and the various state variables are stacked together into a column vector, called the *state vector*, which is denoted $\boldsymbol{x}$.

For example, if we were doing a localisation problem on a single robot moving on the surface of the earth, then we might have a state vector defined as

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}, \text{ where } \begin{cases} x_1 \text{ is the latitude of the robot} \\ x_2 \text{ is the northwards velocity of the robot} \\ x_3 \text{ is the longitude of the robot} \\ x_4 \text{ is the eastwards velocity of the robot} \end{cases}$$

State Space Models – Estimates

Our job when performing SLAM, or other *state estimation* tasks is to determine an approximation for $\boldsymbol{x}$, which we will denote as $\hat{\boldsymbol{x}}$.

In doing simulation work we know what $\boldsymbol{x}$ really is, but of course in practical applications we do not (otherwise we would not be doing SLAM!). As a result it is useful to keep track of how accurate we think that our estimate might be.

We characterise our uncertainty about $\hat{\boldsymbol{x}}$ by a covariance matrix $\boldsymbol{P}$.

- Large diagonal elements within $\boldsymbol{P}$ indicate that we are uncertain about the corresponding state variable. For example, in our 2D robot above, a high value for p_{22} would indicate that we are uncertain about how fast the robot is moving northwards (or southwards).
- Note that we estimate $\boldsymbol{P}$. It tells us how certain we are about our guess, not anything about the underlying world.

State Space Models – Sensors

One of the ways we can estimate our state is to take measurements. That is, we use some sensors to attempt to measure some combination of the state. Typically in a SLAM application this would include things like odometry and range measurements to beacons or landmarks.

These measurements will be functions of the true state of the world. That is, we take a set of measurements y, where $y = h(\boldsymbol{x})$ for some function h. Sometimes we can simplify things by assuming that h is linear, in which case we usually write $\boldsymbol{y} = \boldsymbol{C}\boldsymbol{x}$ where $\boldsymbol{C}$ is called the sensor matrix.

- Unfortunately h is often *not* linear in SLAM problems, which causes much of the complexity that we will encounter later. We will begin with linear sensor functions to keep things manageable.

In practice we find that our sensor measurements always have a degree of uncertainty associated with them. For example, we might have a distance sensor that gives a distance ± 2 m. Again we will characterise this uncertainty with a covariance matrix, $\boldsymbol{R}$.

State Space Models – Evolution

We will assume that we know how to write an equation that describes how each state variable changes with time. That is, we would like to describe what will happen at the next time instant as a function of the present state ($\boldsymbol{x}$) and any inputs that we choose to apply to the system (denoted $\boldsymbol{u}$).

Let's return to our 2D robot example, and assume that our time samples are separated by a timestep of length Δt. Let's assume a (fairly silly) model where it is on a frictionless plane, so simply carries on at its current velocity for all time.

$$
\begin{aligned}
\boldsymbol{x}(t+1) &= f(\boldsymbol{x}, \boldsymbol{u}) \\
\rightsquigarrow\ x_1(t+1) &= s_{\mathrm{NS}} + v_{\mathrm{NS}}\Delta t && \equiv x_1(t) + x_2(t)\Delta t \\
x_2(t+1) &= v_{\mathrm{NS}} && \equiv x_2(t) \\
x_3(t+1) &= s_{\mathrm{EW}} + v_{\mathrm{EW}}\Delta t && \equiv x_3(t) + x_4(t)\Delta t \\
x_4(t+1) &= v_{\mathrm{EW}} && \equiv x_4(t)
\end{aligned}
$$

State Space Models – Evolution

Note that in this case we can write linear equations to describe how the state evolves. That is, in this case

$$\boldsymbol{x}(t+1) = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix} \boldsymbol{x}(t)$$

Where possible we will denote the state evolution equation as

$$\boldsymbol{x}(t+1) = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}$$

where $\boldsymbol{A}$ and $\boldsymbol{B}$ are matrices arising from the physics of the situation.

Again, we will generalise to the more complicated, nonlinear case later.

State Space Models – Inputs

There are two types of inputs that act on our system. The first, denoted $\boldsymbol{u}$ we have already seen. It captures things that we deliberately do to the system. For example, we might choose to drive forwards 2 m, or turn to the left by 15°.

The second set of inputs are *disturbances*. They model the random influences that the universe has on our system. For example, while we would struggle to write a simple model for them, we could statistically model wind, or wheel slippage, or uneven ground.

- We will call these random influences *plant disturbances* and denote them $\boldsymbol{w}$.
- We will characterise the size of the random effects with a covariance matrix $\boldsymbol{Q}$.

Kalman Filter Introduction

The Kalman filter allows us to build state estimators that account for the presence of noise in the measurements and disturbances in the plant.

The Kalman filter has access to two methods for inferring the state.

1. Use a prediction from an internal model.
2. Determine the state from measurements.

The Kalman filter finds the 'best' way to combine these two sources of information.

Outline

- A Kalman filter works by keeping track of the uncertainties in the model-based state estimate and the measurements. It was for this reason that we began our study of the Kalman filter by looking at how we describe the uncertainty of vector quantities.
- We will next see how the presence of feedback reduces the uncertainty in our estimate of the state.
- We will then derive expressions for the Kalman Filter gain. We will see that this leads to a set of equations that must be solved for each time step to see how strongly we should believe new measurements.
- We will finally look at the steady-state Kalman Filter, which is a simplified version of the full iterative filter that can often be used with little penalty.

Kalman Filter Introduction

Consider a discrete-time, linear dynamical system that is subject to a disturbance signal $\boldsymbol{w}$ and has measurements of its outputs contaminated by noise $\boldsymbol{v}$.

- $\boldsymbol{w}$ is called the process or disturbance noise.
- $\boldsymbol{v}$ is called the measurement noise.

For now we will assume that $\boldsymbol{v}$ and $\boldsymbol{w}$ are zero-mean, white, gaussian noise sources.

$$\begin{aligned}
\boldsymbol{x}(t) &= \boldsymbol{A}\boldsymbol{x}(t-1) + \boldsymbol{B}\boldsymbol{u}(t-1) + \boldsymbol{G}\boldsymbol{w}(t-1) \\
\boldsymbol{y}(t) &= \boldsymbol{C}\boldsymbol{x}(t) + \boldsymbol{H}\boldsymbol{v}(t) \\
\boldsymbol{w} &\sim \mathcal{N}(0, \boldsymbol{Q}) \\
\boldsymbol{v} &\sim \mathcal{N}(0, \boldsymbol{R}) \\
t &\in \mathbb{Z}_+
\end{aligned}$$

Noise coupling in the Kalman filter

The matrices $\boldsymbol{G}$ and $\boldsymbol{H}$ in the above representation describe the way in which noise is coupled into the state and the measurements respectively. Note that $\boldsymbol{G}$ is similar to $\boldsymbol{B}$ in that it couples inputs (unknown inputs in this case) into the state. For simpliciry we will assume that $\boldsymbol{G} = \boldsymbol{I}$ throughout our derivation. However, we will see that using $\boldsymbol{G} \neq \boldsymbol{I}$ is quite useful in practical problems, because there are interrelationships between the ways in which $\boldsymbol{w}$ affects the various state variables. While we assume $\boldsymbol{G} = \boldsymbol{I}$ in the derivations to follow, we could replace $\boldsymbol{Q}$ with $\boldsymbol{G}\boldsymbol{Q}\boldsymbol{G}^{\mathsf{T}}$ throughout to keep the treatment general.

Similarly, the matrix $\boldsymbol{H}$ couples noise into the measurements. It is almost always sensible to use $\boldsymbol{H} = \boldsymbol{I}$, because we typically have independent noise in the various measurements. However, there are situations where we have correlated interfefence in multiple sensors, so using a more interesting $\boldsymbol{H}$ matrix is sometimes useful. Examples of this would be attempting to capture the effecst of mains interefence, or high frequency intereference signal from an MRI machine, or a radio transmitter.

Noise in Kalman filter

The assumption that $\boldsymbol{w}$ and $\boldsymbol{v}$ are zero-mean, white and independent noise sources mean that

$$
\begin{aligned}
\mathrm{E}\{\boldsymbol{w}(t)\} &= 0 \\
\mathrm{E}\{\boldsymbol{v}(t)\} &= 0 \\
\mathrm{E}\left\{\boldsymbol{w}(t)\boldsymbol{w}(t+\tau)^{\mathsf{T}}\right\} &= \boldsymbol{Q}\,\delta_\tau \qquad \Longrightarrow \ \mathrm{E}\left\{\boldsymbol{w}(t)\boldsymbol{w}(t)^{\mathsf{T}}\right\} = \boldsymbol{Q} \\
\mathrm{E}\left\{\boldsymbol{v}(t)\boldsymbol{v}(t+\tau)^{\mathsf{T}}\right\} &= \boldsymbol{R}\,\delta_\tau \qquad \Longrightarrow \ \mathrm{E}\left\{\boldsymbol{v}(t)\boldsymbol{v}(t)^{\mathsf{T}}\right\} = \boldsymbol{R} \\
\mathrm{E}\left\{\boldsymbol{w}(t)\boldsymbol{v}(t+\tau)^{\mathsf{T}}\right\} &= 0 \qquad \text{for simplicity}
\end{aligned}
$$

where δ is the Kronecker delta. The values of $\boldsymbol{w}$ and $\boldsymbol{v}$ at one instant are completely uncorrelated with the values at any other instant because the noise sources are white.

Aim of the Kalman filter

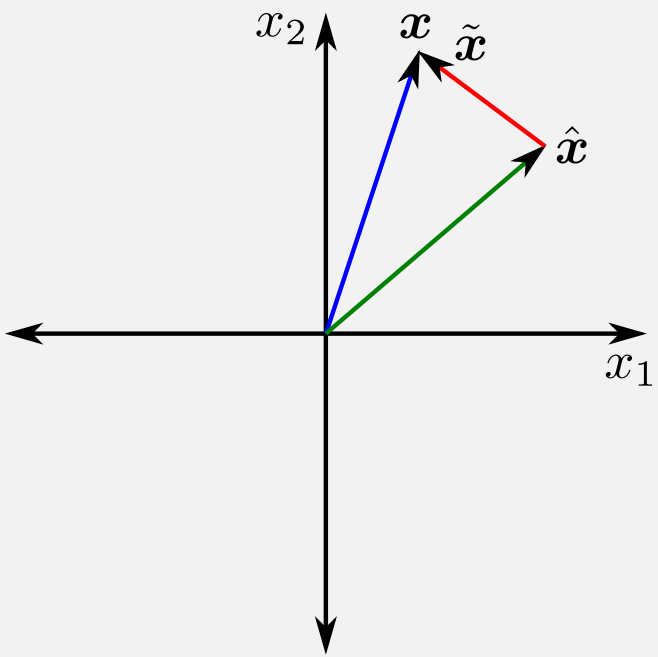
The aim of the Kalman filter is to produce the state estimator with the lowest covariance in the error of the estimate. That is, we seek the 'best' estimate of the state given the system noise.

We make the assumption that we can measure (or guess) the noise covariances for both the plant and the measurements. Sometimes this is easy (for example we can measure the output noise of a sensor), sometimes it is more difficult.

Formally, the Kalman filter attempts to minimize the square of the length of the error vector between $\boldsymbol{x}$ and its estimate $\hat{\boldsymbol{x}}$. That is, we seek to minimise the expected length of $\tilde{\boldsymbol{x}} = \boldsymbol{x} - \hat{\boldsymbol{x}}$.

We choose to minimise $\|\tilde{\boldsymbol{x}}\|$ because it is mathematically tractable. This choice allows us to derive a closed form solution to our problem, whereas many other sensible choices lead to solutions that must be solved using numerical optimization methods.

Aim of the Kalman filter



$\tilde{\boldsymbol{x}}$ has length $\|\tilde{\boldsymbol{x}}\|$.

That is, we are attempting to minimise

$$
\begin{aligned}
J &:= \mathrm{E}\left\{ \|\tilde{\boldsymbol{x}}\|^2 \right\} \\
&= \sum_i \mathrm{E}\left\{ \|\tilde{x}_i\|^2 \right\} \\
&= \sum_i \mathrm{E}\left\{ \|x_i - \hat{x}_i\|^2 \right\} \\
&= \mathrm{var}(\hat{x}_1) + \mathrm{var}(\hat{x}_2) + \cdots + \mathrm{var}(\hat{x}_n) \\
&= \mathrm{tr}(\mathrm{cov}\, \hat{\boldsymbol{x}}) \\
&= \mathrm{tr}\, \boldsymbol{P}
\end{aligned}
$$

with tr being the trace operator, which sums the diagonal of a matrix.

Derivation of the discrete-time Kalman filter

We would like to find some feedback gain $\boldsymbol{L}$ which we can use to update our estimate of the state. That is, we are going to form a model that tells us how to update the state estimate,

$$\hat{\boldsymbol{x}}(t+1) = \boldsymbol{A}\hat{\boldsymbol{x}} + \boldsymbol{B}\boldsymbol{u} + \boldsymbol{L}(\boldsymbol{y} - \hat{\boldsymbol{y}})$$

Plan:

1. Determine how we would make a prediction of the next state given our model of the system and information that we know. That is, $\hat{\boldsymbol{x}}(t+1) = \boldsymbol{A}\hat{\boldsymbol{x}} + \boldsymbol{B}\boldsymbol{u}$.
2. Determine how this estimate can be improved once we take another measurement of the output.
3. Choose a feedback gain $\boldsymbol{L}$ that best combines the prediction and the measurement. We will find that $\boldsymbol{L}$ changes with time (unlike our previous observers).

The derivation for the last part of this procedure becomes quite involved. You do not need to be able to reproduce it, but you should understand the general flow of the argument.

Notation

Prior:
$\hat{\boldsymbol{x}}(t|t-1)$ is our estimate of $\boldsymbol{x}$ at time t, given only our knowledge of the system at time $t-1$. That is, we assume that we know $\hat{\boldsymbol{x}}(t-1|t-1)$, $\boldsymbol{u}(t-1)$ and $\boldsymbol{y}(t-1)$, as well as all of their previous values.

Posterior:
$\hat{\boldsymbol{x}}(t|t)$ is our estimate of $\boldsymbol{x}$ at time t given the information above *plus* $\boldsymbol{y}(t)$, the measurement at time t.

The notion of a prior/posterior refers to our knowledge before/after the incorporation of a measurement.

We can extend this notation to other times and quantities:
$\boldsymbol{P}(t|t)$ is the posterior covariance of $\hat{\boldsymbol{x}}$ at time t.
$\hat{\boldsymbol{x}}(t|t-4)$ is the estimate of $\boldsymbol{x}$ at time t given knowledge of the system up until time $t-4$.

Prediction of the next state

Imagine that we know the state of the system at a time $t-1$ and we want to predict its state at time t. To make this prediction we can use our estimate of the previous state $\hat{\boldsymbol{x}}(t-1|t-1)$ and our previous estimate of its covariance $\boldsymbol{P}(t-1|t-1)$, along with our knowledge of the input that acted on the system in the last time step.

$$\hat{\boldsymbol{x}}(t|t-1) = \boldsymbol{A}\hat{\boldsymbol{x}}(t-1|t-1) + \boldsymbol{B}\boldsymbol{u}(t-1)$$
$$\boldsymbol{P}(t|t-1) = \boldsymbol{A}\boldsymbol{P}(t-1|t-1)\boldsymbol{A}^{\mathrm{T}} + \boldsymbol{Q}$$
$$\hat{\boldsymbol{y}}(t|t-1) = \boldsymbol{C}\hat{\boldsymbol{x}}(t|t-1)$$

The expression for $\hat{\boldsymbol{x}}(t|t-1)$ and $\hat{\boldsymbol{y}}(t|t-1)$ are the familiar state evolution equations with an extension to the notation to make explicit the information leading to our estimates.

Prediction of the next state

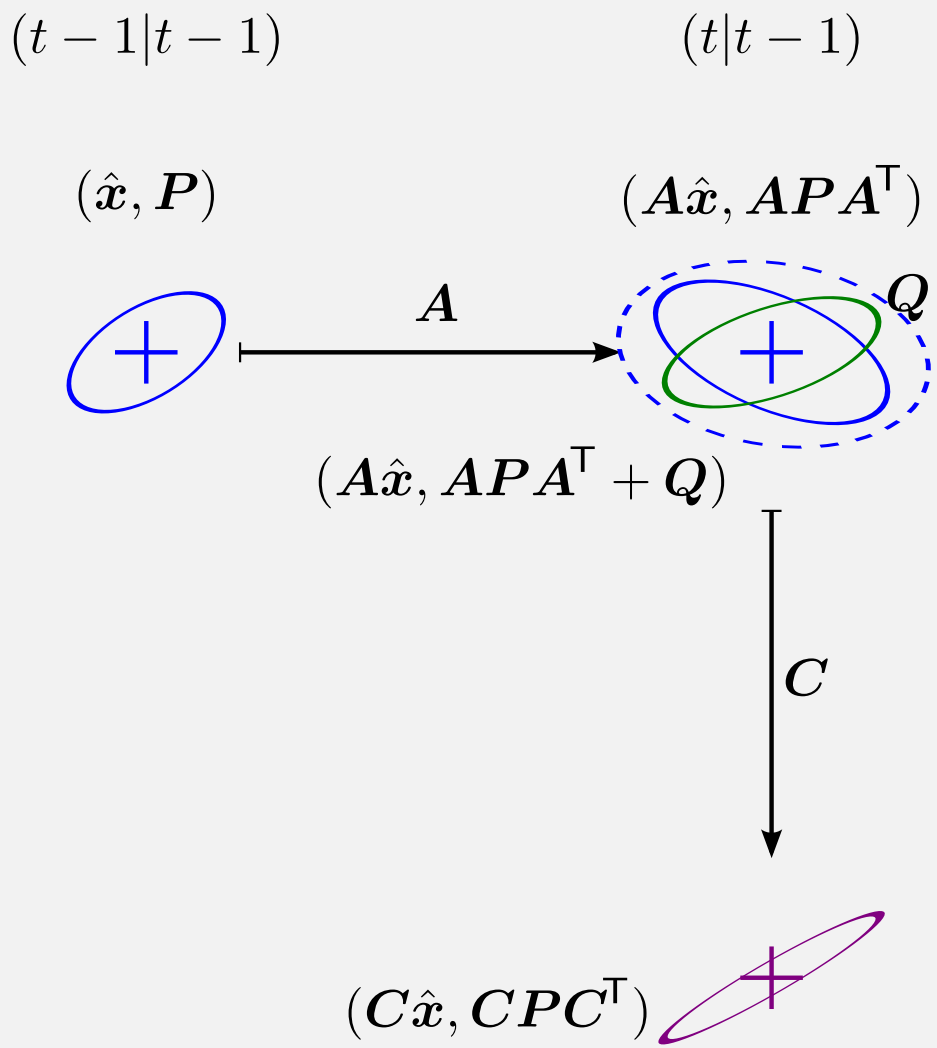
Recall that $\boldsymbol{x}$ is assumed to be disturbed by a noise source $\boldsymbol{w}$ and $\boldsymbol{y}$ is contaminated by the noise source $\boldsymbol{v}$. However, because $\boldsymbol{w}$ and $\boldsymbol{v}$ are zero mean, our predictions contain no noise terms. In other words we predict that that, on average, the noise sources will not perturb the system.

The new covariance of our estimate consists of two components.

- $\boldsymbol{APA}^{\mathsf{T}}$ arises when we take the previous estimate of the covariance $\boldsymbol{P}(t-1|t-1)$ and act on $\hat{\boldsymbol{x}}$ with $\boldsymbol{A}$.
- The second term is the covariance of the process noise $\boldsymbol{w}$. While we cannot predict how this noise source will perturb the system, we do know that it *almost surely* has some effect. Our uncertainty about the effect is described by $\boldsymbol{Q}$, so we add this uncertainty.

Note that it is only correct to add uncertainties in this way when the two effects are uncorrelated.

Prediction of the next state



Output error

We now take a measurement of the output $\boldsymbol{y}(t)$ and we want find the error in our estimate. We can first calculate the error in our estimate of the system output,

$$\begin{aligned}\tilde{\boldsymbol{y}}(t|t) &:= \boldsymbol{y}(t) - \hat{\boldsymbol{y}}(t|t-1) \\ &= \boldsymbol{y}(t) - \boldsymbol{C}\hat{\boldsymbol{x}}(t|t-1)\end{aligned}$$

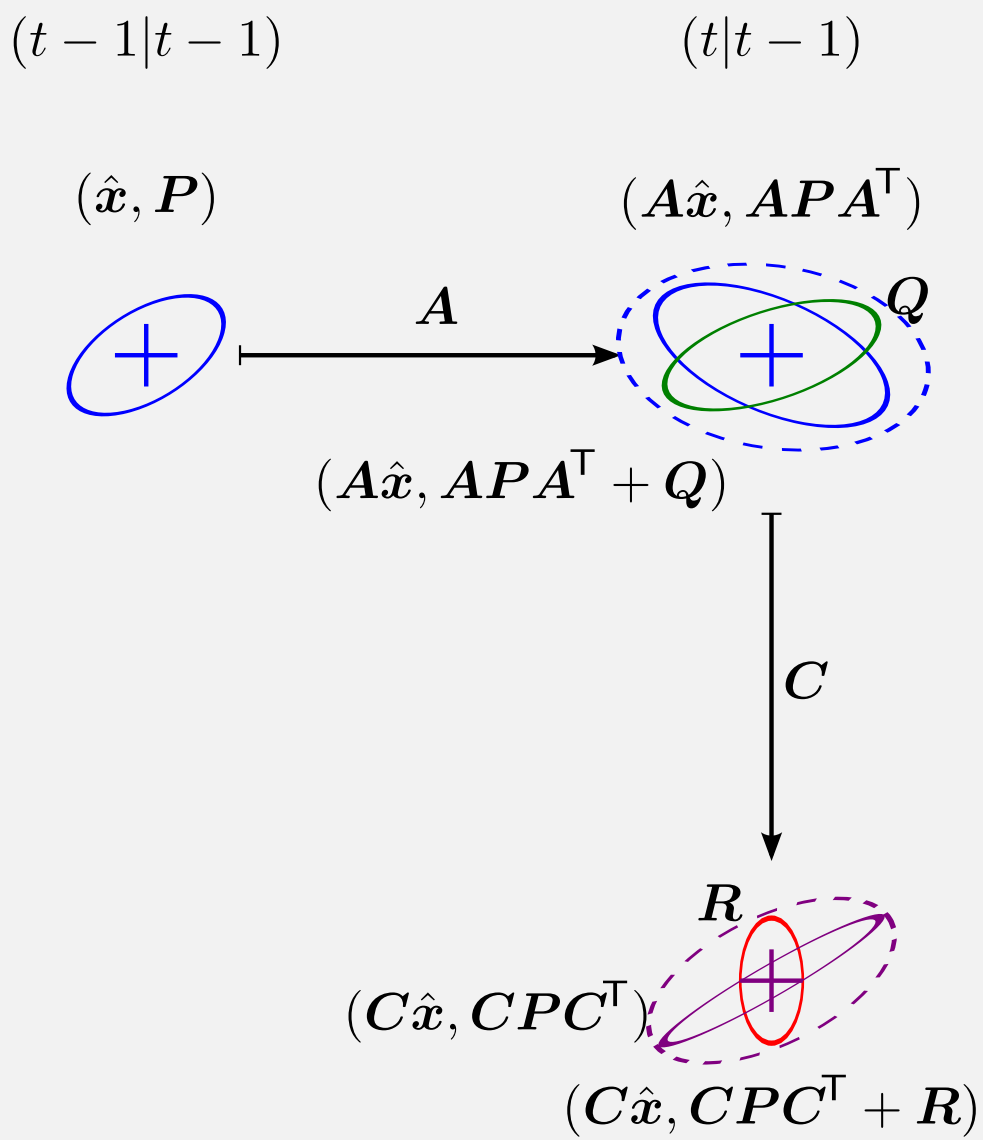
We also need to calculate how certain we are about that error. We know that our uncertainty in the state estimate is $\boldsymbol{P}(t|t-1)$. The resulting covariance in the output error is $\boldsymbol{C}\boldsymbol{P}(t|t-1)\boldsymbol{C}^{\mathsf{T}}$.

However, we also know that $\boldsymbol{y}$ is contaminated by $\boldsymbol{v}$ with covariance $\boldsymbol{R}$, so we must also add this covariance.

Thus, if the covariance of the error in the estimate of $\boldsymbol{y}$ is denoted $\boldsymbol{S}$, we can write

$$\boldsymbol{S} = \boldsymbol{C}\boldsymbol{P}(t|t-1)\boldsymbol{C}^{\mathsf{T}} + \boldsymbol{R} \quad .$$

Output Error



Finding $\boldsymbol{P}(t|t)$ with feedback present

We now use feedback with gain $\boldsymbol{L}$ to correct our estimate of $\boldsymbol{x}$. Let's calculate what effect this has on $\boldsymbol{P}(t|t)$.

$$
\begin{aligned}
\boldsymbol{P}(t|t) &= \operatorname{cov}\left(\tilde{\boldsymbol{x}}\right) \\
&= \operatorname{cov}(\boldsymbol{x} - \hat{\boldsymbol{x}}(t|t)) \\
&= \operatorname{cov}\left(\boldsymbol{x} - \left(\hat{\boldsymbol{x}}(t|t-1) + \boldsymbol{L}\tilde{\boldsymbol{y}}\right)\right) \\
&= \operatorname{cov}\left(\boldsymbol{x} - \hat{\boldsymbol{x}}(t|t-1) - \boldsymbol{L}\tilde{\boldsymbol{y}}\right) \\
&= \operatorname{cov}\left(\boldsymbol{x} - \hat{\boldsymbol{x}}(t|t-1) - \boldsymbol{L}\left(\boldsymbol{y} - \hat{\boldsymbol{y}}\right)\right) \\
&= \operatorname{cov}\left(\boldsymbol{x} - \hat{\boldsymbol{x}}(t|t-1) - \boldsymbol{L}\big(\boldsymbol{C}\boldsymbol{x} + \boldsymbol{v} - \boldsymbol{C}\hat{\boldsymbol{x}}(t|t-1)\big)\right) \\
&= \operatorname{cov}\left(\left(\boldsymbol{I} - \boldsymbol{L}\boldsymbol{C}\right)\left(\boldsymbol{x} - \hat{\boldsymbol{x}}(t|t-1)\right) - \boldsymbol{L}\boldsymbol{v}\right) \\
&= \operatorname{cov}\left(\left(\boldsymbol{I} - \boldsymbol{L}\boldsymbol{C}\right)\left(\boldsymbol{x} - \hat{\boldsymbol{x}}(t|t-1)\right)\right) + \operatorname{cov}(\boldsymbol{L}\boldsymbol{v}) \\
&= \left(\boldsymbol{I} - \boldsymbol{L}\boldsymbol{C}\right)\operatorname{cov}(\boldsymbol{x} - \hat{\boldsymbol{x}}(t|t-1))\left(\boldsymbol{I} - \boldsymbol{L}\boldsymbol{C}\right)^{\mathrm{T}} + \boldsymbol{L}\operatorname{cov}(\boldsymbol{v})\boldsymbol{L}^{\mathrm{T}} \\
\boldsymbol{P}(t|t) &= \left(\boldsymbol{I} - \boldsymbol{L}\boldsymbol{C}\right)\boldsymbol{P}(t|t-1)\left(\boldsymbol{I} - \boldsymbol{L}\boldsymbol{C}\right)^{\mathrm{T}} + \boldsymbol{L}\boldsymbol{R}\boldsymbol{L}^{\mathrm{T}}
\end{aligned}
$$

Finding $\boldsymbol{L}$

Remember that we are trying to minimize $\mathrm{tr}\left(\boldsymbol{P}(t|t)\right)$. That is, we want to find the value of $\boldsymbol{L}$ that makes the trace of the above expression as small as possible.

To do this we will take the derivative of $\mathrm{tr}\left(\boldsymbol{P}(t|t)\right)$ with respect to $\boldsymbol{L}$ and set it to zero. The value of $\boldsymbol{L}$ that results in a zero derivative is the one that will minimize the expected error in out estimate.

We will allow $\boldsymbol{L}$ to be time varying, however for brevity we will not explicitly write out the time variation in the derivation that follows. For every occurrence of $\boldsymbol{L}$ you should read $\boldsymbol{L}(t)$.

We don't know how to find $\frac{\partial}{\partial \boldsymbol{L}} \mathrm{tr}\left(\left(\boldsymbol{I}-\boldsymbol{L}\boldsymbol{C}\right)\boldsymbol{P}(t|t-1)\left(\boldsymbol{I}-\boldsymbol{L}\boldsymbol{C}\right)^{\mathrm{T}}\right)$, so we will begin by simplifying the expression for $\boldsymbol{P}(t|t)$.

Finding L

$$
\begin{aligned}
\boldsymbol{P}(t|t) &= (\boldsymbol{I} - \boldsymbol{LC})\,\boldsymbol{P}(t|t-1)\,(\boldsymbol{I} - \boldsymbol{LC})^{\mathsf{T}} + \boldsymbol{LRL}^{\mathsf{T}} \\
&= \boldsymbol{IP}(t|t-1)\boldsymbol{I}^{\mathsf{T}} - \boldsymbol{IP}(t|t-1)(\boldsymbol{LC})^{\mathsf{T}} - \boldsymbol{LCP}(t|t-1)\boldsymbol{I}^{\mathsf{T}} \\
&\quad + \boldsymbol{LCP}(t|t-1)(\boldsymbol{LC})^{\mathsf{T}} + \boldsymbol{LRL}^{\mathsf{T}} \\
&= \boldsymbol{P}(t|t-1) - \boldsymbol{P}(t|t-1)\boldsymbol{C}^{\mathsf{T}}\boldsymbol{L}^{\mathsf{T}} - \boldsymbol{LCP}(t|t-1) \\
&\quad + \boldsymbol{LCP}(t|t-1)\boldsymbol{C}^{\mathsf{T}}\boldsymbol{L}^{\mathsf{T}} + \boldsymbol{LRL}^{\mathsf{T}} \\
&= \boldsymbol{P}(t|t-1) - \boldsymbol{P}(t|t-1)\boldsymbol{C}^{\mathsf{T}}\boldsymbol{L}^{\mathsf{T}} - \boldsymbol{LCP}(t|t-1) \\
&\quad + \boldsymbol{L}\left(\boldsymbol{CP}(t|t-1)\boldsymbol{C}^{\mathsf{T}} + \boldsymbol{R}\right)\boldsymbol{L}^{\mathsf{T}} \\
&= \boldsymbol{P}(t|t-1) - \boldsymbol{P}(t|t-1)\boldsymbol{C}^{\mathsf{T}}\boldsymbol{L}^{\mathsf{T}} - \boldsymbol{LCP}(t|t-1) + \boldsymbol{LSL}^{\mathsf{T}} \\
&= \boldsymbol{P}(t|t-1) - \boldsymbol{LCP}(t|t-1)^{\mathsf{T}} - \boldsymbol{LCP}(t|t-1) + \boldsymbol{LSL}^{\mathsf{T}} \\
\boldsymbol{P}(t|t) &= \boldsymbol{P}(t|t-1) - 2\boldsymbol{LCP}(t|t-1) + \boldsymbol{LSL}^{\mathsf{T}}
\end{aligned}
$$

Finding L

We now take the derivative of $\mathrm{tr}\left(\boldsymbol{P}(t|t)\right)$ with respect to the Kalman gain $\boldsymbol{L}$.

$$\begin{aligned}\frac{\partial \mathrm{tr}(\boldsymbol{P}(t|t))}{\partial \boldsymbol{L}} &= \frac{\partial \mathrm{tr}(\boldsymbol{P}(t|t-1))}{\partial \boldsymbol{L}} \\ &\quad - 2\frac{\partial \mathrm{tr}(\boldsymbol{LCP}(t|t-1))}{\partial \boldsymbol{L}} \\ &\quad + \frac{\partial \mathrm{tr}(\boldsymbol{LSL}^{\mathrm{T}})}{\partial \boldsymbol{L}} \\ &= -2\left(\boldsymbol{CP}(t|t-1)\right)^{\mathrm{T}} + \boldsymbol{LS}^{\mathrm{T}} + \boldsymbol{LS} \\ &= -2(\boldsymbol{CP}(t|t-1))^{\mathrm{T}} + 2\boldsymbol{LS} \quad \text{as } \boldsymbol{S} \text{ is symmetric}\end{aligned}$$

Finding L

We now equate the derivative to zero and solve for $\boldsymbol{L}$.

$$
\begin{aligned}
0 &= 2\boldsymbol{L}\boldsymbol{S} - 2\big(\boldsymbol{C}\boldsymbol{P}(t|t-1)\big)^{\mathsf{T}} \\
2\boldsymbol{L}\boldsymbol{S} &= 2\big(\boldsymbol{C}\boldsymbol{P}(t|t-1)\big)^{\mathsf{T}} \\
\boldsymbol{L} &= \big(\boldsymbol{C}\boldsymbol{P}(t|t-1)\big)^{\mathsf{T}}\boldsymbol{S}^{-1} \\
\boldsymbol{L} &= \boldsymbol{P}(t|t-1)^{\mathsf{T}}\boldsymbol{C}^{\mathsf{T}}\boldsymbol{S}^{-1} \\
\boldsymbol{L} &= \boldsymbol{P}(t|t-1)\boldsymbol{C}^{\mathsf{T}}\boldsymbol{S}^{-1}
\end{aligned}
$$

Interpreting the L equation

Finally, replacing $\boldsymbol{S}$ with its definition we obtain

$$\boldsymbol{L}(t) = \boldsymbol{P}(t|t-1)\boldsymbol{C}^{\mathsf{T}}\left(\boldsymbol{C}\boldsymbol{P}(t|t-1)\boldsymbol{C}^{\mathsf{T}} + \boldsymbol{R}\right)^{-1}$$

Note that the Kalman filter gain $\boldsymbol{L}$ will change as our certainties about the system evolve.

When the uncertainty in the state is high ($\boldsymbol{P}$ is large) then a greater weighting is given to the measurement.

When $\boldsymbol{P}$ becomes small then $\boldsymbol{L}$ also becomes small. This means that the amount of feedback is small, so the Kalman filter does not give much regard to the measurement, but relies on its own prediction of the state.

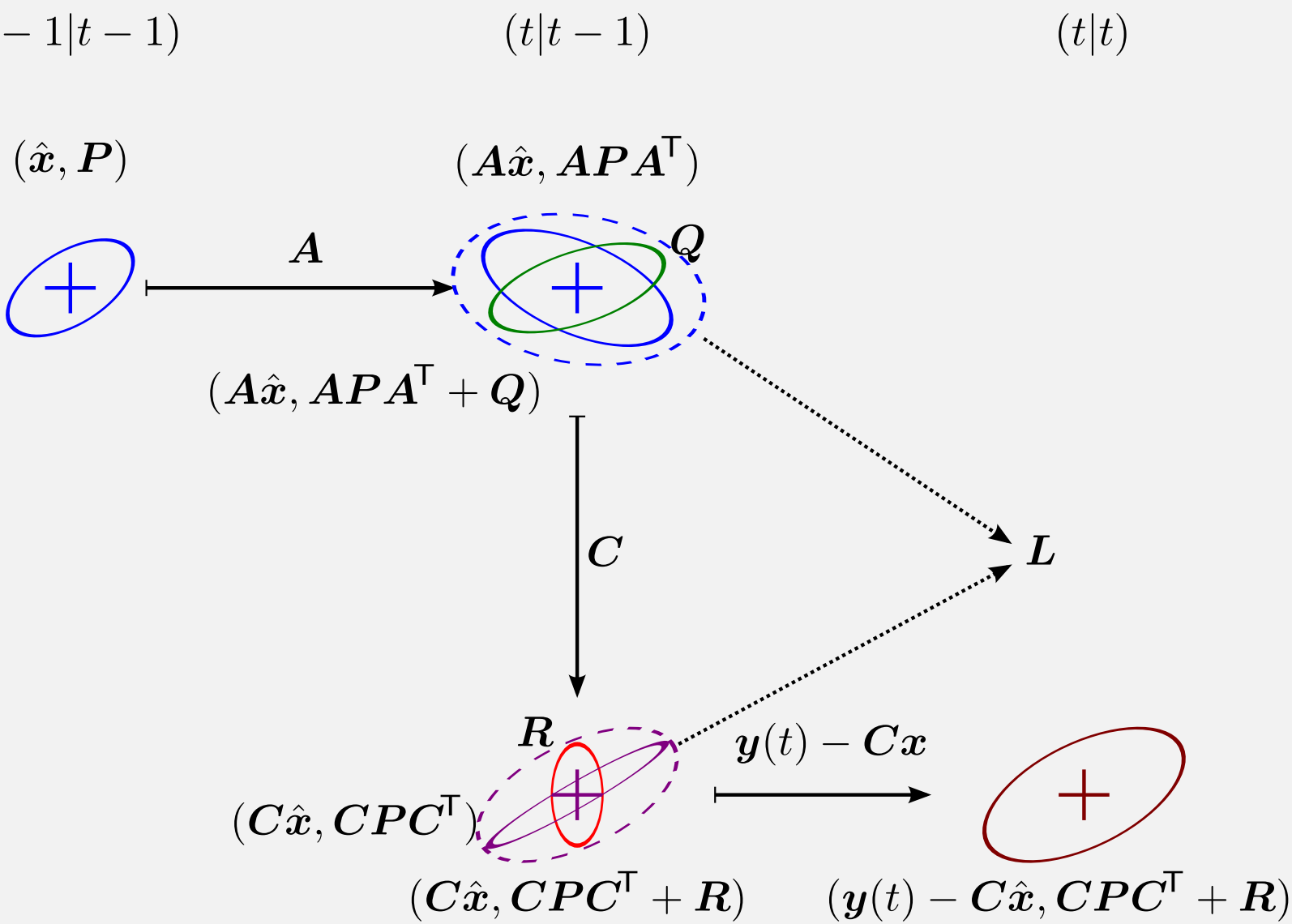
Correcting the state estimate

Knowledge of $\boldsymbol{L}$ allows us to update our state estimate.

$$\begin{aligned}\hat{\boldsymbol{x}}(t|t) &= \hat{\boldsymbol{x}}(t|t-1) + \boldsymbol{L}\tilde{\boldsymbol{y}} \\ &= \hat{\boldsymbol{x}}(t|t-1) + \boldsymbol{L}\big(\boldsymbol{y}(t) - \boldsymbol{C}\hat{\boldsymbol{x}}(t|t-1)\big)\end{aligned}$$

This is identical to our previous observer studies. The only difference is that $\boldsymbol{L}$ is now changing.

Correcting the state estimate



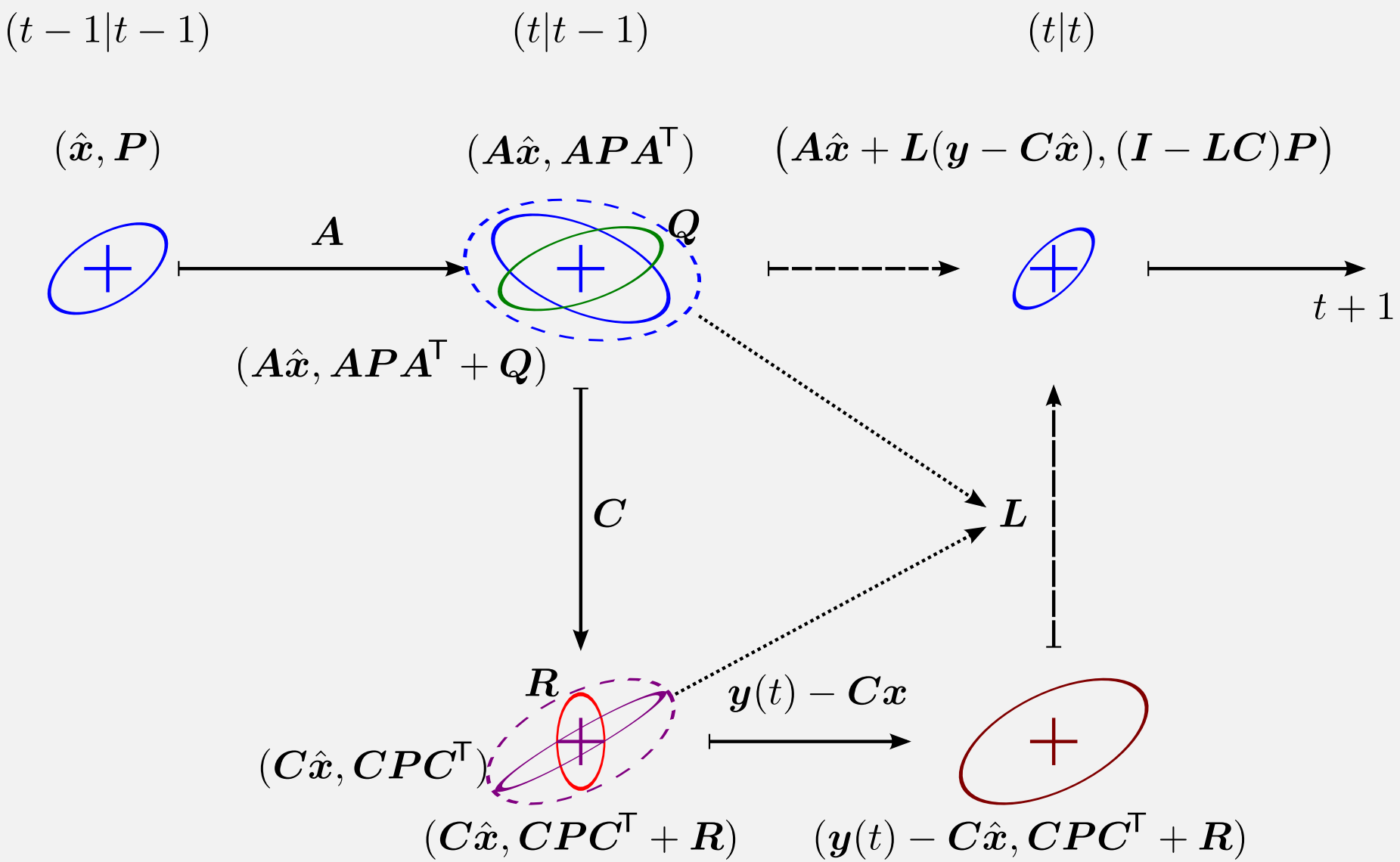
Correcting the covariance estimate

Now that we know $\boldsymbol{L}$ we can also calculate our estimate of the covariance matrix once we have a new measurement $\boldsymbol{y}(t)$.

$$
\begin{aligned}
\boldsymbol{P}(t|t) &= \boldsymbol{P}(t|t-1) - 2\boldsymbol{LCP}(t|t-1) + \boldsymbol{LSL}^{\mathsf{T}} \\
&= \boldsymbol{P}(t|t-1) - 2\boldsymbol{P}(t|t-1)\boldsymbol{C}^{\mathsf{T}}\boldsymbol{S}^{-1}\boldsymbol{CP}(t|t-1) \\
&\quad + \boldsymbol{P}(t|t-1)\boldsymbol{C}^{\mathsf{T}}\boldsymbol{S}^{-1}\boldsymbol{S}\left(\boldsymbol{P}(t|t-1)\boldsymbol{C}^{\mathsf{T}}\boldsymbol{S}^{-1}\right)^{\mathsf{T}} \\
&= \boldsymbol{P}(t|t-1) - 2\boldsymbol{P}(t|t-1)\boldsymbol{C}^{\mathsf{T}}\boldsymbol{S}^{-1}\boldsymbol{CP}(t|t-1) \\
&\quad + \boldsymbol{P}(t|t-1)\boldsymbol{C}^{\mathsf{T}}\boldsymbol{S}^{-1}\boldsymbol{S}\boldsymbol{S}^{-1}\boldsymbol{CP}(t|t-1) \qquad \text{as } \boldsymbol{P}, \boldsymbol{S} \text{ symmetric} \\
&= \boldsymbol{P}(t|t-1) - 2\boldsymbol{P}(t|t-1)\boldsymbol{C}^{\mathsf{T}}\boldsymbol{S}^{-1}\boldsymbol{CP}(t|t-1) \\
&\quad + \boldsymbol{P}(t|t-1)\boldsymbol{C}^{\mathsf{T}}\boldsymbol{S}^{-1}\boldsymbol{CP}(t|t-1) \\
&= \boldsymbol{P}(t|t-1) - \boldsymbol{P}(t|t-1)\boldsymbol{C}^{\mathsf{T}}\boldsymbol{S}^{-1}\boldsymbol{CP}(t|t-1) \\
&= \boldsymbol{P}(t|t-1) - \boldsymbol{P}(t|t-1)\boldsymbol{C}^{\mathsf{T}}\left[\boldsymbol{CP}(t|t-1)\boldsymbol{C}^{\mathsf{T}} + \boldsymbol{R}\right]^{-1}\boldsymbol{CP}(t|t-1) \\
&= \boldsymbol{P}(t|t-1) - \boldsymbol{LCP}(t|t-1) \implies \boldsymbol{P}(t|t) = (\boldsymbol{I} - \boldsymbol{LC})\,\boldsymbol{P}(t|t-1)
\end{aligned}
$$

This equation allows us to update our confidence in the estimate for $\boldsymbol{x}$.

Correcting the covariance estimate



Kalman Filter Summary

We now have a complete procedure for managing our estimate.

Initialisation: (To reflect knowledge of the starting point)

$$\hat{\boldsymbol{x}}(0|-1) = \boldsymbol{x}_0$$
$$\boldsymbol{P}(0|-1) = \boldsymbol{P}_0$$

Prediction: (Prior estimates)

$$\hat{\boldsymbol{x}}(t|t-1) = \boldsymbol{A}\hat{\boldsymbol{x}}(t-1|t-1) + \boldsymbol{B}\boldsymbol{u}(t-1)$$
$$\boldsymbol{P}(t|t-1) = \boldsymbol{A}\boldsymbol{P}(t-1|t-1)\boldsymbol{A}^{\mathsf{T}} + \boldsymbol{Q}$$

Correction: (Posterior estimates)

$$\boldsymbol{L}(t) = \boldsymbol{P}(t|t-1)\boldsymbol{C}^{\mathsf{T}}\left[\boldsymbol{C}\boldsymbol{P}(t|t-1)\boldsymbol{C}^{\mathsf{T}} + \boldsymbol{R}\right]^{-1}$$
$$\hat{\boldsymbol{x}}(t|t) = \hat{\boldsymbol{x}}(t|t-1) + \boldsymbol{L}\big(\boldsymbol{y}(t) - \boldsymbol{C}\hat{\boldsymbol{x}}(t|t-1)\big)$$
$$\boldsymbol{P}(t|t) = (\boldsymbol{I} - \boldsymbol{L}\boldsymbol{C})\,\boldsymbol{P}(t|t-1)$$

Practical considerations

In practice the calculation of $\boldsymbol{P}(t|t) = (\boldsymbol{I} - \boldsymbol{LC})\,\boldsymbol{P}(t|t-1)$ is not very well behaved numerically. Accumulation of errors as the filter is run multiple times means that $\boldsymbol{P}$ can lose its necessary positive definiteness. One reasonable effective strategy for dealing with this is to ensure that $\boldsymbol{P}$ remains symmetric at every time step (or occasionally) by setting

$$\boldsymbol{P}(t|t) = \frac{1}{2}\left(\boldsymbol{P}(t|t) + \boldsymbol{P}(t|t)^{\mathrm{T}}\right)$$

However, if the filter is run for long periods it may be safer to update $\boldsymbol{P}$ using the general equation

$$\boldsymbol{P}(t|t) = (\boldsymbol{I} - \boldsymbol{LC})\,\boldsymbol{P}(t|t-1)\,(\boldsymbol{I} - \boldsymbol{LC})^{\mathrm{T}} + \boldsymbol{LRL}^{\mathrm{T}}$$

which has much better numerical properties.

Updating $\boldsymbol{P}$ with suboptimal $\boldsymbol{L}$

Recall that the full equation for updating the covariance matrix,

$$\boldsymbol{P}(t|t) = (\boldsymbol{I} - \boldsymbol{LC})\,\boldsymbol{P}(t|t-1)\,(\boldsymbol{I} - \boldsymbol{LC})^{\mathrm{T}} + \boldsymbol{LRL}^{\mathrm{T}}$$

makes no assumptions about the value of $\boldsymbol{L}$. This equation gives a correct estimate of the covariance with arbitrary $\boldsymbol{L}$.

The simpler expression, $\boldsymbol{P}(t|t) = (\boldsymbol{I} - \boldsymbol{LC})\,\boldsymbol{P}(t|t-1)$, was derived using an assumption that $\boldsymbol{L}$ is optimal, so you should not use this expression for general $\boldsymbol{L}$. If you do then the Kalman filter will underestimate $\boldsymbol{P}$.

There are a number of scenarios where we might choose to use suboptimal $\boldsymbol{L}$, one of which we shall see shortly. In such circumstances take care to use the full update equation if you need an accurate $\boldsymbol{P}(t|t)$.

Precalculation of the Kalman Gain

Notice that $\boldsymbol{L}(t)$, $\boldsymbol{P}(t|t-1)$ and $\boldsymbol{P}(t|t)$ do not depend on any measurements, so if you know $\boldsymbol{P}(0|0)$ then you can calculate them for all time. This can lead to significant speed increases in real systems.

This works only if we know the statistics of the noise sources ahead of time. $\boldsymbol{Q}$ and $\boldsymbol{R}$ can even be time varying if the variations are known (that is, if needed we could use $\boldsymbol{Q}(t)$ and $\boldsymbol{R}(t)$ in our equations.)

Initialising the Kalman filter

A Kalman filter needs to be initialised with values for $\boldsymbol{x}(0|0)$ and $\boldsymbol{P}(0|0)$.

If you know precisely the initial state of the system then you set $\boldsymbol{x}(0|0)$ to the appropriate initial state and make $\boldsymbol{P}(0|0)$ very small or zero.

If you have no idea about the initial state then set $\boldsymbol{x}(0|0) = 0$ and $\boldsymbol{P}(0|0) = 1 \times 10^4 \boldsymbol{I}$, depending on the application. This will ensure that the Kalman filter will depend heavily on early measurements so that $\hat{\boldsymbol{x}}$ quickly converges to $\boldsymbol{x}$. However, don't make this too large, as large changes are the most likely to cause numerical problems with the algorithm.

Obviously if you know the variance in some initial state estimate you should enter that into the appropriate element of $\boldsymbol{P}$.

Sometimes you can't trust the model

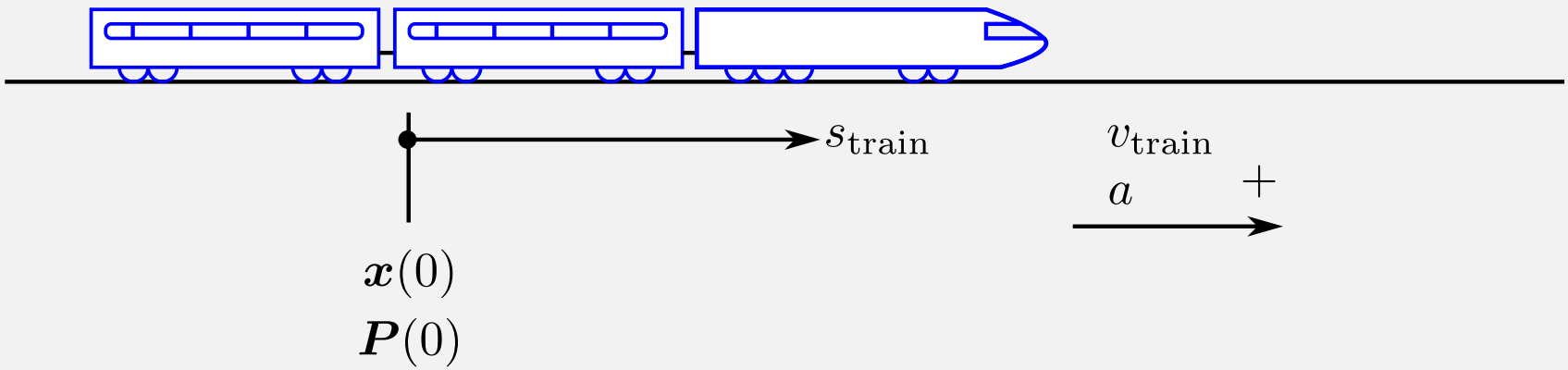
Like our previous observer designs, the Kalman filter assumes that we have a perfect model of the system.

However, we rarely have a perfect model. One way to get around this problem is to add some extra covariance to $\boldsymbol{Q}$. In effect we model the difference between the plant and our model as if were a noise source.

This approach is quite ad-hoc, but can work as it prevents the Kalman filter from showing too much faith in the model.

Train estimation

Consider a train travelling down a straight and level stretch of track. The train's throttle is set for a constant speed but the train is subject to random accelerations. We wish to estimate the train's speed and position as a function of time. The only sensor that we have is a speedometer.



We will use the state vector $\boldsymbol{x} = \begin{bmatrix} s_{\text{train}} & v_{\text{train}} \end{bmatrix}^{\mathsf{T}}$ to describe the train's motion. The random acceleration to which the train is subject will be denoted $a(t)$.

Model

We set up a discrete-time state space model as usual. In this case there is no input signal from the driver, so we omit $\boldsymbol{u}$ from the model.

$$\begin{aligned} \boldsymbol{x}(t+1) &= \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{w}(t) \\ \boldsymbol{y}(t) &= \boldsymbol{C}\boldsymbol{x}(t) + \boldsymbol{v}(t) \\ \boldsymbol{w} &\sim \mathcal{N}(0, \boldsymbol{Q}) \\ \boldsymbol{v} &\sim \mathcal{N}(0, \boldsymbol{R}) \end{aligned}$$

Here $\boldsymbol{w}$ describes the effects of the accelerations on the train and $\boldsymbol{v}$ describes the speedometer noise.

Setting up the model

We write some difference equations to describe the train's motion.

$$\begin{aligned} s(t+1) &= s(t) + v(t)t_s + \tfrac{1}{2}a(t)t_s^2 \\ v(t+1) &= v(t) + a(t)t_s \\ y(t) &= v(t) \end{aligned}$$

where t_s is the sampling time for the system.

$$\boldsymbol{x}(t+1) = \begin{bmatrix} 1 & t_s \\ 0 & 1 \end{bmatrix} \boldsymbol{x}(t) + \begin{bmatrix} \frac{t_s^2}{2} \\ t_s \end{bmatrix} a(t)$$

We choose $t_s = 0.5$ s for these simulations,

$$\begin{aligned} \boldsymbol{x}(t+1) &= \begin{bmatrix} 1 & 0.5 \\ 0 & 1 \end{bmatrix} \boldsymbol{x}(t) + \begin{bmatrix} 0.125 \\ 0.5 \end{bmatrix} a(t) \\ \boldsymbol{y}(t) &= \begin{bmatrix} 0 & 1 \end{bmatrix} \boldsymbol{x}(t) \end{aligned}$$

The plant disturbance model

Notice that our plant disturbance is not yet in a simple form. We would like an equation

$$\boldsymbol{x}(t+1) = \begin{bmatrix} 1 & 0.5 \\ 0 & 1 \end{bmatrix} \boldsymbol{x}(t) + \boldsymbol{w}(t)$$

In this case we have $\boldsymbol{w}(t) = \boldsymbol{G}a(t)$ with $\boldsymbol{G} = \begin{bmatrix} 0.125 \\ 0.5 \end{bmatrix}$, so

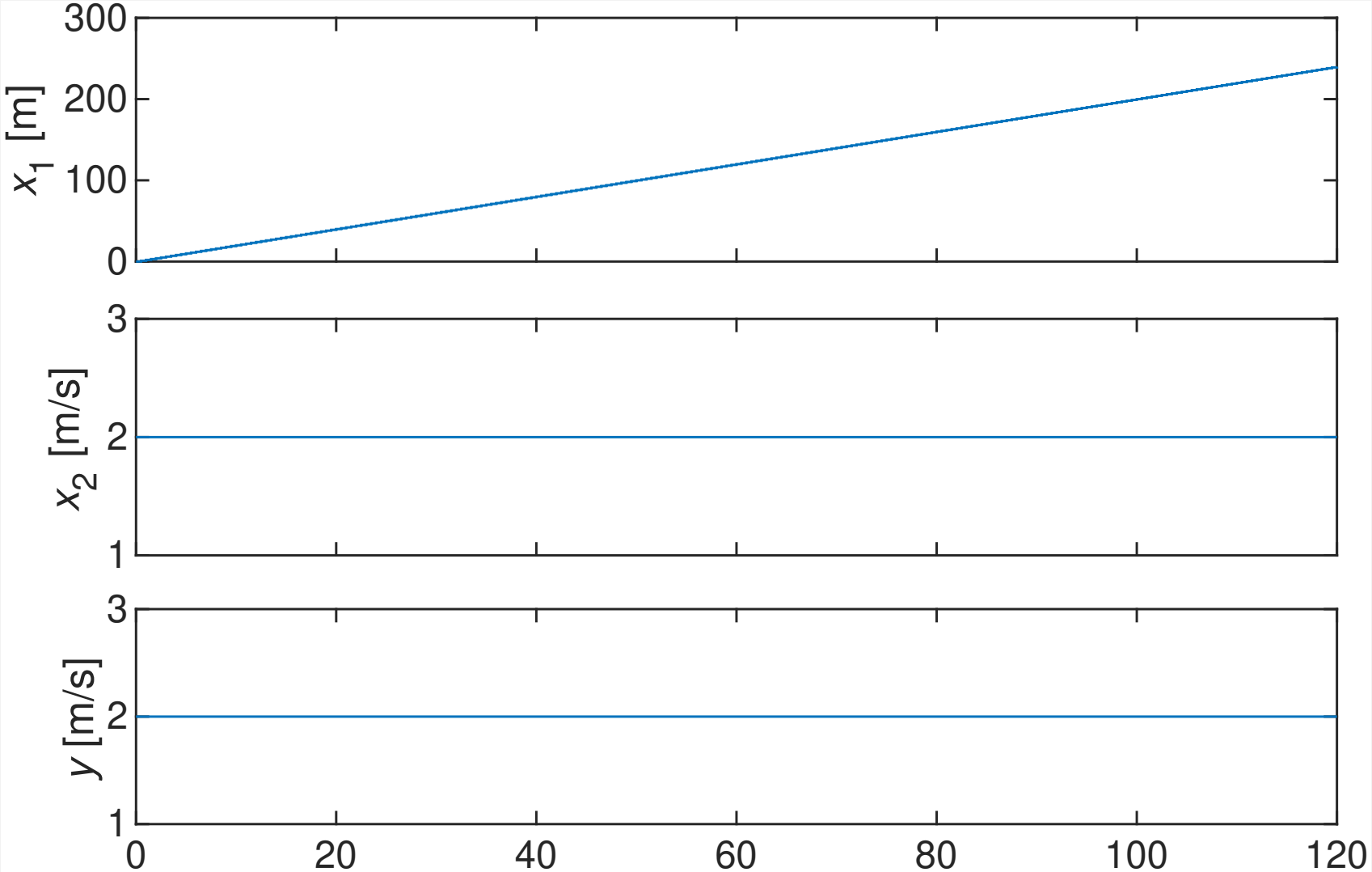
$$\boldsymbol{w}(t) = \begin{bmatrix} \frac{a(t)}{8} \\ \frac{a(t)}{2} \end{bmatrix}$$

$$\boldsymbol{Q} = \boldsymbol{G}\sigma_a^2\boldsymbol{G}^\mathsf{T} = \begin{bmatrix} \frac{1}{8} \\ \frac{1}{2} \end{bmatrix} \sigma_a^2 \begin{bmatrix} \frac{1}{8} & \frac{1}{2} \end{bmatrix} = \begin{bmatrix} \frac{\sigma_a^2}{64} & \frac{\sigma_a^2}{16} \\ \frac{\sigma_a^2}{16} & \frac{\sigma_a^2}{4} \end{bmatrix}$$

That is, we have found the disturbance in the state variables by looking directly at the physical cause of the variations.

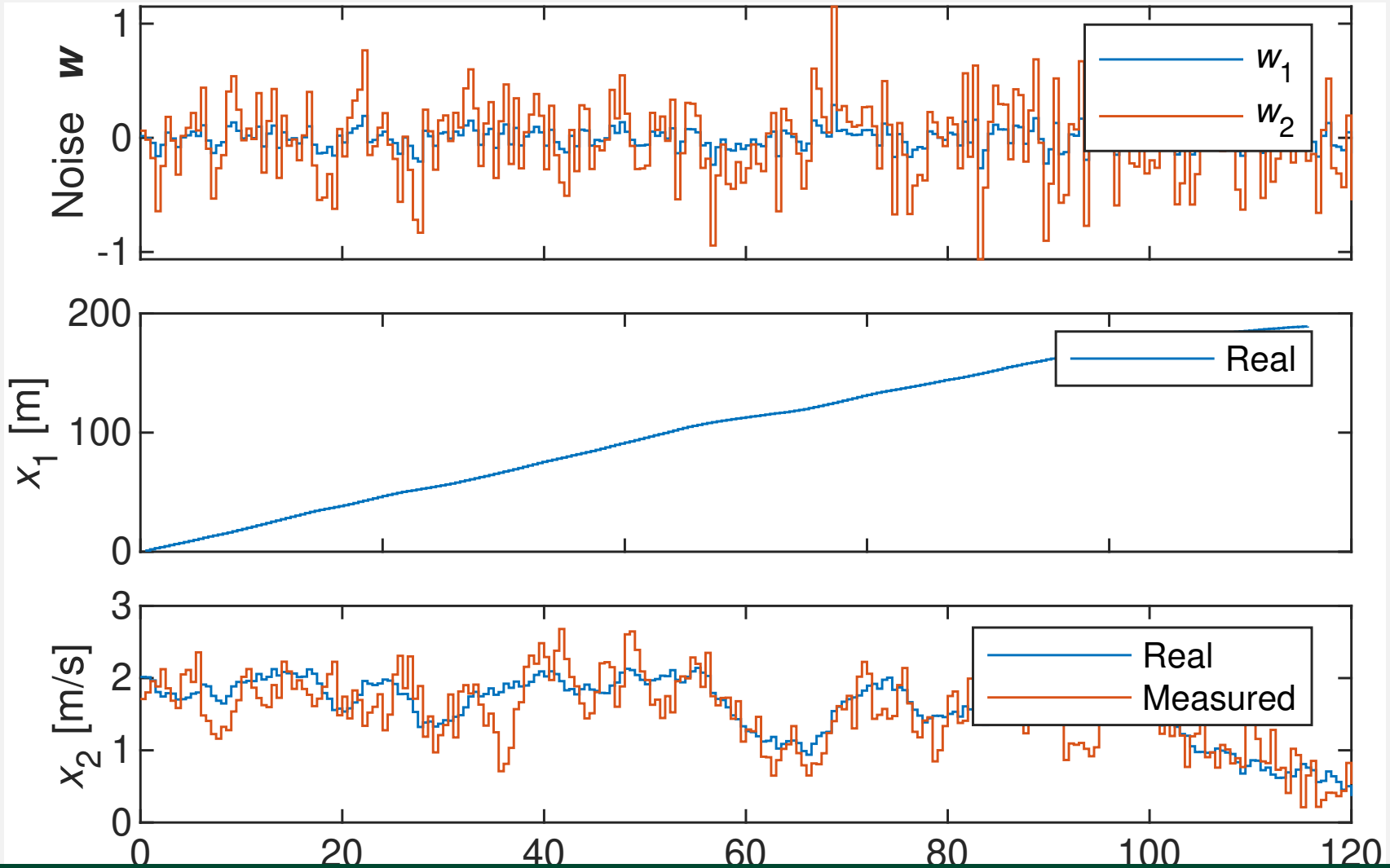
Noiseless case

To check our model, let's begin with $\boldsymbol{R} = \boldsymbol{Q} = 0$. That is, we will not disturb the system or the measurements. We set $\boldsymbol{x}(0) = \begin{bmatrix} 0 & 2 \end{bmatrix}^{\mathrm{T}}$.

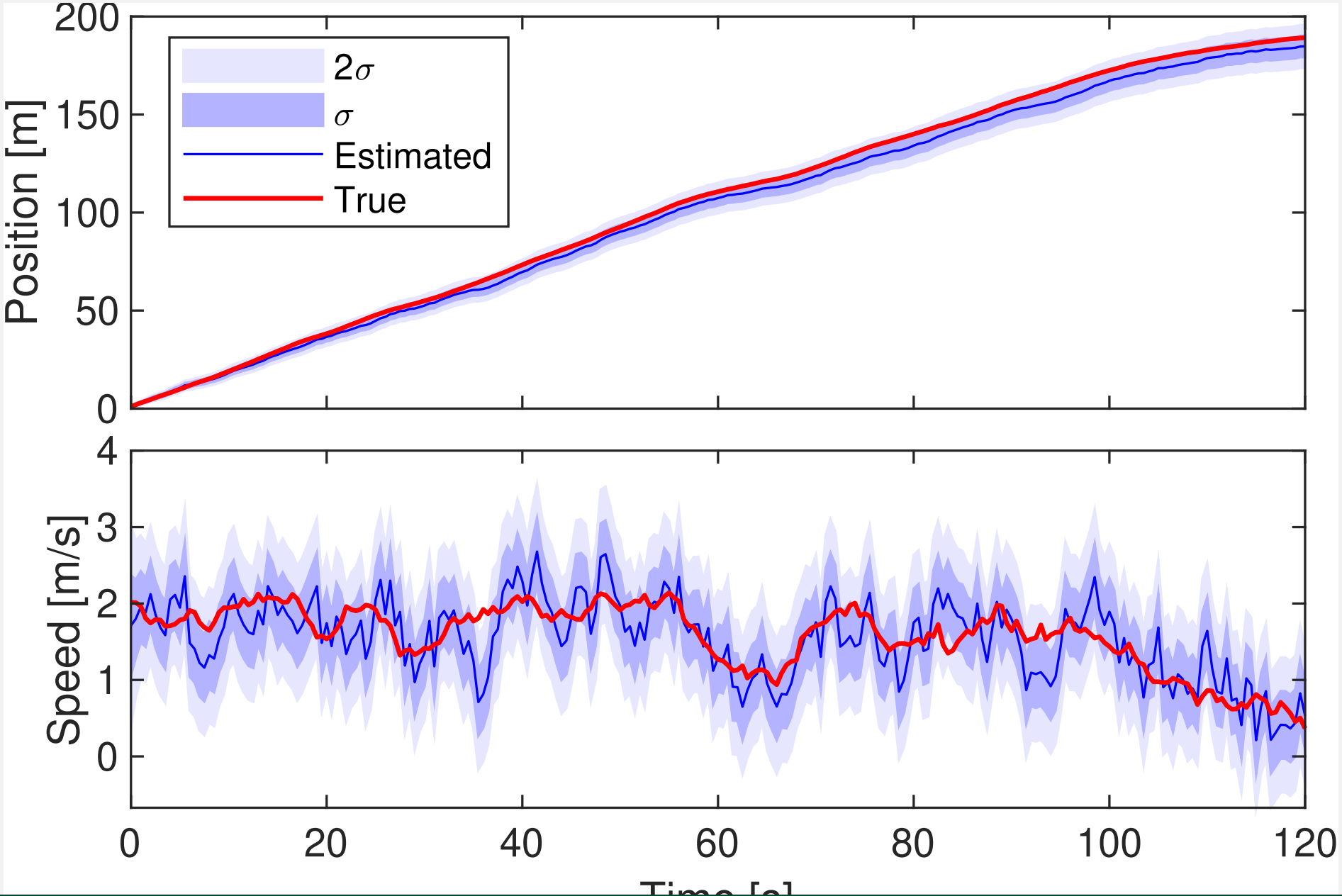


Noisy case

We now add $\boldsymbol{w}$ with $\sigma_a^2 = 0.5 \implies \boldsymbol{Q} = \begin{bmatrix} 0.0078125 & 0.03125 \\ 0.03125 & 0.125 \end{bmatrix}$, and $\boldsymbol{v}$ with $\boldsymbol{R} = 0.5$.

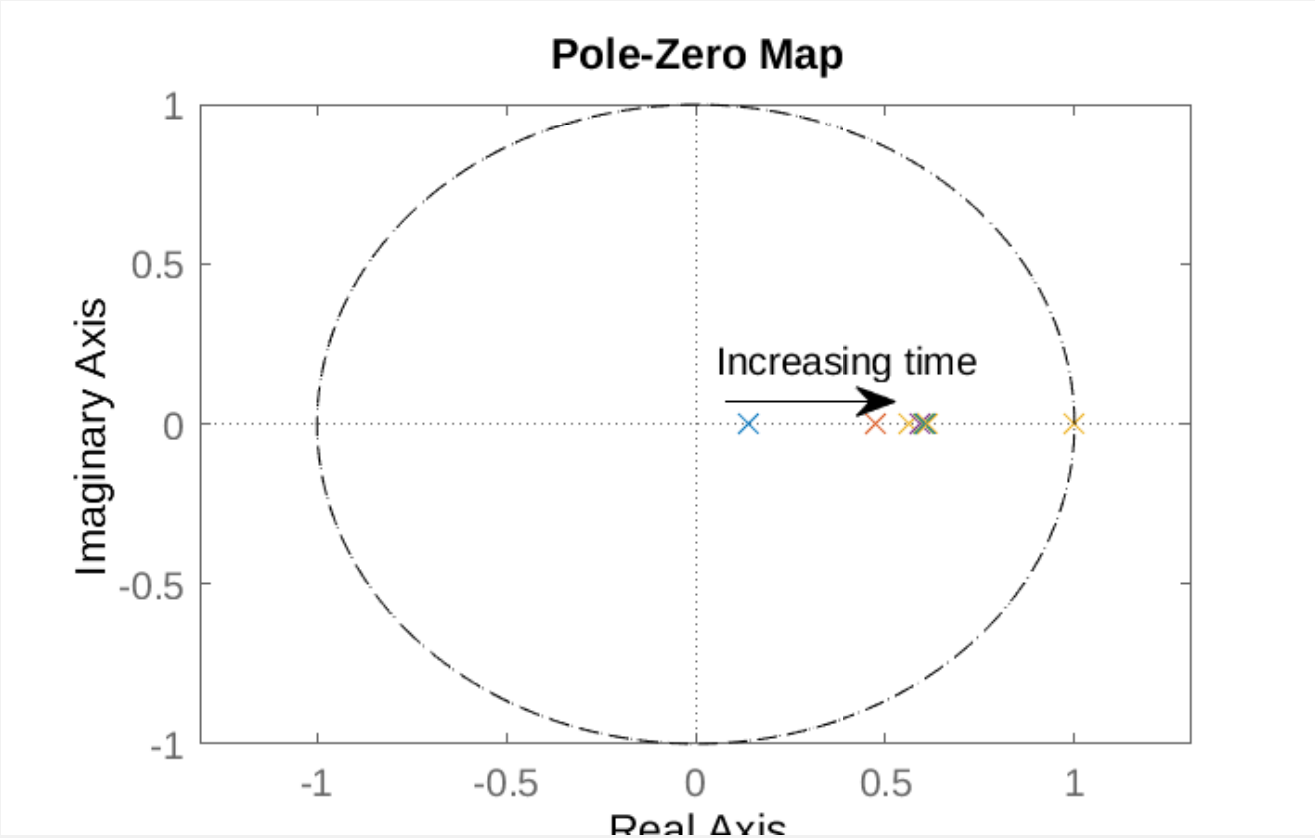


Operation of the estimator



Variation in filter poles

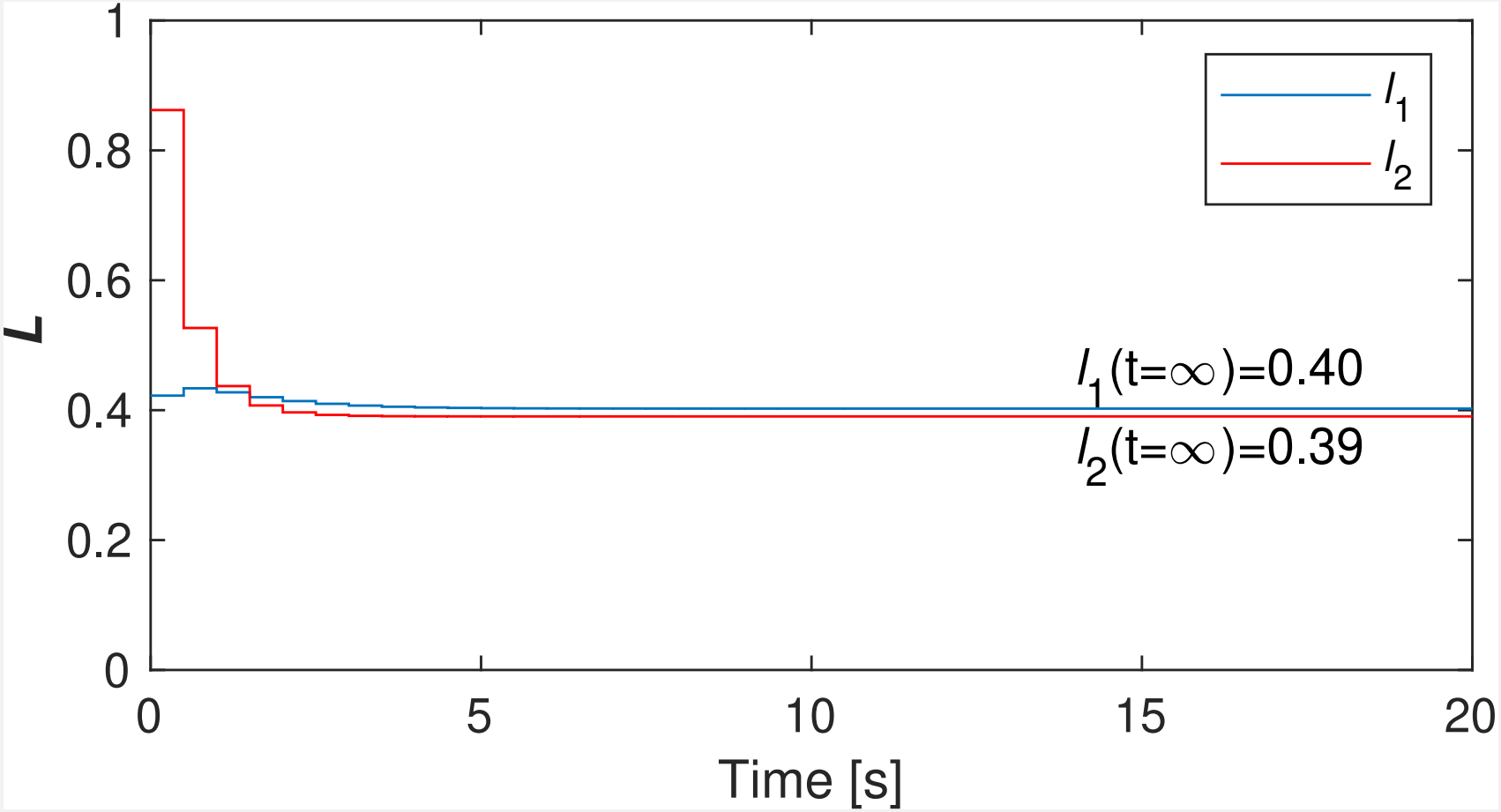
We can look at the pole location of the Kalman filter.



The filter begins with a large bandwidth (allows the measurement to pass) but becomes increasingly slow to smooth measurement noise.

Variation in Kalman gain L

Let's examine the elements of the Kalman feedback gain.



We can see that both elements converge to steady state values very quickly.

Behaviour of Kalman filter gain

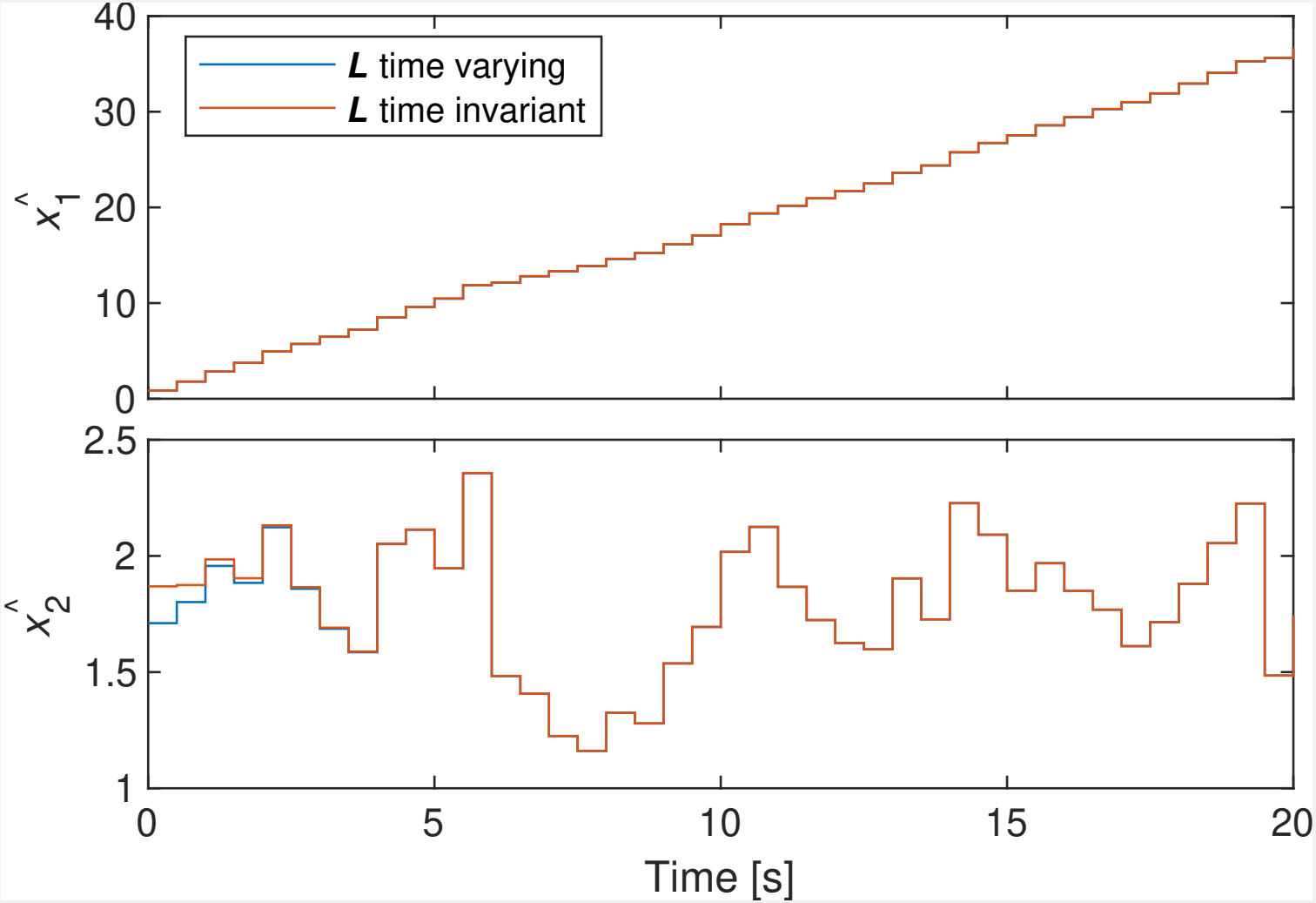
In practice the Kalman filter gain $\boldsymbol{L}$ seems to be asymptotic to some steady state value. In many cases we find that $\boldsymbol{L}$ converges to this final value quite rapidly.

In essence, if the covariances $\boldsymbol{Q}$ and $\boldsymbol{R}$ are constant then there is an optimal balance between believing the model and believing the measurements. This corresponds to a certain $\boldsymbol{L}$. The initial transient in $\boldsymbol{L}$ arises from the fact that we might initially have a large $\boldsymbol{P}$.

It seems natural to wonder how a system might perform if we just use the final value of $\boldsymbol{L}$ and avoid the complication of calculating $\boldsymbol{L}(t)$.

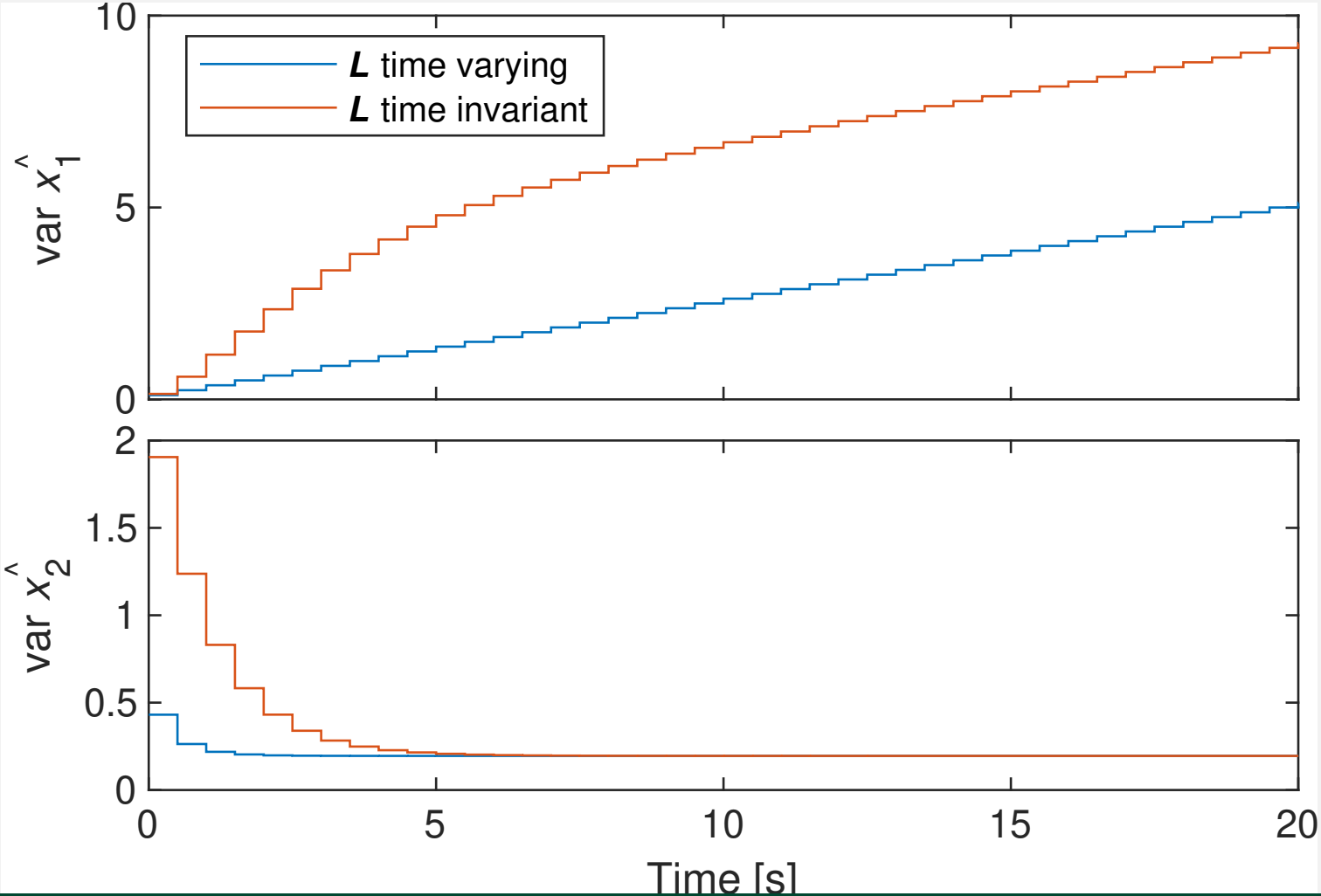
Performance with steady-state Kalman gain

We can compare the filter when we use the steady state $\boldsymbol{L}$ value.



Performance with steady-state Kalman gain

The variance in our estimates is shown in the figure. As can be seen the time varying filter reduces our uncertainty about the state more rapidly than the steady state filter.



The steady state Kalman filter

Once the initial transient has passed we will have a constant $\boldsymbol{L}$ and also a constant $\boldsymbol{P}$. We would like to find an expression for the steady state $\boldsymbol{P}$, as that would allow us to calculate the steady state $\boldsymbol{L}$ using $\boldsymbol{L} = \boldsymbol{P}\boldsymbol{C}^{\mathsf{T}}\boldsymbol{S}^{-1}$.
Let's take the expression for the prior estimate of $\boldsymbol{P}$ and substitute for the previous posterior estimate for $\boldsymbol{P}$.

$$\begin{aligned}\boldsymbol{P}(t|t-1) &= \boldsymbol{A}\boldsymbol{P}(t-1|t-1)\boldsymbol{A}^{\mathsf{T}} + \boldsymbol{Q} \\ &= \boldsymbol{A}\left[(\boldsymbol{I} - \boldsymbol{L}\boldsymbol{C})\boldsymbol{P}(t-1|t-1)\right]\boldsymbol{A}^{\mathsf{T}} + \boldsymbol{Q}\end{aligned}$$

Now, we know that $\boldsymbol{P}$ is unchanging, so we can drop the time subscripts.

$$\begin{aligned}\boldsymbol{P} &= \boldsymbol{A}\Big[(\boldsymbol{I} - \boldsymbol{L}\boldsymbol{C})\boldsymbol{P}\Big]\boldsymbol{A}^{\mathsf{T}} + \boldsymbol{Q} \\ &= \boldsymbol{A}\boldsymbol{P}\boldsymbol{A}^{\mathsf{T}} - \boldsymbol{A}\boldsymbol{L}\boldsymbol{C}\boldsymbol{P}\boldsymbol{A}^{\mathsf{T}} + \boldsymbol{Q}\end{aligned}$$

Discrete Ricatti equation

We now substitute for $\boldsymbol{L} = \boldsymbol{PC}\left(\boldsymbol{CPC}^{\mathsf{T}} + \boldsymbol{R}\right)^{-1}$.

$$\boldsymbol{P} = \boldsymbol{APA}^{\mathsf{T}} - \boldsymbol{APC}^{\mathsf{T}}\left(\boldsymbol{CPC}^{\mathsf{T}} + \boldsymbol{R}\right)^{-1}\boldsymbol{CPA}^{\mathsf{T}} + \boldsymbol{Q}$$

This is called the Discrete Algebraic Ricatti Equation (DARE), which can be solved numerically to find $\boldsymbol{P}$.

Thus, when $\boldsymbol{R}$ and $\boldsymbol{S}$ are time-invariant, we can solve the DARE to find $\boldsymbol{P}$ and hence $\boldsymbol{L}$ without needing to use the iterative predict-correct procedure. Use of this $\boldsymbol{L}$ is sub-optimal, but approaches the optimal solution.

If you wish to use Matlab's DARE solver `idare`, you will find it makes more sense when you substitute $\boldsymbol{A}^{\mathsf{T}}$ for `A` and $\boldsymbol{C}^{\mathsf{T}}$ for `B`. Matlab's syntax is tailored to the regulator problem, so we need to make the same duality motivated substitutions we saw when comparing the `place` command for observers and regulators.

Matlab for steady-state Kalman filters

The model used by Matlab is more general than ours:

$$\begin{aligned} \boldsymbol{x}(t+1) &= \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t) + \boldsymbol{G}\boldsymbol{w}(t) \\ \boldsymbol{y}(t) &= \boldsymbol{C}\boldsymbol{x}(t) + \boldsymbol{D}\boldsymbol{u}(t) + \boldsymbol{H}\boldsymbol{w}(t) + \boldsymbol{v}(t) \\ t &\in \mathbb{Z}_+ \\ \mathrm{E}\{\boldsymbol{w}\boldsymbol{w}^\mathsf{T}\} = \boldsymbol{Q}, &\quad \mathrm{E}\{\boldsymbol{v}\boldsymbol{v}^\mathsf{T}\} = \boldsymbol{R}, \quad \mathrm{E}\{\boldsymbol{w}\boldsymbol{v}^\mathsf{T}\} = \boldsymbol{N} \end{aligned}$$

For our purposes we would typically have $\boldsymbol{D} = \boldsymbol{H} = \boldsymbol{N} = 0$ and often $\boldsymbol{G} = \boldsymbol{I}$ (though $\boldsymbol{G} \neq \boldsymbol{I}$ is useful if we can model the source of the state disturbance as we did for the train example).
Matlab conveniently has a function to calculate the steady state Kalman filter. `[Kest,L,P] = kalman(SYS,Q,R,N)` with `SYS=ss(A,[B G],C,[D H])`. Be sure to include $\boldsymbol{G}$ (even if $\boldsymbol{G} = \boldsymbol{I}$). However, $\boldsymbol{H}$ is often zero and can be omitted.

$\boldsymbol{P}(t)$ in the steady state Kalman filter

The steady state Kalman filter is one in which we use the value of $\boldsymbol{L}$ that the Kalman filter approaches in the steady state. This does not imply that anything else will be at steady state. In particular, $\hat{\boldsymbol{x}}$ and $\boldsymbol{P}$ will still (in general) be time-varying.

Because the steady state and optimal Kalman filters are identical if we wait long enough, we should expect that $\boldsymbol{P}$ would eventually agree for the two filters. However, the steady-state Kalman filter is sub-optimal, so it will approach this final $\boldsymbol{P}$ value more slowly than the normal Kalman filter.

Extensions to the Kalman filter

There are many obvious extensions that can be made to a Kalman filter, particularly in its iterative form. These rely on the fact that we can change the system on the fly.

Some examples of things we could do include

- Use different update times for predictions and measurements.
- Use different sensing strategy at different times (ie. change $\boldsymbol{C}$ on the fly).
- Reject measurements that are inconsistent with the predictions (validation gating). This allows outliers to be removed, or even a faulty sensor to be detected.
- Cope with changes in the model (for example, an aircraft went from sub to supersonic, so its model changed.)
- Estimate some parameter of a model.

Sometimes you can't trust the model

Like our previous observer designs, the Kalman filter assumes that we have a perfect model of the system.

However, we rarely have a perfect model. One way to get around this problem is to add some extra covariance to $\boldsymbol{Q}$. In effect we model the difference between the plant and our model as if were a noise source.

This approach is quite ad-hoc, but can work as it forces the Kalman filter to increase the amount of feedback from the sensors, and hence reduces its dependence on the model.