CHAPTER 39

The Kalman Filter

Here is an abstraction underlying tracking by detection (Chapter 21.3). You see a frame in a video, and make an estimate of where some objects are. Typically, but not always, you do so with some form of detector. Now the next frame arrives. You must update your estimates. In some cases, you know something about how the objects move. For example, you can make a fairly reliable estimate of the acceleration of an object falling under gravity, and its velocity is going to be fairly well estimated by integrating this acceleration. Similarly, integrating velocity will get you position. These estimates aren't perfect – for example, the position and velocity of the falling object may be affected by air resistance or by wind. You will want to work this information into the estimate of position of the object.

In each of these cases, there is a clock, which might be abstract. There is some underlying state – write $\mathbf{X}_i$ for the state at tick i. This is a random variable, which is shown indicated by the capital letter. You do not know the true value of this random variable. The value changes each time the clock ticks. The way the value changes is referred to as the *dynamics*. Immediately after the value of the state changes, you receive a measurement, which is a known value of the state. You must build a representation of what you know about the state, and maintain it as measurements arrive. Doing so is known as *filtering*. In tracking, the state is the position (and perhaps velocity, acceleration or other properties) of the objects being tracked.

39.1 FILTERING: ONLINE INFERENCE

Filtering is best discussed in the language of probability. The state of the object at the ith frame (or clock tick; or incoming image) is typically written as $\mathbf{X}_i$. The measurement obtained in the ith frame is a value of a random variable $\mathbf{Y}_i$. Write $\mathbf{y}_i$ for a value taken by $\mathbf{Y}_i$. On occasion, I will write $\mathbf{Y}_i = \mathbf{y}_i$ for emphasis. For the moment, assume you can represent any probability distribution you care to, and concentrate on the question of which distributions you need to represent.

39.1.1 Independence Assumptions

Filtering is difficult without the following assumptions:

- **Only the immediate past matters:** Formally, require that

$$P(\mathbf{X}_i|\mathbf{X}_{i-1}, \text{any other conditioning information}) = P(\mathbf{X}_i|\mathbf{X}_{i-1}).$$

  This assumption – sometimes known as a Markovian assumption – hugely simplifies the design of algorithms because the only thing that affects the next state is the current state. Furthermore, it isn't terribly restrictive (next section).

- **Measurements depend only on the current state:** This means that

  $$P(\mathbf{Y}_i|\mathbf{X}_i, \mathbf{X}_0, \ldots, \mathbf{X}_k) = P(\mathbf{Y}_i|\mathbf{X}_i)$$

  for any k. Again, this isn't a particularly restrictive or controversial assumption, but it yields important simplifications.

39.1.2 Online Inference

Assume you have a model for $P(\mathbf{X}_0)$, which represents everything you know about $\mathbf{X}_0$ in the absence of measurements. You then obtain $\mathbf{y}_0$, a measurement from the initial state. Forming $P(\mathbf{X}_0|\mathbf{Y}_0 = \mathbf{y}_0)$ is a straightforward application of Bayes' rule **exercises**. The rest follows by an induction, using two steps. Assume you have $P(\mathbf{X}_{i-1}|\mathbf{Y}_0 = \mathbf{y}_0, \ldots, \mathbf{Y}_{i-1} = \mathbf{y}_{i-1})$. This distribution is often called the *prior*.

To know what the measurements you have seen – that is, $\mathbf{y}_0, \ldots, \mathbf{y}_{i-1}$ — predict for the ith frame, you need a representation of $P(\mathbf{X}_i|\mathbf{Y}_0 = \mathbf{y}_0, \ldots, \mathbf{Y}_{i-1} = \mathbf{y}_{i-1})$. Forming this representation is *prediction*, and the distribution is sometimes known as the *predictive distribution*. Notice

$$\begin{aligned} P(\mathbf{X}_i|\mathbf{y}_0, \ldots, \mathbf{y}_{i-1}) &= \int P(\mathbf{X}_i|\mathbf{X}_{i-1}, \mathbf{y}_0, \ldots \mathbf{y}_{i-1}) \\ &\quad P(\mathbf{X}_{i-1}|\mathbf{y}_0, \ldots, \mathbf{y}_{i-1}) d\mathbf{X}_{i-1} \\ &= \int P(\mathbf{X}_i|\mathbf{X}_{i-1}) P(\mathbf{X}_{i-1}|\mathbf{y}_0, \ldots, \mathbf{y}_{i-1}) d\mathbf{X}_{i-1} \end{aligned}$$

(using the first assumption) **exercises**.

Now $\mathbf{y}_i$ — the i'th measurement — arrives, and you need to compute a representation of $P(\mathbf{X}_i|\mathbf{Y}_0 = \mathbf{y}_0, \ldots, \mathbf{Y}_i = \mathbf{y}_i)$. Forming this distribution is *correction*, and the distribution is often known as the *posterior*. Notice

$$P(\mathbf{X}_i|\mathbf{y}_0, \ldots, \mathbf{y}_i) \propto P(\mathbf{y}_i|\mathbf{X}_i) P(\mathbf{X}_i|\mathbf{y}_0, \ldots, \mathbf{y}_{i-1})$$

using the independence assumptions **exercises**. This posterior now becomes the prior for the next step.

39.1.3 Linear Models

In a linear dynamical model, the state is advanced by multiplying it by some known matrix (which may depend on the frame) and then adding a normal random variable of zero mean and known covariance. In a linear measurement model, the measurement is obtained by multiplying the state by some matrix (which may depend on the frame) and then adding a normal random variable of zero mean and known covariance. There is an extremely useful interaction between normal distributions, linear dynamical models and linear measurement models. If $P(\mathbf{X}_0)$ is normal, the dynamical model is linear, and the measurement model is linear, and all noise is gaussian, then every distribution you will construct while filtering will be a normal distribution and so easy to represent (just provide the mean and covariance). This property is quite special, and is useful because representing probability distributions can be very difficult, particularly in high dimension.

Use the notation

$$\mathbf{x} \sim N(\mu, \Sigma)$$

to mean that $\mathbf{x}$ is the value of a random variable with a normal probability distribution with mean μ and covariance Σ; notice that this means that, if $\mathbf{x}$ is one-dimensional — so $x \sim N(\mu, v)$ — that its standard deviation is $\sqrt{v}$.

Remember this: *A linear model is given by*

$$\mathbf{x}_i \sim N(\mathcal{D}_i \mathbf{x}_{i-1}; \Sigma_{d_i});$$

$$\mathbf{y}_i \sim N(\mathcal{M}_i \mathbf{x}_i; \Sigma_{m_i}).$$

Notice that the covariances could be different from frame to frame as could the measurement matrix $\mathcal{M}_i$ and the dynamical matrix $\mathcal{D}_i$.

exercises. Although this model appears limited, it is in fact extremely powerful.

39.1.4 Examples of Linear Dynamics

Drifting: Assume that $\mathbf{x}$ encodes the position of a point. If $\mathbf{D}_i = Id$, then the point is moving under random walk — its new position is its old position plus some Gaussian noise term. It may seem this dynamics just applies to stationary objects, but it is quite commonly used for objects for which no better dynamic model is known

Constant velocity: Write $\mathbf{p}$ for the position and $\mathbf{v}$ for the velocity of a point moving with constant velocity. In this case, $\mathbf{p}_i = \mathbf{p}_{i-1} + (\Delta t)\mathbf{v}_{i-1}$ and $\mathbf{v}_i = \mathbf{v}_{i-1}$. Stack position and velocity into a single state vector, and obtain

$$\mathbf{x} = \left\{ \begin{array}{c} \mathbf{p} \\ \mathbf{v} \end{array} \right\}$$

and

$$\mathcal{D}_i = \left\{ \begin{array}{cc} Id & (\Delta t)Id \\ 0 & Id \end{array} \right\}.$$

Constant acceleration: Write $\mathbf{p}$ for the position, $\mathbf{v}$ for the velocity, and $\mathbf{a}$ for the acceleration of a point moving with constant acceleration. In this case, $\mathbf{p}_i = \mathbf{p}_{i-1} + (\Delta t)\mathbf{v}_{i-1}$, $\mathbf{v}_i = \mathbf{v}_{i-1} + (\Delta t)\mathbf{a}_{i-1}$, and $\mathbf{a}_i = \mathbf{a}_{i-1}$. Stack position, velocity and acceleration into a single state vector to obtain

$$\mathbf{x} = \left\{ \begin{array}{c} \mathbf{p} \\ \mathbf{v} \\ \mathbf{a} \end{array} \right\}$$

and

$$\mathcal{D}_i = \left\{ \begin{array}{ccc} Id & (\Delta t)Id & 0 \\ 0 & Id & (\Delta t)Id \\ 0 & 0 & Id \end{array} \right\}.$$

Periodic motion: Write p for the position of a point moving on a line with a periodic movement. Then p satisfies a differential equation like

$$\frac{d^2p}{dt^2} = -p.$$

This can be turned into a first order linear differential equation by writing the velocity as v and stacking position and velocity into a vector $\mathbf{u} = (p, v)$ to obtain

$$\frac{d\mathbf{u}}{dt} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \mathbf{u} = \mathcal{S}\mathbf{u}.$$

Integrate this equation with a forward Euler method with steplength is Δt, to obtain

$$\mathbf{u}_i = \begin{pmatrix} 1 & \Delta t \\ -\Delta t & 1 \end{pmatrix} \mathbf{u}_{i-1}.$$

This method works for points on the plane, in 3D, and so on, as well as for other integrators **exercises**.

Higher order models: Assume the point $\mathbf{p}$ moves with a model $\mathbf{p}_i = \sum_{u=1}^{u=k} a_u \mathbf{p}_{i-u}$ + gaussian noise for some constants a_u. It may appear that this does not meet the requirement that $P(\mathbf{x}_i|\mathbf{x}_1, \ldots, \mathbf{x}_{i-1}) = P(\mathbf{x}_i|\mathbf{x}_{i-1})$, but it does. Write

$$\mathbf{x}_{i-1} = \begin{bmatrix} \mathbf{p}_{i-1} \\ \mathbf{p}_{i-2} \\ \cdots \\ \mathbf{p}_{i-k} \end{bmatrix}$$

then

$$\mathbf{x}_i = \begin{bmatrix} a_1 & \ldots & \ldots & a_k \\ 1 & 0 & 0 & 0 \\ & \ldots & \ldots & \\ 0 & 0 & 1 & 0 \end{bmatrix} \mathbf{x}_{i-1}$$

You can fit other models in this section into this framework **exercises**.

You don't need to measure every aspect of the state of a moving point at every step. For example, assume that the point is in 3D: Now if $\mathcal{M}_{3k} = (0, 0, 1)$, $\mathcal{M}_{3k+1} = (0, 1, 0)$, and $\mathcal{M}_{3k+2} = (1, 0, 0)$, then at every third frame you measure, respectively, the z, y, or x position of the point. You can still expect to estimate the state of the point. Similarly, if you observe the position of a point whose state contains position, velocity and acceleration, you can reconstruct velocity and acceleration from those observations. In each case, the state is *observable*.

However, there are some cases where measurements do not constrain the state. These tend not to be common in vision problems. The exercises explore this point in a little more detail.

39.1.5 Kalman Filtering

If $P(\mathbf{X}_0)$ is normal, the measurement and dynamic models are linear, then

$$P(\mathbf{X}_i|\mathbf{y}_1, \ldots, \mathbf{y}_{i-1}) \text{ and } P(\mathbf{X}_i|\mathbf{y}_1, \ldots, \mathbf{y}_i)$$

are normal, too. This means that they are relatively easy to represent — all you need to do is maintain representations of the mean and the covariance for the prediction and correction phase. The procedure for maintaining these representations is known as the *Kalman filter*.

I will use the following notation for the Kalman filter. Write

$\overline{\mathbf{x}}_i^-$ for the mean of $P(X_i|y_0, \ldots, y_{i-1})$
Σ_i^- for the covariance of $P(X_i|y_0, \ldots, y_{i-1})$
and
$\overline{\mathbf{x}}_i^+$ for the mean of $P(X_i|y_0, \ldots, y_i)$
Σ_i^+ for the covariance of $P(X_i|y_0, \ldots, y_i)$

The superscripts on terms suggest that they represent our belief about X_i immediately before and immediately after the ith measurement arrives. Recall that the covariance of a 1D normal distribution is its variance, and I will write, for example, $(\sigma_i^-)^2$ for the variance of the predictive distribution in 1D (and so on).

39.1.6 The Kalman Filter for a 1D State Vector

The model is

$$x_i \sim N(d_i x_{i-1}, \sigma_{d_i}^2);$$

$$y_i \sim N(m_i x_i, \sigma_{m_i}^2).$$

Think of the m_i in the measurement model as a change of unit from the state to the measurement. Prediction is straightforward. $P(X_{i-1}|y_0, \ldots, y_{i-1})$ is normal and has

$$\text{mean } \overline{\mathbf{x}}_{i-1}^- \text{ and variance } \sigma_{i-1}^-)^2).$$

Further, $x_i = d_i x_{i-1} + \xi$ (where $\xi \sim N(0, (\sigma_{n,i})^2)$). Now

$$d_i x_{i-1} \sim N(d_i \overline{\mathbf{x}}_{i-1}^-, d^2(\sigma_{i-1}^-)^2)$$

(**exercises**), which means $P(X_i|y_0, \ldots, y_{i-1})$ is normal and has

$$\text{mean } d_i \overline{\mathbf{x}}_{i-1}^+ \text{ and variance } d^2(\sigma_{i-1}^+)^2 + \sigma_{n,i}^2$$

(**exercises**). This means

$$\overline{\mathbf{x}}_i^- = d_i \overline{\mathbf{x}}_{i-1}^+ \text{ and } (\sigma_i^-)^2 = d^2(\sigma_{i-1}^+)^2 + \sigma_{n,i}^2.$$

Updating to take the measurement into account is also straightforward. From Bayes' rule,

$$\begin{aligned}\log P(X_i|y_0,\ldots y_i) &= \log P(X_i|y_0,\ldots,y_{i-1})+\log P(y_i|X_i)+K\\ &= -\frac{(X_i-\overline{\mathbf{x}}_i^-)^2}{2(\sigma_i^-)^2}-\frac{(y_i-m_iX_i)^2}{2(\sigma_{m,i})^2}+K\\ &= -\frac{X_i^2}{2}\left[\frac{m_i^2(\sigma_i^-)^2+(\sigma_{m,i})^2}{(\sigma_{m,i})^2(\sigma_i^-)^2}\right]+\\ &\quad X_i\left[\frac{m_iy_i(\sigma_i^-)^2+\overline{\mathbf{x}}_i^-(\sigma_{m,i})^2}{(\sigma_{m,i})^2(\sigma_i^-)^2}\right]+K''\end{aligned}$$

Now pattern match to the expression for a normal distribution to recover

$$\overline{\mathbf{x}}_i^+=\left[\frac{m_iy_i(\sigma_i^-)^2+\overline{\mathbf{x}}_i^-(\sigma_{m,i})^2}{m_i^2(\sigma_i^-)^2+(\sigma_{m,i})^2}\right]\text{ and }(\sigma_i^+)^2=\left[\frac{(\sigma_{m,i})^2(\sigma_i^-)^2}{m_i^2(\sigma_i^-)^2+(\sigma_{m,i})^2}.\right]$$

You can rearrange the expression for $\overline{\mathbf{x}}_i^+$ to obtain the highly suggestive form

$$x_i^+=\overline{x}_i^-+k_i\left(y_i-m_i\overline{x}_i^-\right)$$

where

$$k_i=\frac{m_i(\sigma_i^-)^2}{m_i^2(\sigma_i^-)^2+(\sigma_{m,i})^2}.$$

So you correct the mean with a weighted difference between the measurement and the value predicting by the predicted mean. This should seem natural to you – a bigger difference should require a larger correction. You should check that if the measurement is wholly unreliable – so $\sigma_{m,i} >> \sigma_i^-$ – the weight will be very small. Further, check that if $\sigma_i^- >> \sigma_{m,i}$, the posterior mean will be very largely the measurement corrected for units.

Procedure: 39.1 *The 1D Kalman filter updates estimates of the mean and covariance of the various distributions encountered while tracking a one-dimensional state variable using the given dynamic model.*

Dynamic Model:

$$x_i \sim N(d_i x_{i-1}, \sigma_{d_i})$$

$$y_i \sim N(m, i x_i, \sigma_{m,i})$$

Start Assumptions: $\overline{x}_0^-$ and σ_0^- are known. From these, using y_0, construct:

$$k_0 = \frac{m_0(\sigma_i^-)^2}{m_i^2(\sigma_i^-)^2 + (\sigma_{m,i})^2}$$
$$x_i^+ = \overline{x}_i^- + k_0 \left(y_0 - m_0 \overline{x}_0^-\right)$$

$$(\sigma_i^+)^2 = \left(\frac{\sigma_{m,i}^2(\sigma_i^-)^2}{(\sigma_{m,i}^2 + m_0^2(\sigma_i^-)^2)}\right).$$

Update Equations: Prediction

$$\overline{x}_i^- = d_i \overline{x}_{i-1}^+$$

$$(\sigma_i^-)^2 = \sigma_{d_i}^2 + (d_i \sigma_{i-1}^+)^2$$

Update Equations: Correction

$$k_i = \frac{m_i(\sigma_i^-)^2}{m_i^2(\sigma_i^-)^2 + (\sigma_{m,i})^2}$$
$$x_i^+ = \overline{x}_i^- + k_i \left(y_i - m_i \overline{x}_i^-\right)$$

$$(\sigma_i^+)^2 = \left(\frac{\sigma_{m,i}^2(\sigma_i^-)^2}{(\sigma_{m,i}^2 + m_i^2(\sigma_i^-)^2)}\right)$$

39.1.7 The Kalman Update Equations for a General State Vector

I obtained a 1D filter without having to do any integration using the properties of normal distributions. This approach works for a state vector of arbitrary dimension, but the process is a good deal more elaborate than that shown in Section 39.1.6.

I omit the necessary orgy of notation — it's a tough but straightforward exercise for those who really care — and simply give the result in Algorithm **??**. Notice the form: the prediction is corrected by a linear function of the difference between measurement and predicted measurement.

Procedure: 39.2 *The Kalman filter updates estimates of the mean and covariance of the various distributions encountered while tracking a state variable of some fixed dimension using the given dynamic model.*

Dynamic Model:

$$\mathbf{x}_i \sim N(\mathcal{D}_i \mathbf{x}_{i-1}, \Sigma_{d,i})$$

$$\mathbf{y}_i \sim N(\mathcal{M}_i \mathbf{x}_i, \Sigma_{m,i})$$

Start Assumptions: $\overline{\mathbf{x}}_0^-$ and Σ_0^- are known. From these, using $\mathbf{y}_0$, construct:

$$\mathcal{K}_0 = \Sigma_0^- \mathcal{M}_0^T \left[\mathcal{M}_0 \Sigma_0^- \mathcal{M}_0^T + \Sigma_{m,i}\right]^{-1}$$

$$\overline{\mathbf{x}}_0^+ = \overline{\mathbf{x}}_0^- + \mathcal{K}_0 \left[\mathbf{y}_0 - \mathcal{M}_0 \overline{\mathbf{x}}_0^-\right]$$

$$\Sigma_0^+ = \left[Id - \mathcal{K}_0 \mathcal{M}_0\right] \Sigma_0^-.$$

Update Equations: Prediction

$$\overline{\mathbf{x}}_i^- = \mathcal{D}_i \overline{\mathbf{x}}_{i-1}^+$$

$$\Sigma_i^- = \Sigma_{d,i} + \mathcal{D}_i \Sigma_{i-1}^+ \mathcal{D}_i^T$$

Update Equations: Correction

$$\mathcal{K}_i = \Sigma_i^- \mathcal{M}_i^T \left[\mathcal{M}_i \Sigma_i^- \mathcal{M}_i^T + \Sigma_{m,i}\right]^{-1}$$

$$\overline{\mathbf{x}}_i^+ = \overline{\mathbf{x}}_i^- + \mathcal{K}_i \left[\mathbf{y}_i - \mathcal{M}_i \overline{\mathbf{x}}_i^-\right]$$

$$\Sigma_i^+ = \left[Id - \mathcal{K}_i \mathcal{M}_i\right] \Sigma_i^-$$

The matrix $\mathcal{K}_i$ is known as *the Kalman gain matrix.*

Remember this: *When the prior is normal, dynamics are linear with Gaussian noise and measurement is linear with Gaussian noise, both predictive distribution and posterior are normal. The Kalman filter maintains the mean and covariance of these normal distributions.*

39.1.8 Examples

Figures **??**, 39.2 and 39.4 show examples of a Kalman filter estimating state for a 2D state space under various conditions. The state is position and velocity of a point moving on a line. The initial state is 0, 0.5 – so at the origin, moving in the positive direction at velocity 0.5. The clock ticks every 20'th of a second, and there are 20 ticks in total. The **blue** circles give the true state, and the **blue** arrows indicate which state follows which. The filter is initialized with a gaussian with mean at the origin, and isotropic covariance with covariance 0.5 in each direction. The mean of each distribution is shown with a marker, and the covariance is indicated with an ellipse which contains 90% of the weight of the distribution. **Red** shows predictive distributions, and **green** shows the posterior. The **inset** uses a scale that shows the prior, which has relatively large covariance.

Figure 39.4 shows the filter operating where the measurement is sum of position and velocity. The measurements are somewhat inaccurate. The velocity is now quite noisy (the states move up and down). Notice the prior and posterior estimates have quite variance in the position - velocity direction, and this remains quite large. This direction is orthogonal to the measurement direction, so measurements acquire no information about it – the only control on the estimate of this direction is dynamical information. You might expect that the state is not observable in this case, but this is wrong. The variance in the direction orthogonal to the measurement direction does go down slowly. The exercises explore this point.**exercises**

Figure 39.4 shows the filter operating with an unusual measurement model. In this case, at every fifth tick of the clock, the measurement supplies position (these states are **yellow**). All other measurements are of velocity. The velocity is now quite noisy (the states move up and down), but the filter produces rather good posterior estimates. At A and B, the predictive distribution is poor, because the previous measurement was of position and the velocity update is noisy. The posterior is much better, because the filter received a velocity measurement **exercises**. At C1, the prediction of velocity is fairly poor, but the measurement corrects it to a better posterior; this is propagated to C2, where the measurement is one of position, and so the error in velocity does not improve much. Finally, at C3 the prediction is a noisier version of that at C2, but the measurement provides a velocity estimate and the posterior corrects it significantly.

39.2 THE EXTENDED KALMAN FILTER

A car moving on the plane has interesting, but resolutely non-linear, dynamics. The difficulty is caused by the car's steering. The car can only move in the direction its front wheels are pointed. Figure **??** shows notation for this case. The position of the car's centre is (x, y); the car's orientation is θ; and the orientation of the steering relative to the orientation of the car is ϕ. Encoding the position and orientation of the car requires three variables, but you can represent the movement of the car using only two variables: the speed at which it moves along its axis (s) and the rate at which its steering angle changes (r). The "missing" term occurs because the car can't translate sideways (look at Figure **??** if you're uncertain).

In this notation, you can derive (or look up, at `https://msl.cs.uiuc.edu/planning/node658.html`) the model

$$\frac{d}{dt}\left[\begin{array}{c} x \\ y \\ \theta \end{array}\right] = \left[\begin{array}{c} s\cos\theta \\ s\sin\theta \\ \frac{s}{L}\tan r \end{array}\right].$$

Now assume this car has a measurement device attached to it. This device – which could be LiDAR or stereo – measures the location of a set of beacons relative to the car. Assume the i'th beacon is at (u_i, v_i) (and, for the moment, there is no noise). Then the measurement this device produces for the i'th beacon is

$$\mathcal{R}_{-\theta}\left[\begin{array}{c} (u_i - x) \\ (v_i - y) \end{array}\right].$$

The car should build (a) a map of the world and (b) a representation of where it is in the map. It should do so online – every time it moves, it should update the map and its pose to the best available estimate. This is clearly a filtering exercise, but the Kalman filter won't work because neither dynamics and measurement are linear. The extended Kalman filter deals with this by linearizing dynamics and measurement about the current operating point. It is often, but not always, successful.

39.2.1 Linearizing a Non-Linear Function of a Random Variable

You have a non-linear function $\mathbf{h}$ of a random variable $\mathbf{x}$. The random variable is normal, so

$$\mathbf{x} \sim N(\mu_{\mathbf{x}}, \Sigma_{\mathbf{x}}).$$

The function is d-dimensional, and accepts a second argument $\mathbf{n}$. Here $\mathbf{n} \sim N(\mathbf{0}, \Sigma_{\mathbf{n},i})$. Write

$$\mathbf{u} = \mathbf{h}(\mathbf{x}, \mathbf{n}).$$

Write $\mathcal{J}_{\mathbf{h},\mathbf{x}}$ for the matrix of first partial derivatives of $\mathbf{h}$ with respect to the first argument, and $\mathcal{J}_{\mathbf{h},\mathbf{n}}$ for the matrix of first partial derivatives of $\mathbf{h}$ with respect to the second argument. Now linearize to find

$$\mathbf{u} \sim N(\mathbf{h}(\mu_{\mathbf{x}}, \mathbf{0}), \mathcal{J}_{\mathbf{h},\mathbf{x}}\Sigma_{\mathbf{x}} + \mathcal{J}_{\mathbf{h},\mathbf{x}}^T + \mathcal{J}_{\mathbf{h},\mathbf{n}}\Sigma_{\mathbf{n}}\mathcal{J}_{\mathbf{h},\mathbf{n}}^T).$$

(**exercises**)

39.2.2 Linearizing State Update and Measurement

Linearizing the state update and the measurement is now a question of substituting terms. The state is updated by $\mathbf{f}$. Write

$$\mathbf{x}_{i+1} = \mathbf{f}(\mathbf{x}_i, \mathbf{n}).$$

Here $\mathbf{n}$ is noise, $\mathbf{n} \sim N(\mathbf{0}, \Sigma_{s,i})$, and $\mathbf{n}$ has the same dimension as the state. Model $\mathbf{x}_i$ using

$$\mathbf{x}_i \sim N(\overline{\mathbf{x}}_i, \Sigma_{\mathbf{x}_i})$$

and linearize to get

$$\mathbf{x}_{i+1} \sim N(\mathbf{f}(\overline{\mathbf{x}}_i), \mathcal{J}_{\mathbf{f},\mathbf{x}} \Sigma_{\mathbf{x}_i} + \mathcal{J}_{\mathbf{f},\mathbf{x}}^T + \mathcal{J}_{\mathbf{f},\mathbf{n}} \Sigma_{s,i} \mathcal{J}_{\mathbf{f},\mathbf{n}}^T)$$

exercises. Here $\mathcal{J}_{\mathbf{f},\mathbf{x}}$ is analogous to $\mathcal{D}_i$. You expect that the value of $\mathbf{x}_{i-1}$ is close to $\overline{\mathbf{x}}_{i-1}^- + \delta$ from the properties of a mean. If $\mathbf{x}_{i-1} = \overline{\mathbf{x}}_{i-1}^- + \delta$, then $\mathbf{x}_{i+1} \approx \overline{\mathbf{x}}_{i-1}^- + \mathcal{J}_{\mathbf{f},\mathbf{x}} \delta$. Similarly, $\mathcal{J}_{\mathbf{f},\mathbf{n}} \Sigma_{s,i} \mathcal{J}_{\mathbf{f},\mathbf{n}}^T$ is analogous to $\Sigma_{d,i}$.

Similarly, the observation is obtained with $\mathbf{g}$. Write

$$\mathbf{y}_i = \mathbf{g}(\mathbf{x}_i, \mathbf{n}).$$

Here $\mathbf{n}$ is noise, $\mathbf{n} \sim N(\mathbf{0}, \Sigma_{m,i})$, and $\mathbf{n}$ has the same dimension as the state. Model $\mathbf{x}_i$ using

$$\mathbf{x}_i \sim N(\overline{\mathbf{x}}_i, \Sigma_{\mathbf{x}_i})$$

and linearize to get

$$\mathbf{y}_i \sim N(\mathbf{g}(\overline{\mathbf{x}}_i, \mathbf{0}), \mathcal{J}_{\mathbf{g},\mathbf{x}} \Sigma_{\mathbf{x}_i} + \mathcal{J}_{\mathbf{g},\mathbf{x}}^T + \mathcal{J}_{\mathbf{g},\mathbf{n}} \Sigma_{\mathbf{m},\mathbf{i}} \mathcal{J}_{\mathbf{g},\mathbf{n}}^T)$$

exercises. Here $\mathcal{J}_{\mathbf{g},\mathbf{x}}$ is analogous to the matrix $\mathcal{M}_i$ in the case of linear measurements, and $\mathcal{J}_{\mathbf{g},\mathbf{n}} \Sigma_{\mathbf{m},\mathbf{i}} \mathcal{J}_{\mathbf{g},\mathbf{n}}^T$ is analogous to $\Sigma_{m,i}$.

39.2.3 The Extended Kalman Filter

The steps in the extended kalman filter are easily obtained by analogy. Assume $\overline{\mathbf{x}}_{i-1}^+$ and Σ_{i-1}^+ are known. To predict, apply the linearized dynamics to obtain

$$\overline{\mathbf{x}}_i^- = \mathbf{f}(\overline{\mathbf{x}}_{i-1}^+, \mathbf{0}) \text{ and } \Sigma_i^- = \mathcal{J}_{\mathbf{f},\mathbf{x}} \Sigma_{i-1}^+ \mathcal{J}_{\mathbf{f},\mathbf{x}}^T + \mathcal{J}_{\mathbf{f},\mathbf{n}} \Sigma_{s,i} \mathcal{J}_{\mathbf{f},\mathbf{n}}^T$$

exercises. The mean of the predicted measurement will be $\mathbf{g}(\overline{\mathbf{x}}_i^-, \mathbf{0})$, so analogy suggests correctly that

$$\overline{\mathbf{x}}_i^+ = \overline{\mathbf{x}}_i^- + \mathcal{K}_i \left[\mathbf{y}_i - \mathbf{g}(\overline{\mathbf{x}}_i^-, \mathbf{0}) \right]$$

and

$$\Sigma_i^+ = \left[Id - \mathcal{K}_i \mathcal{J}_{\mathbf{f},\mathbf{x}} \right] \Sigma_i^-$$

(where the matrix of derivatives is evaluated at $\overline{\mathbf{x}}_i^-$). The important question is the form of the Kalman gain matrix. Using the analogies above yields the correct form, in the box.

Procedure: 39.3 *The Kalman filter updates estimates of the mean and covariance of the various distributions encountered while tracking a state variable of some fixed dimension using the given dynamic model.*

Dynamic Model:

$$\begin{aligned} \mathbf{x}_i &= \mathbf{f}(\mathbf{x}_{i-1}, \mathbf{n}_{d,i}) \\ \text{where } \mathbf{n}_{d,i} &\sim N(\mathbf{0}, \Sigma_{d,i}) \end{aligned}$$

$$\begin{aligned} \mathbf{y}_i &= \mathbf{g}(\mathbf{x}_i, \mathbf{n}_{m,i}) \\ \text{where } \mathbf{n}_{m,i} &\sim N(\mathbf{0}, \Sigma_{m,i}) \end{aligned}$$

Start Assumptions: $\overline{\mathbf{x}}_0^-$ and Σ_0^- are known
Update Equations: Prediction

$$\overline{\mathbf{x}}_i^- = \mathbf{f}(\overline{\mathbf{x}}_{i-1}^+, \mathbf{0})$$

$$\Sigma_i^- = \mathcal{J}_{\mathbf{f},\mathbf{x}} \Sigma_{i-1}^+ \mathcal{J}_{\mathbf{f},\mathbf{x}}^T + \mathcal{J}_{\mathbf{f},\mathbf{n}} \Sigma_{s,i} \mathcal{J}_{\mathbf{f},\mathbf{n}}^T$$

Update Equations: Correction

$$\mathcal{K}_i = \Sigma_i^- \mathcal{J}_{\mathbf{g},\mathbf{x}}^T \left[\mathcal{J}_{\mathbf{g},\mathbf{x}} \Sigma_i^- \mathcal{J}_{\mathbf{g},\mathbf{x}}^T + \mathcal{J}_{\mathbf{g},\mathbf{n}} \Sigma_{m,i} \mathcal{J}_{\mathbf{g},\mathbf{n}}^T \right]^{-1}$$

$$\overline{\mathbf{x}}_i^+ = \overline{\mathbf{x}}_i^- + \mathcal{K}_i \left[\mathbf{y}_i - \mathbf{g}(\overline{\mathbf{x}}_i^-, \mathbf{0}) \right]$$

$$\Sigma_i^+ = \left[Id - \mathcal{K}_i \mathcal{J}_{\mathbf{f},\mathbf{x}} \right] \Sigma_i^-$$

The extended Kalman filter is often extremely useful (Chapter 21.3), but should be adopted with caution. If the linearization of either **f** or **g** is unreliable at any state you encounter, there is a good chance it will behave badly. Examples are fairly easy to construct **exercises**.

Remember this: *The extended Kalman filter is often useful when either dynamics or measurement are non-linear, but can behave badly.*

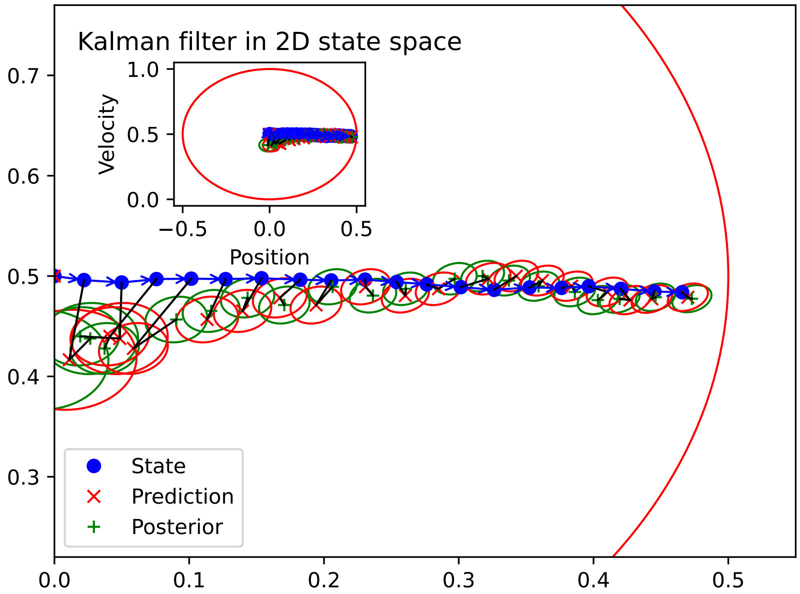

FIGURE 39.1: *A Kalman filter with relatively little noise in the dynamics and rather inaccurate measurements will fairly quickly estimate state accurately. The state is position and velocity of a point moving on a line. The initial state is 0, 0.5 – so at the origin, moving in the positive direction at velocity 0.5. The clock ticks every 20'th of a second, and there are 20 ticks in total. The* **blue** *circles give the true state, and the* **blue** *arrows indicate which state follows which. The filter is initialized with a gaussian with mean at the origin, and isotropic covariance with covariance 0.5 in each direction. The* **inset** *uses a scale that shows the prior, which has relatively large covariance. The mean of each distribution is shown with a marker, and the covariance is indicated with an ellipse which contains 90% of the weight of the distribution.* **Red** *shows predictive distributions, and* **green** *shows the posterior. The measurement is position and velocity, with substantial noise. You can tell the dynamics have very little noise because the velocity (vertical variable) is very largely constant and the positions (horizontal variable) are evenly spaced.*

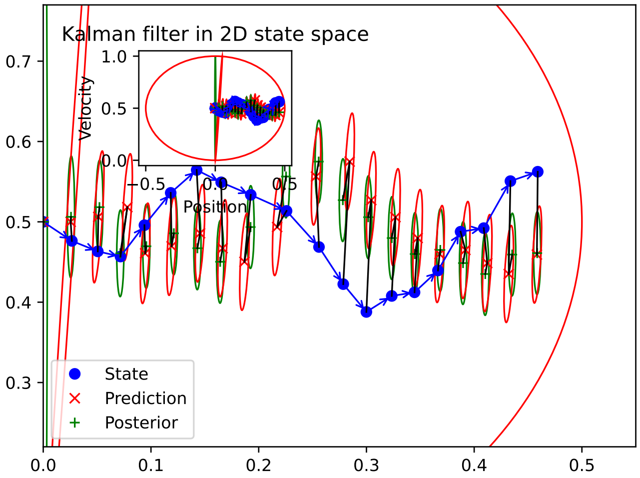

FIGURE 39.2: *A Kalman filter uses measurements effectively. Here only position is measured. Notice how the posterior covariance is very narrow in position – the measurement is accurate – and broader in velocity. Notice also that the posterior covariance contracts in velocity, because the measurement supplies some information about velocity (think about the derivative). There is substantial noise in the dynamics but position measurements are quite accurate. The state is position and velocity of a point moving on a line. The initial state is 0, 0.5 – so at the origin, moving in the positive direction at velocity 0.5. The clock ticks every 20'th of a second, and there are 20 ticks in total. The* **blue** *circles give the true state, and the* **blue** *arrows indicate which state follows which. The filter is initialized with a gaussian with mean at the origin, and isotropic covariance with covariance 0.5 in each direction. The* **inset** *uses a scale that shows the prior, which has relatively large covariance. The mean of each distribution is shown with a marker, and the covariance is indicated with an ellipse which contains 90% of the weight of the distribution.* **Red** *shows predictive distributions, and* **green** *shows the posterior. The measurement is position and velocity, with substantial noise. You can tell the dynamics have substantial noise because the velocity (vertical variable) changes significantly. There is not much noise in position (horizontal variable).*

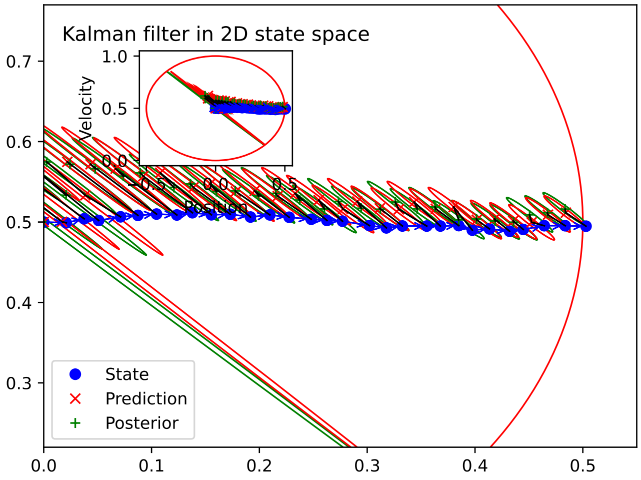

FIGURE 39.3: *The Kalman filter works with quite strange measurement models. Details in the text.*

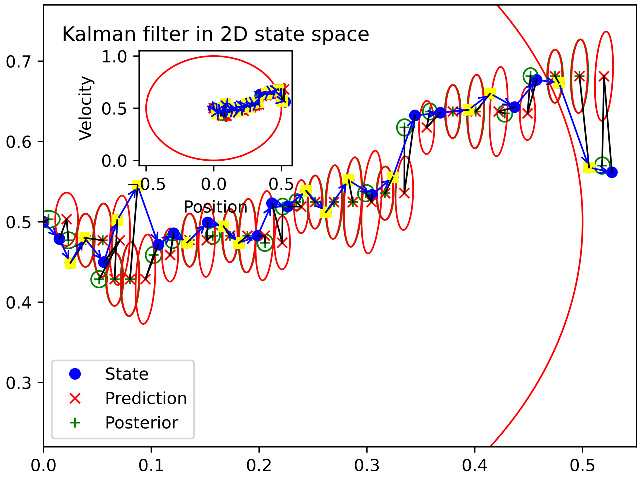

FIGURE 39.4: *The Kalman filter easily adapts to missing measurements. Details in the text.*

39.3 YOU SHOULD

39.3.1 remember these facts:

The form of a linear model of dynamics and measurement.	641
The Kalman filter maintains the mean and covariance of predictive and posterior distributions	648
The extended Kalman filter is often useful when either dynamics or measurement are non-linear, but can behave badly.	651

39.3.2 remember these procedures:

The 1D Kalman filter updates estimates of the mean and covariance of the various distributions encountered while tracking a one-dimensional state variable using the given dynamic model.	645
The Kalman filter updates estimates of the mean and covariance of the various distributions encountered while tracking a state variable of some fixed dimension using the given dynamic model.	647
The Kalman filter updates estimates of the mean and covariance of the various distributions encountered while tracking a state variable of some fixed dimension using the given dynamic model.	651

39.3.3 be able to:

- Build a simple TbD application using a detector.
- Exploit a Kalman filter in this application.

EXERCISES

QUICK CHECKS

39.1. Use Bayes' rule to construct $P(\mathbf{X}_0|\mathbf{Y}_0 = \mathbf{y}_0)$ from $P(\mathbf{X}_0)$ and $P(\mathbf{Y}_0 = \mathbf{y}_0|\mathbf{X}_0)$.

39.2. Check that

$$\int P(\mathbf{X}_i|\mathbf{X}_{i-1}, \mathbf{y}_0, \dots \mathbf{y}_{i-1}) P(\mathbf{X}_{i-1}|\mathbf{y}_0, \dots, \mathbf{y}_{i-1}) d\mathbf{X}_{i-1} = \int P(\mathbf{X}_i|\mathbf{X}_{i-1}) P(\mathbf{X}_{i-1}|\mathbf{y}_0, \dots, \mathbf{y}_{i-1}) d\mathbf{X}_{i-}$$

39.3. Show that

$$P(\mathbf{X}_i|\mathbf{y}_0, \dots, \mathbf{y}_i) \propto P(\mathbf{y}_i|\mathbf{X}_i) P(\mathbf{X}_i|\mathbf{y}_0, \dots, \mathbf{y}_{i-1})$$

39.4. Give a (reasonably practical) example of a case where the dynamical matrix $\mathcal{D}_i$ depends on the clock tick.

39.5. Show that if $x_{i-1} \sim N(\overline{\mathbf{x}}_{i-1}^{-}, d^2(\sigma_{i-1}^{-})^2)$, then $d_i x_{i-1} \sim N(d_i \overline{\mathbf{x}}_{i-1}^{-}, d^2(\sigma_{i-1}^{-})^2)$.

39.6. Show that if $x_{i-1} \sim N(\overline{\mathbf{x}}_{i-1}^{-}, d^2(\sigma_{i-1}^{-})^2)$; and $x_i = d_i x_{i-1} + \xi$; and $\xi \sim N(0, \sigma_{n,i}^2)$; then

$$\overline{\mathbf{x}}_i^{-} = d_i \overline{\mathbf{x}}_{i-1}^{-} \text{ and } (\sigma_i^{-})^2 = d^2(\sigma_{i-1}^{-})^2 + \sigma_{n,i}^2.$$

39.7. Pattern match to the expression for a normal distribution to derive

$$\overline{\mathbf{x}}_i^{+} = \left[\frac{m_i y_i (\sigma_i^{-})^2 + \overline{\mathbf{x}}_i^{-}(\sigma_{m,i})^2}{m_i^2(\sigma_i^{-})^2 + (\sigma_{m,i})^2}\right] \text{ and } (\sigma_i^{+})^2 = \left[\frac{(\sigma_{m,i})^2(\sigma_i^{-})^2}{m_i^2(\sigma_i^{-})^2 + (\sigma_{m,i})^2}.\right]$$

39.8. Rearrange the result of the last exercise to find

$$x_i^{+} = \overline{x}_i^{-} + k_i \left(y_i - m_i \overline{x}_i^{-}\right)$$

where

$$k_i = \frac{m_i(\sigma_i^{-})^2}{m_i^2(\sigma_i^{-})^2 + (\sigma_{m,i})^2}.$$

39.9. Check that if the measurement is wholly unreliable – so $\sigma_{m,i} >> \sigma_i^{-}$ – the weight k_i will be very small.

39.10. Further, check that if $\sigma_i^{-} >> \sigma_{m,i}$, the posterior mean will be very largely the measurement corrected for units.

39.11. Section 39.1.8 has: "The posterior is much better, because the filter received a velocity measurement." Explain.

39.12. Check that if

$$\mathbf{x}_{i+1} = \mathbf{f}(\mathbf{x}_i, \mathbf{n}) \text{ and } \mathbf{x}_i \sim N(\overline{\mathbf{x}}_i, \Sigma_{\mathbf{x}_i}),$$

$\mathbf{n}$ is noise, $\mathbf{n} \sim N(\mathbf{0}, \Sigma_{s,i})$, then linearizing with a Taylor series yields the approximation:

$$\mathbf{x}_{i+1} \sim N(\mathbf{f}(\overline{\mathbf{x}}_i), \mathcal{J}_{\mathbf{f},\mathbf{x}} \Sigma_{\mathbf{x}_i} + \mathcal{J}_{\mathbf{f},\mathbf{x}}^T + \mathcal{J}_{\mathbf{f},\mathbf{n}} \Sigma_{s,i} \mathcal{J}_{\mathbf{f},\mathbf{n}}^T)$$

39.13. Section 39.2.3 has the following expressions for prediction in an extended Kalman filter:

$$\overline{\mathbf{x}}_i^{-} = \mathbf{f}(\overline{\mathbf{x}}_{i-1}^{+}, \mathbf{0}) \text{ and } \Sigma_i^{-} = \mathcal{J}_{\mathbf{f},\mathbf{x}} \Sigma_{i-1}^{+} \mathcal{J}_{\mathbf{f},\mathbf{x}}^T + \mathcal{J}_{\mathbf{f},\mathbf{n}} \Sigma_{s,i} \mathcal{J}_{\mathbf{f},\mathbf{n}}^T.$$

Check this expression is correct.

39.14. Check that, for the EKF, $\mathcal{J}_{\mathbf{f},\mathbf{x}}$

LONGER PROBLEMS

39.15. This exercises shows that models of points moving with constant velocity or constant acceleration fit into the framework where $\mathbf{p}_i = \sum_{u=1}^{u=k} a_u \mathbf{p}_{i-u} +$ gaussian noise for some constants a_u. Write $\mathbf{p}$ for position, $\mathbf{v}$ for velocity and $\mathbf{a}$ for acceleration.

(a) In a constant velocity model, $\mathbf{v}_i \sim N(\mathbf{v}_{i-1}, \Sigma_{v,i-1})$. Show that this means that $\mathbf{v}_i \sim N(\overline{\mathbf{v}}_0, \sum_{u=0}^{u=i-1} \Sigma_{v,u})$.

(b) In a constant velocity model, $\mathbf{v}_i \sim N(\mathbf{v}_{i-1}, \Sigma_{v,i-1})$ and $\mathbf{x}_i \sim N(\mathbf{x}_{i-1} + \delta t \mathbf{v}_{i-1}, \Sigma_{x,i-1})$. Show that $\mathbf{x}_i \sim N(\mathbf{x}_{i-1} + \delta t \overline{\mathbf{v}}_0, \Sigma_{x,i-1} + \sum_{u=0}^{u=i-1} \Sigma_{v,u})$. Conclude that, for a given constant velocity model, there is an equivalent model in the framework where $\mathbf{p}_i = \sum_{u=1}^{u=k} a_u \mathbf{p}_{i-u} +$ gaussian noise.

(c) Repeat the previous construction for a constant acceleration model.

39.16. You have a non-linear function $\mathbf{h}$ of a random variable $\mathbf{x}$. The random variable is normal, so

$$\mathbf{x} \sim N(\mu_{\mathbf{x}}, \Sigma_{\mathbf{x}}).$$

The function is d-dimensional, and accepts a second argument $\mathbf{n}$. Here $\mathbf{n} \sim N(\mathbf{0}, \Sigma_{\mathbf{n},i})$. Write

$$\mathbf{u} = \mathbf{h}(\mathbf{x}, \mathbf{n}).$$

Write $\mathcal{J}_{\mathbf{h},\mathbf{x}}$ for the matrix of first partial derivatives of $\mathbf{h}$ with respect to the first argument, and $\mathcal{J}_{\mathbf{h},\mathbf{n}}$ for the matrix of first partial derivatives of $\mathbf{h}$ with respect to the second argument.

(a) Use a Taylor series to show that, for sufficiently small Σ_x and Σ_n, the approximation

$$\mathbf{u} \sim N(\mathbf{h}(\mu_{\mathbf{x}}, \mathbf{0}), \mathcal{J}_{\mathbf{h},\mathbf{x}} \Sigma_{\mathbf{x}} + \mathcal{J}_{\mathbf{h},\mathbf{x}}^T + \mathcal{J}_{\mathbf{h},\mathbf{n}} \Sigma_{\mathbf{n}} \mathcal{J}_{\mathbf{h},\mathbf{n}}^T).$$

is reasonable.

(b) The previous exercise uses "sufficiently small Σ_x and Σ_n", but is vague about what this means. Assume that each of these covariances is diagonal. What could go wrong with the approximation if a term on the diagonal was large?

PROGRAMMING EXERCISES

This exercise investigates the effects of non-linearity in dynamical models.

(c) Consider the dynamical model $x_i = x_{i-1} + \epsilon \sin(2\pi x_{i-1})$. There is no noise, and the model appears to be only "slightly" non-linear ($(x_i - x_{i-1})^2 \leq \epsilon^2$). For a range of intervals, $x_i > x_{i-1}$; for another range of intervals, $x_i < x_{i-1}$; and for a set of discrete points, $x_i = x_{i-1}$. Write down expressions for each of these three sets.

(d) Build a simulation of this model, where $x_0 \sim N(0, 9)$. Do this by drawing samples for x_0, then propagating them forward. Simulate until $i = 10$. What does the resulting distribution look like? How would you model it? You will find $\epsilon = 0.001$ and $\epsilon = 0.1$ informative.

(e) Now consider $x_i = x_{i-1} + \epsilon \sin(2\pi x_{i-1}) + \delta\xi$, where $\xi \sim N(0, 1)$. Simulate this model until $i = 10$ for various values of ϵ and δ. How big does δ have to be so that the resulting distribution looks normal? You will find $(\epsilon, \delta) = (0.1, 0.1)$ and $(\epsilon, \delta) = (0.1, 0.01)$ informative.