A SHORT INTRODUCTION TO KALMAN FILTERS

Alan Washburn
Naval Postgraduate School
August, 2015

Introduction

Kalman filtering is a method for recursively updating an estimate $\boldsymbol{\mu}$ of the state of a system by processing a succession of measurements $\mathbf{Z}$. After each measurement, a new state estimate is produced by the filter's *measurement* step. $\mathbf{Z}$ and $\boldsymbol{\mu}$ do not necessarily have to have the same dimensionality. For example, $\boldsymbol{\mu}$ could be an estimate of the two dimensional position of a target, which would be represented as a 2×1 matrix, whereas $\mathbf{Z}$ could be a bearing to the target (a scalar). Provision is made in the filter's *movement* model for the possibility that the state of the system may change between measurements. These notes describe the details of both the measurement and movement steps, using a succession of examples.

The reader is assumed to be familiar with fundamental matrix properties and algebra. See exercise 1 for a review of the essentials. It will be useful to have access to Excel, especially the workbook *Kalman.xlsm*.

Throughout these notes, symbols for random variables will be in uppercase font, as will certain other symbols. Vector and matrix symbols will be bold.

Multidimensional random variables have covariance matrices, rather than variances. For consistency, we may refer to covariance matrices even in scalar examples where the simpler notion of variance would suffice, and will continue to use bold symbols such as $\boldsymbol{\mu}$ and $\mathbf{Z}$ even in scalar examples. See Appendix B for important properties of covariance matrices.

The *state* $\mathbf{X}$ is assumed to be a multivariate normal random variable with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$, which we abbreviate $\mathbf{X} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. $\mathbf{V}$ is the measurement noise and $\mathbf{W}$ is the movement noise; with $\mathbf{V} \sim N(\boldsymbol{\mu}_\mathbf{V}, \mathbf{R})$ and $\mathbf{W} \sim N(\boldsymbol{\mu}_\mathbf{W}, \mathbf{Q})$. Random variables $\mathbf{X}$ and $\mathbf{W}$ are n-dimensional and $\mathbf{V}$ is m-dimensional.

All computations can be thought of as manipulations of multivariate normal probability distributions. In fact, there are only two essential facts on which the whole structure is built, the first being associated with movement and the second with measurement. These facts are stated below using random variables $\mathbf{X}$, $\mathbf{V}$, and $\mathbf{W}$, all independent of each other, with means and covariance matrices as introduced above. If $\boldsymbol{\phi}$

and **H** are appropriately dimensioned movement and measurement matrices, the two vital facts are (the superscript^t means transpose and **I** is the identity matrix):

1) If $\mathbf{X}' = \boldsymbol{\phi}\mathbf{X} + \mathbf{W}$, then $\mathbf{X}' \sim N(\boldsymbol{\mu}', \boldsymbol{\Sigma}')$, where $\boldsymbol{\mu}' = \boldsymbol{\phi}\boldsymbol{\mu} + \boldsymbol{\mu}_\mathbf{W}$ and $\boldsymbol{\Sigma}' = \boldsymbol{\phi}\boldsymbol{\Sigma}\boldsymbol{\phi}^t + \mathbf{Q}$.

2) If $\mathbf{Z} = \mathbf{H}\mathbf{X} + \mathbf{V}$, then, conditional on **Z** being given, $\mathbf{X} \sim N(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}})$, where $\hat{\boldsymbol{\mu}} = \boldsymbol{\mu} + \mathbf{K}(\mathbf{Z} - \boldsymbol{\mu}_\mathbf{V} - \mathbf{H}\boldsymbol{\mu})$ and $\hat{\boldsymbol{\Sigma}} = (\mathbf{I} - \mathbf{K}\mathbf{H})\boldsymbol{\Sigma}$. The matrix **K** is called the Kalman gain, and is given by $\mathbf{K} = \boldsymbol{\Sigma}\mathbf{H}^t (\mathbf{H}\boldsymbol{\Sigma}\mathbf{H}^t + \mathbf{R})^{-1}$.

Discussion of Fact 1:

The purpose of a KF is to keep track of the state of a system by making a sequence of measurements. It is permitted to have the state of the system change randomly from **X** to **X**′ between measurements, with the attractive feature of the movement model $\mathbf{X}' = \boldsymbol{\phi}\mathbf{X} + \mathbf{W}$ being that normality is preserved. The formula for $\boldsymbol{\mu}'$ should make intuitive sense, given the movement model. Recall that in the scalar case $\mathrm{Var}(\boldsymbol{\phi}\mathbf{X}) = \boldsymbol{\phi}^2 \mathrm{Var}(\mathbf{X})$, so the presence of both $\boldsymbol{\phi}$ and $\boldsymbol{\phi}^t$ in the formula for $\boldsymbol{\Sigma}'$ should come as no surprise. **Q** is additive because **W** is independent of **X**. $\boldsymbol{\phi}$ and **Q** are $n \times n$ matrices, and $\boldsymbol{\mu}_\mathbf{W}$ is an n-vector. If the system state does not change between measurements (vacuous movement step), then $\boldsymbol{\phi} = \mathbf{I}$, $\boldsymbol{\mu}_\mathbf{W} = 0$, and $\mathbf{Q} = 0$.

Discussion of Fact 2:

Suppose you were told that $\mathbf{X} \sim N(-1, 4)$, and asked to guess **X**. You would probably guess "–1", the mean value. Suppose you were also told that $\mathbf{Z} = -3.1$, and that Z was obtained by adding a measurement error **V** to **X**, where $\mathbf{V} \sim N(0, 2)$. In other words, –3.1 is a noisy measurement of the unknown **X**. Given this information, what would you guess for **X**? Since **Z** is smaller than –1, and since **Z** represents a reasonably accurate, unbiased ($\boldsymbol{\mu}_\mathbf{V} = 0$) measurement of **X**, your intuition would tell you to revise your estimate of **X** downward. In fact, since $2 < 4$, you would probably conclude that the best guess at **X** would be closer to –3.1 than to –1. The best way to make these intuitive considerations precise is to employ Bayes' Theorem, as is done in appendix A. The result of that application is Fact 2. In our scalar example, $\boldsymbol{\Sigma} = 4$, $\mathbf{H} = 1$, and $\mathbf{R} = 2$. It follows

that $\mathbf{K} = 2/3$, $\hat{\boldsymbol{\mu}} = (-1) + (2/3)(-3.1 + 1) = -2.4$, and $\hat{\boldsymbol{\Sigma}} = (1 - 2/3)\boldsymbol{\Sigma} = 4/3$. Not only does Bayes tell you to guess –2.4, but he tells you how accurate the guess is!

Fact 2 states that the best way to process the information in **Z** is to revise the "inputs" $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ to the "outputs" $\hat{\boldsymbol{\mu}}$ and $\hat{\boldsymbol{\Sigma}}$. The state of the system is still normal after the measurement is processed — the mean and covariance matrix have simply changed. The fact that normality is preserved is important, since $\hat{\boldsymbol{\mu}}$ and $\hat{\boldsymbol{\Sigma}}$ may themselves be the inputs to similar calculations in the future.

The simplicity of the way in which Kalman revises $\boldsymbol{\mu}$ to $\hat{\boldsymbol{\mu}}$ after a measurement is significant. Note that $\mathbf{H}\boldsymbol{\mu} + \boldsymbol{\mu}_\mathbf{V}$ is the mean or best guess of the measurement **Z**, so that $\mathbf{Z} - \boldsymbol{\mu}_\mathbf{V} - \mathbf{H}\boldsymbol{\mu}$ is the "shock" caused by the measurement*. If the measurement is not shocking, Kalman sets $\hat{\boldsymbol{\mu}} = \boldsymbol{\mu}$; otherwise, he makes a correction that is proportional to the shock. The matrix **K** is simply the proportionality constant. This method of revising $\boldsymbol{\mu}$ to $\hat{\boldsymbol{\mu}}$ is so simple and natural that the robustness of the procedure with respect to modeling errors should not be surprising.

Employment of Fact 2 requires one to know **H**, $\boldsymbol{\mu}_\mathbf{V}$, and **R**. If **Z** has *m* components, then **H** is $m \times n$, $\boldsymbol{\mu}_\mathbf{V}$ is $m \times 1$, and **R** is $m \times m$. In general, computation of **K** requires a matrix inverse.

Operation of the KF

There are two more required inputs: $\boldsymbol{\mu}_0$ and $\boldsymbol{\Sigma}_0$ are the initial values for $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, respectively. These two inputs determine the prior distribution that Bayes' theorem requires. Once $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are initialized, all calculations correspond to either movement or measurement, as shown in the diagrams in Figure 1, where the replacement symbol ← makes it possible to dispense with the ' and ˆ notation used in stating Facts 1 and 2. The ← notation emphasizes that operation of a Kalman Filter can be thought of as a sequence of updates to $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$. Sufficient memory to store one copy of $\boldsymbol{\mu}$ and one copy of $\boldsymbol{\Sigma}$ is all

* There is a different formula for shock if the KF is *extended* (see p. 16).

that is really required when these updates are made by computer, as is usually the case. However, for tutorial purposes it is sometimes useful to let $(\boldsymbol{\mu}_i(-), \boldsymbol{\Sigma}_i(-))$ be $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ with all updates up to time i except for the measurement at time i, and to let $(\boldsymbol{\mu}_i(+), \boldsymbol{\Sigma}_i(+))$ be similarly defined except that the update for the measurement at time i is included. Thus the measurement block of Figure 1 updates $(\boldsymbol{\mu}_i(-), \boldsymbol{\Sigma}_i(-))$ to $(\boldsymbol{\mu}_i(+), \boldsymbol{\Sigma}_i(+))$, while the movement block updates $(\boldsymbol{\mu}_i(+), \boldsymbol{\Sigma}_i(+))$ to $(\boldsymbol{\mu}_{i+1}(-), \boldsymbol{\Sigma}_{i+1}(-))$. Other matrices will also be subscripted for time in this expanded notation.

MOVEMENT
$\boldsymbol{\mu} \leftarrow \boldsymbol{\phi}\boldsymbol{\mu} + \boldsymbol{\mu}_w$
$\boldsymbol{\Sigma} \leftarrow \boldsymbol{\phi}\boldsymbol{\Sigma}\boldsymbol{\phi}^t + \mathbf{Q}$

MEASUREMENT (Z)
$\mathbf{K} \leftarrow \boldsymbol{\Sigma}\mathbf{H}^t(\mathbf{H}\boldsymbol{\Sigma}\mathbf{H}^t + \mathbf{R})^{-1}$
$\boldsymbol{\mu} \leftarrow \boldsymbol{\mu} + \mathbf{K}(\mathbf{Z} - \boldsymbol{\mu}_v - \mathbf{H}\boldsymbol{\mu})$
$\boldsymbol{\Sigma} \leftarrow (\mathbf{I} - \mathbf{K}\mathbf{H})\boldsymbol{\Sigma}$

Figure 1. Showing the calculations corresponding to movement and measurement in a Kalman Filter.

Summary of Notation

$\boldsymbol{\phi}$ is the **movement matrix**, and is part of the description of how the state changes between measurements.

$(\boldsymbol{\mu}_W, \mathbf{Q})$ is the mean and covariance of the **movement noise**. If you accidentally make **Q** too large, the filter will be high strung; that is, the filter's estimates will bounce around too much becauses **K** is too large. If you make **Q** too small, the filter will be lethargic.

H is the **measurement matrix** that describes how the measurement depends on the state.

$(\boldsymbol{\mu}_V, \mathbf{R})$ is the mean and covariance of the **measurement noise**. The filter's tendencies with respect to **R** are the opposite of those with **Q**.

$(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is the mean and covariance of the **state** of the system. $\boldsymbol{\mu}$ can also be interpreted as a guess at the state **X**. The initial values $(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$ must be provided; after that, it is the filter's job to continually update $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$.

Z is the **measurement**.

K is the **Kalman gain** that is used to update ($\boldsymbol{\mu}$, $\boldsymbol{\Sigma}$) by processing **Z** .

The best way to become familiar with the notation and the use of the movement and measurement models is to carefully track through the details of some examples. Several are provided below. In the examples that follow, we will describe the actions of a Kalman filter as those of the man Kalman himself, as if the filter were someone sitting at a desk processing inputs (see cover). The man Kalman is simply a surrogate for the matrix algebra outlined above.

Example 1 (random walk)

A target moves (or appears to move) in a one-dimensional random walk, adding an increment to its position between observations that is normal with mean 1 mile and standard deviation 2 miles; that is, $\mathbf{X}_{i+1} = \mathbf{X}_i + \mathbf{W}_i$, where $\mathbf{W}_i$ is the increment. It follows that $\boldsymbol{\phi} = 1$, $\mathbf{Q} = 4$ miles^2, and $\boldsymbol{\mu}_{\mathbf{W}} = 1$ mile. The state or position of the target is basically increasing with time, but the randomness of $\mathbf{W}_i$ will cause occasional exceptions where the state decreases instead of increasing. We also assume $\mathbf{H} = 1$, $\boldsymbol{\mu}_{\mathbf{V}} = 0$, and $\mathbf{R} = 9$ miles^2, which is the same as saying that unbiased measurements of **X** are available that are accurate to within about $\sqrt{\mathbf{R}} = 3$ miles standard deviation. Kalman's initial guess at the target's position is $\boldsymbol{\mu}_0 = 0$, $\boldsymbol{\Sigma}_0 = 10000$ miles^2, the large value for $\boldsymbol{\Sigma}_0$ indicating that whoever was forced to make the initial guess had basically no idea where the target was. Suppose the first three measurements are 84, 83, and 88, from which we might conclude that the target's position is somewhere in the 80's even without Kalman's help. Since a measurement is made before the target moves, we take ($\boldsymbol{\mu}_1(-)$, $\boldsymbol{\Sigma}_1(-)$) to be ($\boldsymbol{\mu}_0$, $\boldsymbol{\Sigma}_0$). Kalman would use the measurement and movement blocks alternately, with the results for ($\boldsymbol{\mu}$, $\boldsymbol{\Sigma}$) shown in Figure 2. Since the first Kalman gain is (10000/10009) ≈ 1, Kalman forgets the initial guess entirely as soon as the first measurement is available, guessing instead that the target's position is $\boldsymbol{\mu}_1(+) = 84$, with the associated accuracy being the same as the accuracy of a measurement. He then adds 1 to $\boldsymbol{\mu}_1(+)$ and 4 to $\boldsymbol{\Sigma}_1(+)$ to obtain $\boldsymbol{\mu}_2(-)$ and $\boldsymbol{\Sigma}_2(-)$, reflecting the idea that the best guess of the target's position right before the second measurement is one unit larger than the best guess right after the first, but that $\boldsymbol{\mu}_2(-)$ is a worse guess than $\boldsymbol{\mu}_1(+)$ because of the unpredictable part of the target's motion. Turn the crank to make sure you see how the rest of the numbers are obtained. Letting $\mathbf{K}_i$

be the Kalman gain for the i^{th} measurement, you should get $\mathbf{K}_3 = 9.32/(9.32 + 9) = 0.51$, $\mu_3(+) = 84.82 + 0.51\ (88 - 84.82) = 86.44$.

i	**Before Measurement *i***			**After Measurement *i***	
	$\mu_i(-)$	$\mathbf{\Sigma}_i(-)$	$\mathbf{K}_i$	$\mu_i(+)$	$\mathbf{\Sigma}_i(+)$
1	0	10000	1.00	84	9
2	85	13	.59	83.82	5.32
3	84.82	9.32	0.51	86.44	4.58
4	87.44	8.58	0.49	?	4.39
	?	⋮	⋮	?	⋮
∞	?	8.33	0.48	?	4.33

Figure 2. Employment of a Kalman Filter to track a target.

Note that $\mathbf{K}_4$ and $\mathbf{\Sigma}_4(+)$ can be computed even before the fourth measurement is made; in fact, the entire sequence of Kalman gains and covariances is completely independent of the measurements. A close inspection of Figure 1 shows that this will always be the case; neither $\mathbf{Z}$ nor $\boldsymbol{\mu}$ is ever used in computing $\mathbf{\Sigma}$ or $\mathbf{K}$. This could be an important feature in a situation where measurements have to be processed rapidly, since the Kalman gains can all be computed beforehand.

The movement block is a variance increasing operation, while the measurement block is a variance decreasing operation. This is evident in Figure 2. It sometimes happens that the net result of these opposite forces is that the covariance matrices and Kalman gains approach steady state limits. Assuming they exist, the steady state limits $\mathbf{\Sigma}(-)$, $\mathbf{K}$, and $\mathbf{\Sigma}(+)$ must satisfy the equations:

$$\mathbf{\Sigma}(-) = \boldsymbol{\phi}\mathbf{\Sigma}(+)\boldsymbol{\phi}^t + \mathbf{Q}$$

$$\mathbf{K} = \mathbf{\Sigma}(-)\mathbf{H}^t\left(\mathbf{H}\mathbf{\Sigma}(-)\mathbf{H}^t + \mathbf{R}\right)^{-1}$$

$$\mathbf{\Sigma}(+) = (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{\Sigma}(-)$$

In our scalar example with $\boldsymbol{\phi} = \mathbf{H} = 1$, $\mathbf{R} = 9$, and $\mathbf{Q} = 4$, the only positive solution is

$$\boldsymbol{\Sigma}(+) = \left(\sqrt{\mathbf{Q}^2 + 4\mathbf{QR}} - \mathbf{Q}\right)/2 = 4.33$$
$$\boldsymbol{\Sigma}(-) = \boldsymbol{\Sigma}(+) + \mathbf{Q} = 8.33$$
$$\mathbf{K} = \boldsymbol{\Sigma}(-)/\left(\boldsymbol{\Sigma}(-) + \mathbf{R}\right) = 0.48$$

Comparison of these three numbers with the fourth row of Figure 2 shows that the steady state limit is approached rather quickly in this case. Note that the steady state accuracy of Kalman's estimate right after a measurement ($\sqrt{\boldsymbol{\Sigma}(+)}$ = 2.08 miles) is better than the accuracy associated with the latest measurement (3 miles). In fact, $\boldsymbol{\Sigma}(+)$ would be 0 if either **R** were 0 (that's obvious) *or* if **Q** were 0 (that's not obvious, but think about what happens when you can make lots of measurements of an unknown but *fixed* quantity). Note also that the steady state equations do not involve $\boldsymbol{\mu}_0$ or $\boldsymbol{\Sigma}_0$, which is a relief.

A particularly simple filter would use the steady state **K** at every stage and dispense with the covariance matrix calculations. Such a filter will typically behave poorly in the early stages unless the initial guess $\boldsymbol{\mu}_0$ happens to be close to the truth. But if only steady state performance is important, the simplicity of the technique could make it attractive. Try repeating the calculation of $\boldsymbol{\mu}_4(-)$ in example 1 using $\mathbf{K}_1 = \mathbf{K}_2 = \mathbf{K}_3 = 0.48$. Better yet, use $\mathbf{K}_1 = \mathbf{K}_2 = \mathbf{K}_3 = 0.5$, since the basic point is that the filter will still do a good job even if the gain schedule isn't precisely as indicated in Figure 2. The technique of smoothing a scalar times series with a constant Kalman gain is sometimes called "exponential smoothing".

Example 2 (massive target with velocity)

Suppose now that the target of example 1 is actually quite massive, so that the erratic type of random walk motion postulated there is implausible over the time period Δ between measurements. Suppose, instead, that the target can actually be thought of as having a velocity that changes by only a small amount (say q miles/hr standard deviation) over time Δ. The "state" of the target is now $\mathbf{X}=(X, V)^t$, a 2×1 vector where V is the velocity. The velocity itself behaves like the position in example 1, but with $\mathbf{Q}=q^2$ and $\boldsymbol{\mu}_\mathbf{W}=0$. Since the update equation for X is $X_{i+1} = X_i + \Delta V_i$, the first row of □s [1 Δ]. Each row of $\boldsymbol{\phi}$ expresses how one of the state variables is movement-updated as a function of the others. Altogether,

$$\boldsymbol{\phi} = \begin{bmatrix} 1 & \Delta \\ 0 & 1 \end{bmatrix} \quad \mathbf{Q} = \begin{bmatrix} 0 & 0 \\ 0 & q^2 \end{bmatrix} \quad \boldsymbol{\mu}_{\mathbf{W}} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Assuming that the measurement is still of the target's position, **H** is now the 1×2 matrix **H** = [1 0], and **R** is as before. After initializing $\boldsymbol{\Sigma}_0$ and $\boldsymbol{\mu}_0$ (which are now 2×2 and 2×1 matrices), the movement and measurement blocks can be employed as before (see exercise 9).

If the movement model of example 2 is a better representation of how targets move than was the simple random walk of example 1, then the position estimates (first component of $\boldsymbol{\mu}$) will be more accurate in example 2. This trick of improving accuracy by augmenting the state vector will come as no surprise to a reader familiar with Markov Chains. The main computational effort in implementing a Kalman Filter is in calculating **K**, where a matrix with the same dimensions as **R** must be inverted. Since the dimensions of this matrix are independent of the size of the state vector, the computational impact of state augmentation is small.

Example 3 (relative measurement)

Let **X** be the location of a target, and suppose $\boldsymbol{\mu}(-)=0$ and $\boldsymbol{\Sigma}(-)=1m^2$. A sensor located at $x_0=10m$ measures the location of the target relative to itself. The sensor's accuracy as a standard deviation is $2m$, and the measurement is $-8m$. What are $\boldsymbol{\mu}(+)$ and $\boldsymbol{\Sigma}(+)$? To answer this question, we first state the equation that determines the measurement. We take this equation to be $\mathbf{Z}=\mathbf{X}-x_0+\mathbf{V}$, with **V** being the measurement error. We suppose measurements are unbiased, so $\boldsymbol{\mu}_{\mathbf{V}}=0$, and we assume $\mathbf{R}=4m^2$. The measurement equation is not supposed to have a constant in it, but this difficulty can be remedied by simply eliminating it from the equation for **Z** and making $\boldsymbol{\mu}_{\mathbf{V}}=-x_0$, which is analytically equivalent. Since $\mathbf{H}=1$, the Kalman gain is $\mathbf{K}=0.2$. The shock is $\mathbf{Z}-\mathbf{H}\boldsymbol{\mu}(-)-\boldsymbol{\mu}_{\mathbf{V}}=-8-0+10=2m$. Therefore $\boldsymbol{\mu}(+)=\boldsymbol{\mu}(-)+0.2(2m)=0.4m$. Kalman expects the measurement to be $-10m$, since the sensor is at $10m$ and he expects the target to be at 0. When the measurement turns out to be $-8m$, Kalman increases his estimate of the target's location by an amount proportional to the shock. Since the measurement is not very accurate compared to the initial uncertainty of the target's location, the adjustment is

small, and $\mathbf{\Sigma}(+)$ is $0.8m^2$, not much smaller than $\mathbf{\Sigma}(-)$. Note that all of the algebraic manipulations involving x_0 affect only the shock, and could have been avoided by using the principle that *the shock is always the difference between the actual measurement and what Kalman expects it to be*.

Example 4 (unknown sensor location)

This is the same as example 3, except that the sensor itself has an unknown location. Specifically, let the sensor's location be the random variable X_0, and suppose $E(X_0)=10m$ and $Var(X_0)=5m^2$. There are two ways of proceeding. The first is to simply increase **R** to $9m^2$, since the uncertainty about the sensor's location has the effect of adding $5m^2$ to the measurement's variance. The result of this is that $\mathbf{K}=0.1$, $\boldsymbol{\mu}(+)=0.2m$, and $\mathbf{\Sigma}(+)=0.9m^2$. The second (and recommended) way of proceeding is to augment the state vector to include the sensor's location, so that $\mathbf{X}=(X, X_0)^t$. The measurement equation is now $Z=X-X_0+V$, with $\mu_V=0$. This is $\mathbf{Z}=\mathbf{HX}+\mathbf{V}$ if we take **H** to be [1 –1]. We also have $\boldsymbol{\mu}(-)=\begin{bmatrix}0\\10m\end{bmatrix}$ and $\mathbf{\Sigma}(-)=\begin{bmatrix}1m^2 & 0\\0 & 5m^2\end{bmatrix}$ — the zeros in the covariance matrix represent our assumption that the errors in the target and sensor locations are independent. We can now perform the measurement update, obtaining $\mathbf{K}=\begin{bmatrix}0.1\\-0.5\end{bmatrix}$, $\boldsymbol{\mu}(+)=\begin{bmatrix}0.2m\\9m\end{bmatrix}$, and $\mathbf{\Sigma}(+)=\begin{bmatrix}0.9m^2 & 0.5m^2\\0.5m^2 & 2.5m^2\end{bmatrix}$. This is consistent with the first way of proceeding, but superior for two reasons. One is that the second approach offers more information — note that Kalman revises the location of the sensor, as well as the target, and indicates through the covariance matrix that his errors in estimating the two components of **X** are positively correlated after the measurement is processed. More importantly, the first approach could easily lead to wrong answers if a stationary sensor were to subsequently make another measurement. One of the assumptions on which Kalman filtering is based is that measurement errors at different times are independent. In the first approach, we would implicitly be assuming that the sensor relocates itself between measurements, which is not true. The second approach correctly reflects the idea that the sensor's location is unknown, but also unchanging. As in example 2, the best course of action turns out to be enlargement of the state vector.

The IOU Model and MTST

Let V_t be the velocity of a target at time t, and suppose that $V_{t+1} = V_t + W_t$ for $t \geq 1$, where $W_1, W_2, \ldots$ is a sequence of independent identically distributed normal random variables with mean 0 and variance Q. This is a model of a random walk, as employed in example 2. Since $\text{Var}(V_{t+1}) = \text{Var}(V_t) + Q$, the sequence $V_1, V_2, \ldots$ has a progressively increasing variance. In the long run, according to this movement model, target speeds that exceed (say) 1000 miles/hr are not only possible but likely. Most real targets on earth cannot achieve such speeds, so this feature must be considered a modeling defect. However, a simple revision can keep $\text{Var}(V_t)$ within bounds, and thereby render such extreme deviations from 0 unlikely.

The revision is $V_{t+1} = cV_t + W_t$, where $0 \leq c < 1$. Since $\text{Var}(V_{t+1}) = c^2 \text{Var}(V_t) + Q$, the shrinkage factor c will prevent $\text{Var}(V_t)$ from growing large with t. In fact, $\text{Var}(V_t)$ now has a limit s^2 as t approaches infinity, and this limit must satisfy the equation $s^2 = c^2 s^2 + Q$. If c and s are known, this equation can be solved for $Q = s^2(1 - c^2)$.

It is not hard to quantify or estimate s for a real target, since s is the target's root-mean-square velocity. However, it is also necessary to quantify c. To do so, consider forecasting V_{t+n} from a knowledge of V_t. By applying the movement model n times, it can be shown that $V_{t+n} = c^n V_t + (\text{noise})$, where (noise) is a linear combination of W_t, $W_{t+1}, \ldots, W_{t+n-1}$. The noise is 0 on the average. Now let $c = \exp(-\Delta/\tau)$, where Δ is the length of a time step, so that $c^n = \exp(-n\Delta/\tau)$. Since $n\Delta$ is the length of time over which the forecast is to be made, the parameter τ can be recognized as a *relaxation time* for velocity. Thus the two target motion parameters that need to be quantified are τ and s. Given τ, s, and the time step Δ, one has only to solve $c = \exp(-\Delta/\tau)$ and $Q = s^2(1 - c^2)$ for the inputs to the movement model. For example, the East-West component of a ship's velocity might have $s = 5\ kt.$ and $\tau = 1hr$. If $\Delta = 0.1hr$ (possibly because a measurement of the ship's position is made every 6 minutes), then $c = .905$ and $Q = 4.532(kt)^2$. Figure 3 shows a Monte Carlo simulation of this movement model over an 8-hour period. See sheet "OUSim" of *Kalman.xlsm* to resample, or to see the effect of changing the parameters.

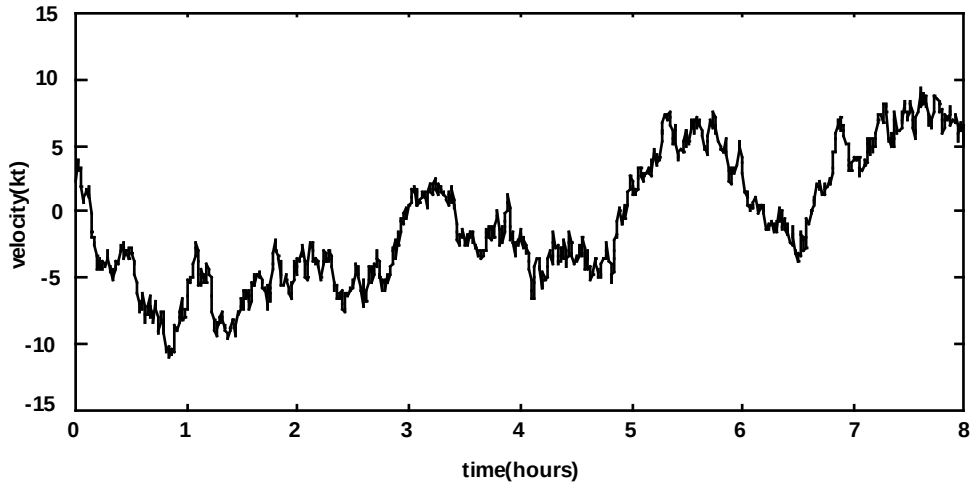

Figure 3. An O-U process fluctuating about 0.

The revised model described above is a discrete version of the Ornstein-Uhlenbeck (O-U) process, the most general normal Markovian stationary stochastic process that is zero on the average. The O-U process is often used as a model of velocity because most velocities are zero on the average, being negative as often as positive. Integrating velocity gives position, so the position of a target whose velocity is O-U is sometimes called IOU.

The U.S. Navy makes use of a Kalman Filter called the Maneuvering Target Statistical Tracker (MTST) for modeling the motion of objects moving in two dimensions. MTST has a four component state vector, the first two being target location and the last two being target velocity. The two components of velocity are assumed to be independent O-U processes with the same (s, τ) parameters. The movement model is thus

$$\boldsymbol{\phi} = \begin{bmatrix} 1 & 0 & \delta & 0 \\ 0 & 1 & 0 & \delta \\ 0 & 0 & c & 0 \\ 0 & 0 & 0 & c \end{bmatrix}, \mathbf{Q} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & Q & 0 \\ 0 & 0 & 0 & Q \end{bmatrix}, \mu_W = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

Parameters c and Q have already been explained. Parameter δ is the multiplier for velocity used in updating position. Since velocity can fluctuate over the measurement interval Δ, and since the predictable part of velocity relaxes toward 0 with relaxation time

τ, the velocity multiplier should be $\delta = \int_0^{\Delta} \exp(-u/\tau)du = \tau(1-c)$. δ is always smaller than Δ, but there is very little difference between the two when Δ is small relative to τ. This completes the description of the three-parameter (s, τ, Δ) MTST movement model.

The O-U process has been described as a model of velocity, but it can be adapted to any phenomenon that fluctuates around 0. For example, suppose that one component of the state vector is the target's course in degrees, and that the target's course fluctuates around 0 with time constant $\tau = 96$ seconds and standard deviation $s = 4$ degrees. Over an 8 second time interval, $c = \exp(-8/96) = .92$, and $q = s^2(1 - c^2) = 2.456$ in units of squared degrees. These are the parameters that would be used in the movement model for 8 seconds of movement. An O-U process can model the deviation of a quantity that is not 0 on the average by dealing with fluctuations from the average, so this model could be easily adapted to deal with a target whose average course is not 0 (see example 8).

Vacuous Movement Step

If the state does not change between measurements, then the movement step has no effect on either $\boldsymbol{\mu}$ or $\boldsymbol{\Sigma}$, and operation of the filter amounts to processing a sequence of measurements on the same unknown state. In these circumstances it is usually best to keep track of the *inverse* $\mathbf{P} = \boldsymbol{\Sigma}^{-1}$ of the covariance matrix, rather than $\boldsymbol{\Sigma}$. **P** is often called the *precision matrix*. In terms of **P**, the measurement step (see appendix A) is

MEASUREMENT
$\mathbf{P} \leftarrow \mathbf{P} + \mathbf{H}^t\mathbf{R}^{-1}\mathbf{H}$ $\mathbf{K} \leftarrow \mathbf{P}^{-1}\,\mathbf{H}^t\,\mathbf{R}^{-1}$ $\boldsymbol{\mu} \leftarrow \boldsymbol{\mu} + \boldsymbol{K}(\mathbf{Z} - \boldsymbol{\mu}_\mathbf{V} - \mathbf{H}\boldsymbol{\mu})$

The simplicity of the update for **P** makes it obvious that any measurement must literally add to the precision with which the state is known. Note that **K** is calculated *after* **P** is updated — the sequence is important.

Since **P** can only grow as more and more measurements are processed, the gain matrix **K** will eventually, in most cases, approach 0. This phenomenon is sometimes referred to as "covariance collapse." There is nothing wrong with this if the state is

indeed known to be unchanging, but if the state *should* change, a collapsed filter will not be able to track the change. Covariance collapse can be prevented by making **Q** nonzero, but in that case the movement step is no longer vacuous.

In a sequence of measurements, let $\mathbf{P}_i$ be the reciprocal of $\mathbf{\Sigma}_i$ (the ± modifiers are no longer needed when there is no movement). Taking $\mathbf{P}_0$ to be 0 is common, since doing so has the effect of giving zero weight to *a priori* judgments about the state.

Example 5 (artillery registration)

Consider a gun shooting at a target that it cannot see, but with an observer who can see the fall of each shot relative to the target. The observer reports the miss distance for each shot, and the gun wishes to use these reports to zero in on the target. Since the target is stationary, the movement model is null. Let **X** be the location of the target and let μ_i and $\mathbf{P}_i$ be the mean and precision of **X** just after the i^{th} shot, with μ_0 and $\mathbf{P}_0$ being the initial mean and precision. Also let a_i be the i^{th} aimpoint. The i^{th} shot lands at $a_i - V_i$, where V_i represents the shot's dispersion error, so the i^{th} measurement is $\mathbf{Z}_i = \mathbf{X} - a_i + \mathbf{V}_i$. We assume $\mu_V = 0$, but that **R** is positive. The aimpoints are immaterial as far as Kalman is concerned, but we assume that $a_{i+1} = \mu_i$ because the gun, after all, is trying to hit the target. Thus the first shot is aimed at μ_0, etc. This aiming policy also means that the expected measurement is always 0, so the i^{th} shock is simply $\mathbf{Z}_i$. Kalman always expects to hit the target because he always aims at the latest estimate of its location.

For simplicity, assume that **X** is one dimensional (if **X** has two or more dimensions, then each can be treated independently). Since $\mathbf{H}=1$, $\mathbf{P}_i$ is $\mathbf{P}_0 + i/\mathbf{R}$. Therefore $\mathbf{K}_i$ is $1/(i + \mathbf{R}\mathbf{P}_0)$. In the special case where $\mathbf{P}_0 = 0$, $\mathbf{K}_i$ is simply $1/i$, and the corresponding aiming procedure is sometimes called "Whistler's rule". In general, the observer reports are taken less and less seriously as the precision of the state estimates increases. Each shot has two purposes. The shot might, of course, hit the target, but even if it misses, it is still the basis of an observer report that serves to refine the estimate of the target's location.

Sheet "ArtillerySim" of *Kalman.xlsm* is a side-by-side simulation of the shooting method described above and a different one called 3CAL. See that sheet or exercise 13 for details.

Example 6 (linear regression)

Consider the linear regression of Y on X, where $(x_i, y_i) = (2,7), (0,2), (5,14)$ for $i = 1, 2, 3$. There is no line in the (X,Y) plane that goes through all three points exactly, but some lines fit better than others. The usual approach is to find the best linear fit of the form $Y = aX + b$, where a and b are chosen to minimize the least squares expression $\sum_{i=1}^{3}(y_i - a - bx_i)^2$. The solution is $(a, b) = (2.395,2.079)$, obtained using the regression function on a hand calculator. The same result can also be obtained using a Kalman filter where $(a, b)^t$ is regarded as the unknown (but unchanging, so there is no movement) state vector $\mathbf{X}$, with $\boldsymbol{\mu}_0$ and $\mathbf{P}_0$ both taken to be 0. The three measurements can be thought of as one super-measurement $\mathbf{Z} = (y_1, y_2, y_3)^t$, with $\mathbf{H} = \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ x_3 & 1 \end{bmatrix}$ We take $\mathbf{R}$ to be $r\mathbf{I}$, where r is a scalar representing the variance associated with each observation and $\mathbf{I}$ is an identity matrix. Carrying out the measurement step, we find that

$$\mathbf{P}_1 = 0 + \frac{1}{r}\mathbf{H}^t\mathbf{H} = \frac{1}{r}\begin{bmatrix} \sum_{i=1}^{3} x_i^2 & \sum_{i=1}^{3} x_i \\ \sum_{i=1}^{3} x_i & 3 \end{bmatrix} = \frac{1}{r}\begin{bmatrix} 29 & 7 \\ 7 & 3 \end{bmatrix},$$

$$\mathbf{K}_1 = \frac{r}{38}\begin{bmatrix} 3 & -7 \\ -7 & 29 \end{bmatrix}\mathbf{H}^t\left(\frac{1}{r}\right) = \frac{1}{38}\begin{bmatrix} -1 & -7 & 8 \\ 15 & 29 & -6 \end{bmatrix}, \text{ and}$$

$$\boldsymbol{\mu}_1 = 0 + \mathbf{K}(\mathbf{Z} - 0) = \mathbf{K}\begin{bmatrix} 7 \\ 2 \\ 14 \end{bmatrix} = \frac{1}{38}\begin{bmatrix} 91 \\ 79 \end{bmatrix} = \begin{bmatrix} 2.395 \\ 2.079 \end{bmatrix},$$

the same result as obtained by linear regression.

The point here is not that Kalman filtering is an easier way to do linear regression (far from it), but that a Kalman filter behaves as expected in a familiar situation. Note that the unknown variance r doesn't enter into calculating $\boldsymbol{\mu}_1$ because it cancels in the computation of $\mathbf{K}_1$. The initial estimate $\boldsymbol{\mu}_0$ was taken to be 0 for convenience in calculating $\boldsymbol{\mu}_1$, but any other estimate would have produced exactly the same result.

In linear regression problems where the index *i* represents time, analysts sometimes worry about the parameters *a* and *b* changing slightly as time goes by. Intuitively, recent measurements should play a stronger role than old ones in estimating the state. This is exactly what would happen in a Kalman filter with a non-vacuous movement step where **Q** is positive, rather than 0.

Extended Kalman filters (EKF'S)

If the measurement is a nonlinear function of the state variables, then the matrix **H** must be obtained by *linearizing* the nonlinear function. Formally, if $\mathbf{Z} = f(\mathbf{X}) + \mathbf{V}$, then we deal with the approximation $\mathbf{Z} \cong f(\boldsymbol{\mu}) + \mathbf{H}(\mathbf{X} - \boldsymbol{\mu}) + \mathbf{V}$, where $\mathbf{H} = df(\mathbf{X})/d\mathbf{X}|_{\mathbf{X}=\boldsymbol{\mu}}$ is the matrix of first partial derivatives (Jacobian). In other words, the nonlinear function $f(\mathbf{x})$ is approximated by the first order terms of a Taylor series expansion about the point $\boldsymbol{\mu}$. The approximation is a linear function of **X**, and so, except for the fact that **H** now depends on $\boldsymbol{\mu}$, the measurement step can be employed as before. The matrix **H** is used in calculating the Kalman gain, but not in calculating the shock, which is $\mathbf{Z} - f(\boldsymbol{\mu}) - \boldsymbol{\mu}_{\mathbf{V}}$ (the nonlinear function itself is used in calculating the shock, rather than the linear approximation). Similarly, if the movement model $\mathbf{X}' = g(\mathbf{X}) + \mathbf{W}$ includes a nonlinear function $g()$, then $\boldsymbol{\phi} = dg(\mathbf{X})/d\mathbf{X}|_{\mathbf{X}=\boldsymbol{\mu}}$ is the $n \times n$ Jacobian of g, and the movement step can be employed as before except that $\boldsymbol{\mu} \leftarrow g(\boldsymbol{\mu}) + \boldsymbol{\mu}_{\mathbf{W}}$. The matrix $\boldsymbol{\phi}$ is not used in updating $\boldsymbol{\mu}$, but is required in updating $\boldsymbol{\Sigma}$. In either case, whether the nonlinearity occurs in the measurement model or the movement model, the result is called an EKF.

Example 7. Triangulation

The problem of estimating the position of a stationary target from several inaccurate bearing measurements can be solved by employing an EKF with a vacuous movement step. Let the state be $(X, Y)^t$, with observers located at known points (x_i, y_i), i=1,2,3.

Let (D_i, θ_i) be the true range and bearing from observer *i* to the target, as pictured in Figure 4. Consider the measurement $Z_1 = \theta_1 + V_1$, where $\theta_1 = \arctan((Y_1 - y_1)/(X_1 - x_1))$. The measurement is a nonlinear function of the state. Since $d\theta_1 / dY_1 = \cos(\theta_1)/D_1$ and $d\theta_1 / dX_1 = -\sin(\theta_1)/D_1$, **H** is $[-\sin(\theta_1)\ \cos(\theta_1)]/D_1$, a 1×2 matrix. Since θ_1 and D_1 depend on the unknown state, **H** must in practice be evaluated by inserting the latest

estimates of θ_1 and D_1. Assuming $\mathbf{P}_0 = 0$, the precision matrix after one observation is (omitting trigonometric parentheses for brevity):

$$\mathbf{P}_1 = \mathbf{H}_1^t \mathbf{R}_1^{-1} \mathbf{H}_1 = \mathbf{R}_1^{-1} \begin{bmatrix} \sin^2\theta_1 & -\sin\theta_1\cos\theta_1 \\ -\sin\theta_1\cos\theta_1 & \cos^2\theta_1 \end{bmatrix} D_1^{-2} = \begin{bmatrix} \sin^2\theta_1/d_1 & -\sin\theta_1\cos\theta_1/d_1 \\ -\sin\theta_1\cos\theta_1/d_1 & \cos^2\theta_1/d_1 \end{bmatrix}$$

where $\mathbf{R}_1$ is the variance of the angular measurement in radians^2. The product $d_i = \mathbf{R}_i D_i^2$ is the variance of the i^{th} measurement expressed as a distance in the vicinity of the target, and plays a central role because this one factor incorporates all information about bearing accuracy and range.

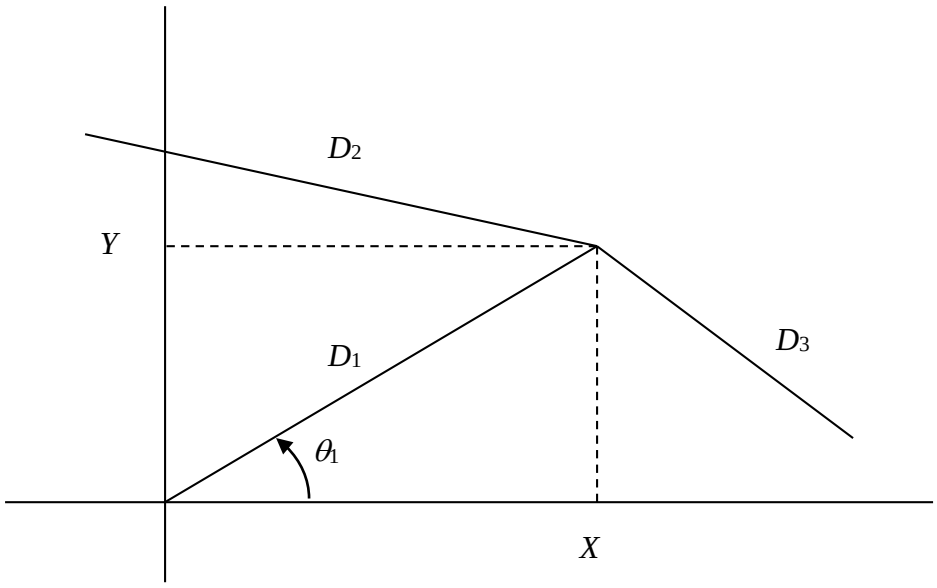

Figure 4. Three stations measure the bearing to a target located at (*X*,*Y*).

After n measurements, the precision matrix will be the sum

$$\mathbf{P}_n = \begin{bmatrix} \sum_{i=1}^{n} \frac{\sin^2\theta_i}{d_i} & -\sum_{i=1}^{n} \frac{\sin\theta_i\cos\theta_i}{d_i} \\ -\sum_{i=1}^{n} \frac{\sin\theta_i\cos\theta_i}{d_i} & \sum_{i=1}^{n} \frac{\cos^2\theta_i}{d_i} \end{bmatrix}.$$

We have not tracked through the state updates, but the shape and orientation of the bivariate normal distribution of the position of the target is already implied in **P**. The

standard way of representing this distribution graphically is to show the "two sigma ellipse" in a rotated coordinate system. This is an equiprobability contour that contains the state with probability 1 – exp(–2) = 0.865. This ellipse has its major axis at inclination I, a major diameter of length $4s_1$ and a minor diameter of length $4s_2$, where the inclination I and the two standard deviations s_1 and s_2 in the rotated coordinate system can be determined from the unrotated precision or covariance matrix shown above. For the sake of completeness, we record the formulas, letting $\mathbf{P}^{-1} = \Sigma = \begin{bmatrix} a & h \\ h & b \end{bmatrix}$:

$$s_1^2 = \left[\left(\frac{a+b}{2} \right) + \sqrt{\left(\frac{a-b}{2} \right)^2 + h^2} \right]$$

$$s_2^2 = \left[\left(\frac{a+b}{2} \right) - \sqrt{\left(\frac{a-b}{2} \right)^2 + h^2} \right]$$

$$I = \begin{cases} \frac{1}{2} \tan^{-1}\left(\frac{2h}{a-b} \right) & \text{if } a > b \\ \frac{1}{2} \tan^{-1}\left(\frac{2h}{a-b} \right) + \frac{\pi}{2} & \text{if } a < b \\ \frac{\pi}{4} \operatorname{sign}(h) & \text{if } a = b \end{cases}$$

In an EKF such as this it is not true that a collection of measurements can be processed as a single vector of measurements, as was done in the linear regression example. The measurements must be processed one at a time because **H** depends on previous measurements through the current state estimates. In an EKF, the order in which measurements are processed may be significant.

Dimensionless Shock

Since it is derived via Bayes Theorem, a (non-extended) Kalman filter makes estimates that are optimal in almost any reasonable sense of the word. This is not true of an EKF. Since the matrices **H** and/or ϕ depend on current state estimates and are used to

obtain revised state estimates, there is a potential for bad estimates to get worse, and complete loss of track is possible. A practical filter will recognize when this has happened, and take steps to correct the problem. Dimensionless shock plays a vital role.

The shock $\mathbf{S}_i$ used in making the measurement update at time i in an EKF is $\mathbf{Z}_i - f(\boldsymbol{\mu}_i(-)) - \boldsymbol{\mu}_\mathbf{V}$, the difference between what is actually measured ($\mathbf{Z}_i$) and the best prediction of $\mathbf{Z}_i$ based on all history previous to the i^{th} measurement ($f(\boldsymbol{\mu}_i(-)) + \boldsymbol{\mu}_\mathbf{V}$). One symptom of being out of control is that $\mathbf{S}_i$ is unusually large. A useful measure of tracking quality can be built on this observation, provided a scale can be found on which $\mathbf{S}_i$ can be judged to be "unusually large".

The measurement model is that $\mathbf{Z}_i = f(\mathbf{X}_i) + \mathbf{V}_i$. Approximating $f(\mathbf{X}_i)$ by $f(\boldsymbol{\mu}_i(-) + \mathbf{H}_i(\mathbf{X}_i - \boldsymbol{\mu}_i(-))$ leads to the approximation $S_i \cong \mathbf{H}_i(\mathbf{X}_i - \boldsymbol{\mu}_i(-)) + (\mathbf{V}_i - \boldsymbol{\mu}_V)$, a linear combination of the two independent random variables $\mathbf{X}_i$ and $\mathbf{V}_i$. The expected value of S_i in this approximation is 0, and the covariance is $\mathbf{H}_i\boldsymbol{\Sigma}_i(-)\mathbf{H}_i^t + \mathbf{R}$, which the reader may recognize as the denominator of the Kalman gain computation. The dimensionless shock is defined to be $DS_i \equiv \mathbf{S}_i^t\left(\mathbf{H}_i\boldsymbol{\Sigma}_i\left(-\right)\mathbf{H}_i^t + \mathbf{R}\right)^{-1}\mathbf{S}_i$, a Mahalanobis distance (appendix B). If $\mathbf{S}_i$ has m components, DS_i should be a scalar random variable that has a Chi-square distribution with m degrees of freedom. The mean of such a random variable is m. Thus, if DS_i becomes large compared to m, the likely explanation is that the filter has lost track. Since the covariance of $\mathbf{S}_i$ is already required to compute the Kalman gain, very little additional effort is required to also calculate DS_i. The computation and testing of this important statistic should therefore be automatic in an EKF. If $m = 3$, for example, one might declare loss of track if the dimensionless shock exceeds (say) 50. Once loss of track occurs, but not until then, *DS* will quickly increase to levels far exceeding that.

In addition to its use in recognizing when an EKF has lost track, the dimensionless shock *DS* also has a use in associating data with targets. Suppose that several targets are being tracked simultaneously, and that any of them might have caused a particular measurement. Other things being equal, the best target to associate with the measurement will be the one for which the dimensionless shock is smallest. Or, if *DS* is large for all targets, then the measurement might have been caused by some previously undiscovered

target. Associating data with targets is an important part of *data fusion*. See Bar-Shalom and Fortman [7] for an in-depth treatment, or search the world-wide-web for "Mahalanobis Distance".

A good way to observe the track loss phenomenon is through Monte Carlo simulation. Sheet "KFSim" of *Kalman.xlsm* does this for a scalar Kalman filter that is extended on account of the measurement. The user controls the nonlinear measurement function through VBA code, and can experiment to see exactly how stable the resulting filter is. In addition to dimensionless shock, the sheet also computes and graphs two other diagnostic Mahalanobis distances, each one a dimensionless version of the distance from $\mathbf{X}$ to $\boldsymbol{\mu}$. While these distances are interesting in an experimental setting, they could not be used in an actual EKF because $\mathbf{X}$ is never actually known. See exercise 14.

Example 8 (Doppler shift)

Active sensors can often measure a Doppler shift that is equivalent to observing the rate at which range is changing. Like any measurement, the Doppler shift can be used in improving state estimates. This example gives details for a filter whose state is given in polar coordinates.

Suppose $\mathbf{X} = (R, \theta, V, \Psi)^t$, with the four components of state being range, bearing, speed and course; i.e., the state is position and velocity in polar coordinates. In terms of these state variables, the range rate is $V\cos(\theta - \psi)$. Suppose that noisy measurements of range and bearing are available, in addition to range rate, so that each active pulse results in a three-dimensional measurement: $\mathbf{Z} = (R, \theta, V\cos(\theta - \psi))^t$ + (measurement noise). Since the cosine is a nonlinear function, an extended Kalman filter is required. Letting $S = \sin(\theta - \Psi)$ and $C = \cos(\theta - \Psi)$, and taking derivatives as necessary in the third row,

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -VS & C & VS \end{bmatrix}.$$

Since the quantities *V, S,* and *C* depend on the best current estimate of the state vector, the filter is an EKF. **H** will be a different matrix for each measurement, and can no longer be calculated beforehand, since its evaluation must await a state estimate.

For the movement model, suppose that the target's speed and course fluctuate according to independent O-U processes. The time interval between measurements is assumed to be Δ=8 seconds. The stochastic process representing the target's course is the one described earlier in the O-U section, so its two parameters are c_4=0.92 and Q_{44}=(0.028radians)^2 (angular measurements will be in radians in this example, rather than degrees). The target's speed process parameters are taken to be c_3=0.95 and Q_{33}=(0.1)^2, but we wish to have speed fluctuate around the positive level v=5m/s, rather than 0. To accomplish this, we add a constant $\mu_3=v(1-c_3)$ to the update equation for speed. We now have a target that basically moves East at speed v, with fluctuations.

The target's position is assumed to be the time integral of its velocity. Since range rate is $V\cos(\theta-\psi)$, the updated range is $R+\Delta\, V\cos(\theta-\psi)$. By taking derivatives of this expression with respect to all four state variables, we obtain the first row of the $\boldsymbol{\phi}$ matrix. The updated bearing is $\theta-\Delta\, V\sin(\theta-\psi)/R$, and the second row of $\boldsymbol{\phi}$ is again obtained by taking derivatives. Letting *C* and *S* be defined as above, the final result is

$$\boldsymbol{\phi}=\begin{bmatrix} 1 & -VS\Delta & C\Delta & VS\Delta \\ VS\Delta/R^2 & 1-VC\Delta/R & -S\Delta/R & VC\Delta/R \\ 0 & 0 & c_3 & 0 \\ 0 & 0 & 0 & c_4 \end{bmatrix}, \text{ and } \mathbf{Q}=\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & Q_{33} & 0 \\ 0 & 0 & 0 & Q_{44} \end{bmatrix}.$$

The first two rows of $\boldsymbol{\phi}$ will require current estimates of the state variables for their evaluation, so the movement model is also extended. The movement model would *not* be extended if Cartesian coordinates had been used (recall MTST) — the choice of coordinate systems is significant in Kalman filtering. We persist with polar coordinates here because of the simplified measurement model implied by that choice.

A reader with access to MATLAB and the script *KDopp.m* may wish to experiment (see exercise 12). The script performs a Monte Carlo simulation of the filter in action, much like page "KFSim" of *Kalman.xlsm* for scalar measurements. MATLAB is easier to use and understand in this case because of its algebraic treatment of matrices.

Further Reading

Kalman filtering was invented by and for the most part has been used by electrical engineers, even though there is nothing especially electrical about it. This explains the name, and also the fact that a great deal of the literature is in the IEEE Transactions series. Kailath [3] provides an excellent historically based review with 390 (!) references. He traces the underlying ideas back to the work of Kolmogorov, Krein, and Weiner in the 1940s, and even before. Kalman's essential contribution was to recognize that the required computations can be done recursively, the seminal paper being [4].

References [1, 2, 5] are textbooks. The movement and measurement steps can be justified as being optimal in a least squares sense even without the assumptions of normality made here, and that approach is in fact the one more commonly pursued. There is also a continuous-time KF that is obtainable as a limiting form of the discrete filter; it involves ordinary differential equations for $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ instead of the replacement ($\leftarrow$) operation of Figure 1.

Most applications of KFs are to tracking problems where the state is a more or less elaborate description of the position of something. There are several examples in Gelb [1] and Titus [6].

Appendix A(Proof of the second basic fact)

Theorem: Let $\mathbf{Z} = \mathbf{HX} + \mathbf{V}$, where $\mathbf{X}$ and $\mathbf{V}$ are independent, multivariate normal random variables for which $E(\mathbf{V}) = 0$, $\mathrm{Cov}(\mathbf{V}) = \mathbf{R}$, $E(\mathbf{X}) = \boldsymbol{\mu}$, and $\mathrm{Cov}(\mathbf{X}) = \boldsymbol{\Sigma}$. It is assumed that the inverses of $\mathbf{R}$ and $\boldsymbol{\Sigma}$ exist. Then, conditional on $\mathbf{Z}$ being given, $\mathbf{X}$ is multivariate normal with mean $\hat{\boldsymbol{\mu}} \equiv E(\mathbf{X}|\mathbf{Z}) = \boldsymbol{\mu} + \mathbf{K}(\mathbf{Z}-\mathbf{H}\boldsymbol{\mu})$ and covariance $\hat{\boldsymbol{\Sigma}} \equiv \mathrm{Cov}(\mathbf{X}|\mathbf{Z}) = (\mathbf{I}-\mathbf{KH})\boldsymbol{\Sigma}$, where $\mathbf{K} \equiv \boldsymbol{\Sigma}\mathbf{H}^t(\mathbf{H}\boldsymbol{\Sigma}\mathbf{H}^t + \mathbf{R})^{-1}$.

Proof: The joint density function of $\mathbf{Z}$ and $\mathbf{X}$ is (const.) $\exp(-q/2)$, where $q = (\mathbf{x}-\boldsymbol{\mu})^t\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}) + (\mathbf{z}-\mathbf{Hx})^t\mathbf{R}^{-1}(\mathbf{z}-\mathbf{Hx})$. By simple expansion by terms, one can show that $q = (\mathbf{x}-\hat{\boldsymbol{\mu}})^t\hat{\boldsymbol{\Sigma}}^{-1}(\mathbf{x}-\hat{\boldsymbol{\mu}}) + (\text{const.})$, where (const.) does not depend on $\mathbf{x}$, $\hat{\boldsymbol{\Sigma}}^{-1} = \boldsymbol{\Sigma}^{-1} + \mathbf{H}^t\mathbf{R}^{-1}\mathbf{H}$, and $\hat{\boldsymbol{\mu}} = \boldsymbol{\mu} + \hat{\boldsymbol{\Sigma}}\mathbf{H}^t\mathbf{R}^{-1}(\mathbf{z}-\mathbf{H}\boldsymbol{\mu})$. This verifies that $E(\mathbf{X}|\mathbf{Z})$ and $\mathrm{Cov}(\mathbf{X}|\mathbf{Z})$ are $\hat{\boldsymbol{\mu}}$ and $\hat{\boldsymbol{\Sigma}}$, respectively, but in most cases formulas that do not require inversion of $\boldsymbol{\Sigma}$ will be more convenient. To obtain such, we use the fact that $(\mathbf{I} + \mathbf{UH})^{-1}\mathbf{U} = \mathbf{U}(\mathbf{HU} + \mathbf{I})^{-1}$ for any matrix $\mathbf{U}$—note that the formula connects two matrix inversions of different dimension. In the following, we will define $\mathbf{U} = \hat{\boldsymbol{\Sigma}}\mathbf{H}^t\mathbf{R}^{-1}$. We have

$$\begin{aligned}\hat{\boldsymbol{\Sigma}}\mathbf{H}^t\mathbf{R}^{-1} &= (\boldsymbol{\Sigma}^{-1} + \mathbf{H}^t\mathbf{R}^{-1}\mathbf{H})^{-1}\mathbf{H}^t\mathbf{R}^{-1} \\ &= (\mathbf{I} + \mathbf{UH})^{-1}\mathbf{U} = \mathbf{U}(\mathbf{HU} + \mathbf{I})^{-1} \\ &= \boldsymbol{\Sigma}\mathbf{H}^t(\mathbf{H}\boldsymbol{\Sigma}\mathbf{H}^t + \mathbf{R})^{-1} \\ &\equiv \mathbf{K},\end{aligned}$$

thus showing that the formula given for $\hat{\boldsymbol{\mu}}$ in the statement of the theorem is correct. To show that the formula for $\hat{\boldsymbol{\Sigma}}$ is correct, we will use the fact that $(\mathbf{I} + \mathbf{UH})^{-1} = \mathbf{I} - (\mathbf{I} + \mathbf{UH})^{-1}\mathbf{UH}$. Since we showed above that $(\mathbf{I} + \mathbf{UH})^{-1}\mathbf{U} = \mathbf{K}$, it follows by substitution that $(\mathbf{I} + \mathbf{UH})^{-1} = \mathbf{I} - \mathbf{KH}$. We therefore have $\hat{\boldsymbol{\Sigma}} = (\boldsymbol{\Sigma}^{-1} + \mathbf{H}^t\mathbf{R}^{-1}\mathbf{H})^{-1} = (\mathbf{I} + \mathbf{UH})^{-1}\boldsymbol{\Sigma} = (\mathbf{I} - \mathbf{KH})\boldsymbol{\Sigma}$, showing that the expression for $\hat{\boldsymbol{\Sigma}}$ is also correct.

QED

Appendix B (Covariance matrices)

If a random column vector $\mathbf{X}$ has m rows and mean $\boldsymbol{\mu}$, then $\boldsymbol{\Sigma}=\text{Cov}(\mathbf{X})$ is by definition the $m\times m$ matrix $\text{E}((\mathbf{X}-\boldsymbol{\mu})(\mathbf{X}-\boldsymbol{\mu})^{\text{t}})$. The entry in the i^{th} row and j^{th} column is $\text{E}((\mathbf{X}_i-\boldsymbol{\mu}_i)(\mathbf{X}_j-\boldsymbol{\mu}_j))$, so two immediate properties of such matrices are that the entries on the main diagonal must be nonnegative (since squares of scalars are nonnegative), and that the matrix must be symmetric about the main diagonal (since scalar arithmetic is commutative). Thus $\begin{bmatrix}1 & 2\\ 3 & 4\end{bmatrix}$ is not a covariance matrix, since $3\neq 2$. These elementary properties of covariance matrices can be of use in debugging Kalman filters, since virtually any mistake in applying one of the update formulas will result in a clearly faulty "covariance matrix".

Two useful facts about covariance matrices are

1. If $\mathbf{Y}=\mathbf{C}\mathbf{X}+\mathbf{D}$, where $\mathbf{X}$ is random and $\mathbf{C}$ and $\mathbf{D}$ are compatibly dimensioned constant matrices, then $\text{Cov}(\mathbf{Y})=\mathbf{C}\text{Cov}(\mathbf{X})\mathbf{C}^{\text{t}}$. Note that D is not involved.

2. If $\mathbf{X}$ and $\mathbf{Y}$ are independent, then $\text{Cov}(\mathbf{X}+\mathbf{Y})=\text{Cov}(\mathbf{X})+\text{Cov}(\mathbf{Y})$.

These two facts can be combined. In a Kalman filter, the quantity $\boldsymbol{\phi}\boldsymbol{\Sigma}\boldsymbol{\phi}^{\text{t}}+\mathbf{Q}$ can be seen to be $\text{Cov}(\boldsymbol{\phi}\mathbf{X}+\mathbf{W})$, and $\mathbf{H}\boldsymbol{\Sigma}\mathbf{H}^{\text{t}}+\mathbf{R}$ can be seen to be the $\text{Cov}(\mathbf{H}\mathbf{X}+\mathbf{V})$.

Covariance matrices do not necessarily have inverses. A counterexample would be where Y is a standard normal random variable, and $\mathbf{X}=\begin{bmatrix}Y\\ Y\end{bmatrix}$. In this case $\text{Cov}(\mathbf{X})=\begin{bmatrix}1 & 1\\ 1 & 1\end{bmatrix}$, which does not have an inverse. The problem here is that the components of $\mathbf{X}$ have perfect correlation, which is usually not the case. As long as the inverse exists, we have a third useful fact about covariance matrices:

3. $(\mathbf{X}-\boldsymbol{\mu})\boldsymbol{\Sigma}^{-1}(\mathbf{X}-\boldsymbol{\mu})^{\text{t}}$ is a chi-square random variable with m degrees of freedom.

In general, this quadratic form is the square of the *Mahalanobis distance* of $\mathbf{X}$ from its mean. In Kalman filtering, the dimensionless shock *DS* is such a distance of $\mathbf{Z}$ from Kalman's expectation. All three facts generalize similar facts for scalar random variables. In the case of fact 3, $(X-\mu)/\sqrt{\Sigma}$ is a standard normal random variable, and its square is a chi-square random variable with one degree of freedom.

References

[1] Gelb, A. (1974), *Applied Optimal Estimation*, MIT Press.

[2] Jazwinski, A. H. (1970), *Stochastic Processes and Filtering Theory*, Academic Press.

[3] Kailath, T. (1974), "A view of three decades of linear filtering theory," *IEEE Trans. Info. Theory*, IT-20, 146-180.

[4] Kalman, R. E. (1960), "A new approach to linear filtering and prediction problems," *Trans. ASME, J. Basic Engineer*, 82D, 34-45.

[5] Meditch, J. S. (1969), *Stochastic Optimal Linear Estimation and Control*, McGraw-Hill.

[6] Titus, H. (1977), *Advances in Passive Target Tracking*, NPS report NPS62Ts-77071, Naval Postgraduate School.

[7] Bar-Shalom, Y. and Fortman, T. (1988), *Tracking and Data Association*, Academic Press.

[8] Bar-Shalom, Y. and Li, X. (1995), *Multitarget-Multisensor Tracking: Principles and Techniques*, Academic Press.

Exercises (solutions follow):

1. Suppose $\mathbf{A} = \begin{bmatrix} 1 & 2 \end{bmatrix}$, $\mathbf{B} = \begin{bmatrix} 4 & 1 \\ 1 & 2 \end{bmatrix}$, and $\mathbf{C} = \begin{bmatrix} -1 & 2 & 0 \\ 1 & 1 & 1 \end{bmatrix}$. Then

   a) What is **AB**, the matrix product of **A** and **B**?

   b) What is $\mathbf{C}^t$, the transpose of **C**?

   c) There are nine possible two-factor matrix products like **AB** or **AA**. Which ones are well defined?

   d) Which of the three matrices have inverses, and what are they?

   e) There are six possible two-factor matrix sums like **A+B**. Which ones are well defined?

   f) Is matrix multiplication commutative; that is, does **AB**=**BA** in general?

   g) Is matrix multiplication associative; that is, does (**AB**)**C**=**A**(**BC**) in general?

2. Suppose $\boldsymbol{\mu}(+) = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$, $\boldsymbol{\Sigma}(+) = \begin{bmatrix} 4 & 2 \\ 2 & 1 \end{bmatrix}$, $\boldsymbol{\mu}_{\mathbf{W}} = \begin{bmatrix} 5 \\ 5 \end{bmatrix}$, $\phi = \begin{bmatrix} 0 & 1 \\ 2 & 2 \end{bmatrix}$, and $\mathbf{Q} = \begin{bmatrix} 1 & 0 \\ 0 & 3 \end{bmatrix}$.
Apply the movement update to find $\boldsymbol{\mu}(-)$ and $\boldsymbol{\Sigma}(-)$ right before the next measurement.

3. If $\boldsymbol{\mu}(-) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, $\boldsymbol{\Sigma}(-) = \begin{bmatrix} 4 & 2 \\ 2 & 1 \end{bmatrix}$, $\boldsymbol{\mu}_{\mathbf{V}} = \begin{bmatrix} 3 \end{bmatrix}$, $\mathbf{H} = \begin{bmatrix} 1 & 1 \end{bmatrix}$, and $\mathbf{R} = \begin{bmatrix} 1 \end{bmatrix}$, and if the measurement turns out to be **Z**=5, apply the measurement update to find $\boldsymbol{\mu}(+)$ and $\boldsymbol{\Sigma}(+)$.

4. (formulation) Jack Kalman has recently been employed by the torpedo tracking department at an underwater testing facility. The facility uses five hydrophone arrays, each of which reports the torpedo's position in 3-D every time the torpedo pings (which it does every Δ=1 second), provided the torpedo is within range of the array. Experience has shown that the variance of each component of each 3-D measurement is 4 square meters, independently of the other components.

Jack is not even distantly related to the famous R. E. Kalman, but this has not prevented his boss from asking him to develop a Kalman filter to process the measurements and keep track of the torpedoes as they move from array to array. Jack feels that velocity as well as position should be part of the torpedo's state vector, so he currently has a state vector with six components: (X_1, X_2, X_3) for

position in meters and (U_1, U_2, U_3) for velocity in meters/second. He has simulated the process $U_1(t+\Delta)=.9U_1(t)+N_1(t)$, where $N_1(t)\sim N(0, 10 \text{ meters}^2/\text{second}^2)$, and found that the results agree closely with what torpedoes actually do when tested on the range. The second velocity component behaves similarly, but the third component U_3 represents vertical velocity, a relatively slowly changing quantity. Jack has found that the motion model $U_3(t+\Delta)=.95U_3(t)+N_3(t)$, where $N_3(t)\sim N(0, 3 \text{ meters}^2/\text{second}^2)$, comes reasonably close to behaving like a torpedo's vertical velocity.

But now Jack needs your help:

a) What are $\boldsymbol{\phi}$ and **Q**? Hint: $\boldsymbol{\phi}$ has as many rows as there are components to the state, and each row expresses the deterministic part of how its component is updated. Information about variance goes in the covariance matrix **Q**.

b) Assuming that only array number 1 is within range, each ping will produce three numbers, a measurement of the torpedo's 3-D position. What are **H** and **R**? Hint: **H** has one row for each component of the measurement, and each row expresses the deterministic part of how that component depends on the state. Information about variance goes in covariance matrix **R**.

c) If arrays 1 and 2 are both within range, each ping will produce six numbers, three from each array. Now what are **H** and **R**?

d) Suppose that the arrays themselves are imperfectly located, with the initial location error variance being known. Kalman filtering is still applicable, but the definition of "state" must change. What is the new definition of state, and what will the new dimensions of $\boldsymbol{\phi}$ and **Q** be? You do not need to write out $\boldsymbol{\phi}$ and **Q**.

5. Confirm that the second approach in example 4 produces the $\boldsymbol{\Sigma}(+)$ matrix that is stated there.

6. Using the data of example 3, try processing two measurements in a batch, and then processing the remaining measurement. Regardless of which two you select to process first, you should get the same final result as in example 3. If you attempt to process only one measurement first, you will find that the first calculated precision matrix doesn't have an inverse. This is Kalman's way of protesting the task you have set him, namely the task of estimating two numbers from only one measurement when the prior estimate is valueless.

7. Suppose that $Z=f(X)+V$, where $f(x)=\sin(x)$, so that an extended Kalman Filter is required to process a measurement. Specifically, suppose that $\mu(-)=1$, $\Sigma(-)=0.5$, $\mu_V=0.1$, $R=0.03$, and $Z=0.6$. What are $\mu(+)$ and $\Sigma(+)$?

8. (advanced continuation of exercise 7) The best estimate of X after the measurement has been processed in exercise 7 is 0.476, but processing that measurement has linearized the function $\sin(x)$ about the point $\mu(-)=1$. If we believe that X is approximately 0.476, rather than 1, then why not linearize about 0.476? And if changing the linearization point changes K, which in turn makes our estimate of X something else again, then we can continue to re-linearize about out current belief until convergence occurs. Doing that is called an *iterated* EKF. After convergence, what are $\mu(+)$ and $\Sigma(+)$? Notationally, let $\mu_k(+)$ and $\Sigma_k(+)$ be the k^{th} iteration, with $\mu_1(+)=0.476$ and $\Sigma_1(+)=0.085$.

Hint: Ten iterations suffice, and can be accomplished in ten lines of a spreadsheet.

9. Let $\boldsymbol{\mu}_1(-)=\begin{bmatrix}0\\0\end{bmatrix}$, $\boldsymbol{\Sigma}_1(-)=\begin{bmatrix}10 & 0\\0 & 100\end{bmatrix}$, $\Delta=1$, $q=1$, and other inputs as in example 2. Calculate the Kalman gains and both covariance matrices up to and including $\boldsymbol{\Sigma}_3(+)$. It is suggested that you do this by hand, with accuracy to four decimal places. Accuracy is necessary because the formula for $\boldsymbol{\Sigma}(+)$ involves a numerically unfortunate subtraction of nearly equal quantities.

10. The same as exercise 9, except make **H**=[0 1] instead of [1 0]. In exercise 9, Kalman estimates velocity from multiple estimates of position. In exercise 10, he estimates position from multiple estimates of velocity. Which is the easier task for him? To check hand calculations (or avoid them), change **H** on sheet "KF2Sim" of *Kalman.xlsm*.

11. The same as exercise 9, except test the robustness of a Kalman Filter to bad assumptions. Add a cell to sheet "KF2Sim" of *Kalman.xlsm* to calculate the statistic $(X_2-\mu_2(+))\text{^}2$, the squared error of the second component of **X**, right after the third measurement is processed. On the average, this quantity should be $\Sigma_{22}(+)$, the variance computed by Kalman right after the third measurement. Verify that this is so by making several replications of the new cell. Try inserting a bad assumption to see what effect it has. For example, you might set **R** to 1 and sqrt(**R**) to 4. Since sqrt(**R**) is used in producing the actual measurements, this will have the effect of making the measurements less accurate than they are supposed to be.

12. This exercise requires access to MATLAB and the script *KDopp.m*. That script performs a Monte Carlo simulation of the Doppler filter described in example 8. Read the script to see the implementation details, and run it to see how well the filter performs its tracking task. You should see that the filter's estimates are usually close to the truth, and that the dimensionless shock is never alarmingly large. Next, make some robustness experiments and report on one of them. Possibilities include

a. Increase one of the input variances. The filter is extended, so there will be a limit to how much you can do this. If you go too far, you may see Kalman lose track.

b. Deliberately lie to Kalman. The MATLAB code has a mechanism for having the target make a sudden, one radian turn that is not in accord with the motion model. There are other possibilities.

13. Page "ArtillerySim" of *Kalman.xlsm* is a side-by-side simulation of two artillery registration procedures. One is the Kalman procedure described in example 5. The other (3CAL) simply averages the first three miss distances, adjusts the aimpoint once, and then "fires for effect" at the adjusted aimpoint. By repeating the simulation, you will see that there are instances where 3CAL beats Kalman in hitting the target, as well as vice versa. Show that the Kalman procedure is nonetheless superior in the sense that it has a higher kill probability on every shot.

14. Page "KFSim" of *Kalman.xlsm* simulates the operation of a scalar EKF, showing dimensionless shock and two other Mahalanobis distances as diagnostic tools. The measurement is Z=m_func(X)+V, where m_func(x) and its derivative d_func(x) are under control of the user through VBA code. Basically X increases erratically from 1 to 10 while Kalman tries to keep track of it. The exercises below require changing m_func(x). Don't forget to also change d_func(x) to be the derivative. To be definite, define "loss of track" to be where any Mahalanobis distance exceeds 1000.

a. Make m_func(x) a linear function, in which case d_func(x) is a constant and the filter is not extended. You should never see loss of track in this case. The dimensionless shock behaves like a Chi-square random variable with one degree of freedom. See what happens if you insert one bad measurement or unexpected movement.

b. Make m_func(x) a slightly nonlinear function, perhaps exp(x/10). You should see a filter that behaves as in part a).

c. Make m_func(x) a highly nonlinear function, perhaps sin(x), being careful not to use a function like 1/x where division by 0 is possible. Experiment with the parameters of the motion model until loss of track happens on some replications, but not on others.

Exercise Solutions

1.

a) $\mathbf{AB}=[6\ \ 5]$.

b) $\mathbf{C}^t=\begin{bmatrix}1 & -1\\ 1 & 2\\ 1 & 0\end{bmatrix}$. The transpose is defined for every matrix.

c) **AB**, **BC**, **BB**, and **AC** are well defined. **AA, CC**, **BA**, **CB**, and **CA** are not defined because the number of columns in the first factor is not the same as the number of rows in the second.

d) Only **B** has an inverse. Except for square matrices, matrices do not have inverses.

e) None of the sums are defined. Only matrices with identical dimensions can be summed.

f) No, matrix multiplication is not commutative in general. You have to get used to this when doing matrix algebra, since the more familiar scalar multiplication is commutative.

g) Yes, matrix multiplication is associative. Therefore we usually skip the parentheses.

If matrix arithmetic is a completely new subject for you, it would be best to find a textbook and learn the rules. The actual algebra can be efficiently performed in MATLAB or (with more trouble) Excel.

2. $\boldsymbol{\mu}(-)=\begin{bmatrix}7\\ 11\end{bmatrix}, \boldsymbol{\Sigma}(-)=\begin{bmatrix}2 & 6\\ 6 & 39\end{bmatrix}$.

3. $\boldsymbol{\mu}(+)=\begin{bmatrix}1.2\\ 0.6\end{bmatrix}, \boldsymbol{\Sigma}(-)=\begin{bmatrix}0.4 & 0.2\\ 0.2 & 0.1\end{bmatrix}$.

4.

a) $\boldsymbol{\phi}$ is a 6×6 matrix with nine nonzero entries, three of which are Δ. **Q** is a 6×6 covariance matrix with three positive entries, all on the main diagonal.

b) **H** is a 3×6 matrix with three positive entries, all of which are 1. **R** is a 3×3 covariance matrix with three nonzero entries, all on the main diagonal.

c) **H** has six rows, with the top 3 and bottom 3 both being the same as in part b. **R** is a 6×6 covariance matrix. Since the measurements are independent, all of the off-diagonal entries in **R** are 0.

d) Putting all of the array locations into the state vector requires 15 more components, so the state vector has 21 components. $\boldsymbol{\phi}$ and **Q** are both 21×21.

5. The answer is given in the text.

6. The answer is given in the text.

7. 0.476 and 0.085.

8. 0.616 and 0.041. A spreadsheet implementation is on page "IEKF" of *Kalman.xlsm*. Do not conclude that the difference between an EKF and an iterated EKF is always as large as in this example, which has been rigged to make the difference noticeable

9. The answer is on sheet "KF2Sim" of *Kalman.xlsm*.

10. Kalman is essentially unable to keep track of position based only on noisy velocity measurements, as should be clear from inspecting the covariance matrices. It would be even harder to keep track of position through acceleration measurements. This is why submarines occasionally come to the surface for navigation reasons.

11. For the proposed experiment, you should find an experimental variance that is larger than $\Sigma_{22}(+)$. Kalman is optimistic about the accuracy of his state estimates when the measurements are more noisy than he thinks they are.

12. There are many solutions possible.

13. With parameters as distributed, Kalman's kill probability on the tenth shot exceeds 3CAL's by about 0.04. This is based on 3000 replications.