16-642 The Kalman Filter

George Kantor

26 October 2020

1 Discrete Time State Space Systems

State space formulation also works for discrete time systems. The only difference is that the ODE is replaced by a difference equation.

We start by defining a state vector $x \in \mathbb{R}^n$, i.e.,

$$x[k] = \begin{bmatrix} x_1[k] \\ x_2[k] \\ \vdots \\ x_n[k] \end{bmatrix}.$$

We also define an input vector $u \in \mathbb{R}^m$ and output vector $y \in \mathbb{R}^p$:

$$u[k] = \begin{bmatrix} u_1[k] \\ u_2[k] \\ \vdots \\ u_m[k] \end{bmatrix}, \qquad y[k] = \begin{bmatrix} y_1[k] \\ y_2[k] \\ \vdots \\ y_p[k] \end{bmatrix}.$$

A linear discrete time state space system is represented by the following equations: And her

$$x[k+1] = Fx[k] + Gu[k]$$

$$y[k] = Hx[k].$$

where the matrix $F \in \mathbb{R}^{n \times n}$, $G \in \mathbb{R}^{n \times m}$, and $H \in \mathbb{R}^{p \times n}$. The first equation is called the state equation and the second equation is called the output equation.

1.1 Unforced System

As we did with continuous time systems, when we set $u = 0$ we can explicitly write the solution to the state equation. If

$$x[k+1] = Fx[k], \qquad x[0] = x_0,$$

then we find the solution:

$$x[1] = Fx_0$$

$$x[2] = Fx[1] = F^2 x_0$$

$$x[3] = Fx[2] = F^3 x_0$$

So generally,

$$x[k] = F^k x_0$$

We can use the eigenvalues of F to tell whether or not the system is stable (they are just like the poles!):

- $|\lambda_i| < 1 \quad \forall i \quad \Leftrightarrow$ the equilibrium point $x_e = 0$ is asymptotically stable (i.e., $\|x[k]\| \to 0$ as $k \to \infty$.)
- $|\lambda_i| > 1$ for some $i \quad \Leftrightarrow$ the equilibrium point $x_e = 0$ is unstable (i.e., $\|x[k]\| \to \infty$ as $k \to \infty$).
- if $|\lambda_i| \leq 1 \quad \forall i$ with the $|\lambda_i| = 1$ for some i then the result is analOgous to the continuous case with poles on the imaginary axis. If the poles are not repeated then the system is stable, and there is a stronger result that includes repeated poles, but is beyond the scope of this class.

1.2 Controllability and State Feedback

The concept of controllability of discrete time state space systems is analogous to that of continuous time state space systems. The test to see whether a pair (A, B) is controllable is exactly the same as for the continuous case.

The concept of using state feedback is also the same. Using the state feedback law $u[k] = -Kx[k]$ yields the closed loop system

$$x[k+1] = (F - GK)x[k]$$

The pole placement theorem holds, so that you can find a feedback matrix K to make the closed loop system matrix $A - BK$ have any allowable set of eigenvalues.

You can solve the pole placement problem using the MATLAB `place` or `acker` commands.

There is also a discrete time version of LQR. It is just like the continuous time version, except the integrals in the cost function are replaced by summations. MATLAB has solution to the discrete time infinite time horizon problem implemented with the function `dlqr`.

1.3 Observability and Observers

This also is very similar to the continuous time case. Specifically, the definition of observability and the observability test are exactly the same. A discrete time observer is of the form

$$\hat{x}[k+1] = F\hat{x}[k] + Gu[k] + K_o(y[k] - H\hat{x}[k])$$

Analysis of the dynamics of the error signal $e[k] = x[k] - \hat{x}[k]$ yields

$$e[k+1] = (F - K_o H)e[k]$$

So observer design boils down to a pole placement problem.

2 Gaussian Random Variables

A *random variable* is a variable whose value depends on the outcome of some random event. There are discrete random variables (e.g., coin flip or die roll) and continuous random variables, which are often used to model noise. Here we care exclusively about continuous random variables

A random variable has a *probability distribution* which defines the likelihoods of the possible values that the variable can take. For continuous random variables, the probability distribution can be defined by a *probability density function (pdf)*. For a scalar random variable, a pdf is a function $p : \mathbb{R} \to \mathbb{R}$ with the property that

$$P(x \in [c_0, c_1]) = \int_{c_0}^{c_1} p(z)dz$$

Some comments:

- big P is the probability function, in English $P(x \in [c_0, c_1])$ is pronounced "the probability that x is in the interval $[c_0, c_1]$.
- $\int_{-\infty}^{\infty} p(z)dz$ must be equal to 1.
- $P(x = c_0)$ is always zero.
- we call $p(c_0)$ the "likelihood" of c_0, it is a *relative* measure of how likely it is that x will be c_0 relative to other possible outcomes.

For $x \in \mathbb{R}$, the Gaussian pdf (also known as the "normal pdf") is

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

The *expectation* (or mean) of a random variable x with pdf $p(x)$ is

$$E(x) = \int_{-\infty}^{\infty} zp(z)dz$$

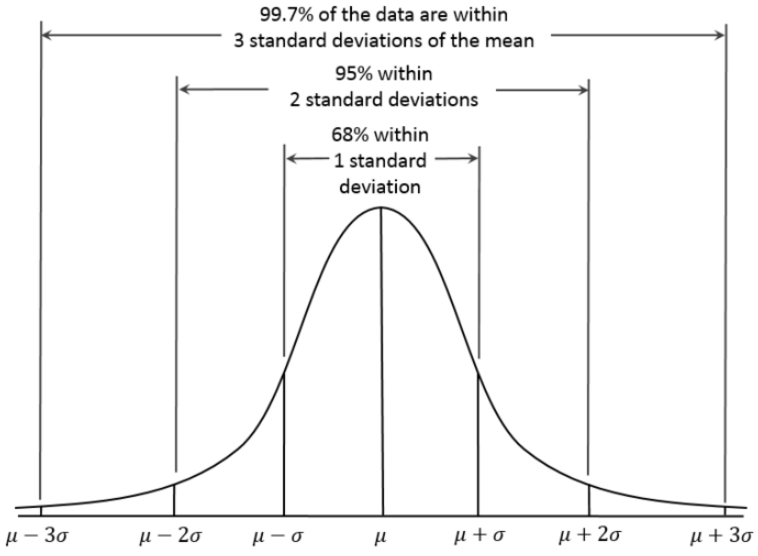
For the Gaussian, you can do the math to show that $E(x) = \mu$.

The *variance* of a random variable x with pdf $p(x)$ is

$$\text{Var}(x) = E((x - \mu)^2)$$

For the Gaussian, you can do the math to show that $\text{Var}(x) = \sigma^2$. We call σ the standard deviation. Sometimes we use the shorthand $\mathcal{N}(\mu, \sigma^2)$ to denote a Gaussian (aka "normal") distribution with mean μ and variance σ^2.

So the Gaussian distribution is completely described by its mean and variance. The mean defines the peak of the pdf, or the "most likely" value of x, the variance defines how "spread out" the curve is:



For higher dimensional random variables (sometimes called random vectors) $x \in \mathbb{R}^n$, the story is very similar. The distribution is defined by a pdf, the only difference is that you need to integrate over multiple variables in order to get the probability. For example, if $x \in \mathbb{R}^2$, the a pdf $p : \mathbb{R}^2 \to \mathbb{R}$ so that

$$P(x_1 \in [c_0, c_1], x_2 \in [d_0, d_1]) = \int_{c_0}^{c_1} \int_{d_0}^{d_1} p(z) dz_2 dz_1.$$

More generally, if $x \in real^n$ and $C \subset \mathbb{R}^n$, then

$$P(x \in C) = \int_{x \in C} p(z) dz.$$

The pdf we care about for Kalman filters is the Multivariate Gaussian pdf with $x \in \mathbb{R}^n$:

$$p(x) = \frac{1}{\sqrt{(2\pi)^n \det(\Sigma)}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)}$$

The mean vector $E(x) = \mu \in \mathbb{R}^n$. The covariance matrix $\text{Var}(x) = E((x-\mu)(x-\mu)^T) = \Sigma \in \mathbb{R}^{n \times n}$ is square, symmetric, and positive definite.

The multivariate Gaussian is a bell curve in n dimensions. The mean μ is the peak of the bell, the covariance matrix Σ defines the shape of the bell.

To get some intuition about Σ, consider the case where $n = 2$. Here

$$\mu = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}.$$

By definition,

$$\text{Var}(x) = E\left((x-\mu)(x-\mu)^T\right) = E\left(\begin{bmatrix} x_1 - \mu_1 \\ x_2 - \mu_2 \end{bmatrix} \begin{bmatrix} x_1 - \mu_1 & x_2 - \mu_2 \end{bmatrix}\right)$$

$$= E\left(\begin{bmatrix} (x_1 - \mu_1)^2 & (x_1 - \mu_1)(x_2 - \mu_2) \\ (x_2 - \mu_2)(x_1 - \mu_1) & (x_2 - \mu_2)^2 \end{bmatrix}\right)$$

$$= \begin{bmatrix} E\left((x_1 - \mu_1)^2\right) & E\left((x_1 - \mu_1)(x_2 - \mu_2)\right) \\ E\left((x_2 - \mu_2)(x_1 - \mu_1)\right) & E\left((x_2 - \mu_2)^2\right) \end{bmatrix}$$

The diagonal terms $E((x_i - \mu_i)^2)$ are just the variances of the individual components of x, e.g.,

$$E\left((x_1 - \mu_1)^2\right) = \text{Var}(x_1) = \sigma_1^2.$$

A couple of things to note about the off-diagonal terms:

- Remember that Σ is diagonal, which can be seen by observing that

$$E\left((x_1 - \mu_1)(x_2 - \mu_2)\right) = E\left((x_2 - \mu_2)(x_1 - \mu_1)\right)$$

- We denote $\sigma_{ij} = E\left((x_i - \mu_i)(x_j - \mu_j)\right)$
- σ_{ij} is called the *covariance* between x_i and x_j
- $\sigma_{ij} = 0$ if and only if x_i and x_j are statistically independent.

An intuitive way to think about statistical independence is this: suppose the $x \in \mathbb{R}^2$ is drawn from a Gaussian distribution. Your job is to estimate the value of x_2. Does knowledge about the value x_1 affect your estimate of x_2? If so, then the two variables are not independent. If the variables are independent, then knowing x_1 tells you nothing about x_2.

So, how does Σ affect the shape? Let's look at the independent and dependent cases separately:

Independent case ($\sigma_{12} = 0$): In this case x_1 and x_2 are drawn from two independent 1D Gaussian distributions. x_1 has mean μ_1 and variance σ_1^2, x_2 has mean μ_2 and variance σ_2^2.

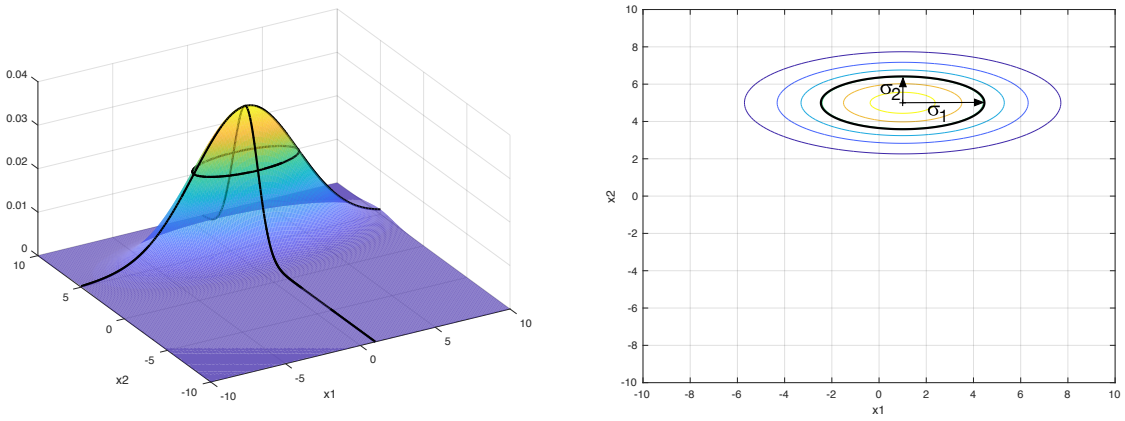
The 1σ ellipse is the set of points where

$$(x - \mu)^T \Sigma^{-1} (x - \mu) = 1.$$

Two facts about this ellipse:

- The principal axes of the ellipse will be aligned with the original variable axes.
- The length of the principle axes will be given by the standard deviation associated with that variable.

The picture looks like this:

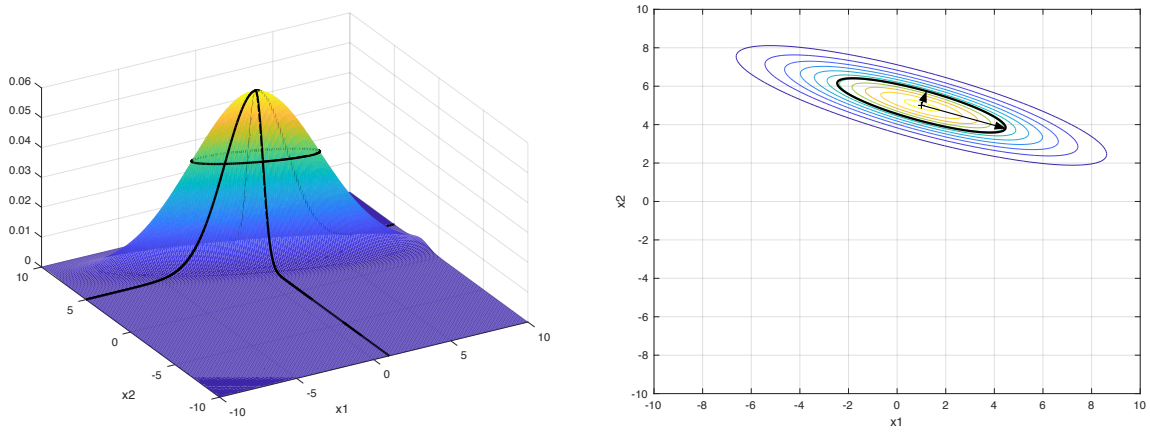


These facts will hold true for dimensions greater than n as well.

Non-independent case ($\sigma_{12} \neq 0$): In this case, we can still think about the "shape" of the distribution using the 1-σ ellipse, but it will have different properties since the two varialbes are statistically dependent:

- The directions of the principle axes will be given by the eigenvectors of Σ. These will always be real and orthogonal for a symmetric matrix.
- The lengths of the principle axes will be given by the square roots of the associated eigenvalues of Σ. These will always be positive since Σ is positive definite.

The picture looks like this:



3 Basic Operations with Gaussian Random Variables

For multivariate Gaussian distribustions (aka normal distributions), we use the expression

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

to mean "the random variable x is drawn from a multivariate Gaussian distribution with mean μ and covariance matrix Σ." This section describes what happens when we perform a few basic operations on Gaussian random variables.

Adding a random variable and a deterministic vector: Let x be a Gaussian random variable of dimension n

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

and let y be a fixed deterministic vector in $\mathbb{R}^n$. Then the quantity $(x + y)$ is a new random variable, and

$$(x + y) \sim \mathcal{N}(\mu + y, \Sigma).$$

The mean part of this comes directly from the fact that he expectation is linear: $E(x + y) = E(x) + E(y) = \mu + y$. For the covariance, intuitively, the non-random vector does not add any uncertainty, so the covariance is unchanged. A more formal argument can be made by looking at the definition of covariance (similar to what is done for the next operation, below).

Adding independent random variables: Let

$$x \sim \mathcal{N}(\mu_x, \Sigma_x) \quad \text{and} \quad y \sim \mathcal{N}(\mu_y, \Sigma_y).$$

The quantity $(x + y)$ is a new random variable, and

$$(x + y) \sim \mathcal{N}(\mu_x + \mu_y, \Sigma_x + \Sigma_y).$$

As in the above case, the result for the mean comes directly from the fact that expectation is linear. For the covariance, we start with the definition of covariance:

$$\text{Cov}(x + y) = E\left((x + y - E(x + y))(x + y - E(x + y))^T\right)$$

$$= E\left((x + y - \mu_x - \mu_y))(x + y - \mu_x - \mu_y)^T\right)$$

$$= E\left(((x - \mu_x) + (y - \mu_y))((x - \mu_x) + (y - \mu_y))^T\right)$$

$$= E\left((x-\mu_x)(x-\mu_x)^T + (x-\mu_x)(y-\mu_y)^T + (y-\mu_y)(x-\mu_x)^T + (y-\mu_y)(y-\mu_y)^T\right)$$
$$= \underbrace{E\left((x-\mu_x)(x-\mu_x)^T\right)}_{\Sigma_x} + \underbrace{E\left((x-\mu_x)(y-\mu_y)^T\right) + E\left((y-\mu_y)(x-\mu_x)^T\right)}_{=0 \text{ since } x,y \text{ independent}} + \underbrace{E\left((y-\mu_y)(y-\mu_y)^T\right)}_{\Sigma_y}$$

Multiplying a random variable by a matrix: Let x be a Gaussian random variable of dimension n

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

and let A be an $n \times n$ matrix. Then Ax is a new random variable, and

$$Ax \sim \mathcal{N}\left(A\mu, A\Sigma A^T\right)$$

The result for the mean comes from linearity of the expectation ($E(Ax) = AE(x)$). For the covariance we start with the definition:

$$\text{Cov}(Ax) = E\left((Ax - E(Ax))(Ax - E(Ax))^T\right)$$
$$= E\left((Ax - A\mu)(Ax - A\mu)^T\right)$$
$$= E\left(A(x-\mu)(x-\mu)^T A^T\right)$$
$$= AE\left((x-\mu)(x-\mu)^T\right) A^T$$
$$= A\Sigma A^T.$$

Merging distributions: Suppose that you know both of the following are true:

$$x \sim \mathcal{N}(\mu_1, \Sigma_1) \quad \text{and} \quad x \sim \mathcal{N}(\mu_2, \Sigma_2).$$

"Merging" means finding a single distribution that represents both conditions. In general, merging can be achieved by multiplying and renormalizing the underlying pdfs. For the Gaussian case, it turns out that this multiplication will yield a Gaussian distribution $\mathcal{N}(\mu_3, \Sigma_3)$ with mean

$$\mu_3 = \mu_1 + Q(\mu_2 - \mu_1)$$

and covariance matrix

$$\Sigma_3 = \Sigma_1 - Q\Sigma_1,$$

where

$$Q = \Sigma_1 \left(\Sigma_1 + \Sigma_2\right)^{-1}$$

4 Kalman Filter Derivation

Consider a discrete time linear state space system with added noise in the state and output equations:

$$x[k+1] = Fx[k] + Gu[k] + v[k]$$

$$y[k] = Hx[k] + w[k],$$

where $v[k]$ and $w[k]$ are independently distributed random variables with zero mean and covariance matrices $V[k]$ and $W[k]$, respectively.

$v[k] \sim \mathcal{N}(0, V[k])$ is called the *process noise*.

$w[k] \sim \mathcal{N}(0, W[k])$ is called the *measurement noise*.

The Kalman filter assumes that the initial state x_0 is drawn from a distribuiton with known mean $\hat{x}[0|0]$ and covariance $P[0|0]$:

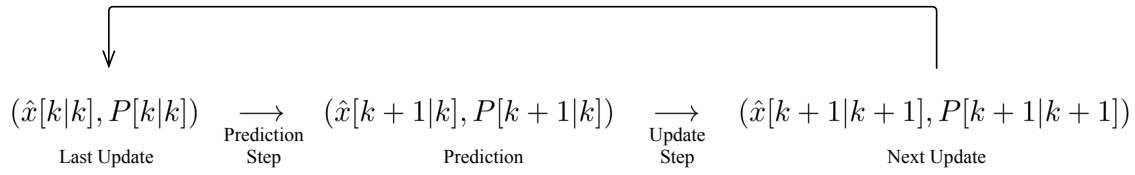
$$x_0 \sim \mathcal{N}(\hat{x}[0|0], P[0|0]).$$

After that, it just updates the estimated distribution by updating the mean and covariance as new inputs $u[k]$ are applied and new outputs $y[k]$ are measured.

Notationally, the pair $(\hat{x}[k|k], P[k|k])$ gives the estimate mean and covariance at time k given all inputs up to time $k-1$ and all outputs up to time k. This estimate is often called the "update", for reasons that will become apparent soon.

The pair $(\hat{x}[k+1|k], P[k+1|k])$ gives the estimate mean and covariance at time $k+1$ given all inputs up to time k and outputs up to time k. This estimate is often called the "prediction" since it is predicting the state at time $k+1$ before the measurement at time $k+1$ has been received.

At the highest level, the Kalman filter is an iterative sequence of predictions and updates:



So we need to work out the math for the prediction and update steps.

Prediction Step: We start with

$$x[k] \sim \mathcal{N}(\hat{x}[k|k], P[k|k])$$

and we apply the next input $u[k]$ using the process model, which says

$$x[k+1] = Fx[k] + Gu[k] + v[k]$$

The three terms on the right hand side are

- $Fx[k]$, a matrix F times a random variable $x[k] \sim \mathcal{N}(\hat{x}[k|k], P[k|k])$
- $Gu[k]$, a deterministic known vector in $\mathbb{R}^n$,
- $v[k]$, a random variable $v[k] \sim \mathcal{N}(0, V[k])$.

So after applying the input, we can compute the predicted distribution using a combination of the addition and matrix multiplication operations described above.

The mean $\hat{x}[k+1|k]$ must be

$$\hat{x}[k+1|k] = E(Fx[k] + Gu[k] + v[k])$$

$$= FE(x[k]) + E(Gu[k]) + E(v[k])$$

$$= F\hat{x}[k|k] + Gu[k] + 0.$$

The covariance $P[k+1|k]$ must be

$$
\begin{aligned}
&P[k+1|k] = \\
&E\left((Fx[k]+Gu[k]+v[k]) - E(Fx[k]+Gu[k]+v[k]))(Fx[k]+Gu[k]+v[k]) - E(Fx[k]+Gu[k]+v[k]))^T\right) \\
&= E\left((Fx[k]+Gu[k]+v[k]) - FE(x[k]) - Gu[k])(Fx[k]+Gu[k]+v[k]) - FE(x[k]) - Gu[k])^T\right) \\
&= E\left((Fx[k] - FE(x[k]) + v[k])(Fx[k] - FE(x[k]) + v[k])^T\right) \\
&= E\left((F(x[k] - E(x[k])) + v[k])(F(x[k] - E(x[k])) + v[k])^T\right) \\
&= E\left(F(x[k] - E(x[k]))(F(x[k] - E(x[k]))^T + F(x[k] - E(x[k]))v[k]^T + v[k](F(x[k] - E(x[k]))^T + v[k]v[k]^T\right) \\
&= \underbrace{E\left(F(x[k] - E(x[k]))(x[k] - E(x[k]))^T F^T\right)}_{=P[k|k]} + \underbrace{2E\left(F(x[k] - E(x[k]))v[k]^T\right)}_{=0,\,(x,v)\text{independent}} + \underbrace{E\left(v[k]v[k]^T\right)}_{=V[k]}
\end{aligned}
$$

So

$$P[k+1|k] = FP[k|k]F^T + V[k].$$

Update Step: Now we start with

$$x[k+1] \sim \mathcal{N}(\hat{x}[k+1|k], P[k+1|k])$$

and we receive the next input $y[k+1]$. From the output equation, we know that

$$y[k+1] = Hx[k+1] + w[k+1],$$

where $w[k+1]$ is zero mean Gaussian with covariance matrix $V[k+1]$. This means that

$$Hx[k+1] \sim \mathcal{N}(y[k+1], V[k+1]).$$

From the update starting condition, we get another condition on $Hx[k+1]$:

$$Hx[k+1] \sim \mathcal{N}\left(H\hat{x}[k+1|k], HP[k+1|k]H^T\right)$$

We can combine these two indpendent pieces of information to get a single distribution for $Hx[k+1]$ by merging $\mathcal{N}(y[k+1], V[k+1])$ and $\mathcal{N}\left(H\hat{x}[k+1|k], HP[k+1|k]H^T\right)$. From the merging formula above (we temporarily drop the indices on $x[k+1]$, $\hat{x}[k+1|k]$, $P[k+1|k]$, $y[k+1]$, $v[k+1]$, and $V[k+1]$):

$$Hx \sim \mathcal{N}\left(H\hat{x} + Q(y - H\hat{x}), HPH^T - QHPH^T\right),$$

where

$$Q = HPH^T\left(HPH^T + W\right)^{-1}.$$

To make the math easier to follow, define $S = HPH^T + W$, so that Q can be written more compactly as

$$Q = HPH^T S^{-1}.$$

Now we substitute for Q and do a bunch of math:

$$Hx \sim \mathcal{N}\left(H\hat{x} + HPH^T S^{-1}(y - H\hat{x}), HPH^T - HPH^T S^{-1} HPH^T\right),$$

$$Hx \sim \mathcal{N}\left(H\left(\hat{x} + PH^T S^{-1}(y - H\hat{x})\right), H\left(P - PH^T S^{-1} HP\right)H^T\right)$$

The above gives the distribution of Hx. Given what we know about matrix multiplication with Gaussian random variables, we can conclude that

$$x \sim \mathcal{N}\left(\hat{x} + PH^T S^{-1}(y - H\hat{x}), P - PH^T S^{-1} HP\right),$$

giving the equations we need to update the mean and covariance.

Kalman Filter equation summary:. Restating the prediction step and re-including the indices in the update step yields the following summary of the Kalman filter equations:

prediction step:

$$\hat{x}[k+1|k] = F\hat{x}[k|k] + Gu[k]$$

$$P[k+1|k] = FP[k|k]F^T + V[k].$$

update step:

$$S = HP[k+1|k]H^T + W[k]$$

$$\hat{x}[k+1|k+1] = \hat{x}[k+1|k] + P[k+1|k]H^T S^{-1}(y[k+1] - H\hat{x}[k+1|k])$$

$$P[k+1|k+1] = P[k+1|k] - P[k+1|k]H^T S^{-1} HP[k+1|k].$$