Explaining the Prediction and Update Equations for the Kalman Filter:

Notation and Models Used:

The Kalman Filter has the same prediction and update equations as the g-h and discrete bayes filter. The improvement the Kalman Filter has over both of the previous filters is that the Kalman Filter can account for a control input into the system. In the case of a car, the control input can be the throttle and brake of the car to make it accelerate faster or slow down. Another innovation of the Kalman Filter is that we can input mathematical models to represent the physics of the system or other dynamics such as financial equations. We'll assume that all the equations in the system are linear since the physics models we're going to use are linear as well as for simplicity's sake.

Let's say we want to predict our state matrix X_t. We know that X_t is going to be a function of the previous state matrix X_{t-1} as well as a control matrix. Let's call the control matrix U_t. We also know that there will be noise in the system as well which we can represent by using the variable ϵ_t. ϵ_t is assumed to be gaussian with a mean of 0 and covariance R_t. It's represented by the statement $\epsilon_t \sim N(0, R_t)$. How a covariance matrix is calculated is by having the variance along the diagonal of the matrix and the covariance between different variables everywhere else. For an n-dimensional matrix, n being the number of states you're observing, the covariance matrix will look like

$$\begin{bmatrix} Var(x_1) & Cov(x_1, x_2) & \cdots & Cov(x_1, x_n) \\ Cov(x_1, x_2) & Var(x_2) & \cdots & Cov(x_2, x_n) \\ \vdots & \vdots & \ddots & \vdots \\ Cov(x_1, x_n) & Cov(x_2, x_n) & \cdots & Var(x_n) \end{bmatrix}$$

What we have currently is the equation $X_t = X_{t-1} + U_t + \epsilon_t$. But, we can't use the matrices by themselves since there are no scaling factors to model the dynamics of the system. In high school physics class one is taught that the position of an object is determined by the equation $x_t = x_{t-1} + v_{t-1}\Delta t + \frac{1}{2}a_t\Delta t^2$. The state matrix is represented by the equation

$$X = \begin{bmatrix} x \\ \dot{x} \end{bmatrix}$$

x dot is the derivative of the position and is therefore the velocity of the system. Knowing this we can multiply a weight matrix A_t to X_{t-1} to get the new state X_t. The equation would look like $X_t = A_t X_{t-1} + U_t + \epsilon_t$. When $A_t X_{t-1}$is replaced with its matrix values and evaluated it looks like 63

$$\begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_{t-1} \\ \dot{x}_{t-1} \end{bmatrix} = \begin{bmatrix} x_{t-1} + \dot{x}_{t-1}\Delta t \\ \dot{x}_{t-1} \end{bmatrix}$$

The top value of the matrix looks like the first part of the kinematic equation used to find position. All that is left to do is to account for the acceleration of the system.

We can put the acceleration of the system in the control matrix because of the equation $F = ma$ and we apply a force to the system to change its state. U_t would then be represented by the vector $U_t = [a_t]$. We can then scale the control matrix by another weight matrix B_t to get the full equation $X_t = A_t X_{t-1} + B_t U_t + \epsilon_t$. The expression $B_t U_t$ when expressed with its matrix values looks like

$$\begin{bmatrix} \frac{\Delta t^2}{2} \\ \Delta t \end{bmatrix} \begin{bmatrix} a_t \end{bmatrix} = \begin{bmatrix} a_t \frac{\Delta t^2}{2} \\ a_t \Delta t \end{bmatrix}$$

When we combine both expressions together we find we get the matrix

$$\begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_{t-1} \\ \dot{x}_{t-1} \end{bmatrix} + \begin{bmatrix} \frac{\Delta t^2}{2} \\ \Delta t \end{bmatrix} \begin{bmatrix} a_t \end{bmatrix} + \epsilon_t = \begin{bmatrix} x_{t-1} + \dot{x}_{t-1}\Delta t + a_t \frac{\Delta t^2}{2} \\ \dot{x}_{t-1} + a_t \Delta t \end{bmatrix} + \epsilon_t$$

The top equation represents our predicted position and the bottom equation represents our predicted velocity. The kinematic equation for velocity is $v_{t-1} + a_t \Delta t$ which matches our result in the matrix. To generalize the equation $X_t = A_t X_{t-1} + B_t U_t + \epsilon_t$ and clarify the dimensions of the system, X_t and X_{t-1} are n×1, U_t is m×1, A_t is n×n, B_t is n×m, and ϵ_t is n×1.

Next, we need to derive the measurement equation. This is relatively simple since we can apply the same logic to our measurement equation. Our measurement Z_t is determined by our state matrix X_t. We can convert X_t into Z_t using another matrix C_t to convert how our measurements affect the state. We will also add the random error δ_t which is determined by the statement $\delta_t \sim N(0,\ Q_t)$ where Q_t is the covariance matrix of the gaussian function. Together the equation is $Z_t = C_t X_t + \delta_t$. C_t is a k×n matrix.

Another variable we need to calculate is the belief/probability distribution of the state at time t. The belief is assumed to be a multivariate gaussian since gaussian distributions are the distributions often assumed to model the real world accurately. It's represented by the expression $bel(X_t)$. For gaussian distributions we only need to know the mean and covariance matrix to construct our distribution. We use μ_t to represent our mean at time t and Σ_t to represent our covariance at time t. We use Σ instead of σ or σ^2 because we are using covariance instead of standard deviation or variance. One note to make is that since the Kalman Filter relies on previous measurements we must initialize our beliefs beforehand when starting the filter. Our initial beliefs will be represented by $bel(X_0)$ and we initialize our beliefs using $X_0 \sim N(\mu_0,\ \Sigma_0)$. The exact choice of the mean and covariance doesn't matter all too much as they will get updated as the filter runs its cycle.

The Prediction Equations:

Now that we have defined and explained every variable we need, let's define the proper prediction equations. To project the state of the system into the future without using our observations we can use the prediction equation we found earlier on. We can represent our projection as $\overline{\mu}_t = A_t X_{t-1} + B_t U_t$. μ_t and X_t are often interchangeable in the literature because they both represent the state values of the system. To project the covariance of the system into the future we use the equation $\overline{\Sigma}_t = A_t X_{t-1} A_t^T + R_t$. R_t being the covariance of the prediction equation as stated before. The way the equation was derived was through the assumption that the system can be represented by markov chains and deriving the equations using the properties of markov chains. The bar on top signifies that the variable is used in the prediction step. In the literature this is also known as the prior belief because the belief is prior to the update. The update is known as the posterior belief because it is calculated at the very end of the algorithm.

The Update Equations:

The next step is to derive the update equations. We can start by calculating the kalman gain. Using the g-h filter as an analogy, the kalman gain serves as both the g and h value of the filter. The equations for the kalman gain is $K_t = \overline{\Sigma}_t C_t^T (C_t \overline{\Sigma}_t C_t^T + Q_t)^{-1}$ where $K_t \in (0, 1)$. The closer the kalman gain is to one, the more trust is put into the measurements. The closer the kalman gain is to 0, the more trust is put into the predictions. How the equation works is if the noise in the measurement, Q_t, is small then the measurements can be trusted more. If the noise was big then the trust in the measurement is small. The result is the kalman gain getting closer and closer to 1. If the noise in the prediction is big then the result would be the kalman gain getting smaller as the value of the denominator increases.

The kalman gain will be used in the final estimate of the mean and covariance of the filter. To calculate the mean of the system we use the equation $\mu_t = \overline{\mu_t} + K_t(Z_t - C_t\mu_t)$. μ_t can be interpreted the same as X_t because it represents the state of the system. The only difference in the notation is that it represents the mean to the gaussian distribution while X_t does not imply a gaussian distribution. The reason why the update equation for the mean works is because. To calculate the covariance of the system we use $\Sigma_t = (I - K_t C_t)\ \overline{\Sigma_t}$.

Extended Kalman Filter:

The purpose of the extended kalman filter is to fix the problem of linearization. The original kalman filter assumes that the system can be modeled linearly i.e. can be modeled using only addition and multiplication. But, many real-world problems can't be modeled linearly. Instead most real-world problems have to be modeled non-linearly. The way the extended kalman filter compensates for the linear nature of the original kalman filter is by using the first-order Taylor series approximation of the non-linear model. The taylor series takes a function and turns it into a polynomial. Using the first-order Taylor polynomial can approximate the nonlinear function using a linear function.

We can represent our model as a function of the state and control matrix using the expression $f(X_t, U_t)$. We also need to change the observation to a nonlinear function which we can represent as h(x). Knowing this the full system can be represented as

$$X_t = f(X_{t-1}, U_t) + \epsilon_t$$

$$Z_t = h(X_t) + \delta_t$$

The A_t matrix can be represented as the partial derivative of $f(X_t, U_t)$ with respect to X_t. We can do a similar process with the $h(X_t)$. The resulting matrices are

$$A_t = \frac{\partial f(X_{t-1}, U_t)}{\partial X}\Big|_{X_{t-1}}$$

$$C_t = \frac{\partial h(X_t)}{\partial X}\Big|_{X_t}$$

The final equation we need to modify is the mean update equation which will now be represented by $\mu_t = \overline{\mu_t} + K_t(Z_t - h(X_t))$.

Conclusion:

The Kalman Filter is an effective way of removing statistical noise from measurements using linear algebra, statistics, and system modeling. The Kalman Filter doesn't suffer as much of a lag when compared to other filtering techniques and can accurately model the true signal. The only limitation of the original filter is that it's only limited to linear systems. Most of the physical models of the world are nonlinear which means that the Kalman Filter is not the optimal filter for those situations. The effects can be remedied using the Extended Kalman Filter, but that too has its limitations. Despite this limitation, the Kalman Filter has been used on rocketships and lunar landers which is a testament to how effective this filter can be.