9. Kalman Filter

The Kalman Filter theory has an almost cult status and it definitely has a place among the great scientific discoveries. The blunt formulas that describe the solution to the filter hide its rather simple basic principles and make it look like magic. And its ease of application make it feel even more magic. It is usually the last chapter in many Signal Processing courses and this makes it look like the pinnacle of the area. This, coupled with the fact that it is usually proved or derived using unnecessarily fancy math, makes it look like a realy challenging topic. It is not.

In any case the Kalman filter is a great idea and all great ideas are simple. It is applied whenever we want to compute the state of a system, like the position of a robot. If, from previous measurements, we know the position of the robot with some uncertainty and we make a few measurements and infer the position again with some uncertainty, we can average the two and get a better estimate. This is more or less what the Kalman filter does, but it does it very robustly. Even if the uncertainty in the measurements is infinite in one or more dimensions, the filter still works.

Let's look at that simple example again. A robot knows that its position in the x and y direction is 5 and 7 with a variance of 1 and 10 respectively. It can also fire its sonars and get its distance from the walls and find that its x and y position is 3 and 5 with variance of 10 and 1 respectively. We can compute the estimate of its position using the formulas for the weighted average. The variance-covariance matrix for the position of the robot is

$$\mathbf{C}_1 = \begin{bmatrix} 1 & 0 \\ 0 & 10 \end{bmatrix}$$

and the variance-covariance matrix for the sonar measurement is

$$\mathbf{C}_2 = \begin{bmatrix} 10 & 0 \\ 0 & 1 \end{bmatrix}.$$

and from these we can compute the $\mathbf{C}_{avg}$ and $\hat{x}$

$$\mathbf{C}_{avg} = \left(\mathbf{C}_1^{-1} + \mathbf{C}_2^{-1} \right)^{-1} = \begin{bmatrix} 0.91 & 0 \\ 0 & 0.91 \end{bmatrix}$$

The best estimate is then

$$\hat{x} = \begin{bmatrix} 4.81 \\ 5.18 \end{bmatrix}.$$

This kind of computation is the heart of the Kalman filter.

Before we can discuss the filter and its magical powers we need some formalism to make things concrete. The formalism will include two equations. One to describe the position of the robot motion and the other to describe the behaviour of the sensors. With

a good such formalism, the only thing we need to describe the robot is to fill in the various parameters of the equations, and everything else is taken care by the filter.

To keep the discussion short and general let us define first the *state vector* of a robot to be vector of all quantities needed to describe its state. A simple slow robot that roams a lab might need a state vector that has only three elements: its position x and y and its orientation θ. A space probe on the other hand could have many more: 3 for its position, 3 for its orientation, 3 for its velocity, 3 for its angular velocity, and a number of other parameters like the weight of its fuel tanks, state of its solar panels etc. It is obvious that another fitting name for state vector could be generalized position vector. Somehow the name *state* won and this is what we will use.

If the robot does not plan to stay in the same state forever, we have to have a way to specify how it changes its state, e.g. how it moves, unfolds its panels, etc. These state transitions can be dictated by the physics of the problem (if it moves with 100 miles an hour now, it will cover a certain distance until the next measurement, for example). We call this physics model *plant model* or *state transition model*. The state of a space robot for example, will change in various ways. The position component of the state will change according to its velocity and its velocity will change according to the external gravitational fields, etc.

The state might also change due to commands that it might receive from the human operator or the controling computer (by shoveling some more trilithium in the burner, or simply by stepping on the accelarator in the car or turning the steering wheel). We call the vector of these commands *control input* and we have to include it in the state transition equation.

So, it seems, that if we define the state of our robot, the physics of state transitions and the control input mechanism, we have described the model completely. Unfortunately it only seems so. In practice there are usualy many things that we did not account for, like wheel slippage of a lab roaming robot, headwinds for a flying robot, poorly known gravitational field for a moon orbiter etc. The reasons that we do not account for them can be ignorance of the mathematical models, inability to handle the complexity, stinginess, lazyness and other typicaly human qualities. In order to protect the public image of science and safeguard the trust of the common folks towards science we call all these things that we do not account for *noise*. And so we complete our state transition equation by incorporating some noise in it.

The other half of the story is the measurement model, where we write down an equation that relates the state of our robot to a set of measurements. Again, for the same reasons, we have to take into account the presense of noise. It is also very rare that these measurements will give us an estimate for the whole state vector, which incidentally is where the Kalman filter is the most useful. This means that we can guess the measurements if we know the state (if there was no noise we should be able to compute the measurements), but usualy we cannot find the state given the measurements even if we have no noise. For example, if we have a lab roaming robot with sonars we can use the measurements to find where the robot is (if we know where the sonar beams are bouncing off), but in general we cannot use this directly to find the robot orientation directly. So

stay tuned for the next section to see how the Kalman filter can help.

9.1. State and Measurement Models

A simple and slow robotic train moving on a simple track has a simple 1-D state vector that is expressed by a real number $\mathbf{x}$ which specifies the position on the track. If the robotic train is controlled in a "stop and go" fashion then the next state is

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{u}_k + \mathbf{v}_k$$

where the subscript k in the state x_k refers to time k and similarly for x_{k+1}. The second term $\mathbf{u}_k$ is the so called control input it is simply the distance it was ordered to travel. The last term is the noise in the system. Usually the only thing we know about the noise is its mean and variance. Whenever possible we prefer to deal with zero mean noise for convenience and this is what we do here. For this simple robot we only need the variance ${\sigma_v}^2$ to completely specify the plant model.

A more evolved train that moves fast enough to have some kinetic energy has a more elaborate state description: a vector

$$\mathbf{x}_k = \begin{bmatrix} x_k \\ u_k \end{bmatrix}.$$

The vector has two elements x_k, the position along the track and u_k, the speed of the train. Using some kindergarten physics the next state should be

$$\mathbf{x}_{k+1} = \begin{bmatrix} x_k + u_k \delta t \\ u_k + \Delta u_k \end{bmatrix} + \mathbf{v}_k$$

where δt is the time interval and Δu_k is the acceleration at the end of each time interval. The noise $\mathbf{v}_k$ is a 2-D vector now with zero mean and variance covariance matrix $C_{v,k}$. We notice that we can write the same equation as

$$\begin{aligned} \mathbf{x}_{k+1} &= \begin{bmatrix} 1 & \delta t \\ 0 & 1 \end{bmatrix} \mathbf{x}_k + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \Delta u_k + \mathbf{v}_k = \\ & \Phi_k \mathbf{x}_k + \Gamma_k \mathbf{u}_k + \mathbf{v}_k \end{aligned} \tag{9.1}$$

which not only makes us look more sophisticated but it is more convenient because we can give meaningful names to the different parts. Matrix Φ_k is called state transition matrix and Γ_k is called control input matrix.

Our robot also has some position sensors that give us some measurements that are related to the state. In general these measurements are not enough to deduce the state accurately. But if we know the state of the robot (e.g. its position and speed) and the physics of the sensor, then we can guess the output of the sensor. In our simple example, let's assume that the train is equipped with an odometer that measures distance traveled. Then the measurement vector $\mathbf{z}_{k+1}$ is 1-D and it relation with the state is

$$\mathbf{z}_{k+1} = [1 \quad 0]\mathbf{x}_{k+1} + \mathbf{w}_{k+1} = \Lambda_{k+1}\mathbf{x}_{k+1} + \mathbf{w}_{k+1}. \tag{9.2}$$

This relation is nice and simple but it cannot give us the complete state directly. Of course we might estimate the velocity by taking two measurements at times k and $k+1$ and subtracting. Not suprisingly, this is what the Kalman filter does essntialy with all the formulas that we develop later. Only it does it consistently and optimaly in all the extremely different situations that it is used.

If in our simple example we assume that the train is equipped with both a speedometer and odometer so that the measurement vector $\mathbf{z}_{k+1}$ is

$$\mathbf{z}_{k+1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\mathbf{x}_{k+1} + \mathbf{w}_{k+1} = \Lambda_{k+1}\mathbf{x}_{k+1} + \mathbf{w}_{k+1} \tag{9.3}$$

where Λ_{k+1} is the measurement matrix. In this case the measurement matrix is a very friendly identity matrix, but we will not always be that lucky. In most cases matrix Λ is more similar to the one in Eq. (9.2) rather than to Eq. (9.3) so it is not even square not to mention invertible. This is though when the Kalman filter is most useful. To complete our model we need to specify the properties of the vector $\mathbf{w}_{k+1}$, the omnipresent noise. It is usualy assumed zero mean so we only need to specify the variance covariance matrix $C_{w,k+1}$.

9.2. Next State Prediction

Now that we have these two equations, the state transition equation (9.1) and the measurement equation (9.3), we can try to find some good use for them. We will use the following scenario. Every time we apply the filter we start from the previous state which we assume we have computed as accurately as possible and use the equations to compute the current state again as accurately as we can. And we know from our statistics that whenever we estimate a quantity we have to estimate the uncertainty of our estimate as well which usually means we have to estimate the variance covariance matrix.

We do this in two steps. In the first step we use the state transition equation to predict the current state and its variance covariance matrix without incorporating the measurements and then we incorporate measurements to get the final estimate for the state along with the variance covariance matrix.

In most cases the best estimate of an actual value underlying a random variable is its expected value, so

$$\hat{\mathbf{x}}_{k+1}^{k} = E\{\mathbf{x}_{k+1}\}$$

where $\hat{\mathbf{x}}_{k+1}^{k}$ is the estimate of the state $\mathbf{x}$ at time $k+1$ based on measurements up until time k. Substituting the state transition equation (9.1) we get

$$\hat{\mathbf{x}}_{k+1}^{k} = E\{\mathbf{x}_{k+1}\} = E\{\Phi_k\mathbf{x}_k + \Gamma_k\mathbf{u}_k + \mathbf{v}_k\} =$$
$$E\{\Phi_k\mathbf{x}_k\} + E\{\Gamma_k\mathbf{u}_k\} + E\{\mathbf{v}_k\}.$$

The third of these terms is the expected value of the noise which we have thoughtfully assumed to be zero. The second term is not even a random variable so its expected value is itself. And the first term is the product of a constant and a random variable so

$$\hat{\mathbf{x}}_{k+1}^k = \Phi_k E\{\mathbf{x}_k\} + \Gamma_k \mathbf{u}_k = \Phi_k \hat{\mathbf{x}}_k^k + \Gamma_k \mathbf{u}_k \tag{9.4}$$

where $\hat{\mathbf{x}}_k^k$ is the estimate of the state $\mathbf{x}$ at time k based on measurements up until time k which is what we got by applying Kalman filter for the transition from time $k-1$ to k. We have to compute now the variance covariance matrix of the state $\mathbf{x}_{k+1}$. Substituting Eq. (9.1) and Eq. (9.4) in the definition of the variance covariance matrix we get

$$P_{k+1}^k = E\left\{\left(\mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1}^k\right)\left(\mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1}^k\right)^T\right\} =$$

$$E\left\{\left(\Phi_k\mathbf{x}_k + \Gamma_k\mathbf{u}_k + \mathbf{v}_k - \Phi_k\hat{\mathbf{x}}_k^k - \Gamma_k\mathbf{u}_k\right)\left(\Phi_k\mathbf{x}_k + \Gamma_k\mathbf{u}_k + \mathbf{v}_k - \Phi_k\hat{\mathbf{x}}_k^k - \Gamma_k\mathbf{u}_k\right)^T\right\} =$$

$$E\left\{\left(\Phi_k\mathbf{x}_k + \mathbf{v}_k - \Phi_k\hat{\mathbf{x}}_k^k\right)\left(\Phi_k\mathbf{x}_k + \mathbf{v}_k - \Phi_k\hat{\mathbf{x}}_k^k\right)^T\right\} =$$

$$E\left\{\left(\Phi_k\left(\mathbf{x}_k - \hat{\mathbf{x}}_k^k\right) + \mathbf{v}_k\right)\left(\Phi_k\left(\mathbf{x}_k - \hat{\mathbf{x}}_k^k\right) + \mathbf{v}_k\right)^T\right\}.$$

The next step is to expand the product inside the expected value and get four terms. Two of these terms contain both $\mathbf{x}_k$ and $\mathbf{v}_k$ which are random variables which we assume independent, so the expected value of their product is the product of their expected values

$$E\left\{\Phi_k(\mathbf{x}_k - \hat{\mathbf{x}}_k^k)\mathbf{v}_k{}^T\right\} = E\left\{\Phi_k(\mathbf{x}_k - \hat{\mathbf{x}}_k^k)\right\}E\left\{\mathbf{v}_k{}^T\right\} = 0$$

because the expected value of both factors are zero. The same is true for the other term that contains both random variables. So we are left with the other two terms

$$P_{k+1}^k = E\left\{\Phi_k(\mathbf{x}_k - \hat{\mathbf{x}}_k^k)(\Phi_k(\mathbf{x}_k - \hat{\mathbf{x}}_k^k))^T\right\} + E\left\{\mathbf{v}_k\mathbf{v}_k{}^T\right\} =$$

$$E\left\{\Phi_k(\mathbf{x}_k - \hat{\mathbf{x}}_k^k)(\mathbf{x}_k - \hat{\mathbf{x}}_k^k)^T\Phi_k{}^T\right\} + E\left\{\mathbf{v}_k\mathbf{v}_k{}^T\right\} =$$

$$\Phi_k P_k^k \Phi_k^T + C_{v,k}$$

where P_k^k is the variance covariance matrix of $\mathbf{x}_k$ at time k incorporating measurements up to time k.

9.3. Incorporating Measurements

It is time now to bring the measurements into the picture before they go stale. The problem is that this is not at all easy because matrix Λ_k, the measurement matrix, is not

always invertible. So what! We will assume it is, convince ourselves that the formulas we get make sense and then open a real book and verify that the same formulas apply for non invertible matrices Λ_k. Now if this does not convince everybody and we need some more solid proof we can always derive the equations from scratch using χ^2. The math is not terribly convolved and all the derivations are clean and beautiful. In fact, people that tend to use the words mathematics and aesthetics in the same sentence, would consider it poetry.

Assuming that Λ_k is invertible we can use a few simple facts from statistics. If we want to combine two equally trustworthy estimates to get a better one we take the average. You do not need to be a statistician to know that. If the estimates are not equally trustworthy then we take the weighted average. Again, nothing unexpected here. And this is true for any dimensions.

In our problem we have an estimate of the state of the robot at time $k+1$, which we got from the estimate of the state in time k and a set of measurements. Since we assume that matrix Λ_{k+1} is invertible we also have an estimate of the position of the robot from the measurements. We can then take the weighted average of these two estimates and get a better estimate in exactly the same manner as above but instead of handling matrices with real numbers, we handle matrices in symbolic form. The only tricky thing is this assumption about Λ_{k+1}, which only means that we have to be careful not to leave any inverses of Λ_{k+1} in the final equations.

We already know $\hat{\mathbf{x}}_{k+1}^k$ the expected value of the state and P_{k+1}^k the variance covariance matrix, which are based on knowledge up until time k. We can compute the state from the measurements using Eq. (9.3)

$$\mathbf{x}_{k+1} = \Lambda_{k+1}^{-1}(\mathbf{z}_{k+1} - w_{k+1})$$

and from this we can easily compute the expected value

$$\hat{\mathbf{x}}^{k+1} = E\{\mathbf{x}_{k+1}\} = E\left\{\Lambda_{k+1}^{-1}(\mathbf{z}_{k+1} - w_{k+1})\right\} =$$

$$\Lambda_{k+1}^{-1}\mathbf{z}_{k+1}$$

and the variance covariance matrix

$$C_{x,k+1} = E\left\{(\mathbf{x}_{k+1} - \hat{\mathbf{x}}^{k+1})(\mathbf{x}_{k+1} - \hat{\mathbf{x}}^{k+1})^T\right\} =$$

$$E\left\{(\Lambda_{k+1}^{-1}(\mathbf{z}_{k+1} - w_{k+1}) - \Lambda_{k+1}^{-1}\mathbf{z}_{k+1})(\Lambda_{k+1}^{-1}(\mathbf{z}_{k+1} - w_{k+1}) - \Lambda_{k+1}^{-1}\mathbf{z}_{k+1})^T\right\} = \Lambda_{k+1}^{-1}C_{w,k+1}\Lambda_{k+1}^{-T}.$$

Now we can compute the expected value and variance covariance matrix using the weighted average. We start from the variance covariance matrix because it is easier. The formula for this is

$$\left(\mathbf{C}_1{}^{-1} + \mathbf{C}_2{}^{-1}\right)^{-1}$$

which can be more useful if written as

$$\mathbf{C}_1(\mathbf{C}_1 + \mathbf{C}_2)^{-1}\mathbf{C}_2.$$

In our case the two sources of information are the previous state and the measurements so $\mathbf{C}_1$ and $\mathbf{C}_2$ are P^k_{k+1} and $\Lambda^{-1}_{k+1}C_{w,k+1}\Lambda^{-T}_{k+1}$ respectively. So

$$P^{k+1}_{k+1} = P^k_{k+1}\left(P^k_{k+1} + \Lambda^{-1}_{k+1}C_{w,k+1}\Lambda^{-T}_{k+1}\right)^{-1}\Lambda^{-1}_{k+1}C_{w,k+1}\Lambda^{-T}_{k+1} =$$

$$P^k_{k+1}\left(\Lambda_{k+1}P^k_{k+1} + C_{w,k+1}\Lambda^{-T}_{k+1}\right)^{-1}C_{w,n+1}\Lambda^{-T}_{k+1} =$$

$$P^k_{k+1}\Lambda^T_{k+1}\left(\Lambda_{k+1}P^k_{k+1}\Lambda^T_{k+1} + C_{w,k+1}\right)^{-1}C_{w,n+1}\Lambda^{-T}_{k+1}.$$

If we define the *Kalman Gain* to be

$$K_{k+1} = P^k_{k+1}\Lambda^T_{k+1}\left(\Lambda_{k+1}P^k_{k+1}\Lambda^T_{k+1} + C_{w,k+1}\right)^{-1} \tag{9.5}$$

we can conclude that

$$P^{k+1}_{k+1} = K_{k+1}C_{w,k+1}\Lambda^{-T}_{k+1}.$$

Are we done already? Well, no. We still have a stubborn Λ^{-T}_{k+1} and we do not want it. It turns out that

$$K_{k+1}C_{w,k+1}\Lambda^{-T}_{k+1} = P^k_{k+1} - K_{k+1}\Lambda_{k+1}P^k_{k+1} \tag{9.6}$$

because we can show that their difference is zero

$$K_{k+1}C_{w,k+1}\Lambda^{-T}_{k+1} - P^k_{k+1} + K_{k+1}\Lambda_{k+1}P^k_{k+1} = K_{k+1}\left(C_{w,k+1}\Lambda^{-T}_{k+1} + \Lambda_{k+1}P^k_{k+1}\right) - P^k_{k+1} =$$

$$K_{k+1}\left(C_{w,k+1} + \Lambda_{k+1}P^k_{k+1}\Lambda^T_{k+1}\right)\Lambda^{-T}_{k+1} - P^k_{k+1} =$$

$$P^k_{k+1}\Lambda^T_{k+1}\left(\Lambda_{k+1}P^k_{k+1}\Lambda^T_{k+1} + C_{w,k+1}\right)^{-1}\left(C_{w,k+1} + \Lambda_{k+1}P^k_{k+1}\Lambda^T_{k+1}\right)\Lambda^{-T}_{k+1} - P^k_{k+1}.$$

One can notice the matrices that are neighboring their inverses and mutually annihilate leaving $P^k_{k+1} - P^k_{k+1} = 0$. So

$$P^{k+1}_{k+1} = P^k_{k+1} - K_{k+1}\Lambda_{k+1}P^k_{k+1}.$$

And now we are done with the variance covariance. On to $\hat{\mathbf{x}}^{k+1}_{k+1}$ the estimate of the state. The formula for this is

$$\left(\mathbf{C}_1{}^{-1}+\mathbf{C}_2{}^{-1}\right)^{-1}\left(\mathbf{C}_1{}^{-1}x_1+\mathbf{C}_2{}^{-1}x_2\right)$$

We notice that the first factor is the variance covariance matrix P_{k+1}^{k+1} we just computed. So

$$\hat{\mathbf{x}}_{k+1}^{k+1}=\left(P_{k+1}^{k}-K_{k+1}\Lambda_{k+1}P_{k+1}^{k}\right)\left((P_{k+1}^{k})^{-1}\hat{\mathbf{x}}_{k+1}^{k}+\left(\Lambda_{k+1}^{-1}C_{w,k+1}\Lambda_{k+1}^{-T}\right)^{-1}\Lambda_{k+1}^{-1}\mathbf{z}_{k+1}\right)=$$

$$\left(P_{k+1}^{k}-K_{k+1}\Lambda_{k+1}P_{k+1}^{k}\right)\left((P_{k+1}^{k})^{-1}\hat{\mathbf{x}}_{k+1}^{k}+\left(\Lambda_{k+1}^{T}C_{w,k+1}^{-1}\Lambda_{k+1}\right)\Lambda_{k+1}^{-1}\mathbf{z}_{k+1}\right)=$$

$$\left(\hat{\mathbf{x}}_{k+1}^{k}-K_{k+1}\Lambda_{k+1}\hat{\mathbf{x}}_{k+1}^{k}\right)+\left(P_{k+1}^{k}-K_{k+1}\Lambda_{k+1}P_{k+1}^{k}\right)\left(\Lambda_{k+1}^{T}C_{w,k+1}^{-1}\Lambda_{k+1}\right)\Lambda_{k+1}^{-1}\mathbf{z}_{k+1}$$

and if we make use of Eq. (9.6)

$$\left(\hat{\mathbf{x}}_{k+1}^{k}-K_{k+1}\Lambda_{k+1}\hat{\mathbf{x}}_{k+1}^{k}\right)+K_{k+1}C_{w,k+1}\Lambda_{k+1}^{-T}\left(\Lambda_{k+1}^{T}C_{w,k+1}^{-1}\Lambda_{k+1}\right)\Lambda_{k+1}^{-1}\mathbf{z}_{k+1}$$

and finally

$$\hat{\mathbf{x}}_{k+1}^{k+1}=\hat{\mathbf{x}}_{k+1}^{k}+K_{k+1}\left(\mathbf{z}_{k+1}-\Lambda_{k+1}\hat{\mathbf{x}}_{k+1}^{k}\right)$$

Now our job is complete. Or kind of complete. We still have to prove that these formulas apply in case Λ_{k+1} is not invertible. One could put together a quick and dirty proof by using a sequence of invertible matrices Λ_{k+1} that converge to a non invertible matrix. Or do the whole thing the proper way starting from χ^2. Or prove it in a Bayessian fashion.

9.4. Bayessian Derivation

From the state transition equation

$$\mathbf{x}_{k+1}=\Phi_k\mathbf{x}_k+\mathbf{v}_k$$

where for convenience we omitted the $\Gamma_k\mathbf{u}_k$ term. Since we know that the distribution of $\mathbf{v}_k$ is a zero mean Gaussian with variance $C_{v,k}$, the probability density function of $\mathbf{x}_{k+1}$ given $\mathbf{x}_k$ is

$$p(\mathbf{x}_{k+1}\mid\mathbf{x}_k)\sim N(\Phi_k\mathbf{x}_k,C_{v,k}). \tag{9.7}$$

The probability density function of $\mathbf{x}_k$ given all the past measurements is assumed known from the previous step and is

$$p(\mathbf{x}_k\mid\mathbf{z}_{0..k})\sim N(\hat{\mathbf{x}}_k^k,P_k^k). \tag{9.8}$$

In the first step we want to determine the probability density function of the next state $\mathbf{x}_{k+1}$ given all the past measurements up to time k which can be done by using marginalization

$$p(\mathbf{x}_{k+1} \mid z_{0..k}) =$$
$$\int p(\mathbf{x}_{k+1}, \mathbf{x}_k \mid z_{0..k}) d\mathbf{x}_k =$$
$$\int p(\mathbf{x}_{k+1} \mid \mathbf{x}_k, \mathbf{z}_{0..k}) p(\mathbf{x}_k \mid \mathbf{z}_{0..k}) d\mathbf{x}_k =$$

and under the Markov assumption, whereby if we have the last state we need no other information about the past

$$\int p(\mathbf{x}_{k+1} \mid \mathbf{x}_k) p(\mathbf{x}_k \mid \mathbf{z}_{0..k}) d\mathbf{x}_k =$$

which if we use Eqs. (9.7) and (9.8) and ignore the multiplicative constants becomes

$$\int e^{-\frac{(\mathbf{x}_{k+1}-\Phi_k\mathbf{x}_k)^T C_{v,k}^{-1}(\mathbf{x}_{k+1}-\Phi_k\mathbf{x}_k) + (\mathbf{x}_k-\hat{\mathbf{x}}_k^k)^T (P_k^k)^{-1}(\mathbf{x}_k-\hat{\mathbf{x}}_k^k)}{2}} d\mathbf{x}_k \, . \tag{9.9}$$

To avoid carrying around the whole integral we work, for now, on the numerator of the exponential

$$(\mathbf{x}_{k+1} - \Phi_k\mathbf{x}_k)^T C_{v,k}^{-1}(\mathbf{x}_{k+1} - \Phi_k\mathbf{x}_k) + (\mathbf{x}_k - \hat{\mathbf{x}}_k^k)^T (P_k^k)^{-1}(\mathbf{x}_k - \hat{\mathbf{x}}_k^k) =$$
$$\mathbf{x}_{k+1}^T C_{v,k}^{-1}\mathbf{x}_{k+1} - 2\mathbf{x}_{k+1}^T C_{v,k}^{-1}\Phi_k\mathbf{x}_k + \mathbf{x}_k^T\Phi_k^T C_{v,k}^{-1}\Phi_k\mathbf{x}_k +$$
$$\mathbf{x}_k^T (P_k^k)^{-1}\mathbf{x}_k - 2\mathbf{x}_k^T (P_k^k)^{-1}\hat{\mathbf{x}}_k^k + (\hat{\mathbf{x}}_k^k)^T (P_k^k)^{-1}\hat{\mathbf{x}}_k^k =$$

and after we drop the term that does not contain either $\mathbf{x}_{k+1}$ or $\mathbf{x}_k$, we separate the terms that contain $\mathbf{x}_k$ and have

$$\mathbf{x}_k^T\left(\Phi_k^T C_{v,k}^{-1}\Phi_k + (P_k^k)^{-1}\right)\mathbf{x}_k -$$
$$2\left(\mathbf{x}_{k+1}^T C_{v,k}^{-1}\Phi_k + (\hat{\mathbf{x}}_k^k)^T (P_k^k)^{-1}\right)\mathbf{x}_k +$$
$$\mathbf{x}_{k+1}^T C_{v,k}^{-1}\, \mathbf{x}_{k+1} =$$

and if we multiply and divide, and add and subtract the appropriate factors and terms

$$\mathbf{x}_k^T\left(\Phi_k^T C_{v,k}^{-1}\Phi_k + (P_k^k)^{-1}\right)\mathbf{x}_k -$$
$$2\left(\mathbf{x}_{k+1}^T C_{v,k}^{-1}\Phi_k + (\hat{\mathbf{x}}_k^k)^T (P_k^k)^{-1}\right)\left(\Phi_k^T C_{v,k}^{-1}\Phi_k + (P_k^k)^{-1}\right)^{-1}\left(\Phi_k^T C_{v,k}^{-1}\Phi_k + (P_k^k)^{-1}\right)\mathbf{x}_k +$$
$$\left(\mathbf{x}_{k+1}^T C_{v,k}^{-1}\Phi_k + (\hat{\mathbf{x}}_k^k)^T (P_k^k)^{-1}\right)\left(\Phi_k^T C_{v,k}^{-1}\Phi_k + (P_k^k)^{-1}\right)^{-1}\left(\Phi_k^T C_{v,k}^{-1}\mathbf{x}_{k+1} + (P_k^k)^{-1}\hat{\mathbf{x}}_k^k\right) -$$
$$\left(\mathbf{x}_{k+1}^T C_{v,k}^{-1}\Phi_k + (\hat{\mathbf{x}}_k^k)^T (P_k^k)^{-1}\right)\left(\Phi_k^T C_{v,k}^{-1}\Phi_k + (P_k^k)^{-1}\right)^{-1}\left(\Phi_k^T C_{v,k}^{-1}\mathbf{x}_{k+1} + (P_k^k)^{-1}\hat{\mathbf{x}}_k^k\right) +$$
$$\mathbf{x}_{k+1}^T C_{v,k}^{-1}\, \mathbf{x}_{k+1} =$$

we get the first three terms in a perfect square that contains all the occurrences of $\mathbf{x}_k$ which will integrate to a constant (if you still remember the integral in Eq. (9.9)) and both the perfect square and the integral will disappear leaving us with

$$
\begin{aligned}
&-\left(\mathbf{x}_{k+1}^T C_{v,k}^{-1}\Phi_k + (\hat{\mathbf{x}}_k^k)^T (P_k^k)^{-1}\right)\left(\Phi_k^T C_{v,k}^{-1}\Phi_k + (P_k^k)^{-1}\right)^{-1}\left(\Phi_k^T C_{v,k}^{-1}\mathbf{x}_{k+1} + (P_k^k)^{-1}\hat{\mathbf{x}}_k^k\right) + \\
&\mathbf{x}_{k+1}^T C_{v,k}^{-1}\ \mathbf{x}_{k+1} = \\
&\mathbf{x}_{k+1}^T C_{v,k}^{-1}\ \mathbf{x}_{k+1} - \mathbf{x}_{k+1}^T C_{v,k}^{-1}\Phi_k\left(\Phi_k^T C_{v,k}^{-1}\Phi_k + (P_k^k)^{-1}\right)^{-1}\Phi_k^T C_{v,k}^{-1}\mathbf{x}_{k+1} \\
&-2\mathbf{x}_{k+1}^T C_{v,k}^{-1}\Phi_k\left(\Phi_k^T C_{v,k}^{-1}\Phi_k + (P_k^k)^{-1}\right)^{-1}(P_k^k)^{-1}\hat{\mathbf{x}}_k^k \\
&-(\hat{\mathbf{x}}_k^k)^T (P_k^k)^{-1}\left(\Phi_k^T C_{v,k}^{-1}\Phi_k + (P_k^k)^{-1}\right)^{-1}(P_k^k)^{-1}\hat{\mathbf{x}}_k^k =
\end{aligned}
$$

and after combining the first two terms we get

$$
\begin{aligned}
&\mathbf{x}_{k+1}^T\left(C_{v,k}^{-1} - C_{v,k}^{-1}\Phi_k\left(\Phi_k^T C_{v,k}^{-1}\Phi_k + (P_k^k)^{-1}\right)^{-1}\Phi_k^T C_{v,k}^{-1}\right)\mathbf{x}_{k+1} \\
&-2\mathbf{x}_{k+1}^T C_{v,k}^{-1}\Phi_k\left(\Phi_k^T C_{v,k}^{-1}\Phi_k + (P_k^k)^{-1}\right)^{-1}(P_k^k)^{-1}\hat{\mathbf{x}}_k^k \\
&-(\hat{\mathbf{x}}_k^k)^T (P_k^k)^{-1}\left(\Phi_k^T C_{v,k}^{-1}\Phi_k + (P_k^k)^{-1}\right)^{-1}(P_k^k)^{-1}\hat{\mathbf{x}}_k^k\ .
\end{aligned}
\tag{9.10}
$$

If we now apply the Sherman-Morrison-Woodbury identity on the first term we get

$$
C_{v,k}^{-1} - C_{v,k}^{-1}\Phi_k\left(\Phi_k^T C_{v,k}^{-1}\Phi_k + (P_k^k)^{-1}\right)^{-1}\Phi_k^T C_{v,k}^{-1} = \left(C_{v,k} + \Phi_k P_k^k \Phi_k^T\right)^{-1}
$$

and so Eq. (9.10) becomes

$$
\begin{aligned}
&\mathbf{x}_{k+1}^T\left(C_{v,k} + \Phi_k P_k^k \Phi_k^T\right)^{-1}\mathbf{x}_{k+1} \\
&-2\mathbf{x}_{k+1}^T C_{v,k}^{-1}\Phi_k\left(\Phi_k^T C_{v,k}^{-1}\Phi_k + (P_k^k)^{-1}\right)^{-1}(P_k^k)^{-1}\hat{\mathbf{x}}_k^k \\
&-(\hat{\mathbf{x}}_k^k)^T (P_k^k)^{-1}\left(\Phi_k^T C_{v,k}^{-1}\Phi_k + (P_k^k)^{-1}\right)^{-1}(P_k^k)^{-1}\hat{\mathbf{x}}_k^k =
\end{aligned}
$$

and after multiplying and dividing the first term with the appropriate expression and noticing that the third term is constant with respect to $\mathbf{x}_{k+1}$ we get

$$\mathbf{x}_{k+1}^T\left(C_{v,k}+\Phi_k P_k^k\Phi_k^T\right)^{-1}\mathbf{x}_{k+1}- \\ 2\mathbf{x}_{k+1}^T\left(C_{v,k}+\Phi_k P_k^k\Phi_k^T\right)^{-1}\left(C_{v,k}+\Phi_k P_k^k\Phi_k^T\right)C_{v,k}^{-1}\Phi_k\left(\Phi_k^T C_{v,k}^{-1}\Phi_k+(P_k^k)^{-1}\right)^{-1}(P_k^k)^{-1}\hat{\mathbf{x}}_k^k+const\ . \tag{9.11}$$

We notice that

$$\left(C_{v,k}+\Phi_k P_k^k\Phi_k^T\right)C_{v,k}^{-1}\Phi_k\left(\Phi_k^T C_{v,k}^{-1}\Phi_k+(P_k^k)^{-1}\right)^{-1}(P_k^k)^{-1}= \\ \left(\Phi_k+\Phi_k P_k^k\Phi_k^T C_{v,k}^{-1}\Phi_k\right)\left(P_k^k\Phi_k^T C_{v,k}^{-1}\Phi_k+\mathbf{1}\right)^{-1}= \\ \Phi_k\left(\mathbf{1}+P_k^k\Phi_k^T C_{v,k}^{-1}\Phi_k\right)\left(P_k^k\Phi_k^T C_{v,k}^{-1}\Phi_k+\mathbf{1}\right)^{-1}=\Phi_k$$

so Eq. (9.11) becomes

$$\mathbf{x}_{k+1}^T\left(C_{v,k}+\Phi_k P_k^k\Phi_k^T\right)^{-1}\mathbf{x}_{k+1}- \\ 2\mathbf{x}_{k+1}^T\left(C_{v,k}+\Phi_k P_k^k\Phi_k^T\right)^{-1}\Phi_k\hat{\mathbf{x}}_k^k+const= \\ (\mathbf{x}_{k+1}^T-\Phi_k\hat{\mathbf{x}}_k^k)^T\left(C_{v,k}+\Phi_k P_k^k\Phi_k^T\right)^{-1}(\mathbf{x}_{k+1}^T-\Phi_k\hat{\mathbf{x}}_k^k)+const\ = \\ (\mathbf{x}_{k+1}^T-\Phi_k\hat{\mathbf{x}}_k^k)^T {P_{k+1}^k}^{-1}(\mathbf{x}_{k+1}^T-\Phi_k\hat{\mathbf{x}}_k^k)+const$$

where

$$P_{k+1}^k=C_{v,k}+\Phi_k P_k^k\Phi_k^T$$

and $\Phi_k\hat{\mathbf{x}}_k^k$ would be $\Phi_k\hat{\mathbf{x}}_k^k+\Gamma_k\mathbf{u}_k$ if we hd not dropped $\Gamma_k\mathbf{u}_k$ for convenience. This means that $\mathbf{x}_{k+1}$, given the measurements up until time k has mean

$$\hat{\mathbf{x}}_{k+1}^k=\Phi_k\hat{\mathbf{x}}_k^k+\Gamma_k\mathbf{u}_k$$

and variance

$$P_{k+1}^k=C_{v,k}+\Phi_k P_k^k\Phi_k^T\ .$$

So after a couple of pages of mathematical scribbling we showed that the Bayessian approach gives us the right result.

Off to derive the second step. We want now to find the probability distribution for the next state given all the previous measurements, and we use the Bayess theorem for this:

$$p(\mathbf{x}_{k+1} \mid \mathbf{z}_{0..k+1}) = p(\mathbf{x}_{k+1} \mid \mathbf{z}_{0..k}, \mathbf{z}_{k+1}) = \frac{p(\mathbf{z}_{k+1} \mid \mathbf{x}_{k+1}, \mathbf{z}_{0..k}) p(\mathbf{x}_{k+1} \mid \mathbf{z}_{0..k})}{p(\mathbf{z}_{k+1} \mid \mathbf{z}_{0..k})} .$$

From the measurement equation

$$\mathbf{z}_{k+1} = \Lambda_{k+1}\mathbf{x}_{k+1} + \mathbf{w}_{k+1}.$$

we can infer that

$$p(\mathbf{z}_{k+1} \mid \mathbf{x}_{k+1}, \mathbf{z}_{0..k}) = p(\mathbf{z}_{k+1} \mid \mathbf{x}_{k+1}) \sim N(\Lambda_{k+1}\mathbf{x}_{k+1}, C_{w,k+1})$$

and from the first step we know that

$$p(\mathbf{x}_{k+1} \mid \mathbf{z}_{0..k}) \sim N(\hat{\mathbf{x}}^k_{k+1}, P^k_{k+1}) .$$

We do not need to worry about the expression in the denominator because its only purpose in life is to make $p(\mathbf{x}_{k+1} \mid \mathbf{z}_{0..k+1})$ a proper probability density, that is integrate to unity. We can achieve the same thing if we omit all the multiplicative constants and in the end, since we know the end result is a Gaussian the property will be satisfied automatically by virtue of Gaussianity. So omitting unnecessary constants

$$-2 \ln p(\mathbf{x}_{k+1} \mid \mathbf{z}_{0..k+1}) =$$

$$(\mathbf{z}_{k+1} - \Lambda_{k+1}\mathbf{x}_{k+1})^T {C_{w,k+1}}^{-1} (\mathbf{z}_{k+1} - \Lambda_{k+1}\mathbf{x}_{k+1}) + (\mathbf{x}_{k+1} - \hat{\mathbf{x}}^k_{k+1})^T {P^k_{k+1}}^{-1} \left(\mathbf{x}_{k+1} - \hat{\mathbf{x}}^k_{k+1}\right) =$$

$$\mathbf{x}^T_{k+1} {\Lambda_{k+1}}^T {C_{w,k+1}}^{-1} \Lambda_{k+1}\mathbf{x}_{k+1} + \mathbf{x}^T_{k+1} {P^k_{k+1}}^{-1} \mathbf{x}_{k+1} -$$

$$2\left(\mathbf{x}^T_{k+1} {\Lambda_{k+1}}^T {C_{w,k+1}}^{-1} \mathbf{z}_{k+1} + \mathbf{x}^T_{k+1} {P^k_{k+1}}^{-1} \hat{\mathbf{x}}^k_{k+1}\right) + const =$$

$$\mathbf{x}^T_{k+1}\left({\Lambda_{k+1}}^T {C_{w,k+1}}^{-1} \Lambda_{k+1} + {P^k_{k+1}}^{-1}\right)\mathbf{x}_{k+1} -$$

$$2\mathbf{x}^T_{k+1}\left({\Lambda_{k+1}}^T {C_{w,k+1}}^{-1} \mathbf{z}_{k+1} + {P^k_{k+1}}^{-1} \hat{\mathbf{x}}^k_{k+1}\right) + const .$$

It is obvious that

$$P^{k+1}_{k+1} = \left({\Lambda_{k+1}}^T {C_{w,k+1}}^{-1} \Lambda_{k+1} + {P^k_{k+1}}^{-1}\right)^{-1}$$

which is fine for most purposes but we want to arrive to the standard expression, so we use the Sherman-Morrison-Woodbury identity again and get

$$P^{k+1}_{k+1} = P^k_{k+1} - P^k_{k+1}{\Lambda_{k+1}}^T\left(C_{w,k+1} + \Lambda_{k+1}P^k_{k+1}{\Lambda_{k+1}}^T\right)^{-1}\Lambda_{k+1}P^k_{k+1} =$$

$$P^k_{k+1} - K_{k+1}\Lambda_{k+1}P^k_{k+1}$$

where K_{k+1} is the Kalman Gain as before and from which we get that

$$-2\ln p(\mathbf{x}_{k+1} \mid \mathbf{z}_{0..k+1}) =$$
$$\mathbf{x}_{k+1}^T \left(P_{k+1}^k - K_{k+1}\Lambda_{k+1}P_{k+1}^k \right)^{-1} \mathbf{x}_{k+1} -$$
$$2\mathbf{x}_{k+1}^T \left({\Lambda_{k+1}}^T {C_{w,k+1}}^{-1} \mathbf{z}_{k+1} + {P_{k+1}^k}^{-1} \hat{\mathbf{x}}_{k+1}^k \right) + const =$$

and multiplying and dividing by $P_{k+1}^k - K_{k+1}\Lambda_{k+1}P_{k+1}^k$ we get

$$\mathbf{x}_{k+1}^T {P_{k+1}^{k+1}}^{-1} \mathbf{x}_{k+1} -$$
$$2\mathbf{x}_{k+1}^T \left(P_{k+1}^k - K_{k+1}\Lambda_{k+1}P_{k+1}^k \right)^{-1} \left(P_{k+1}^k - K_{k+1}\Lambda_{k+1}P_{k+1}^k \right) \left({\Lambda_{k+1}}^T {C_{w,k+1}}^{-1} \mathbf{z}_{k+1} + {P_{k+1}^k}^{-1} \hat{\mathbf{x}}_{k+1}^k \right)$$
$$+ const =$$
$$\left(\mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1}^{k+1} \right)^T {P_{k+1}^{k+1}}^{-1} \left(\mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1}^{k+1} \right)$$

where

$$\hat{\mathbf{x}}_{k+1}^{k+1} = \left(P_{k+1}^k - K_{k+1}\Lambda_{k+1}P_{k+1}^k \right) \left({\Lambda_{k+1}}^T {C_{w,k+1}}^{-1} \mathbf{z}_{k+1} + {P_{k+1}^k}^{-1} \hat{\mathbf{x}}_{k+1}^k \right).$$

Again we want the standard expression which we can get by playing a little bit

$$\hat{\mathbf{x}}_{k+1}^{k+1} =$$
$$P_{k+1}^k {\Lambda_{k+1}}^T {C_{w,k+1}}^{-1} \mathbf{z}_{k+1} + \hat{\mathbf{x}}_{k+1}^k - K_{k+1}\Lambda_{k+1}P_{k+1}^k {\Lambda_{k+1}}^T {C_{w,k+1}}^{-1} \mathbf{z}_{k+1} - K_{k+1}\Lambda_{k+1}\hat{\mathbf{x}}_{k+1}^k =$$
$$\hat{\mathbf{x}}_{k+1}^k - K_{k+1}\Lambda_{k+1}\hat{\mathbf{x}}_{k+1}^k + \left(P_{k+1}^k {\Lambda_{k+1}}^T {C_{w,k+1}}^{-1} - K_{k+1}\Lambda_{k+1}P_{k+1}^k {\Lambda_{k+1}}^T {C_{w,k+1}}^{-1} \right) \mathbf{z}_{k+1}$$

and we notice that if we substitute K_{k+1} with its value from Eq. (9.5), the factor multiplying $\mathbf{z}_{k+1}$ can be written as

$$P_{k+1}^k {\Lambda_{k+1}}^T {C_{w,k+1}}^{-1} - P_{k+1}^k \Lambda_{k+1}^T \left(\Lambda_{k+1} P_{k+1}^k \Lambda_{k+1}^T + C_{w,k+1} \right)^{-1} \Lambda_{k+1} P_{k+1}^k {\Lambda_{k+1}}^T {C_{w,k+1}}^{-1} =$$
$$P_{k+1}^k \Lambda_{k+1}^T \left(\Lambda_{k+1} P_{k+1}^k \Lambda_{k+1}^T + C_{w,k+1} \right)^{-1} \left(\left(\Lambda_{k+1} P_{k+1}^k \Lambda_{k+1}^T + C_{w,k+1} \right) - \Lambda_{k+1} P_{k+1}^k \Lambda_{k+1}^T \right) {C_{w,k+1}}^{-1} =$$
$$P_{k+1}^k \Lambda_{k+1}^T \left(\Lambda_{k+1} P_{k+1}^k \Lambda_{k+1}^T + C_{w,k+1} \right)^{-1} \left(C_{w,k+1} \right) {C_{w,k+1}}^{-1} =$$
$$P_{k+1}^k \Lambda_{k+1}^T \left(\Lambda_{k+1} P_{k+1}^k \Lambda_{k+1}^T + C_{w,k+1} \right)^{-1} = K_{k+1}$$

so

$$\hat{\mathbf{x}}_{k+1}^{k+1} = \hat{\mathbf{x}}_{k+1}^{k} - K_{k+1}\Lambda_{k+1}\hat{\mathbf{x}}_{k+1}^{k} + K_{k+1}\mathbf{z}_{k+1} =$$
$$\hat{\mathbf{x}}_{k+1}^{k} + K_{k+1}\left(\mathbf{z}_{k+1} - \Lambda_{k+1}\hat{\mathbf{x}}_{k+1}^{k}\right)$$

which after shedding so much of the holy blood of the photocopier we rederived the Kalman equations this time in a Bayessian fashion.

9.5. Algorithm for Kalman Filter

Putting everything together, we start from the state transition equation and measurement equation

$$\mathbf{x}_{k+1} = \Phi_k\mathbf{x}_k + \Gamma_k\mathbf{u}_k + \mathbf{v}_k$$

and

$$\mathbf{z}_{k+1} = \Lambda_{k+1}\mathbf{x}_{k+1} + \mathbf{w}_{k+1}.$$

We are given the matrices Φ_k, Γ_k and Λ_{k+1} as well as $C_{v,k}$ and $C_{w,k+1}$ the variance covariance matrices of the noise terms. We are also given the starting points $\hat{\mathbf{x}}_0^0$ and P_0^0. Then at every time instance k we compute

$$\hat{\mathbf{x}}_{k+1}^{k} = \Phi_k\hat{\mathbf{x}}_k^k + \Gamma_k\mathbf{u}_k$$
$$P_{k+1}^{k} = \Phi_k P_k^k\Phi_k^T + C_{v,k}$$
$$K_{k+1} = P_{k+1}^{k}\Lambda_{k+1}^T\left(\Lambda_{k+1}P_{k+1}^{k}\Lambda_{k+1}^T + C_{w,k+1}\right)^{-1}$$
$$P_{k+1}^{k+1} = P_{k+1}^{k} - K_{k+1}\Lambda_{k+1}P_{k+1}^{k}$$
$$\hat{\mathbf{x}}_{k+1}^{k+1} = \hat{\mathbf{x}}_{k+1}^{k} + K_{k+1}\left(\mathbf{z}_{k+1} - \Lambda_{k+1}\hat{\mathbf{x}}_{k+1}^{k}\right)$$