Foundations of Cyber Physical Systems

Prof. Soumyajit Dey

Department of Computer Science and Engineering

Indian Institute of Technology - Kharagpur

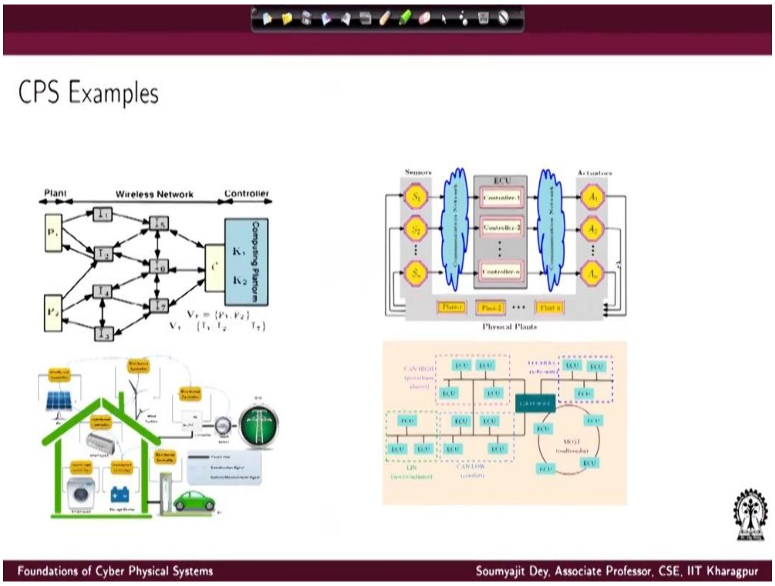
Module - 11

Lecture - 50

State Estimation using Kalman Filters (KF)

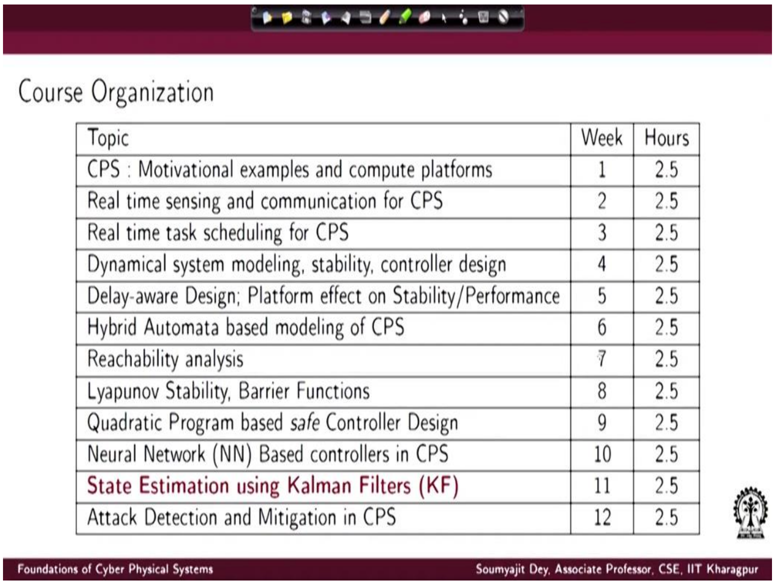
Hello and welcome back to this lecture series on Computational Foundations of Cyber Physical Systems. So, this week we will be starting with Kalman filters, how they are used for state estimation in cyber physical systems and several other applications also exist for Kalman filters. So, just to uh have some recap uh

(Refer Slide Time: 00:49)



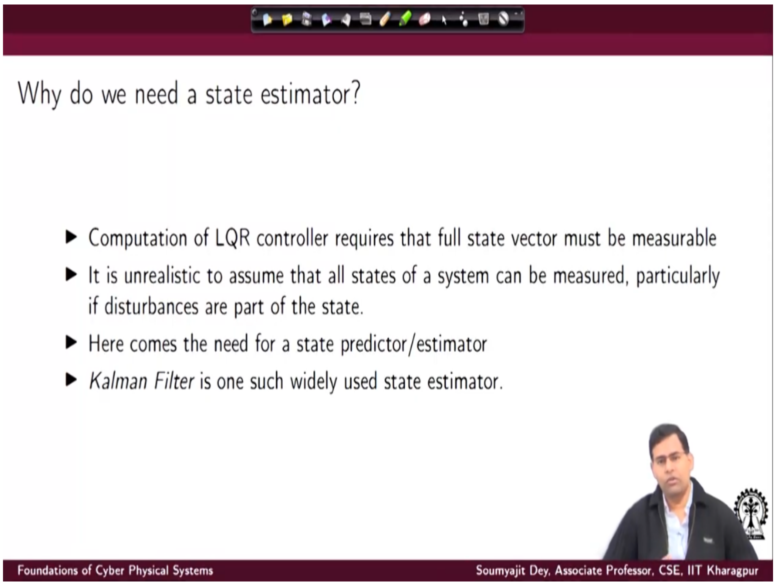
what we have here is uh well, this was my standard example of cyber physical system examples. Right.

(Refer Slide Time: 00:54)



And uh so, this is, this was our course uh organization, right, so, till date we have we have actually covered all these topics up to week 10 and we have learnt about basic control design, how to model the formal methods part on hybrid automata-based modelling and reachability analysis. We have learnt how to analyze complex non-linear CPS using Lyapunov stability and barrier functions, how to design optimal controllers using uh quadratic expressions and stuff like that, and how to how to actually use neural network-based controllers also.

(Refer Slide Time: 01:34)

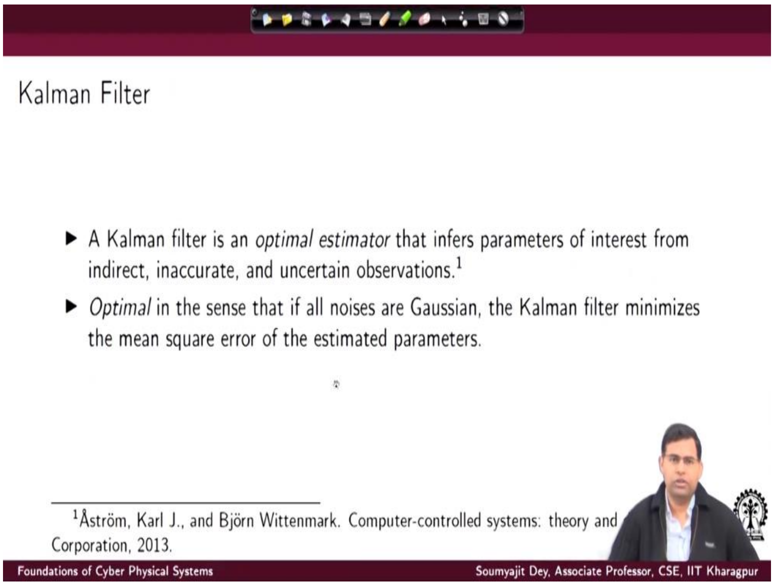


So, uh what, one important thing that we did not talk about there is, (refer time: 01:34) well, if when we speak of optimal controllers like LQR and several other kinds of controllers like model prediction and others, so, they are based on optimization functions and we have shown how such optimizations can be solved in real time. But in many cases we also assume that the full state of the system is uh known to the controller but it is unrealistic. So, it is also possible in many cases that the full state is not known.

What you have access to is just a set of measurements about the system. So, this is precisely where you need to understand that well, from such measurements how one can actually estimate the state and this is where the role of observers or predictors and estimators will come in. And Kalman filter as it happens is one such widely used state estimator. So, this was invented by professor Rudolf E. Kalman long back and it has still till till date it has found widespread application into domains of control and several other interesting electrical engineering discipline oriented domains and so we, I mean, we are just trying to give you a mathematical derivation of Kalman filter here and we are just trying to show you how it is to be put in practice.

Let us remember that uh this is a quite involved topic. There are there are lot of deep mathematics and from a complex foundations of this topic but we are just taking a simple linear algebra approach and we are trying to figure out well, how the basic Kalman filter equation can be derived. So, that it will be our approach here during this coverage.

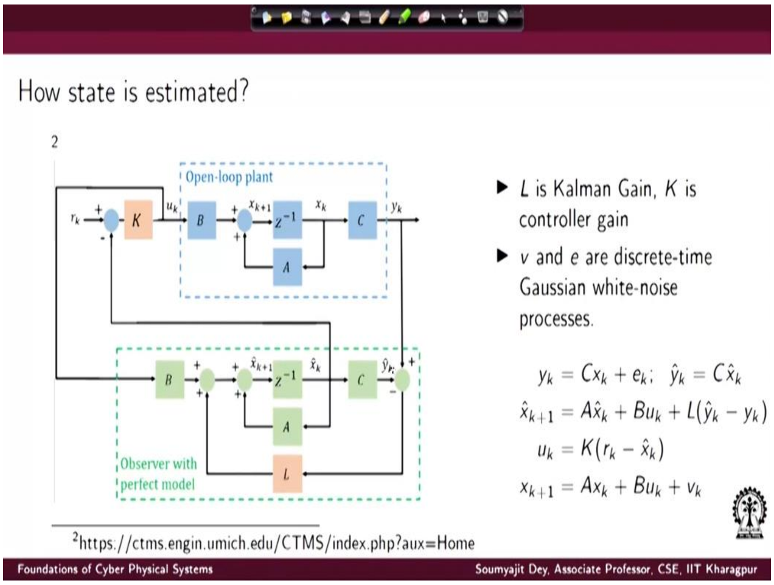
(Refer Slide Time: 03:11)



So, it is an optimal estimator in case of assuming Gaussian noises, right, and it is trying to infer parameters of interest from inaccurate noisy observations, okay.

And when we say noisy observations, we are assuming that the noises are Gaussian, okay, and it will minimize the mean square error of these estimated parameters. Due to such Gaussian model noises, whatever is the error coming out in a mean square form, the the an expression will be created and we will try to figure out what is the exact filter which minimizes that expression. So, that that is the basic approach here.

(Refer Slide Time: 03:48)



So, that that is the (refer time: 03:48) basic approach here. So, in general, how is state estimated and how are such state estimates put into practice? So, typically, that is how the control works. So, suppose let us assume that you have designed, you have indeed designed such a Kalman gain or a Kalman estimator, okay, and your original controller is having some gain K, okay, so, typically this is how you put it into practice. You will have the open loop plant.

So, this is your plant. You can see that this is in the form of Ax plus Bu, right, x_{k+1}, the update is happening based on Bu plus Ax here, right, and you are getting the measurement y equal to Cx, and this measurement comes to your Kalman gain, right, and so this is part of the observer circuit here.

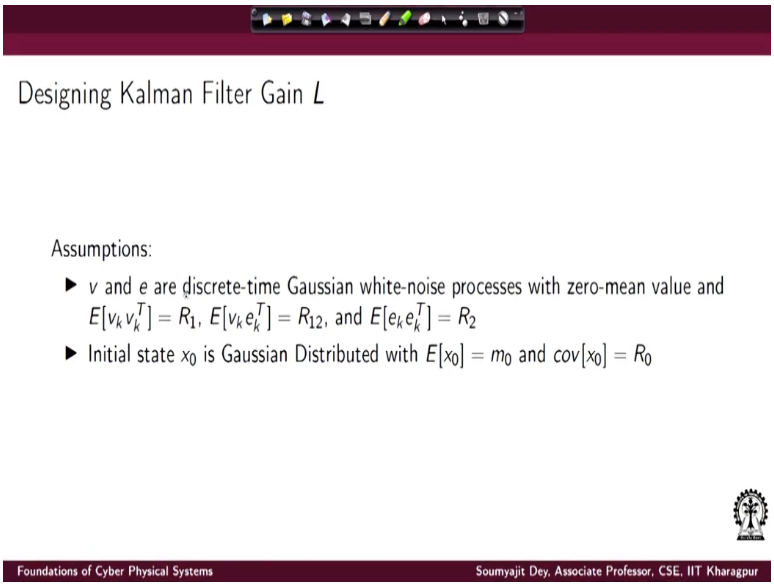
And what it is, this circuit is, is entirely working on the estimates of the system like you can see. So, if this is the estimate x_{k+1} based on the previous estimate, then what we can write is well, x_{k+1} hat is equal to Ax_k hat plus Bu_k. Okay. uh So, x_k plus x_k A x_k hat plus B u_k plus L multiplied by this difference that is the error uh between y_k and the measurement estimate that the observer can generate, okay. So, L y_k minus y_k hat minus y_k, where this u_k that we have, uh what is, I mean, what really is it?

So, this u_k is a in this case u_k happens to be nothing but this controller gain K times the reference r minus uh x, this estimate of x here. Okay. In case your reference is 0 in this place for the controller, so, you simply have a standard equational form that is minus k x_k; but instead of minus k x_k, you are having minus k x_k hat. So, that is the more practical scenario because we cannot expect my controller like I was saying that it is going to have access to x_k, right?

So, for all practical purpose, this controller will have access to x_k hat and this observer is the system which is responsible to generate these estimates x_k hat recursively in each step based on its uh its sampling of the measurement y_k. So, this is what you observe and through this observer, this is the estimate you generate. And this observer as you can see it is kind of mimicking your original system through the structure of A's and B's, right, but along with that, what you have is L which is the Kalman gain.

The the entire purpose of this gain is to kind of minimize this difference, right. It will it will it will try to figure out a nice estimate of x so that based on x_k's estimate if I derive a y_k's estimate, the difference between y_k hat and y_k must be very small. Okay. So, that is that is how this entire equational setting comes in. So, this is my observer as you can see and it is generating x_k hat. And like I said, for all practical purpose this is, x_k hat is what with the controller will take as input, is because it will ideally not have access to the x_k's. Okay.

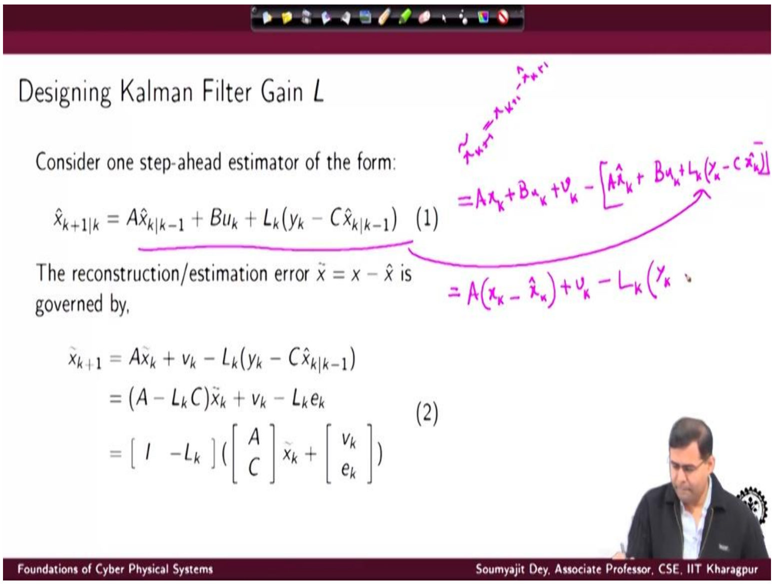
(Refer Slide Time: 07:10)



So, uh we will have some assumptions here, like uh you will have uh noises disturbing the system. For example, for the measurements, you see y_k equal to C x_k. You are adding some noise here, e_k, right, so, uh this e_k is a measurement noise. And similarly if you see, this is your revolution equation of the state and for this evolution of uh x_k, uh you have also added one noise term that is v_k. So, that is your process noise. So, you are considering this v and e the process and measurement noise as discrete time Gaussian white noise processes who have some 0 mean and uh their variances are like this.

So, you have the variances and covariances given by these terms. So, variance of process noise is R_1; variance of uh measurement noise is R_2 and their mutual covariance is R_{12}. Also you will have the initial state of the system x naught assumed to be Gaussian distributed with some mean m_0 and a covariance value R_0. Okay.

(Refer Slide Time: 08:19)



So, uh so, overall, from this picture of uh actual deployment that how the system is really going to work with an observer that is generating these estimates, uh I can write this update equation, right, uh because you see this is how I I am able to generate x_{k+1}'s estimate using A x_k hat B u_k and I am expecting the Kalman gain to do a good job so that it does try to minimize this difference.

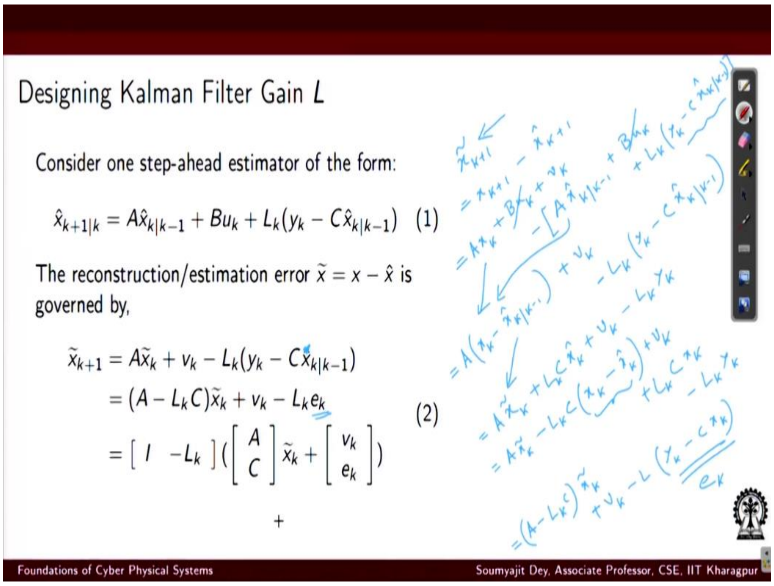
So, whatever is this difference between its measurement that is the sampling and the measurement that it estimated based on previous measurements, so, this residue value I am trying to think that well, it is doing a good job so that this measurement measurement error is as small as most possible here. Okay. So, uh based on this uh we can write this update equation like I said. So, this notation let me explain. So, what it this means is that well, uh based on the kth step measurement, we are trying to estimate x in the k plus 1th step and it can be written through these equations where I can derive this using the previous estimate of k x in the kth step which was generated using measurements that were observed in the k minus 1th step, right.

So, that is how I write this equation. And now, what do I have? I have 2 sets of equations, right, because I have this equation which is trying to uh kind of uh model the evolution of the state estimate. And as you can see, this idea of Kalman filter gain is coming as a multiplication factor or the gain corresponding to the residual because this is my original measurement and this is my estimated measurement and that gives me the residual and I am using this to be, I mean, to be handled by the filter here and the rest of the dynamics is here or that was there originally and with this I am generating the estimate, right.

So, this estimate and my original state, if I just subtract from the original state the estimate, what I can write is a reconstruction error, right. So, let x tilde denote the reconstruction error. So, essentially, uh I can write something like this, right. I mean, well, how it comes is very simple. Let me just correct the orientation of the writing here. So, if I just apply the first equation for the state update and then I have this equation here, okay. So, with these two. So, uh as we were discussing. Let us look at the situation that well, how I can really construct the reconstruction error here.

So, uh we already have these equations of uh the next state with respect to the system model as well as uh using the estimate with respect to the estimator, right. uh So, let us proceed from there.

(Refer Slide Time: 13:16)



So, this is a reconstruction error which is, minus the estimate. Now, if we try to do this math here, so, we will reproduce the original state update equation. Here goes the processed noise. Now we will replace this estimate by the update equation of the estimator.

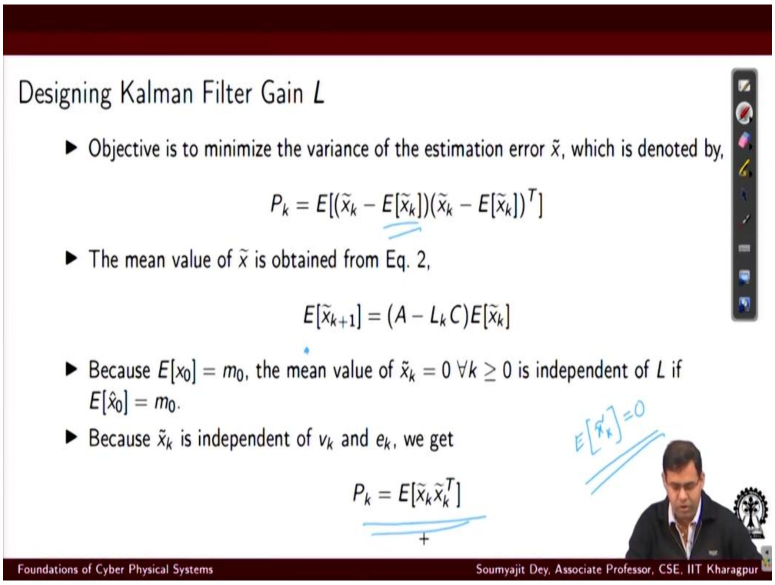
Here, for the time being, so, reproducing this thing here and the estimate of the previous step goes in here. So, that is pretty much it. So, this is the residue which gets multiplied by this Kalman gain which we are supposed to design. So, if you now simplify this equation, let us see what we get. So, of course, these two will get cancelled out and these two gets combined in here and you are left with the Kalman gain multiplied by the residue term. So, now you see, if this is your reconstruction error in the k plus 1th step, then the reconstruction error in the kth step is this, right.

All right, so, we are doing something like uh, so, this is same and the process noise remains, and as you can see, we have added a term L_k C x_k and here and again subtracted it to create this form here so that now if you see, this is again kind of the reconstruction error. So, I can write this thing, I can write this entire dynamics of the reconstruction like this. So, that is the intention, okay, for why I am writing it in that way. So, this is how this derivation comes, right.

So, as you can see, uh this is what we have here, right, A x_k tilde plus v_k minus L_k y minus uh this C_k here. This is, this is the x actually. So, if you recall, this is nothing but my measurement noise here, right. So, that is how we defined uh the measurement noise. So, that is why we can start writing it like this through the e_k. And then, if I write it in a suitable matrix form, I can write it like this, right, x_k uh plus 1 tilde equal to uh I, the identity matrix minus L_k.

In a vector form I have A and C in terms of. So, x_{k+1} is getting expressed in a dynamical equation using x_k with the additive of, additive term of the 2 noise values, right. So, that is like a noise matrix vector representation of the dynamics of the reconstruction here. So, this is what we intended to do here all the time. So, that is the derivation and that is how uh it is happening. So, just to recall, yeah, this is where we had e_k being defined, right. So, fine, uh just recall that this is; yeah.

(Refer Slide Time: 20:50)



Now, if we just go forward, (refer time: 20:50) so, let us understand what we are really trying to do.

So, we, I mean, we have set up a template equation here, right, an equation which is kind, which is which is set up with the objective that it will give me an estimate of x based on some previous estimate and the estimate should be defined in such a way that it minimizes the error between the estimate and the original value, right. So, the idea is that we will we will define an objective function which will minimize this estimation error here uh and the condition of the minimization should be such that, that will give me this uh gain value L. Okay.

So, that is how the Kalman filter is indeed defined here. So, this is my objective then. I want to minimize the variance of the estimation error, right, because it is an error, right, it can be positive or negative. So, in the long run, I want to do a least square minimization, so, I am going to minimize the variance of this estimation error which is a x_k tilde, right? I hope the objective is clear. So, just to recall that I am trying to minimize this value, absolute magnitude. Now, this can be positive or negative.

So, just like the standard technique we have in combinatorial optimization, we will try to do a least square minimization here. In effect, that means I am going to minimize the the the variance of the estimation error, right. So, the formula for variance as we all know is the estimator estimation of uh this expectation over this uh this difference, right, of this value of the variable minus the mean multiplied by these things transpose, right. So, basically, that is how we we are going to always write an square term when we are doing it over a matrix.

So, uh in this way we define this P_k, right, I mean, we are trying to minimize P_k here at the kth step and the definition of P_k is the variance of the estimation error and this is the standard formula for variance. Now, uh the mean value of x, uh if I try to obtain it, so, we can check out the equation. So, if you take this equation and if you take uh the expectation on both sides, then uh well, this is what you are going to get, right. uh From this this equation, take expectation on both sides e of x_{k+1} tilde should be, uh where these are all constant, A minus L_k C multiplied, then expectation of x_k tilde.

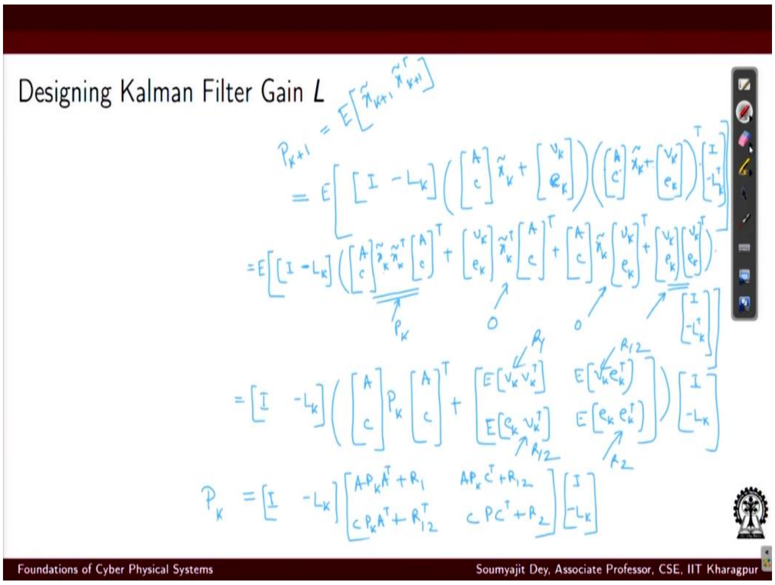
And since uh the expectation will also be of this and this and they are all 0 because these are v_k and e_k are all thus as as per our definition, these are all 0 mean Gaussian noises, right. So, in that way, this is what we get. Now, observe one thing, we we assume that the initial state x_0 that we have here, the initial state x_0 is Gaussian distributed and it has a mean m naught, right. Now, given this uh x_k, let us see how is x k evolving. So, the way x_k is evolving is that it is a linear transformation over a series x_{k-1}, x_{k-2}, like that, right.

So, that would really mean this reconstruction error x_k tilde, right, I mean, since the initial value is Gaussian distributed and we are essentially doing linear transformations over this multiple sequence of linear transformations over this initial value which is Gaussian distributed, we would expect that this reconstruction errors mean should be 0, right. I mean, uh this reconstruction error will have a 0 mean, right. So, that is why you can always write this thing that expectation, once you are going to take it over uh x_k tilde at 0, right. So, this term will become 0.

I can always write this as 0 because of this Gaussian property followed by linear transformations. I mean, physically speaking, if you think the error will be positive and negative around the 0 mean value and that is how this is getting set to 0, so, then, all that will remain is you want to figure out uh this thing. What is the expectation of x_k tilde x_k tilde transpose? Okay. So, that is the, with the 0 mean of this variable, what you are left to do is check the variance and the variance will be given by expectation of this.

So, let us now then go about designing this thing. So, first thing we will do that uh well, I have to minimize this thing. So, let us first create a dynamical equation of P_k. That would be my first step here.

(Refer Slide Time: 25:48)



So, you see, just like we have P_k plus P_k, I can simply write P_{k+1} something like this, right. Right. So, this would give me. So, this is the vector form we have already designed for x_k tilde, right. And then it is the transpose, so, that would really mean this thing, this same quantity with a transpose. So, or we can just write it down here, A C x_k tilde followed by a transpose of this. That means this row representation would get changed to a column representation.

Along with that, this quantity would have a transpose, right, and the bracket closes. So, this is what we are looking at. So, now, so, this I and the second term is minus L_k. Do not think this is I minus L of course. So, essentially, you have, you can, you are going to now multiply these terms inside and that would generate multiple terms here, a multiple additive terms. So, if we just write them down. See, you have a quadratic term, right, it has a nice form, some matrix followed by something, so, this would be a scalar quantity and then this original matrix transpose.

So, that is a quadratic form. Then at the end you have; yeah. So, you have got these 4 terms, all in a; so, you have two of them are in a quadratic form like this and then you have the other 4 terms and overall this is getting multiplied by this. So, now what you can do is you can carry the expectation inside, right, because then this expectation will be applied on here and that would if you have applied on this e applied on here, then it would generate the term P_k.

And similarly you see this term, the expectation would be applied here, right, because the others are constant, I mean, these are all 0 means, right, and so uh this would give you a 0. similarly, this would also give you a 0, right. You have the expectation of this and the expectation of this that would remain. So, then what we can write would be something like this, right. The expectation is applied here. So, you get P_k and then you have things here.

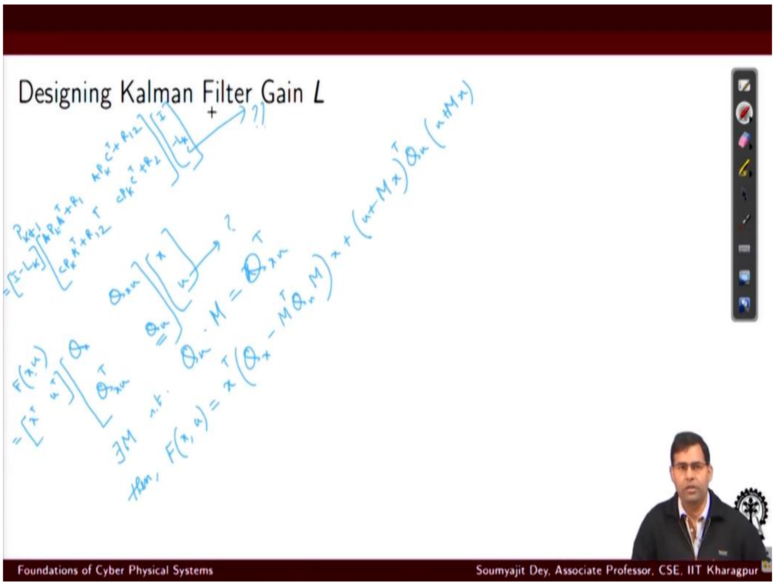
So, you will have 4 terms when you are you are you have a 1 cross 2 and this transpose will give you a 2 cross 1 matrix. So, you have you will in effect have a 2 cross 2 matrix and the expectation should get inside here, right. So, what you will get is something like this. Yeah. So, this is the final thing that I get and of course if you remember, we have already designated these things, these covariance terms. So, this is the variance of v_k and this is the variance of e_k and this is the covariance of v_k and e_k which we have all defined earlier.

So, this is R_1, uh this was our R_2 and these two are our R_{12}'s, right. So, we can just write this thing as in a nice form which would be like. So, if we just break down these expressions, we will get stuff like this. So, essentially, this will be getting multiplied, this will be generating 4 terms and they are just row wise going to be added, right. So, so, here, the first term will be getting added to the first term. This is the first term of the product here, right, and plus R_1, right.

Similarly, the last term would be C P; here the term would be C P_k A transpose here, right, plus this is your R_{12}. These are all defined earlier, the covariances. And similarly here and here, the, this position term is C P_k C transpose. So, this would be your dynamical update expression for P_k here, right. So, fine, we have got a dynamical update equation for P_k and looks like it is in a in a quadratic form because well, here at the start, whatever you have, here you have the same thing in a transposed way.

And this quadratic equation has a specific interesting form which is already known in optimization theory and we will try to use it up to carry on with our uh our goal of coming up with a solution to this. So, uh looking at this, uh let me just recreate the equation again on the side.

(Refer Slide Time: 34:52)



So, what you have is P_{k+1}. So, this is what you have. And if you see, this is a specific form which is like this F(x,y), a function uh which looks like. So, you see, this is what you have like you have a form like this, right, because if you see this and this quantities are transpose of each other.

The covariances are, I mean, so, here you have a transpose, right, so, I mean, these are all transposes of each other, so, that works out, right. So, we are trying to find a well-known form and we are saying that this dynamic update equation conforms to a well-known form and what it what it is known is we want to minimize this with respect to u here, this form, right, just like we want to minimize these variance of the error with respect to L_k. We trying to, we are trying to identify what is the gain such that this error variance is minimum, right.

That is what we, that is what the other target here. Now, what is known from this mathematical optimization theory is that now if I am trying to minimize this form F(x,y) with respect to this variable u, uh uh I mean, if there exists some M, if there exist some M such that I can write this Q_u, this thing as multiplied by, I mean, Q_u times M to be equal to, Q_u times M to be equal to Q; sorry again, Q_{xu} transpose, if I can write something like this, then the form of this function becomes. So, this is not y, this is u of course, there is no y here in this function.

Then, well, F(x,u), so, what we are really saying here is that this dynamic equation is of this form and we are trying to minimize this thing with respect to this u, okay, and if I if I am doing applying the completion of squares technique, then uh under the situation that this kind of an M exists, like if I multiply xu by M, I am getting this term, if that happens, then I can simply write this up in this form. So, you see, that can that can happen, right. So, you will just need to replace this with this form.

And if you simplify, something like this would come, right. So, this this comes. And now if you see, uh this is a special form and we will be trying to figure out that under this assumption that if the M exists, what is the value of u so that this form is minimized. Because then, if we can cast the same problem here because it is of the same structure, we can also find out what is the value of L so that P_k P_{k+1} is minimized. Okay. So, fine, we will end this lecture right here and in the next week we will start from here. Thanks for your attention.