Foundations of Cyber Physical Systems

Prof. Soumyajit Dey

Department of Computer Science and Engineering

Indian Institute of Technology - Kharagpur

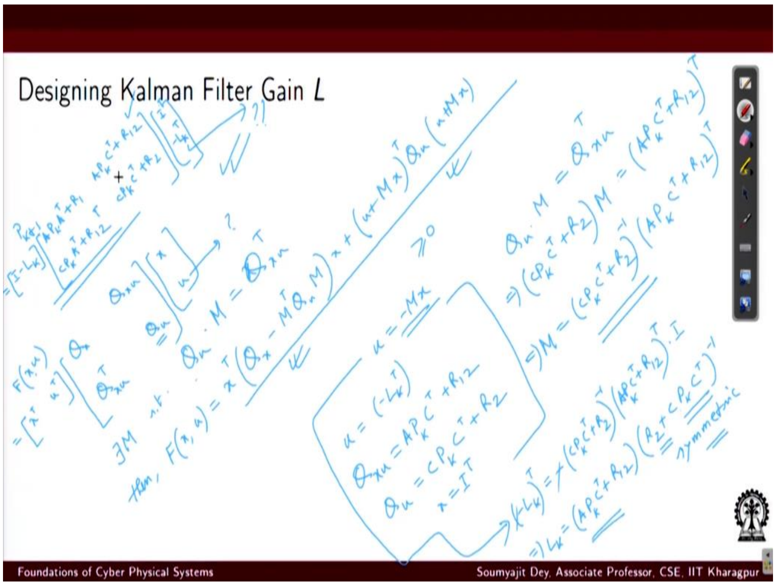
Module - 11

Lecture - 51

State Estimation using Kalman Filters (KF) (Contd.)

Hello and welcome back to this lecture series on Foundations of Cyber Physical Systems. uh If you recall, in our last lecture we were trying to deduce an expression for Kalman gain, so, maybe we can start right from that reduction that we are going on. Right.

(Refer Slide Time: 00:42)



uh So, if you recall, we created a reconstruction error dynamics and then we were saying that well, the expected value of the reconstruction error, I mean, the, I mean, basically the expectation over the variance, why because this error can be positive or negative, so, we will apply the classical least square minimization technique.

So, that would really mean we will try to uh minimize the variance of that x_k tilde term. So, before that, what we did is we created a dynamical equation here uh which represents uh how this reconstruction error is changing, right, plus through the previous, in the previous slide we have already done that, right, and that that derivation tells me that well, P_k is the reconstruction error in the previous row 1 and then what is the P_{k+1}. So, basically the, this is the variance of the reconstruction error in the kth step and the variance of the reconstruction error in the k plus 1th step, how do they relate?

We have figured that out. Now we are trying to say that well, we want to figure out what is the value of the Kalman gain such that this variance of the reconstruction error would get minimized. So, as we can see, uh this dynamical equation tells me that how this variance in the kth step is related to, you can help me to compute the variance in the k plus 1th step. So, in that way, I can get a recurrent relation and that gives me a series of values of P_k and I can see these variances value, how that is evolving.

Now I want to tell how that will be minimized. So, for doing that, we use a special functional form which is this F(x,u), okay, and we are saying that well, this looks like to be in this form, right, and the known result is well, for this functional form, for this functional form, uh if such an M exists through which I can relate this Q_u and Q_{xu} terms, right, through this relation, then, of course, I can represent F(x,u) in this kind of a nice expression. Okay. Now observe something here. In this expression, uh you have a quadratic term. Okay.

Now, this expression is quadratic in u and the both and the both of the terms are greater than 0. If you see, this term is also quadratic, x transpose then something and x and this term is also quadratic, right. So, uh it is kind of that both of the terms are greater than equal to 0, right? So, uh how can this be minimized. Then the only option that we have is it will be minimum when u is equal to minus M x, right. Now, if under the situation that if we can, if we are always we are able to uh generate this thing in a way that Q_u and Q_{xu} will be related by this kind of a linear coefficient, then we can always now say that since the resulting expression is having all quadratic positive terms, then the only way to reduce it would be to set u equal to minus M x.

And if we now apply this well-known combinatorial theory result to our update equation on the variance of the reconstruction error. So, we have this update equation which is relating the variance of the reconstruction error in the kth and k plus 1 h step. So, I can say that well, how will the variance be minimized? So, sim in a similar way, I can say that the variance will be minimized in that setting uh and that would mean that well, uh you replace u uh with the corresponding uh L_k term. So, what is really your u and L_k term, let us see here.

So, this was a transpose actually uh because this is in this is basically a transpose of the first term; so, this is a transpose. So, that means u is basically nothing but minus L_k transpose and then, right, and what are the other terms here? We have to see, right. So, this is what is your u and what are my other terms from the original expression, let us just write them down first. So, Q_{xu}, this term is this thing, right. uh Q_{xu} is this term, A P_k, the variance of the reconstruction error in the kth step, Q_u is this term. And x is your I transpose, that is I only, I mean, yeah, I mean, in the transpose manner of course, so, because uh it is changing from this to the other format, so, of course it should be like this.

Now if we apply these things, these substitutions, right, if you apply these substitutions, first into this requirement that Q_u times M should be equal to Q_{xu} transpose, right, so, if you apply there, you prove, you first put the value of Q_u, so, we know what is uh x Q_{xu}, right. So, let us take the transpose of that. So, it is this. That would mean. This is fine. This is just coming from here. This is just coming from here. And all these values, you are just correlating here and you are just substituting here. And then you you need an M to exist, so, that is why in this relationship you are now substituting these values, right.

So, I am just recapping here. So, your x transpose is I's or x is I transpose or that is same as I, no problem; and u transpose is your minus L_k or u is minus L_k transpose; so, so, just uh you substitute those values and this is what you have. Now, that means we have to have this inverse to exist and as long as that does which is fine, uh you will have an M. Okay. So, as long as you get the M, you can now simply put that value of M here and you can get your value of L k. So, let us see.

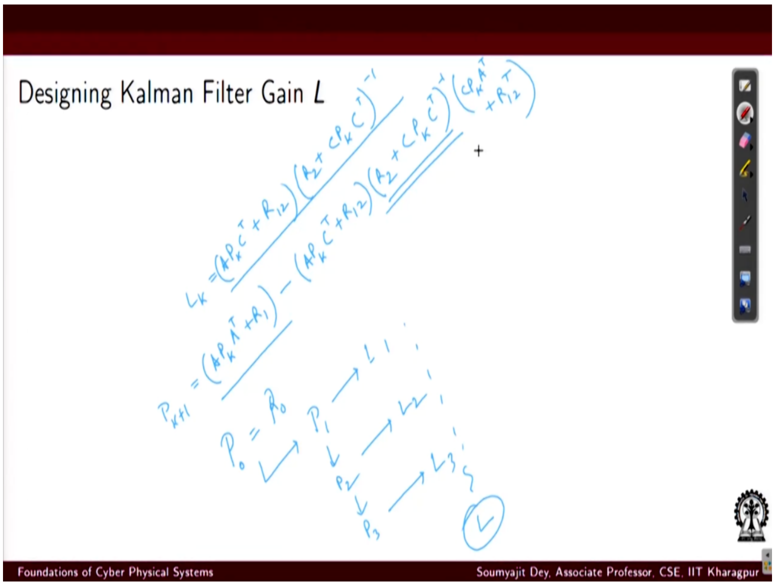
So, L is minus L_k transpose, u is minus L_k transpose and minus of, now push your value of M here and your x is I or I transpose, the same thing. So, this is what you have, right. So, now if you write L_k, so, you have to take a transpose on both sides. That would simply mean this thing double transpose and its comes to the forward. So, A P_k C transpose plus R_{12} and on this thing you have to again carry out a transpose. Now these covariances are such all the same. So, that is what you have.

Now of course uh, the reason we are able to write this is, this is symmetric and also this is symmetric, so, they are same under transpose. I hope we will see that we are taking a transpose on both sides, right, so, this comes to the front and this transpose cancels out and this goes to the back and again you apply a transpose here but since these are noise covariances and this C if you recall, this is also a symmetric matrix, so, and this C P_k C transpose, so, this form, this is a quadratic form, not C, but this quadratic form will also be symmetric and so they remain invariant under the transpose operation.

So, that is what we have, right. Now if you insert the. So, in that way, what you get is a uh expression of the Kalman gain, but observe one thing, this expression is here in terms of the variance of the estimation error in terms of P_k, right, but that is that is not something we really want, right. Eventually, we want just the Kalman gain. So, how do I get a value of Kalman gain? So, that means it is related to P_k and I have to relate it with the update equation of P_k also.

So, if I insert this value of L_k into the previous equation of P_{k+1}, right, you had this update equation of P_{k+1} and here you had an unknown L_k. So, you push this value to this expression and what you will get. Now let me clear this thing and write cleanly.

(Refer Slide Time: 12:36)



What you will get is something like this. So, first let me write down the Kalman gain. So, L_k and we put this value in the update equation of P_{k+1}. So, then the reconstruction error dynamics can be written with this value as something like this.

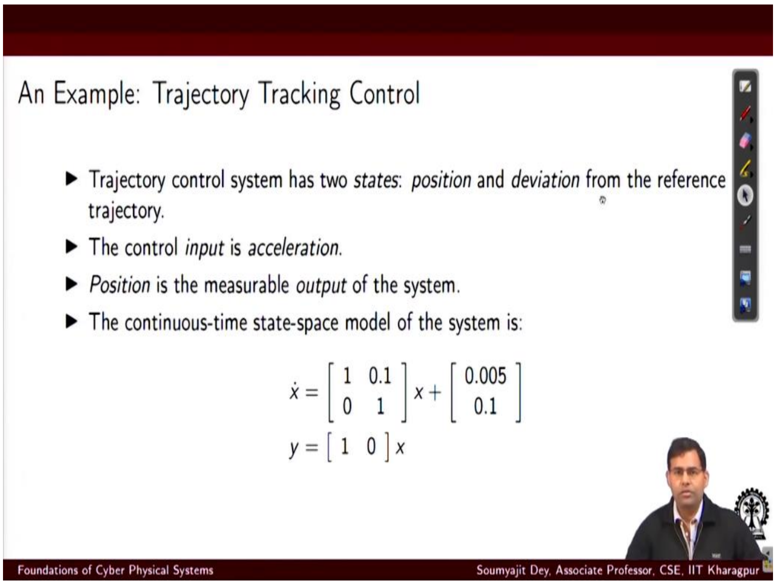
So, this is what you have, P_{k+1} update equation, you can now express in one line with this term and this. So, as you can see here, again you have this quadratic form, right, with A P_k C transpose plus R_{12} and its transpose coming up again here with C P_k transpose plus R_{12}

transpose and in between you have this, right. So, uh the thing is you have 2 recurrent equations here just like our previous expressions for LQR. uh What we have here is, well, you know the initial value and the initial mean value, right, so, you can start with P_0 as R_0 some estimate here and then you can calculate P_{k+1}, right.

So, with this you can calculate P_1. With this uh you can calculate L_1, right. Now, with P_1 you can calculate P_2; with this you can calculate L_2 and you can proceed in this way and eventually what you are expected to see is that this L sequence will kind of kind of get to a fixed point, right. When the gains converge, that is what the final Kalman gain value L is what you will get. Okay. So, this is your Kalman filter equation and this subsidiary equation is going to give you this error dynamics.

And uh what MATLAB does for you is, MATLAB solves it in one step by doing this convergence internally.

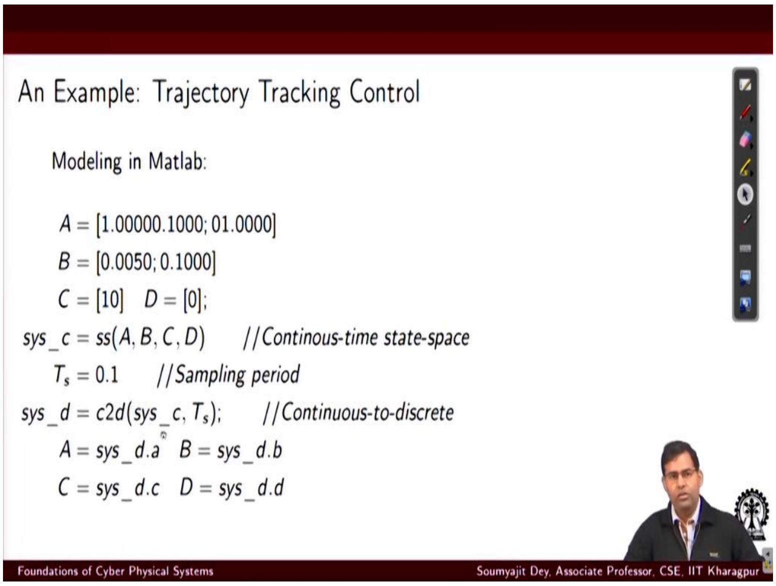
(Refer Slide Time: 15:45)



So, let us take an example here. Let us take this example of a trajectory control system. It has got 2 states, position and deviation from reference and let us say your control input is acceleration and position is the measurable output of the system. So, not all states are feasible, like deviation, we are not checking here. And the continuous state space system model is this.

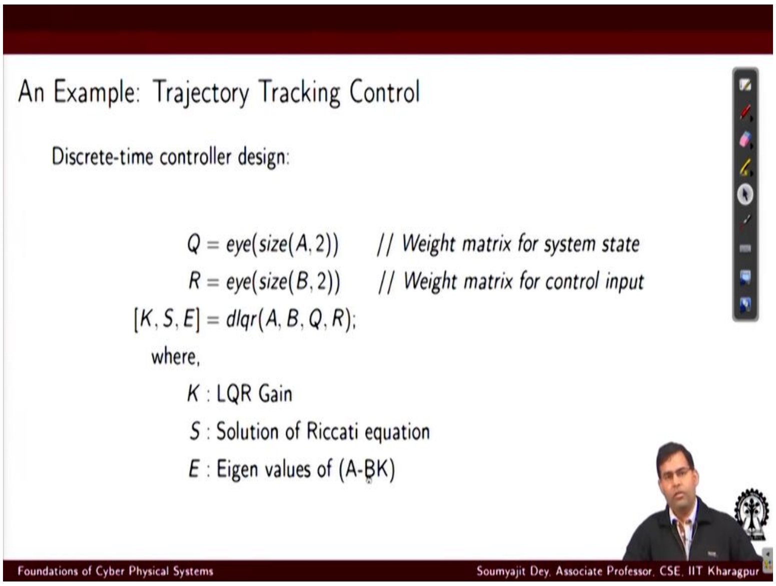
uh It is a, this is a well-known example you can get in books like Angstrom's book and other books which you have referenced earlier.

(Refer Slide Time: 16:14)



And if you create the, if you just create this model in MATLAB, so, you can code it up like this, you can define the A,B,C,D matrices and then you can generate the continuous state space model. You can choose a sampling period; you can create the discrete dynamics now using this c2d conversion with respect to the sampling period. Now, this discrete dynamics, you can then assign to each of these A,B,C,D matrices in the discrete time domain.

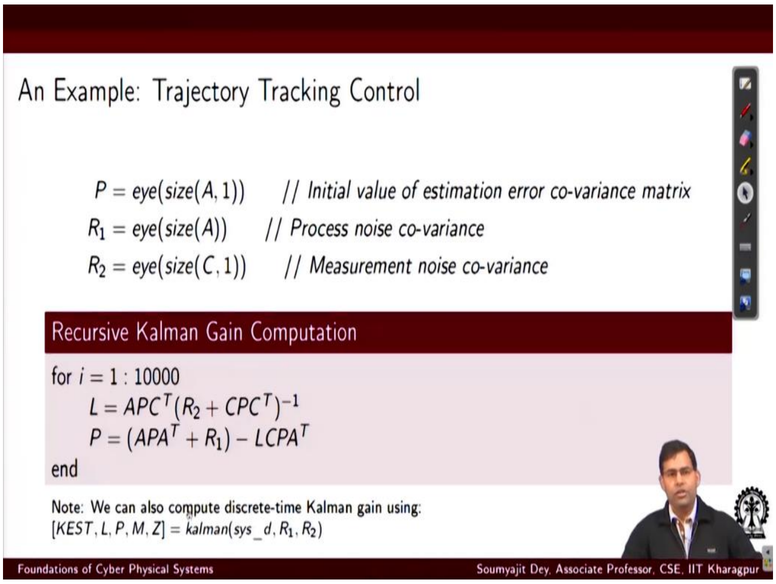
(Refer Slide Time: 16:40)



And once that is done, uh you can create the uh weight matrices of system state and weight matrices of control input, the Q and R matrices, right, like this. And uh so, basically, for A using, you can use A and B itself with this Eigen Eigen Eigen function of MATLAB, right, uh to choose the size of a the sub, a subset of, the sub size of A and create Q and R like this, right.

And then you can use the LQR command here uh to create the corresponding uh LQR gain uh for the, for this system. So, you are basically generating an LQR controller here, and as a side effect, you also get the solution for the Riccati equation, the Eigen values of these, etcetera, etcetera. Okay. So, overall you get this controller designed here.

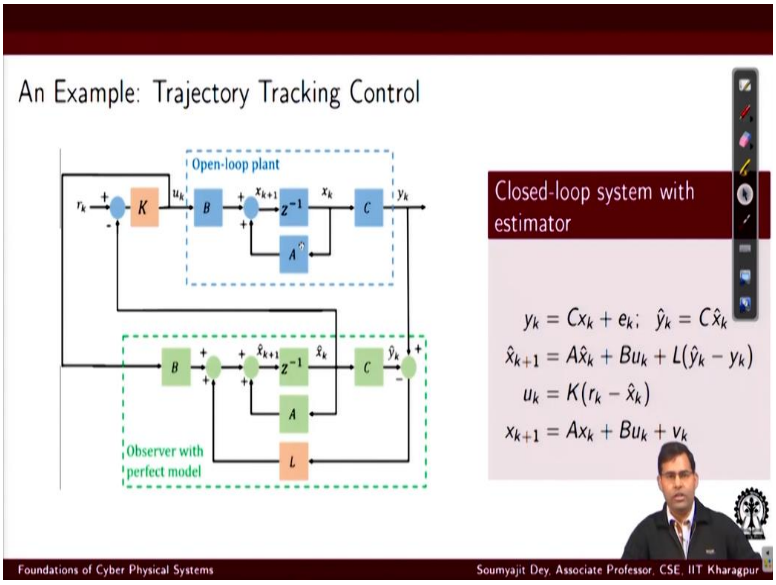
(Refer Slide Time: 17:31)



Now uh you can, you will need to create this uh Kalman filter. So, if you see, you can generate an initial estimate of the error covariance. Again you can have an initial value of P uh which you can create using this Eigen function of MATLAB again and you can create this process noise covariance, the R_1 and R_2, the noise variance and noise covariance values again. And then uh you can do a recursive Kalman gain computation, right. So, you can write this MATLAB code. So, essentially, you are just creating these matrix transformations like we discussed earlier. So, if you see here, we have taken that R_{12} term as 0, right, and you have just defined that R_1 and R_2 here, uh right.

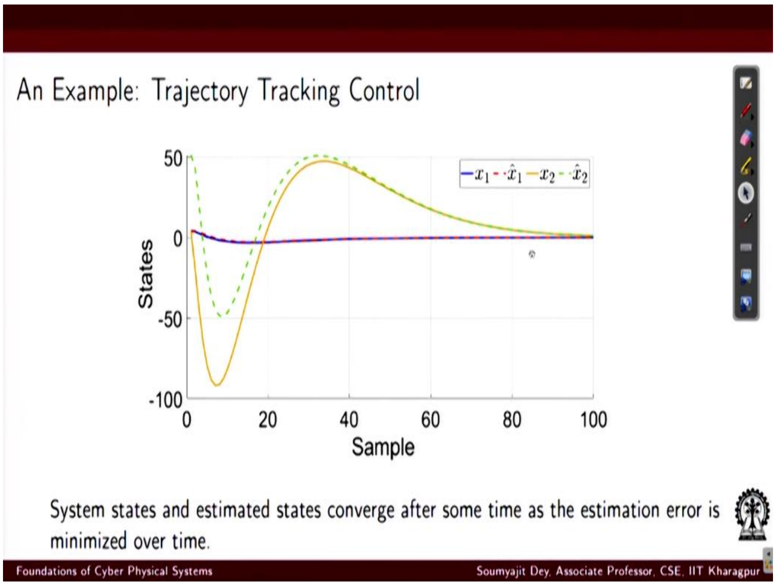
For that, if you just iterate over this, you will you will see that the gain value is getting generated and finally the gain value will converge or in other way, you can just directly use this Kalman command of MATLAB and you can get that Kalman estimator gain estimated directly, okay, and you will see that those values are almost same.

(Refer Slide Time: 18:38)



So, that is the way to check. So, this is again our picture where we can see that well, how the closed loop of the trajectory control can be designed along with this estimator in the loop, right. The estimator is basically part of an or part of an observer here and the observer uh is assumed to known this, all the system matrices of A, B and C and it is in incorporating this Kalman gain also here. Okay. So, this is the closed loop with the estimator.

(Refer Slide Time: 19:07)



And uh if you just try to plot this system state and the estimate states, right. So, as you can see, x_1 and x_2 are like this, so, they are almost similar, right. x_1 and x_1 hat from the estimates of x_1 are almost similar and x_2 initially has a deviation.

So, if you want to recall what was x_2 here. So, it is the deviation from the reference, right, and x_1 is the position here, right. So, what you can see here is that the deviation will eventually cancel out and it is able to track the system's trajectory properly. Okay. So, that is about it. With this we end our derivation of how a Kalman filter based estimator can be designed. So, as you can see is basically a minimization of a least square setting and the way we derive the filter gain is using a standard uh minimization technique uh of optimization. And well, with this we will end our lecture. Thanks for your attention.