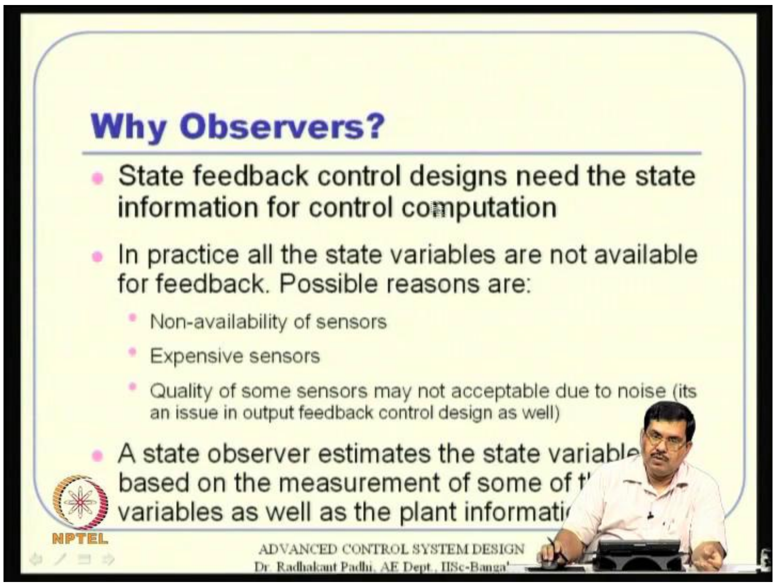
Advanced Control System Design
Prof. Radhakant Padhi
Department of Aerospace Engineering
Indian Institute of Science, Bangalore

Lecture No. # 40
An Overview of Kalman Filter Theory

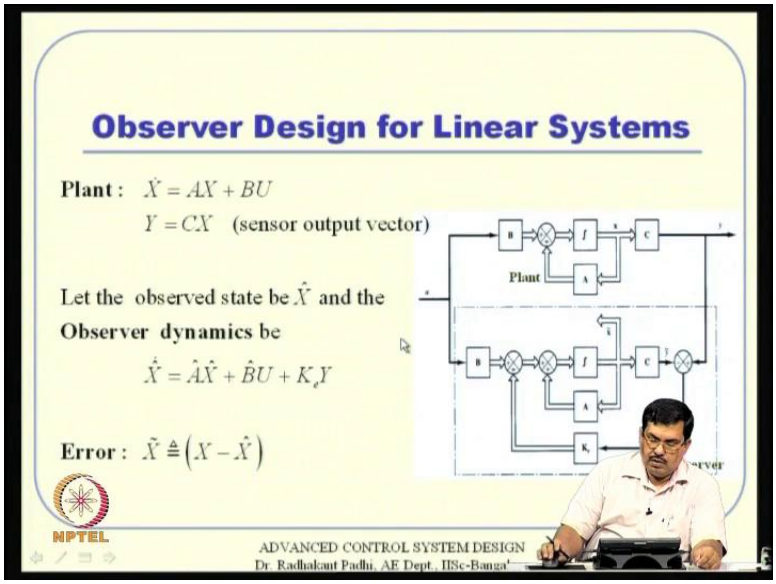
Hello everyone, we will continue with our last lecture today; and before end of this course, I thought an appropriate topic is some review of Kalman Filter Theory, because without that probably in my view, no control theory is complete. Now obviously, I will not be able to do a justice with all the derivations and everything about Kalman Filter Theory, but I will be able to give you some sort of a summary or a little bit theory round that, so that you can actually take this and implement in your problems actually. So, let us do that, but before before doing that, let us go through a little bit on what we discussed last time. And that was about this LQ observer, that we discussed last time actually, so a very quick review of that.

(Refer Slide Time: 01:06)



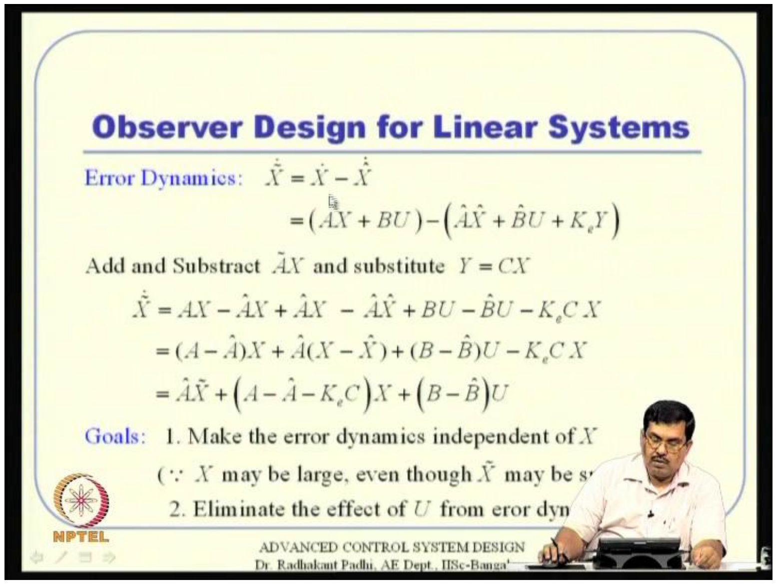
So, this is this is what we did this, observers and all we are need, because I mean the need for the observer or estimator is because we typically proposed a feedback control design, and state information is needed for control computation. And most of the time, you may not have sensor rich systems. In other words either the sensor is not available expensive sensors, all that actually whatever we discussed last last class.

(Refer Slide Time: 01:29)



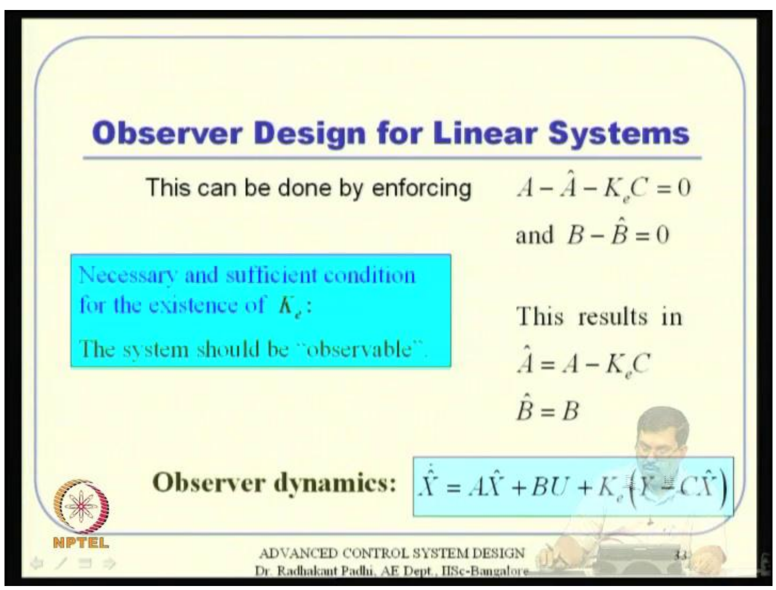
So, what we did did here, in an observer for linear system design, we have a system plant like that, we propose that observer dynamics we constructed like this, where K is an estimator gain. And then we did discussed about the error, error being X minus X hat, where hat is the estimated information, X is a true information.

(Refer Slide Time: 01:47)



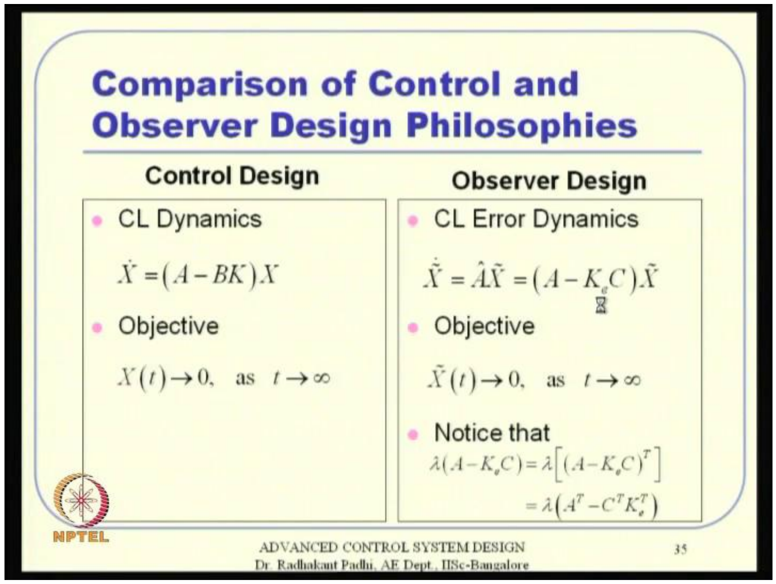
Then what happens is like, you go through this error dynamics and think like that, and try to make sure that, it is not a function of the error dynamics is I mean the X tilde dot is not a function of state and control. So, we enforce this coefficient to be 0.

(Refer Slide Time: 02:00)



And then we derive this estimated dynamic, observer dynamics to be of this form. So, it is exactly falls into this this form that X plus B U, but we have an additional term K e, that is estimated gain times, Y that the actual output minus estimated output C X hat actually, so that is the innovation term (Refer Slide Time: 02:07). So, you have that filter dynamics as like linear system dynamics plus Kalman gain times, innovation term actually.

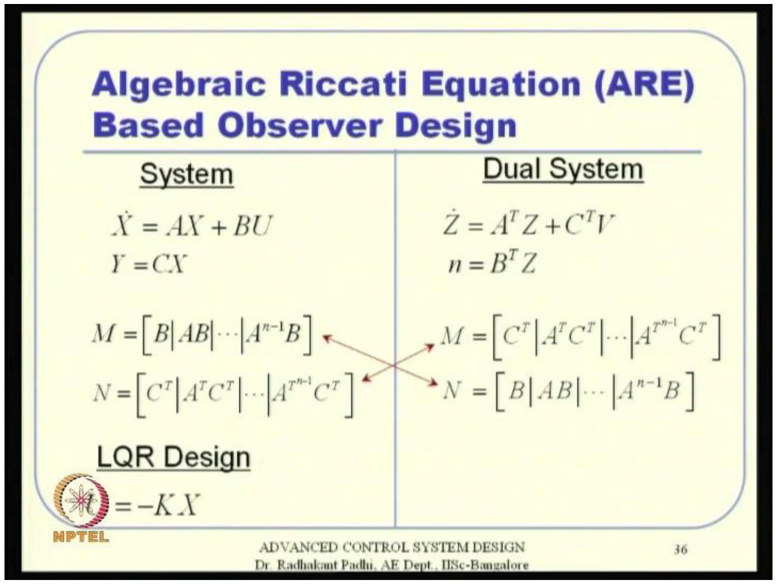
(Refer Slide Time: 02:28)



Then we went and saw a comparison between control design, and observer design where you tell the close dynamics of that, for the control system design and like that, and close look error dynamics for the observer, I mean turns out to be very close to that actually. When we discussed the objective here is like, X (()) go to 0 and that is the keeping that objective, we have designed an (()) controller, and objective here is also X tilde should go to 0. So, fairly similar objectives, only problem was K e happens to be in the left hand side here whereas, it happens to be right hand side here.

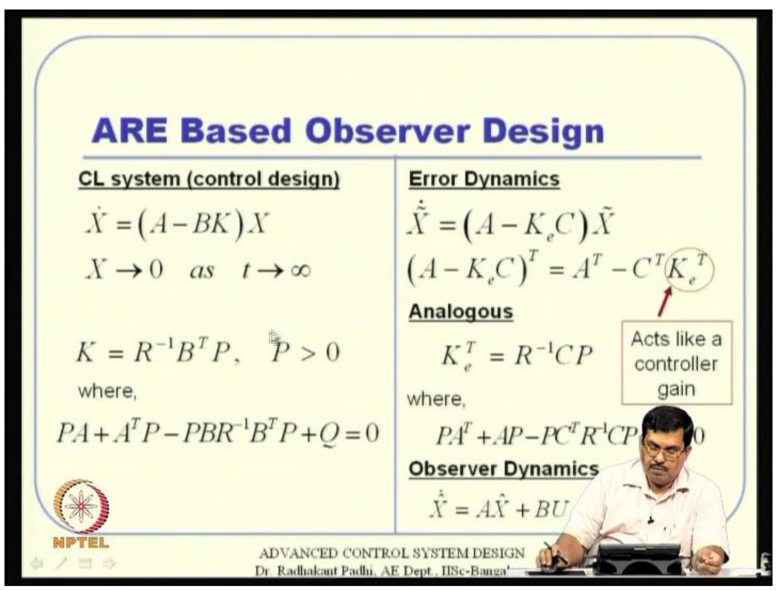
So, we took transpose of this matrix and while doing the transpose K e transpose turns out to be the right, and then we told we can we can actually treat this A transpose and C transpose is equivalent A and (()). And then design a K e transpose instead instead of directly designing K e, you will be able to design K e transpose, but the once you design that then taking transpose means, we have done with the design actually.

(Refer Slide Time: 03:21)



So, for that we have this system dynamic original system, and had a dual system, and observe that control availability for one is observablity for the other and vice versa.

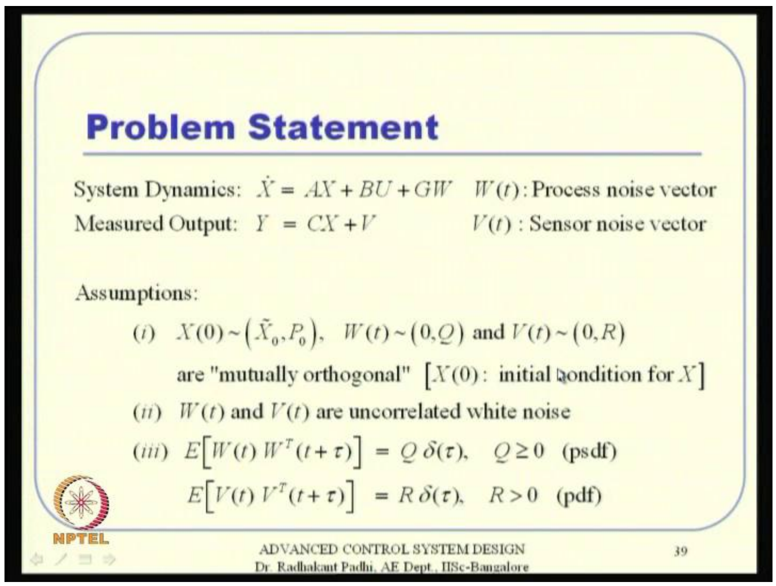
(Refer Slide Time: 03:32)



So, if you see this close looks to system dynamics and all, so we propose that this K e transpose is should be designed, that is similar to what we design is Kalman Kalman gain actually, I mean for this, for the control design part of it (()). So, instead of R B transpose T we should have R inverse C times P; and this P should be a is like a solution of this matrix, and think like that actually.

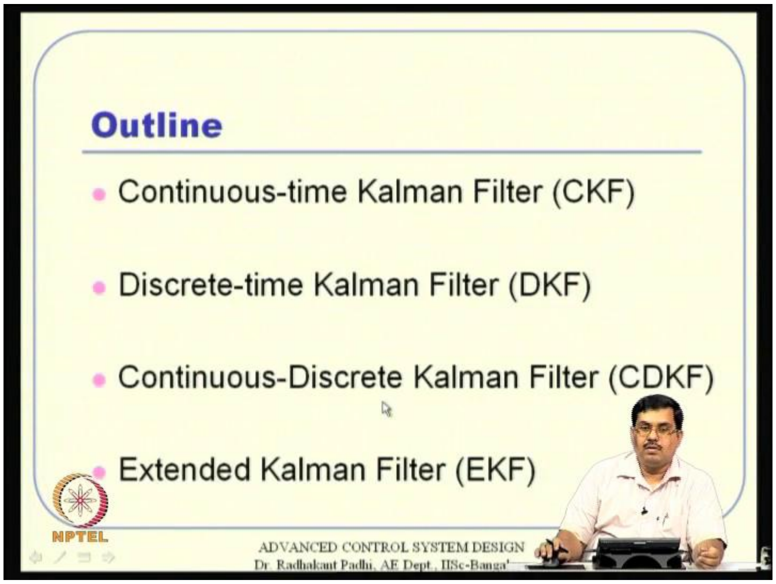
So, any way, we continued with that and then the observer dynamic turns out to be like this, with the with the way to design and observe again actually. So, this is this is all turns out to be like a like the L Q observer design, and then we will say, I mean we have also told that towards the end of the last lecture is like.

(Refer Slide Time: 04:18)



If you consider the system dynamics with a noise, process noise and sensor noise, then it then it also turns out that the filter dynamics and all remain similar, and then that is the little bit more retail will will see this, in this class and continue further actually. So, this this is about this lecture is about Kalman filter theory, so we will revisit that that towards the end of the lecture, what we discussed last time, and then continue further actually.

(Refer Slide Time: 04:47)



So, outline of this particular lecture, will be something like this, we have continued time Kalman filter, that we talked upon in last class, then we will extend that to discrete time Kalman filter. So, the system dynamics is given in discrete form now, and the measurement is also given in discrete time. Then we will try to combine the (()) discrete Kalman filter; in other words, system dynamics is continuous, because measurements comes at discrete times.

So, that that is what reality is actually most of the time, and this will give us a platform to talk about something called extended Kalman filter, which is used heavily in in practice. Most of the time when people tell I have used the kind of Kalman filter, that is what the mean they have actually implemented and EKF in the into this design basically. So, to up to that, we will discuss in this lecture and then probably that will be about summary, before we wind up this course alright.

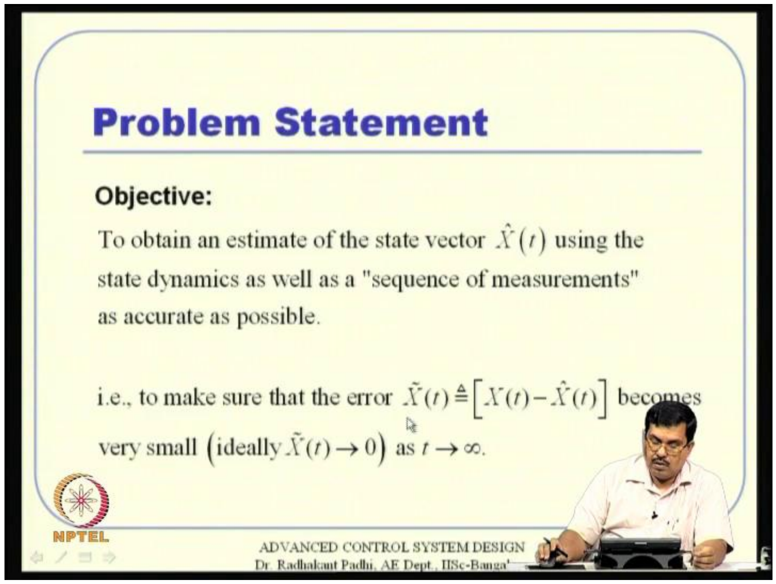
So, continuous time Kalman filter design for liner time invariant system, that is the that is the basic starting point and that is what we discussed last class also. So, we we have a system dynamics which is like X dot is A X plus B U plus G W, and then measurement equation turns out to be Y equal to C X plus V, where W and V are like, W is process noise vector and V is sensor noise vector. Remember, this these are effecting the system anyway, because the W is nothing but, an input which is directly effecting the system state, and if you really want to estimate a state into the feedback control design; that means, U becomes estimated for I mean function of an estimated state; then for the estimation process you will use Y, and Y is corrupted by sensor noise anyway. So, this two will will affect the system dynamic, the effect I mean it will affect the performance of the controller like that actually. So, if it proceed further there are some some of the assumptions involved, and what it means is initial condition of X is given like this, W is given like this, pair Q 0 Q and V is like 0 R, what it means in this parenthesis is, the first thing is mean the second is variance actually or there is a vector it is called covariance matrix (Refer Slide Time: 06:48).

So, that means, X of 0 initial condition mean mean value is X tilde 0 and the covariance matrix is P naught similarly, W T is is a process noise has 0 mean, and Q as the covariance matrix for this, what is the covariance matrix by the way, this is all given here. Expected value of W times W transpose is actually Q, if tau is equal to T and sorry tau is 0; if tau is not 0 then it will happen to be 0 actually, because delta function is defined like that any way.

So, this is Q means that expected value of W time, W transpose at the same time basically, that is what it means, anyway so we also assume that this W and V are uncorrelated white noise, that is a fundamental back bone Kalman filter theory, that this noise thing that that are accounted for are assume to be white. They may or may not be white, but in the entire theory W of they are assume to be white, and what you mean by white is like there supposed to be like uncorrelated that means, if I take I mean the correlation process and all are are defined like this.

If I take any (()) other time like T and T plus tau, tau is a non 0 quantity, then I should be at 0 actually, so there is nothing, if I multiply the same process noise with the same point of time, then I will get Q, but if any other time if I multiply then I will get 0 actually. So, these are like, what is what are called is white noise actually, and we also they are assume to be 0 mean actually, W and V are assume to be 0 mean alright.

(Refer Slide Time: 08:44)

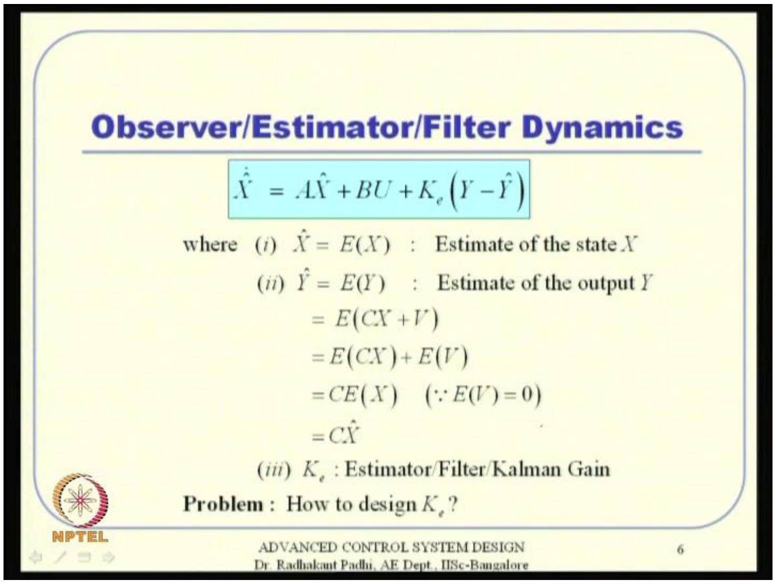


So, objective here what is the problem statement, they are the objective here to estimate the state vector X hat T using the state dynamics, as well as a sequence of measurement actually. We are not talking about only a single measurement, you have to have a sequence of measurement and using all those sequence of measurements, if observablity is there that means, if I mean if I keep on taking sequence of measurement as a combination actually. The sequence of measurements, so will have will be effected by system state actually, let us the meaning of observablity.

Assuming that observablity is there, for the system output and system dynamic (()) that we are talking about, then what we what we really talk is, we take sequence of measurements, they are corrupted by noise alright, still we will be able to get a good estimate X hat of T. And what do you mean by good estimate, it again means that, if I take X tilde that is error between true and estimated values, that will become very small I mean I mean ideally X tilde will go to 0 X T tends to infinity. But, in a in a stochastic sense, where this noise vector is there and all this will not happen in this you know, even though we would like to happen actually.

So, what will what will happen, then the expected value of X tilde will go to 0, if I as I take expected value means, it is large large I mean as the average of large number of cases actually. So, if I keep on I mean taking this X tilde T for a large number of discrete point of time probably, then if I take the mean value of that, and that should go to (()) at least. So, expected value sense it should go to 0; that is all we are demanding actually. So, let me probably write it here, what we are telling here is expected value of X tilde t, so it will go to 0, that is that is our objective actually, totally speaking alright.

(Refer Slide Time: 10:47)

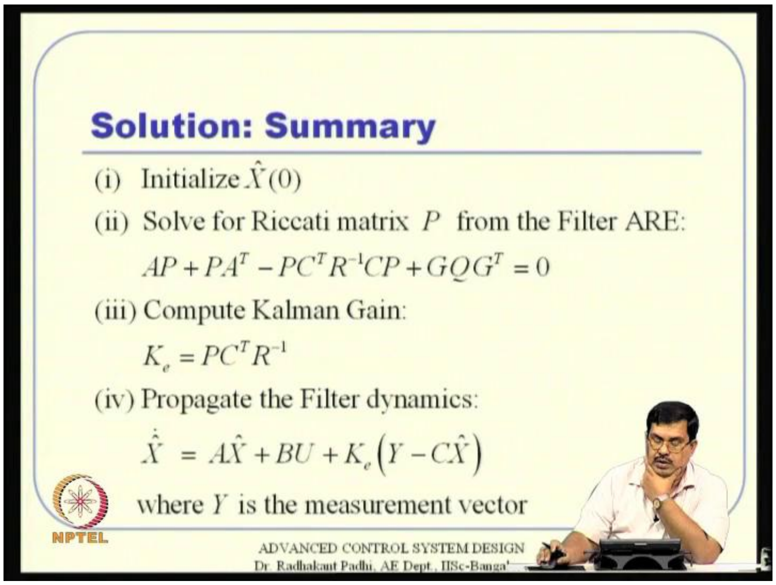


So, let us let us see that, how does it happen, we will we will get motivated by this L Q observer, that we discussed last class and all. And take K, we will assume the same observer dynamics rather; it is it is not similarly, it is same actually. So, we will just take a X hat plus B U plus gain times estimated gain times Y minus Y hat, where Y hat is estimated output, and estimated output will turn out to be C X hat again actually.

So, this derivation is fairly straight forward, because see Y I mean, X hat by definition is expected value of X, what you mean by X hat, here is not I mean this X hat is nothing but, expected value of (No audio from 11:36 to 11:46) (Refer Slide Time: 11:25) this X this X hat is nothing but, expected value of X actually. So, in that sense the Y, if you continue with Y hat, and Y hat is expected value of Y and Y is nothing but, C X plus V and I get property of that expectation operation is it is a linear operator. So, we will be able to split it out and take C out of this operation.

So, it turns out to be expected value of V is anyway 0, because it is 0 mean white noise. So, this is, because of that the expected value of V turns out to be 0, and expected value of X is except by definition. So, Y hat turns out to be C X hat, and that is how it will operate actually, now the problem is how do you how do you come up with this design of K e. So, this is what what is required anyway, without that we will not be able to propagate the system dynamic actually. So, we will not go through the entire derivation, I mean it it will require probably a full class and all that.

(Refer Slide Time: 13:00)



So, what we what we what I will tell is just a summary part of it, and summary part is like the gain, what we are talking about here, Kalman gain can be computed like this, P C transpose R inverse, and then P C transpose T inverse, this (()) that after you compute this, then X hat dot is nothing but, A X hat plus B U plus K e times Y minus C X hat actually. So, what do you compute this P, P is given like you can solve this Riccati equation, it is called filter Riccati equation actually.

And if you see this Riccati equation, I mean before what we discussed, so the only difference is probably this term actually. So, this term I mean the L Q observation same this this term, earlier you did not have a G in the left, G transpose in the right actually, it was only Q (Refer Slide Time: 13:42). So, if you assume that G is actually, like a identity matrix here or G W is the noise, not not W parse, then it is nothing but, an L Q observer actually.

In other words, L Q observer is also like a Kalman filter, where you are assuming that that G is actually identity matrix, may not happen of course, I mean the W, may not W is like a control input, control input may not alter the X dot directly with V; it passes through a influence matrix anyway G, it is like a V matrix. So, if you assume that G is actually identity then it is nothing but, (()) L Q observer actually, anyway so this is this is how it operate, so if you really want to mechanize a Kalman filter like this, all that you have to do is initialize X hat 0 with the guess. Then you have to solve this Riccati matrix P for the filter algebraic Riccati equation, because this is in the in the framework of infinite time that means, it may take a little longer time to stabilize actually, if your T F is infinity, then this is 0 otherwise, it will not be actually 0.

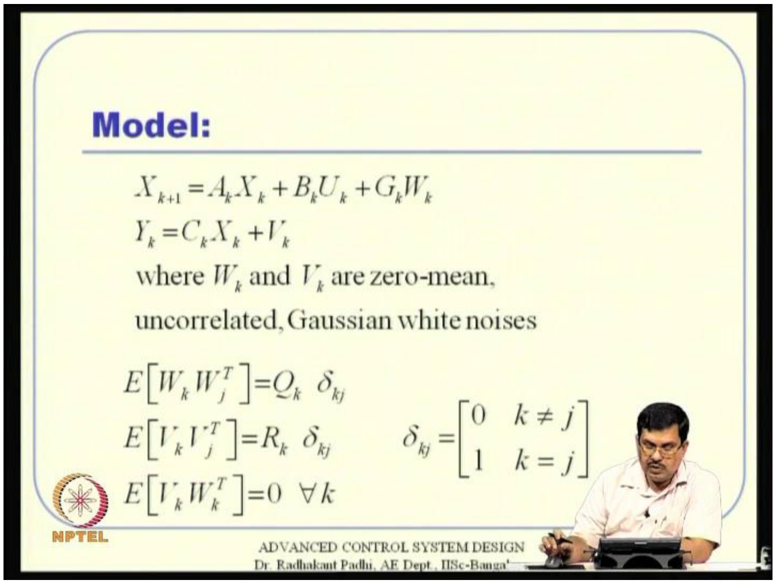
Anyway assuming that it is, if this algebraic equation, Riccati equation is solved this is called filter Riccati equation, and then this after we solve for P, we compute the Kalman gain that something like this, and once you are there then you can propagate the system dynamic. Because, you have already have an initial condition there, and that it will also lead to this good stability behaviour, and think like that means, the error that we are talking about X tilde, which is expected value of X tilde and all that what we discussed here, it is all guaranteed to happen actually, because, it is a linear time invariance system.

And there are also nice property such as, in the sense of this is of something called separation principle and all, so that means, you can design a controller and of the estimator separately each other. So, it it will not have the stability behaviour, each of the thing even though, you want to operate it based on the feedback system, I mean if you want to operate U based on a feedback of estimated state, then all nothing is going to happen drastically wrong, basically.

Because, the close loops Eigen values, will be Eigen values of this system I mean and the Eigen values of error dynamics and all that. So, those theorems are, there I will not discuss too much on that actually. So, this is the summary of how do you mechanize a continuous time Kalman filter. But, that is not the problem here is it, this this entire formulation, assumes a continuous availability of measurement Y is C X plus V; that means, continuously the measurement is is coming to us actually.

And suddenly, that is not reality; measurements are taken only a discrete point of time actually. So, if if that is the case then it also makes sense to probably discretise, this system equation, system dynamic equation; and we have several methods, any way we can use 0 order either integration also the thing. I mean what about (()) method. So, once you (()) the system dynamic, and the measurement is anyway coming in in discrete manner; then probably we will have a compatible system to talk about actually. So, that is what this discrete time Kalman filter theory talks about.

(Refer Slide Time: 17:12)



So, let us let us talk about that, so what we are assuming here is as a discrete time system model, again linear time linear system, and when you talk about discrete time system normally, there is no difference rather you take like time varying system or time invariant system, so we will just take time varying system actually. So, X k plus 1 is A k X k plus B k U k plus G k times W k Y k C k X k plus B k, so remember A k, B k, C k, G k all that they are then is not necessarily be constant, they can be also the time getting actually.

Again we assume that, W k and V k are 0 mean uncorrelated Gaussian white noises, this is a these are like little bit strong assumption, but then these are standard assumption for Kalman filter theory. So, this they they are corrupted by noise alright, but the noise that is this property, both of the noise have 0 mean, they are uncorrelated and the Gaussian process is also (()), Gaussian makes it probability the assumption of Gaussian (()) I mean done to both address realistic situation. That most of the time, then as happens to be Gaussian, and once you assume Gaussian noise there there is lot of nice properties actually, for which thus the theory becomes complete.

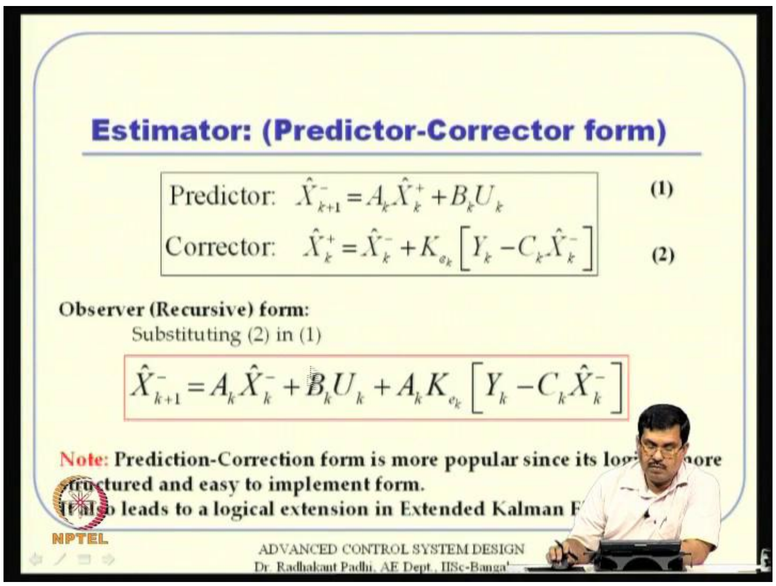
For example, if you if you have a Gaussian distribution then mean and variance gives us the entire meaning, I mean more than that there is nothing actually. So, if once you know the noise is actually Gaussian, and once you know the meaning and it is variance, then we are probability done actually, theory we complete easily. But, for entire derivation remember, I mean in my view all that is required is that it has 0 mean non correlated thing, so 0 mean white noise that is all we need actually for the derivation part. So, let us proceed with that, then because of this assumptions, these are these are all good and what it means is if I take expected value of W times W transpose, if they are not taken at the same instant of time that is 0, nothing happens there.

If they are taken at the same instant of time, then this and this Q k matrix and this Q k is called process noise covariance matrix, actually. Similarly, if you take sensor noise covariance matrix, that is the relationship between (()), so if they are not taken at the same time that is 0, if they are taken at the same time then this and these are connected, is R k matrix. And if you take even at the same time, what about different time between V and W, then it is 0, so they are totally uncorrelated to each other actually.

It is either auto correlated autocorrelation sense self correlation sense that that happens, that way, but if you take cross correlation sense V k, W k like two different things, I can together on that no matter, what about time we are talking about they are all 0 actually. So, these are like, if you think that is slightly strong essences rather, there are also tricks and techniques to is not really a white noise, how do you how do you make use of it there are ideas like say filter design like, you can situate some sort of a small system, sub system rather, where where you take white noise input.

And output of that artificial sub system should be really the colour noise that we are talking about actually; then you can argument the original system dynamic with that artificial system, which will like in filter design and all. Then then you can talk about estimating the entire state vector actually, so those (()) summary is interested they are encouraged to study details of those anyway, filter theory is a fascinating subject it is a complete theory by itself and all that actually. So, there are many tricks and techniques available on the various actually.

(Refer Slide Time: 20:53)



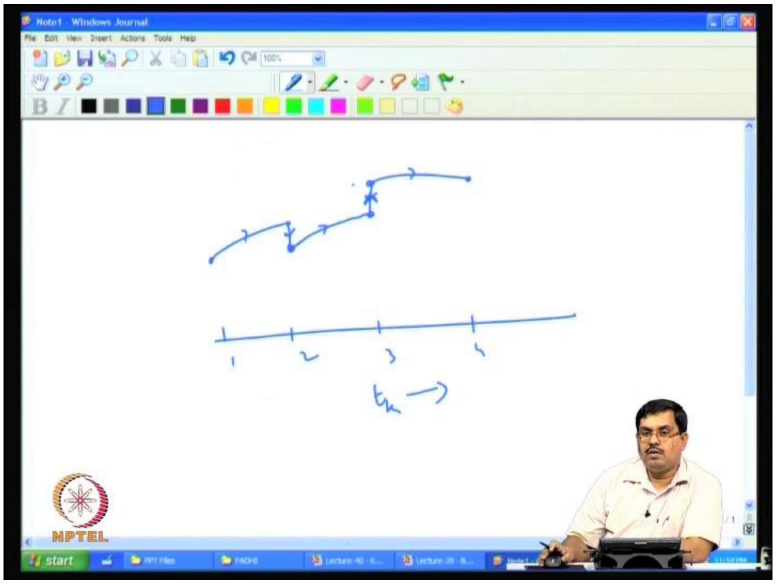
Anyway, so going back this is what you what is assumed, now what is our I mean what is the way to proceed further, so we need some sort of a filter dynamics anyway. And the filter dynamics is an artificial system dynamics, which simply needs to be propagated, with some initial condition and some then computation actually. So, there are two ways of doing this here, and the discrete time, one is like I will assume it is a it is a predictor corrector form, which is actually very popular.

There are several reasons for that of course, and then there is a observer form what you have this what you have seen in the, this continuous time framework, so observer form or recursive form can be derived from this. So, for I mean this from the predictor corrector form easily actually rather, anyway what you are doing here first of all you predict for the next time state; let us talk about this, let us assume that I know some state information already, at k the instant of time and I also know the controller at k the instant of time.

So, I will be able to I will be able to take advantage of this system dynamics, without the noise of course, and then I will be able to predict what is going on actually that means, if the noise does not happen to be there, then I should have a better prediction here. Whatever I am predicting here, but remember this prediction part assume any any sensor information actually, and once the sensor information is there that means, sensor has given me some value; I will be able to update this value from from that k plus 1 time (()). And same thing happens in T the instant of time also, if I start with k minus 1, then I this I will get k actually here, and then k will be updated here. So, first prediction, then update, prediction update like, that it will happen actually. Now, if you the reason it is written here (()) k plus 1, but k is easy to see, that if I put substitute this this expression here, then I will be able to derive this actually; this is this is observer form or or recursive form actually. So, prediction correction form is more popular since, it is more logical structured way, and easy to implement also actually, we will see that actually.

They also leads to a logical logical extension, extended Kalman theory, which is which is primary requirement (()); if (()) once have definitions remember, we are talking about plus minus plus minus all sort of things here. So, we will be able to do some of these actually anyway, so before we proceed further, this in this let me explain a little bit here it is ok.

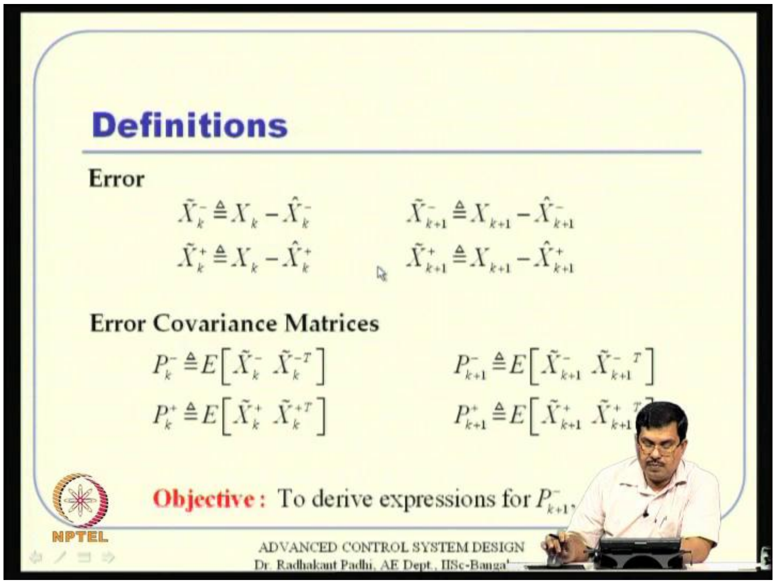
(Refer Slide Time: 23:46)



So, what we are telling here is something like, we have this time sequence of k this is 1, this 2, this is 3 like that, well we have this picture one more time alter anyway. So, I start with some information here, and then I will be able to predict using the system dynamic that I know, from step one to step two this is that time exist actually. Then when I when I have this measurement information coming then, I will update this here, then I will again predict it here from two to three, I will get some value, then I will be able to update using using some sort of sensor information, then I will continue that.

So, this is that the prediction part and then there is a correction part actually involved, so this is this is the correction part sorry this is, this is this way. So, predict correct, predict correct like that it will happen actually, so this that is the meaning of that aright, so let us continue this.

(Refer Slide Time: 24:44)



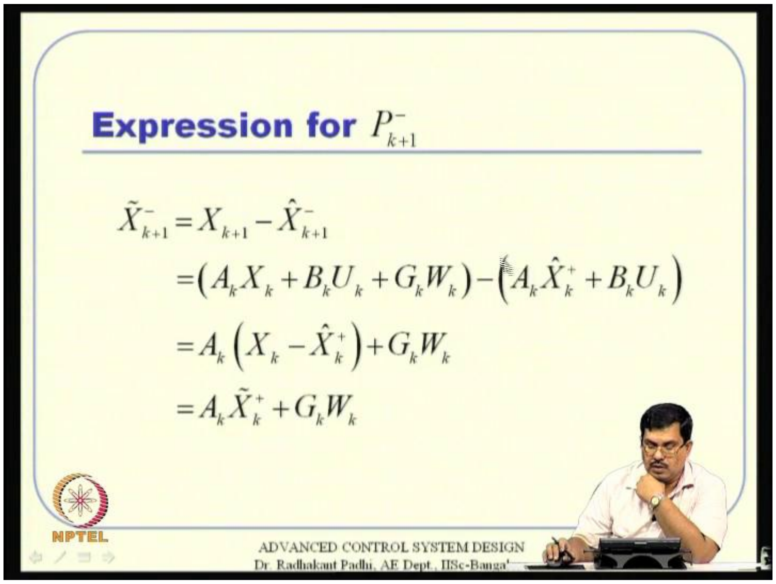
So, before before proceeding further let us talk about certain certain error values, and all that, so remember I have so, the prediction part is is written normally like minus, and the correction part is written as plus actually. So, when you see minus that means, it is actually coming from the prediction, when you see plus, it is actually corrected taking the output information actually, so prediction part and then correction part like that.

So, I mean we will define this this error quantities, this error covariance matrices and think like that before proceeding further. So, this X tilde minus X tilde k minus is nothing, but X k minus X k hat minus similarly, X tilde k plus is X k minus X hat k plus and similarly, k plus 1 and think like that. Now, error covariance matrix, it we also needs to define, I mean we need to define at this way P k minus that is the error covariance matrix at time step k, in the prediction part of it, is given like this, expected value of X tilde k minus times X tilde k minus transpose like that.

They are all outer product anyway, because it so each of that is (()) vector, so if I get an outer product, it becomes like (()) matrix basically. So, this error this error values are error states are defined like that, and error covariance matrices are defined like that, they are simply definition to (()). And obviously, objective here is to derive expressions for this actually, we ultimately need to derive expressions for Kalman gain, k E k, but also we need to derive this expression, P k P k plus 1 minus and P k plus because, k E k is a function of this. So, we need to derive expressions for that actually, and we cannot really start with the definitions, because this is expected value means we have to we should have I mean infinite number of this I mean values available to us.

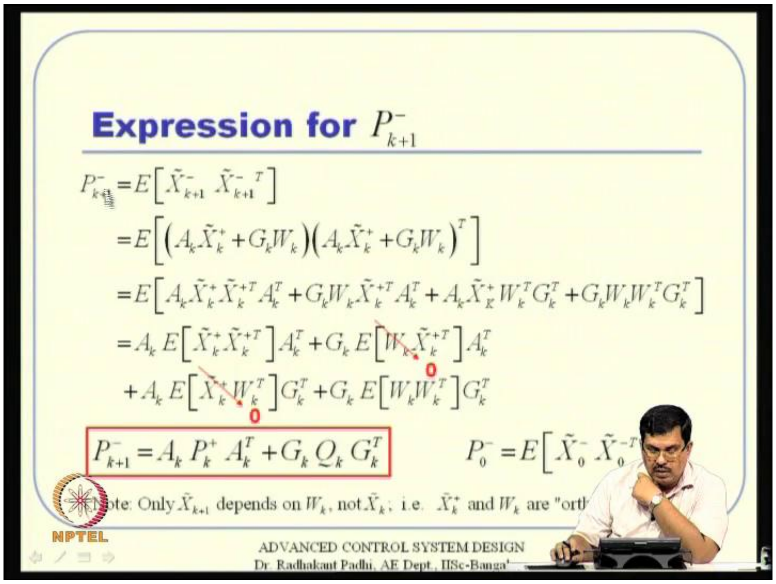
And then we have to take average value of that and all, so from definition it will not be it is not advisable, it is not possible also. We need a kind of precise information like; separate expressions to derive this actually, so let us do let us see whether we can do that. Now, expressions for P k plus 1 minus, so remember P k plus 1 minus is is given like this, so we need to derive an expression for that one first, and then type the expected value of this outer product and all that.

(Refer Slide Time: 27:12)



So, what is this, this is by definition is this two and then this this X k plus 1 is like this, then this this part is given like that from the definition, this is coming from this actually. So, this this part can from that, and then this part is nothing but, the system dynamic remember, estimated 1 is the known wise, but the true one is given with noise anyway. So, now we see that these two are are cancelled out, and this this B k U k, will go from B k U k and you are left out with only the rest of the terms; and that is given something like this, so if I see X tilde k plus 1 minus it turns out to be like that.

(Refer Slide Time: 28:01)



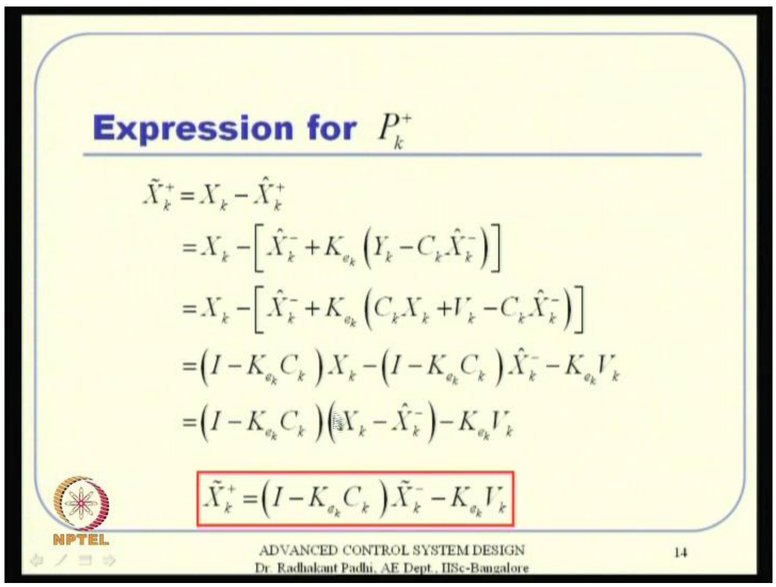
So, now, I can take P k plus 1 minus which is expected value of this this times the transpose, and this one I have just derived, this is like this, so I will take a transpose of that also, then I will I will expand this transpose and then multiply to the do the algebra and all that actually. And remember, these are all like cross correlation is 0, so expected value of this W k and X tilde k this is 0, and this is also 0 anywhere, we see a across correlation term, these are all 0 by assumption in Kalman filter theory.

So, it take (()) out those and we are left out with these, and these values remember whatever you have here this is nothing but, definition if you go back to definition this turns out to be like that, so this is nothing but, P k plus actually right. So, this this one what you see here is nothing but, P k plus, so what you what you have here A k times P k plus times A k transpose (()).

(()) simply like good book keeping basically, if you know what you are doing substitute, expand and then cancel out put something, I mean C wise all times 0 and deal with rest of the term that are left out actually. Similarly, if you see this one, expected value of W k times W k transpose, that happens to be Q k by definition, and then it turns out to be like that actually.

So, if you really want P k plus 1 minus, then it is a function of P k plus and think like that actually. So, this happens to be because, this W k and X k these are like what is called as an orthogonal, orthogonal means this expected value of that happens to be 0. So, this process will start start from P P 0 minus, and P 0 minus is like initial covariance matrix of this error vector and all that, and that is supposed to be a selected by the design basically. That is how you start with along with your X tilde information, like guess for the initial condition for the estimated state; you also start with a guess for your covariance matrix actually.

(Refer Slide Time: 30:06)



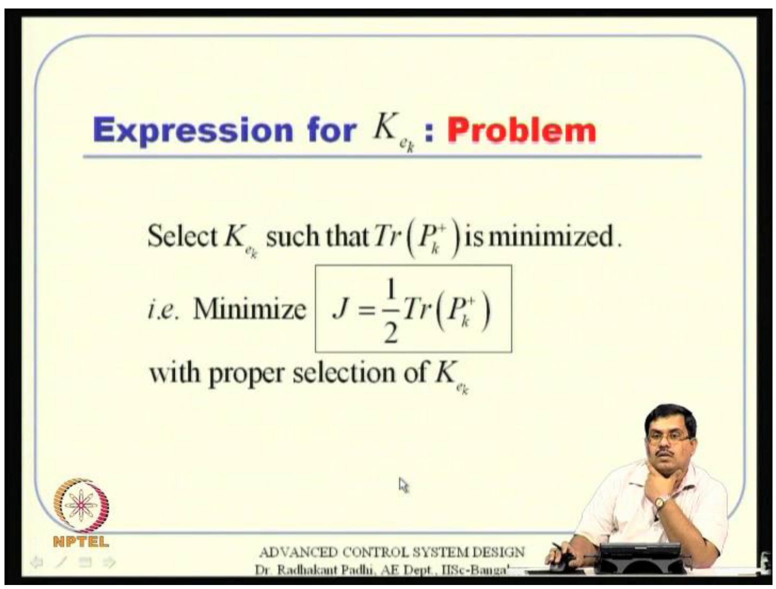
So, this expression is derived like that, what about the other one that P k plus remember P k plus is required here, so we need that, so P k plus is by definition is expected value of X k I mean, X tilde k plus times X tilde k plus transpose. So, we need to have X tilde k plus first, so this one is given like this by definition, and actually that k plus is given by this, this is the predictor corrector form, so this is this expression comes there.

So, if you put this this expression this, this part here and then try to simplify this it turns out, that I can probably simplify that way actually, so I will combine this X k term is here this X k term is here, so this is I minus K E times C k times X k (Refer Slide Time: 30:36). And similarly, I have this X k hat minus here and X k hat minus here, so I will put it there actually that way. So, I will try to simplify this, and I got some sort of a formula like that, so

P k plus is expected value of this expression times that, so I will put this one times that that transpose.

And then again do the same thing, it will expand this this bracketed term, and then multiply and see, what all in equation then I can do. So, it turns out that again this expected value of this, these are constant mean these are like expected value is not a function that, so I will be able to take out and these are orthogonal to each other, so they will go. So, what are left out, these two expressions see this (()) told here is this two expressions will not be required, so first and last will be there actually (Refer Slide Time: 31:31). And first and last by definition will will turn out to this part, by definition is P k minus and this part by definition is actually, R k anyway, so P k plus happens to be like this.

(Refer Slide Time: 31:50)



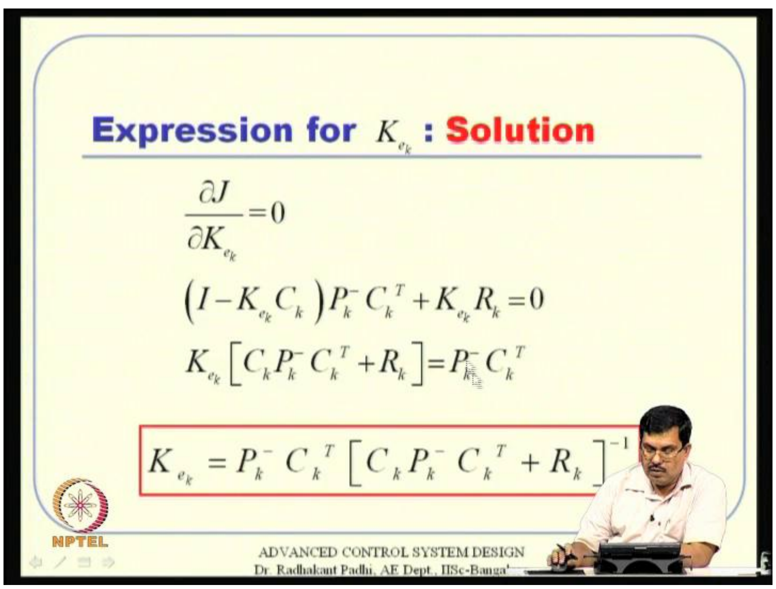
So, now we need to derive A k, E k like that estimated problem remember, what is our power problem actually, I mean the problem here is the P k plus which by definition, will go back to P k plus is error times, the error basically, the transpose of course. And this is after update that means, this this if you see X k tilde plus, this is after this X hat k plus that means, after I correct this using this innovation terms and all that whatever, I get is that and then error term happens to be true minus that.

So, obviously, if I take a kind of a covariance matrix for that, I want that to be minimum actually, as minimum as possible ideally 0. So, what I formulated problem is let me go back and try to see, whether I can minimize this this operator, I mean this P k plus and minimizing this P k plus means, I one of that is like trace operator remember this is a P k plus is a matrix.

So, trace is nothing but, kind of a norm operator and the trace is like, if you I mean if you see that a little bit that is individual terms, so will happen to be in diagonal, and trace is nothing but, summation of diagonals actually. So, if I just take this expression P k plus happens to be like this this one, and (()) if I take second norm square of this this X tilde plus X tilde k plus second norm square and that is nothing but, trace of P k plus actually.

So, the I want to minimize this expression half of trace of P k plus, so whatever happens, whatever the solution I will get is K e k basically. So, how do I how do I do that, if I really want to minimize this J, the necessary condition is J remember, this J happens to be the function of K e k; where P k plus there is a K e k expression and all (Refer Slide Time: 33:35). So, I consider this as a function of K e k, and then try to minimize that actually.

(Refer Slide Time: 33:49)

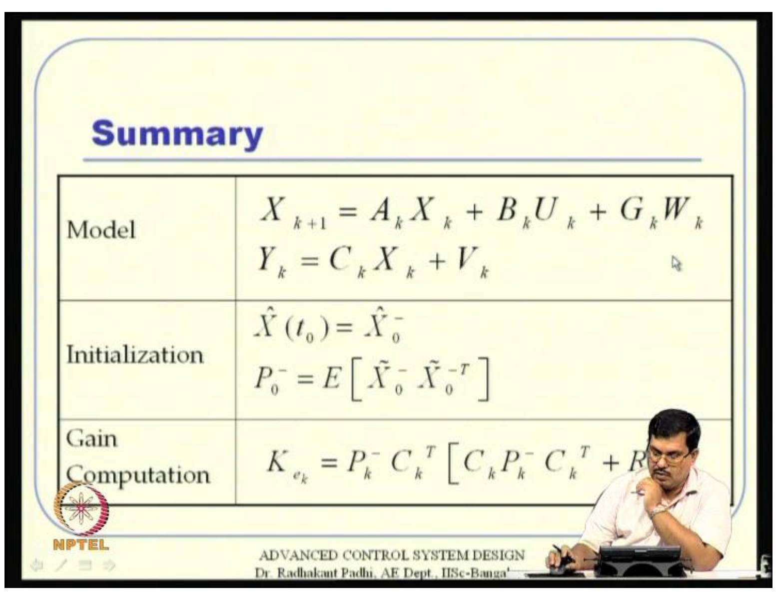


So, how do I mean do that, the necessary condition turns out to be like that, and J is this, where P k plus is this expression anyway. So, I substitute that, and then try to I mean take this also do this algebra, I turns out to be like this, and K e k happens to be left hand side of both , so I will be able to take common out of that. And then I will solve for K e k is like this. If I write multiply with this inverse, this this side we will get K e k this side we will happen like that.

So, K e k which is Kalman gain in discrete time formulation turn out to be like this actually; we have also an idea that K e k as what we derived here, one it seems slightly complicated, one does not have to live with that, you can actually simplify this expression, you can expand this with thing, and then cancel out a few terms and it turn out to be as simple as that. But later we will see that this form is advisable because of numerical condition is conditioning is used and all that.

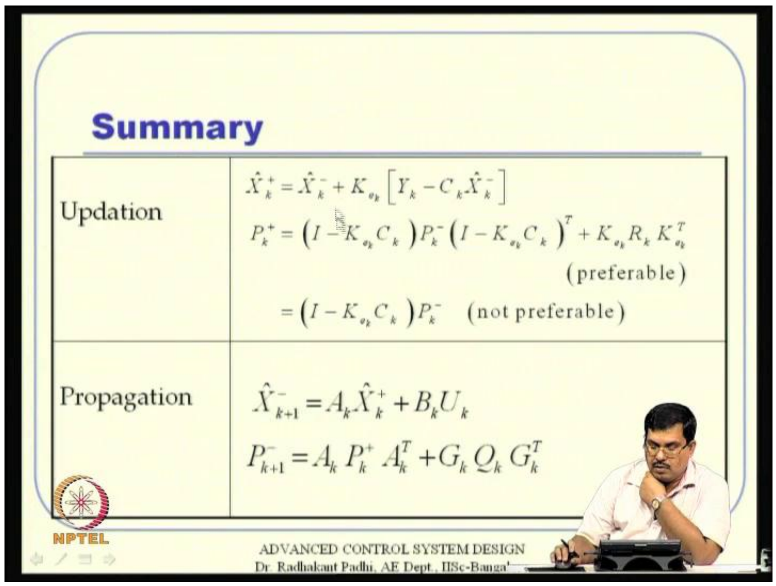
Now, this is like a quadratic form what you see here, so the symmetricity of P k will not be compromise even for even, because of numerical inaccuracy, same computation and all that. So, even though it is a slightly little more kind of computation, we still I mean we still advisable to use the (()) expression anyway.

(Refer Slide Time: 35:04)



So, as the summary summary is problem is like this, I have a have a discrete time formulation, where X k plus 1 is like this Y k is like that, where W and V k W k and V k are uncorrelated, white noise 0 mean orthogonal to state vector all that is there. And then we start this process by initializing this two, we initialize the estimated value of the state as well as, a guess value for the error covariance matrix.

(Refer Slide Time: 35:37)



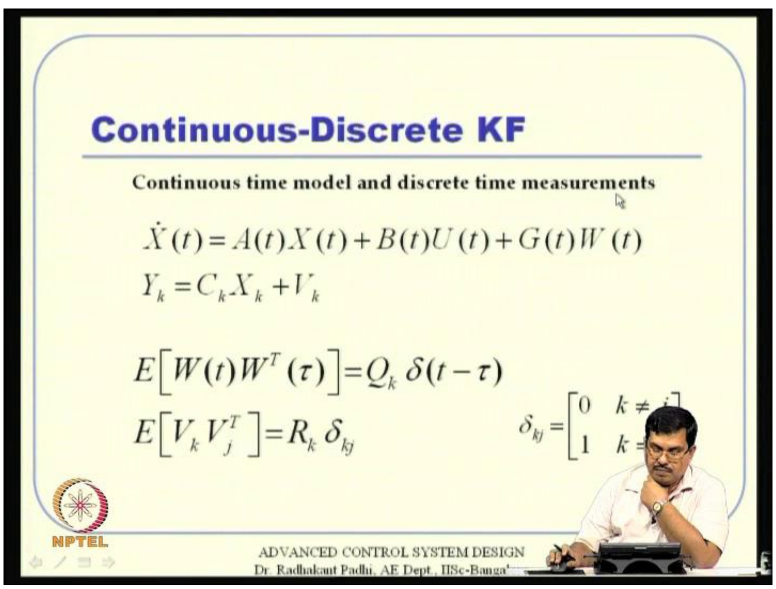
Then the gain computation can be done this way, where V k minus can be computed I mean using this this propagation equation and all that actually. This is this is what you see here is nothing but, propagation equation, basically what you see here that means, if I know P k minus I will I will be able to compute P k plus provided I know K e k. So, this is the I mean, if I have an initial condition I will be able to propagate this covariance matrix anyway, so this is what is going on here.

So, we are initialization which is done, then gain can be computed like this and update equations will proceed that way, so this is after get a measurement, and I will update the state vector that way. So, it is my propagated thing, that the prediction part of it plus the correction term which is coming from the measurement multiplied with a Kalman gain term actually.

And then there are covariance matrix can be propagated this way or that way, but this is preferable anyway, so we propagate that and get ready for the next time state actually. So, after we are done with this, then we predict it again one more time state, so both state as well as covariance matrix. So, this is because of this easy of implementation is logical flow of the equations and all that, this discrete form is kind of very popular actually, predictor corrector form is is a natural way of implementation actually.

Now, we will move further, and then tell we have obviously, we have seen continuous time Kalman filter we have seen discrete time Kalman filter; neither of that is actually a very good reality, because what happens is in pure continuous time Kalman filter the assumption is the sensor noise, I mean sensor measurement is is available in a continuous time manner, which is not realistic. And may appear discrete form the system dynamic needs to be written in a discrete form, which is approximation to reality. So, I do not know I mean this given a choice will like to work with a continuous time state equation, but discrete time measurement equation; and that is where it will lead to continuous discrete gamma filter.

(Refer Slide Time: 37:47)

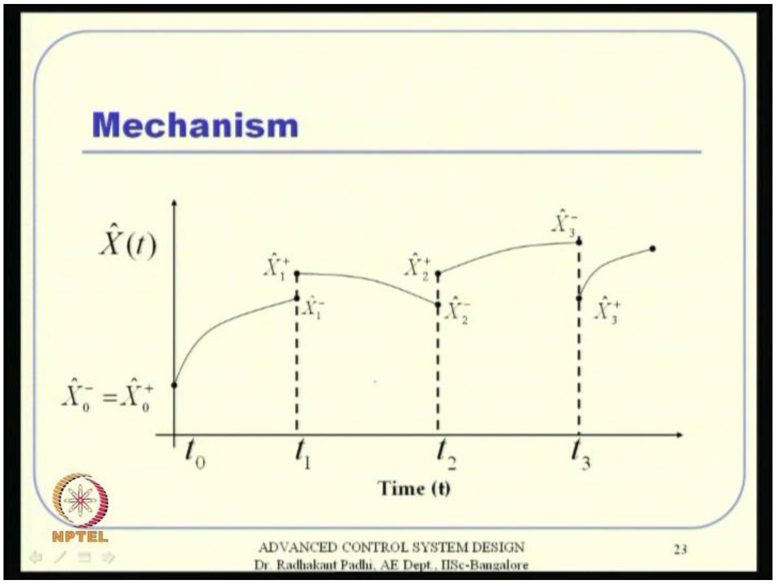


So, let us see that in brief, so what you are talking here is the state equation is is given in the in terms of continuous time, still it is linear, but now let us talk about time varying system dynamics and all. And the measurement equation is given in terms of discrete time, that is in

W T is continue versus, and V k is discrete sensor noise actually. So, they also satisfies this similar properties, and then remember one will say satisfy (()) the delta sin, and other one is derived delta, this is derived delta this is kind of delta.

So, that means, as long as your T is not equal to tau this is this is 0, if T equal to tau this is 1, and similarly if you k is not equal to 0, then it is 0 and if it is equal to 0 this is 1, that is that way (Refer Slide Time: 38:29). So, this is done because, it is a discrete I mean, this is a continuous time formulation and this is done, because it is a discrete time formulation. So, probably this, there is a small mistake out this k is not equal to (()).

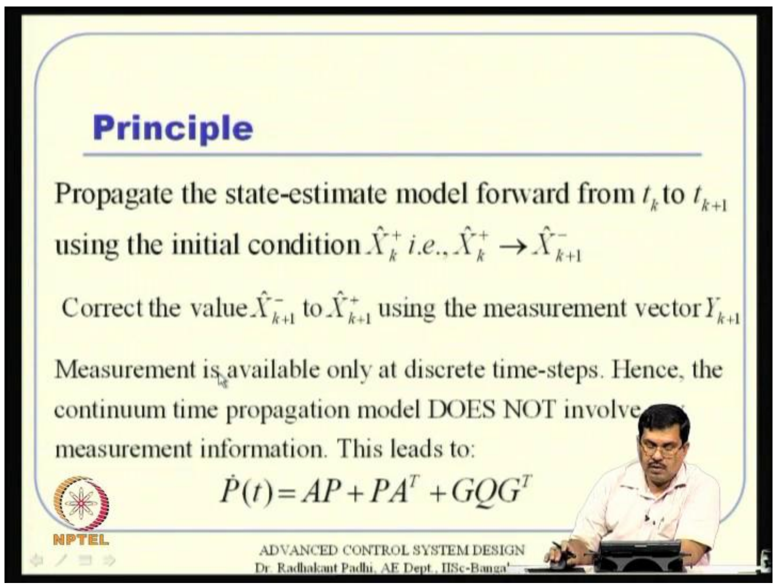
(Refer Slide Time: 38:55)



And this is what I was talking earlier, this is the discrete I mean, this prediction correction mechanism is what is popular here, and so assuming some values are T 0, we will be able to predict in a fairly continuous time manner, using any numerical scheme not necessarily 0 out of think like that; you can you can just use any continuous time propagation iterations and all to predict this. So, once you predicted, then the sensor information is available I will I will be able to correct it, I will I mean, because the sensor will give you assuming that that is, that will give me correct information and all that.

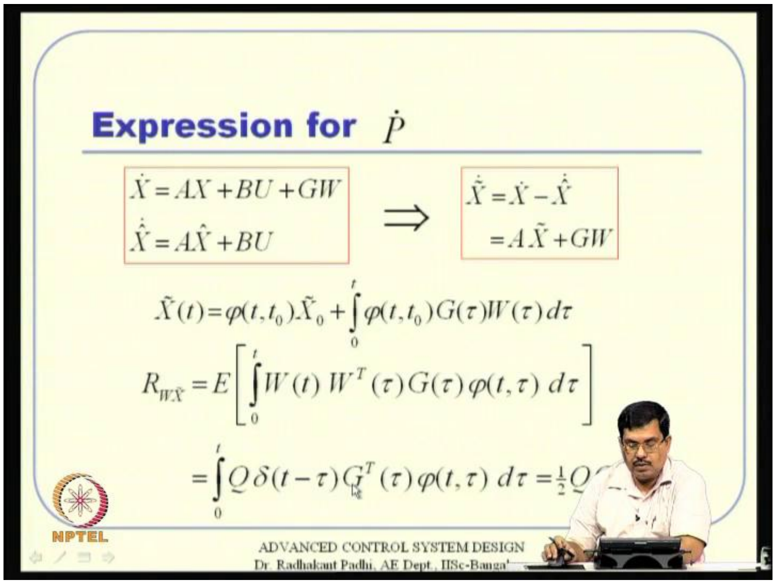
I will be able to make take the advantage of that, and then tell I have already given here what let me not operate further on that, and let me correct myself a little there and then operate further. So, this is prediction, this is correction, again prediction, because in between I do not have any sensor information, I have to simply rely on what I have, so starting with this initial condition, I will be able to predict (Refer Slide Time: 39:43). (()) sensor information again, so let me correct again from here, so like that it will proceed actually.

(Refer Slide Time: 40:01)



So, in principle, it is similar to what you have seen before, so what it tells is propagate the state estimate model forward from T k to T k plus 1 using the initial condition, and correct the information at one of the measurement is available as far I told you before. So, what it runs out is that, unlike this continuous time Kalman filter for expressions and all, because the measurement is not available in between in between measurement is simply not this. It turns out that the covariance matrix which is actually a continuous time expression, this time is given in a simplified form that that non-linear term, what you had earlier is not there basically.

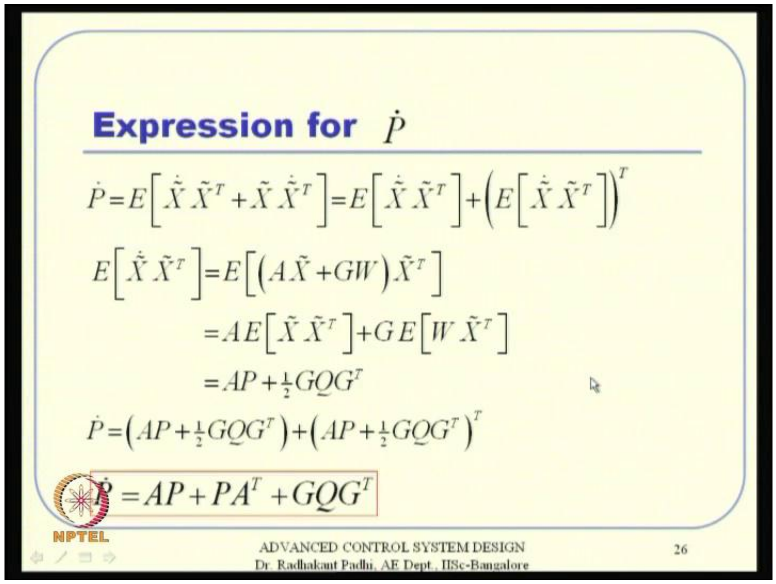
(Refer Slide Time: 40:50)



So, this can we derive fairly rather easily, actually if you see this expression for P dot this X dot is given like that, X hat is given like that, X hat dot is a prediction part and remember that there is no noise in that actually. So, now, if I see that the error definition, and error dynamics and think like that X tilde are X minus X hat, so X tilde dot is X dot minus X hat dot. Because, I just take difference between that to B U B U goes and I left I am left out with A times X minus X hat which is X tilde plus G W actually. So, I will be able to operate based on this X tilde dot is A X tilde plus G W.

So, remember this time varying system dynamics and time varying system dynamics with W being the input matrix and think like that, I will be able to write the solution of that in a continuous time manner. Where this phi T 0 happens to be like a state transition matrix and think like that, and then R W X tilde, now is X tilde is available. So, covariance matrix for R W tilde will be given something like this, carryout the expression simplified by simplify that and between and thing like that we will left we will be left out with this this expression really.

(Refer Slide Time: 42:01)



Now, what you do now, P dot is is by definition like this this two terms, so we have one term and the same term transpose, and one term we have we need to derive. So, this one term is is can be derived simply by definition, with X tilde dot we just derive here, so you put it there, and X tilde is we keep it, then we proceed further like that way; and then at E times expected value of X, see E times expected value of X tilde times X tilde transpose this by definition is P matrix anyway. So, we will put P matrix here than this one we just derived it the half times Q C transpose and then G is already there here, so this is half time Q C transpose, so that will happen to be like that. So, P dot is this expression plus the same expression transpose, so this expression what you have here plus the same expression transpose.

Now, you are simplify these two terms, and half half will be combined together think like that, so P dot happens to be like that basically, that is what I told you here P dot happens to be like that. So, what is the summary part of it again here, the model is given like this, the continuous time model measurement is like this discrete time, initialize the state vector as well as the covariance matrix, we will compute the gain which is actually discrete time manner k is like this (Refer Slide Time: 43:05).

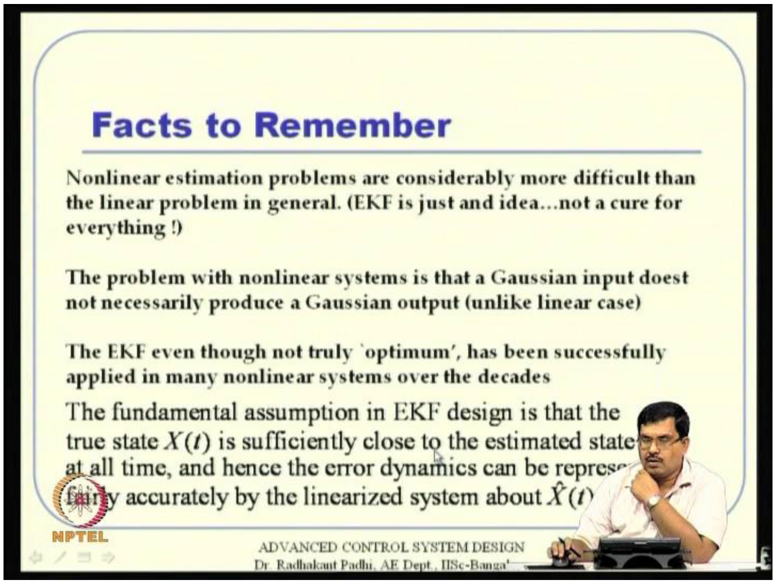
Well then once you once you compute the gain, you will be able to correct the state vector, starting from the predicted value you can get the updated value, using this innovation term. Then we need to get ready with the next time step, next time step you can propagate that way, the state equation can be propagated using this this part of a system dynamic; not including noise. And this covariance matrix can also be propagated actually again this this symmetric form is preferable.

So, propagation sense we will make use of this propagation as I told and we will also make use of this propagation term, and I mean this continuous time covariance matrix to propagate actually alright. So, this is what I mean is this this covariance matrix can can also be updated there, let me (()) not only updating this state equation state information here with the measurement, we will also update the covariance matrix here. Then we will propagate the covariance matrix using the continuous time expression that is available, so this is update stage (Refer Slide Time: 43:52).

Both the both the state vector as well as covariance matrix will be updated, then they will be propagated by one more time state actually, remember the way it is mechanized it does not really require uniform delta t that means, the propagation time state and updated time state need not be same actually. You keep on propagating as long as the sensor information is not available, the moment sensor information is available, you update the information that you have actually, so that is a where you need to have mechanization actually.

And so this what I already told, and then this is what you told is like P dot expression is a continuous time (()) equation, if you see this there is no more Riccati equation basically, it is in a discrete, I mean continuous discrete framework, it terms out to be nothing but, a (()) equations sort of thing. Now, finally, with this information we will be able to touch up on this this this external command filter, which is very popular in industry, and most of the time people use that, that extended Kalman filter.

(Refer Slide Time: 45:37)



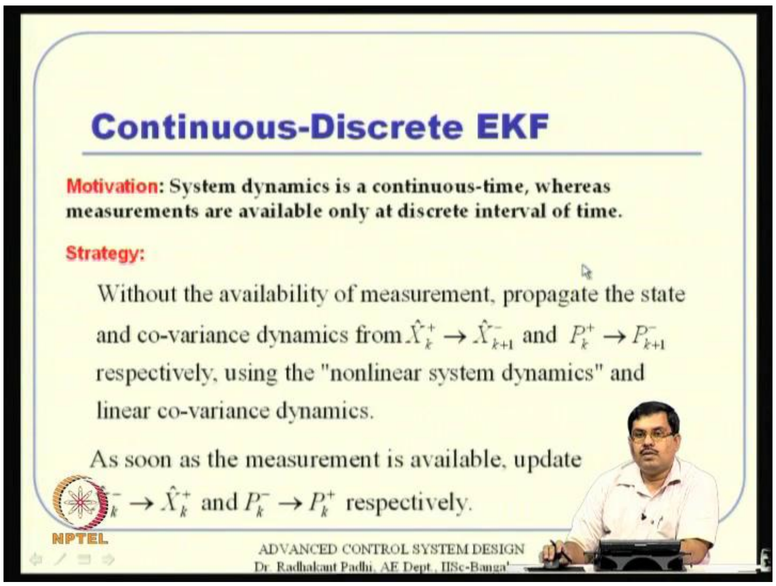
So, first you remember, first is non-linear estimation problems are are considerably more difficult, than the linear problems in general, and EKF turn out to be an idea, but not a cure for everything that means, that you just cannot have, because I know EKF, I will I will be able to solve everything. What surprisingly turns out that EKF actually, works for a variety of problems, actually large number of problems EKF does work, but some cases it may not work also because, EKF is not a very rigid precise, theory it is it is an extensive idea really.

So, what the problem, problem is like if the non-linear system, even if you take Gaussian input, it really does not translate, it does not retain in the Gaussian nature actually, if it is a linear system the nature of the Gaussian nature remains in that what the non-linear case does not actually. What is the assumption, fundamental assumption, what you are assuming in in external Kalman filter, is that the true state is sufficiently close to the estimated state actually already; an implementation, it may not be actually initially you will not be able to guess a very close state information and all that; that is one of the most one of the difficulties of course. Because, universal stability like linear system theory is not available, guess will play a role and if your guess is too far away you will have a problem actually.

But, it turns out that you can have a fairly large amount of error in the initial guess, you can go through a large amount of transient, what it will ultimately settles out nicely actually.

Anyway, coming back what it assumes that true state is sufficiently, close to the estimated state, and hence we can actually linearize the non-linear system, about the estimated set and then use the linear theory that we have used discussed before actually, that is what the idea. So, let us talk about the most popular form continuous discrete EKF and that is most natural form also system dynamics is continuous, but measurement is discrete again.

(Refer Slide Time: 47:37)



So, what you do here, so similar idea, like we will be able to, like with without the availability of measurement, we will be able to propagate the state information as well as the covariance information for to the next time state. And as long as the system dynamics is available will be able to correct it actually sorry, as long as the measurement information is available, we can update this this predicted values to corrected values really, so that is that is the way it will really precede actually. Now, so what is the summary of this the entire derivation, we will not go through probably, but the summary part is something like this, we have the system dynamics like this, where f of X is T U this is the non-linear system dynamics remember that. But, we still assume that the noise part is additive and noise part appears linearly rather, and this output is is again a non-linear expression plus V k. Remember, the system dynamics is continuous, but like measurement is actually discrete probably you can put here k.

So, again the with mechanize, we need some initial condition and then that initial condition we have to guess, and so except k sorry, except minus T 0 is X 0, that is a guess, and P 0 minus expected value of that, and that also value of this matrix needs to be guessed, most of the time these values are guessed as diagonal matrices anyway. So, this is this is what is required, then we will have again computation like fairly similar to what we discussed in the continuous discrete form of regular Kalman filter for linear systems, we will do that, but remember these are matrices now.

So, matrices are this C k minus what you are using here is to be estimated this way actually that means, you take linearization of this output equation. So, C k minus is linearized form of this equation actually, so that is what we will go here. And similarly, we will need, I mean once you compute this Kalman gain, you will be able to use that for the update equations, and this expressions and all we have derived before. So, that is exactly same thing that we will be using, and then the after we are done with this we need to have propagation thing, and propagation happens to be like that.

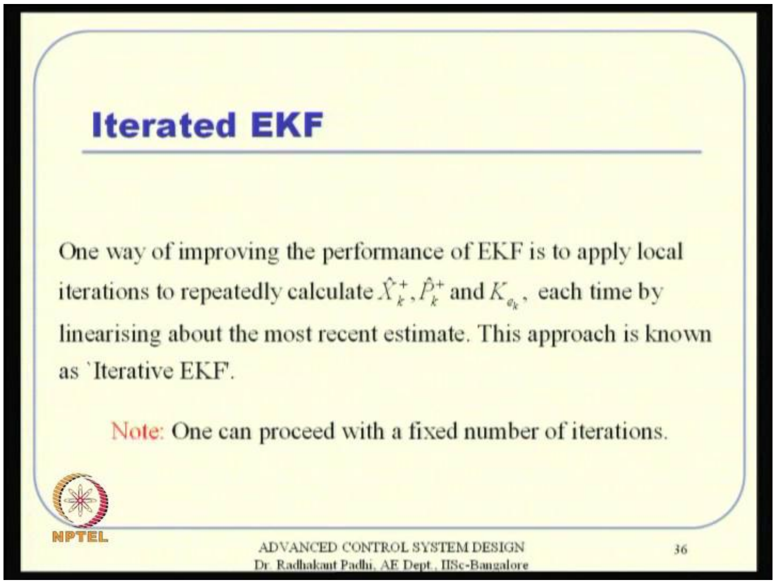
So, we have a state equation which can be propagated without noise part of course, and then it is a covariance matrix, I mean differential equation which will need to integrate each other. So, that using that will be able to propagate, and here this a matrix is again a Jacobean matrix this is derived del f by del X, and which is evaluated about the current estimate X hat T. So, A T and C T will be required here, C k minus and all that, and these are actually a nothing but, the linearization of the system dynamics and output equation what you have, about the current estimate what you have actually.

So, once you are ready with these matrices, you can keep that as a that point of time that is my time varying system dynamics matrix that I have, and for that I already know how to deal with actually, because continuous discrete Kalman filter for linear system is available to be, so this is how how you operate actually. So, every point of time you generate A and C matrices about it something like a time varying linear matrix sort of thing, and then using your like regular linear system Kalman theory Kalman filter theory you will be able to compute the necessity matrices and all, so so again this operates based on this prediction correction, prediction correction like that actually. So, we also have an idea of this something called iterated EKF, that is the fundamental EKF anyway, but you will see that, the plenty of ideas of various extensions. And then various arts and all like how do you implement, what do you do depends on your system dynamics, depends on your experience, depends on your insight to the problem, many many many things will happen around that actually.

So, one of that happens to be this iterated EKF however, that tell once I update this equations here, let me again go back and tell can I can I update one more time, can I update one more time like that (()); you get the system matrix if this is talking about some sort of a linearization here, and linearization is about my predicted value. So, let me do the linearization again the I mean with respect to the corrected value also, and operate the filtering equation one more time actually. So, that one you can keep on doing that, because you keep on updating your state in covariance matrix anyway.

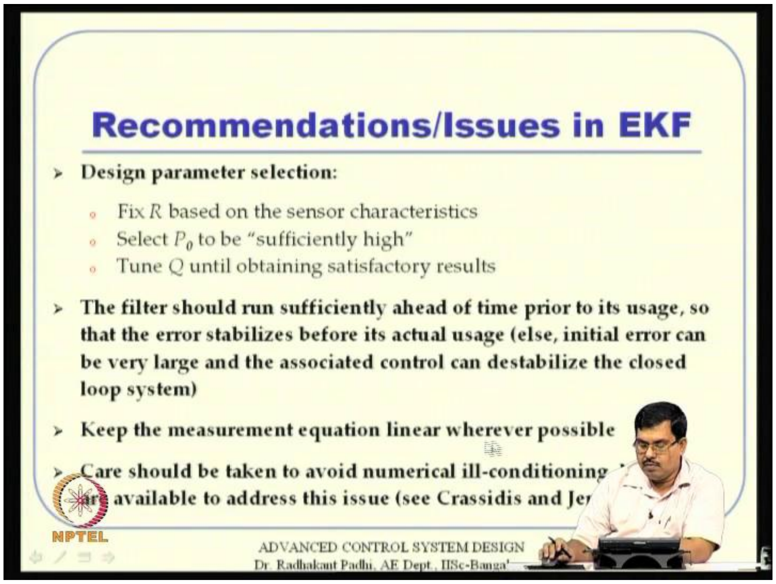
So, I will just update it and go back and tell I will reevaluate this this C k X k minus matrix about this except k plus norm and then I will use that one actually norm, so you keep on updating several times. So, before you proceed, I mean talk somewhere preferably after some conversions happen, and then you start your propagation equations actually, you start using a propagation equation.

(Refer Slide Time: 52:43)



This is the whole idea of iterated EKF and one can proceed with a fixed number of iterations, does not have to really wait for convergence and all, we can simply use some (()) and all that, before you proceed further; this is remember, these are all like like real time computation is should be addressed also, because without that this (()) also we will see. So, we can use it based on fixed number of iteration then go to the next step, go to the next step like that way actually alright.

(Refer Slide Time: 53:11)



So, there are various recommendations and issues in EKF, so before talking I mean before stopping this lecture, and winding up this course let me talk, several small small recommendation that are given for successful implementation of EKF, and again I will talk in a very generic sense basically. So, first thing is the design choice and all that, so design parameters lot of times you will see in a literature, that they are the recommendations that will perform.

So, remember there is the tuning sense, what we really need is a initial condition guess for the state vector, and other than that we need values for this matrices R P naught and Q right, P Q and I mean what they call P naught Q and R sort of thing, but you can think of that. So, what they tell is like first thing is you fix your R matrix, because R is nothing but, sensor providence matrix. So, depending on what type of sensors that you have, that will that will be fixed from there, because you study the nature of your sensors and then find out the covariance matrix from that, and that is the reality, so do not try to that actually.

So, keep your R matrix fixed based on the sensors that you are using, then you select a P naught with sufficiently high, that the lot of times you will see that P naught value is recommended to be very high, and that actually helps in some sort of a stability behaviour actually. That means, even though your initial condition guess is large enough, (()) if your P naught is sufficiently high in terms of the filter thus converge, after some initial transient of course.

Now, once you fix R from the sensors and fix P naught sufficiently high, the only flexibility that you are left out with Q basically. So, you have to tune Q until I mean until you obtain satisfactory result basically, so that is the standard recommendation that that is there in in much literature actually. So, you can be say, if you see that, as I told this EKF and all, you it will go through initial large transient actually. So, initially it will got through large transient, and then ultimately it will try to settle and after that the error will be small actually.

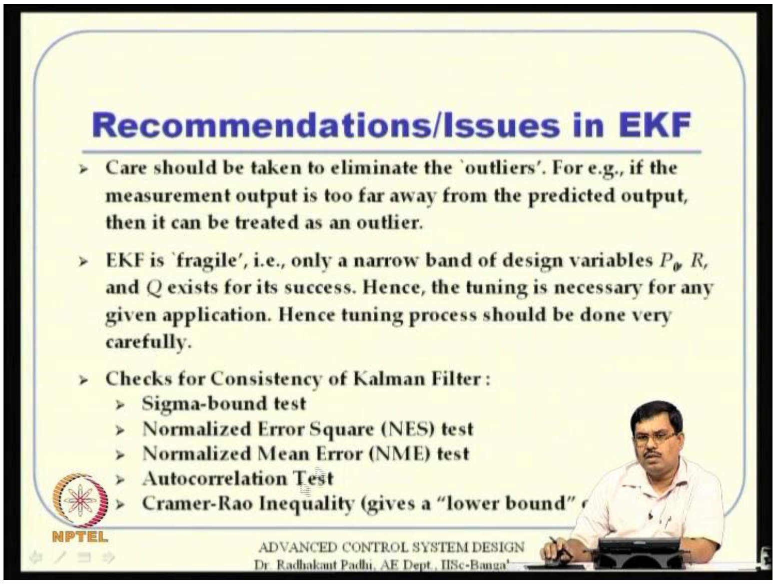
So, obviously, do not want to use this transients values, which are error nears any way, so what you what you have to do is probably you should you should run the filter sufficiently ahead of time prior to its usage. So, that the error stabilizes before it is usage actually. So, especially when you want to close in that means that may you want to operate the estimation into the control design and think like that. So, recommendation is used, you use the filter for a, I mean you start operating your filter in the background for some time, and then after sometime only you close the loop for the controller actually. So, that means, by that time the error is stabilized error is small, and then it will operate as if it will like to estimate the state and all that (()). So, anyway we will see that there is some sort of an initial time log given to the filter to stabilize, before it is actually usage actually.

There is another recommendation, I have seen many times, that we lot of people are in favor of the idea of that measurement equation is to be linear as much as possible, for example, I mean the theory tells that I can linearize the measurement equation as well and obtain a C matrix. But, in general what lot of practicing engineers actually work, but observe that if you if you keep the measurement equation linear wherever possible, then it leads to better stability for produce of the filter, that is just some sort of an experienced recommendation sort of thing, there is no no theory reasoning to make it off actually.

So, one example is for example, if you are talking about something like a (()) gradient problem (()) rather just information in polar coordinate, then the system dynamic associated with that is you should also use in polar coordinate; estimate the states in polar coordinate, and then transform it to Cartesian coordinate if you need actually. But, if you start with a Cartesian coordinate itself, and use polar coordinate information as of sensor output, then it will the measurement equation becomes non-linear, once you start linearizing it you are truncating the information. So, these are some of the issues that you want to avoid actually, so that is another recommendation.

(()) should be taken to avoid numerically conditioning actually, that you can see some of this accommodation in this crass dies, and book these are all of my discussion from this book, and you can see that one of the recommendation, as I told is you need to use this expression for for this P k plus instead of that simplified expression. If you use this this will have better numerical property, even though both these expressions are identical, mathematical, so that is another care, that you need to take actually.

(Refer Slide Time: 57:59)



One more big recommendation is we need to eliminate out layers actually, so sometimes this this sensor information are not really good, and we and this EKF, actually gives us a platform, because we have a predicted state, we have a measurement equation actually. Which is like Y and Y hat sort of thing, and if the innovation term is really very big, then I do not want to come for that, if it is reasonably small and all I will assume the sensor is correct I will update.

If it is too high too big, then probably momentarily I need to ignore that, I will consider that some sort of an out layer and just taken out that actually. Also remember that, EKF in general is fragile that that means, the only a narrow brand of design variables will be very good for for successful operation; if you do not lose your presence what I mean, keep on tuning this and ultimately, you will see that EKF actually works wonderfully for a for a number of problem.

But, you need to have presence for making it operation or (()) actually and there are ultimately, there are consistency checks of the Kalman filter, there are various checks actually, one is sigma bound test, normalized error square test, normalized mean square test, for autocorrelation test and all that, (()) inequality test and all that. So, there are various test available, and at least where few of them needs to be done to have confidence in your estimation and all that, that is a really recommended and they were need to do that actually (()).

There are limitations where linearization is not good general convergence guarantee convergence guarantee is not there and many issues, because of that there are other ideas available beyond EKF actually. And I will not discuss too many on that, if you are interested you can see EKF you can see particular filter, you can say actually kind of filtering lot and lot, lot things etcetera.

So, with that probably I will stop here but, then the reference books are available the first book is my most favorite book and that is what I have taken from; second one is also good, and you can see some other books available or very good application, book there are theory books like that actually, so with that I will stop here thanks a lot.