Optimal Control Guidance and Estimation

Prof. Radhakant Padhi

Department of Aerospace Engineering

Indian Institute of Science, Bangalore

Module No. # 12

Lecture No. # 26

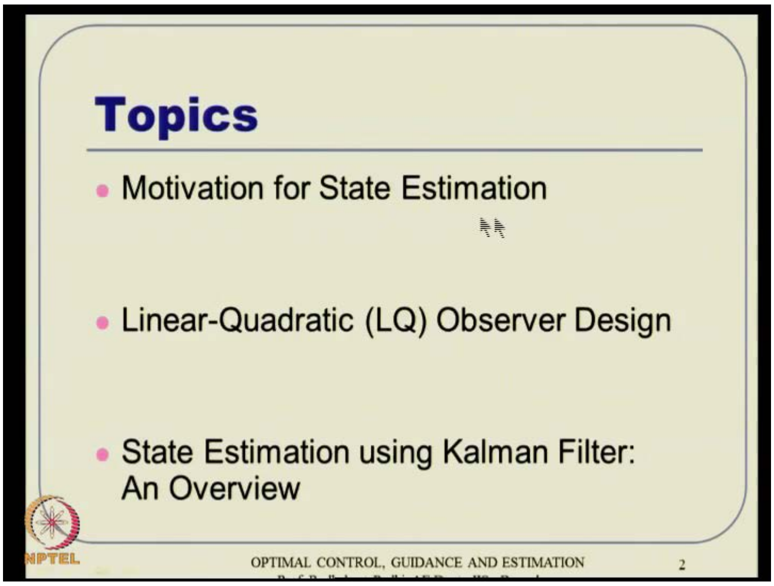
Linear Quadratic Observer & an Overview of State Estimation

All right. Let us continue with our lecture series. This is a different topic, since we have seen all these 25 lectures before and here, we are talking about estimation ideas or observer ideas and all that actually.

So, let us see further begin with little bit on linear quadratic observer and some sort of and as overview of state estimation. I am not talking about detailed derivations and all here which we will do in next couple of lectures anyway ok, but here we will simply see some of fundamental ideas behind this LQ observer and it also turns out to be some sort of a Kalman filter whether you realize it or not actually, ok.

Then, it also gives much more formalization of this state estimation technique, especially the Kalman filtering ideas in next couple of lectures. However, this particular lecture we will simply see what the basic idea is and what the mechanism are to operate it and all that actually, ok.

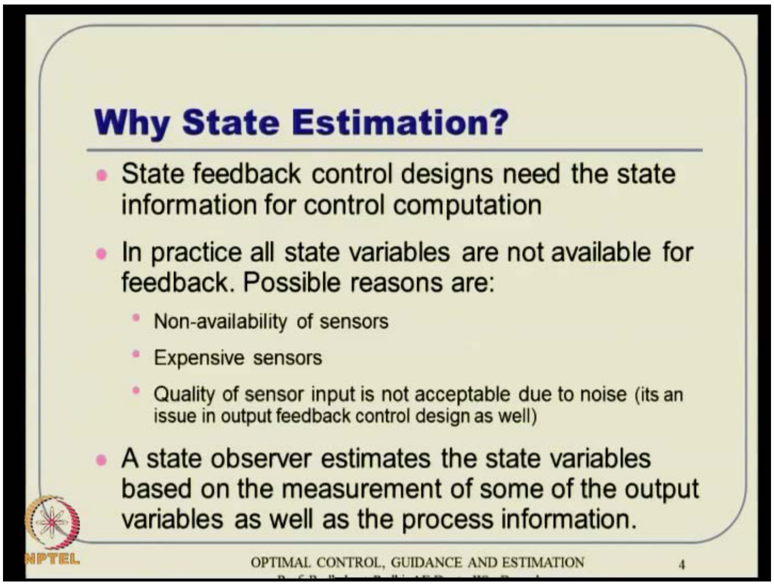
(Refer Slide Time: 01:25)



So, topics are motivation of state estimation, why do we want to do that, then LQ observer and some sort of an overview of state estimation, especially using Kalman filter is very very popular. It works for lot of problem something like that. All right. So, this motivation of state estimation first. Why do we want to do state estimation actually? Here, first thing is for state feedback control design, we really need information about state and hence, we need to do state estimation actually.

In practice, all state variables are not available for feedback and possible reasons can be non-availability of sensors and sensors simply not there or even if they are there, that can be sometimes quite expensive. For example, a seeker in the missile is typically very very expensive.

(Refer Slide Time: 01:43)



So, unless you really see out that everything else works, do not put it there, but you want to have some sort of validation before, I mean on that. So, preliminary launches and all typically are done with respect to lot of data basically, not necessarily with respect to seekers. That is the reason, the sensor is very expensive. We do not want to kind of just like that use it actually.

Sometimes, the quality of the sensor is not acceptable because of the noise content. Especially, if you see this memes level sensor which is there, it will give some data, some information about what is going on, but the information content is very nice actually, ok. So, when you have that, obviously, we want to use that sensor or may be many sensors like that, but then we will have to have a filter in the loop which will kind of strike the good information out of it actually, ok. So, these are some of the reasons why we want to do.

So, what is the destination actually? What is the idea here? A state observer estimates the state variables based on the measurement of some of the output variables as well as the process information, ok. So, process information means system plant model. It will use that process information as well as the output of the sensors for a finite segment of time actually. Just not at the one instant of time, but sequence of that type starts coming. Then, from there it will be able to kind of estimate the state.

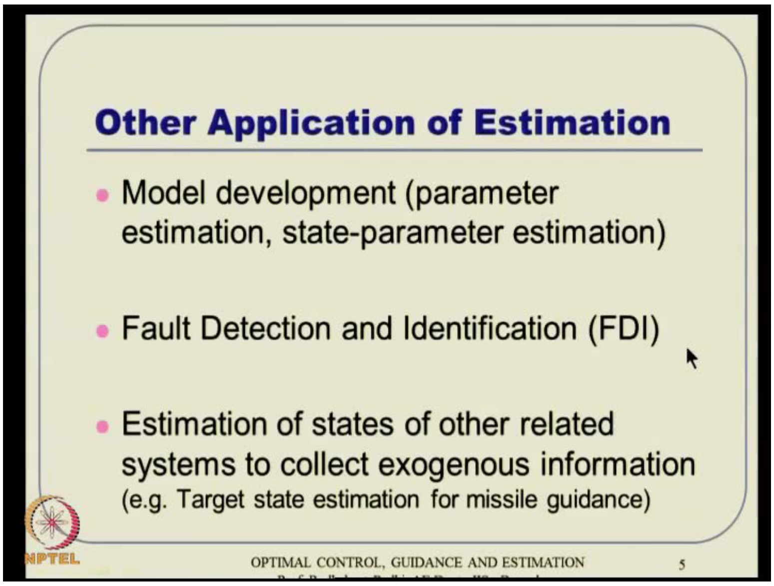
So, application of estimation is not just confined to feedback control, it is also you can think about model development, especially parameter estimation and combine state and parameter estimation sort of thing. So, that will be model development and somewhat sometime we are interested in fault detection and identification. That means, if you estimate the state, I mean we are not doing anything. We are simply observing the number and observation of that number can tell us many things about the status of the plant, status of the system. Then, it will also turn out to be that there is something wrong in that fault detection actually.

For example, probably nuclear power plant if we think about temperature control, our temperature monitoring rather, if you do, the temperature keeps on rising to some level, then it is probably very dangerous and need to sort on the part actually. So, there is something faulty going on there ok. So, that kind of information can be obtained from using this FDI techniques, Fault Detection and Identification techniques. Essentially, these are estimation thing.

Hence, estimation of states of other related systems to collect exogenous information and this is the thing that I am talking about that target information in missile gradients problems is typically done through our own sensor. Otherwise, target is never going to declare its own position velocity and tends of anymore thing like that. It is the missile of do find out what is going on in the target actually, ok. So, sometimes this exogenous input what is necessary for computing the command is done through this estimation actually.

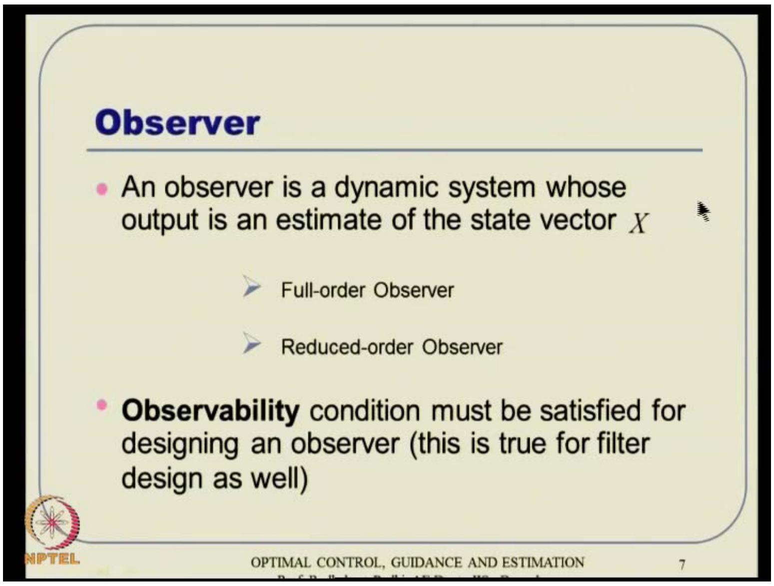
You can see various applications of estimation thing and unless we know something about estimation, the entire control theory is kind of only half explored actually, may not be completed actually. So, something, some ideas of estimation we must try it. All right.

(Refer Slide Time: 03:40)



So, LQ Observer Design first. Let us see what the idea is and here, we do not really require any random variable concept and we do not want to go the filtering ideas in this particular development actually. So, let us talk about some sort of noise pre-situation and in other words, system dynamics is non-perfectly and the system variables are also known properly basically, ok. Then, the output is also noise free, then what you do actually. That is the all idea here.

(Refer Slide Time: 05:56)



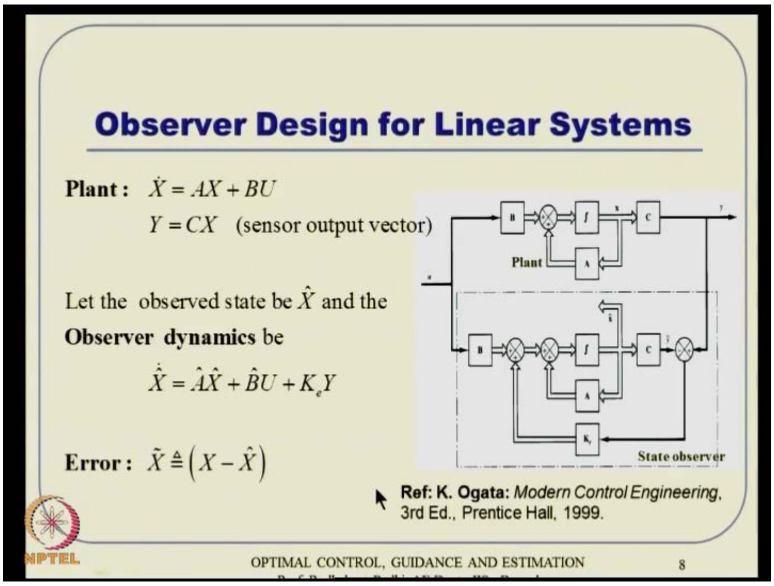
So, an observer is a dynamic system whose output is an estimate of the system vector x. That is the definition actually. So, essentially when we talk about why we got an observer, that what we are telling is we also got some sort of artificial system dynamics and that system dynamic output is nothing, but the state vector x actually, ok. Then, there ideas are full-order observer and reduced-order observer.

In other words, full-order observer tells, I will go ahead and find out the entire state all the time. Reduced-order observer tells some of the output vectors are actually nothing, but the state information and that output vector does not contain (()), I do not want to kind of estimate that. I will just simply take it and whatever I do not know, I will try to estimate that. So, that is called reduced-order observer actually and also this, remember this observability condition must be satisfied for designing an observer. No matter what you do, observability has to be here actually, ok.

Now, let us see this. We have got a linear system plant. We are talking about linear system dynamics here. So, you have AX dot is AX plus BU and Y is C X. So, motivated by this plant, what you have telling is you also propose an observer dynamics something like this, very similar towards the plant is x dot is AX plus BU is also X hat dot is a hat X hat plus B hat U plus this K e times Y because Y contains the output information. So, we want to put it that way. This our observer where the error is still the X defines as X minus X hat, ok. Now, the question is what is this A hat and B hat actually using that way, ok.

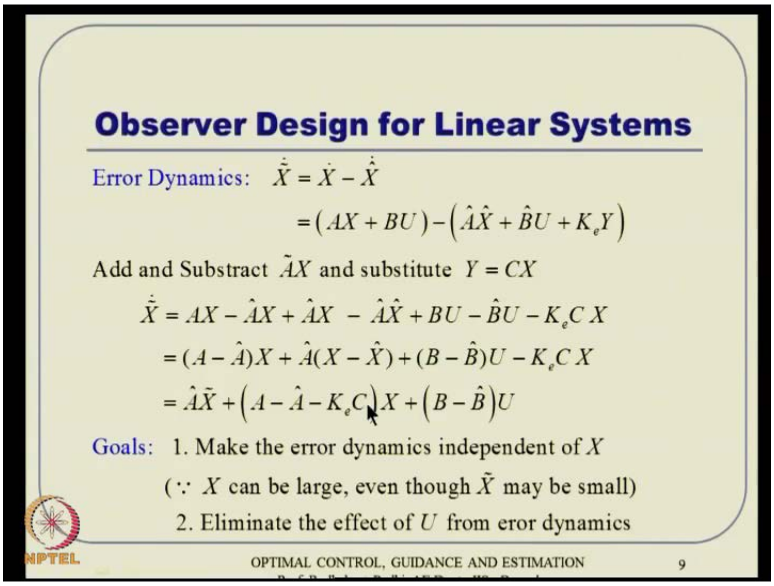
You can also think these two things. You can pictorially represent something like this. This top plant, top portion of the thing is actually nothing, but the original system dynamics X dot is something here is nothing, but A times X plus B times U. B times U coming here. That is what X dot U and Y is nothing, but CX. So, that part is the entire plant, the actual plant. This plant is the observer. So, in your one hand, you have got Y tilde, I mean this will be tilde actually, ok.

(Refer Slide Time: 07:05)



So, there is some notational things with respect to this is Y hat and then, there is Y basically. So, using this error information which is nothing, but let us Y tilde I put it on that way. Then, it can be observed, I mean this part of the dotted box, what you see here is nothing, but observer dynamic. So, this is also X hat in our notation. So, this is what happens here.

(Refer Slide Time: 08:51)

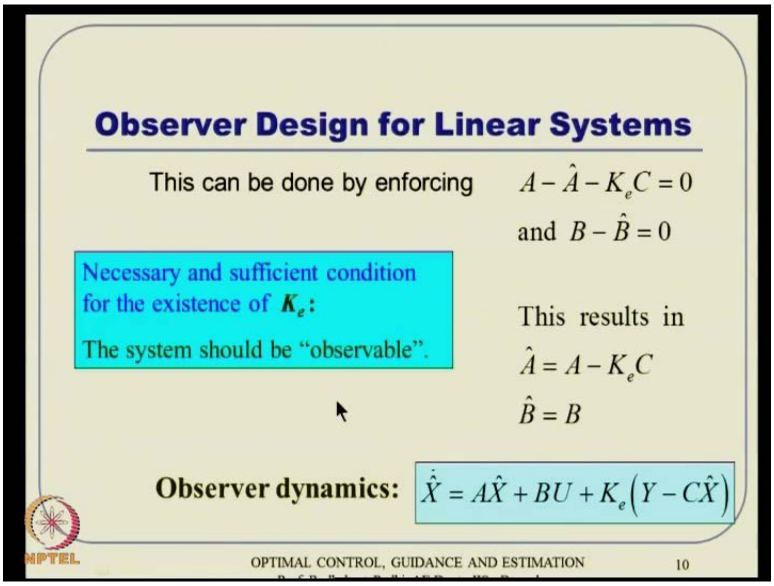


Now, we talk about error dynamics. Error dynamics is X tilde dot which is X minus X hat. So, X tilde dot is X square minus X hat. That X dot is AX plus BU and X hat dot is this expression what we had, I mean kind of proposing here. Then, we can add and subtract Y. Add and subtract these terms A tilde X and substitute Y equal to CX actually, ok. All right. So, whatever expression we have here right hand side, we add and subtract this term A tilde X and then, carry out the algebra like that.

So, we have AX plus BU, which is coming from here and then, this expression is there. So, I mean I am not going to explain all that whether the TF anyway. Also, to do that it turns out that is this part of that here is nothing, but something like A dot X tilde basically plus this term into S plus this one actually, ok. Now, this gives us some ideas actually that what is going on here basically, ok. Let us go step by step.

So, we are talking about this expression first and this expression. Now, if you look at this error dynamics X tilde dot, it has to be a function of X tilde only ideally speaking. In other words, whether X is big or U is big or small or X is big or small, it does not matter basically.

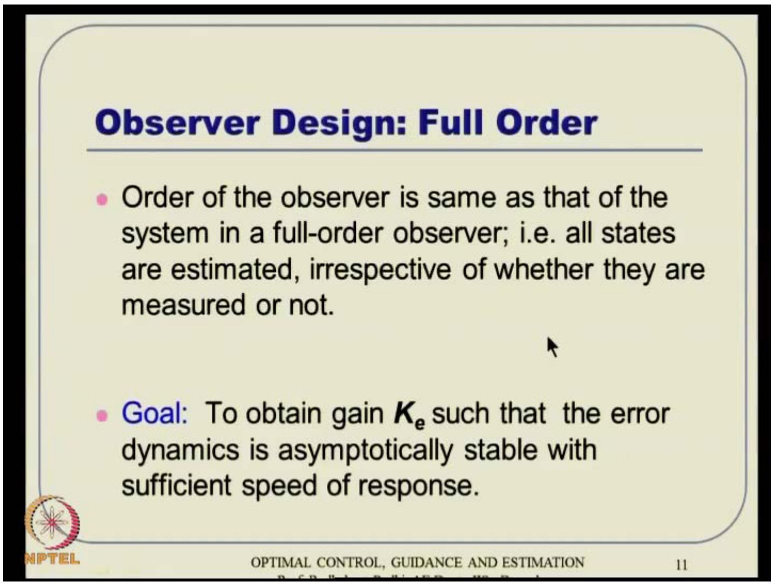
(Refer Slide Time: 09:58)



So, for that reason, we want to make the coefficients to be 0. That means, we got an expression for B hat and A hat from here. So, B equal to B hat and A equal to A hat plus K e C basically, sorry A hat equal to A minus K e C actually, ok. So, B equal to B hat and A hat equal to A minus K e C. Now, you substitute that till that K, this is my X hat dot. That is why started with only A hat and B hat was not known. So, now, B hat is B and A hat is A minus K e C actually, ok.

So, now, I substitute that and that A minus K e C, this component I will put it somewhere here now basically and then, this particular term Y minus CX hat turns out to be something called innovation actually. This is well estimated or predicted output actually and this is the actual output. So, the difference between that is something, some new information which is coming to this observer. If the difference is not there, then it is as good as the original plan actually, ok and it turns out that necessary and sufficient condition for existence of this gain K e is that the system should be observable, ok.

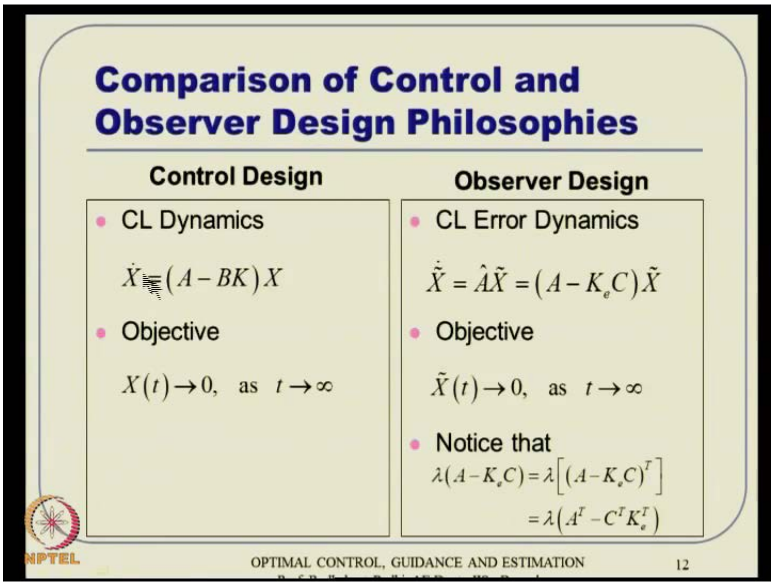
(Refer Slide Time: 11:45)



So, the observer design if you are interested in a full order observer, then order of the observer is same as that of the system of full order observer. That means, all states are estimated irrespective of whether they are measured or not. So, that is what we are talking about here at this order of the observers are there, but we are not going to discuss that because we are slowly interested in Kalman filter actually, ok.

So, we will go out and do that way actually. The idea here is to obtain this expression K e. Anyway, we have got a structure already and we are known because that is what the system dynamics is given to us actually. Typically, those are to be known things. What is unknown in the entire thing is only K e actually if you ask me. So, this K e thing is the only thing, which is interested in designing actually because a, b and c are known to us.

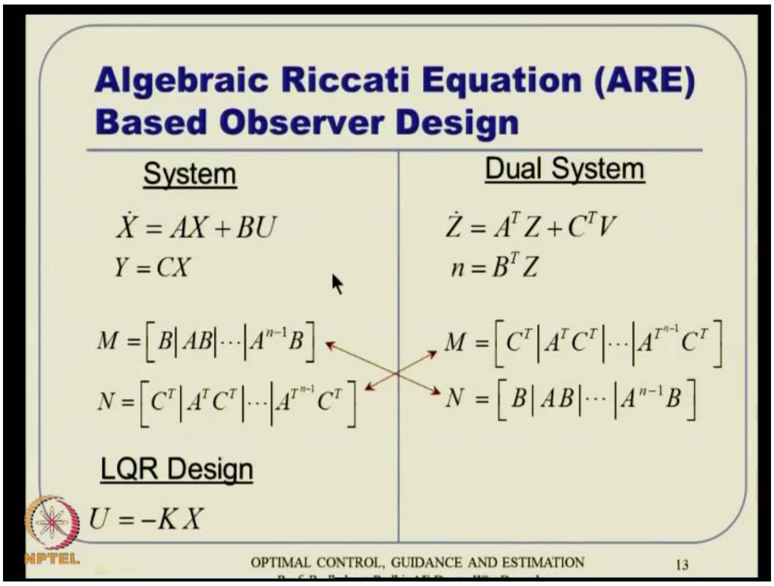
(Refer Slide Time: 12:45)



Then, the question is how do you do that? Now, here we can see an interesting analogy. We do not want to go through the total thing actually. Now, the analogy turns out to be something like that. We know the control design philosophy already and in closed loop control design; this is the system dynamics in closed loop. Remember, X dot equals to X close BU and U equal to minus KX. So, that one if you substitute, it turns out to be something like this. Then, the objective there were X of t goes to 0 and that is the whole philosophy based on which we deigned this k basically and we want to extract this benefit out of this thing in the observer designed actually. We do not want to, I mean kind of go back and redo the entire method again. So, this is the objective here and in this process, we know to design k basically, either well placement, I mean what here we are talking about an LQ design. So, we are talking about knowing this k from LQR sort of ideas actually.

So, we know how to design this k, but coming to the error dynamics in the observer side, this is what we got, x tilde dot, after doing that tilde dot is AX tilde dot. That is all we have got actually, ok and if we substitute for A, that is what you get. So, if you look at this expression X dot equal to A minus B X into X and here objective was X tilde dot A minus K e c minus X tilde. Here the objective is X should go to 0 and here the objective is that X tilde should go to 0. A very similar objective actually what you say of similar system dynamics and similar objectives. Again I emphasise the word similar because here it is not really same. The difference here is this gain K appears in the right hand side, whereas it appears in the left hand side here. Thus, the whole difficulty first, but however, it tells out that if you talk about a transpose of any matrix, then the Eigen values remains same. Eigen values of this matrix A minus K e C is same as Eigen values of A minus K e C transpose. Now, this transpose if you expand, it turns out to be like this A transpose minus C transpose K e transpose because this transpose will tend to alter actually.

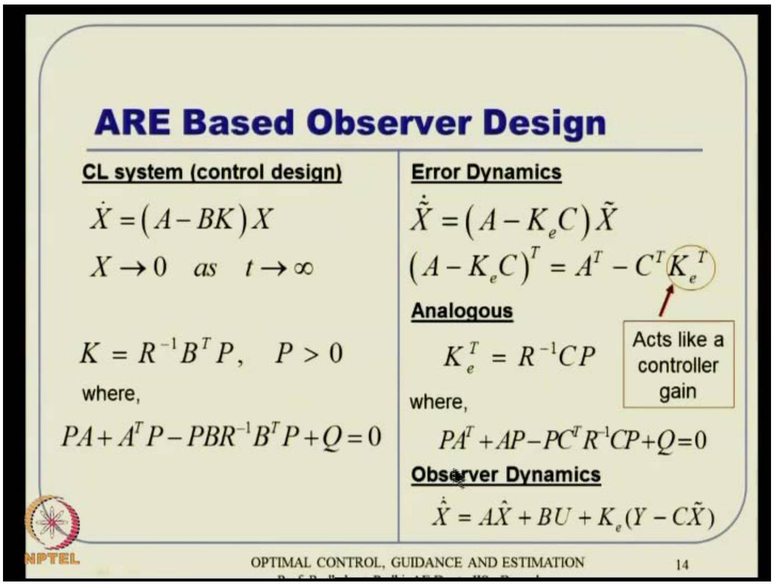
(Refer Slide Time: 15:19)



Now, suddenly we have got some hope because K e T, I mean K e transpose happens to be appear in the right hand side only. Here it was left hand side; sorry here it was left hand side, they got right hand side and this right hand side is compatible to what you know in control theory basically. Now, you go back to the analogy part of it. We have got this as a system dynamics and here was the control ability matrix, here the observe ability matrix and here constitute dual system like that. Z dot is a transpose Z plus c transpose v and output N equal to B transpose Z.

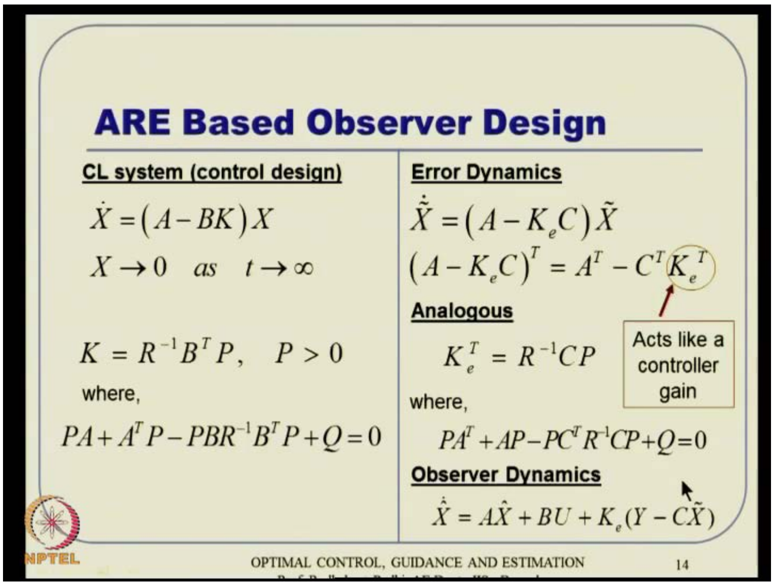
Suppose, you contribute some constitute something like a dual system here in this form, then you can construct this controllability of this particular dual system and observability of this particular dual system that way, but it turns out that the controllability matrix for the system, original system is actually observability of the dual system and observability of the original system happens to be controllability matrix of the original system actually, sorry dual system.

(Refer Slide Time: 16:10)



So, here we got LQR design U equal to minus KX and that what we want to extract actually, all right. So, control design sense which is X dot equal to A minus BKX. So, objective was X has to go to 0 and we know how to design K in terms of LQR. K is nothing, but R inverse B transpose P, where P is a positive definite matrix and P is solution of this Riccati equation also basically.

(Refer Slider Time: 16:10)



Now, that side of the story we have got this aero dynamics and this transpose is nothing, but similar and something like if you look at it a little bit, this K e transpose acts like a controller gain for this system actually. So, this K e transpose, we can actually put the formula here, similar formula K e transpose R inverse C transpose R inverse CP, where this P matrix what you are looking, it is designed based on this Riccati equation rather. We are now here to remove A transpose and C transpose are here for matrix and system matrix actually.

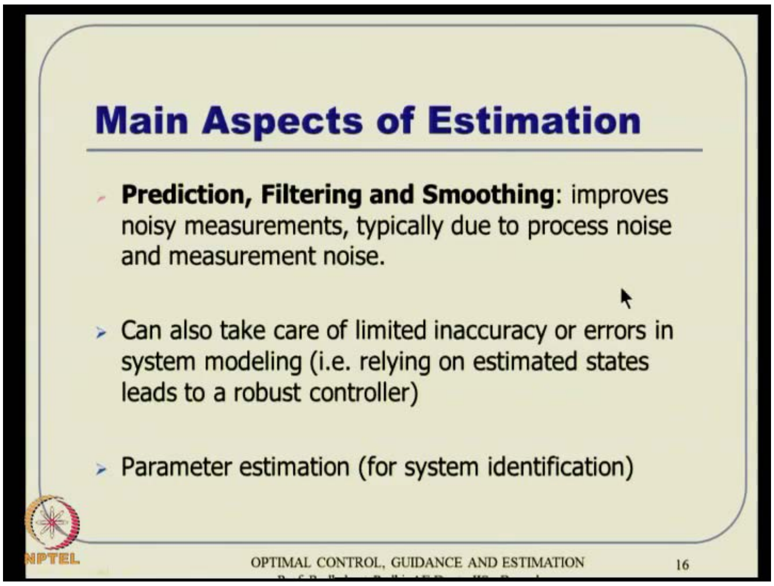
So, using that it turns out that this Riccati equation is something like this, all right. So, the whole idea is what it turns out is something like filter Riccati equation what you call or observer Riccati equation. So, we solve this Riccati matrix from these equations and then, compute K e transpose like this and once you compute K e transpose, we have got K e obviously and once you got K e, the observer dynamics is given to us actually. Thus, the observer dynamics we started something like this, right. So, the whole point is to know K e.

K e is now known actually and remember, we did not go to again the fundamental philosophy of LQR and things like that. We have simply tried to exact it by constructing a dual system, all right. So, this is your observer dynamics. In other words, you start with any any initial condition except 0 and when this system dynamics is parallel in computer only, the only thing that you have to keep on using is y actually. Anything else is kind of (()). Well, there is a small error again. There is some mismatch of notations and sort of thing will correct this. This is again this H hat here, not error, but the observer state actually, all right. Now, this is all about Kalman filter and LQ observer and continues time actually, ok.

Now, moving on to filter. The whole idea here is we do not get too much of flexibility actually by doing this. We simply try to take advantage of LQR theory, but what is Q and what is R that remains kind of this, I mean kind of filter are observer tuning in matrix variables which you do not get a clue of how to adjust it properly sort of thing. Just it happens to work actually. That is not a problem, all right.

Now, moving on to Kalman filter ideas which is the extension of these ideas and all that, again will need some sort of random variables concepts and things like that, which we are going to discuss I mean subsequently basically.

(Refer Slide Time: 19:43)



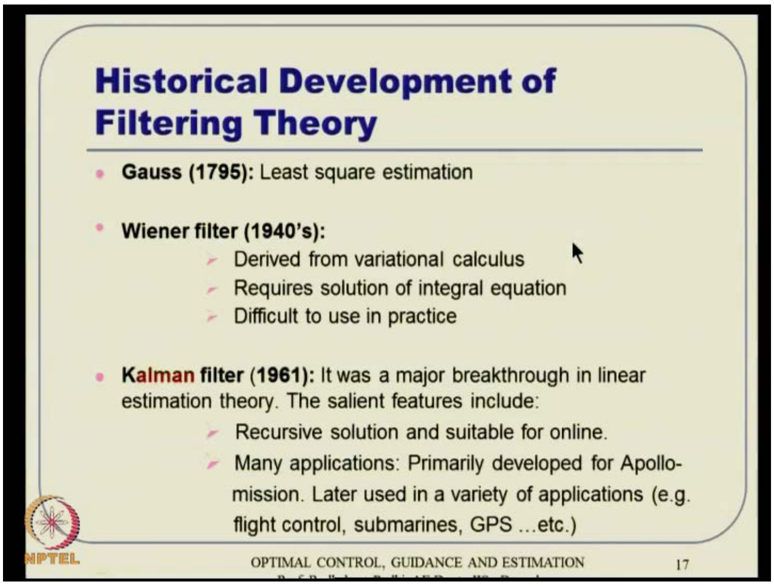
Now, let us see a little bit overview of what is this Kalman filter about actually. So, main aspect of estimation if you see, it talks about prediction, filtering and smoothing. So, prediction is what is going to happen in the future, filtering is what is happening right now and smoothing is what has happened before and this typically, these three algorithms are based on somewhat similar mathematics, but they have different simplification actually. Especially, when you have this process noise and measurement noise coming into picture and some noise characteristic are available to us, why not using that explicitly, ok. Instead of realizing on simply some observer thing which does not talk about any noise anywhere actually. Thus, all idea there.

So, prediction, filtering and smoothing are the three things. Prediction is what will happen, based on what I have observed until now, using that can I predict what is going to happen in future and obviously, I lead a state predict for that. In other words, state model for that actually and filtering is I got a sequence of data already and based on which can I estimate something what is in current state right now. Smoothing is process is done, but I want to see what has happened before in a much more better sense actually. There are the applications also for smoothing and all. Especially, when you talked about parameter identification, you go for something like a test flight basically. So, flight is already done. So, we have got the data already and come back and try to fit some parameters. So, why not using smoothing radius? Why should you use only filtering there actually?

So, they are the things which can think about where it is. I mean what is applicable, where actually and it turns out nicely that this particular formulation of the estimation can also take care in a limited accuracy sense the errors in the system modeling. That means, if you have some sort of system dynamics in accuracy, you can think that in accuracy part of it is nothing, but a noise actually.

So, in other words, it can handle that part is a kind of much more better sense actually. Especially, in structural vibration sense, we typically take about six third or I mean even, I mean finite order model. Then, whatever remains after that is actually noise, but ideally speaking, they are not noise really because there is a physics process goes behind that, but you can interpret that is a noise for particular applications point of view actually because in that sense, it can take care of limited inaccuracy or errors in the system modeling actually. So, it can also be used some in parameter estimation for system identification. I have already talked about that.

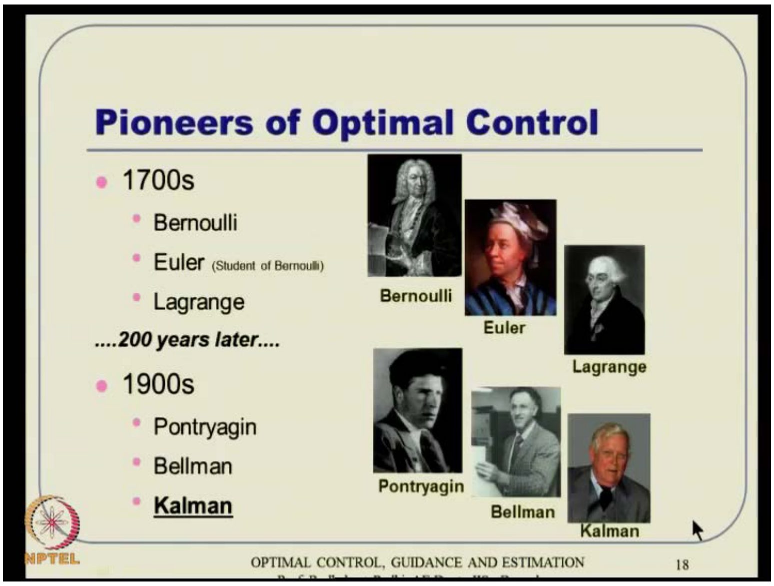
(Refer Slide Time: 22:28)



Now, if you see a history part of it and it goes all the way back in 1795 with when Gauss first proposed this least square estimation. Then, for a long time nothing happened. People used Gauss least square estimation. Even now it is being used, especially for static optimization ideas and all that. But then, in 1940 Wiener proposed this new idea which is something like filter and he derived from variation of calculus point of view and unfortunately, it requires solution of integral equation and turns out it is difficult to use in practice. However, the idea content was good and hence, Kalman got interested in that and he proposed radically. He got quite motivated by his ideas, but he reproposed in a very different sense, in a time domain sense actually, whereas wiener filter was in frequency domain lastly, ok. So, he proposed something which was readily understood and it actually had recursive solutions. So, hence, computational it is very less demanding and things like that. That is why, it became quite popular actually.

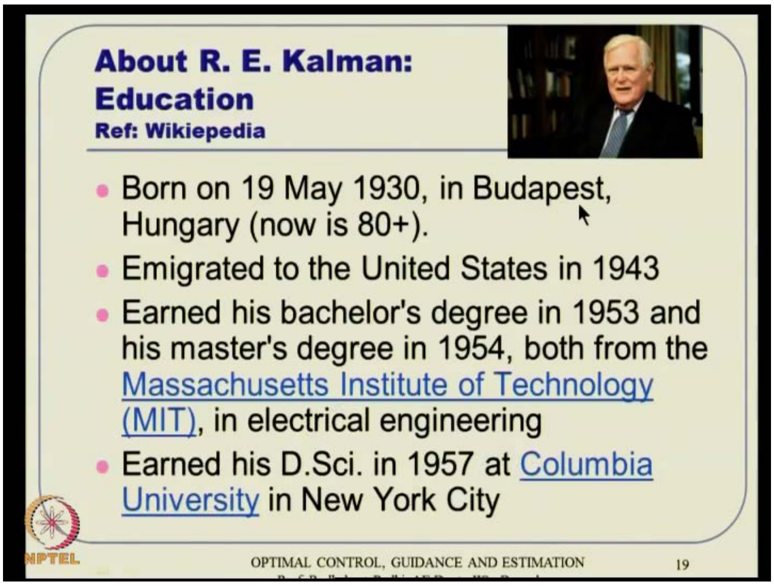
So, around 61, it was a major breakthrough sort of thing in linear estimation theory and the salient features of that including recursive solution and in suitable for online applications and subsequently, it has been applied in almost all applications of engineering actually, all domains of engineering. Even though, it was primarily developed during Apollo mission that the moon landing mission actually of NASA, but later it was used for variety of applications, flight control, submarines, GPS, I mean parameterized estimation system identification, I mean automobiles you can talk about, electrical application, robotics, anywhere you think about Kalman filter has been used actually, ok.

(Refer Slide Time: 24:35)



So, this is the little bit history part of it. Again, if you revisit the pioneers of optimal control in a historical sense, then something like Bernoulli, Euler, Lagrange comes under 1700 century, but 200 years later, all these 1900 or something like Pontryagin, Bellman and Kalman. Kalman has done lot of work in linear system theories point of view, especially in LQ design of control light, I mean LQR theory or essentially, LQ design, linear quadratic design and various ideas associated with that using controllability, observability and things like that. Many things he has proposed, but he is fundamentally known for Kalman filter actually.

(Refer Slide Time: 25:19)



Also little bit on Kalman just to side of see historical and sort of thing. He was born on 1930 in Hungary and obviously, now he is 82 plus actually and emigrated in to US in 1943 and around 50, that is where the Second World War was going on actually that time lot of immigration happened from Europe to US. He was one of that and then, he earned his bachelor degree in 53 and his master’s degree in 54, both from MIT in electrical engineering.

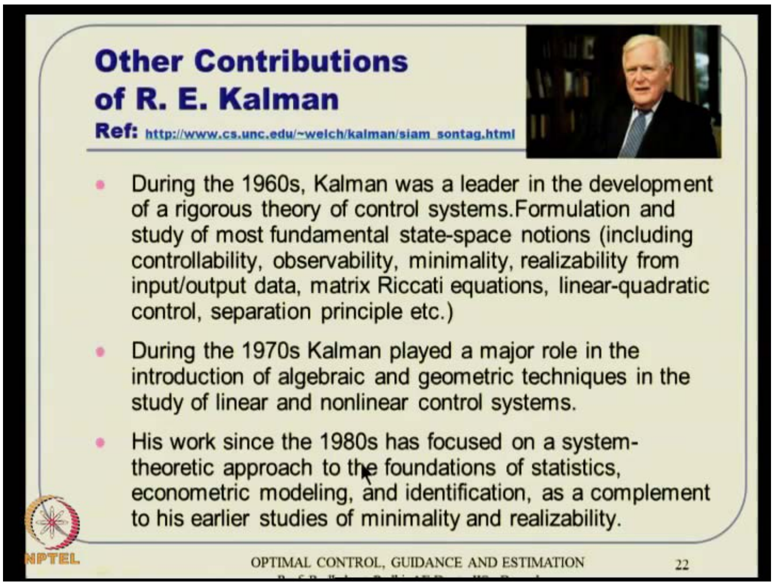
Then, he earned his doctor of science in 57 from Columbia University in New York City and then, he was actually a research mathematician in something called RIAS. That means, Research Institute for Advanced Studies in Baltimore, Maryland from 58 until 64 and this is where his legendary papers appeared actually and then, Stanford university professor from 64 to 71. Then, he migrated to university of Florida from 71 to 92 he was there and then, from I mean he got slowly associated from 73 onwards with Swiss Federal Institute of Technology, which is famously known as ETH in Zurich, Switzerland. Currently he is an emeritus professor in all the three places Swiss Federal

Institute of Technology again, ETH and University of Florida as well as Stanford University actually.

So, that is the history or background of Kalman and essentially is Kalman's ideas on filtering were initially met with some skepticism when lot of people thought this is not a good idea actually and so much, so that he was his paper was not accepted in IEEE and he was actually forced to kind of resubmit in publication, Journal of Basic Engineering from mechanical engineering. So, that what it is actually, ok, but he had lot more success in presenting his ideas while visiting this NASA Lab as now commercially known as NASA Ames Research Center in 1960. This led to the use of filter in Apollo program and then, it followed by space shuttle program, navy submarines, aerospace vehicles, various aerospace unmanned aerospace vehicles and weapons systems, cruise missiles and rest is history basically. Many things are applied after that, many problems actually.

During 60s, Kalman was a leader in the development of a rigorous theory and during, I mean it is actually, he has proposed many things based on controllability, observability, minimality, realizability, matrix Riccati equations, linear-quadratic control, separation principle various things which are very well path looking sort of ideas basically and in 70s, he played a major role in introducing this algebraic and geometric techniques in study of linear and non-linear control systems as well actually. In 80s, he has focused on very different ideas on economic modeling, identification and things like that actually, ok.

(Refer Slide Time: 28:00)

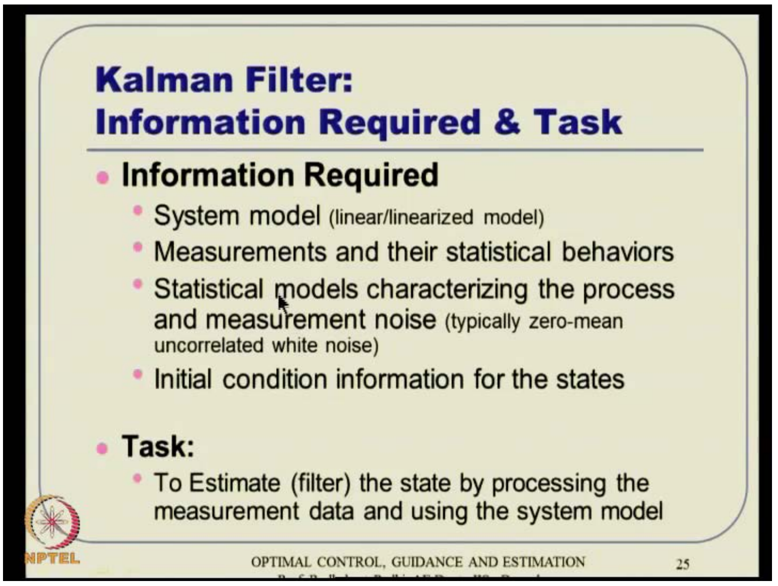


So, the all these information I have taken from Wikipedia or other. Then, any of you are interested in like this part of these has been taken from this website and then, this is University of North Carolina and that is where many things are well documented for Kalman filter actually. Anyway, about his pioneering work for various things, he has got several very good medals.

First thing is even though his paper was not published in IEEE and subsequently, he has never published in IEEE also. Finally, IEEE gave him the medal of honour in 74 and that is IEEE's highest honor and he also got Centennial medal in 84 where IEEE completed its 100 year. It is a kind of honor to few people and Kalman was one of that and then, there is a Kyoto prize from Japanese noble prize. It is a kind of regarded that way. He has got that also. Then, this American Mathematical Society something called Steele prize, the highest prize in that society, he got in 87. Then, Richard E. Bellman control heritage award which American Automatic Control Council gives. That also he has got in 97. Then, he has got some Charles Stark Draper prize in mathematics, I mean he got from National Academy of Engineers and that is somewhat considered as equivalent noble prize that must have monetary value as well. Finally, the National Medal of Science from 2000 or I mean in 2009 from US that is typically given by US president actually. So, he got several several very well credit on his shoulder and primarily because of his prize winning contributions and out of that the Kalman filtering turns out to be most well recognized thing actually.

Now, I just did a Google search today morning. So, just the day of recording of this lecture today and if I just do Kalman filtering that about comes something like 20,90,000 results. If I do for professor R.E. Kalman, it turns out to be something like 8,75,000 results actually. You can see the impact worldwide.

(Refer Slide Time: 30:34)

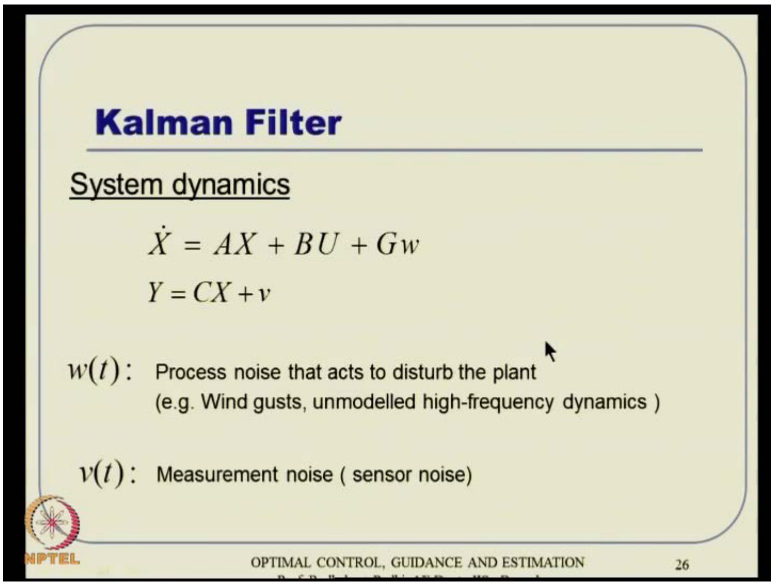


Application domains include aerospace, electrical, chemical, mechanical image processing robotics and virtually every field actually. So, that is why, it is is recognized so much actually and rich source can be found in this repository which is inversely of North Carolina as a good repository about Kalman filtering. There you can also download his first fundamental basic paper actually. All right.

So, let us move on with what is scientific content here. So, Kalman filter are the information required and task is something like this. So, what is required for for operating Kalman filter is something like this. It needs to have a system model. We should have a linear model or other linearised model and then, you should have measurements and their statistical behaviors. We should have statistical models characterizing the process and measurement noise and typically, they are considered as zero-mean uncorrelated white noise. What is that? I will talk about that in subsequent lectures. Then, we got initial condition information for the states are also available. That is what we are seen here. That means, these are essentially the required information and using this information, the task is to estimate or filter the state by processing the measurement data and using the system model.

Regard to measurement data, various statistics about that, we also have process model, typically a linear model. So, using that you are able to I mean the task is to estimate the states by processing this processing a kind of using certain formulas or sequence of formulas and things like that, so that you get estimated state information actually. Again, without any derivations and all, let us see how do we kind of implement it, how do you, what is the basic idea here and what the final result actually, ok.

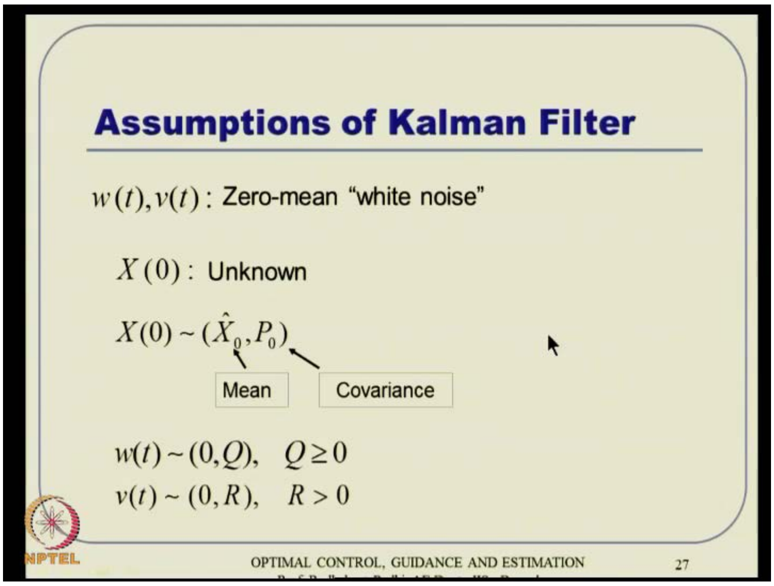
(Refer Slide Time: 32:52)



So, the system dynamics here we are talking about AX dot is AX plus BU plus G times w. That means, this is like a noise influence matrix sort of thing. This is control influence matrix B, but this is noise influence matrix. Remember process noise is a physical noise which is coming and acting on the system dynamics. So, that is the w of t which is process noise that acts on the system to disturb the plant actually, ok.

In practice is some aircraft is going, this some gusts noise from atmosphere that is noise on this noise input. If you are driving on the road, the the pot holes, the pot holes on the road, the road is not very smooth. So, that also is a kind of a noise actually. So, similar things are there for distress the noise, but that is an impact of the system and that impact is realized through this this noise influence matrix G basically, ok. All right. So, that is what w of t is. It is a process noise that acts on the system to disturb the plant and v of t is a measurement noise and typically, it is a sensor noise, Y is CX plus V. So, this is the model now, where w and v are are noise quantities actually.

(Refer Slide Time: 34:15)



Now, what you are assuming is w and v has to be 0, mean white noise again. What it means is white and all that. This talks about auto-correlation function being delta basically. That is what auto-correlation is a delta function and I will talk about that in in next class in a little more detail, but essentially it turns out that if you multiply w of t with w of w of t 1 and w of t 2, it turns out to be 0 and v of t 1 into v of t 2 also turns out to be 0 unless t 1 equal to t 2. So, if you talk about the same time instant, then this auto-correlation has some value. Otherwise, not actually that for white noise.

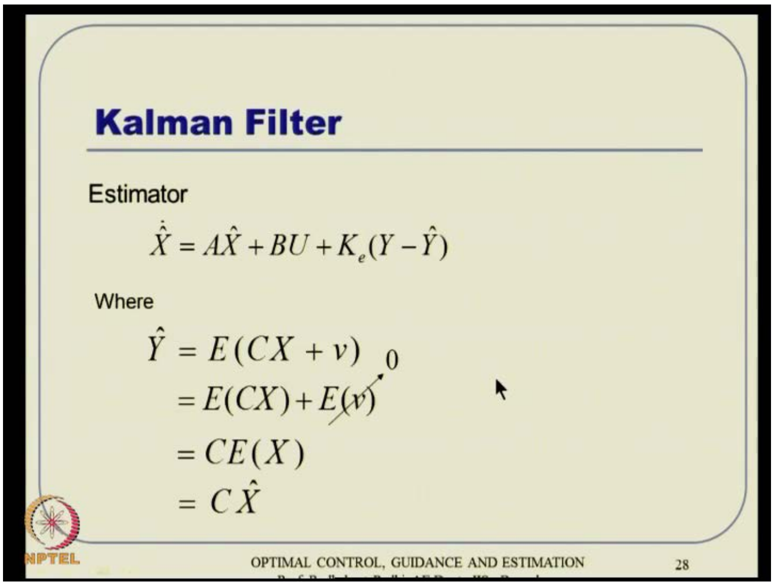
We also know some initial condition of that is unknown. However, we know some sort of a mean value of the initial condition and some sort of a co-variance matrix of the initial condition. What is P naught is if you talk about, well X tilde which is X minus X hat if you defined that way, then P is nothing, but an expected value of X tilde times X tilde transpose. Remember this is an out of product X tilde times X tilde transpose. That is a matrix actually. By definition something like this. So, initially, initial conditions means you put x 0 here instead of X and you will let let becomes X 0 tilde times X 0 tilde transpose.

So, that is a process noise co-variance matrix initial condition. That is kind of we are assuming that we know a value. In other words, while recognizing this, we need to put some values for this actually and w of t are characterizing something like this. W of t is 0, Q. That means, 0 mean the co-variance is Q basically. So, in other words, Q it turns out to be something like expected value of w times w transpose actually, ok. Similarly, R is also expected value of V times V transpose. These are all out of orders actually. So, this is w and v are assumed to be 0 mean and they have this Q and R are their co-variance matrices respectively.

Q is assumed to be positive semi-definite and R is assumed to be positive definite actually and also, this has a little bit odd factor. Remember, R cannot to be a positive semi-definite matrix, which means whether the sensor is noisy or not, we are assuming that there is certain amount of noise in the sensor output, ok. So, that is fundamental fact of gamma field triangle. If you are very sure that the sensor output solve really noise free or that noise is something very low and do not want to determine kind of account for and all, then you probably go back to the observer ideas instead of relying on the filtering ideas. Filter does not normally degrade to observer automatically. That is somewhat difficult actually, ok.

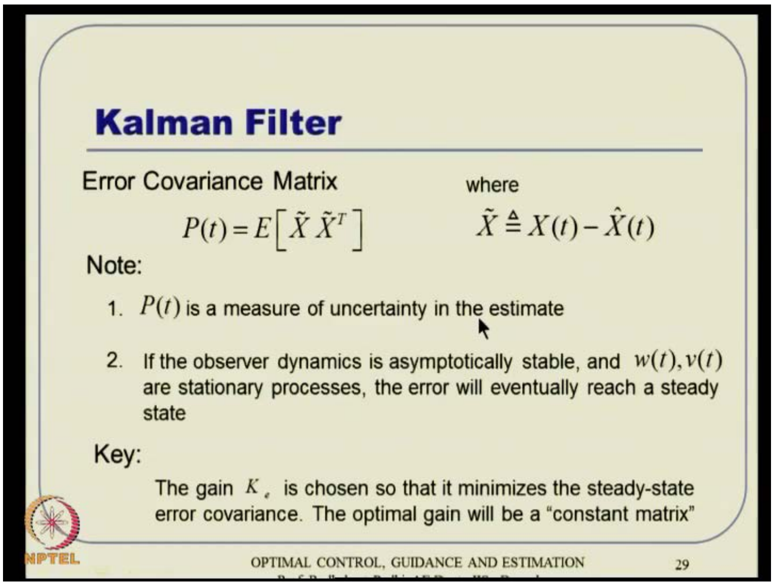
Anyway, coming back, we are assuming the same estimation or same observer dynamics X dot is AX plus BU plus k times Y minus Y hat, where Y hat is nothing, but expected value of CX plus V. Then, if an expected value being on A, we need linear operation, you can separate it all that way expected value of CX plus expected value of V, but expected value of V is remember, it is a 0 mean noise that is what we are assuming.

(Refer Slide Time: 37:17)



So, this is 0 and C is again expected value, the linear operator. So, C can come out and this turns out to be CX hat. So, Y hat is nothing, but CX hat. Even though definition tells that is expected value of CX plus V. So, you substituted back and present for that basically. That is what the message is here, ok. Now, what happens is this error co-variance matrix P of P is expected value of X tilde times X transpose there is like that one at before that.

(Refer Slide Time: 38:04)



We also assume that P of t is a measure of uncertainty in the estimate. I mean why assume, it is the definition actually, ok. So, if you talk about X tilde times X tilde transpose and it has the expected value of that; that means, it is externally nothing, but a major of uncertainty and the estimated states, ok.

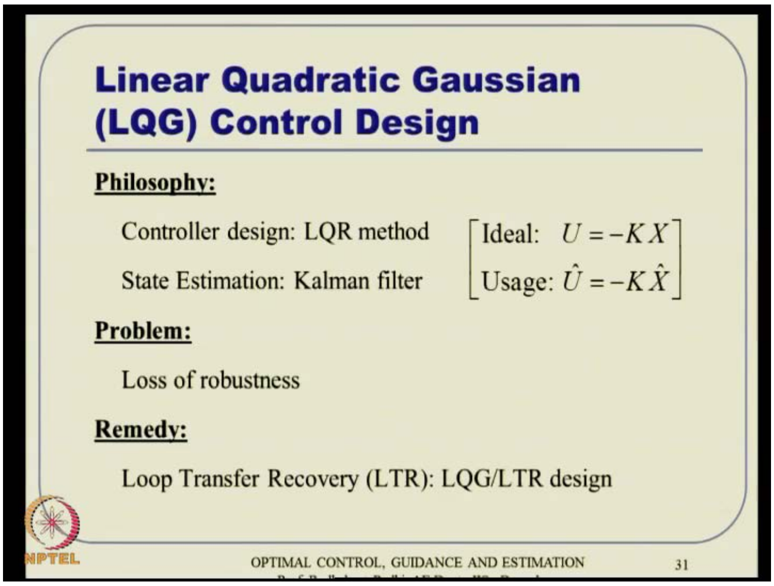
If the observer dynamic is asymptotically stable and this case, this noise vectors double of t and v of t are stationary processes, that is their characteristics do not change, the correct the magnitude keep on, I mean the number of where w of t and v of t keeps on changing, but there probability density function or probably distributive function does not change with time PDF actually. There that do not change with time actually. That cost stationary processes. That means, the characteristic noise itself does not change. The value can change basically. So, in other words, if the observer dynamic is asymptotically stable and this w of t and v of t are stationary process, then eventually the error will reach a steady state value, ok.

So, that is what the observation is all about actually. Now, it turns out that this gain K e is chosen, so that it minimizes that steady state error co-variance actually, ok. So, you have to leave some sort of steady value of error co-variance, but the idea here is how do I design K e, such that it minimizes the steady state error co-variance thing actually and optimal gain will be some sort of a constant matrix actually, ok.

Now, how do you design and all, I am not going to talk here, but the procedure mechanization of that is something similar. So, we have got initially this Kalman gain is some you compute that way, where the P matrix is computed from this algebraic Riccati equation or Riccati equation. Then, you operate this observer dynamics this way. Now, there you can see, if you go back to this, let me go back to that. If you see this matrix, there is no G here. So, here the filter it will turn out to be G times Q times G transpose. You can see that very quickly here G times Q times G transpose, ok. So, if G happens to be identity sort of thing, then essentially it is nothing, but same LQ observer.

Now, the thing is in general, it gives us little more freedom compared to just as a simply and LQ observer, ok. So, estimated dynamic turns out to be what is given here. X hat is AX hat plus BU plus K e times Y minus CX hat. This factor is innovation term, Y is the actual sensor output minus, CX hat is the predicted system output actually, all right.

(Refer Slide Time: 41:33)



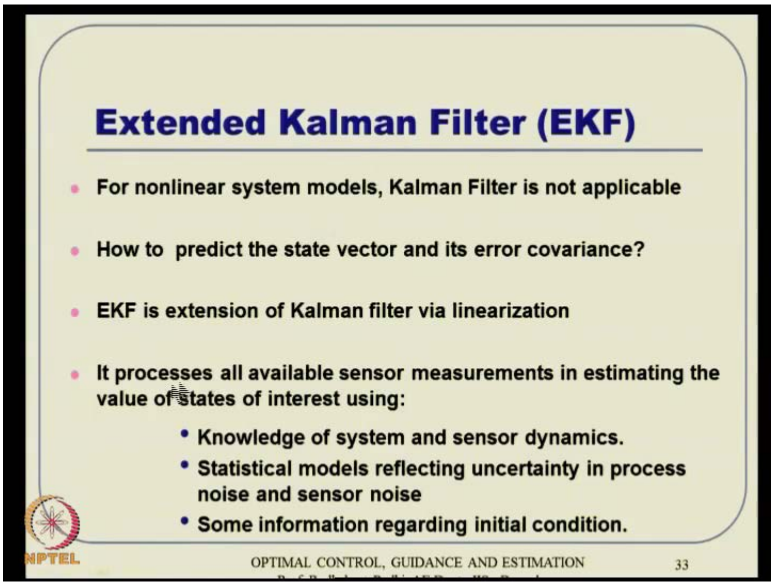
So, what is the concept of LQG design then? This turns out that ok you apply LQR method of control design, but X is not available. So, what you do is, I will estimate the state and that let me call that is X hat. So, now, the control V U hat equal to minus K times X hat really. This will be always available. The problem happens to be in LQ, this is actually called LQG method linear quadratic Gaussian, where the control turns to be minus K times X hat and X hat is designed from the filter point of view and K is designed from control point of view basically.

Problem turns out to be loss of robustness. The regions represent principle by the way which which tells us that it does not matter. It can be actually done independently and simply you can, in other words, the control and filter design can be done independently and you can simply operate that way. Nothing bad will happen that way. The problem happens to be a loss of robustness and remedy for that happens to be something like LQG/ LTR designs, loop linear quadratic Gaussian with loop transfer recovery, but I will not talk too much on that. Anyone interested can read some appropriate book and some more is probably one of the good books for this particular region actually, all right.

Now, moving on. What is EKF then? That is fundamental thing the people keep on using and when somebody tells I am using Kalman filter and they does not tell you what kind of filter he is using, what form of Kalman filter he is using, it typically by default it means external Kalman filter actually, ok and the whole idea here is nothing, but the

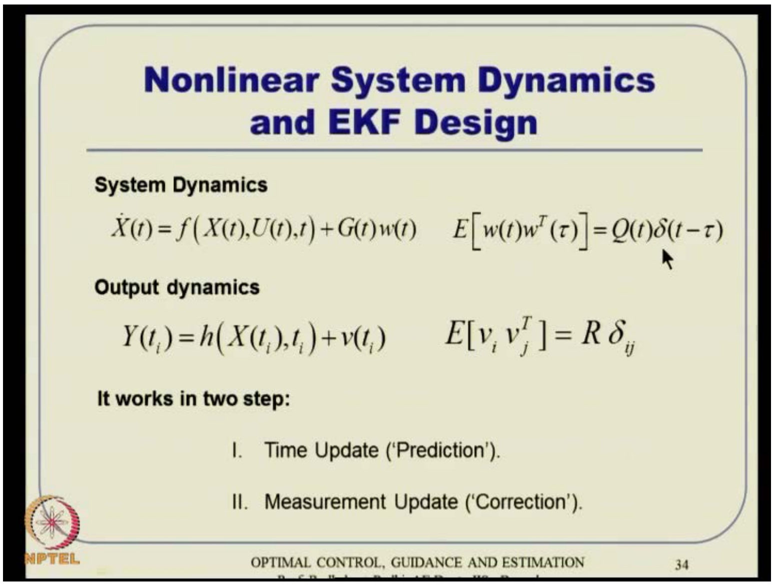
Kalman filter, but remember Kalman filter assumes and linear model and all. So, this actually is a linear estimated, where the linearised system dynamics is obtained about the most current updated value of the estimated state. That becomes here something like operating point and from which we will be able to extract this, the system matrices actually. Motivation for that for non-linear system models, Kalman filter is certainly not applicable.

(Refer Slide Time: 43:38)



So, we want more than that and then, the question is how to predict the state vector and its error co-variance and EKF is some sort of an extension of Kalman filter via linearization. It proposes all variable, all available sensor measurements in estimating the values of states of interest and it uses these three things. It uses the knowledge of the system and sensor dynamics, it uses the statistical models reflecting the uncertainty of the process noise and sensor noise and also, it uses some information regarding initial conditions, ok.

(Refer Slide Time: 44:28)

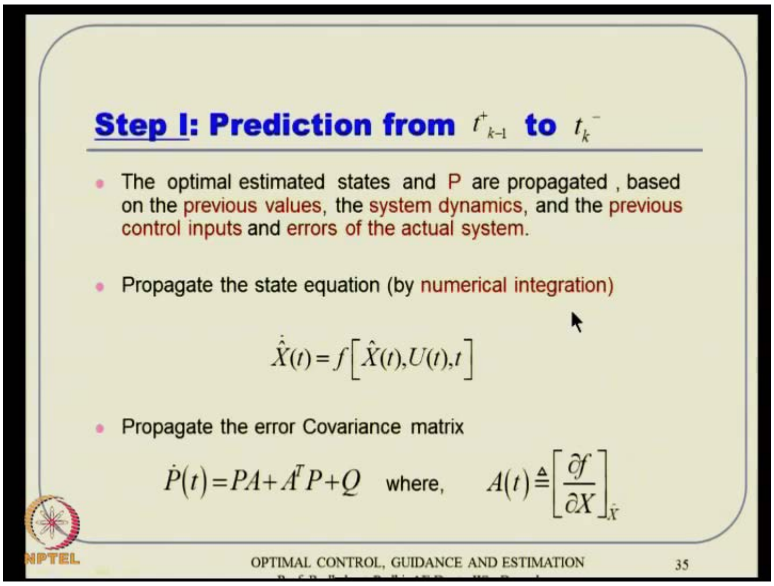


So, using these things, it will be able to compute what is an estimated state. Now, the (()) turns out to be a something like this, X dot is now f of X U plus G times w t dot A X plus B U. It is in general knowledge actually. Now, put also something similar and typically E cap is various fonts, but what is most commonly used is continuous discrete form. In other words, system dynamics is continuous, but the output dynamic is restricted actually. Again we assume this zero mean white noise sort of things, so this happens to be something like this and this happens to be something like this. Remember, these notations are very quick and gives us a mathematically platform to do something actually. So, delta is, this is a direct delta function and this is a tonic current of function actually.

In other words, if phi equals to j, the delta is 1. Otherwise, i not equal j, then a 0 basically. Similarly, if t equal to if 4 equal to t, then its value is 1 and if it is not there, anything different the value 0 actually, ok. All right. So, essentially we are talking about is continuous discrete form of EKF and where the system dynamics is continuous with noise in foot of course, the output dynamics is discrete with its sensor noise, I mean sensor noise in that basically.

So, it works in two steps. First thing is time update. That means, it talks about prediction and then correction, ok. So, whatever is the current estimate, you predict for what will happen in the future, but whenever the data comes in that point of time, it will be time for correction and hence, you do some sort of corrective steps actually, ok.

(Refer Slide Time: 46:11)

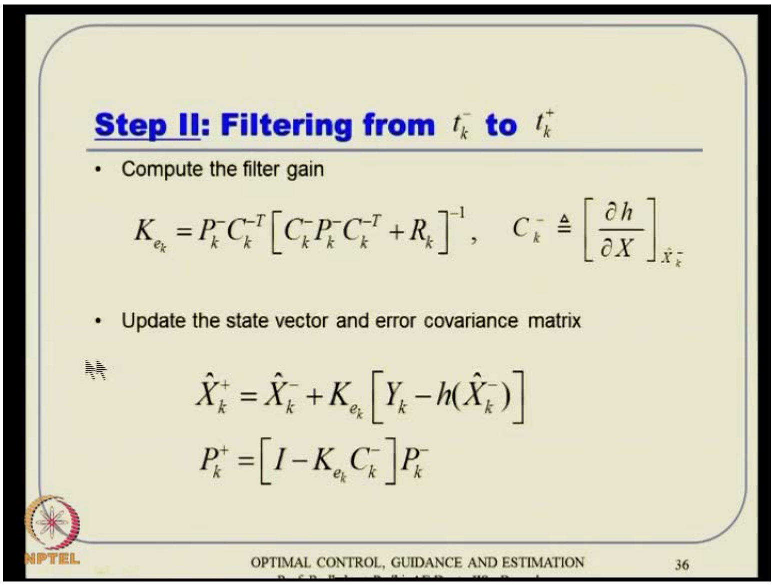


So, how do you predict? From t k minus 1 plus to t k minus is what is t k and plus minus and all that pictorially is something like this. Suppose, you have A t and this is the t k, this is the t k plus 1 t k minus 1 and this is t k and this is t k plus 1 let us say, ok. You got something to predict it here. So, this is called t k minus, this is t k minus 1 plus. Then, we update here from A to there somewhere and then, you talk ok predict actually.

So, this is nothing, but t k plus and this is nothing, but t k minus t k plus 1 minus. Then, again it will be updated and then again, it will go like that actually, ok. So, this is the notation here. So, how do you do the prediction from t k minus 1 plus to t k minus, this path what you are saying here? How do you do that actually? So, that is done by simply propagating the state equation without noise.

So, you construct a state equation without noise and simply, if you know this value, you can propagate to that. Then, along with that you also propagate the error covariance matrix P of t in this way. So, where it will come and all will discuss later, but then you use this equation P dot of t is nothing, but PA plus A transpose plus Q, where A t is given defined as del f by del X about X hat. That is more important.

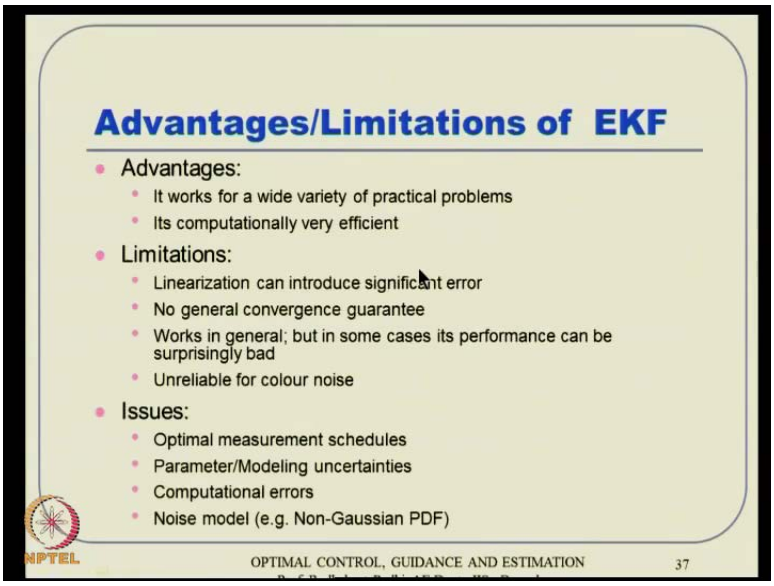
(Refer Slide Time: 48:00)



So, linearization is carried out at the most estimated current value of state. That is what is A t. That what it means pi of t basically, alright. Now, compute the filter gains. That means, K e k is something like P k minus C k and things like that. This is again how it comes and all we will talk later, but once you get T k minus, it is time to update to T k plus also. So, that T k that is on that way. K e k you compute this, this entire expression, where C k minus is computed by that actually because we have already come here. So, whatever see is C k minus is there, you compute in that expression.

Then, you update the state and co-state and error covariance matrix this way. So, X at k plus turns out to be something like this, where P k plus turns out to be something like this actually and these are different expressions for that basically, all right. So, advantage is first of all it works for a wide variety of practical problems and it is computationally very efficient, but limitations are also there and some of them are something like linearization.

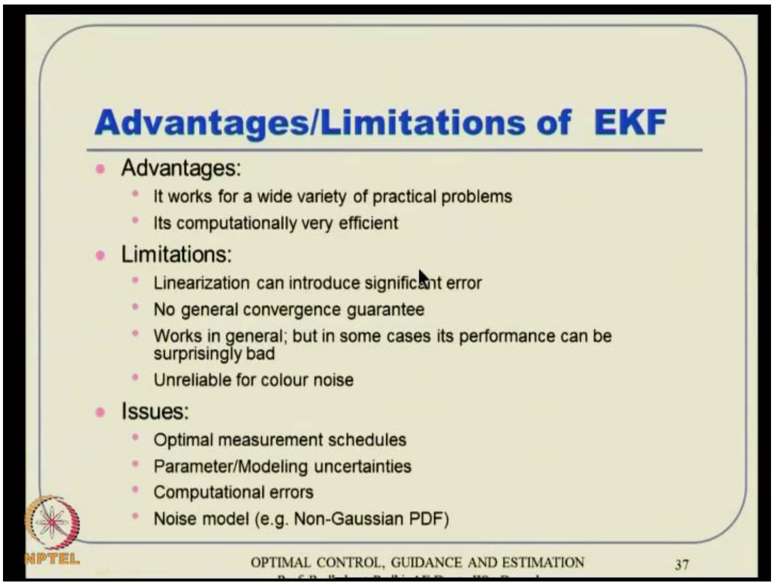
(Refer Slide Time: 48:48)



Linearisation can introduce significant error and hence it cannot work very well. No general convergence guarantee. I mean it works we know, but convergence guarantee and theoretical proof is probably not there and hence, it works in general, but in some cases, the performances can be very bad actually. So, you need to be slightly careful about that and it also turns out to be a kind of unreliable for colour noise, where colour noise is something that is not really white.

In other words, the self correlation while the self correlation is not, I am sorry auto-correlation is not a delta function. Then, what actually? Then, there are issues something like this. How do you kind of even if somewhat EKF works like this, so how do you optimally place the issues like these? So, how do you optimally place the measurement schedule basically? So, how do you measure it? What is the optimal sequence actually? That we do not know. That is the tuning it requires.

(Refer Slide Time: 50:25)



Then, parameter or modeling uncertainty. If they, I mean you are relying on some sort of a model to predict the behavior, but if the model itself is wrong, then what you do actually. Then, there are issues of computational errors. Then, there are issues of a noise model. In other words, if the noise model turns out to be non-Gaussian PDF, then what do you do actually? These are some of the issues. Because of this, there is a need for going beyond EKF and here we talk about the need being non-linear systems. Typically systems are non-linear. We have got this non-Gaussian noise are inputs. We got this correlated noise also and if the uncorrelated, what we talk is self correlation and all or what are auto-correlation. That is something also called cross correlation. Again we will talk about in the subsequent lectures.

So, cross correlation means, I mean if I take X 1 of t 1 and X 2 of t 1, again the same time, but two different values, then if I take multiplication of that, it also turns out to be 0 actually, but in general, it is not so. One has an impact on others and things like that. So, in that particular case attribute this is called uncorrelated noise and typically, there are many filters which have been proposed beyond EKF. So, the very first thing that comes to mind is something like a linear equation EKF, linear equation Kalman filter or something very popularly known as UKF, Uncentered Kalman filters actually. So, then people talk about the infinity filters actually. Then, there are a radius of practical filter and things like that actually and characteristics of such filters are typically are often approximate and the sacrifices theoretical accuracy in favor of particular constraints and considerations like something like robustness, adaptation, numerical feasibility. All sort of things we will talk about, but then you have to sacrifice theoretical accuracy will be there.

Essentially, the attempts are there to cover the limitations of EKF. That is the fundamental idea actually, but here in this particular course, we will not talk anything beyond EKF. We will see whether EKF can be kind of discussed here, but essentially we talk about EKF in general. Then, we will see whether UKF ideas can be brought in and more than that, we will not be able to discuss and not necessary also. Many times UKF or maximum UKF will do the job and remember particular filtering and all that nice. Many things you can do that way, but your computational demand turns out to be much more, then UKF actually. UKF and EKF is not too much out of the magnitude difference. This will be a little bit more, but not in the order of different. However, the practical filters and all will demand a lot of computational time actually, where essentially all of them try to kind of cover the limitations of EKF. So, that is what it is, all right.

I think this one I want to talk about. In the next lectures onwards, we will see various derivations, theoretical foundations and all for Kalman filtering. Thank you.