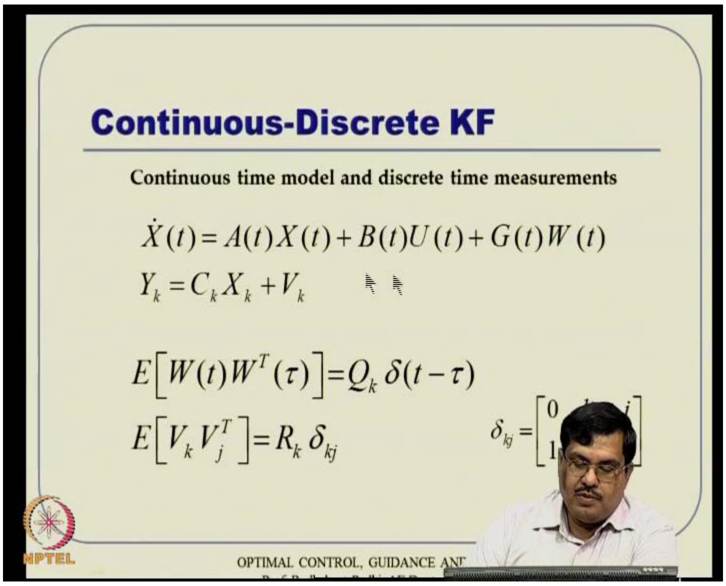
Optimal Control Guidance and Estimation
Prof. Radhakant Padhi
Department of Aerospace Engineering
Indian Institute of Science, Bangalore

Module No. # 12
Lecture No. # 30
Kalman Filter Design - III

Hello everybody. We will continue our lecture on Kalman filter and logically, we have derived in the linear domain first followed with discrete time as well and logically, we will extend this to something called extended Kalman filter and then, followed with little bit concept on unscented Kalman filter which are typically used these days in particular application.

So, the problem is something like this. First is continuous, ok before that a little bit recapitulation of what we call is continuous discrete Kalman filter is still in the linear domain, but system dynamics can be continuous, where as measurement is discrete and that is the one we shall actually, this platform is the one which will help us for external Kalman filter and all that.

(Refer Slide Time: 01:08)

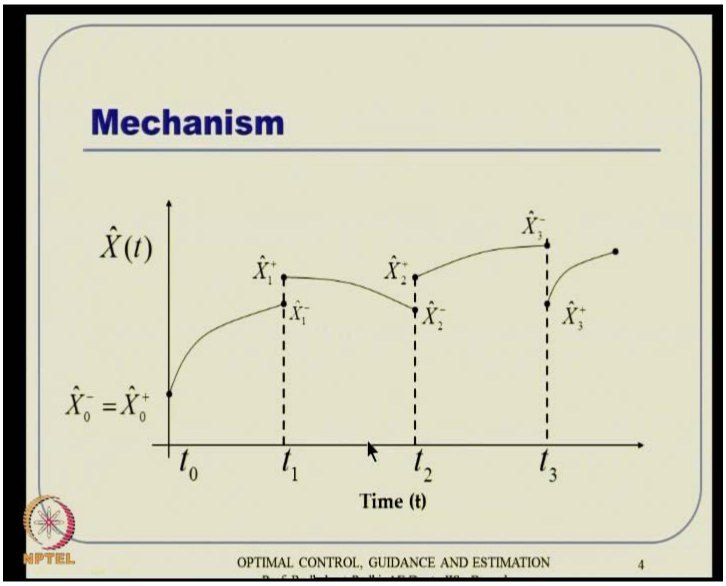


So, the problem is like this now. There is a continuous time linear system time varying of course, where is I mean system dynamics is continuous time, whereas the measurement equation is discrete time, ok.

Now, there is a kind of a mix situation, but that is typically real life. In real life, we have system dynamics represented by differential equations naturally, whereas, measurement equations are typically discrete equations where it comes out what it is sample of, I mean sample of measurement sort of thing, ok. So, here is W which is actually continuous time. So, it is represented as this expression, where as Q is t minus tau. Where is this? Essentially, well, we can think of this is something. Well, anyway we will not bother about that. We will bother about Q k is when t goes t is t k, otherwise Q k is moving actually.

Having said that, if t is equal to tau, then only it says 1, otherwise it is 0 basically. Remember that. Then, expected value of this quantity V k times V j transpose is R k chronicle delta now. So, unless this k is equal to j everywhere else, I mean unless k is equal to j, it is 0. If k equal to j, it is 1 actually.

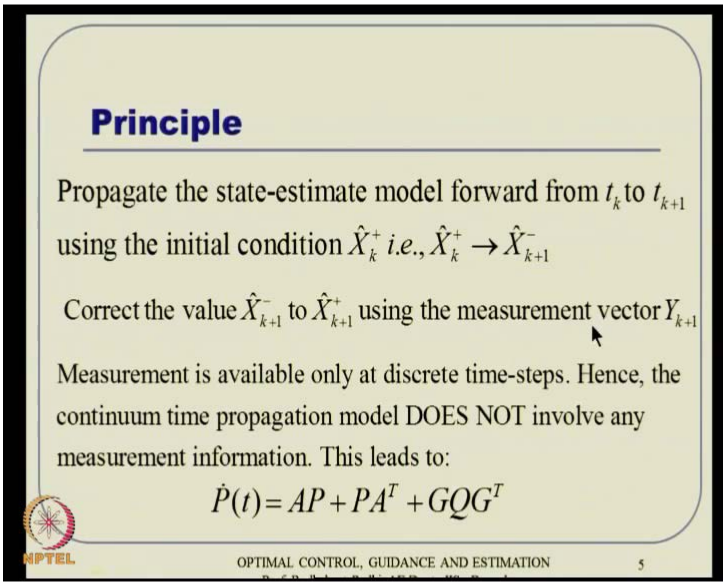
(Refer Slide Time: 02:37)



So, using this kind of notation, we will go to the mechanism. The idea is j again this prediction-correction, prediction-correction like that actually. So, start with some value at t naught minus value, which will also assume that is the corrected value. That is the initial guess sort of thing. Then, we propagate it and get a measurement here. So, update it and by the way, if you get measurement at t naught itself, then you can update it also here itself and then proceed.

So, again as coming back, this is the propagation. Then, we predicted based on the measurement this correction happens here and then, again we predict and based on measurement, we have to update here and think like that actually.

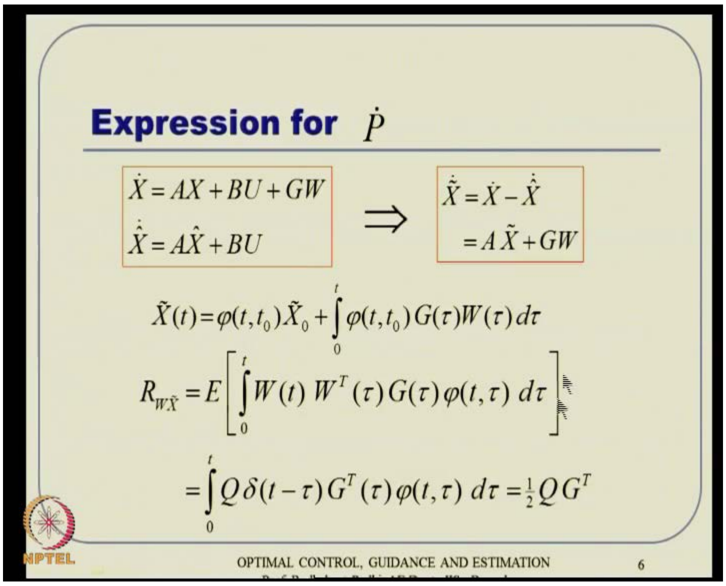
(Refer Slide Time: 03:17)



So, principle is like this. Propagate the state-estimate model forward from t k to t k plus 1 using the initial condition X k hat plus and then, correct the value from minus to plus hat k plus 1 using the measurement vector Y k plus 1, ok.

Now, the measurement is available at discrete time steps only and hence, the continuous time propagation does not involve any measurement information actually. That is you can derive it also and it turns out that earlier there was some, I mean it had some measurement information as well actually, if you remember the previous lecture and all that, but now it does not actually. So, P dot takes a rather relatively simpler expression like this, right in this form actually.

(Refer Slide Time: 04:13)

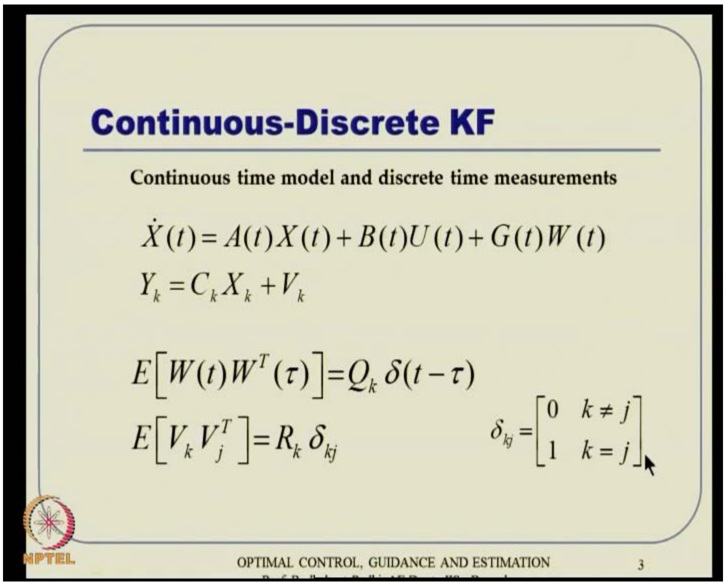


So, we have this now the expression for P dot, something what we need to derive and let us say, we have this X dot is something like this and X hat dot is this form because noise cannot be taken into propagation equations. So, we drop the noise term while operating on the prediction mode. So, obviously, the error between that is nothing, but this X dot minus X hat dot which turns out. If you take the difference between that, BA BU will go, the rest of the terms will become something like A times X minus X hat, which is XA times X tilde plus G times W, something like this.

So, this is the time varying input. So, time varying input we can solve this X hat of t using this. What you remember these are all time varying system. So, we cannot use exponential term, you can always use state transition matrix. So, transition matrix phi t naught you put it back here and then, get a solution for that. Then, RWX tilde turns out to be expected value of this expression, right. It is AW times X tilde. X tilde is like this. So, put it there, but again because our orthogonality, this expression does not, this term does not matter, only this term will matter and then, it will excite this again delta function and things like that. Ultimately, it will turn out be something like this only. Here is the delta function which will come into place when expected operator goes inside the integral.

So, this is nothing, but by definition Q. So, it turns out to be QG transpose sort of things actually. Now, going back to this actually, strictly speaking this there is, so that is no Q there. I mean Q k does not make too much sense because continuous time is evaluated at time t k than is Q k.

(Refer Slide Time: 05:58)

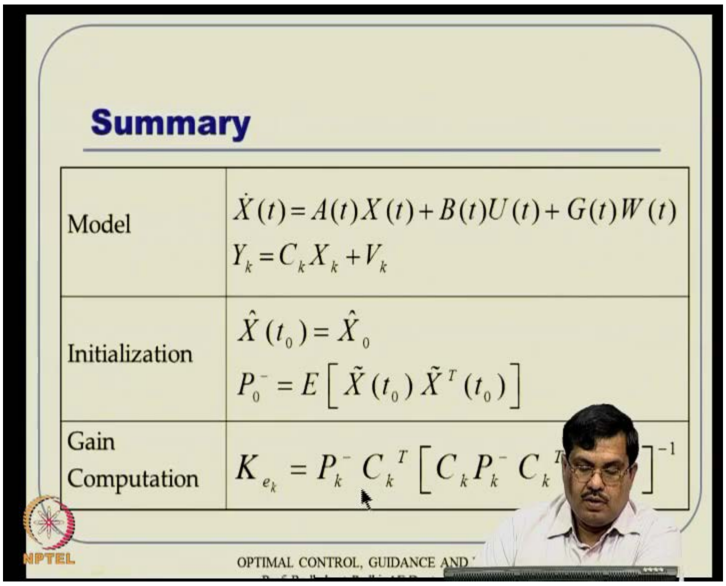


So, this is where it is actually. So, all right. Now, expression for P dot is like this by definition. So, again you separate it out these two terms and this term plus the same term transpose again. So, this term happens to be something like this. So, expected value of all that and then, expect AE, expected value X tilde. X tilde transpose is something like this plus GE W times X actually.

The expected value will come all the way, hence here actually. W is a random variable we know. That is why it will get coupled with that, but this quantity is nothing, but P by definition. So, A times P plus this quantity is nothing, but we just derived it is half of QG transpose. So, put that there and it will land up with this. So, P dot happens to be this term plus the same term transpose. So, that is why you get AP plus PA is a symmetric matrix. So, P transpose A transpose P times A transpose here, ok.

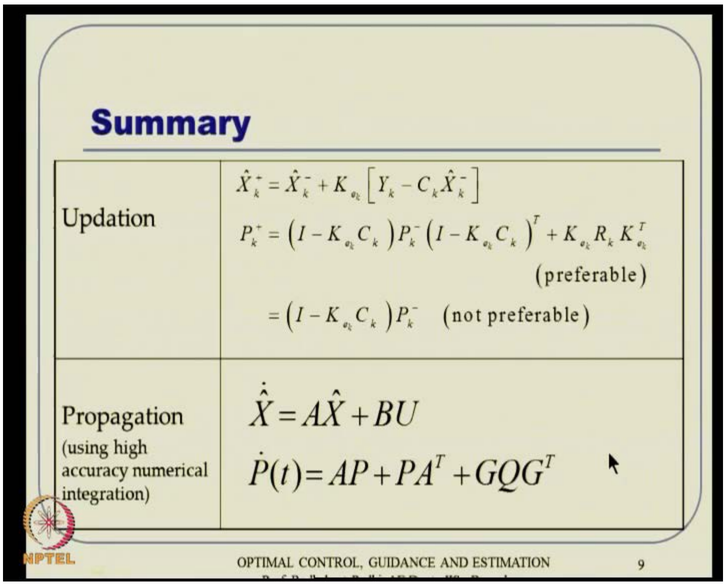
Once you expand this transpose, it will become P transpose A transpose. P transpose is same as P. So, that is P and the transpose is here. So, AP plus PA transpose plus this quantity will remain exactly same, but half will go now. So, that is what will happen, thing like this. That is what we told. I mean this P dot takes the expression of like this actually.

(Refer Slide Time: 07:49)



So, we got an expression for P dot now basically. So, that is all we probably needed to implement that actually. So, how do you implement really? We have this system dynamics here and we have this measurement equation in discrete time. We initialize it again and P naught minus also need to be initialized. Then, computation can happen this way something as before, ok.

(Refer Slide Time: 08:09)

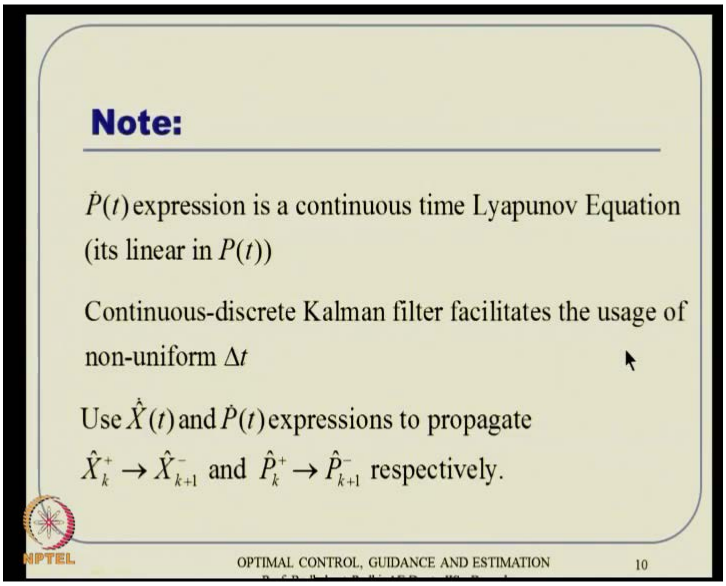


So, this update can also happen as something as before. We do not have to derive again. It will derive everything. These are very similar to what we have done in the discrete time system, I mean discrete time platform. Only thing is the update equation, sorry the prediction equation, the propagation of equation sort of thing can happen now using some sort of higher accuracy numerical integration of high accuracy basically. In other words, this can be retained as something like continuous time expression and then, using all order numerical integration scheme, you can actually propagate it with much lesser error actually. That is the only difference, but when you do this P dot expression, be careful. P dot expression like, LTI system do not, I mean like continuous time as system do not use it.

This additional term will not be there. It stops here actually. P dot is PA plus, sorry AP plus PA transpose plus GQG transpose. That one more additional term what we had in, I mean pure continuous time derivation process, where measurement was also seemed to be like purely continuous variable is not there basically, ok. So, just be aware of that. So, otherwise the implementation is fairly similar to what we do, very similar to what we do in discrete time domain. The only difference is probably in the propagation stage, you use continuous time expression and then, excite some sort of higher order numerical integration thing to propagate it in much better sense basically. That is how this platform.

Now, going back to this idea for this prediction-correction, prediction-correction thing like that, we can now graduate our self centre. Well, we know something, so that we can actually think about handling knowledge on system as well. Basically let us see in a short while now.

(Refer Slide Time: 09:53)



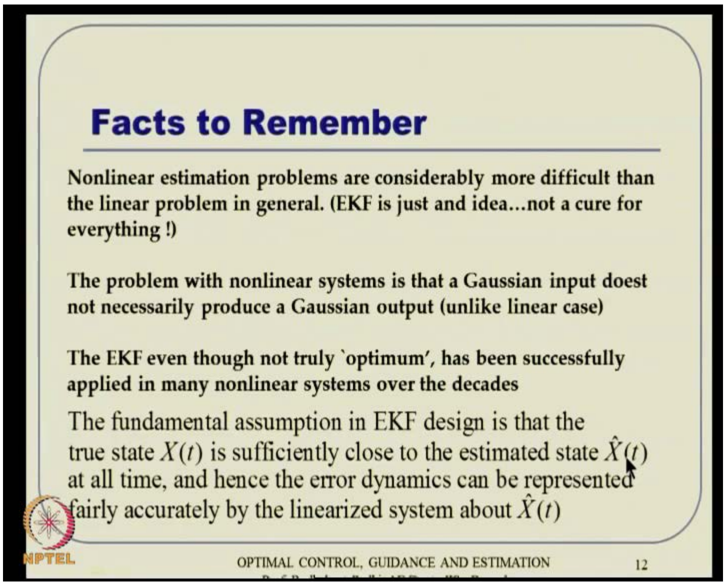
So, couple of comments here. First of all, this P dot expression is something is a continuous time Lyapunov equation now because nor linear term is no more there in terms of P basically, ok. So, it happens this P dot expression is type PP transpose and things like that. That term is no more there. So, it will turn to be something like linear differential equation in the form of P and it takes somewhat close to this expression of what is called is Lyapunov equation, ok. That is just an observation actually. Then, this continuous discrete Kalman filter facilitates the usage of non-uniform delta t also. That is another important point. We really do not have that this up this interval t naught to t 1, t 1 to t 2 and all the intervals will not be same basically.

So, prediction can go on as long as there is no measurement information and as soon as there is the measurement information or it is valid information by the way, we will talk about that what is called as out layer and all that is not there. So, if you consider that is valid information, then as long as it does not look on, we can continue to operate on prediction mode. As long as this valid information comes to the sensor, we can update it actually. So, this does not tell you that the rest to be in uniform delta t. There is no reason for doing that actually. So, that is another advantage, ok.

So, you use this continuous time expression to propagate this way. So, do not get confuse that these expressions are not there. They are still there embedded into this actually. This equation basically, starts with some mutual condition propagate to the next time state actually. So, when it propagate, essentially start is more condition at k and it propagate k plus 1 minus implies it happens actually.

So, finally, after all this we are testers. We can actually talk about E k and if you remember this, my very first comment about Kalman filter is, Kalman filter has become quite popular in the end, quite vast applications and things like that because of this platform, this external Kalman filter. Unfortunately, it works for a variety of problem actually. Even though, it is a concept and it has no rigorous proof and things like that, still it operates. I mean it works successfully for many cases actually. That is why it is quite popular. So, let us see that before you derive or I plan the concepts, some facts to remember.

(Refer Slide Time: 12:36)



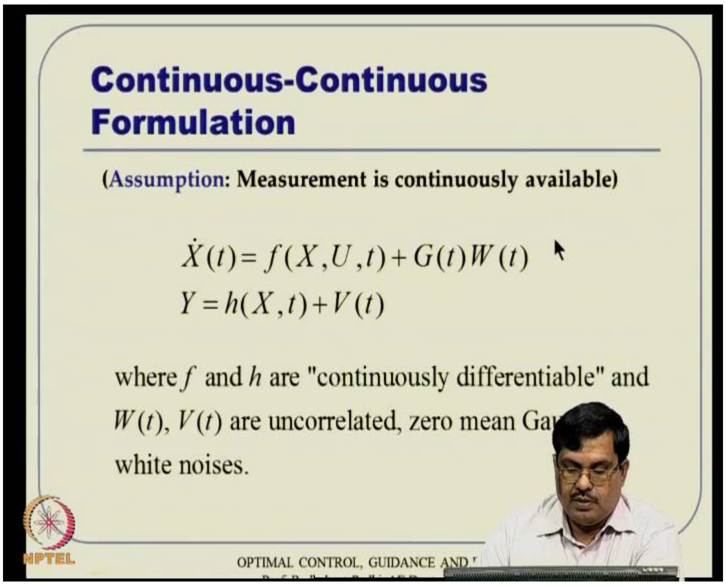
First thing is nonlinear estimation problems are considerably much more difficult than the linear problem in general. EKF is just an idea and it is really not a cure for everything. In other words, if you have a typical peculiar problem with high nonlinearity or may be this kind of non-Gaussian noise unlike linear noise things like that, then EKF may fail also. So, just be aware of that.

The problem with nonlinear or linear system is that the Gaussian input does not necessarily produce a Gaussian output. So, thus the one of the major difficulties of these nonlinear systems in general actually, ok, but I have heard the nice thing is EKF, even though is not really optimum, but it has been successfully applied in many nonlinear systems over the decades and in a variety of applications and that is what I have been insisting on actually. What are the fundamental assumptions? There is a nice idea here. It tells us that the fundamental assumptions in EKF design is that the true state X t is sufficiently close to the estimated state X hat of t at all time, ok.

So, our estimate is not really read at any time including initial guess by the way. That is the fundamental assumptions and hence, it requires that we really have rather reasonable good initial guess for X hat of t as well actually. All the time, it will assume that if my true state is close to the estimated state and hence, I can actually represent the system dynamics, some sort of linearised system dynamics around the systemated value because the error between estimated and true is not high. That is the assumption actually.

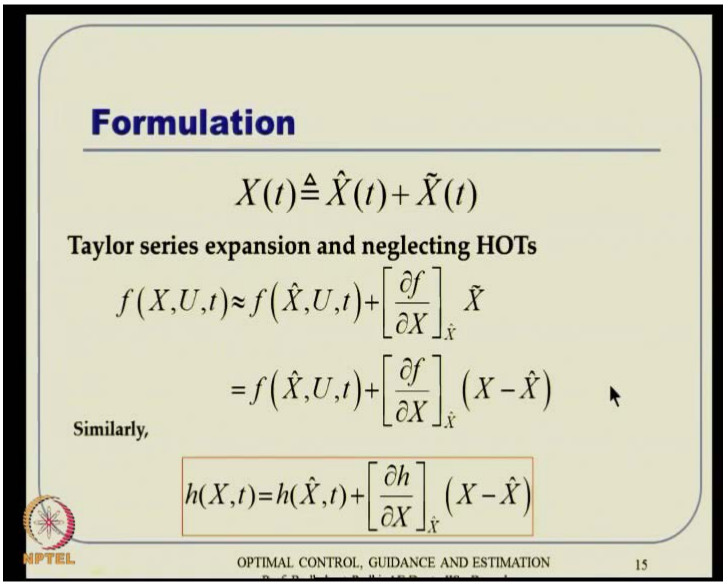
So, with all that fact in mind let us go to EKF domain and first, we will see this continuous EKF, that is everything happens in continuous domain sort of thing, where ultimately we will land up with discrete form also basically.

(Refer Slide Time: 14:48)



So, when we talk about continuous formulation, you have this problem X dot is f of XU t plus G t W t and Y is nothing, but h of X t plus V t. Everything is continuous, even measurement is assumed to be kind of a continuous variable with varying sample grade and all that actually. The other assumption is f and h are also continuously differentiable functions and as usual, W and V are assumed to be uncorrelated, zero mean Gaussian white noises actually.

(Refer Slide Time: 15:23)

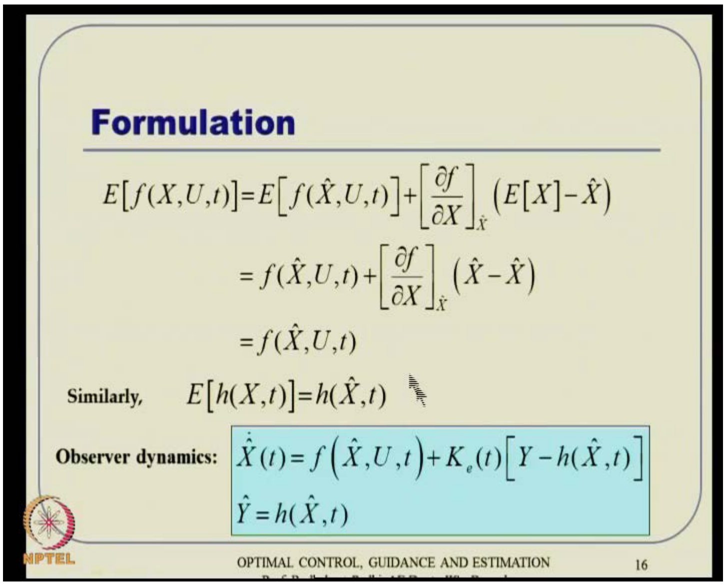


So, those standard assumptions are there with us. Now, this is the concept that facilitates EKF. What you are telling is the true state, each can be represented as something like this. Obviously, it can be represented always like this, something like true state is estimated state plus error in estimation that is by definition, but what your resuming is that X tilted is small. That is where you can bring in these linearised concepts and all that actually.

So, when you write something like this, this expression what you had in system dynamics now can be written something like this using Taylor's series. So, using Taylor's series and neglecting higher order terms, we can go back and tell that this expression I want to expand and in terms of Taylor series around X light of t. The first term will take X set. The second term will be del f by del X evaluated at X is equal to X set times X minus X set and X minus X set is X tilde basically.

So, this is where it is the first term and then, this term Jacobean matrix sort of things minus to X minus X set. Similarly, you can write this nonlinear or linear function also. If you expand using Taylor series, we can write it something like these actually.

(Refer Slide Time: 16:43)



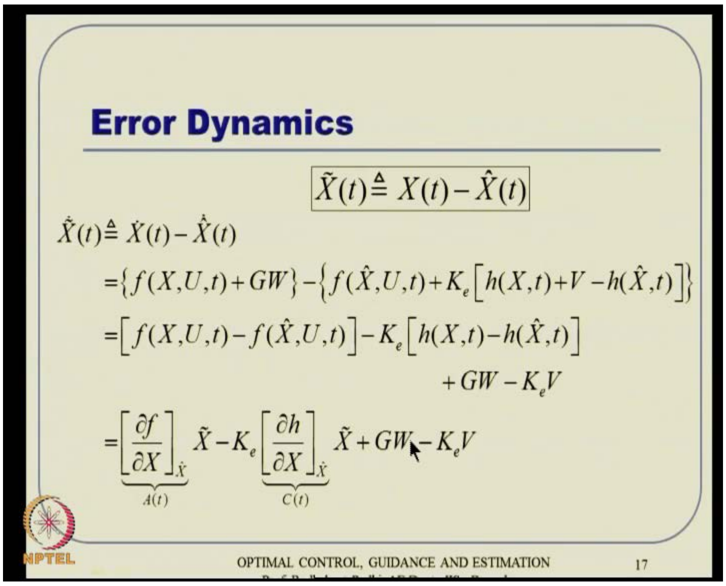
Now, what? We evaluate that expected value of this function what your derived here. If you take an expected value of both side and it turns out to be expected value of first term plus this is a constant and this kind of known numbers basically, there is no unknown variable here. So, this can be I mean, the expected evaporated can be first inside here, where this is deterministic quantity sort of thing.

Now, X E expected value of X is by definition it is X set. So, this is the nice observation actually. By definition it is nothing, but AX set. So, it all turns out that X set minus X set. So, it gets cancelled out all together basically. So, essentially what it tells you that expected value of this quantity which contains some set of renown value quantities and all is nothing, but that quantity only as long is this linear expression remains valid. I mean linearised expression remains valid up to first order term basically.

So, we take expected value of these is nothing, but the first term plus this term into Z of X as actually and thus, one beautiful observation there. So, this gives us a lot of simplicity basically. So, expected value of these is nothing, but that. Similarly, the expected value of h is nothing, but this. Exactly same thing is that again expected value of X will come here X set minus. X set will be 0 basically. So, it turns out to be like this.

So, what you can refer with this is if we go back to our observer dynamic sort of thing, we will get motivated by this observation because expected value of sense, these does not play role and it only has that part of it, so how about proposing an estimated dynamics are observed dynamics to be something like these. The first term is kept as it plus Kalman gain times innovation. Innovation is a true observation, I mean true output minus the expected output or predicted output basically and predicted output which is expected value of this is nothing, but that, I mean that observation again helps us actually putting it there. So, we have except that is nothing, but f of these plus Kalman gain times innovation which is minus h of these basically for all Y yet is nothing, but that this is predicted output sort of thing. So, this is our observed dynamics.

(Refer Slide Time: 19:10)



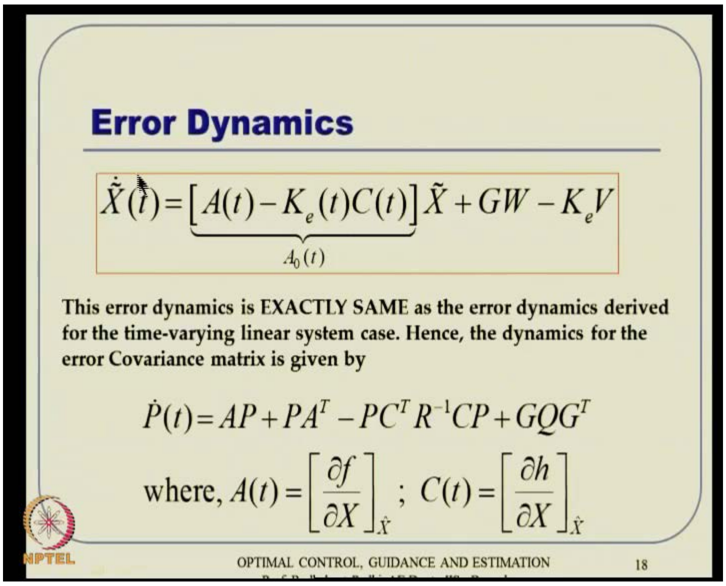
Now, what is error dynamics? Now, error is defined as X minus X set. So, X set X tilde dot is X dot minus X set dot and X dot is this quantity minus X set dot is nothing, but that quantity now. This is our observed dynamics. So, put it back here and whatever we know here and then, even first term can be put it this way and rest of the term can be kept out actually.

So, this first term and first term will combine here and then, Kalman gain terms which is one going from here and one coming from here will put them to gather plus GW minus K e V. GW coming from here and K e V, K e times V comes from here with positive sign here and ultimately, this is a negative sign here, these qualities. Now if you see this, now if you see these quantities, you go back to this minus. I mean this quantity, this minus that one. So, this minus that is nothing, but this quantity basically. So, we put it back there that quantity.

Similarly, if we talk about these two quantities is nothing, but this quantity now. So, we put it back. The total quantity what I mean is this quantity, the full and this quantity the full. So, put it back plus GW minus K e V. Now, by definition, we define this is del f by del X evaluated at X hat is nothing, but A of t. It keeps on changing because X hat keeps on changing. It is not a constant matrix, but it is time varying A of t sort of thing.

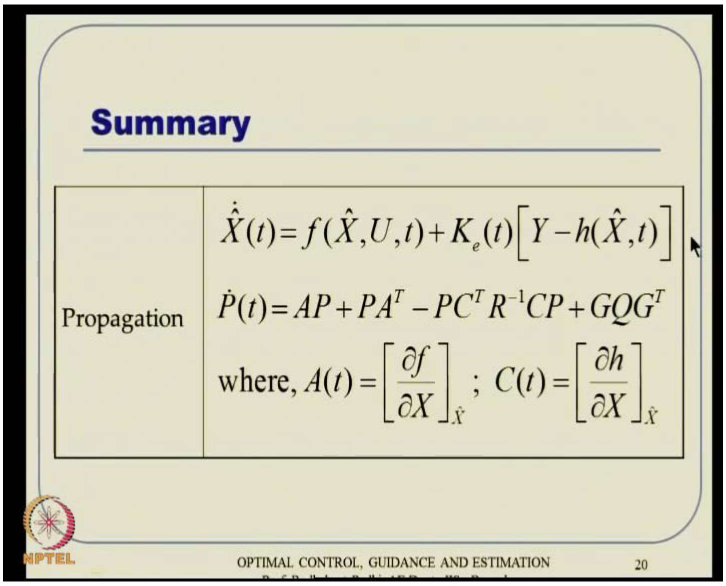
Similarly, this is nothing, but C of t. So, ultimately, what you will end up with is X hat as X tilde dot error dynamics is A minus K e C times X tilde plus GW minus K e V, where this is all time varying and this can be defined as something like A naught.

(Refer Slide Time: 20:49)



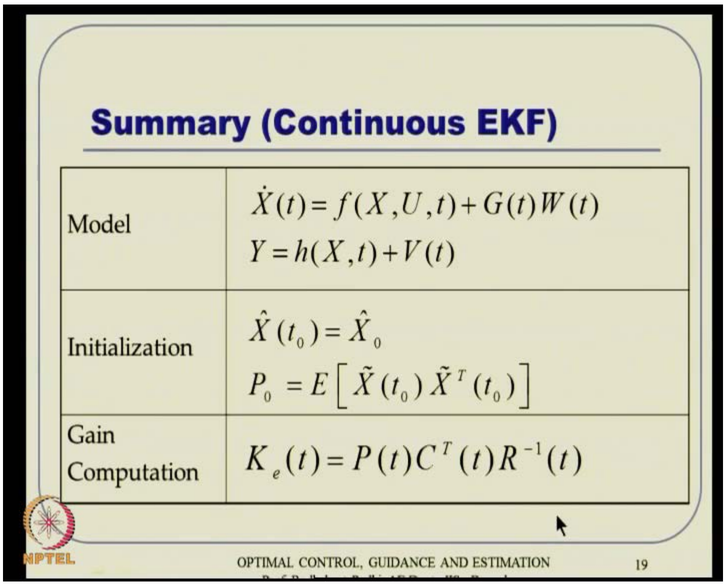
So, the beautiful observation is the error dynamics is exactly same, exactly same as the error dynamics derived for time varying linear system case. Hence, everything else remains same. We do not have to derive it again. We just simply can write it actually. So, hence, the dynamics of the co-variance matrix we can straight away write it using this linear time, I mean this time varying linear system case. We just go there, see that and write it exactly same, where the error dynamics is exactly same. So, we think there are co-variance matrix dynamics also remain exactly same as already are there.

(Refer Slide Time: 22:09)



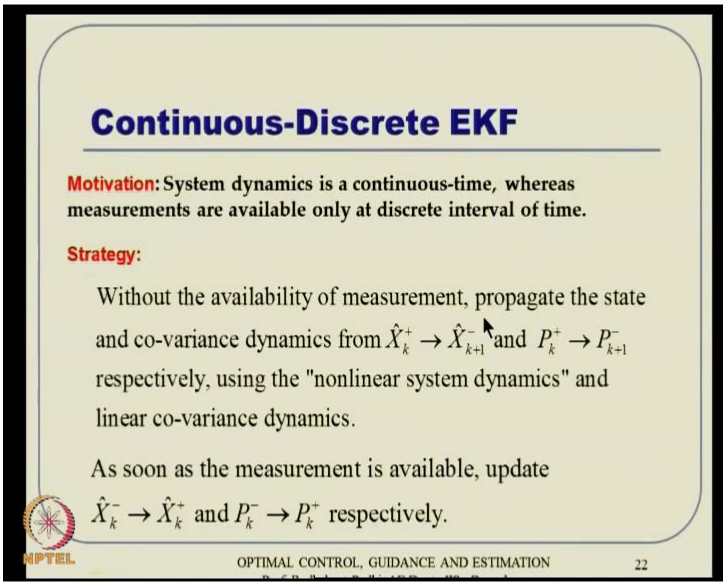
The only thing is, remember A of t and C of t are not really given to us, but they have to be evaluated every time at the most updated value of this X hat actually and that is the only difference. So, what is the summary of continuous time EKF? We have this system dynamics and our initialization has to happen first. Then, we compute the gain that way and then, we propagate it that way. You have this filter dynamics or propagation dynamics. We can use it to propagate X hat of t.

(Refer Slide Time: 21:56)



We have this co-variance matrix dynamics which can be used to propagate P of t actually. Only difference is A and C has to be evaluated every time around this X hat basically. So, that is the summary of continuous time EKF.

(Refer Slide Time: 22:34)



Now, however, continuous discrete EKF where we can mark these concepts again and here the question is, you have this system dynamics in continuous time, whereas the measurements are available only at discrete interval of time. So, that is the idea here. When there is no measurement, that means, without the availability of measurement, we can propagate the state of co-variance dynamics from XK plus to XK plus XK plus 1 minus and similarly, PK plus 2, PK plus 1 minus actually, but we can propagate now using the nonlinear system dynamics. We really gather to truncated linear dynamics and all that actually.

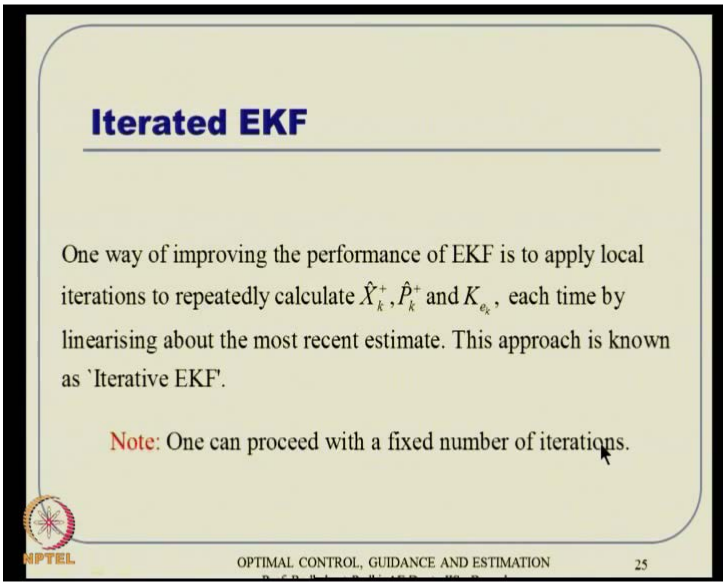
We can go back to that and tell ok, this is anyway propagation, so we can use that nonlinear system dynamics to propagate, but the co-variance dynamics, we cannot derive it that way because that is difficult. So, we can still use the linear co-variance dynamics actually, but even though it actually turns out because there is no measurement in between actually because this co-variance dynamics had this term which made it non-linear. It happens primarily because we have this continuous time measurement.

Measurement is there everywhere actually, but if it is not there, this term is not there, so we will land up with this P dot is some sort of a linear differential equation, this one plus this one plus the last one actually. That is what we will use. So, we propagate this P dot using this linear co-variance dynamics sort of thing, but as soon as the measurement is available, we will update it and we will update from minus to plus value here for the state and minus to plus value for the co-variance matrix also.

So, what is the summary again? In discrete time continuous discrete EKF, we have this form which is X dot is something like f of XU t plus G t W t Y is like this. This is continuous time where this is discrete time. So, initialize both the state and co-variance matrix and you compute the gain. Wherever there is a measurement coming that part of time, it will operate. We quickly compute the gain, use it for update and update co-variance matrix also and then, go back to this dynamics in continuous time that we know and propagate it actually. So, using the non-linear dynamics, we propagate from X k hat plus to X k plus 1 hat minus and using this linear error co-variance matrix equation, we propagate from P k plus P k hat plus 2 PK plus 1 hat minus actually. The thing is we have to evaluate this matrix here and this has to be evaluated all the time actually using this X hat of t.

So, this quantity this CK minus have to be evaluated and A t has to be evaluated while propagating that actually. This is the only difference. Otherwise we are pretty clear how to operate EKF now actually. So, the equations are not difficult or it is not difficult to understand really, but there are many issues there which make implementation of EKF. It will be tricky here in some sense. The little bit of art involves and experience comes a very handy actually here, but there are several other ideas around EKF. The first thing that comes to mind is something called iterated EKF.

(Refer Slide Time: 25:56)



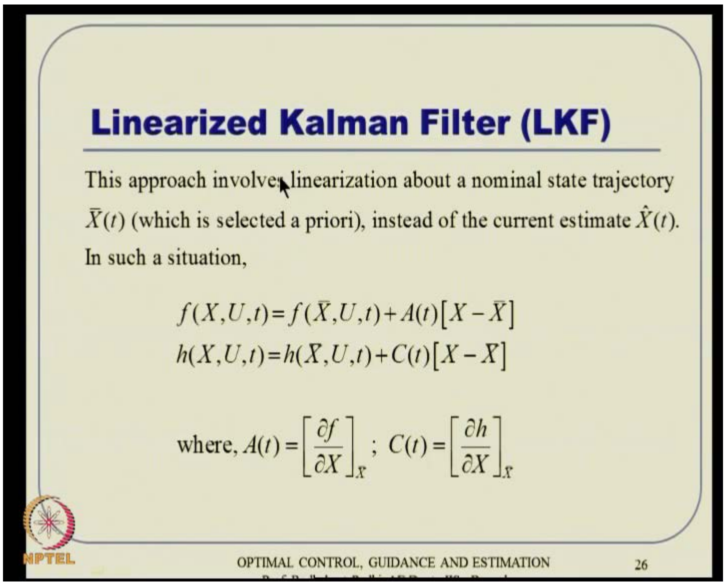
So, this idea is like that one way improving the performance of EKF is to apply local iterations to repeatedly calculate this and each time by linearising about the most recent update. Now, what is the whole idea here? Whenever there is a measurement, we quickly compute the gain, you quickly compute K minus 1. I mean C k minus plus, sorry C k minus first and then, quickly compute Kalman gain using this C k minus and then, we compute this. Now, the moment we update this equation X k plus X hat k plus, we have updated this value to this value.

Now, the moment we update this value, then again you can evaluate it about that value. Why operating it based on a previously predicted value? We do not have to do that. We have an updated value. So, we take that and put this updated value here. So, we get a new C. Using the new C, you can compute your new gain and then, again come back and update and this cycle can continue basically. So, that is called iterated EKF. We can keep on doing that and that is called iterated EKF and you can actually fix a number of iteration. There is no point in waiting until stability and things like that and then proceed further.

So, like some day some people can have to fight and you can fix based on your computational platform and computational capability and things like that. Then, operate it based on that, but it is still based on EKF, external Kalman filter, but it is now doing several iteration based on the same measurement actually just because your c matrix is changing basically. The moment you update it, there is a different value of X and hence, your C matrix changes and again, you keep on iterating on that basically.

So, that is the first idea iterated EKF and there is a parallel development which is also called linearised Kalman filter. Well, it is actually people thought about it. Initially, it is a kind of a competitor for EKF and all that actually.

(Refer Slide Time: 27:58)

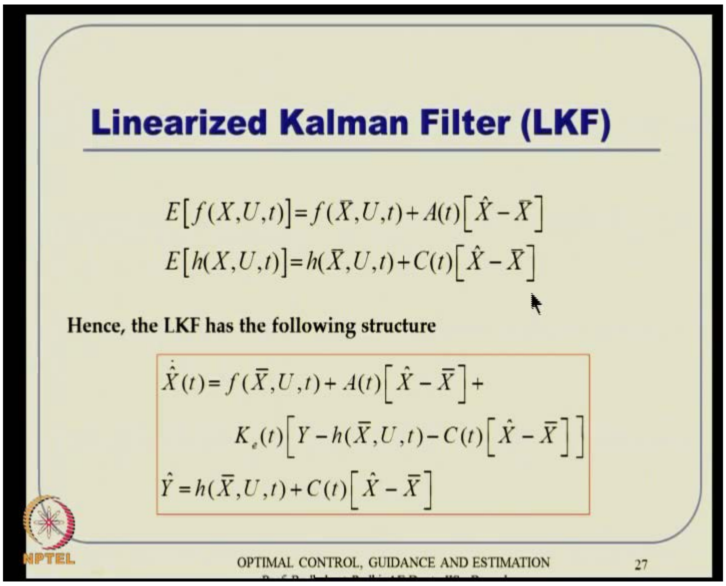


Well, let us see what the idea is there. So, idea here is something like this. This approach involves linearization about a nominal state trajectory which is selected a priori instead of the current estimate. So, what happens in estimating about current estimate is that transients becomes quite by initially. You do not have any idea about what you are doing or the arrow value can be very high. So, the very fundamental assumption that your true state is close to the estimated state or best available information is not there. So, that is the fundamental difficulty of EKF. So, people thought, well in that is because we are bothered about that, I really do not have to do that. All that I need to know is some sort of a linearised approximation of the nonlinear system dynamics.

So, in that sense, they told I can estimate something like AX bar of t which is selected a priori and I will linearise it about it actually. How do what the meaning of X bar of t? Let us say a satellite orbits and things like that. So, you talk about orbital corrections and all that actually if suppose you want to do that. So, it is a satellite is a kind of a known orbit. The known orbit information is already available basically. So, using that information, we can think of operating based on I mean, you can think of having your Kalman filter implementation based on that assumption really, ok.

So, then what you tell is f of this f of XU is nothing, but f of X bar because that is what I know, but the fact is that these guys will not cancel out anymore. If it is X hat, then X hat minus X hat was there, but now, it will not cancel out. It will remain actually. So, A of t is evaluated at X bar. Now, del f by del X evaluated XX bar and C of t is del h del X evaluated X bar. Again, I am telling if you take expected value of all these expressions, then expected value of f of X, this term will not go basically. This will become X hat minus X hat bar which will see this entry. So, if I take now this expression is from Taylor series expansion and keeping first order terms only, ok.

(Refer Slide Time: 30:19)



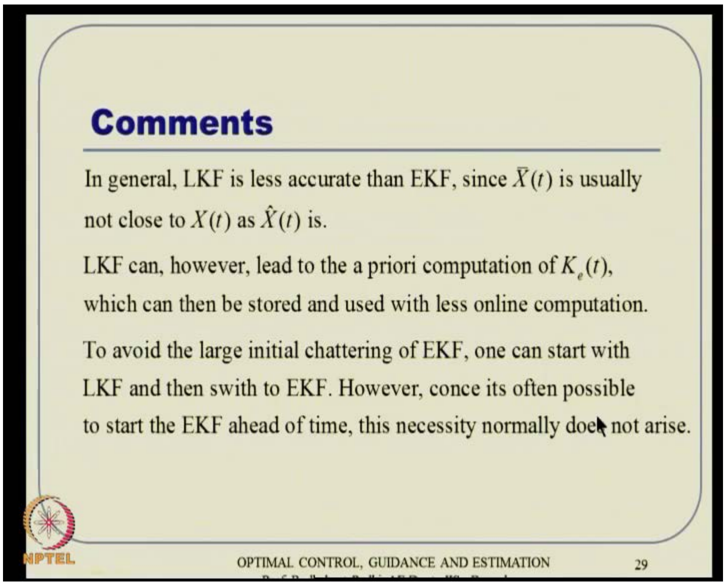
Now, if you expect operator here, then this first term is as it is, but second term, this term will not go. This will remain actually. This is no more XX bar. This is no more X hat. Earlier it was X hat. So, X hat minus X hat was 0 and now, it will stay. So, keep that A t times X hat minus X bar and all that, so that is to propose this equation now. So, we tell X hat dot is this quantity. What you see here, what you see here plus first Kalman gain times the actual measurement minus predicted measurement. Predicted measurement is this quantity, so minus these over everything actually. So, put it that way.

So, Y hat is obviously, if you go back to this expression, this is by definition is Y hat is something, but this one plus C times, this quantity actually what you see here. So, the co-variance matrix can be derived and you can derive it fully. Now, you can go back and derive this is something like P dot will turn out to be the full expression again, where A and C have to be evaluated around X bar, not X hat. All right. So, that is linearised LKF, linearised Kalman filter basically, but in principle, it does not give too much of a benefit at, I mean unnecessarily it gives us algebra complexity and things like that. So, people thought why to bother so much about it actually.

So, the most popular form still there is Kalman filter, but sometimes LKF is probably beneficial to use it initially to get some form, some sort of conditioning of these numbers before UX hat EKF and all that. So, the transient information can transient performance can be relatively better. So, that is the utility of this linearised Kalman filter in general.

So, as I told before, Kalman filter has several concerns and all that so far let us see some of these comments and other things. So, first of all, LKF is less accurate than EKF since, X bar t is usually not close to X of t. So, what happens if you do this LKF? In that form, your trajectory is always there. So, even though we have a better estimate, you are unable to use it actually subsequently. That is the whole idea there, but you can like LKF can lead to this a priori computation of K e t, which can be stored and use in online computation which is not possible in EKF.

(Refer Slide Time: 32:31)

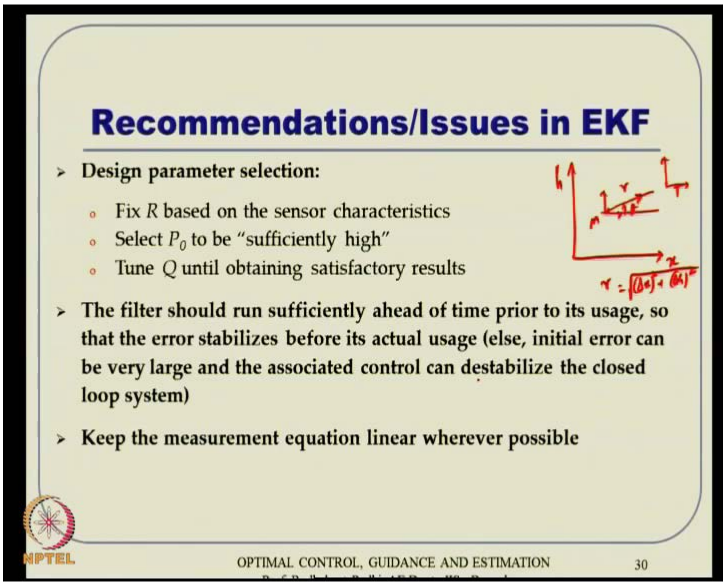


So, sometimes as I told to avoid large initial chattering of EKF, one can start with LKF and then, switch over to EKF. However, it is often possible this over it is often possible to start the EKF ahead of time. So, that means, when you want to kind of close in or modify your control guidance whatever based on the estimated information, so you can actually start the EKF little more time in advance basically.

So, then these transients are expected to write down by the time you are interested to operate based on this estimation information. For example, if you think about missile gradients application and all that in the terminal phase, you would largely be interested in seeker based information that is much more accurate, but that filtering can be actually initialized based on radar data. So, little before, little time before, so that your better values of gains and other things are available by the time. Thus, require starts operating or you close your guidance based on the require data basically.

So, that kind of application, I mean mutually it is done that way invariably because the initial transients can be very bad. So, you start with little bit ahead of time and then, tell by the time I want to use filtered information transients and write down actually. So, essentially, if you do that, then this necessity of first starting with LKF and then, switching over to EKF and all that also normally does not arise actually and the philosophy of local iterations, that is LKF can also be implemented here and that leads to this iterated LKF concepts sort of things actually, but very rarely again, we will hear terms like LKF and literature and all that. It is not very popular anymore because there is no specific advantage over EKF actually. So, that is what I write here.

(Refer Slide Time: 35:26)



So, there are various recommendations and issues for successful implementation of EKF. Let us study one by one. First of all, how do you tune it actually? That is the major issue and the very first thing that comes to mind is something like this fix R based on sensor characteristics. So, based on whatever sensors are available, we fix a number for R. Then, select P naught to be sufficiently high. In other words, we have information about initial value and it tells whatever initial value of the state what we are guessing, this is large error actually.

So, we take P naught to be very high, sufficiently high and that is some more amplification that you, if take P naught high, then it is going to convert much faster than all that actually, whereas the transient can be high, but it can convert much faster. So, after these two, after you fix R based on the sensor characteristics and then, select AP to be high. Then, what remains is Q. So, then you have to vary this Q up and down. In other words, tune Q obtains until obtaining sufficiently satisfactory results actually.

So, that is kind of a universal recommendation for implementation of EKF. Again going back to that something, that filter should run sufficiently ahead of time prior to its usage, so that the error stabilizes before its actual usage. Otherwise, the initial error can be very large and the associated control guidance, everything can go back. It can destabilize the closed loop system. So, this small kind of comment that is very critical for practical application actually also keeps the measurement equation linear wherever possible because see the two choices, typically either we work with linear system dynamics or nonlinear measurement equation or you work with linear measurement equation and nonlinear system dynamics.

Typically, if you have this formula, I mean again going back to error space application, you have this probably a kind of missile guidance problem, where in X and h are looking at the error dynamics and all that. Suppose, the missile is here and some target is here and the information typically comes in the form of something like r theta, this angle. The sensors will look in the frame work r theta, but the inertial coordinates in the form of cartesian coordinates and hence, if you formulate these dynamics in terms of X direction Y, I mean X direction and also like dynamics of target in terms of X and h direction and things like that, it turns out to be something like a kind of cartesian problem actually.

So, in other words, system dynamics can be in the form of cartesian coordinates where the measurements can come from the polar coordinates actually. So, now we have two sizes. If you stick to that, what happens is the polar coordinate information has to be converted back into the state space equation. That means, the r has to be expressed in the form of something like delta X whole square plus delta h, the whole square like that actually a square root and theta also, I mean 10 theta is del x by del h and all that.

So, in other words, the measurement equation becomes nonlinear actually. So, what about looking at in alternate sense? In other words, you tell my system dynamics is already in the form of polar coordinate. So, instead of this cartesian coordinate dynamics, I will talk about polar coordinate dynamics. Then what? The measurement is already coming in the polar coordinates and that will make my measurement equation linear actually.

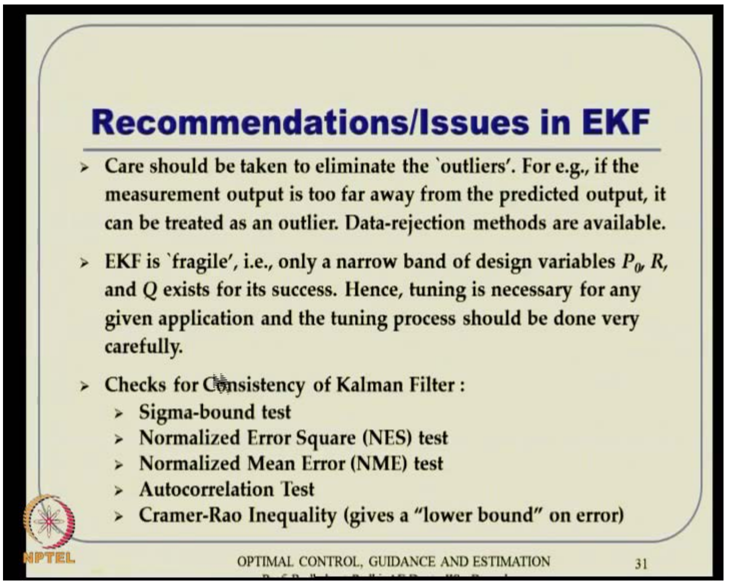
There are many observations from practicing people and some literature and things like that, that people have observed. Many times, it may not give a specific advantage looking at this way or that way, but sometimes, it can have some advantage. When we look, when we formulate the problem where the output equations are typically linear, where the state equation can go nonlinear. So, again you have to truncate anyway. So, somewhere you have to evaluate C and somewhere you have to evaluate A.

So, the moment you evaluate that Eigen Taylor series, there is some sort of truncation actually. Another question is where you want to do that and the recommendation from practicing people and things like that is that typically you prefer that the output equation remains linear actually, but there is no theoretical justification for that. Let me be very clear on that actually, all right.

So, this is what the next one is. Care should be taken or sufficient care should be taken like that to avoid in numerical ill conditioning and that can happen due to several reasons including truncation errors and all that, especially an early computers. The truncation errors are punishing because for this digit, I mean this floating point computational are all very limited actually. So, we are truncating all the numbers very quickly and then, that was a major issue. Now, this 64 bits and all that if you all do real, can implement may not be a major issue, but still there are issues because of numerical problem.

For example, many of these things can be written in several books including this book actually which will list out at the end of this lecture, but there are several improvements of algorithms had also happened over the period of time which will try to avoid this kind of issues actually. For example, this P dot equation or this update equation that we discussed in discrete time domain, we want to use the symmetric expression of propagation rather than asymmetric expression basically. Even though, that is the simplified formulae. So, things like that behavior of that actually.

(Refer Slide Time: 41:13)



Here, again the care should be taken to eliminate outliers. An outlier is something like a kind of innovation actually. So, you have this innovation coming into picture in our implementation thing. Sorry, here somewhere, ok. This term is actual output minus predicted output. So, this term contains innovation actually. If it is very high for whatever reason at the particular list out time or couple of time and interval and things like that, you tell something wrong might have happened in the sensor or somewhere, so it is not giving proper value.

So, let me not correct everything. We have done everything blindly waste on whatever I see. If I do that, my thing can go very weird actually. So, let us not do that. So, I mean I will do that based only if the error is not very high, ok. The error is reasonably ok, then I will update thinking that my measurements are good. The error is variant point out of time, then it will look something that is really happen. I do not want to kind of account for those thing. So, I will ignore that rather continue on the prediction mode basically, ok.

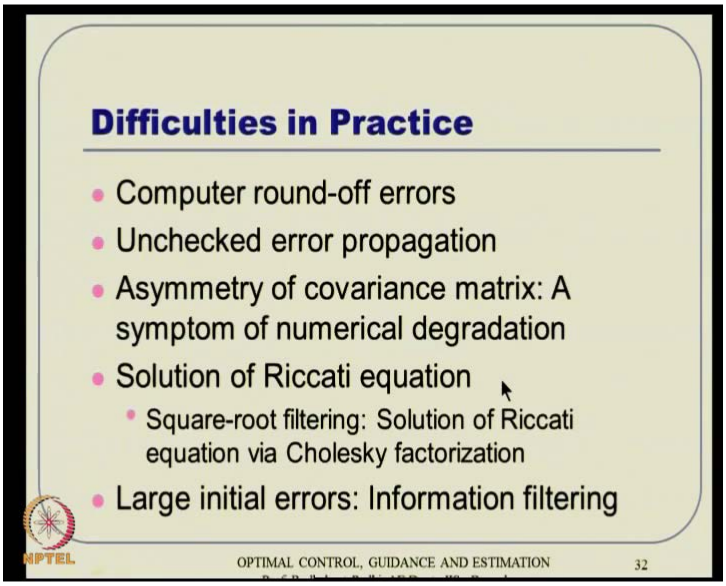
So, that is other thing, but also remember, some data rejection methods are also available, a little bit formal methods and all that actually. You can see that and bottom line is EKF is little bit fragile. In other words, only a narrow band of design variables P naught R and Q will exist, but do not lose hope and have sufficient patience in tuning these variables for any given application. There are some recommendations that we discussed here. This way, you can start like that and then, change values and all that . So, that is the tuning plus here actually. That needs to be done a little bit carefully with some sense of, I mean should have sufficient patience to do that actually.

Ultimately, it is going to work and when it works, it works wonderfully actually. So, have some patience to tune that actually. Then, there are lot of consistency issues in Kalman filter and there are lot of checks and bounds in Kalman filter. So, something like sigma bound test, something like normalized error square test, something like normalized mean error test, something like auto-correlation test, something like Cramer Rao inequality and things like that. So, there are various stages and always subject your results with respect to somewhat these tests and invariably, it should have sigma bound test. In other words, if you have P matrix, it gives us lot of information and the diagonal elements of the P matrix are essentially sigma squares basically.

So, if you collect sigma and plus plot this plus or minus sigma or plus or minus 2 sigma and things like that, ultimately starting from any initial condition, the error should come within that actually, ok. So, that is called sigma bound test. Ultimately, I mean in any experiment, these results should be a kind of a must actually, otherwise there is no confidence on what you are getting. You may get a stabilized value which may look good, but your P matrix may tell something and your estimation may be somewhere else actually. If that fells that, then there is no sufficient confidence on the results basically.

Similarly, other things can also be sensed a whiteness text also, just the auto-correlation whiteness test and all. That means, it turns out that innovation quantity if you test it out, we have a very random variable upper from a white noise you have taken out a kind of continuous signal. So, it does not matter. The innovation will again turn out to be white actually. So, you can think about bring this test also and things like that actually, ok.

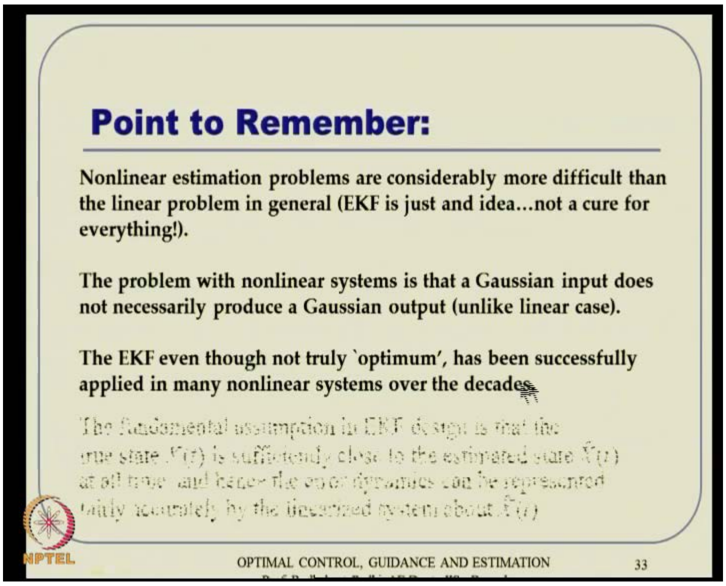
(Refer Slide Time: 44:55)



So, some other difficulties in practice are computer round-off errors are always concern. So, no matter how, whatever digit of accuracy you want to operate for a longer period of time, then this round-off error can still may have major concern. Especially, for more computers, they are still not based on very powerful computing and all that, where round-off errors can be very punishing. So, beware of that and something like if you have some error propagation happening and you do not check it, the unchecked error propagation can cause disasters actually and asymmetry of co-variance matrix is a large symptom of numerical degradation. Can you watch out for P minus P transpose norm, if you evaluate that will give zero. If it is symmetry and if it is asymmetric, it will give you start popping of some numbers and all that which can be thought of some sort of an indication that things are going weird actually.

Solution of Riccati equation. There are ways of doing that and something called square root filtering. This talks about some solution and via Cholesky factorization and all that. I mean, these are all the issues of the implementation really. So, it is not like Eigen value implement something gets some answers actually, ok. So, using this factorization if you solve Riccati equation and then use that, then it turns out to be much better actually. In case, there is some large initial errors and we have absolutely no clue of how to guess it and things like that. There are ideas like information filtering available and more, I mean the filtering is itself you can be a course and all that. So, there are many ideas. I suggest that you use some of that actually.

(Refer Slide Time: 46:35)

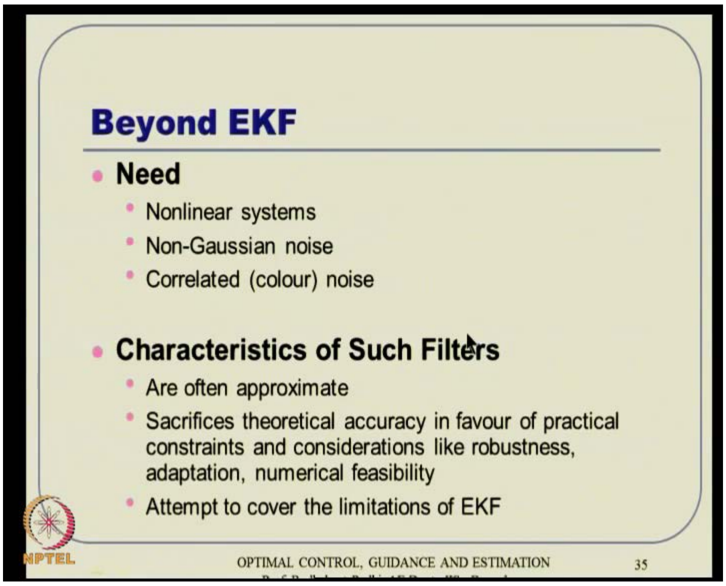


So, some points to remember are nonlinear estimation problems are considerably more difficult than the linear problems in general. Again, EKF is just an idea and not a cure for everything, ok. The problem with nonlinear systems is that a Gaussian input does not necessarily produce a Gaussian output, ok. All these things I have talked about basically. So, let us talk about limitation of EKF and limitations turns out to be significant also. First, limitation is linearization can introduce significant errors. That problem is highly nonlinear. Linearization is not actually typically. There is no general convergence guarantee and convergence observation. There are many times you may get it for in general, there is no convergence going to be at all.

So, it works in general, but in some cases, its performance can be surprisingly bad. So, it typically happens to be unreliable for colored noise and some ideas like shaping filter is just available, where you take a little bit subsystem and then, put it to white noise as input, so that output can be kind of the noise that we are looking for an unnecessarily white. So, that little has blocked, I mean this system dynamics can introduce additional states and things like that. This is called shaping filter actually. We have discussed that before and we will take an example also in the next class somewhat.

Next class will typically take an example in one of the lectures later to see what certain filters is and how to implement that actually. So, there are necessity for beyond EKF and the need essentially turns out to be because of system dynamics is typically nonlinear, noise does not necessarily satisfies goes in PDF, ok.

(Refer Slide Time: 48:09)

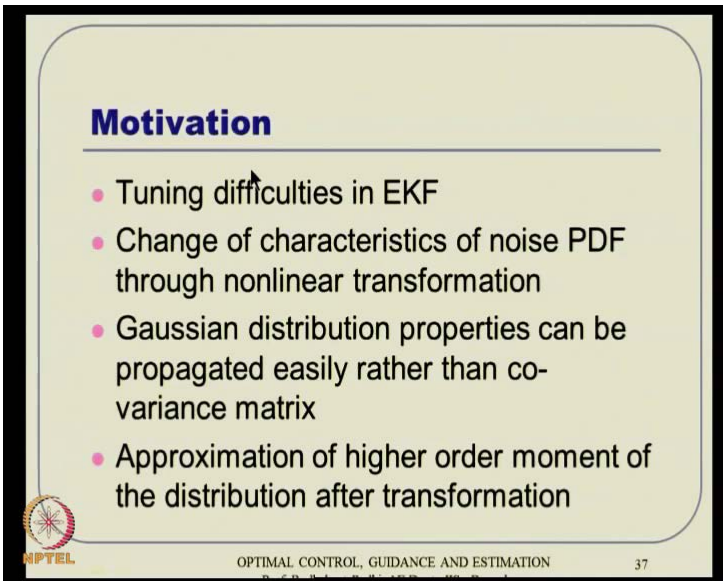


So, Non-Gaussian noise actually and also, we have this noise is also a physical phenomena. It cannot happen extremely randomly basically. Noise happens because of certain characteristics. So, assuming that, it is completely uncorrelated. That means, whatever happens now is completely ignorant of what happen immediately before now is very unrealistic also. So, correlated noise or color noise is also always a concern. White noise, I mean assume white noise, it has given lot of zeros in the derivation and make those derivation lot simpler, but that does not mean that it is close to reality actually.

So, you have this correlated colored noise. Then, what actually? So, this precise will start popping actually, ok. So, characteristics of this advance filter or something like this. They are often approximate, not very good in some sense. The sacrifice theoretical accuracy in favor of practical constraints. So, what we are talking here? We are talking here something like advance filtering, which is something like, something called h infinity filtering and something called particle filtering and things like that actually. Those are available, ok.

The Gaussian based filtering also, Gaussian based particle filtering happens to be one of the phase actually. So, this essentially sacrifice theoretical accuracy in favor of practical constraints and consideration like robustness adaptations, numerical feasibility like that actually will become (()). So, essentially, that attempts to cover the limitations are EKF and run into some of the reuse. If you can solve it, then it turns out to be good actually, all right. So, going too much beyond is not possible, but little bit beyond what is called as Unscented Kalman Filter, we can have glimpse of that, not in a very detailed sense actually.

(Refer Slide Time: 50:06)

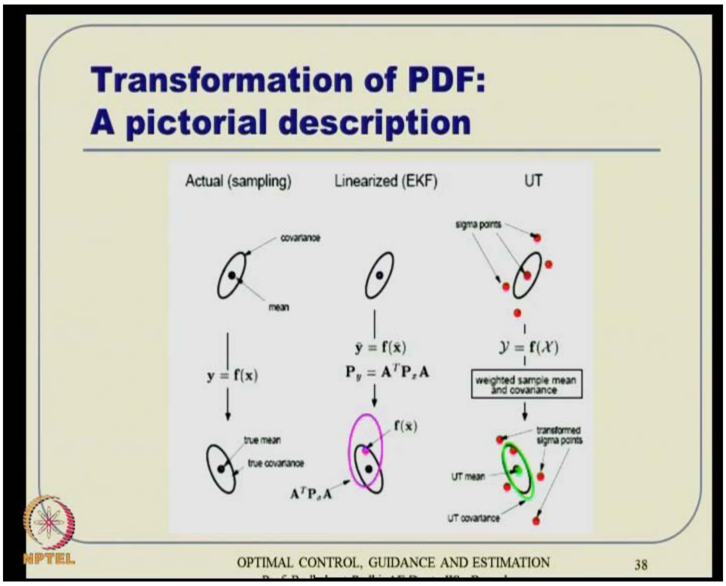


So, what happens here is motivation is something like that. There are tuning difficulties of EKF, we know that. So, can it be relaxed and in other words, can it be little lesser difficulty that way? Then, the second motivation is this change of characteristics of noise PDF through nonlinear transformation, ok. You can still characterize some sort of time varying Gaussian PDF. It is not stationary, ok. It still can be interpreted. That is a Gaussian PDF, but it is not necessarily stationary. In other words, this mean and variance will keep on varying actually, ok.

So, if you consider that way, then is there any idea that you can update this mean, I mean, mean and co-variance of this PDF itself actually. So, one liner through which essentially started from Oxford University, UK something like this. There idea is like, ok. Gaussian distribution properties can rather be propagated easily than the co-variance matrix itself actually. So, we propagate some of this distribution properties, something like sigma mu and all that. Then, evaluate this co-variance there once you do that.

So, there is no need of propagating this co-variance matrix and running into difficulties actually, ok. It also gives us a platform. If necessary, we can approximate the higher order moment of the distribution after transformation. If you take more and more points, what are called a sigma points and all that. If you consider more and more sigma points, it is possible to compute higher order moments. Also, the moment it is higher order moments are also assured to this guardant to be same and things like that. The accuracy become more and more greater basically.

(Refer Slide Time: 51:48)



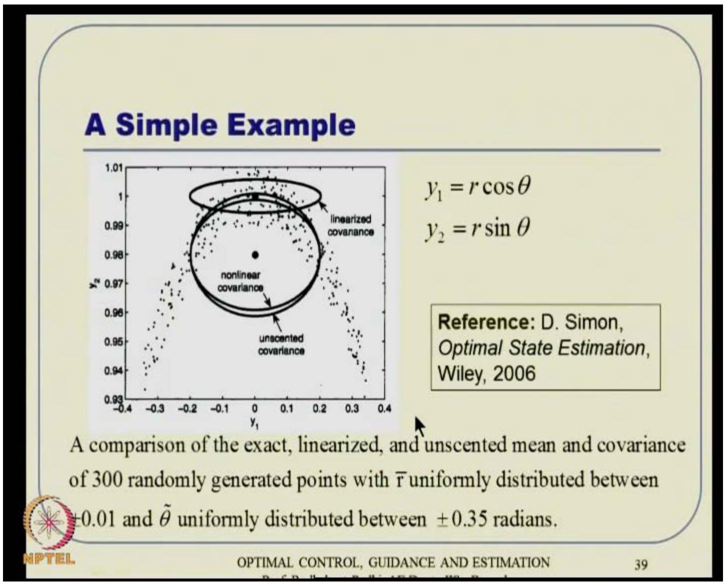
So, this is a typical picture that appears in many of the literature and things like that. I mean, this is conceptually very intuitive picture. It turns out let us start with a estimation with a mean and some sort of a co-variance bound around that and if you really propagate through a nonlinear function, it takes a different shape actually. So, mean can go transfer somewhere else and then, the co-variance can take a different shape altogether actually.

Now, the whole idea is how best you can approximate this. Now, it turns out that if you transform through this propagation ideas, then P Y you can take a very different shape actually. So, essentially the true mean and variance can be somewhere else. This is your turn mean and true variance, whereas predicted thing turns out to be like this. This is the mean and that is the variance. So, obviously, the properties of the Gaussian distribution gets started actually. What you got yesterday is something like this.

Now, the question is how about random sampling? You can do thousands and thousands of sampling and turn pass everything through this nonlinear function and then, re evaluate the mean and co-variance again. Then, it will turn out to be close to that. I mean, that is how you can get an idea of what is going on here actually, but the question is this thousands and thousands points is not a feasible answer. It is not possible at all to do that. So, it turns out that can you do that in a minimum number of samples basically. So, that is the whole thing that turns out to be some sigma points concepts and all that.

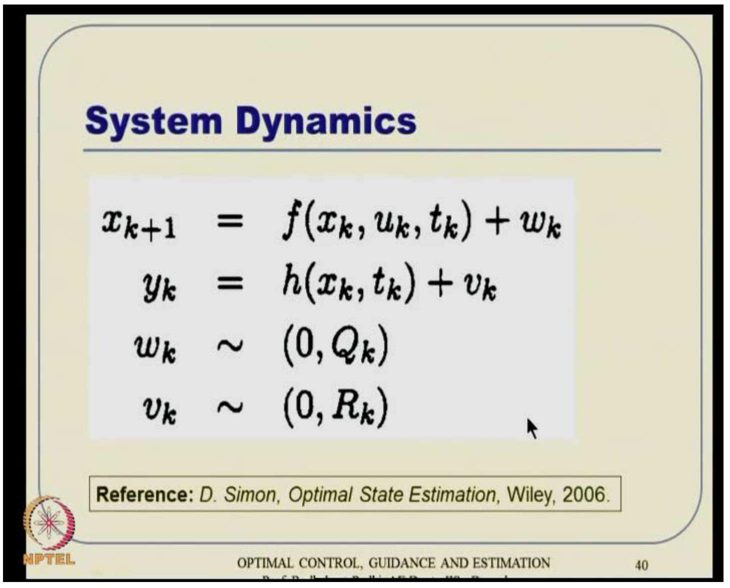
So, if I can selectively select, I mean kind of judiciously select these points and it turns out to be some Eigen vector properties and all that will come into picture here, ok. If I judiciously select these points, these are all sigma points and if I passed these sigma points are evaluated quickly, this mean and co-variance, then it turns out to be very close to what it should be actually, ok. That is the whole beautiful thing here. There is a minimum number of points using which I will pass all those to the same nonlinear function and after I will pass out, they will fall somewhere. Then, I can quickly evaluate this mean and co-variance and again, I will select a bunch of another sigma point based on this linear distribution and then, proceed further actually like that.

(Refer Slide Time: 53:58)



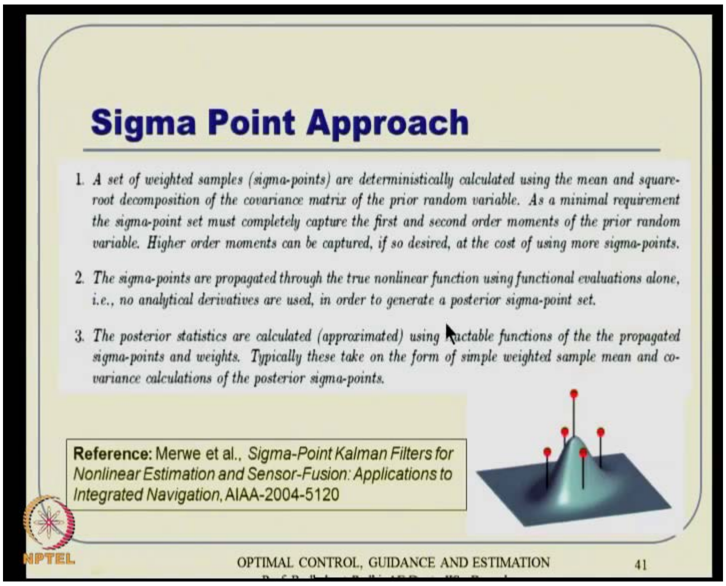
So, there is a simple example to demonstrate these ideas in this D. Simon book. It is a very good book. Again, it is polar coordinate to Cartesian coordinate. You have rth cos theta r sin theta. If you do that, if you do this nearest co-variance, it turns out to be somewhere like this. If you do this sigma point propagation and it find out again, then it turns out to be very close to what you should be and that is what the whole idea is to demonstrate this unscented Kalman filter sort of thing. More details you can see in this book also, ok.

(Refer Slide Time: 54:26)



So, you have this, let me very quickly you have this linear dynamics all and history domain measurement equations also like that. The entire equation concepts are taken from this book. More on that, there is a good chapter around that. You can read it actually. So, this is what it is and then, there are some other thing that I saw from a different literature. I thought of putting it here.

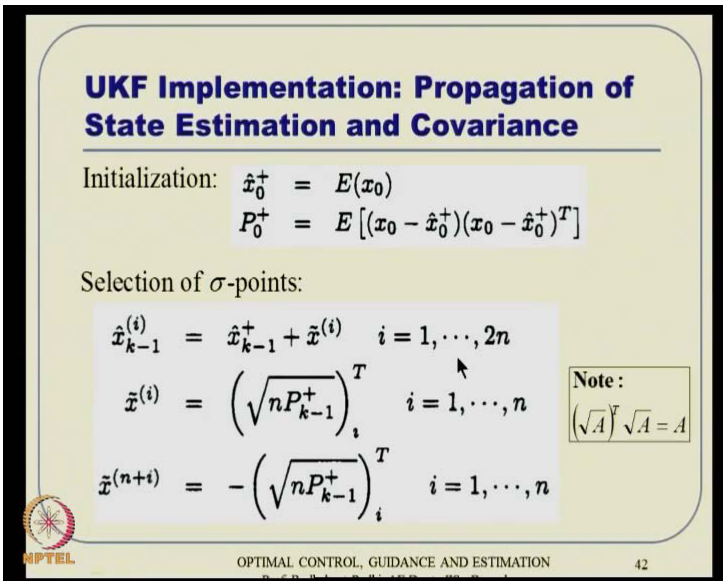
(Refer Slide Time: 54:43)



What you mean by sigma points actually? So, sigma points is a set of weighted sampling points and things like that. So, we have one on the peak and then, four on the outside and things like that. You can see that actually, ok. So, these sigma points are propagated through the true nonlinear function using this functional evaluation alone and there is no analytical derivative used in order to generate this posterior sigma point. These are the selection will pass it through the nonlinear function and there is no derivative information involved. Jacobean matrix is completely avoided actually, ok and the posterior statistics are calculated or rather approximately calculated using this information actually. These definitions are taken from this reference really actually.

So, how do you implement it without derivation again? It initialize the initial condition and co-variance matrix and then select bounds of sigma points or something like this, 1 to 2 n, sometimes 2 n plus 1, people recommend sometimes 2 n. This center point is type of kind sometimes evolving not there basically.

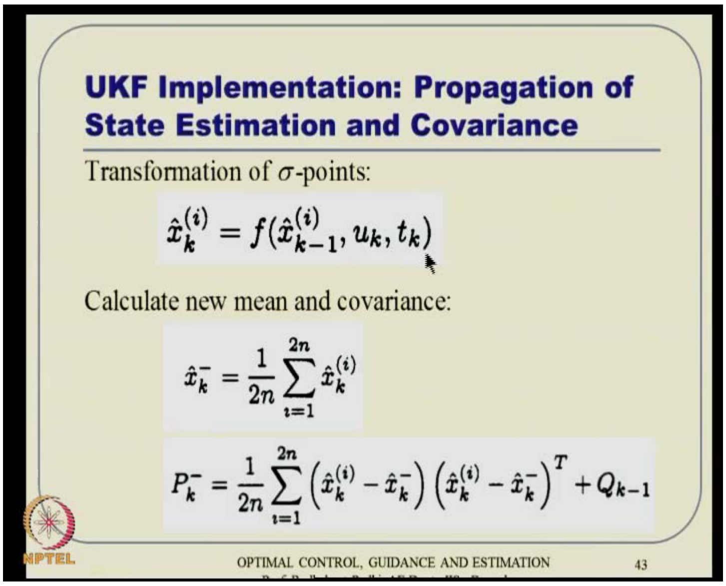
(Refer Slide Time: 55:29)



So, you use the selection of sigma points. I mean select a bunch of sigma points like this 1 to 2 n and n is a number of states, remember that. So, first 2 n, I mean first n are evaluated square root of this and then, the rest evaluated of this actually, k plus 1 is I mean the negative of that. So, they are symmetrically placed basically. Remember as long as there is a positive definite matrix, we can talk about a square root of that also basically.

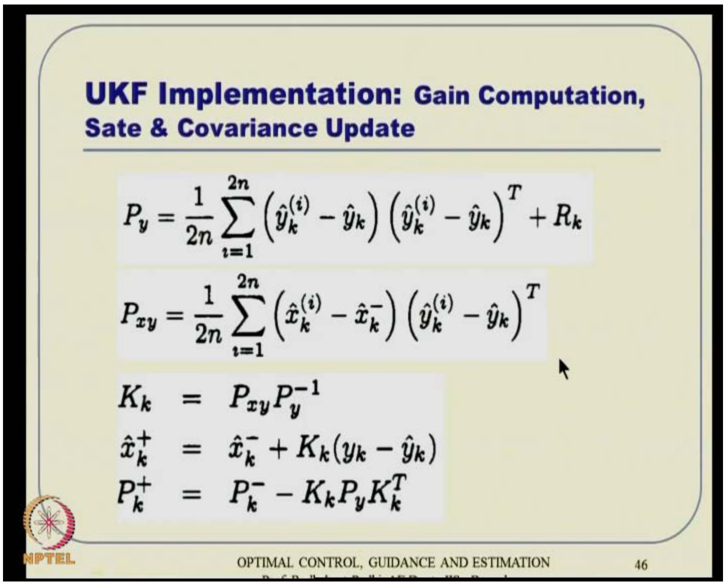
So, we can do that. Then, transform the sigma points and then, calculate the new mean and co-variance. This has to be that and once you have that, you select the sigma points again based on the updated value and then, proceed further. That is the whole idea there basically.

(Refer Slide Time: 56:24)



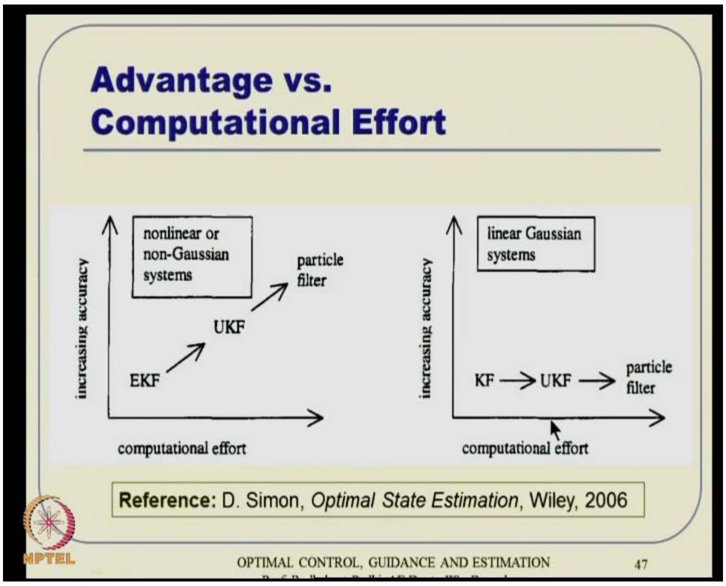
So, transformation of sigma points can be done using the same nonlinear problem equation that you know. So, there will not be too much of a difficulty and based on this, we can have this predicted measurement as well actually.

(Refer Slide Time: 56:55)



So, how to implement it? After you do this transformation there, you evaluate this P y P x y and then, evaluate this Kalman gain and then, update and things like that. Update the state and update the co-variance again. So, it is very quickly. What should be done very easy to implement also in my view and computationally, it is not very taxing either basically.

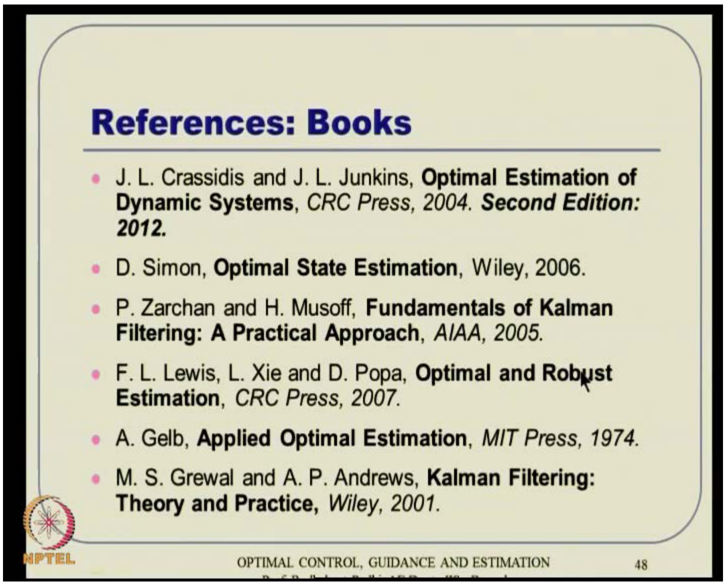
(Refer Slide Time: 57:13)



So, there are other concepts which are not talking here in this course. The concepts like particle filter and all that. We will deviate too much other side instead of retaining this optimality concepts and all that actually. So, these are based on bayson belief and things like that, but there are also nice concepts out there. If you are interested, you can see some of this particle filter ideas again. There is a nice diagram in this book also. I thought I will put it here. There is something like computational effort versus increasing accuracy.

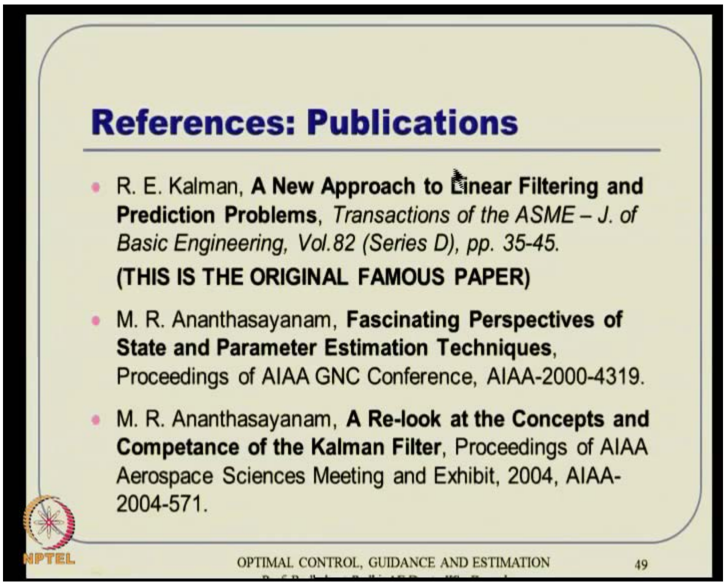
So, UKF well, even though the diagram is like that, I do not think it is too much computationally taxing. It may be little bit more than EKF, but not too much. Remember, EKF, we have to evaluate Jacobean matrix, whereas UKF, we do not actually, but UKF you have to use square root of that matrix is also kind of computationally texting actually. So, that is what the difficulty is. So, little bit computationally more in well, but increasing accuracy, but if you have that advantage is there, only one there is a nonlinear system or non-Gaussian noise and all that. There is a linear Gaussian system. We are unnecessarily wasting things. We are not getting performance, the improvement at all actually. So, be careful about what we are doing.

(Refer Slide Time: 58:18)



So, that is all I want to talk in filtering ideas and all that, but I have listed out a few books and references for your convenience again and those of you are more interested can see these books. These books are classic books actually. There is a second edition available in 2012 also about this book. Many of my derivations have been taken from this book, but also this D. Simon book is also very good and other things are also there. Especially, this one is very particle related to missile guidance and things like that.

(Refer Slide Time: 59:01)



There is a first book that appeared very popular, many several teachers and author and things like that and I can also see this. They are written by practicing people with very good practical implication how do you implement numerical things and something like that and also put some of the publications which appeared. This is the first original famous paper from Kalman and it was appeared in a ASME-J of basic engineering. Then, these two papers are really nice stories out there. Fascinating Perspectives of State and Parameter Estimation Techniques and then, A Re-look at The Concept and Complete Competence of Kalman filter and things like that. Those of you who are more interested to read fun stories and all that, some degree of inside that you want to gain, then you can read some of these. That is all I want to talk about Kalman filtering actually. Thank you. Bye.