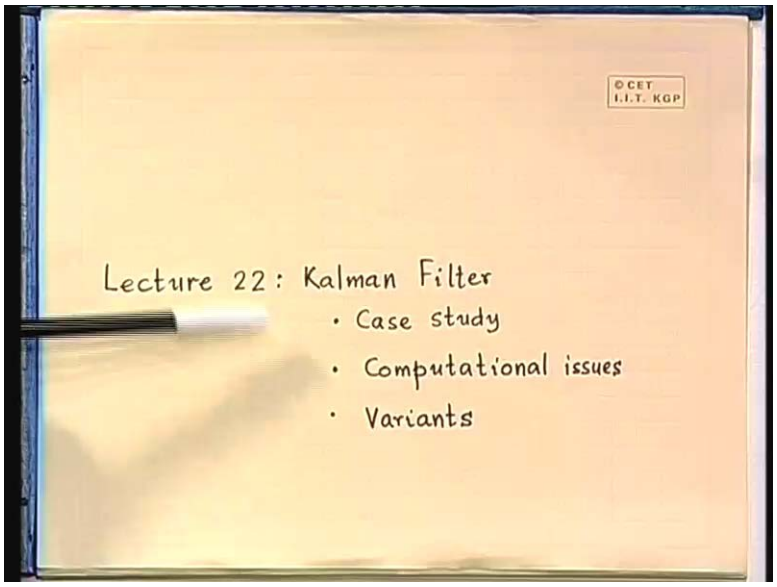
Estimation of Signals and Systems
Prof. S. Mukhopadhyay
Department of Electrical Engineering
Indian Institute of Technology, Kharagpur

Lecture - 22
Kalman Filter – Case Study

Very good morning, this is the last lecture on Kalman filter.

(Refer Slide Time: 00:59)



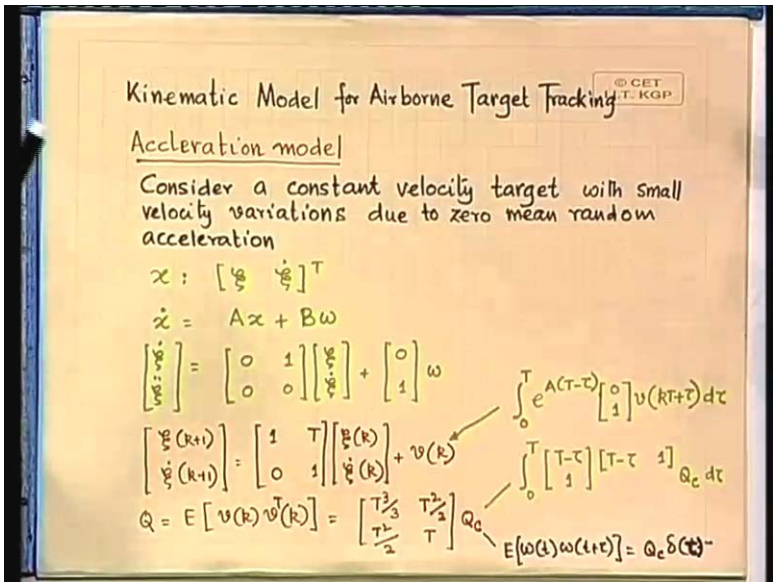
So we have seen most of the mathematical derivations and mathematical properties, that major mathematical properties at the Kalman filter has. So today it is an algorithm; so to get the feel of the algorithm, some feel you can get some by by analysis but, for algorithm seen get a lot of field by simulation. So in fact that is why the assignment was given in the last tutorial class, though the few were absent should take note of it.

So so today we will show a some you know limited simulation studies of the Kalman filter; mainly to substantiate what I have been always, you know telling you that it is that it works very well; it it removes a lot of noise and we will also see the effects of some of the some variations, in the I mean there are some key parameters so if you if you choose them in

various ways, what is the effect on the filter estimate, that is something that will be revealed through this simulations.

Then we are going to touch up on certain other things; we do not have I mean I do not think we can treat them in any depths. So rather than you know describe them in detail more and more like, you know we will talk about that such problem due exists; so beware and there are n number of references you can always look them up. So we are going to look at real computational issues and so some some some of the things that that could give problems, if you really want a apply a Kalman filter in a in an actual case, so there will be mentioned. And at least one variant of the Kalman filter which is which is very popularly known as the extended Kalman filter. There are there are even other variants of the Kalman filter. So we will we will we will have time to discuss only one variant, that is the extended Kalman filter in some detail and that will close or course module on state estimation. So to to begin with the case study;

(Refer Slide Time: 03:14)



we will we if the system is is the as if some there is there is a there is a moving object, okay, It could air borne, it could water borne, whatever but it is moving. So our objective is that we are there is we are assuming that; if you assume that there is some sensor which can get its position information then can we extract, let us say other information like it's velocity. In some cases you may be also interested in in measuring it's acceleration; now it may appear to

you that, what is there I mean if if you have position if you if you differentiate position,n you will get velocity, there is no problem. But it turns out that; if you if you take such a nice approach you land nowhere, you will get absolutely no estimate of velocity at all if you get an noisy measurement of position and if you try to directly differentiate it.

So so you what you need to do is, you need to do more sophisticated estimation which is by Kalman filter and and we will when when we see the our numerical simulation this will be borne out. So here we are we are assuming there are there are various kinds of models we are which are which are used, we are using a simple model where we have not exactly a constant velocity but more or less constant velocity target; with small velocity variations we are assuming that the that the velocity is more or less constant, but there is a small acceleration component which is random and which is causing a slight velocity variation.

So the velocity is nearly constant but then varying in random manner. So if we try to model the kinematic of such a system, then we can define as state the position and the velocity, okay. So xi and xi dot, then we can write this states space equation; this is in in in continuous time that is x dot which is xi dot and xi double dot is equal to A x plus B u. So xi dot is equal to xi dot; therefore this is1 and xi double dot is that random acceleration component, right. So this is my A matrix this is my B matrix in continuous time; because always discretize this if you discretize, this this matrix is nothing but e to the power AT where where where A is this. You can take it turns out, that for this matrix A squared is 0 from from from A square onwards A square A cube or a higher product all higher terms are zero. So in this case e to the power A t is is nothing but i plus A into T, so you get this, right.

So so so this is the discreet time state state matrix and similarly if you take this, how do you get the discreet time input? That is by e to the power integral 0 to T e to the power AT minus tau, B u tau, D tau, tau between T to T plus 1. The effect of on xi dot will be the total accumulated effect of w from T plus kT to to T plus k plus one. T that total accumulated effect you are approximating where constantly as if at the kth instant, you you applied a constant signal v k which remained constant from k to k plus one; and produce the exact the same sample value of this xi dot k plus 1, right, that is how you get it, right. This is a sort of a is what is known as a Stapin variant approximation of continuous times system to get a discreet time signal I mean I mean discreet time model.

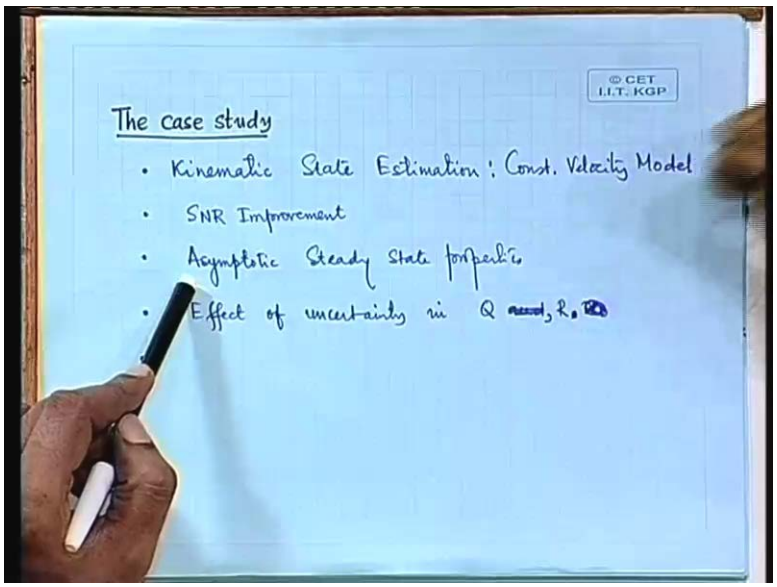
So this is you discreet time model. Now you have to estimate what is Q, you have to now get the discreet time estimate of Q, that is the process noise; because this process noise is not that, this not this process noise. So if you have a continuous time process noise, the discreet time process noise is actually related to the continuous time process noise by this matrix. This is simply just take we have already found this, so if you so this will turn out to be this that is simple. So if you are if you know the continuous time signal covariants, this will be the discreet time signal covariants; so that they will be related by various functions of the sampling time, right. So this is my model, okay. What is my what is my measurement model? That is Measurement model is simple; measurement model is simply one zero into x because I am getting sample measurements of the position, that is what I am assuming, so my c matrix is one, zero.

Now, is this system observable? It is. What is what is c? c is one zero. what is c a? what is c a, zero one. So the matrix c, c a is one zero and zero one, it has full length, therefore it is observable; now which means that if indeed if you are given the position, then you can estimate the velocity, correctly. But if your measurement was the velocity, for example if your c matrix is not one zero but but zero one, then then what is c a? If your c matrix was zero one which means that, you are measuring velocity, it was zero zero. So then the c c a matrix should have been zero one, zero, zero. So it would not have been full length. So if you measure velocity your system is not observing, why? Because from velocity you can never get position; why because you need the initial condition, even if you integrate velocity unless you know the unless you do not know the am I am in a unless you the know initial condition; you cannot estimate the position. So with respect to velocity measurements; this system is not observable but with respect to position measurement, it is okay.

So now we this is our, this is my this this is sometimes called a second order model for obvious reasons because the matrix is has dimension two. Sometimes it is called a constant velocity model, right where you have a random acceleration input. Sometimes you you can if you you can you can you can assume otherwise; you can assume that it is a it is a constant acceleration model also in which case your said dimension will become three, then you will assume that you have a random small random jerk input, jerk is the derivative of acceleration. So so you could have I mean I mean people use such models also especially for various tracking various kinds of targets; which may be you know suddenly trying to speed up again

slow down in a in a random fashions. Sometimes it is useful to use the jerk model or the constant acceleration model I mean three dimensional, third order model.

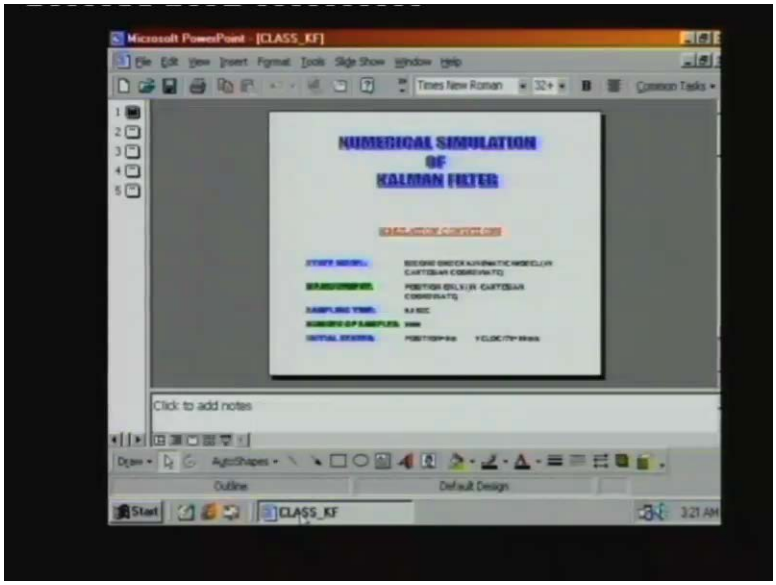
(Refer Slide Time: 11:06)



So we are going to have this case, steady incidentally which was I mean a thankful to Shravani for providing me with all these results. So she has already done an assignment if you note; the assignment was give then that was given is solved by her and I am showing the results, so you have to solve them, yourself. So we have we are going to show a kinematic state estimation case for a constant velocity model and we will show typically, what we want to show is how you get an SNR improvement; we had done a problem on that in in our tutorial class. So now we will see that actually what amount of SNR improvement takes place. We will also we had seen some some of the Asymptotic steady state properties, I mean what does the what does the finally what does the state covariants matrix become?

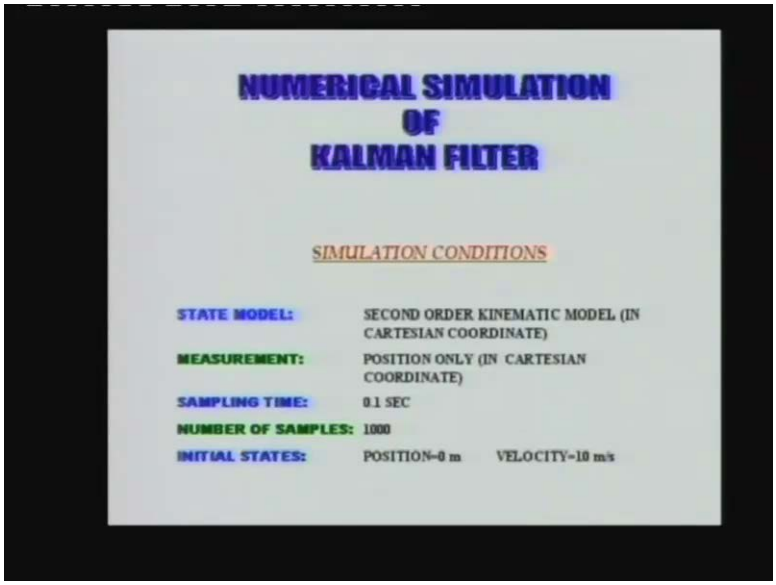
So we will see that even if we do a do an actual stimulation, it does tend to that value it comes very close to that. So our so our analytical values or or expressions will be validated and we will also see effect of uncertainty in Q and R; and we will try to provide intuitive you know interpretations of why this filtered behaves in that way, right. So now we go to the computer and see our case study.

(Refer Slide Time: 12:35)



So we are here and we see, this is it coming is that is good.

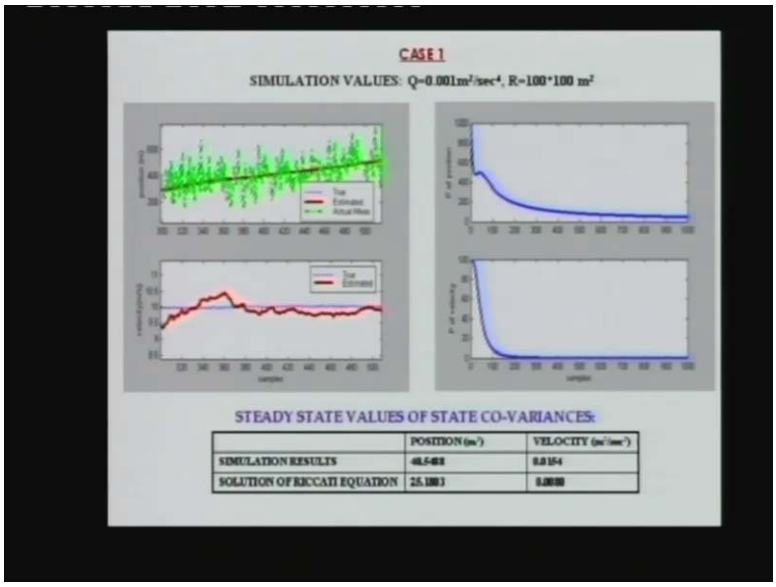
(Refer Slide Time: 12:41)



So here we are showing this one numerical stimulation of of of a Kalman filter, what are the simulation conditions? These these state model is a second order kinematic model, just just the one that I have that we have seen; the measurement is position only assume to be in Cartesian coordinates, that is incidentally this this simulation is in one coordinate only we are we are looking at the motion only in one coordinate; there may be other two other coordinates

which we are not looking at. The sampling time is point one second, the number of samples for which this simulation were were done was thousand; and these are the initial states position zero meter and velocity ten meters per second. So you can see that, the it it really cannot be an airborne target, because the because the velocity is too small. It is probably, it can perhaps be a be a water borne target or even and even a any other moving target. So now let us see the first result, okay.

(Refer Slide Time: 13:41)



So the values is first note the value of Q and the value of R, so typically uncertainty in in in position measurement is about hundred meters; and Q is if you if you want to know the the the the typical you know level of variation in the acceleration input, that is a random input that we are taken you have to take about this square root of that, so it is about point zero three. Note the note the unit here it is meter square per second to the power four, because it is square, right.

So the actually here it is meter per second square for the acceleration. Now look at this the I mean we have we have we have deliberately done this, that there is we have mixed a lot of noise. What is the scale here? If you can see two hundred, four hundred, six hundred; while you can see the you can see the amount of noise; measurement noise that we have deliberately put in in in a in a in a two hundred meter position, we have put in hundred meters of noise, right, so huge noise.

And and can you distinguish between the between the estimate and the trues true directory? Hardly, so the so the estimate is producing merely the true directory, rejecting all the noise. So such such high noise is in position, right. And look at the look at look at the velocity estimate which is here, the scale is this this point is can you see the cursor? So this point is nine point five, this point is ten, this point is ten point five. So you can see the velocity estimate. It is varying between nine point five and ten point five, actually it is it is varying less; this is an extreme case, actually what a what a happened is that here I mean full convergence have not taken place but actually it is probably where varying between nine point seven five and and and ten point two five. So that is a kind of error you are having in estimate, can you imagine what would have happened if you if you try to differentiate this position signal?

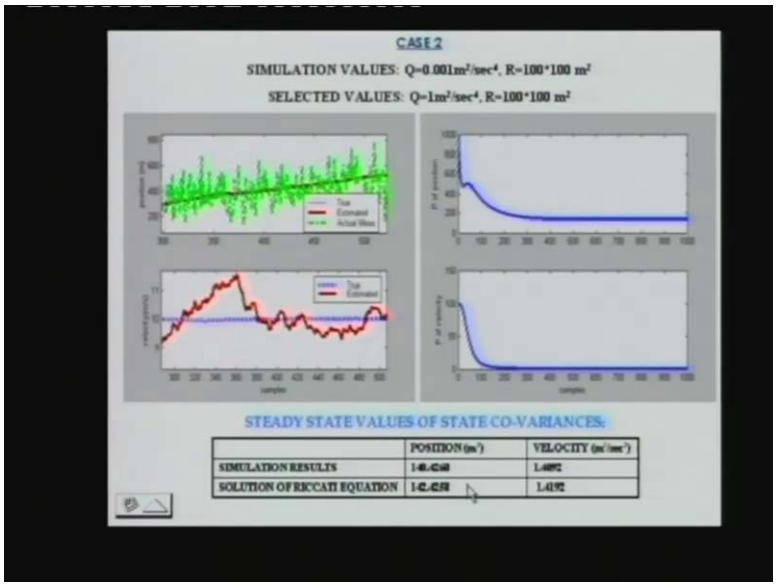
You would have been completely away I mean you would not have got any signal at all; it would have been totally noise, if you differentiate this signal this this this this position measurement directly, where is the cursor? So you can understand what what would have happen to your velocity, okay. Now now let's look at sometimes the cursor is vanishing because this computer is slow. Now here just to show the just to show the asymptotic results; we have plotted the diagonal elements of P, that is with estimation between zero and thousand points; you can see that as these state estimates become I mean better and better the state error covariants sharply falls and then reaches the reaches a steady value. So the P matrix indeed converges, this is an observable system, right control level as well. And so so and the convergence is the note the note the convergence; about three hundred to four hundred it has it has it has mostly converge, it has come to about ninety ninety-five percent of of the steady state value, right.

So this this is important because in some other cases you usually see the convergence will become significant is slower and for good reason, right. So the filter you know the convergence means that the how sure the filter is becoming about the signal properties. So when the filter converges, it has become whatever doubt it had but it it it also you know kinds of kind of learns. So at some point of time it has learnt whatever was there to learn in the data; it cannot reduce uncertainty beyond that, so the faster the convergence the faster the filter learns, okay.

So look at the look at the convergence, look at these figures, so right. If you if you if you solve the Riccati equation actually P equal to xi P; you will get these values twenty-five point one eight, like that. While here it has come to forty point five same scale of things and one experiment which which we did not do is that, this value is actually independent of the value of P zero that we choose. You could have started with the ten times high P zero it will still come to this value, right. And look at the velocity, velocity Riccati position says, point zero zero eight; we have got point zero one five four. So roughly about one point six times, I mean if you really had run it for another two thousand samples probably it would have it would come to point zero zero eight, right.

So now let's see the see the next one. So this is my standard case; where the value used in simulation has been used in the filter, does if the filter knows the true Q and the true R of the data? Right.

(Refer Slide Time: 19:07)



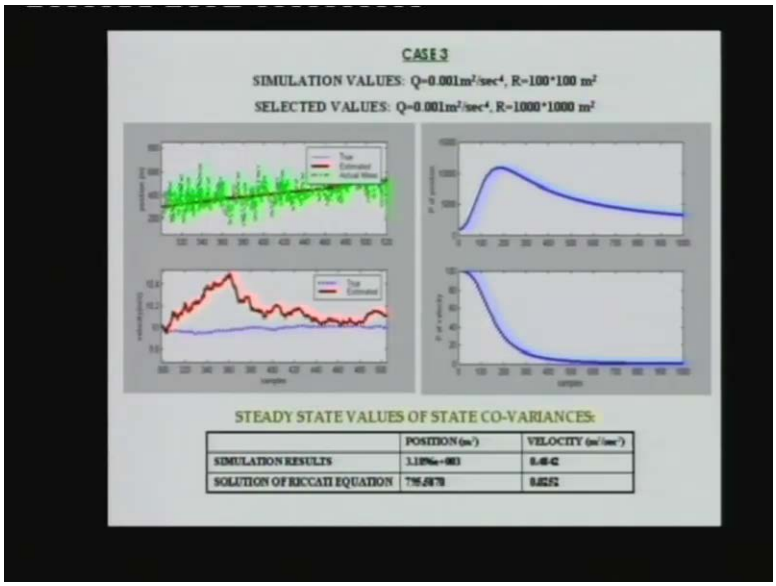
Now, next plot what I have we done here? See what we have done here; what we have done here is that we have we have still use the same data, but now we have told the filter the wrong Q. We have told the filter at much higher value of Q; there is no guarantee that you will know the exact value of Q which which is there in the data, you have to make an estimate. So what happen if you make a grossly wrong estimate? Okay.

This is thousand times Q; so which means that you are about thirty times off in your estimate of the acceleration levels that are coming. See the see the measurement data is not that different, can you see that? So measurement data looks almost like the previous data, why because the measurement data is position; so acceleration gets two times integrated.

So every time you integrate, you are you are actually basically putting a one by omega again. So so power is sharply falling. So the effect of that in much I increase this thing on the on the acceleration level, true true acceleration is is not that high because after all signals are zero. Mean; so when you double integrate, signals do not I mean the position signal is not that much affected but, and the and the and the measurement noise is at the same level, right. So there would be position data looks merely similar, very similar but look at this estimate now; previously remember that that we were between nine point seven five and about ten point two five, now we are this is nine this is this is nine this is eleven. So it it is almost reaching twelve because the filter assures that, there is that the that the velocity should very much; because it has been told that the covariance of the processor of the acceleration is one. So it if the covariance of the acceleration was really one, then it would have a large velocity variation, naturally. So it is interpreting the data in that light, therefore the estimate that it is producing this cursor is vanishing from every now and then.

So so you get a much more varying much more varying estimate; convergence is roughly the same but the steady state values have increased, significantly because the filter assumes that it cannot reduced; if if if the data is so much high Q then velocity and position, you see the Riccati equation values are significantly increase it was fourteen now it has become hundred forty-two. This was point zero zero eight, it has become one point four one. Convergence is pretty fast, so it has merely converged to the steady state value; now let us see the next one.

(Refer Slide Time: 22:15)



In the next case, what we have done is Q you are maintained and R we have increased by ten times, not ten times actually hundred times. So now my measurement uncertainty is of the order of thousand meters, huge. Actually its not in the data, I have told the filter that the that the measurement uncertainty is the is the is this much; I have given a wrong R value to the filter.

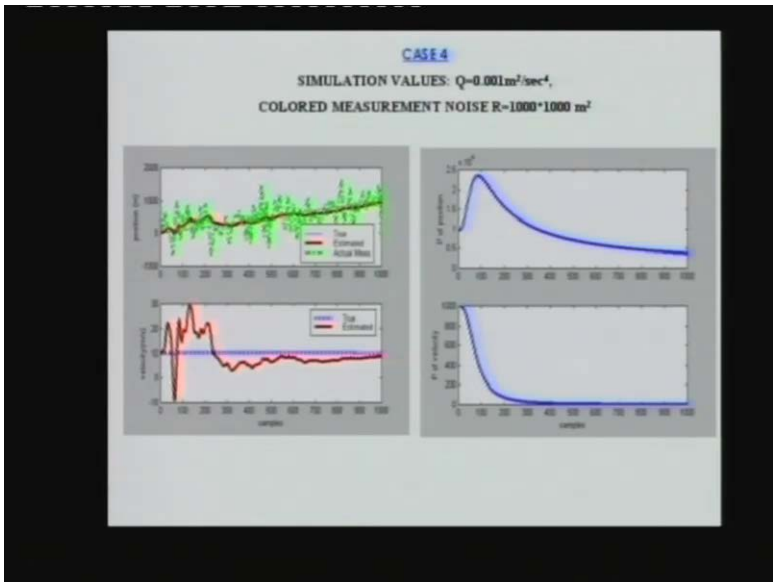
So what is happening? Data is same because data is generated by that. Now you see, first of all look at look at this here you will probably assume that the the mean of the estimate is not matching; now that is not the case actually that is that is that is happening because this is the plot between time instance three hundred to five hundred. Now…. so now now now look at this convergence pattern, why is this mouse stopped?..... Now look at the convergence pattern, now the the now the convergence pattern become has become much slower. Filter does not converge why because the filter is now because it knows that; there is so much error in measurements, see the filter what the eyes on, the filter works on two things. It works on the knowledge that you have given it, values of Q R a b etcetera and the signal it is getting; these are the two things best on which it makes the estimate.

Now this time you are telling it abc, ab r etcetera, this time you are telling it that the that the signal that you are getting is highly uncertain for the same notion; if you perform a second experiment you cannot get a totally difference r, signal that is what you have told the filter because r is so high. So the filter is now much more cautious, it does not want to come to a hasty decision right, because it knows that there is a lot of uncertainty in the data. So it takes time to converts. So the moment you increase r, filter will reduce its gain and will start converging slowly in fact this has happen; because this estimate is not yet converged actually at three hundred, this is far from converged, right.

So so so therefore this is not a I mean you cannot really expect a zero mean. Now the filter is still coming down I mean the especially the effects of three zero etcetera still still getting reduced. And now interestingly look at the look look look at this Riccati equation values; see the position uncertainty steady state has has become from previous it was forty, there it will become one forty, now it is eight hundred. While the while the velocity estimate has not changed, it is still it is it was point zero zero eight in the total ideal case; now it is point zero two, why because the Q is the same. See velocity is not affected by R, velocity is directly position in affected.

So therefore the steady state steady state variance of velocity is not that high, it has not increased previously when it when you gave it a wrong value of Q; it it immediately increase the covariance of the velocity, because Q directly affects velocity, right.

(Refer Slide Time: 26:15)



And then we will see the last result. So so this is the case then, let us see the next one, the last one. Here very interesting thing, that what happens if your noise is coloured? That is measurement noise is actually not white, it is coloured and the filter bandwidth is well within the system band width. And now now in that case, what will happen? What will happen is that; the the filter tries to explain noise you know it finds that, the if the if the noise band width is largely contain in the system bandwidth, then part of the noise can be explain by the system behaviour.

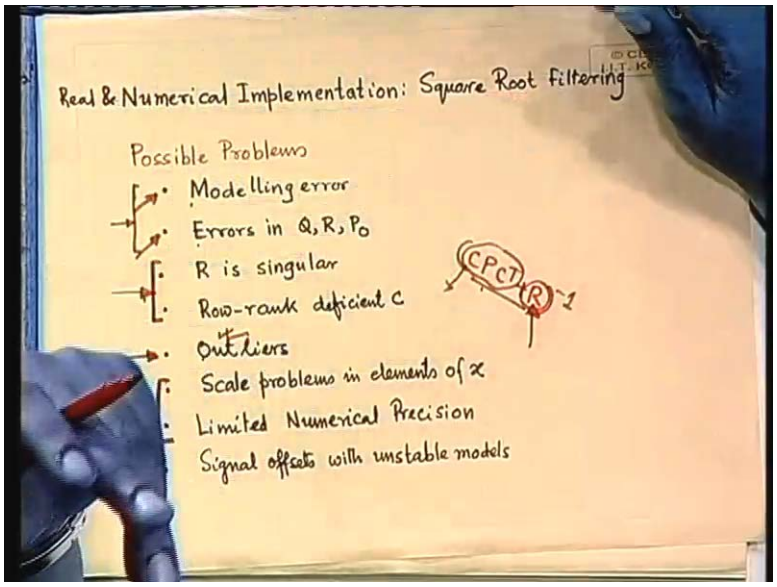
See if noise is white then and if the system bandwidth is very low, then you cannot really knowing put on the no input to the system can really generate that noise; because the noise is white, much of its power lies well well outside the bandwidth of the system. So the so the system whatever input you give it can never generate that sequence which you measured, but if you have noise bandwidth is low; suppose system is also zero to two hertz, noise is also zero to two hertz then the system can explain a lot of noise because there will exist inputs which can produce sequences which has very close to the noisy sequence because the bandwidths are same. So what you are doing now is that you are actually by by coloured by basically you you you generate white measurement noise and then you pass it through a low pass filter.

So the colour measurement noise bandwidth is largely within the system bandwidth. So now the system starts explaining noise; that is the estimate will try to interrupt will try to include the effect of noise. See this scale is this scale is significantly higher again the cursor is gone. This this scale significantly higher, this is zero, this is minus thousand and the top one is plus thousand, previously it was two hundred, four hundred, six hundred.

So the scale is much higher still this estimate is now see when whenever whenever the average value of noise is going up, it is going positive; whenever it will go down, it will go negative. See here the average value of noise is significantly positive, can you see that? That the green lines, if you take an average within that interval, it will become significantly positive and estimate is going on, right. Plus there is correlation; so so so so this so this so the estimate is need not trying to track the noise dizzy, I mean low frequency components.

So if you, this is why you should add in this case, you should actually add the noise states also to the Kalman filter extend the states. And then reformulate the problem such that the noises states will will will also be estimated; then your actual system state estimation will become whiten flatting it, ok. So again you know convergences are still slow because our R is still very high. So this is what I wanted to show about Kalman filter, you can yourself do many experiments with your own code and see better things. So we come back to our notes.

(Refer Slide Time: 29:55)



Now just very briefly, I wanted to discuss that; if you really want to make a numerical implementation of a Kalman filter then there can be there can be several problems and there are several solutions. One very normally people very much very highly recommended implementation of the Kalman filter is this, what is known as the square root filter. So we will just see, what is the philosophy of this square root filter nothing else.

So the so the among the possible problems of, you know not getting filter performance is modelling error; if you are a b c d matrix is the not correct then your your estimate will be great. Sometimes it can even be, it it sometimes it can even diverge especially the if the dynamics is unstable and if you are start having modelling errors. Similarly obviously this is also kind of you know basically error in the input, that you have given Q R P zero. We had seen what happens, if you have errors in Q R and P zero filter will become slow or unnecessarily, right.

So this show shows that incidentally in all in our kinematic model, at least there was no chance of having a modelling error because after all velocity is after all the integration of acceleration; that for that you do not need to model anything it is already model but in the other cases, you if you want good estimates you should spend time. See remember that, algorithm is can never replace experimentation in engineering, experimentation always has

has a very distinct role. So so do not think that I mean just by putting or having having a Kalman filter, you can you can I mean see it in a nice air condition room and get fantastic estimates; that is not going to happen you have to go out there in the field, measure things, model then get data then only you will get good estimate, sitting with the computer otherwise, not.

So so there is need for consider very good modelling and measure I mean estimation of Q R and P 0; if you are do get good estimate, good things. Sometimes what happens is the these two things together, see if you remember the Kalman filter has a term called CPC transpose plus R inverse; so wherever you have inverse there is caution you know is it invertible. So sometimes suddenly you are you are you are running the filter and suddenly it should, if it if it at some point it become numerically singular, the estimate will explored, right. And I mean just imagine that, you are I mean you are in mid-air I mean flying some object which which which constantly works on your target and then in in mid-air you are your Kalman filter has become singular, what a situation to be in, right.

So so this can become singular, why? If the conditions of these deteriorate; so if R is merely singular, or if CPC transpose is singular now. CPC transpose will be singular typically when c is, it will definitely be singular; if C does not have full row rank where is all the rows of C are not independent may be using. So which means that, you are using a third measurement which is really redundant because the third measurement; you can always generate by taking suitable sums of the other two measurements. In that case, you should delete the third measurement, not using because this might lead to singularities.

So, these two can give you numerical problems of non-inevitability of matrices, right. There are there are several solutions, that is check sometimes add a constant non-singular matrix, like that. Sometimes you know there are there you may get outliers, outliers means that you have generally observe that Q and R or I mean xi and eta are random, they have the certain certain kinds of Q and R but sometimes you get for some reason sometimes you get some isolated data points which are suddenly will give you a peaky disturbance, you know glint kind of things. So if you if if you are not aware of those and if you do not know handle them, tackle them with that; first of all see whether you are having such samples, if you have some such samples it is it is better not to update best and those sample because they they do not

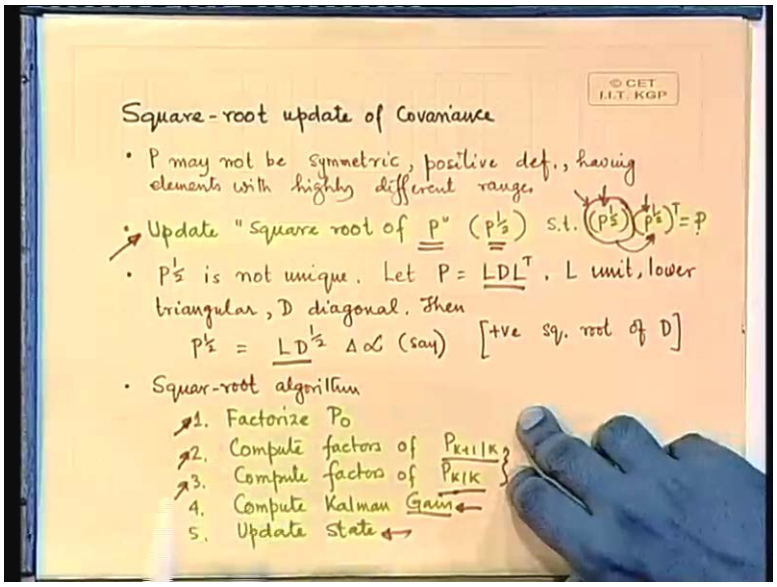
really belong from the probability distribution which you have assume for the filter. They are very sporadic signals which actually belongs to a totally different probability distribution.

See if you if you update based on these outliers, what will happen? Your estimates will suddenly get a big shock, I mean you are arriving at closer through the estimator, suddenly one one data point comes and it shakes your estimate up; because the Kalman filter suddenly finds a very bad data point and it will it will considerably change its estimate. So so so these things should be handled. Then there are sometimes you know numerical problems can come due to various reasons. One reason is the another reason is is due to scaling problem, due basically due to rounding and you know truncation errors; and that can happen if you have you know variables, for examples which are say one variable is of let us thousand meters, that kind of values, another variable is I mean I mean basically various components of a state, another component is point zero zero one millimetre per second square.

So if you have such things then you can imagine that, if you take there covariant is the P matrix, some elements will be ten to the power eight, some elements will be point one. If you want to manipulate such matrices due to due to numerical precision, your your your calculations will will not be will be significantly away from the ideal calculations. This is this itself by itself is a is a signs and requires of very complicated analysis to actually to find out the why it happens, when it happens, how to avoid it. But this can give you considerable problem relating to I mean leading to divergences, especially the see the P matrix is supposed to be at each stage of computation of the Kalman filter, the the the P matrix is supposed to be symmetric and positive definite. It can very well happen, that C is either not symmetric or not positive definite because of these cases. If the C matrix is not not symmetric or nor positive definite, you are your estimates will go as straight. See the health of the P matrix is of critical importance in actually implementation of the Kalman filters.

So there is one algorithm which is widely used which which is which by itself ensures, that the B matrix is always symmetric and positive definite; whatever whatever the precision. It is it is built in in the algorithm and and and that is called, square root filtering. So the whole idea is that, is that do not update P.

(Refer Slide Time: 37:33)



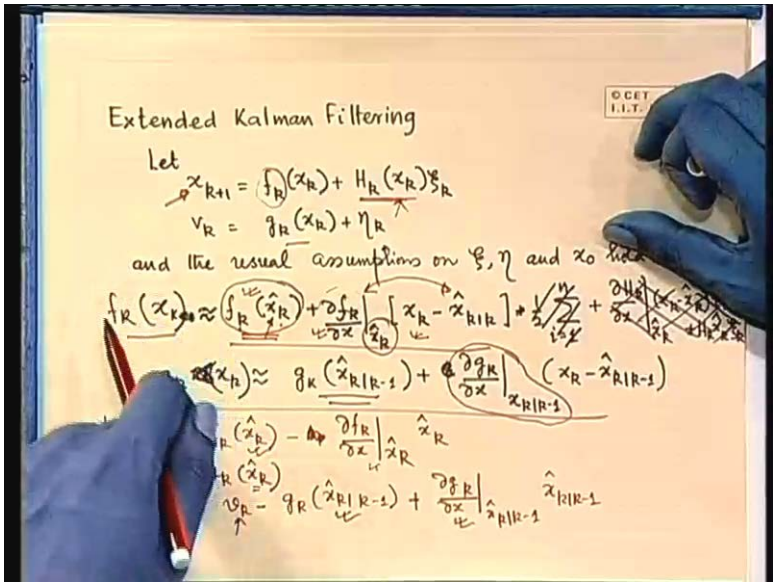
See in the Kalman filter be update P but do not update P, but other update this square root of P; by by this square root of P, I mean such a matrix at that P into P to the power half, that is what I call this square root of P. P to the power half into P to the power half P P, wherever I need P, I actually calculate it out of this, but I update this. So you see by construction it is symmetric and positive definite; it it cannot be otherwise. It cannot become I mean otherwise due to updating; because I am updating this and then whenever I am needing P, I am I am just multiplying a matrix by its transpose.

So it must be positive definite, must be symmetric because because whatever transition will take place; will take place equally in the symmetry elements, so the matrix will be an symmetric. And then you can I mean actually be the the square root of this is not uniquely define. So you can choose one one LDL transpose factorization or some other factorization Q R various kinds of things. So the whole approach is that, you first factorize P0, that is take compute P zero factorize it into P zero to the power half and P zero to the power half transpose, and then go on I mean have a update laws which which which do not update; you have to compute P K plus 1 given K rather than doing that update its square root.

Similarly update this square root of P, P K to the power K and then best once you have this these these square root of P K given K; then you then you construct P K given K and then

you construct Kalman gain, because Kalman gain will require P K given K. So while updating you updates these square roots then while calculating Kalman gain use it using this equation and then updates it. So this avoids many this this algorithms is now fairly stabilize, discovered by one, I mean various people developed it be a matter importer.

(Refer Slide Time: 39:44)



So now it is very stable and well use implementation. To end, we want to just have a look at extended Kalman filtering; for two reasons firstly because a large number of estimation problem. Especially in aerospace, often times out to be non-linear because of the fact that, you see we obtain a very nice linear model; because we assume that our measurement is in Cartesian coordinate. If we are measurement was in polar coordinates as happens often with radon, our our all our equations; theta and R would have been nonlinear differential equation not linear.

So so I mean how do you estimate state? It is that is how do you estimate R and theta? So some some some good solutions are needed, for because they are because this is a this is a very common thing; that a estimating I mean one of the biggest applications of Kalman filter is in target tracky, especially in especially in aerospace applications. So you see that, there is a there is simple formulation available by which you can update I mean do a do a Kalman filter, like it is an it is an it is an approximate Kalman filtering for our linearized model of a non-linear system, so at least you can do something.

So we are assuming now, that we have a non-linear system given like this. This is now not not a not a matrix, but a but a vector function, non-linear function. This is also a so you see that, the b matrix now a function of x. So it itself is not linear, similarly this is not linear, this is that measurement noise. So how do you linearize? The basic idea of linearization of any non-linear system, is to break it into a tellers series approximation. I I am sure you are you are familiar with linearization of non-linear system.

So you so you break break up this into its tellers series, break up this into teller series, F and G. And now you see, what is happening. In this this term is known, can you see this? That is this term is known, what we are doing is we are trying to linearize the actual model around by last estimate. So I have obtained an estimate x at k given k at any point, true state is somewhere close I do not know where. So I am trying to find out the I am trying to find the linear approximation of the true state around my estimate. So the see normally you linearize around x not u not. So you linearize about an operating point, in this case the operating point is my estimate.

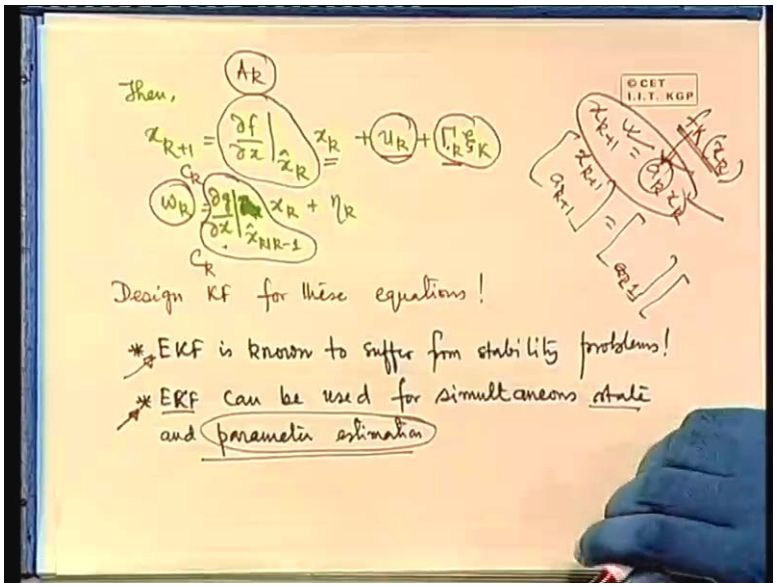
So obviously I am assuming that, my that my estimate is you know kind of good enough and the and the true state indeed lies reasonably close; if it does not lie reasonably close, this approximation is not good. So I have a you know kind of the the the into it you know, it is like you know putting the card before the horse in a sense that; the the before I mean while formulating the estimation algorithm, I am kind of assuming that it will give me a reasonably good estimate, and x k is going to be close to my x hat K. You know this is a chicken and egg problem which which play exists algorithm and it is typically very difficult to prove, it is stability. So what I am doing is that, now is it this is an this is now known because my x k is a known function; may be a non-linear but known. X hat k I have already computed, so this is a pre-computable thing. Similarly this is a pre-computable thing, this is not known, this is known, this is del f k, del x k this is the functional form; which is evaluated again at x hat k, it is it is computable.

Similarly this is computable, this is computable. So these are all at, when you are standing at x hat k these functions are all computable. So anything which is known is like an is like an known input. So now I am redefining let let u k b is equal to this into this. So you see the known two terms; this term and this this term. I am assuming to be a known input because

they are computable, I know them beforehand. Similarly if I know x hat k then then I know gamma k, this term. Similarly if you know if you define a new measurement which is the measurement minus two known quantities; you can always define it as a new measurement because it is computable. This is your measurement, this is computable, this is computable, so you can always define a new measurement.

So now… once you have done that, this you can write this equation in terms of these and it will be a linear Kalman filter that is the whole idea.

(Refer Slide Time: 44:51)



So the whole idea is now, if you write it in terms of these u and gamma and this is and these is this w; now it will be become like the known Kalman filter, I mean linear Kalman filter. So so this is like an like your matrix A k, this is this is x k, this is your known matrix u k, this is gamma k xi, k xi, k is unknown, gamma k is known and this is like c, c k. So in the vicinity of x hat k the the the the this system dynamics behaves like this linear equation.

So around x hat k, you can now at least for one step you can have a linear Kalman filter formulation of this, then you will get a new x hat k plus 1; you will have to again linearize the system around that operating point, find this matrices again new matrices, formulated new Kalman filter around that operating point and do one more step update like this you do. So so

every time you go, you find a new system and you find a new Kalman filter and you make a new update.

So this is how you go when you have a non-linear system, but at least locally for that update is optimum, right. So this is called extended Kalman filtering and it is; it works it does work lot of people have used it, but it sometimes suffer from known to suffer from stability problems. You know mainly because of the fact that is a you know, you are right in the beginning you are sort of assuming; that the the estimate is is is close to the true state.

So it will depend a lot on initialization, if you are starting your estimate very far away then this then this approximate model is very bad. And if it is very bad and if you if you make a one-step update using your Kalman filter based on this model, it might it might take your estimate in a wrong direction, because you are using a very bad model. So it it has a you know Kalman I mean, E K F has a kind of a locking problem; in a some I mean it sometimes cannot lock on to the estimate, once you can lock on it can go on, right.

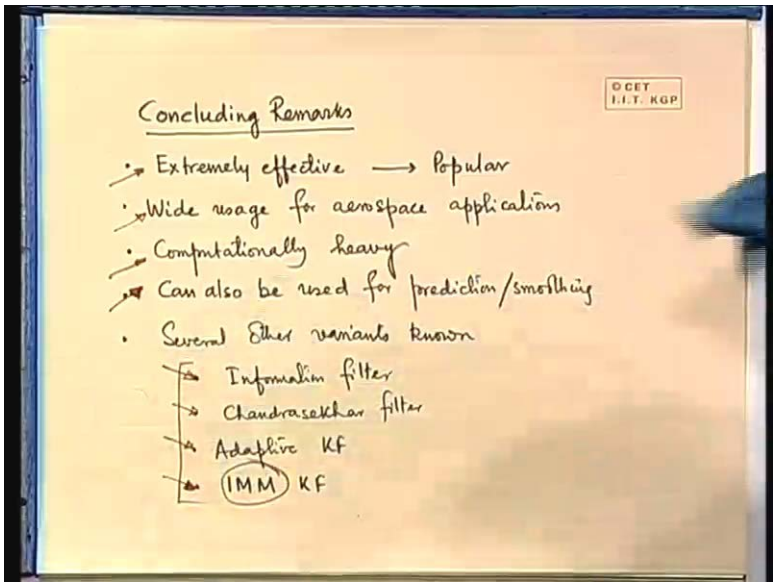
So sometimes you need good initialization for this for this kind of algorithms. But it can be, I mean it is in fact it is a nice point to end our Kalman filtering discussion; because the E K F can be used it has been used for simultaneous state and parameter estimation problems. This is suppose; suppose you are given a system in which you have to estimate the state but you do not know its model or suppose its model keeps on changing. So then what do you do? So then you have to simultaneously estimate the state and the model parameters. So if you want to do that and an obviously model, for example; x k plus 1 is equal to a k x k is linear if a k is known, is completely non-linear, if a k x k are both are un-known because you are you are having product of two unknowns, so it is a non-linear model.

So in such cases you can use the Kalman, you can use the extended Kalman filter and write that; let my states be x k plus 1 a k plus1 then I write that this is, this will be what? Then how do I put it? a k plus 1 you can write as a k equal to equal to a k. So this will be probably be 1. And x k plus 1 will be what in this case? How can we make it a k x k? We cannot make it ask but uh theta problem; we have to we have to it is it is in non-linear but then we have to, yes this is a this is a non-linear system. This will become and the the function f k, which is also a

function of x k. So then so now we have to actually actually we cannot directly find the linear equation, obviously. It is a it is a non-linear equation which for which we have to now formulate the formulate the E K F, but this can be solved using E K F; and this is some sometimes called adapted state estimation, that is when you are trying to estimate the state when you do not know a system.

So we will stop here and incidentally, we have I mean because we have ended with a problem of parameter estimation and in the next module, we are going to that is going to be our subject; in that case we are not interested in state estimation, signals will be known and we will be rather interested to to estimate the system model. So that is system identification or system parameter estimation which will be the next module and the subject of next six to eight lectures. Just one line conclusion on Kalman filter, very effective and popular wide usage for aerospace applications, many other applications; Navy be I mean also in navy mainly for strategic applications.

(Refer Slide Time: 50:25)



Computationally is not a very simply filter, that is why sometimes people use fully knowing that it is not optimal; even for target tracking people people use constant filters known as alpha beta filters or alpha beta gamma filter which are kindly ingredient filter, very simple to compute do not I mean does not give such good results.

Incidentally filtering is not the only problem, that you can define you can also define a prediction problem where you are getting measurements up to j up to k and you are interested in knowing the state at k plus nine, k plus ten in future that is called a prediction problem. We we we only talked about one step prediction, you can do multi step prediction; very effective in in many situation including chemical process control. Sometimes you may like to do smoothy, sometimes you have the full measurement available; offline thing you are doing you have a you have the all the data points available from zero to thousand, so using zero to thousand you can try to estimate x to, x two hundred. So you are not only using past data you are also using future data because the application is offline, right.

So such such cases are called smoothy where using data up to k, you are trying to estimate may be x k minus twenty-five, right. So there there are also formulations of the of the Kalman filter type smoothing problem; you did not consider and there are several variants of the Kalman filter this is not the E K F is not the only one, there is there are filters called information filter, Chandrasekhar filter. There are filters called adaptive Kalman filter where Q and R, are I mean I mean along with the state estimates Q and R are continuously tuned.

There are filters where you do not know what I mean I mean, one one one kind of you know approximation of this will be that; you have three four Kalman filters tuned for three four different use, and and and at any times you use only one of them. So you have all the four filters running and you sometimes switch from here to there, sometimes switch this, sometimes switch that, so you have what is known as an interactive multiple models Kalman filter.

So these are typically use for various kind of things especially for manuring targets, okay. Suppose the suppose the pilot you know, I mean if you are trying to design a let us say guidance system for a for a missile for a aircraft; pilot is not going to sit pretty I mean knowing that missile is chasing it. So it will also detect, it is a it is a cat and mouse game. So it will also know that of if the missile is chasing, so it will do manuring, it will turn right, left, up, go up, down, accelerate, decelerate and evade the missile. So the from point of view of the missile, you have a target whose whose acceleration level will be unknown and which will be suddenly changing.

So if you want to estimate it’s targets, you have to use different filter; you cannot get very good results using the same filter with with one Q. So for such situations you sometime use this this I mean interacting multiple model. So that is all for today, from the next class we will start system identification. Thank you very much.