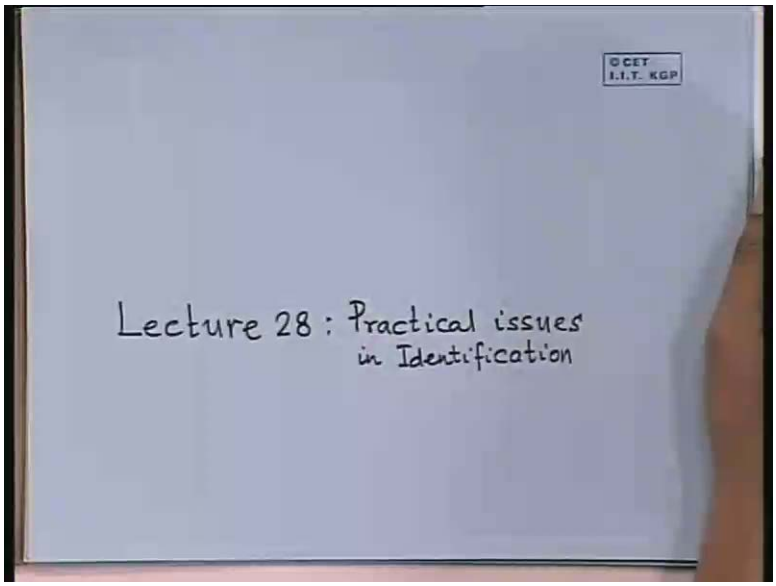
Estimation of Signals and Systems
Prof. S. Mukhopadhyay
Department of Electrical Engineering
Indian Institute of Technology, Kharagpur

Lecture - 28
Practical Issues in Identification

So, we have come to the end of the system identification module, and we will end the whole whole course in a in about two more lectures.

(Refer Slide Time: 00:54)

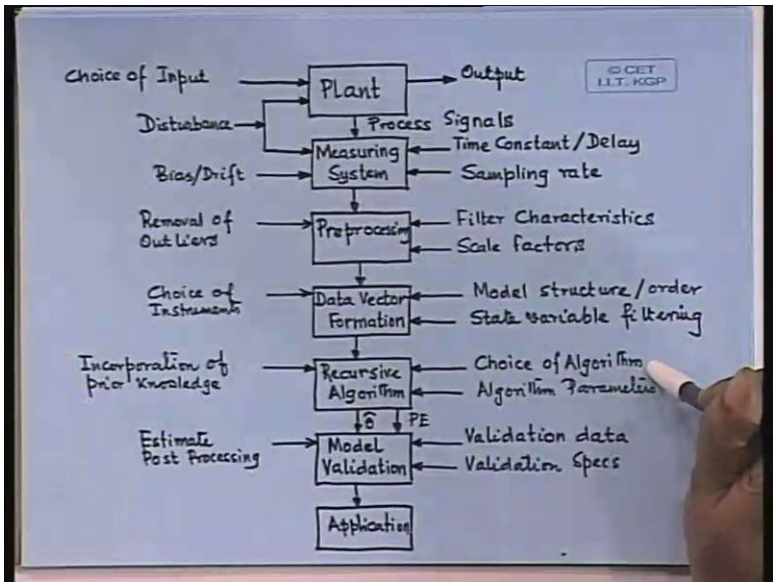


Today, we will discuss what what I have written here as practical issues in identification. So, we we hopefully we want to show that; you know people have a sometimes because academic courses tend to concentrate on algorithms, people think that the algorithms are everything but that is not correct, there are lots and lots of other issues, before you before and after you learn the algorithms.

So, all of that have to be compressed in one day's lecture; so we will again talk somewhat philosophically or take a common sense approach and and see what kind of; look at

identification as whole and try to see what other things are involves, that you actually get models which are useful to you. So that is why I thought it is a nice may be its good to look at the identification process as a whole.

(Refer Slide Time: 02:16)



So, here is here you are here is the here is the gentleman who whom we want to study, the plant. So, you have and here on both sides; I have written various issues that are involved in getting the model, and they are involved where they involved at different stages of getting the model. So these are the, these boxes indicate the stages of getting of getting the model. So the first of all you must understand that, the plant is not the only thing, there is this measuring system.

So you you are you are you are working with the measuring system; and essentially you must remember, that you are building a model between the between the measurements, you might think that you are building a model between this and this, you are not doing that. You are building a model between, what you measure where you what you measure and where you measure this and between where what you measure and where you measure this, not exactly between these two points, right. And and there could be so this measuring system itself needs to

be needs to be understood; because this measuring system is included real model, it it it first it is included and secondly it affects the modeling.

So you need to very carefully understand that and you need to understand; what kind of time constants or delay it involves, what is the what kind of sampling rate it uses, what are the biases drifts which which effect it. Then after you got the after you got the data out of the measuring system, you know you kind of came to the computer. So this is like the data accusation system and then you came to the computer. So from here the the computer starts, you you got those numbers; whatever they are right or wrong then before using them in your algorithm.

You do not blindly plug them into the algorithm; you do a lot of processing before you use them into the algorithm, so that you get good results right. And you you typically like to filter things naturally just because you you can you can model noise; does not means you are going to let in a lot of noise, you will cut out as much noise as you can. So what sort of filters to be used? Sometimes, you use you have to use you know scale factors.

I do not know whether you have ever tried to try a neural network; in a neural network training, if you if you if you do not use scaled data all of which are you know roughly in the same ranges, if you use one data which you know spans between zero and ten thousand and I mean another data which spans between zero and zero point one, you can it it will be very difficult to to train the network because of the interconnectivity. So no set of weights can actually handle such kinds of data, which which I mean whose ranges are widely apart.

So, you need to scale data to bring them roughly in the same ranges; do your estimation and then get back to invert that scaling map, to actually get back to the realistic values, after you estimated, right. So you need to do a lot of scale factors, you you need to do that. After that, you have to form; we have the kind of algorithms we have discussed are linear or pseudo linear regression methods, so all of that involves forming what is known as the data vector or a or a or a regressor. So you need so before you can apply a parameter update; you need to compute the regressor and that involves lot of things.

For example, it it involves choosing your model structure, choosing the order of the model; even some I mean in many cases, even after this preprocessing you you would like to do further filtering as we will see, that the the same transfer function let's say g s can be written in various ways using a phi transpose theta formulation, various various ways are possible. So to basically what exactly what kind of signals, you are you are you are using the regressor depends on what kind of state variable filtering you use. This is a this is a new term, so we will we will see what it is and obviously if you are using instrumental variables then you need to have a choice of instruments at this time; because the because the instrumental variable forms the other data vector, that is an instrument vector which you are also going to use in your algorithm.

After having them that, you come to the actual algorithm which goes you know three or four equations which will update; which will give theta hat k from theta hat k minus one. So there you need to have you need to choose your algorithm, you need every algorithm will involve certain parameters. For example, if you are using exponential; for getting you need to choose what is what is going to be the value of lambda. So, you need to choice choose the algorithm parameters. And you also need to see to what extent you can incorporate prior knowledge.

If you know, that the algorithm is stable or if you know that its dominant time constant is not more than ten seconds; if you if you if you happen to know this, you must use this because if at any point of time your algorithm throws up the time constant which is more than ten seconds, you do not need you you do not update that because you are sure that it is not more than ten seconds. So you need to know, how we you can you can incorporate the existing knowledge about model, that you have based on prior experience, whatever.

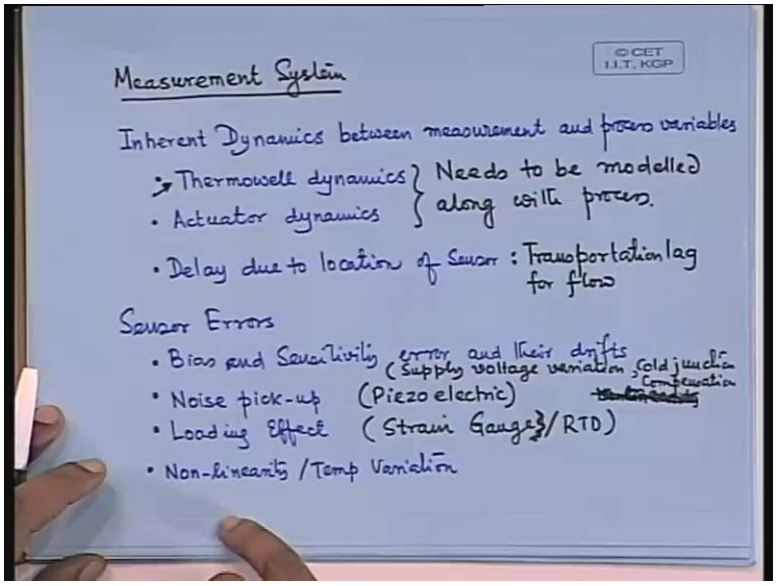
This, so having done that that throws you two things; the algorithms gives you two things, first it gives you the theta hat and second thing it gives you the prediction error. Both are important as you have seen for for you know; checking whether the whether the whether the model is right, whether the estimation has been okay and to do certain kinds of post processing, you you may not even like to you may not even like to use this theta hat directly in your application, you you may like to do do what further. For example, you may like to check whether there has been any

common pole zero, if so cancel it. Why have common pole zeroes? Common means, you will never get exact match again; whether whether you are getting pole zero which are very very close, right.

So, you need to you might like to do model reduction, you might like to do filtering not only on the data; see all these while you have been doing filtering on the data sequences, y case and u case, you might like to do a filtering over theta because theta is continuously getting because of noise but in any given case you know that the parameter cannot getter like that. So, after you have estimated the parameter, you might like to filter it. So so such post processing may be needed before you finally handed over to the application for use.

So, there are so many steps which involved in successfully estimating a model which will serve the purpose of the application. So, we will now look at these in sequence.

(Refer Slide Time: 10:13)



Coming to the measurement system, as I said that you must realize that, you are estimating the model between measured variables; say anything which is being introduced into that is coming any way. So, that is not your system model; you must remember that, that is not exactly your

system model, if you if you are if your measurement system is very good, it will definitely approximate the system model otherwise you are using bad sensors, right. For example, common example is when if you are doing temperature and measurement; you know thermocouples do not have much time constant but thermo-wells could have. Thermocouples are not; Thermocouples, have you seen Thermocouples? They are just bare wires, you do not put bare wires into a in to a in to a furnaces or in to a chamber. You typically put a Thermocouple, you typically have a cover on it, sheet kind of a well on which you put the Thermocouple.

It is that well which could put a significant time constant between the temperature change; which is actually occurring in the furnaces and the temperature change, that you are measuring. So it might introduce an it might introduce a non-trivial time constant and if you are estimating; what are you estimating? You are estimating the the the model between the measured Thermocouple voltage and the and whatever; may be some heat flow, steam flow rate. So what, so obviously the Thermo-well time constant is going to come into your model.

So, you must think of that and you must eliminate that; if you really want to know that how how fast the temperature inside the furnace rises, if I suddenly jack up my steam flow rate then you need to eliminate the time constant, right. Similarly, you will have actuator dynamics, if you are using the u k values which are there in your computer and think that that actually that is going to the plant; it is not going to the plant, it is actually going through the actuator. So, there is an there there could be there could be significant actuator dynamics and when you are estimating what you are thinking as u k is not really u k.

So again you need to strip off the the real actuated dynamics, to know what is going to the plant, right. Similarly, there could be delays due to the location of sensor. You do not know where is your sensor is put in a phi; I mean typically in especially in process control applications, I mean lot of time the the input is the flow variable which turns out that, if you want to change pressure, you use flow, If you want to use if you want to change temperature you use flow, if you want to use if you want to change level, you use flow. So, flow is a lot of time the a very dominant input variable.

Now, now if you what flow you are measuring; very much depends on where you put your flow sensor, how far it is from the from the from the process; it is a pipe right. So, you are going to put it somewhere and that is going to put a delay between the input and the output. And not only that that delay is going to be going to be different at different operating times; after all what is the delay, the delay itself depends on the flow rate and the physical distance. So, it really depends on what is the flow rate that you are using; if you are using a slow flow rate, low flow rate then you are then you are encountered the long delay. If you are using a if you are sending a fluid very fast, you are going to have a small delay, right.

So such things will come in due to the measurements; inherent dynamics between the measurement and the process variables both on the input side and on the output side. Similarly, you could have various kinds of sensor; you know inaccuracies errors, non-idealities. For example, there could be bias and sensitivity error; most most of the instrumentation guys you know, spent lot of time studying the wedge stone bridge, lot of senses used wedge stone bridge Arcades, Strain gauges.

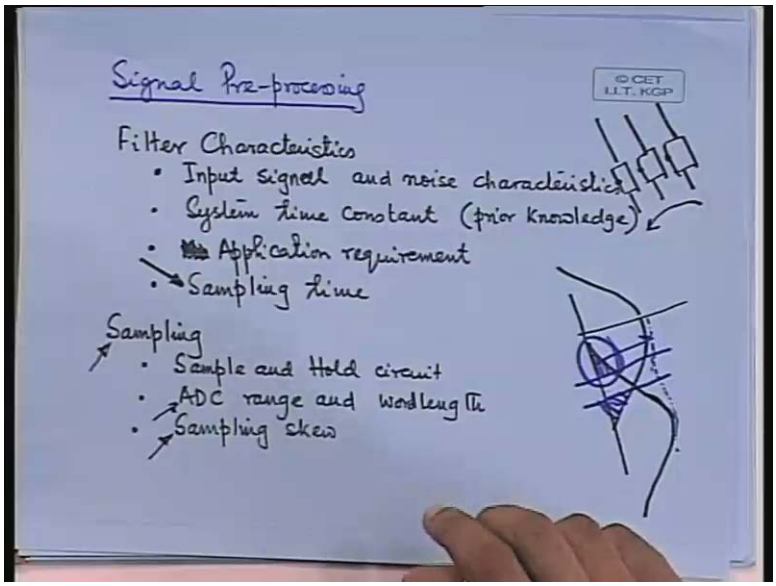
What is the what is what is the biggest source of error on on the on the wedge stone bridge? It is the it is the supply voltage variation. If you, if the if the bridge supply changes little bit; you have a completely different sensitivity and if you do not nullify that then you are basically, your sensor again changes, right. And if you are using a constant gain value then you are off. So there could be bias and sensitivity errors; bias, one of the good very good examples is cold junction compensation, right.

So, if you if you if you have problem in cold junction compensation circuit then you then you have a bias in your temperature, for any thermocouple measurement. Similarly, there could be noise pickups; there could be significant noise pickups. I do not know whether you ever done this but you take a you take a Piezo electric sensor; if you have disturbance and if your shield connection breaks, you have huge noise on the sensor, any high impedance point will tend to pick up noise in any circuit.

So, you could have noise pickups because of welding's going on, because of not well maintained d c motors running and being they may be brushed, spark offs all kinds of things or you know in the I mean heating furnaces; things which are electrical furnaces, they they you know current electromagnetic interference is always caused by current, because flux cannot be created unless you have current, it cannot be created by voltage, right. Magnetic field is created by current. So any high current device, you have is going to create electromagnetic interference, so you you need to think of that.

Loading effects sensors; sensor readings can change significantly due to loading effects. And there can be non-linearity or there can be ambient variations. You know temperature variation; typically instrument characteristics depend on temperature variation. So you need, you must be aware, that the data you are using the sensor fellow; might prep in this kind of errors, so you would like to correct for them, before using them in the, you must correct for them in pre-processing, so so so first of all you must know them.

(Refer Slide Time: 17:09)



So when you are doing signal preprocessing, you have to you you you will do some kind of filtering, right. First, you do so typically process control signals are one one one one good thing

about process control signals; that they are very slow, they are sub one hand signals in generally. And the kind of noises, that you come in being mostly being electrical in nature; they are they are high frequency. So, it is relatively easier to separate that noise out using filter. So you need to know the input signal and the noise characteristics before you can design your filter. You also need to know the system time constant; because because you do not want to alter, you want to keep you want to keep that part of the signal which is within the system bandwidth, you do not want to keep that part of the signal which is outside the bandwidth, right.

Typically, suppose if you are if you are considering let us say; a very large motor motor let's say Thyristor driven motor. Now, the voltage at the terminals are varying at very high speed, they are because they are Thyristor driven or because they are semiconductors which driven they could be at kilo hertz level, but the but the system time constant is is guided by inertia of the motor. So so obviously when you are trying to model speed variations; you do not need to consider, I mean your your your your model bandwidth does not cross some hertz in the systemic electro-mechanical. So, you do not have to really model it up to ten kilo hertz; even if you are giving ten kilo hertz inputs, strictly speaking, right.

So, so actually you can you use a much you can use a much lower time constant filter; even if the input is different, right. So, you really do not need to use the ten kilo hertz filter because the ten kilo hertz filter is going to is going to pass all kinds of measurement noise; which will be in the sensor, which could be in the scheme sensor, right. Similarly, there could be there could be for example; sampling time design. Sampling time design depends on what? Sampling time design depends on the, again it depends on the model bandwidth on the on the expected system bandwidth; why? Because, if you have you know if you if you know the aliasing phenomenon; then if you have a high if you have if you have chosen your sampling frequency is too small, what is the problem with aliasing?

Suppose, you have a system band system bandwidth which is like this and not only that; now if you if you if you sample it at lower than I q straits, I think you are familiar with I q straits? Then what what you are going to have is that the sample signal will have will have spectrum like this,

right. So, now now if you so what is the problem? So so the so the problem is not that so the in general; in any in general in especially in control kind of applications, it is it is not the high frequency in in accuracy of the model, that's of great concern but if you have low frequency in accuracy in the model, that is of much higher concern.

So, what will happen? So, what is the what what is the overall spectrum of this? The overall spectrum of this is actually it is like this because the these these two things add up. So, you can see that; this part this error, this error is not so important because they are finally going to put a filter. So, so this error is not so important, what is important is this error; because the high frequency error of the next band gets folded back onto the low frequencies zone.

So, so now you will have the low frequencies errors which is not good; so so therefore even before you design your sampling time, you you need to remember that its its a it is a kind of chicken and the egg problem. After all you are trying to determine the model itself, but to be able to determine that effectively, you need to know certain things about the model; that is why it is so important to have a priori knowledge and you might like to take up exercises, just to give enough priori knowledge, so that you can design the successful experiments, so that you can get a detailed one. So, you need to do is iteratively, right. It is it is not that nicely, you just when they are put your data actuation, got your data, came home and then you got a model, it cannot be done like that.

Similarly, if you if you are if you are trying to trying to design your sampler then there are certain things; that you need to think about. For example, you need to think about your sampling skew; typically it may happen, that your that your sampler is sampling channel one, channel two, channel three, channel four. So, actually speaking you are sampling channel one at time t-one and sampling channel two at time t plus delta; strictly speaking whether that delta is going to matter or not is you have to decide but it is not at the same instant, unless it could be at same instant unless you use, what is known as simultaneous sample and hold circuit.

If you use the simultaneous sample and hold circuit then the then you know you know; I do not know whether you have seen a seen a seen a sample and hold amplifier, but the sample and hold amplifier has two inputs and if you give a sample signal, then it will start tracking the signal. The movement you give the hold, that value will be held. So, the so the idea is that, you have several sample and hold amplifiers on several channels which have all of which have a common hold signal.

So, you hold them simultaneously; so they become held at one point of time and then you read them sequentially. So so so even if you are reading them one by one after the other, the the values that you have that you are reading have been held at one instant of time. Now, all that is all that may not be important at all; especially for for example, for process control applications it makes you any convertor sampling times of the order of micro seconds, unless. So microsecond is not going to matter, if you are trying to sample a try to sample fifty temperature sequence in all likelihood, it is so slow.

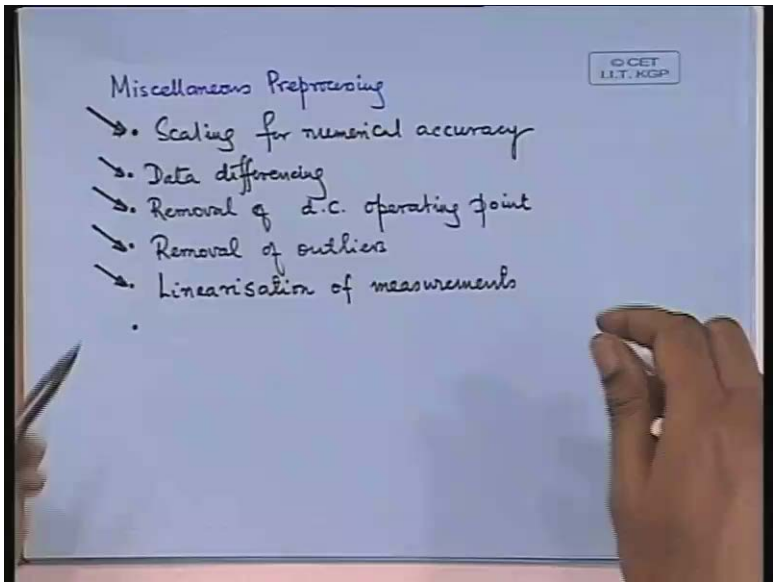
So, it is not that you are going to need it all the time, but you might that depends on your applications, right. So you have to be aware of it. Similarly, A D C range and word length remember that; if you are using an eight bit A D convertor then you are already introducing quantization maths, just by quantization process. So, you have to decide whether you can tolerate that noise and and and what kind of A D convertors; you can actually use but but then you have to remember, you know some people will make this mistake for example, have you ever worried I mean have you ever wondered, why in the market there are there are no there are there are sixty-four bit processors, hundred and twenty bit processors? Why they are not, why there are not sixty-four bit A D convertors in the market?

You can hardly find the A D convertor beyond sixteen beats, eighteen bits are extremely rare; generally it will be using it will be used ten bit to twelve bit, why? What stops it? After all if you know the A D convertor circuit; you could add stages, thirty-two stages why not? Why not because if you because you cannot; see if you have if you have a fivefold reference and if you have sixteen bit convertor then you have sixty-five hundred, five hundred and thirty-five levels by which you are dividing five folds.

So, then one level is what, how many volts? of the order of micro volts. Now, the point is that, if you if you are if you are environmental noise is of that level; then then then what is the what is the significance of the large bit? There is no large bit, there is no significance; because it will if you really see it on a on a on a L C D, you will see the last bit always flickering it has no meaning. So you can artificially add as many as bits you like, only thing is that the the the last two bits will become meaningless; that is why you cannot have an have a have I means, it is very difficult to have especially in an industrial environment, it is very difficult to have more than twelve bit convertor.

So, but you need to be aware, that if you are having a twelve bit convertor and if you are having a ten volts reference voltage; then what is the quantization error you are actually imposing and what does it mean on the actual variable, because it is getting scaled. So, may be zero to thousand degree centigrade is becoming zero to ten volts. So, so to that extent your reading is incorrect, must be you must be aware of that.

(Refer Slide Time: 26:39)



Similarly, you do various other kinds of preprocessing. You could do scaling for numerical accuracy as we have said, this is this is an altogether I mean; this is a different subject, that I

mean fix point and floating point computations. Nowadays, people sometimes use floating point but floating point hardware is expensive, we are talking about control problems; so we are looking for typically many of the micro control as will have will have fix point arithmetic. If you have fix point arithmetic; you have problems problems to compromise between dynamic range and resolution, you cannot do both.

So, if you right, if you have if you have ten bit and if you want to increase your dynamic range, then your resolution will suffer; which which which actually does not happen in floating point. So a lot of times, you are you are going to use fix point hardware. So, scaling will might significantly improve numerical accuracy.

So, you need to scale your data before you use in the regressor. We have not really talked about these issues, you might find some of these in a in a in a standard D S P book; as to what kind of error this kind of scaling can introduce, it is quite mathematical. You might like to do data differencing; that is do not build the model between y and u, rather build the model between delta y and delta u. If your system is linear is the same one, right. So, what is the advantage? Advantage is that, if if u or y had any d c bias, that will go off because the same d c will get subtract.

Similarly, you might have to do that, if you are using a linearize model because in linearize because in a linearize model by definition; it is a model between delta y and delta u, where you have to measure the delta from the operating points. So there is u zero, there is a y zero and your model is always between y minus y zero and u minus u zero, right. That is what is the model in the linearize model. So, you might like to do that.

You will have to remove outliers'. Signals which are down wide right improvable something wrong; for example, if you find exactly, if you find that while you are doing the experiment, you know you are doing it on a on a on a sharp flow. Some guy came and started doing welding somewhere, immediately for whenever that whenever that that welding arc will strike; you are

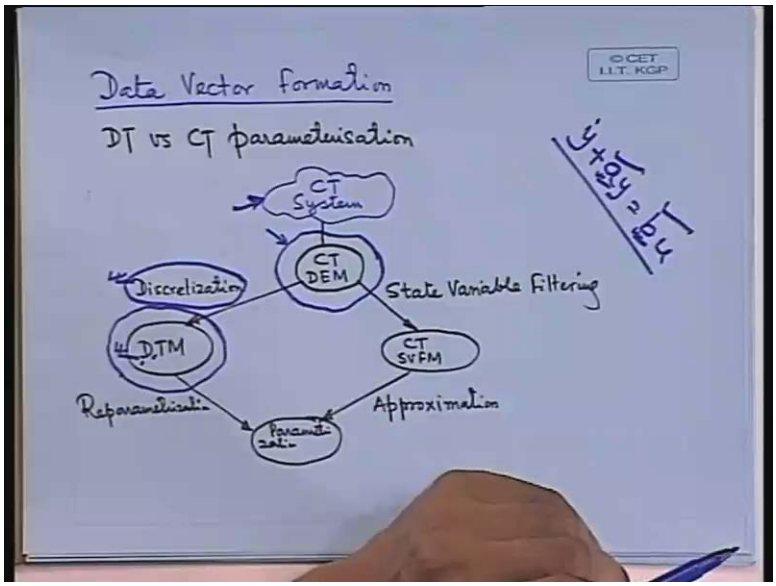
going to very noisy data. So, you must be able to I mean are you going to use the data blindly, no way; so you must remove that data, so such things are needed.

And another is linearization of measurements; sometimes for example, Thermistors. When you use Thermistors; Thermistors have a have a, what is their good property that, they have they are they are actually fast temperature sensors because they are the thermal time constants is very small but they are highly non-linear. They have they have they have very good sensitivity but they are highly non-linear, but you know that characteristics.

So so obviously, you are not going to use that that non-linear measurement; so you will since you know the thermistor characteristics, pre-calibrate, manufacture applied data sheets. So in your computer, you are going to linearize that data and use it, right. So, all these kinds of processing may be needed in a before you before you you do start doing with the algorithm, anything with the algorithm.... Well, they are they are they are the same; roughly speaking they are the same.

Now we come to the data vector formation.

(Refer Slide Time: 30:25)



That, there is you want to now, so so what I am what I am trying to stress at; this is this is something that you need to understand because of the fact that, all the time we were talking about model which are y k is equal to a y k minus one plus b y b u k minus one, that kind of models. Are they the only kind of models? It is it is even if you accept, that you need to use sample data because you are planning to do it in a computer; you are not building an analog circuit which is the least square estimator, so you are doing it in a computer.

Necessarily, means that you have to use sample data but because you have to use sample data; does it mean that, you have to use that so called difference equation model, y k is equal to something into y k minus, is is that only model? Not at all, even with sample data you could so many different kinds of models, for the same transfer function. So, so so just to in fact and it turns out that, if you use the difference equation models you tend to have big problems sometimes; difference equation models are not good, they are they are they not good in some respect. They are they are they are very nice to look at, you know directly you you are taking samples; you do not have to do anything plug that into the into the data vector, data vector computation is extremely simple just values of signals but there are serious problems with the z transfer function models, especially when the sampling time is high, right.

What are the problems? So, before we come to that I would like to emphasize that, this system is that the underlined system is is generally, almost always a continuous time system. All physical systems are continuous time, right; if it is a furnace, if its a discretization column, if it is an aircraft, if it is a circuit as well, right. So, the system is continuous time, right. Now, so ideally speaking, they are they are they are described by continuous time differential equation models; s transfer functions, if they are linear.

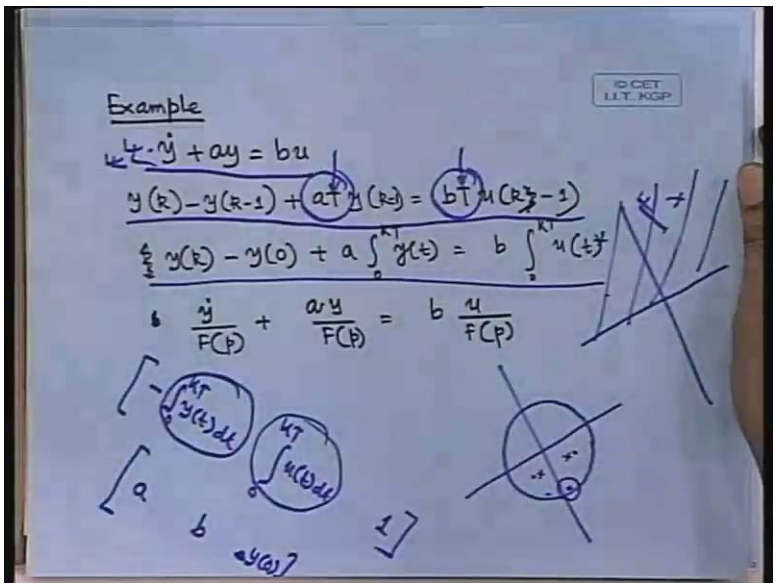
Now, how do we get our difference equation model, by a process called discretization, right? And so what we do, basically, we approximate continuous operators. You know, there are various discretization methodologies; we think that the only methodology is the is the z transform, in fact the z transform is direct z transform is is not a good methodology. Why not? Because, it does not match the the continuous response anywhere; because it is a it is what is

known as impulse in variants transform, you could you could have discretization using an impulse in variant transform or step in variant transform or a ramp in variant transform, various kinds of things.

So, before without going into that, I just like to say that that it what what model you get depends on a process of discretization; some process, there are many methods of doing that. So, this is the model, that we have seen so far in the course, always you know y k a y k minus one. Now, there are big problems in this. What are the big problems? First we will find the parameters; see there are some continuous time parameters here, suppose you have a system y dot plus a y is equal to b u, this is your continuous time difference equation.

This is your continuous, these are your real parameters but the moment you come to a to a difference equation model; the difference the z-transfer function model a b parameters are not only these made of these a b's, they are also made of the sampling time. They are function of the sampling time and that creates huge problems, why we will see. So, what I want to say is that is not necessary, that you have to necessarily go to a discrete to a discrete difference equation model; you could have various other parameterizations.

(Refer Slide Time: 34:26)



For example, without going into without going into any kind of generalization, let’s see this simple system. This is the real C T system; we want to approximate this using sample. right because we work with the computer, we need to approximate this using sample; because I cannot get y dot, who will give me y dot, I am only getting y's and that to a distinct points of time. So, so what is the simplest strategy? Write y dot k is equal to y k minus y k minus one by T; that is the simplest approximation of derivative.

So, if you do that, you get this point. Now look at these, these are your discrete time parameters which you want to identify; they are they are they are functions of T, what happens if you are doing fast sampling? They are all go to zero. So, actually what happens is that, do you remember that this is the F plane and this is the z plane; so this is the left half s plane where does it get map to? It gets map to inside of the unit circle, right. You see this is a semi infinity plane, large such a big area that get that gets map to a small circle, which means; that if you have a small error here, it could mean a huge error here, is it not?

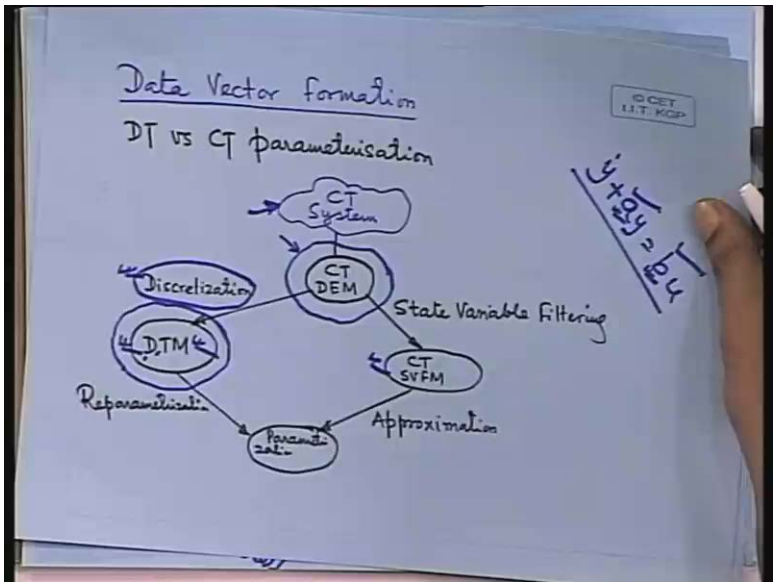
So by due to noise or something, if you are if you are and you are you are your your your system poles and zeros in your discrete time model, you are going to identify here. So, so suppose the suppose the true pole was here and due to noise, you estimated the poles to be here; in your discrete time estimation, the actual poles and zeroes of the of the continuous time system could be could go haver, you could you could you could get double the steady state gain and things like that, people have shown.

So, these especially at fast sampling rates, these models are numerical extremely sensitive. Little error will give you; after all it is the it is the steady state gain which is the reality; whether you have got this, what how you got this Parameterization is your problem. If your models, shows the shows the different steady state gain you add it total controls loop everything will go here where you are having the, either your response will suffer or your steady stability margin will suffer one of them will happen.

So, now what what is alternative? The alternative is simple. The alternative is can you not write the same model also like this? Just integrate between zero and t, this this equation; what you get? You get y k minus y 0, if you integrate y dot. Here you get a integral of 0 to k k t y t. Here you get b integral of 0 to k t u t. Now, can you not have a data vector five which will be which will be minus integral 0 to k t y t d t, this is one element. The other element is integral 0 to k t u t d t and the other element is 1. And then you will then then then what parameter will you get? You will get a, you will get b; which are the same as your continuous parameters, which do not depend on t, do not suffer from that problem and and y 0.

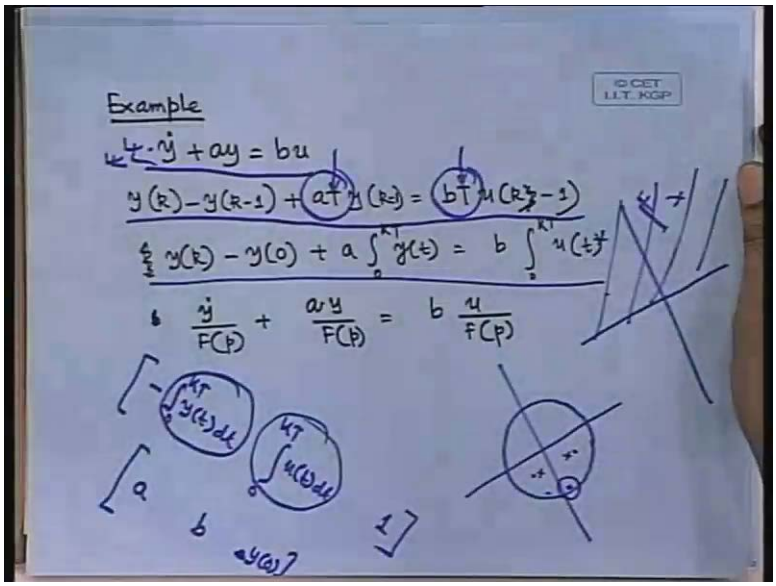
So, you will also estimate an initial condition. See your, see this this you can very well compute using sample data, what is the problem? No problem. If your sampling time is faster, this estimation will go better; because your models will be because your signals will be more accurate. So, all I am trying to say is that, just because this this should be said because all the time, we have used this kind of a model but you must remember; that you do not need to do that all the time at all, you need to just write an equation phi transpose theta, in which is possible in many ways, okay.

(Refer Slide Time: 39:00)



So, you need to while you are doing this data vector formation; you need to know the parameterization, in terms of what are you going to write?

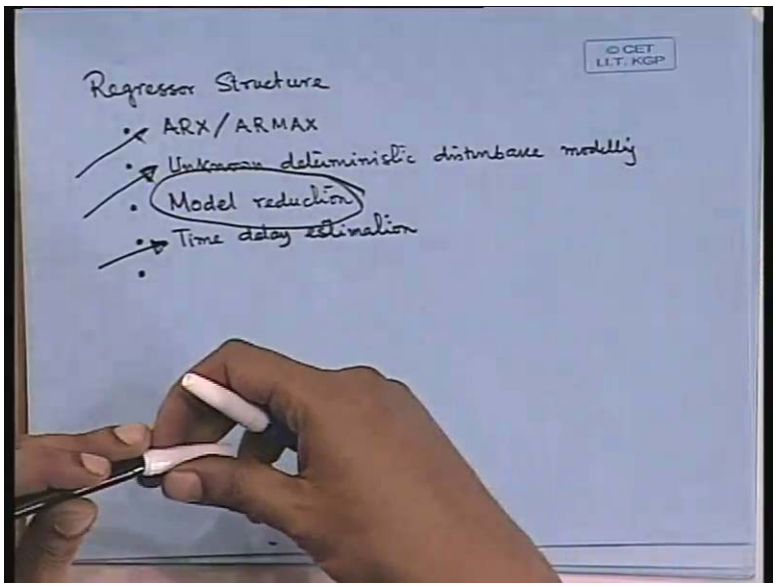
(Refer Slide Time: 39:05)



Are you going to write it in terms of integrals? Problem with integrals is that, if you have a slight bias here in y; that is going to blow up. So so so may be you do not want to like, I mean may be you do not want use integrals, use use any other filter; use a filter which which which will not blow it up, I mean an integral is a is a is a very is an is an unstable filter, itself some originally stable filter.

Another big advantage is that, this will have very little noise because it's an average. So any noise, that you are that you still have it in y and u are going to get average and will go go away; typically you get you get very high accuracy on this, much higher compared to your discrete time model. And discrete time models have been used for for more of our traditional reasons.

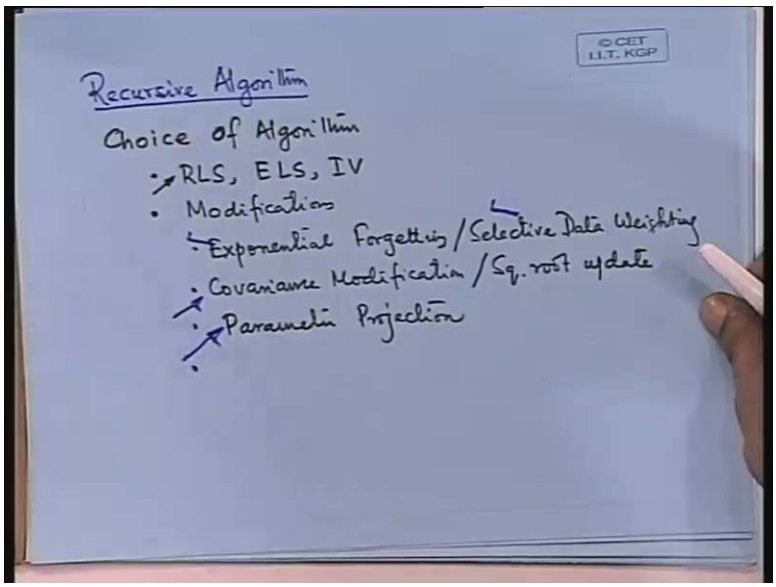
(Refer Slide Time: 39:59)



So, having said that you still have to decide lot of things like, whether you are using an A R X or A R M A X structure, whether you need to model unknown deterministic disturbance; remember that, we said that if you have a sensor bias then then you could also estimate that bias. So you could model the disturbance, deterministic. You could have, you could recite that what is the model order that you want to select, depend I mean; reduce it as you want. And you need to know whether there is any significant time delay involved, you know unknown time delay estimation is a is a is a tricky problem which I have not considered; it requires other kinds of approaches.

So, in general you you time delays may be there but you need to do; I mean spend some effort in trying to know them and then using using in your parameters otherwise time delays time delays lead to very high order models because because e to the power minus s t cannot be approximated as a as a as a finite order transfer function, sufficiently accurate. So, you probably need to take care of it using a priori knowledge.

(Refer Slide Time: 41:14)



Coming to the recursive algorithm; you need to choose your algorithm. What short of algorithm? Recursive least square, sometimes you use there are several versions of it; extended least square, instrumental variable, vast of algorithms. Sometimes, you have you know special kind of algorithms which have very good properties, so you need to carefully choose your algorithm; since we have not read all of them, we cannot discuss them too much.

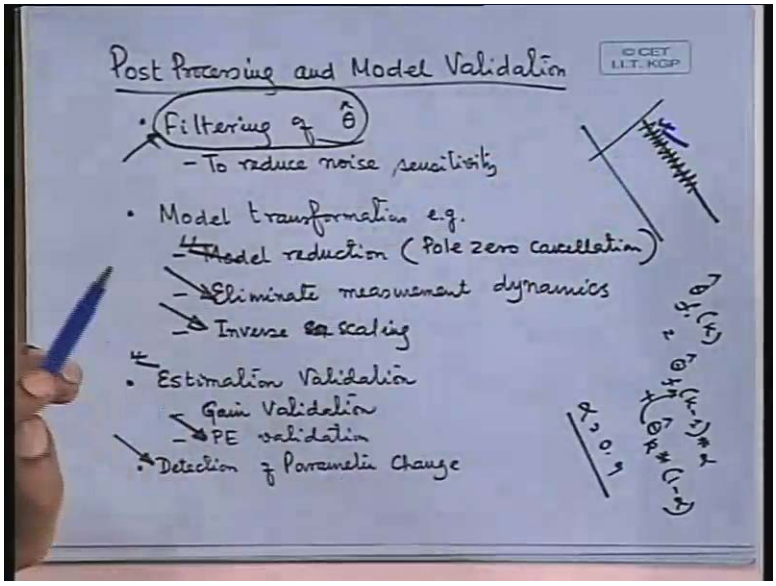
And then you need to also think of various kinds of you know; patches, smart practical I mean, patches that you want to put in your estimator, for two or twelve. For example, exponential forgetting, selective data weighting. You do not want to update your model all the time; if you have too little prediction error, its good do not update parameters you might save time. Similarly, covariance modifications; do you like that your covariance is is does not die, normal recursive least square the covariance almost dies. If you want you can keep it alive, just to keep this estimate again high or you could go for a square root update; to ensure that it always remains positive definite, no numerical problem.

You could use parameter projection, based on I mean always stable models. So, anytime you get an estimate; check stability, if unstable find the nearest stable one. So such things are needed and

they will give, sometimes they can give, what they can give is significantly faster convergence; you know you you you are you are you are preventing it to go astray, normally an estimator, if it wants to go astray, it takes some time for it to come back to the track. If you if you closely monitor it and and prevent it from going astray then you tend to have faster convergence, that is very important especially; when you are if you are trying to use it online or you are trying to use it fall detection because that will directly cut down your fall detection time, if the convergence is faster.

So, after doing this, we hope that you got your parameters; something came out of the algorithm at every times, step. So by the way incidentally; how its its not again necessary that, you need to run your, do you need to run your how often do you need to run your parameter estimator? It is not necessary that you need to run your parameter estimator every sampling time. It is not necessary. You might run it every ten sampling times or you may run it; after all this k need not be the exactly the exactly the sampling time k, you only need to set of that model there can be ten twenty samples later. I mean why do you normally, you do not need to control input in a way to be computed every sampling time, parameter estimation is not be because parameters are generally much much slower varying quantities. Similarly, post processing and model validation.

(Refer Slide Time: 44:14)



This, you you we have seen in some cases; for example recently we used the neural network estimator which is a non-linear estimator. Would you believe it for for estimating, we we have actually used it, for if I if I if a carbide tool is actually cutting metal; its getting worn out, so its cutting edge, dimension is changing, change is of the order of microns, fifty microns, hundred microns, two hundred micron just by sensing vibration, motor current, we had been able to set up a an an an estimator which can estimate that, wear of the tool within twenty microns without measuring it at all, real time.

Now, in such cases, you will find that filtering of the parameters can give you a lot of benefits; that because because the parameters estimate is definitely going to be gettering because of noise, whatever if if there is any amount of noise then your estimate are always going to like that. Now, obviously the estimate is not this, obviously the estimate is much slower. So, after you got this estimate, it makes lot of sense to actually low pass filter the estimate itself, right. So you write theta hat filtered k is equal to theta hat filtered k minus 1 into alpha plus theta hat k minus k into 1 minus alpha. And then you choose alpha equal to 0.9.

So you see, every time you are not changing it, every time you are adding little bit of the estimated parameters, so you are actually doing an averaging, correct. Then you need to check, before you hand over the model; now you are going to hand over the model to the to whoever is going to use it. So you need to see whether there possible possible model reduction, whether you you you need to eliminate measurement dynamics from it; if you know it generally measurement dynamics is is could be available very well from the manufacture as data sheet, because the sensors are well calibrated and tested by manufacturer.

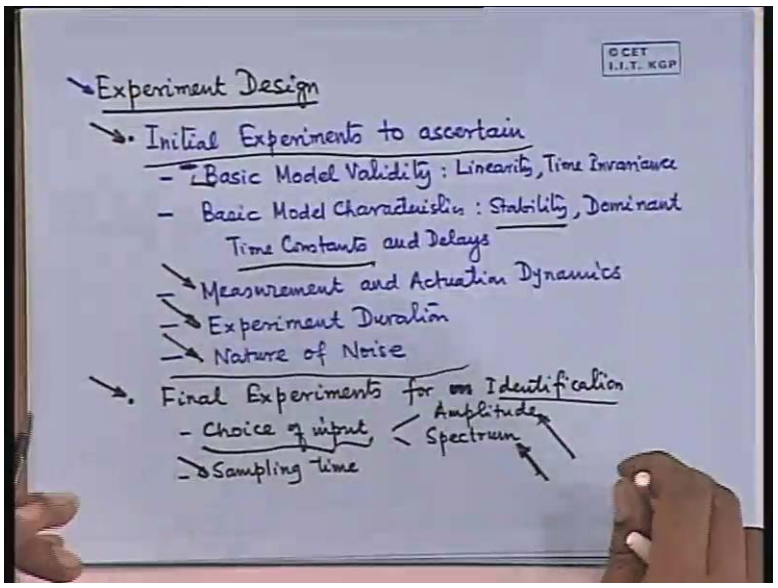
You have to do inverse scaling, remember that you used, you actually used the parameter in terms of scale data; so obviously the parameters also got scaled, so you need to do inverse scaling to actually come back to the units in which it will be useful. And you need to validate, whether your estimation process is going healthy. So you need to know whether your estimator gain is dying, whether your estimator gain is corrupted, you need to check the prediction error to

see whether it is white; whether it is uncorrelated with you, whether you extracted all juice out of the data, you need to see all that.

And sometimes, you you also need to detect parameter changes; because actually what happens is that, you you know this is a this is a this is a contradiction. Parameters are generally likely to be slow, vary; so if you keep your estimated gain high then it will become sensitive to noise and it will start oscillating like this. So while it is while it is steady, you need to reduce the estimated gain but if you reduce the estimated gain then when it will change; you will not be able to track it fast. You cannot have both at the same time.

So, so what you should do is you should have a separate mechanism of checking whether the parameter has changed. If you find that, it has changed or it is very likely that it has changed; you boost again otherwise keep it low, so that you have the best of both volts, so such things you might like to do.

(Refer Slide Time: 48:08)



Finally, coming to experiment design; now you see that you have to design the experiment, you have to say what input you have to give, so that you get the best model which will really suit

your purpose. Sometimes, the if you are doing offline identification, there is one time you have to identify the plant, you can design your experiments. If you are using in a in a online application, you cannot choose your experiments, input is coming; like in like in adaptive control, in an adaptive control any other control is coming, you cannot give a from suddenly give a sinusoid just to identify the model, you have to do it based on existing data but in some cases you might.

So, actually what you have to do is you must remember that; you have to do the experimentation is to be done in at two stages, as I said that you first to be able to have a good estimator you need some a priori knowledge, about the system and its environment. So, you must plan your your your initial experiments to determine certain things before you actually go for data collection. So you need to know basic, whether your whether your basic, your linearity, time in variants, whether these properties are valid on that system.

So, you need to check that, then you need to know; whether it is stable, what is what are that rough ideas about the dominant time constants and delays, I mean as you have seen that, that without these it will be very difficult to choose the pre-filter. So, you cannot even implement your algorithm before you have some idea of this, totally black box approach never works in practice. Then measurement and actuation dynamics, experiment duration, how much data do you need? Remember, in an in an actual data, collecting the data, if you if you if you go to a factory and say; can I try to identify your rolling mill?

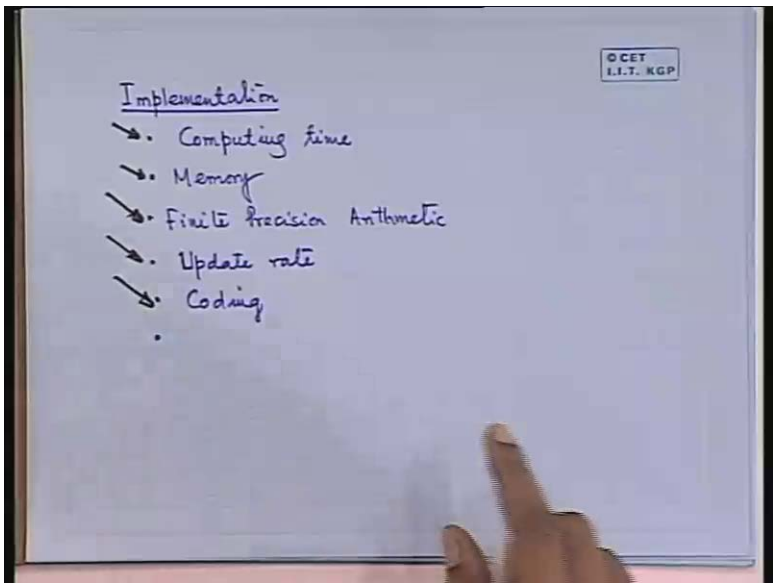
The people will give you, that bloody looks because because if you are if you are going to identify the model for that time; he cannot produce, his manager will shout. And people will not like you; if you say, I want to identify your plant, so you have to be quick, you have to collect just that amount of data as you need but at the same time you have to make sure that, your job is done because if you come back seven days later and said, I need another twenty milliseconds of data, that that will not pleasant, either.

So, so I mean you need to fix experiment duration. So these these are very practically important things, right. Then nature of noise and disturbances which might come, these things are very important. And then after doing these, you will be probably be able to design a kind of final experiment which will which will let you actually build the model; for that you need to choose your inputs and while you are choosing your inputs, you must decide on the amplitude and the spectrum. Why? You must make the amplitude high, because that will make your data resolution high. You have to excite the system properly; if you make it too high, you might excite non-linearity. So you cannot make it too high.

So no clear guideline again, so called coat and coat engineering judgment. How high you need to, what is it is going to be your amplitude and what is going to be your spectrum? Spectrum should be such that; if you are if you want to know, what is the what is the gain margin of your of your process then you do not give a d c because your because your gain margin is I mean, your gain cross over frequency or your or your face cross over frequency might occur at five hertz.

If, you are if you are looking for determining the correct face cross over frequency then you must excite it around three to seven hertz. If you give it zero to two hertz; you are not going to get your gain cross over frequency or face cross over frequency, correctly.

(Refer Slide Time: 52:03)



And of course, sampling time. So, so that is all, only one point before we end is that finally remember that; that is all theory I never remember that, you are going to code it in the computer. So, you are going to need some computing time, you are going to need some memory, are this important as long as you are using a p c, not important, but if you are using an embedded controller; they could be very important.

So, the embedded control are small devices, it might have you know sixteen k of on chip memory; if you have written a grand algorithm, it could mean that it will require separate memory chip, on chip memory does not do which will mean cost test collection, which will mean complexity of hardware, which will mean so many things. If you are a product designer, then it could mean between the success and failure of your of your application.

So, it all depends you you everybody does not use the supercomputer to be able to identify a model, so these could be issues. Finite precision arithmetic, you have to when you are coding your numbers especially in rudimentary processors, using fix point not too high word length arithmetic, you need to be careful about the truncation errors; you know things which go under

what is called the numerical CALPET, there could be a lot of dirt there and could affect your results.

Remember that, especially if you are going to result use your results online for control; it is critically important, if you by some due to some somebody had a my algorithm give a give a divide by zero mid-flight is not pleasant at all. You have to decide an update rates, why because because do not think that, you are going to have a full processor to you just for updating theta hat k; the same processor is is probably going to update three other control loops. So, it is not if you if it is not necessary, you should not update at every sampling time; it all depends on how fast your processor is, how many tasks you should deal on it, what are their sampling update rates?

So, its its generally actually, what is known as the multiprogramming scenario; in which you are going to implement this, if you know what is multiprogramming. And, finally well; coding, so hopefully I have been able to, to the best of my knowledge I tried to draw your attention to certain various factors which really need to be considered, if you want to get a successful model. So, that kind of closes our identification module, some of the major things that we want; I wanted you to remember is that, you must remember that the system is generally much more complex than the model and selection.

Selection of a model actually the the true model is actually a figurative thing; there is there is there is no such thing as a true model and it all depends on; what how good an approximation you can make using resources, which you can afford and for the purpose that you need. So its its actually the engineering decision, there is no right or wrong here; just like in most most most actual engineering decisions, they are only goods good ones and bad ones. Thank you very much.