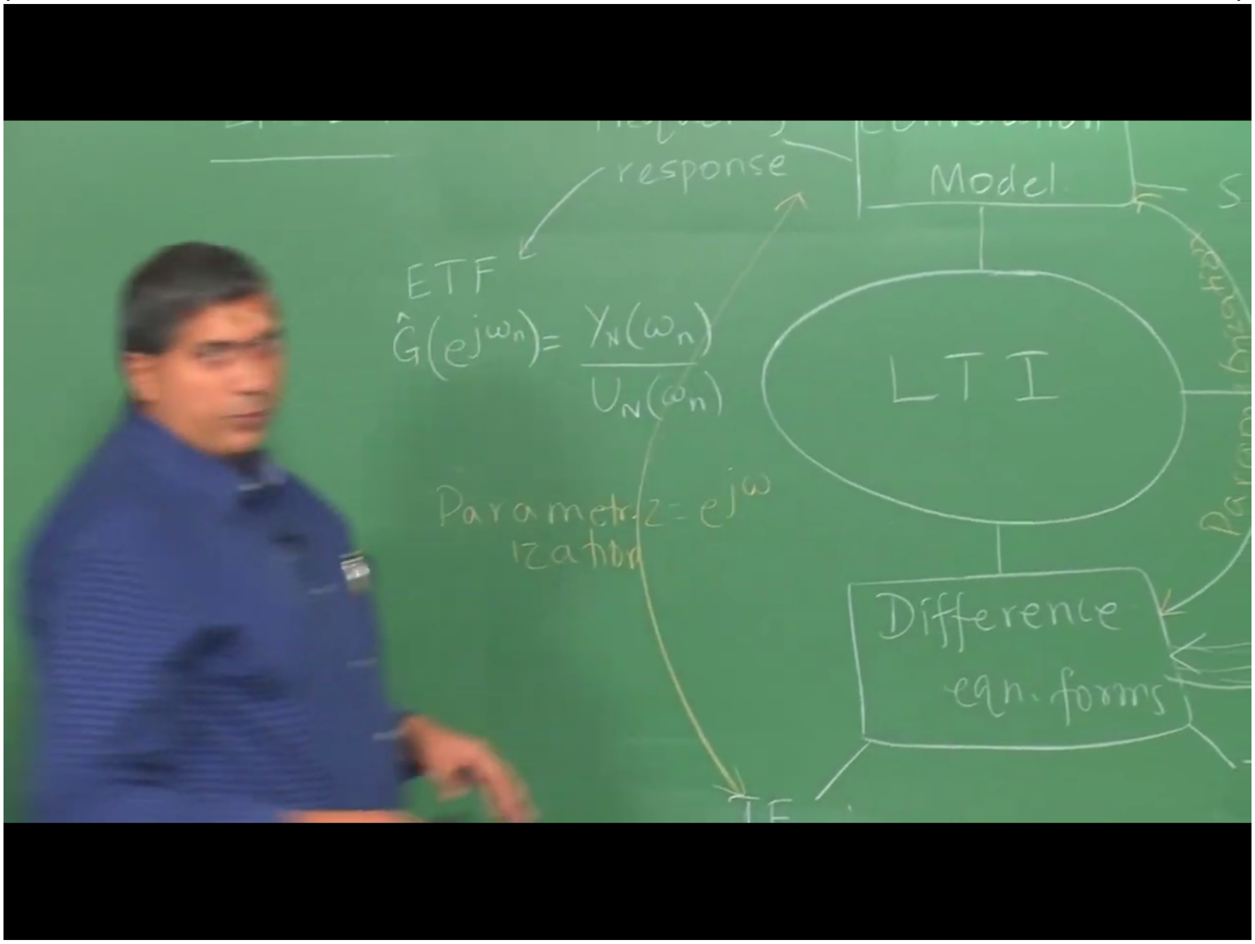
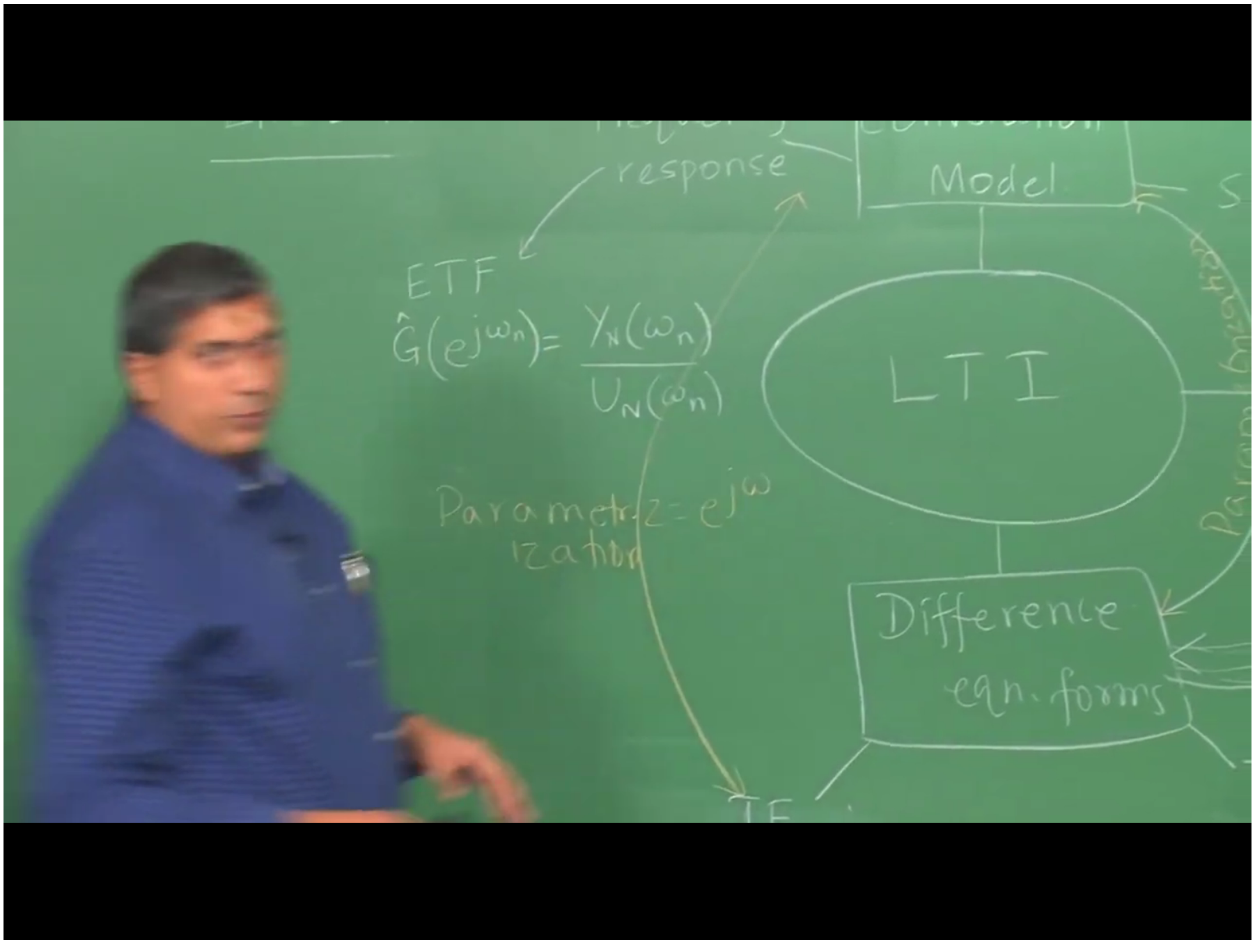
Until last class we have looked at models for linear time invariant system, in particular deterministic systems. And we have learned that there exist different forms of representation and we have done that by starting with the convolution form, which is kind of a natural model for an LTI system. And then we moved on to other models. So here is your LTI system. One of the fundamental models is the convolution model or convolution equation, right. So this is one model of the LTI system and from the convolution model, we could look at three different responses… models. One is the impulse response model, so I am just summarizing things for you, so that you can put all this concept in a good perspective and then you have the step response description. And on the other hand you have the frequency response description, right. I have segregated the impulse and step on one side and the frequency on other side, for reasons you must have guessed, the frequency response is described in the frequency domain, right. So this we call as the non parametric models. And then we said, look from an identification view point there must be something that we have to do in order to reduce the number of unknowns. And one of the things that we did was, we went further and wrote the FIR model, assuming that the system is stable, right. And that was one… that’s the set of approximate models if the system is infinite impulse response. If the system is truly FIR, then it's not an approximation. And on the frequency response side also we studied an approximate model, and what was that? The ETF, right the empirical transfer function. So here we came to the empirical transfer function, where… where we define empirical transfer function as the ratio of the DFTs of the output ring. So this is theory, these are all theoretical definitions that we went through and then we said, if I have to practice this theory, then one of the parts that I have to take is FIR model. If I have to implement this theory in practice, then this is one that I have to… that I can define. This is not the only approximation, you can come up with your own approximations. Even in these kinds of descriptions, you may have a large number of unknowns to estimate or even if you go back to the original impulse response model, if the system is infinite impulse response, then there are infinite number of unknowns. To overcome that and keeping in mind the estimation principle, which is that keep the number of unknowns to a minimum, we resorted to parameterization and that parameterization gives birth to difference equation forms, right. So the root that connects this convolution to difference equation form is parameterization. So if you have to write along this, say parameterize. And from this difference equation form, we define the transfer function operator, so we had the TF operator form, right, and then we had the TF, the transfer function itself, which is in the Z domain, so this is the shift… in terms of the shift operator, but the transfer function is in the form of… in the… defined in the Z domain. In fact the special case of this is your FRF, I could have written, let me do it the other way so that we keep all the time domain on the right hand side and frequency domain on the left hand side. So this is in terms of Q inverse and here we have the transfer function, which is in the Z domain and we know that these two are connected, right. In fact not this, but these two are connected, frequency response and Z domain, the transfer function by working only on the unit circle in the Z domain, if it exist, we know that frequency response function exists only for stable systems. What else is missing here? The mighty state space description, right. So that is missing. So here you have the state space description, which relies on a completely different concept or notion of a state, that is kind of fictitious, some time snot fictitious, but the general understanding is that the states are not necessarily observed, whereas the measurements, the outputs are always observed, so we have therefore another description, and within this you have the non parametric state space models, which we call as unstructured state space models and then you have structured state space models, right. If you were to go from a difference equation form to a state space form, there are infinite pathways, right. There are infinite models that you can get, but the fact is, if you go from a difference equation form to a state space form, you will necessarily end-up with some structured state space form, alright. Whereas in identification, you have the privilege of directly fitting a state space model, in which case you may know the structure, you may not know the structure, and as you will learn later on, state space models, initially its best to identify them without a structure, unless there is a compelling reason, not to pursue the truth. Because the algorithm for identifying so called unstructured or fully parameterized or freely parameterized state space models are very easy to implement, whereas the algorithms for estimating a structured state space models requires you to solve an optimization problem, a constrained kind of optimization problem. And we know that all of this now, I can go from state space models to these forms as well, they are all connected, its… its bidirectional. There is no uni-direction here. Coming from frequency response to transfer function, also is possible, provided you know the structure, so here also you can come, you can arrive at the transfer function representation or the difference equation form, provided you know the parameterization. In fact there are certain fields of research or there are certain applications where people are interested in arriving at the transfer function form, starting with the frequency response form. And what they would describe is, I have a frequency response description, I would like to arrive at a complex frequency description. So you perform your experiments on the pure frequency, that is sinusoids and cosines remember, the transfer function tells you how the system responds to complex frequencies. Damped E to the J omega times E to… E to the J omega K times E to the sigma K, right. So they are called complex frequencies. How to arrive at this from the frequency response, that involves some assumption on the parameterization, on the… some structure. So the story is same here, you can say more or less there is some parameterization involved here as well in arriving at the transfer function. So they are all bidirection and… and the question is now which of these models, right. And not only now, but always, which model structure and as we have discussed many a times, it depends on the end use, the Es of estimation what you are looking for and so on. And that we will learn as we go along. So this is what we have learn, in a nutshell, what we have learnt until now. We have gone through the math and so on and I have also drawn your attention to the MATLAB commands. What I have not discussed are the MATLAB commands for the input output description. So at the top all this corresponds to input output descriptions.

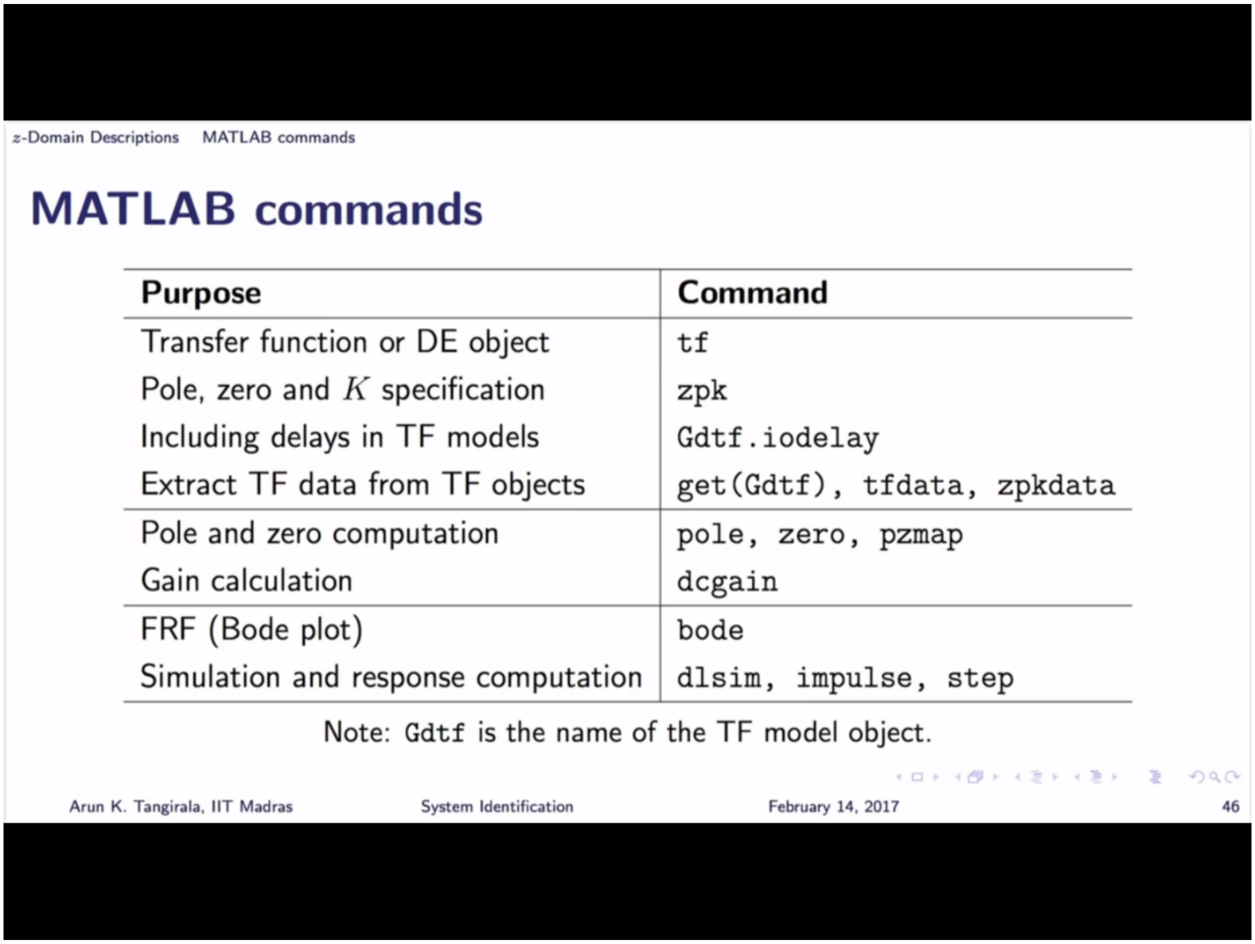
(Refer Slide Time: 11:00)





And you have this, some of this you must have already known in the process of solving your assignment. So you have TF that allows you to describe a transfer function object or a difference equation object. Then you have ZPK. ZPK stands for Zero Pole K. you can rewrite your transfer function by factorizing the numerator into zeros and poles and then there is a constant factor that falls out, that constant factor is called K, right. Don't think K is the gain, gain is different. And you can also include delays in transfer function models. Remember these commands here are applicable to both continuous time and discrete time. Many a times you may have to introduce delays in the continuous time model and you can use this IO delay property. All these objects are essentially structured objects in MATLAB and I hope you are all familiar with structures in MATLAB. The structures have fields, which you access using the dot operator. How do I know what fields are there in each of this, as I said in the last class, you use the get, right. If you use the get command, it will fetch you.

(Refer Slide Time: 12:18)



z-Domain Descriptions MATLAB commands

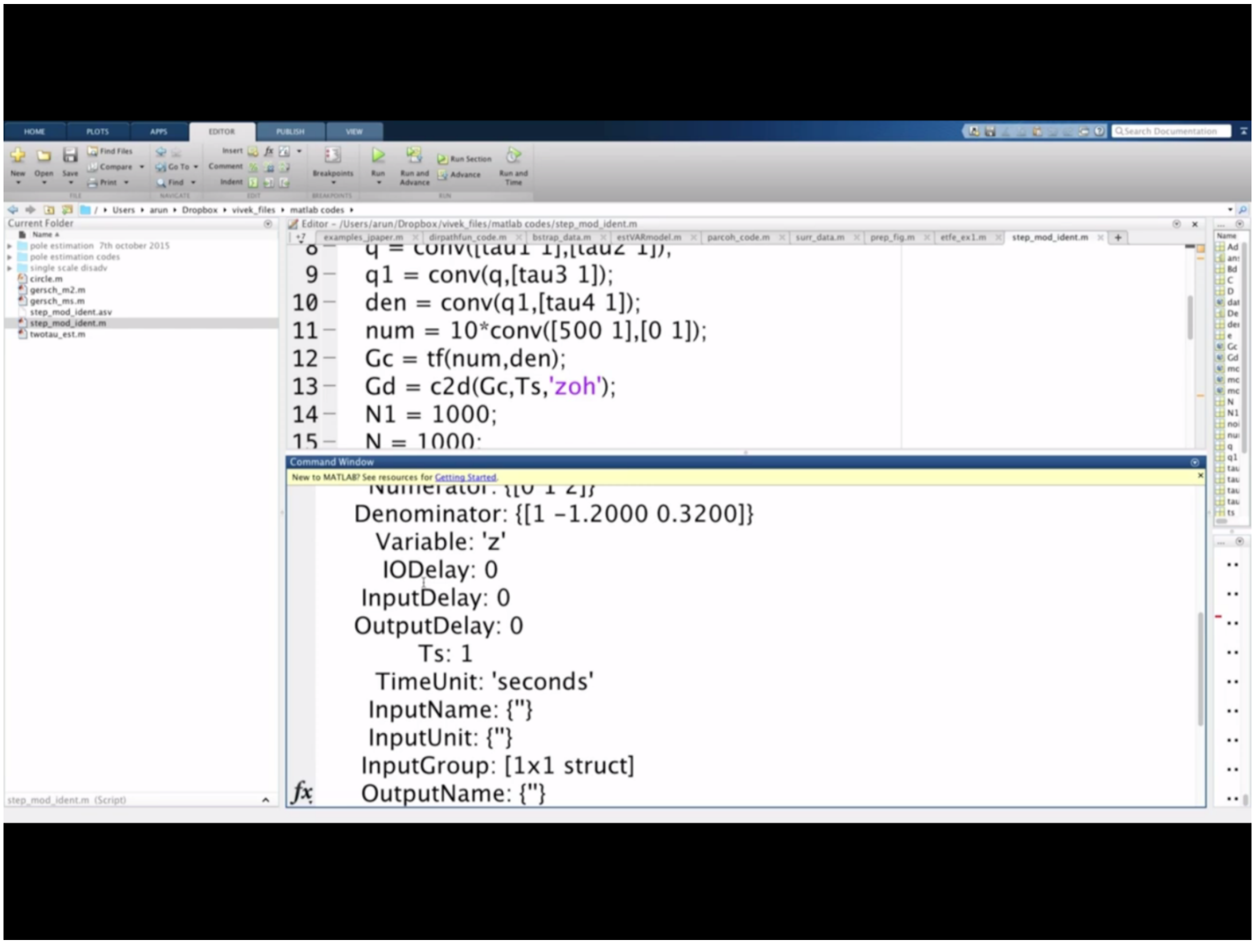
MATLAB commands

Purpose	Command
Transfer function or DE object	`tf`
Pole, zero and K specification	`zpk`
Including delays in TF models	`Gdtf.iodelay`
Extract TF data from TF objects	`get(Gdtf), tfdata, zpkdata`
Pole and zero computation	`pole, zero, pzmap`
Gain calculation	`dcgain`
FRF (Bode plot)	`bode`
Simulation and response computation	`dlsim, impulse, step`

Note: `Gdtf` is the name of the TF model object.

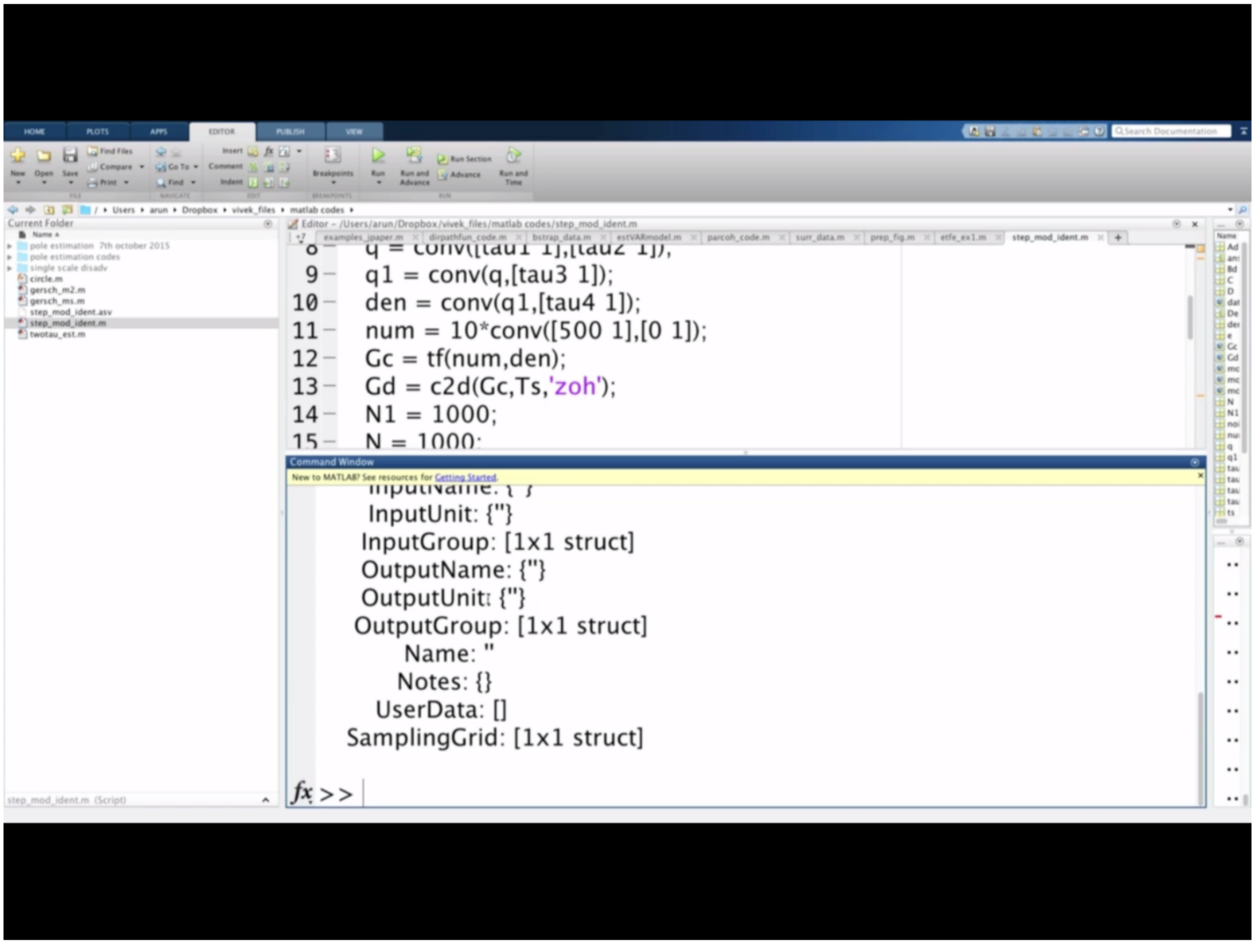
Arun K. Tangirala, IIT Madras System Identification February 14, 2017 46

So let's pick a transfer function here. Let's say I pick this state space model, right, okay, now let me pick this transfer function. So this is my GD. And if I do a GET of GD, it gives me all this properties for that object, right. And you see that there is a numerator which is a vector, in fact it's an array. The reason its enclosed in curly braces is so that if you have multiple input multiple output system, then you would pass vectors of coefficients for the corresponding input output channel, right now we are looking at the seesaw system. So you have only one entry in that salary. And then likewise for denominator, it tells you in which variable, what is the input output delay, it distinguishes between IVO delay and input delay and output delay, for a seesaw system it really does not matter, because you can always transfer the delay from the input side to the output side, that is there is nothing wrong mathematically in doing so.
(Refer Slide Time: 13:31)



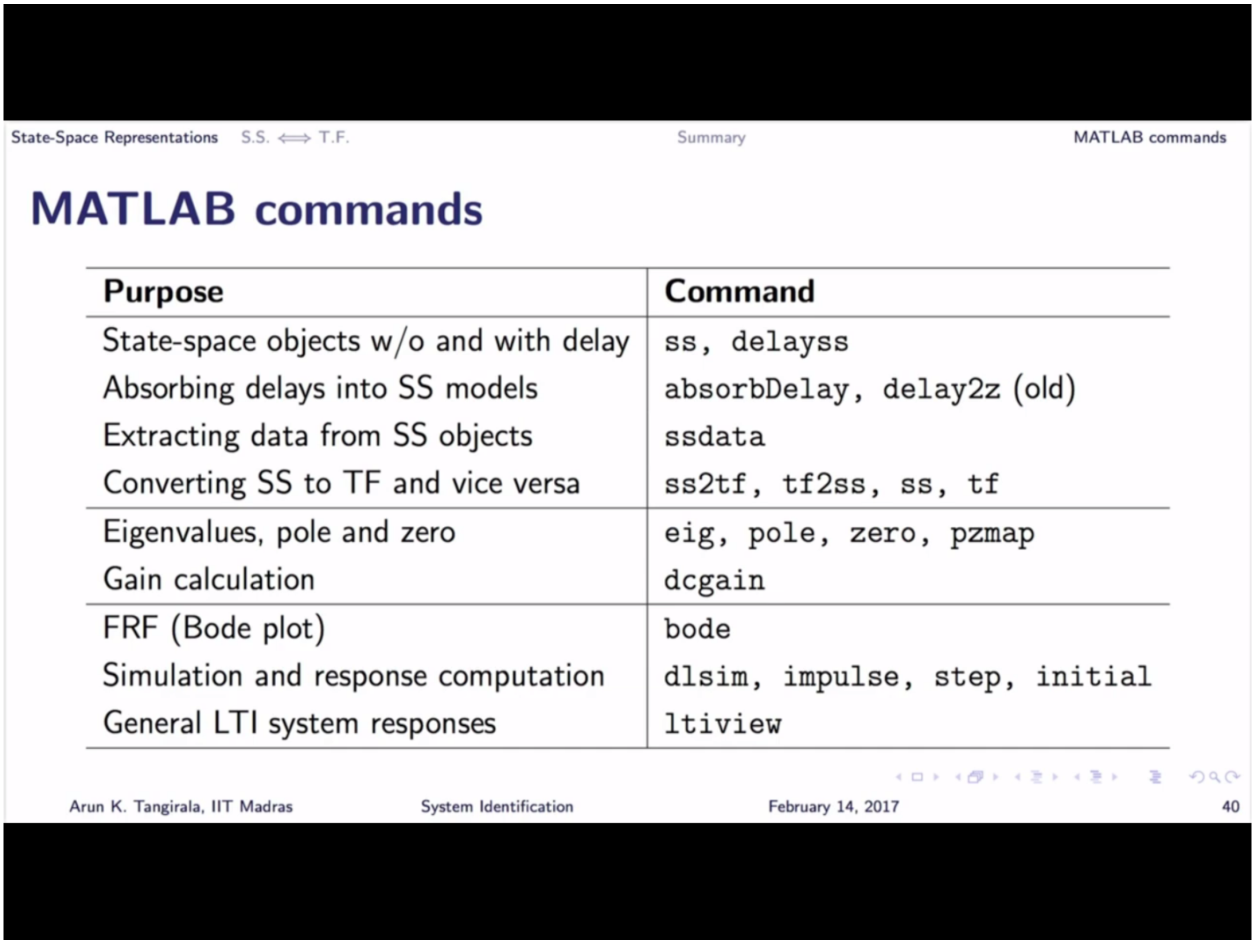
Whereas for mimo systems, input delays you cannot delays, you cannot really transfer just to the output, just like that, because you will have a multiple input multiple output system. Each delay corresponds to that particular channel. If there is a delay in input channel that is specific to that channel, you cannot move it around just like that, you can, but then you have to follow certain rules. So instead of doing that, if you know that there is a delay in input channel, you specify that in the input delay. If you know there is a delay in output channel, it could be due to measurement delay. Why do you have delays in the inputs? May be transportation lacks, they say. That is from the time you… point that you inject to the input or the time that you inject the input, to the time the process is excited, that is the input delay. Likewise you can have measurement delays or other reasons for output delay, you can specify sampling interval of course, and then there are, you can give names to the inputs and outputs, you can group them, and then some user data, right. Where you can store some information that is useful to you, and there are certain other fields that are of interest, not for us at the moment.

(Refer Slide Time: 14:50)



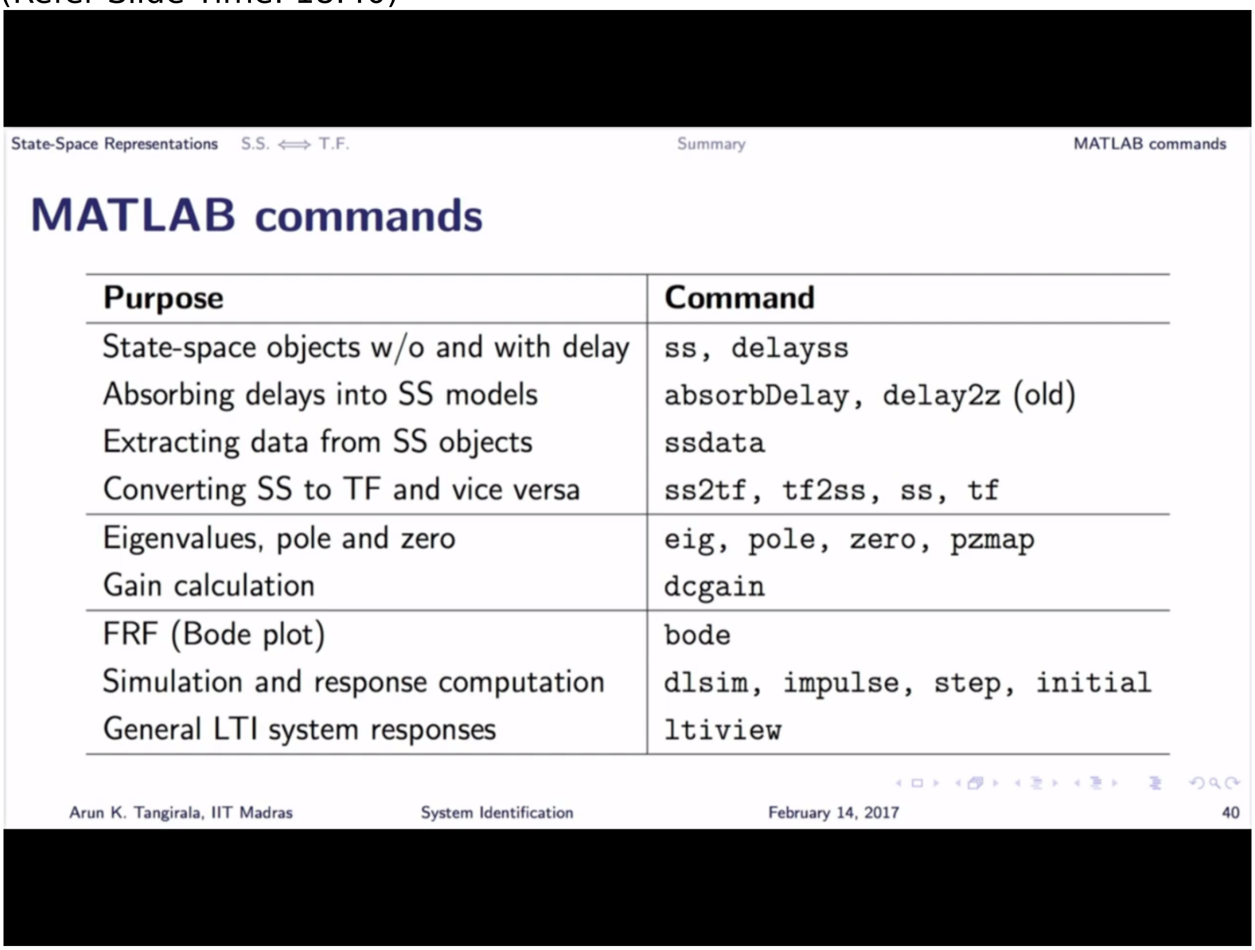
So just to tell you know, basically the get routine gets you all the properties associated with an LTI object in MATLAB. You can also use TF data to get the numerator and denominator, instead of using get, but get is more powerful, because it gets you all the properties. Likewise you have ZPK data, which gets you the poles, zeros, and K, given a ZPK data object. And as far as analysis is concerned, you have pole, then zero, and then PZ map, which gives you the pole and zeros and PZ map draws the poles and zeros for you, depending on whether it's a continuous time system or a discrete time system, it will generate the particular plot. You know DC gain already, bode we have looked at, dlsim or lsim, you can use both, used for simulating the systems with arbitrary inputs, that is user defined inputs. And you know impulse and step routines. Any questions on this?

(Refer Slide Time: 15:58)



Okay so now speaking of MATLAB commands, dealing with state space objects, as I told you in the last class, SS allows you to create a state space object, continuous or discrete time, delay SS which is quite useful in incorporating delays into your model. You can go and look up the help on delay SS, it tells you the syntax through which you can create, you can incorporate delays into the state space model, and an example is given for you, you should follow that example and apply it to the example that we had discussed in the class. If you recall, when we discussed state space models, we learned how to go from a state space model with a delay to a standard state space model, by augmenting the states, correct. Let us that you want to verify that in MATLAB, first you have to create a state space model with the delay, that is without augmenting, just specify the delays in the syntax that MATLAB expects and that is where you use the delay SS routine. SS routine generally gives you the, allows you to generate the state space object, but delay SS is the one that you want to use when you want to verify that example. Then now you want to actually figure out, what is the… that is you want to arrive at the standard state space form by augmenting the states and so on and that’s where you can use the absorb delay routine, which absorbs the delays and comes up with the state space model in a standard form. If there was a need to increase the number of states it will do so and gives you… and it will give you the new state space model. You should try this out and you can even do this using delay to Z, but that's an old one, absorbed delay is the one that is shift with the more recent versions of MATLAB, so you should stick with absorb delay. SS data as usual gets you the state space objects. SS to TF to SS and so on, you know that it allow you to switch between state space and transfer functions. You can also you simply SS on a transfer function object, it will give you the state space representation, like wise TF. Sometimes I prefer that. And then you have the standard eig, pole, zero, pzmap, dcgain, bode, and so on. LTi view, as I have shown you in one of the classes, is a general LTI viewer for handling any LTI system, analyzing any LTI system, where you can generate impulse response, step response, bode plot, nyquist plot and so on.

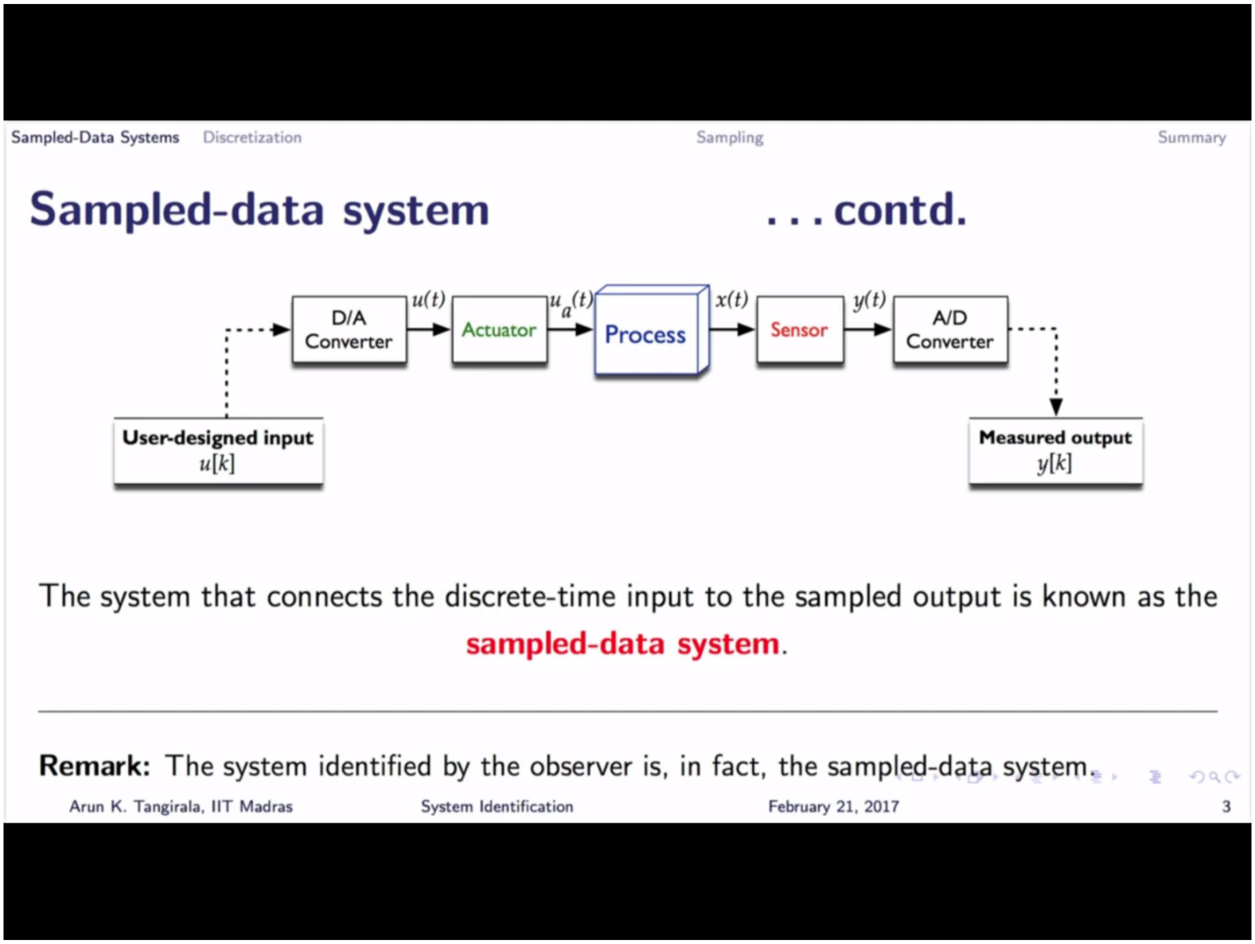
(Refer Slide Time: 18:40)



Purpose	Command
State-space objects w/o and with delay	ss, delayss
Absorbing delays into SS models	absorbDelay, delay2z (old)
Extracting data from SS objects	ssdata
Converting SS to TF and vice versa	ss2tf, tf2ss, ss, tf
Eigenvalues, pole and zero	eig, pole, zero, pzmap
Gain calculation	dcgain
FRF (Bode plot)	bode
Simulation and response computation	dlsim, impulse, step, initial
General LTI system responses	ltiview

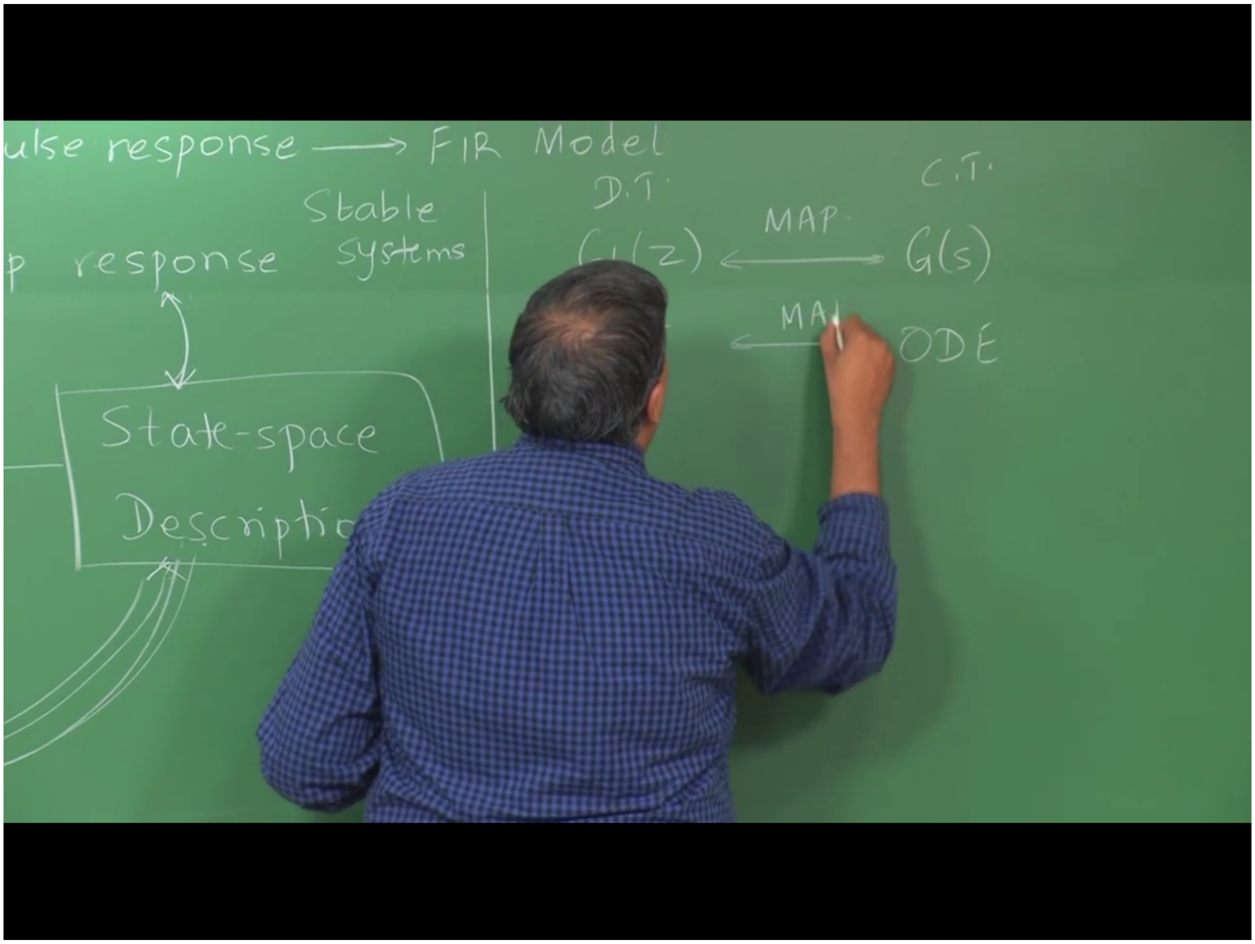
So these are the related MATLAB commands that are relevant to us at this moment. Gradually as we learn more, we will learn the associated MATLAB commands as well. So now we come to the main stay of today's class, which is turning… which is that of discretization and sampling. So until now, as I have described on the board earlier, we have been looking at discrete time LTI systems and the reason was of course to know what models are available and I can choose that model when I am fitting some model to a given input

output data. But now we will go one step deeper and recognize that this input output data, if you recall the diagram that we had, this is the actual story most of the times, that the discrete time process that I am looking at is not necessarily naturally discrete time. Do you understand what is meant by naturally discrete time process? What is the difference between a naturally discrete time process and a discretized process. So can you give an example… okay anything else… our salaries, our stipends, very sadly are discrete time, right population is fortunately discrete time, okay. Popul... that is growth of people. So those are all naturally discrete time processes. They are not a result of any discretizing continuous time process, that is as you said, it is the discretized process is born by observing a continuous time process at some specific instants, it could be regular, it could be irregular, we don't really worry about that. But what that means is, there is an associated continuous time process. And typically that is a situation that we will run into, at least in the last class of identification problems, that doesn't rule out the naturally discrete time processes, whatever, theory we are learning, we will learn, applies to any discrete time process, but by enlarge, we may end up dealing with discretized processes, which are a result of sampling a continuous time process, like the one, and a general ischemia tic is shown for you on the slide. So the process here refers to a continuous time process. You are familiar with the schematic we have discussed this in early lectures of this course and then you have the actuator on the.. on the input site, which realizes the physical signal given the continuous time signal, so actuator is also working in continuous time. Then you have the D to A convert preceding the actuator, which connects the user to the continuous time. So the user designs a discrete time signal, as you know input design is a big… is a big topic in itself. So let us say the user has designed an input, typically the user design in discrete time, tells the system, digital system, that at these instance, these moves have to be made. And the D to A converter constructs an approximate continuous time signal, which is then realized by the actuator in the form of a physical signal that eventually excites the process and the process now responds in continuous time, which we then sense and sample and quantize. So there… there is a lot of things happening there, there is sensing, there is sampling, and then there is quantization, and then that gets recorded in the form of a measured output, right. What I am not showing you of course here are the disturbances and sensor noise, but we will not worry about those at the moment. The point of interest for… the… for us right now is what is a connection between this discrete time model that we have learned, ignoring all these elements? We just drew a block around this and we call that as a discrete time system, wrote down all the models, what is the connection between any of these models, discrete time models and a model for the continuous time process? Why do we want to know this connection?
(Refer Slide Time: 23:25)



That is why we… why do we want to know how the GD said for example, if I am looking a transfer function, how this is connected to let us say the transfer function of the LTI process in continuous time. Why am I interested in this map? If you don't like the transfer function language, you have ODE CI, this is your continuous time process and this is the discrete time process and you have DE CI, what is the map?

(Refer Slide Time: 24:06)



ODE stands for ordinary differential equation, DE stands for difference equation. Why do I care, why should I know, any good reason? Okay, so can I just live with the transfer function, discrete time, GD, G of Z or discrete time state space model or difference equation form and keep going, keep moving on in life?

(inaudible)

There is nothing sense of closedness, there is a discrete time, there is continuous time. What do you mean by closednesss? Any other reason, why we want to study this mapping? I should not just say, alright this is the part of your syllabus, therefore you should. You should be clear in your mind. Can you think of one good reason, that it can help you with, in identification? Suppose I tell you what the connection is between the continuous time process and this discretized process, how are you going to use it, how are you going to use that theory?

(inaudible)

Okay… okay but that discrete time model can also give you, right. It you will give you a sampling intents, not continuous time.

(Inaudible)

So what will you do with that, will you just, you know, share it on Whatsapp and say look I have great piece of information, let's celebrate.

(inaudible)

Okay, but how… I don't know, I am not sure if that information you can gather or, not enough clarity in that answer. What about you?

(inaudible)

Control, no let's not get into control, but that connection is a bit farfetched. Any other ideas? First of all do you understand what is this map that we are going to develop, when we say that we are going to establish a connection between the continuous time process and the discretized one, what is the image… what is the imagination that you have in your mind, am I going to give you some mathematical formula between G of S and G of Z, is that what you are expecting, or you are expecting something else. One standard answer you could have straightaway given is, given the discrete time transfer function if my objective is to find out, if my ultimate objective is to find out what the continuous time process is, this mapping will help me recover that, obviously. You could have straightaway given that answer. I am surprised you have not given me that answer. We have been, in fact I said this in one of the early lectures, that in many applications, it is a continuous time model that the user is seeking, not necessarily the discrete time model, that's only an intermediary. So if I want a continuous time model from the identified discrete time model, I need to know the mapping, right. So that I can go back, of course you still have to answer the question, whether the mapping is unique, that still remains to be answered, but that's okay. At least in the first place, I need to know the mapping. Then I can at least recover the continuous time model, but I would be doing that in an indirect way, because I would be identifying the discrete time model and then using the inverse map to find out the GC, that is the continuous time or G of S. In fact a lot of algorithms rely on this mapping, whether they indirectly identify, crude methods will involve indirect identification, that is you identify the discrete time model first, discretized model first, and then use the mapping the figure out what is the continuous time model. Advanced and good methods for continuous time identification, use this mapping as a part of the optimization, as a part of the estimation algorithm and you should really refer to this literature because of late we have been also looking at, begun to look at continuous time identification and it turns out there is a lot of literature and you will find that there are some excellent results and algorithms that are available for continuous time identification that can overcome some of the limitations that discrete time model identification algorithms have. We will not discuss those algorithms, we will not, we may mention those limitations, but we will not mention… we will not discuss those algorithms, because as I said, that is not really run of the mill as of now, but 10 years down the line, if you come back and if you happen to see whether students are facing the same difficulty as you are currently facing and you just want to sit in the class, I may be teaching continuous time identification. Okay you just want to see, if I have become more kind, so you see that I will be teaching continuous time identification. So all of this rest on this mapping, that is the first reason. Second a very important reason, why we should study this map is because I would like to know, partly the answer was given, whether I have loss of information or I have preserved the information. How could I lose information? What is the operation in this blog diagram that you see, which can result in the loss of operation? Sampling, correct. But you have to elaborate a bit more. You have jus said sampling can result in loss of information. Can you give an example.

(Inaudible)

Sorry if the…

(inaudible)

Sampling rate, you mean, okay so roughly now… so when you say dynamics of the continuous time process, you are… So you see now you are relating, right. So there is a sampling rate and your sampling rate has to be commensurate with the dynamics of the continuous time process. What is commensurate, we will understand shortly, but qualitatively we know, if I observe a very fast process in a slow fashion, I am going to lose out information, right. If there is a rat that’s moving around and I observe it every half an hour, I will miss out all the dynamics easily. In fact you saw this rat here, half an hour later it may be in other part of the city, it just moves so fast. Whereas, sorry… whereas if you are looking at a snail, the movement of a snail, then it's okay, you are not going to miss much, right, you can go have a shower come back, have your breakfast, it could have moved a few inches. That’s why we say at a snail's pace, right. What this tells us is, the sampling rate has to be commensurate with the dynamics. Now I would like to establish a quantitative relationship between the dynamics of the continuous time process and the sampling rate. And make sure that I am choosing some health sampling rates, because we can keep saying qualitatively, if this is, if the process is fast observe more frequently, if the process is slow, observe less frequently, that’s our qualitative rules, but we want some quantitative relationships and that is what this mapping will also help us, right. At least in two different ways we will find this mapping useful, one in recovering the continuous time process, which may or may not be of

interest in this course, predominantly it isn't, but occasionally if we want to talk about the continuous time process, we may explore and exploit this mapping, but more importantly choosing an appropriate sampling rate. If I give you an experimental setup and I say okay, go ahead and perform an experiment for identification, you have to make some important decisions and one of that decision is sampling rate, for that you need to know the mathematical relation between the sampling rate and the dynamics of the process. So at least these are the two top reasons, why this mapping is going to be used. Now in studying this mapping, we shall ignore the actuator dynamics, that is first point that I want to make, we will also ignore the sensor dynamics, that is not going to cause a major problem for you, because if there are those dynamics, you can always club that with the process and redo your exercise, so it's not going to really take away much by assuming or neglecting the actuator sensor dynamics, clear, and we will also ignore the quantization effects for now, quantization can also bring in some effects, remember, quantization is an irrecoverable kind of operation, you know what quantization is, right, take a continuous time signal, chop it off, and I say bin them into some levels, we will ignore those effects. We will simply now focus on the continuous time process and the sample and hold operation, that means, we will only look at this A to D converter as a sampler and we will look at this D to A convert as some kind of a hold device, neglecting the actuator and sensor dynamics and ask how the continuous time process maps on to the discretized process. Now all of this should not sound as very novel and alien to you, because discretizing continuous time equation, differential equations has been something that you have been doing all and all. You have done it in your first year, may be even in your senior, you know, high school, you must have done it, where I give you a continuous time OD… I give you an OD, and I ask you to numerically integrate it. When you cannot do the analytical integration, how do you perform the numerical integration by writing an approximate discretized formula, either using Euler's backward rule, forward rule, whatever, there are number of ways in which you can discretize. So you can think of this as another discretization method that is relevant to sampler and hold, that is now you are given that there is a sampler, there is a hold, in those discretizations, all you know is that I have to come up with an approximate discretized form, and you chose your method. But here the setup is different and therefore you want to stud this discretization in a different way, okay. So remember again that we are looking at discretized systems, whereas you can have discrete time systems by themselves, which can even occur naturally.
(Refer Slide Time: 36:14).

Discretization

Remember: Discrete-time system can also occur naturally (*e.g.*, salary of an individual, stock price series, rainfall). We exclude such systems from our analysis.

On several occasions, it may be required to identify the underlying *continuous-time* process $G_c(s)$ from the identified *discrete-time* model $G_d(z)^1$. Then, it is necessary to know the mapping from the continuous-time space to the discrete-time space and vice versa. Further, this mapping throws light on how we should choose the sampling interval / rate.