Foundations of Cyber Physical Systems

Prof. Soumyajit Dey

Department of Computer Science and Engineering

Indian Institute of Technology, Kharagpur

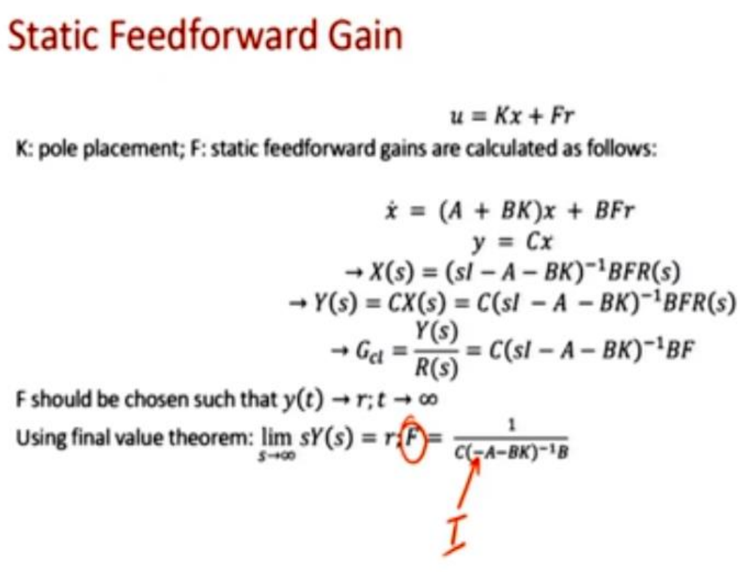
Module No # 05

Lecture No # 24

Delay-aware Design; Platform effect on Stability/Performance

Hello and welcome back to this lecture series on the foundations of cyber-physical systems. So if you recall in the last week we talked about continuous control system design and this week we will be moving on to discrete control system design which is a more common case when we talk about cyber-physical systems because here controllers are software ah mapped to real time processors etc. So before moving on just one small thing we like to point out.

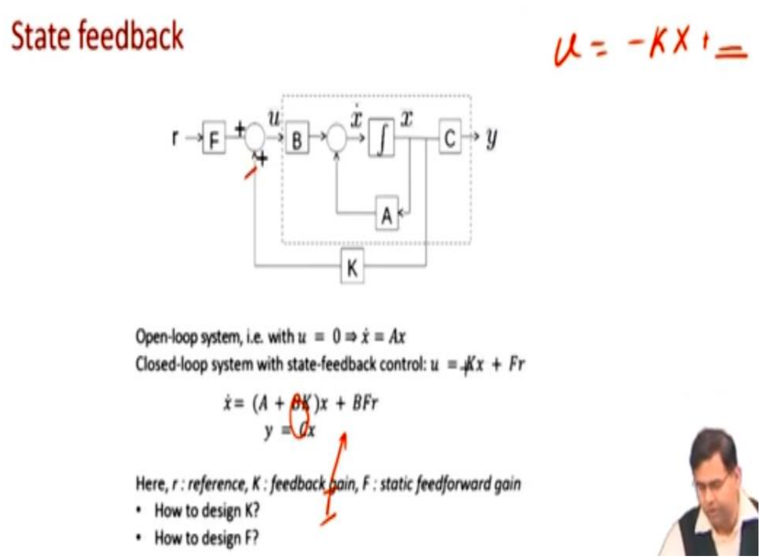
(Refer Slide Time: 00:59)



If you remember from our last lecture we said that the final derivation of the feed-forward gain and here I believe this identity a matrix was missing here in the last equation. So that is the only thing so just in a nutshell, what we discussed here was a continuous system, how to design controllers. So, in general, for an input system here, when you are talking about control design we arrive at some existing methods which can be used to design controllers. It could be used by direct substitution or it can be just this Ackermann's formula-based method through which we can do pole placement best control design.

We can design a controller in such a way that it forces the closed-loop poles below to be located at certain desired locations, which again creates the target system response, whatever nice response we are targeting. And in the process if we also have a reference signal, then we need to design this feed forward signal feed forward gain. If you remember our initial block picture which this 1. So, eventually, what we did was we created this final expression for feed forward gain design, which was this expression $\lim_{n \to \infty} s\,Y(s) = r; F = \frac{1}{c(-A-BK)^{-1}\,B}$. And we said that well this is how you can talk about feed forward and feedback controllers, and the thing was that well this is how you design the K and the s and also one small thing.

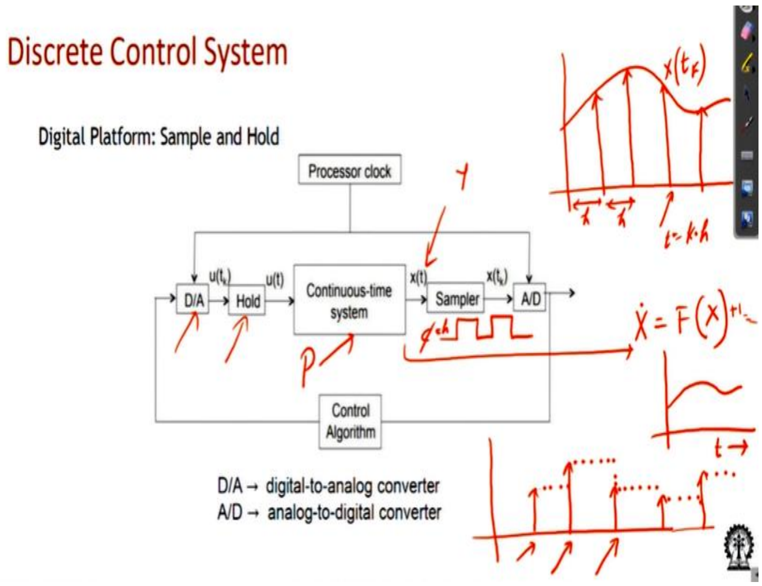
(Refer Slide Time: 02:54)



If you remember us in our derivations, we talked about this model and it is up to your choice for deriving the algebraic expressions typically in books you will find what we do is, we consider the feedback loop to be designed in such a way is well that the control equation is u - K x + the reference signal. And that is how we created this design here for this part, we modified it here so that is about it. So with this we have got a closure to the continuous time control design and we may just go forward to this week's topic, which is designing the controller for digital implementation.

So, for this, the first thing we need to talk about is how really, you discrete the continuous time models for the plants to a discrete type model? Because if you remember in our previous lectures we had a continuous time model and corresponding with the controllers there was also continuous time right. But now if we are going to design controllers which mimic control software which is periodically actuating, then the plant model also needs to be discretized.

(Refer Slide Time: 04:24)



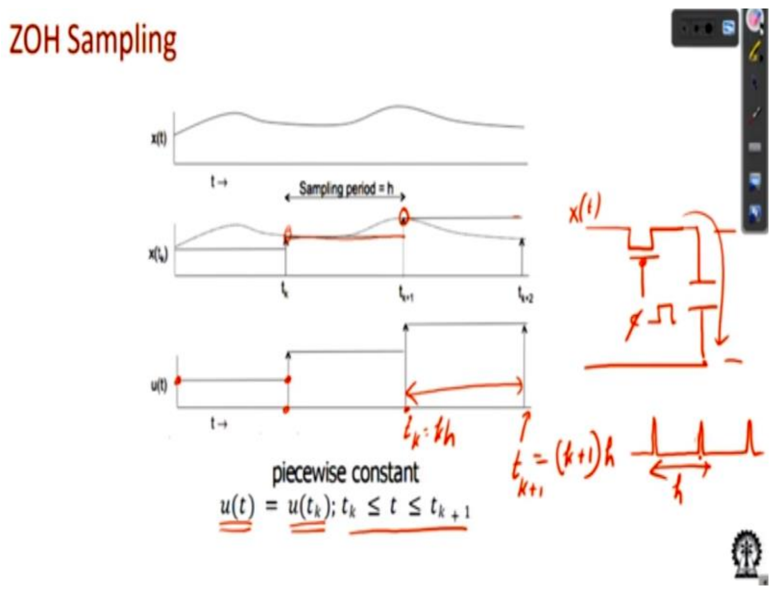
Because what do I really mean by a discretized plant model, let us understand. So, when you have a continuous plant you have this differential equation. Which is giving you the plant's behavior exactly at all real time points, but now what you are going to have is a digital platform controller implementation where you have this as your plant, which is fine, which is being modeled by some equation of this form, maybe.

But we are going to introduce this control term and other stuff which will come with the controller design. But here this continuous plant output Y or X, whatever is the way you write, will be sampled right with a clock of some periodicity h. So that here when you get the sampled output it would be like a sequence of values. So let us say here the response is that the measurements are this, but what you are doing is the same. There is a sampler circuit which samples these responses at specific points with some periodicity. So let us say this is a k h sample sum $X(t_k)$ is how you write it. So, instead of having a continuous time signal, you are having a discrete signal which is defined as those specific sampling points separated by the sampling period and those are the values for which you do A to D conversion.

You go to the control algorithm and then finally you do the digital to analog conversion and you get the control signal. And so, now your control signal is also like sampled values. So you get u values which are updated with this period, so that means this u value will sustain from here up to here. Then this u value will be held forward by the controller from here up to here. That means the actuator will hold on to this u value here and then, with the next control of update the u value will again continue holding to this new updated value unless another value is computed and that is how it works.

So here we will have a hold circuit which will transfer which will kind of transform these periodic control updates which are happening, and the hold circuit will hold that value constant up to that sampling period before the next control update comes which may or may not change the previous control input that was being held by this hold circuit. So how typically do you have the digital equivalent of control systems implementation.

(Refer Slide Time: 07:26)



So this is how we pictorially model it, so this was your original x(t) and then when we have a sampler circuit, then that is going to sample this. So let us say here you have this x coming in this line and you have a sampler right through which you are sending a clock signal. So I am just this, of course, not a real circuit here, so we are just saying that you have a clock signal. Whenever it is going high on the edge of the clock, you are getting the value, of course. This is again. I am repeating. This is just a symbolic representation.

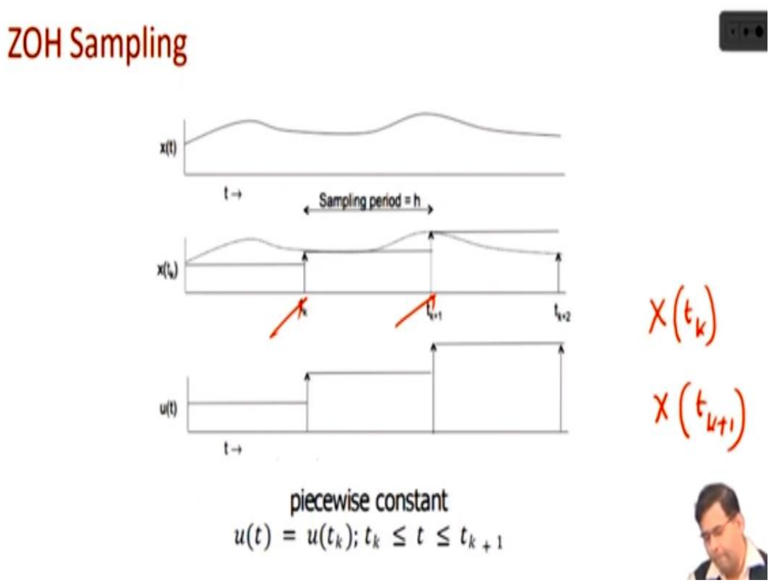
This is not really a sampler circuit. So, what you are really sampling from that is these specific values and you are going to work with these specific values. That means you may have a sample and hold circuit which is connected to this analog signal. Now let us draw a real circuit here, so what you may have is that this is driven, this is driving x(t) and here you may have a gate which is being driven by this clock signal and this output goes to a capacitor. So, this clock signal is being driven to this gate of this MOSFET with this clock periodicity h and whenever this gate closes, this capacitor will be charged and the capacitor will try to hold on to that value. So this is the point where it is instantaneously charged.

It is going to hold to this value and, again, the next clock pulse will come right. So it is like a clock pulse coming, then again a pulse coming and the periodicity is h. That is how it is working and this is the duty cycle, which is quite small in that case, so if the gate is just closing with the edge of the clock, and then the pulse is down. And we going to get another clock cycle another follow upward edge of a small duration and these 2 successive clock going high situations are happening with this periodicity of h. So again, in this next period, you will sample the value of x here and then you are going to hold it here. So that is how this sample and hold will typically operate and this output will go to analog to digital controller, analog to digital converter. So essentially, this is an analog signal here also you have an analog signal right but that is kind of being held to a constant value for a specific interval and inside this interval the control loop has to update of the control input.

Now, similarly, this controller which is producing this output, like we said, that this output is being produced at this specific period. I mean periods, in specific instances separated by a regular interval and those outputs will be held at the constant value using the hold circuit. So now coming to the mathematics of the system, what you really have is a control signal which is piecewise constant, that means you have u(t) which was a continuous signal being modeled as $u(t_k)$. Where for this entire interval, let us say the kth time instant $t_k = k\,h$ and here you have t_{k+1}, k+1 times h inside this interval you have a guarantee that this value will be constant. So that it is a piecewise constant signal and that is how your digital controller this digital implementation will work. What you have here in the abstraction process is that you lose the continuous modifications of x that you had in the continuous model of the system.

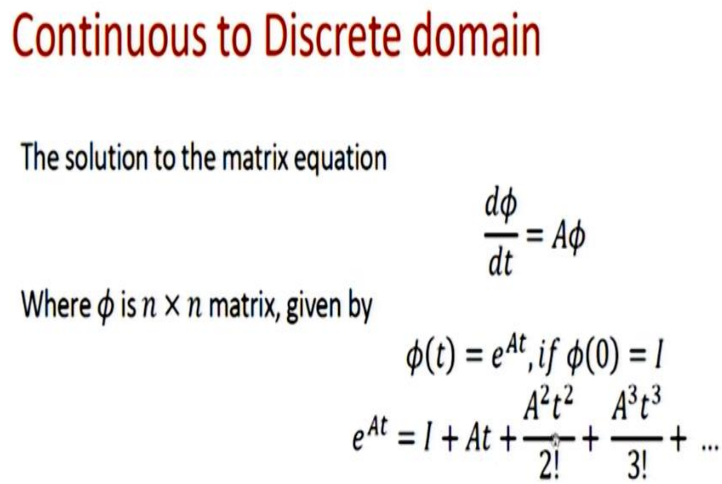
So and accordingly your controller design has to be different, because it should not work with the original continuous type model. Because what you now get is a discrete time system. So you have to create the corresponding discrete type model and you should design your controller corresponding to that discrete type model. So that, for that, the sample system, you are able to derive a continuous corresponding control update. So we need to understand that well for this continuous system when I am sampling it, how really we can derive this discrete time model? By the way, what will the discrete type model tell me?

(Refer Slide Time: 12:22)



Instead of giving this evolution of X continuously, the discrete time model should be evaluated at this specific time point. So I should be able to know what Is T k by querying the discrete time model at t_k. I should be able to know $x(t_{k+1})$ by again querying the model for $t = t_{k+1}$ = K + 1 times h like that.

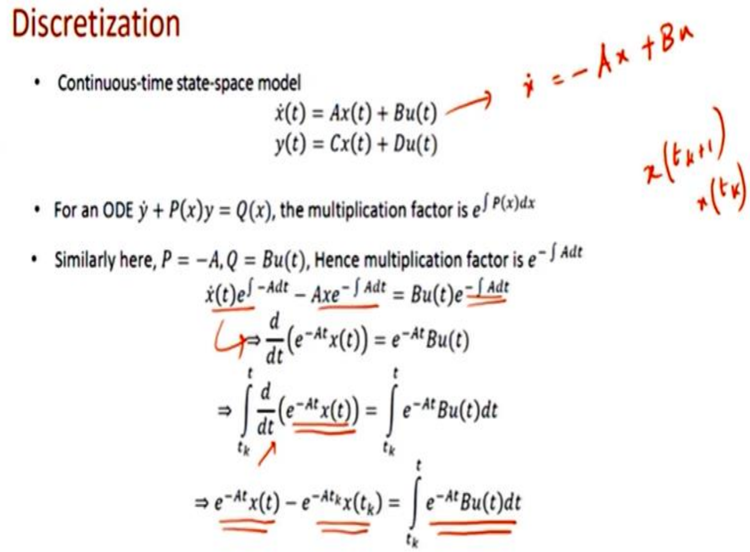
(Refer Slide Time: 12:54)



So for this let us get into some related mathematical machinery so if you recall if you have matrix A, and if you have this equation like this d Phi d t is equal to some matrix A times Phi and for this solution is like this. So Phi(t), Phi being a varying output t should, would be to the power A t so that the standard solution of this well-known form of differential equation considering that of course Phi 0 is I.

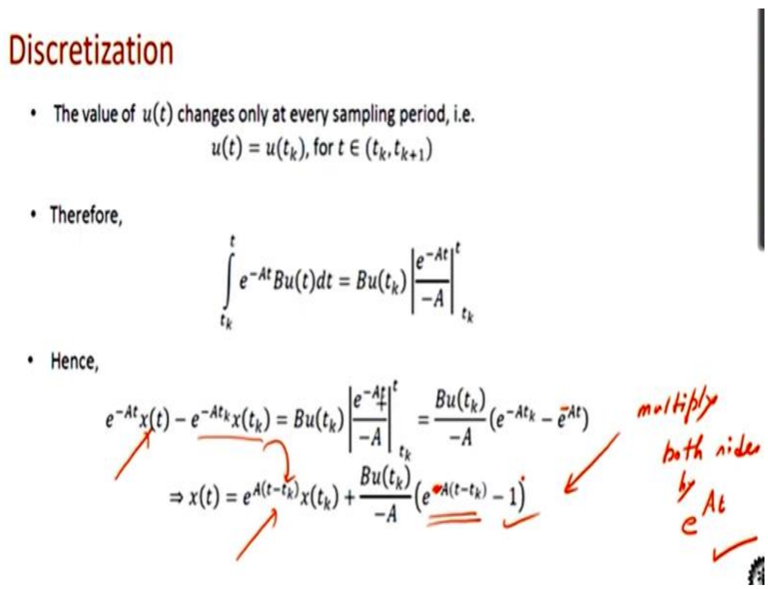
And of course we know that e t e to the power A t has this standard expression right, that is identity matrix + A t. Note that A is not a scalar but a vector but a matrix there and n cross n matrix so I + A t + s square t square by 2 factorial + A cube t cube by 3 factorial like that so that is the form of e to the power t. And given this differential form you can write Phi t's formula like this $\emptyset(t) = e^{At}, if\ \emptyset(0) = 1$. So we are discussing it because we will be using this formula setting here.

(Refer Slide Time: 14:01)



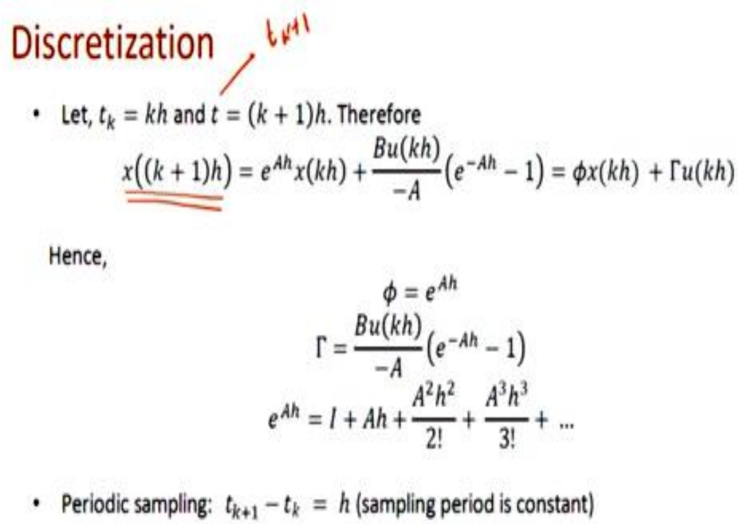
So for this let us get into some related mathematical machinery so if you recall if you have matrix A and if you have this equation like this d Phi d t equal to some matrix A times Phi and for this the solution is like this so Phi t Phi being a time varying output it should it would be to the power A t so that the standard solution of this well known form of differential equation considering that of course Phi 0 is I and of course. We know that e t e to the power a t has this standard expression right that is identity matrix + a t note that a is not a scalar but a vector but a matrix there and n cross n matrix so I + A t + s square t square by 2 factorial + A cube t cube by 3 factorial like that so that is the form of e to the power t and given this differential form you can write Phi t's formula like this so we are discussing it because we will be using this formula setting here So you start with this continuous time state space model and let us also recall one popular OD we which we may have studied in our + 12 + 12 calculus so if you have this form of a differential ordinary differential equation which is like y dot + being a polynomial of x multiplied by y = Q x then for integrating this kind of differential equation form you multiply this with a factor you multiply all the terms with e to the power integral p x d x right. So similarly here if you see the first equation as that form with a being equal to - A and Q being equal to well B u t right that is your form here right. So you have this x dot = - A x + B u you can always write like that well if x and your time varying, so that is why we have this form right. And then we will let us take this form and you multiply both sides with e to the power - integral of A d t. So if you do that then you see the first the 2 terms can be written in this nice form that is its differential of e to the power - A t x(t) which is of course fine. Because the first term will be I mean here in the first term you apply the differential on x and in the next term you apply the differential on e to the power - A t right. I mean you apply it only to the integral of this right so that would be that so that is it in that way it comes. So from here if you look at it so this is of course I mean integral of A dt is nothing but e to the power - A t right I mean with the integration from 0 to sometime t. So we can we can always write it like this and the good thing is now for this specific form you can now take an integral on both sides from some point t k to t right. And since inside the integral you already have this differential so, you can directly take this as the solution on the left hand side. And you can write it to be difference of the solution evaluated at t and the value of the solution evaluated at t = t k and on the right hand side you will have this to be solved out. So what we are really doing is? We are saying that well we have this continuous form but what we want is a form which relates $x(t_{k+1})$ with $x(t_k)$ right we want to go to such a form. Because that would be the set of equations which model the system in the discrete time domain of t_k, t_{k+1} like that right. So fine let us proceed like that. So you have this solved out on the left hand side and from that you can also make one interesting observation exploiting the property of this digital controller implementation.

(Refer Slide Time: 17:34)



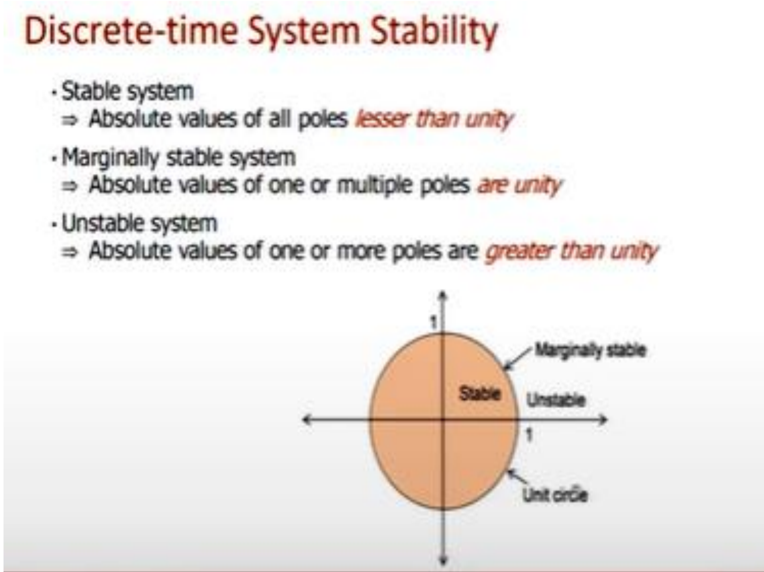
Since I am sampling the system and I am also updating the control only at the sampling period. So I can say that this value $u(t_k)$ is, it is hello u t which is where t is any point between t_k and t_{k+1} is always equal to $u(t_k)$. Because that what we discussed and we also had drawn this figure right that the control input is getting updated only at the sampling period so inside this sampling period u is constant now why is that important? Because if you look at the left and right hand side here you; have this integral over e to the power -A t and u t right. But u t becoming constant you can just take this thing out so you have the integral over then only to the power A t which would then nicely give you to the parity by - A and you take this values at t and t_k. So then you can nicely write this thing that way you have B $u(t_k)$ divided by – A and e to the power - A t_k - e to the power A t right like that. And if we can continue so from this thing where the left hand side we have already seen which is this and the right hand side we have already seen we have evaluated it to be this right and we have simplified the right hand side to this. So then fine we can just we can just take this and from here for we have a way to solve for x(t) right by you just multiply every term by to the power A t. So then for $x(t_k)$ you have e to the power A t - t_k right. This is coming on the right hand side and you are multiplying everything by to the power A t so you get this and then B $u(t_k)$S divided by - A and since we have multiplied Everybody by, e to the power A t. So this becomes 1 and here you have e to the power - A t - t_k like that. So let me just mod make a small modification here so you have e to the power- A t right and - A is there so I believe this should be – A t this works out. So fine, with this you have the discretization done.

(Refer Slide Time: 21:15)



And so with this discretization now if you just start writing t_k as k h because that is the real time period for the instant t_k. And similarly let us say the next k + 1 th time instant is given by what is k + 1 times h right. So that is basically t k +1 so if we choose the time instance to be this. Then for t you are having k + 1 into h right and for t k you are having k into h right. So then this difference here is nothing but. So that gives you h here. So you get an e to the power h. And here you get x as a function of k h + B u(t k). So u(k h), - A land. Here you have e to the power - h because again this is k + 1 into h - k h so that is k(h – 1). So in that way you have 2 new matrices e to the power h you make it Phi and this 1 you consider it as gamma right. So in that way you have these 2 new representations of phi and gamma right. And you can just even simplify them by because e to the power h you can check take a first few terms right. And for those few terms you can just put them here and you can solve out phi and gamma. So then you have the discrete time representation of this continuous time plant.

(Refer Slide Time: 23:18)



Now one important thing is how do we analyze the stability of discrete time systems? So the idea is here in the discrete time you have this representation well you have x(k + 1) equal to this Phi times x k where Phi is e to the power h + gamma times u k where gamma is nothing but this expression. And so the way you are going to see the discrete time system is you consider the discrete time Z domain transfer function and see. If it is going to be stable then all the poles will be located inside the unit circle in terms of absolute value for marginally stable they will be located on the unit circle and for unstable there will be one or more poles which have absolute value that is outside the unit circle. So fine with this we will end this lecture. Thank you for your attention.