Advanced Aircraft Control Systems With MATLAB / Simulink

Prof. Prabhjeet Singh

Department of Aerospace Engineering

Indian Institute of Technology Kanpur

Lecture 50

MATLAB implementation of first order systems

Hello friends, I'm Dr. Prabhjeet Singh. From this week onwards, we'll be mostly working in MATLAB and Simulink environment. I hope you must have tried our simulated examples in MATLAB τ ght by Professor Dipak Giri. Now we'll be focusing on simulation of mass spring damper system. But before that, we need to quickly revise some concepts of first and second order systems and why do we need to do that, because mostly practical systems can be represented in first and second order systems, consider the example of an aircraft the six degree of freedom aircraft equations of motions are represented in first and second order form Although this form has been transformed from second order Newtonian dynamics by introducing additional state variables for velocity and orientation rates. Let us begin with the first order system. The representation of the scalar first order differential equation can be represented as

$$\frac{dy}{dt} = f(t, y,, u) \quad ... Eq(1)$$

where t is continuous time variable u equals to u of t represents system input f represents the derivative function y equals to y(t) represents the system output. Now consider a case where f is a function of input and output by

$$f(t, y,, u) = b_0 u(t) - a_0 y(t) \quad ... Eq(2)$$

So, here a_0 and b_0 are considered as constants. We can write

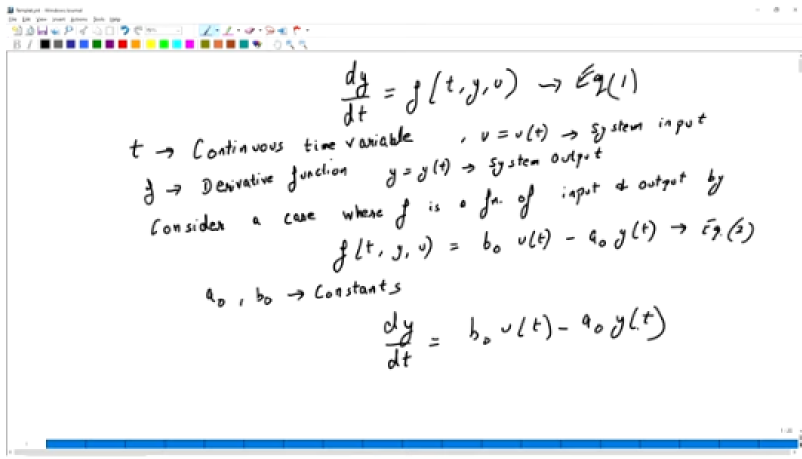
$$\frac{dy}{dt} = b_0 u(t) - a_0 y(t)$$

Now since a_0 b_0 are constants and the behavior is linear the equation is said to be LTI which is nothing but linear time invariant Ordinary differential equation ODE that is ordinary differential equation. Now we can write the above equation as

$$\dot{y} + a_0 y = b_0 u$$

I am dropping the t term for convenience.

(Refer Slide Time: 03:29)

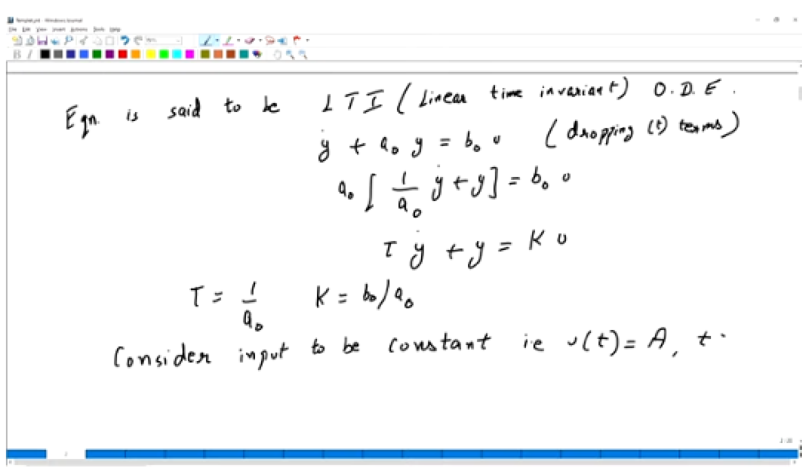


$$\tau\dot{y} + y = Ku$$

Where $\tau = \frac{1}{a_0}$ and $K = \frac{b_0}{a_0}$

We will understand the significance of τ and k in a minute. Now, consider an input to be a constant that is $u(t) = A$. In terms of aircraft, you can assume the elevator is for example, elevator is around 2 degrees for all time greater than zero then the solution is obtained by either using classical time domain methods or by Laplace transform now let us learn how to you know the differentiation in this equation in MATLAB now let us switch to MATLAB now this is how the MATLAB environment looks like I have already written equations. I will not rewrite it again to save time. Now this is CLC clear screen close all.

(Refer Slide Time: 05:52)



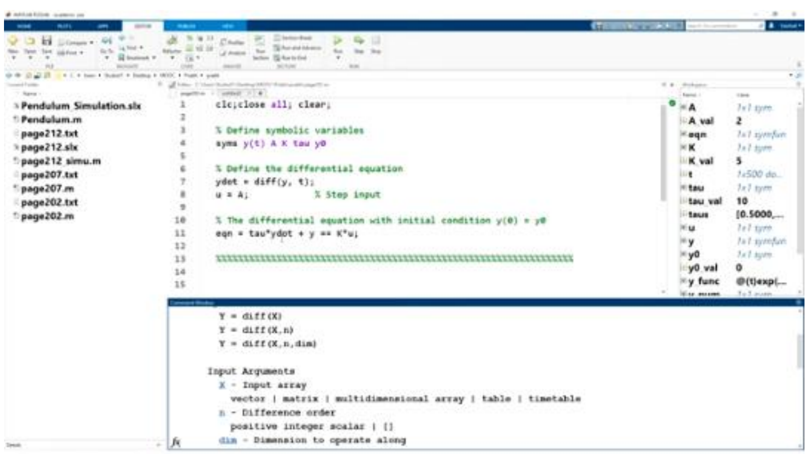
It clears all the variables and closes any figures if there are any open figures. And I am defining the symbolic variables SYMS. The variables are Y of T, A, K, T , and Y naught. Then I am defining the differential equation y dot equals to DIFF y comma t. The syntax for defining any differential equation is DIFF. If you want to know more information about DIFF, you can simply write in the command window: help DIFF.

You can also open the documentation if you like. So this MATLAB function calculates the differences between adjacent elements of x. The syntax is DIFF of x. I have used here DIFF x, n. Here, x is the input array, and n is the difference order, a positive integer scalar. Next, I am defining the step input u equals to a. And finally, the differential equation with initial conditions. y of 0 equals to y naught. The equation is represented as

$$\tau\dot{y} + y = Ku$$

This is the way of defining any equation in the MATLAB environment. Now, we will find the solution of this differential equation directly in MATLAB. Solve the differential equation with the given initial condition. I have written Y solution equals to dsolve. Again, if you want to know information about dsolve, you can look it up here. Solves the system of differential equations.

(Refer Slide Time: 08:42)



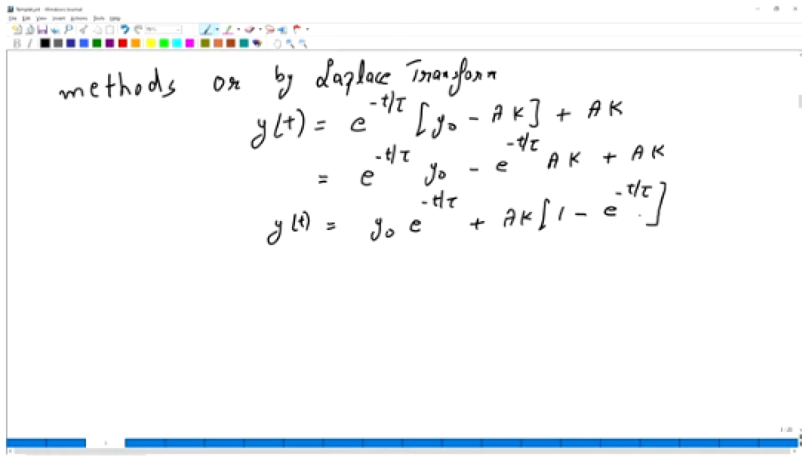
This function solves the differential equation where equation is a symbolic equation. I have removed semicolon here. I just want to show here what is the solution of this differential equation. If I run this program, then the solution comes out to be

$$y(t) = e^{-\frac{t}{\tau}}[y_0 - AK] + AK$$

$$= y_0 e^{-\frac{t}{\tau}} + AK\left[1 - e^{-\frac{t}{\tau}}\right]$$

Now what we will do is we will plot this solution in MATLAB for different values of τ .

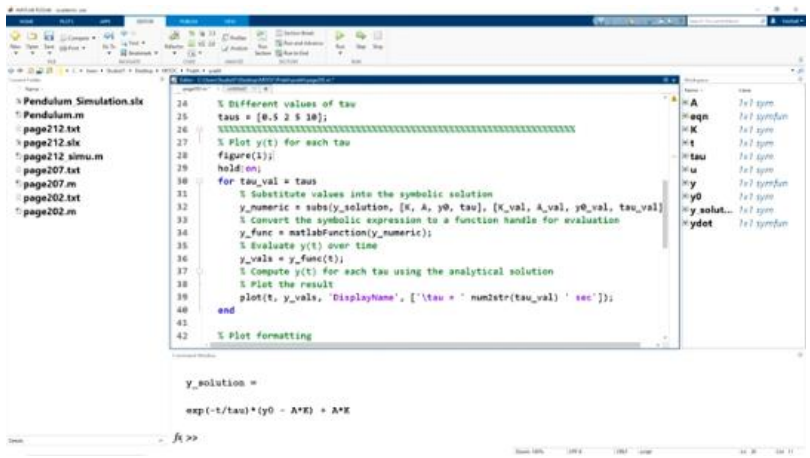
(Refer Slide Time: 10:37)



Let us switch back to MATLAB again. Now I've considered few variables here. Here gain k is given as five, a is given as two, y naught initial condition is zero, and the time range is from zero to 50 seconds with the frequency of 500, and different values of τ I've taken as 0.5, two, five, and 10. Now I'm trying to plot this response. I've named it as figure one,

Hold on, then I have written a for loop for τ value equals to τ s, which is 0.5 to 5 and 10, and then I am substituting these values into the symbolic solution. I have named the variable as y_numeric equals to subs. It will substitute in this equation y_solution. This is the old variable, and this is the new variable. Again, to get more information about these subs, you can simply write it here: help subs. This is the syntax: subs(s), that is the equation, then old and new. Then after this, we need to convert this symbolic expression to a function handle for evaluation. That can be done by simply writing MATLAB function and of bracket y_numeric. Then finally, we evaluate y(t) over time. y_val is equal to y_function of t, and then we plot t comma y_variable. We have here display name: τ equals to. I am converting this number to string, τ value represented in seconds.

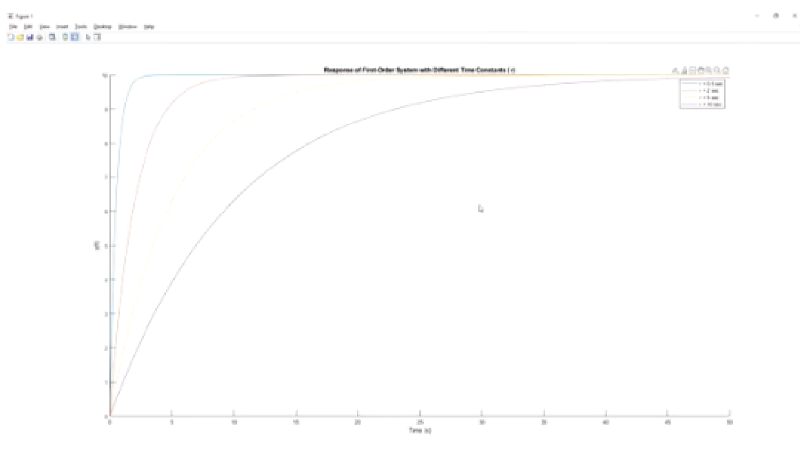
(Refer Slide Time: 11:20)



Then finally, I am plotting it. x_label as time second, y_label is in y, of t. Title is: response of first order system with different time constants. I am showing the legend as show. Grid is on. I am running this code.

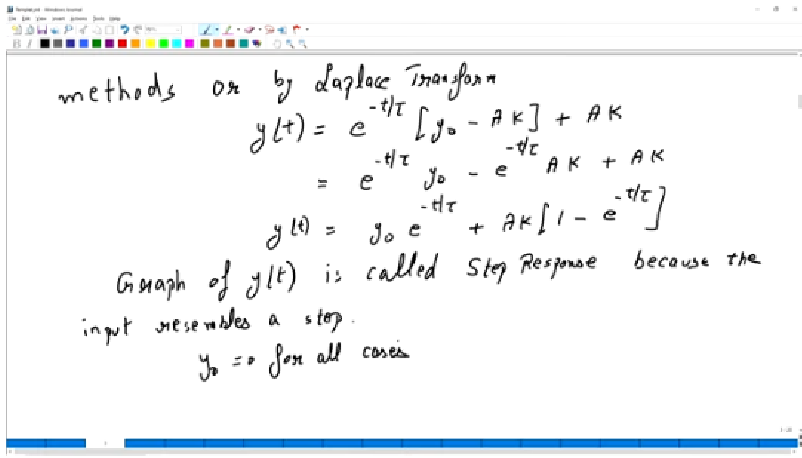
So this is what we get after running the code. So we observe here, we get different responses for different values of τ . Let me just explain the figure here. So this graph of y(t)is called step response because the input resembles a step response.

(Refer Slide Time: 13:00)



And also, we can note here that y(0) is 0 for all cases. Here, A represents the amplitude of the input, which is not a system parameter. k and τ are actually system parameters. That is, they are a function of a_0 and b_0. Here, k is called the steady-state gain.

(Refer Slide Time: 13:58)



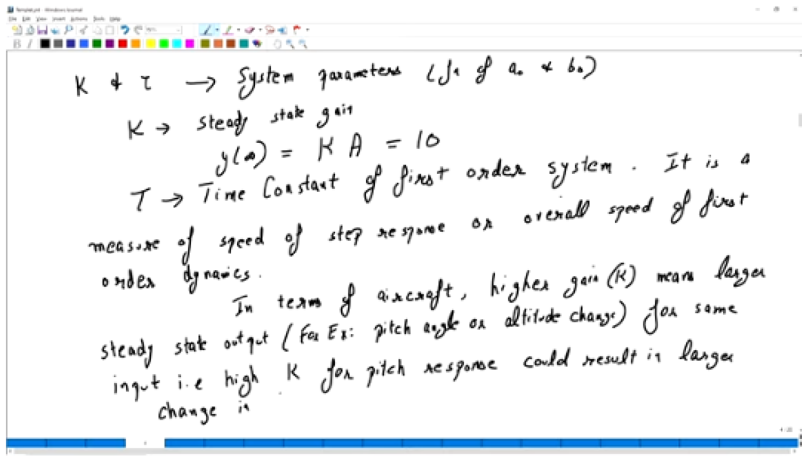
If we see y of infinity, if we substitute infinity in this solution of this equation, we get y naught into e to the power of minus infinity, and we know e to the power of minus infinity is 0. So,

$$y(\infty) = KA$$

So finally, what we get k into a, and I have considered the values of k and a as 2 and 5, so we get $y(\infty) = 10$ y infinity equals to 10. So we observe that the final value, if you see the plot here, we observe that the final value y infinity is unaffected by y naught. However, the graph starts from y naught. Another parameter is τ where the graph is different for all the four values. So there is a noticeable, if you see the figure here, so there is a noticeable difference in the amount of time required for the response to reach y infinity equals to 10. So we can call this τ parameter as the time constant, time constant of first-order system. It is a measure of the speed of step response or overall speed of first-order dynamics.

Now, in terms of aircraft, higher gain K means larger steady-state output. Here, output can be, for example, pitch angle or altitude change. So, higher gain k means larger steady-state output for the same input that is, high gain k for pitch response could result in a larger change in pitch angle per unit elevator input. Similarly, the time constant τ indicates

(Refer Slide Time: 18:53)

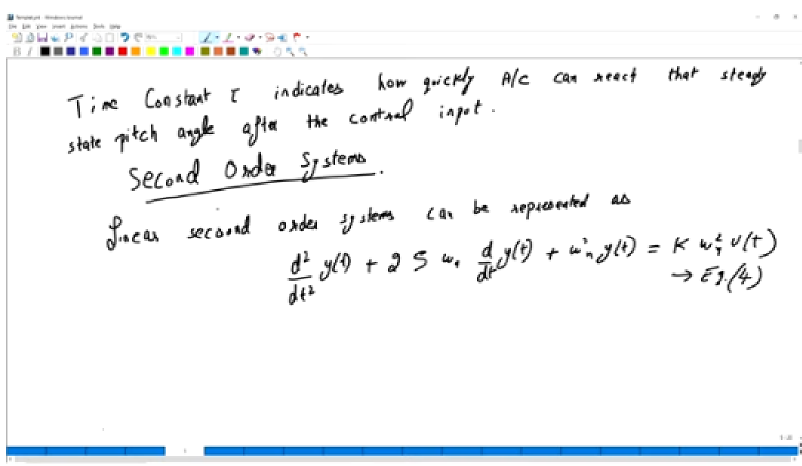


How quickly aircraft—I am representing aircraft as A by C—can reach that steady-state pitch angle. After the control input. Hence, we can infer that both k and τ are important parameters for defining the aircraft control systems. Now, we will move on to the second-order systems. So, linear second-order systems can be represented as

$$\frac{d^2}{dt^2}y(t) + 2\xi\omega_n\frac{d}{dt}y(t) + \omega_n^2\, y(t) = K\omega_n^2\, u(t) \quad \ldots Eq(3)$$

Now, the second-order systems were τ ght at length by Professor Dipak Giri in the first course. I will only touch upon the basics so that we can easily apply those equations in the MATLAB Simulink environment. So, we will continue from here.

(Refer Slide Time: 21:53)



We will stop it right now. We will continue from here in the next lecture. In the next lecture, we will mostly talk about the second-order systems. Thank you.