Advanced Aircraft Control Systems With MATLAB / Simulink

Prof. Prabhjeet Singh

Department of Aerospace Engineering

Indian Institute of Technology Kanpur

Lecture 53

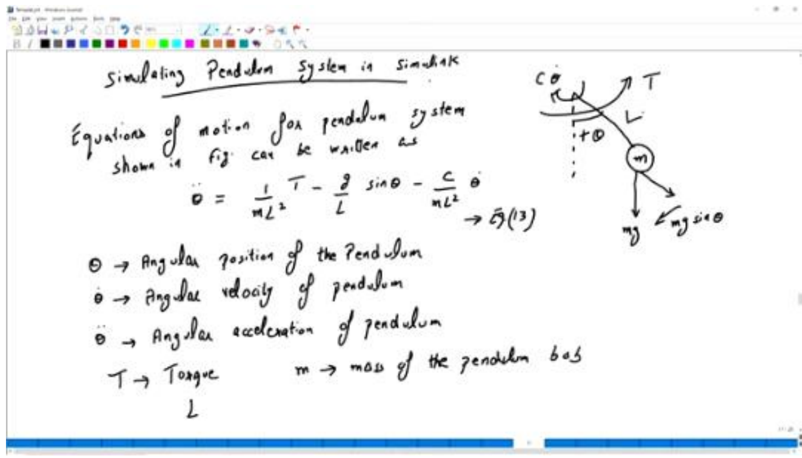
SIMULINK implementation of pendulum system

Welcome friends. Today we will discuss about or rather simulate pendulum system equations of motion in Simulink. If you are wondering which MATLAB version I am using, I am using 2024B. You also will be getting an online version where you can practice in MATLAB and Simulink. Now let us proceed with today's topic that is simulating pendulum motion in Simulink. So we will use different methodologies. First let me draw the figure here. This is a bob of mass m acted upon by the gravity force. Here we have mg sine theta in this direction. this is i am considering this as positive theta this damping will be in the opposite direction c theta dot and this is the torque so we know that the equations of motion for a pendulum system shown in figure can be written as

$$\ddot{\theta} = \frac{1}{mL^2}T - \frac{g}{L}\sin\theta - \frac{c}{mL^2}\dot{\theta} \quad \ldots Eq(13)$$

So, here theta is representing angular position of the pendulum theta dot is angular velocity of pendulum theta double dot is angular acceleration pendulum T represents the torque or the input to the system M is mass of the pendulum bob L this is the length L, L is length of pendulum that is from pivot to the center of mass of bob. And we are left with C, that is the damping coefficient, which is nothing but the resistive forces against motion. So, again I am not going how this equation has arrived that is not the intent of this course.

(Refer Slide Time: 04:06)



So, we will simulate this pendulum motion using different methodologies. First we will try with MATLAB function. So, before that we need to convert this second order system, second order equation to two first order equations. like we did in the last lecture let us follow the same procedure let

$$x_1 = \theta$$

$$x_2 = \dot{\theta}$$

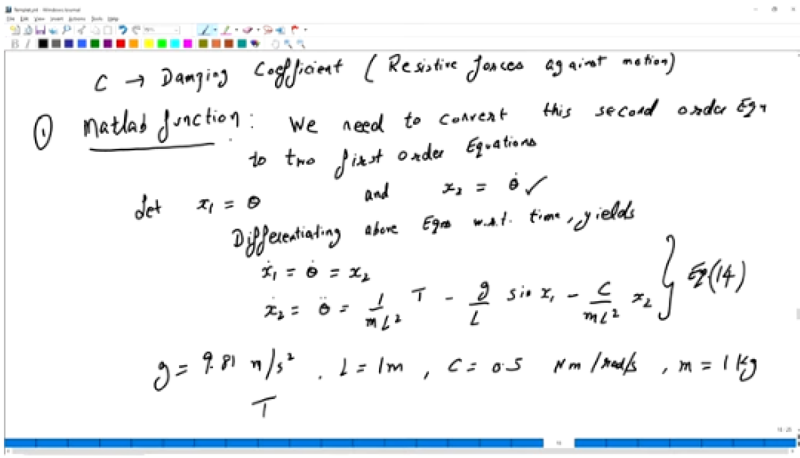
differentiating above equation with respect to time yields

$$\dot{x}_1 = \dot{\theta} = x_2$$

$$\dot{x}_2 = \frac{1}{mL^2}T - \frac{g}{L}\sin\theta - \frac{c}{mL^2}\dot{\theta} \quad \dots Eq(14)$$

Now for simplicity, let us consider the values. For simulation, we need values. Consider g as 9.81 meters per second squared. L as 1 meter. c equals 0.5 Newton meter per radian per second. Mass is 1 kg. Torque, I will consider it as 2 Newton meters. Now, let us jump to Simulink. Now, I have already modeled here and plotted the position and velocity using different methodologies. What we will do is reproduce the results using this Simulink block. So, I will start from scratch.

(Refer Slide Time: 08:21)

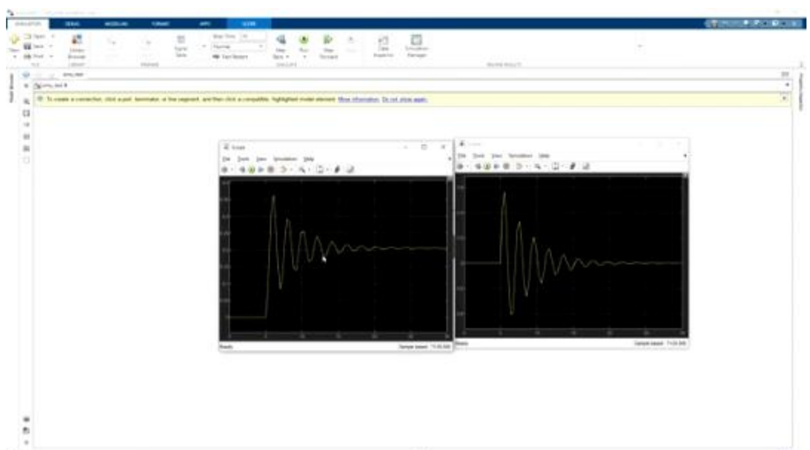


I will take it as a new file or You can go to Simulink either by clicking or writing here. Click on Create Model. And save this model in your working directory. I'm naming it as Simu test. Now let us use the first methodology that is the MATLAB function. I will start typing here in my keyboard MATLAB function. Here you can see the MATLAB function is appearing here. just click on this function and here we will write what is the input and what is the output that is required to run this MATLAB block so it appears automatically here u is the input and y is the output now here we have two outputs x_1 and x_2 this is the MATLAB function that I was talking about let me just copy this code here

So, here input is the x vector, this is x_1 and x_2 and T is the torque and g 9.81, L 1, c 0.5, M is 1. So, here since x is a vector, we need to write what is the first element of x and denoting it as x_1 , x$x_1$1 is x of 1, x_2 is x of 2. and these are the equations $\dot{x}_1 = x_2$ and $\dot{x}_2 = \frac{1}{mL^2}T - \frac{g}{L}\sin\theta - \frac{c}{mL^2}\dot{\theta}$ this is equation number 14. And what is the output that I have written here? Dot equals to $\dot{x}_1$ and $\dot{x}_2$. Now we actually need to integrate this equation. We simply write integrate. We get a block here, integrator. It is denoted by 1 upon S. Which is again fed back. In this manner. And it needs some initial conditions. Let me write the initial conditions as position and velocity. I'm assuming it starts from zero degrees. So right here, zero. Remember, MATLAB always considers the values in radians. If I enter in degrees, convert into radians. Into Pi By 180. And the velocity is 0. Transpose it. We have to write it in a column-wise manner. So these are the initial values. These are the initial values. And we need some torque here. Let me consider the simplest input function, which is a step function. We had written code for creating a step function in MATLAB. So in Simulink, we already have a block called the step block. Now consider a torque of 2 Newton meters, which is applied at 5 seconds. Apply and okay, let me run this for maybe 30 seconds. I am saving And let me see if I'm getting any response, or if it is indeed running or not. Yes, it is actually running. Now, to plot these responses. Now, here, X is a vector.

We need to see the plots for position and velocity. We will actually demux it. So it will demux into two vectors. I am right-clicking and dragging just to create a branch. So the first vector is actually position, and the second is velocity. To see that we have to write here scope. and again to copy this either you can do control c control v or right click this block and simply drag it now if i double click this so nothing is shown here because it is already run and i have plotted use this scope block later on i simply need to run this So here we have position or angular position and this is the velocity. So here we observe there is sort of sluggish response here. What we'll do is we'll increase its frequency. So here we can see after five seconds because of the control torque two Newton per meter. So there is an angular displacement here of about 0.2. And this is again in radians. First we'll convert it to degrees. You'll write here radians to degree.

(Refer Slide Time: 14:50)



That will be converting into degrees. Again right clicking this option and dragging it. As simple as that. So now actually we'll get the response in degrees. All right. Now, let us make this response a little bit smoother. For that, you have to go to Modeling, Model Settings, Model Settings. Right here, maybe Fixed Step Solver. Let us use the Runge-Kutta method and give the frequency as 100 hertz, 0.01. Apply. Okay. I am saving the file, and now let us run this code. It is ready, and we can see here at the bottom left. If you double-click this, now we have a smoother response for angular position as well as velocity. We can go ahead with the Print Display to Figure and get further options to see these data points. We can easily pan it out.

So, you are encouraged to explore these options. And there is another option to actually bring these variables directly to the workspace. So, for that, let me go to MATLAB here. Let me create another file. Let me copy-paste this code here and see here again. Actually, we have to define the variables: g equals 9.81, l c, and so on and so forth.

If you go to this MATLAB function this is a function actually if you have ever worked in function file in MATLAB we know that these variables will remain within this function file so these variables will not come out only the output variable will be dot other than that everything remains in this function hence if you are using a MATLAB or if you are trying to call a simulink from MATLAB we need to redefine these variables now tf is the time I am considering as 30 seconds so let us go here let me write here stop time instead of 30 tf dt 0.01 again go to modeling instead of right directly writing 0.01 I'll write here dt apply okay and x naught initial condition is 0 and 0 0 let me write here x_0 in the place of integrator Apply. OK. And the name of the file I have written is SIMU underscore test. Let me write here SIMU underscore test. Make sure both the files are in the same folder. So I am saving it. SIMU underscore test underscore MATLAB. Now what it will do is after coming to this line number 10, it will run this, it will look for the file name as simi underscore test and it will run that. So let us test it. So it was successful. T out is 300 by 1. But if you see this workspace, we haven't got the vectors x_1 and x_2.

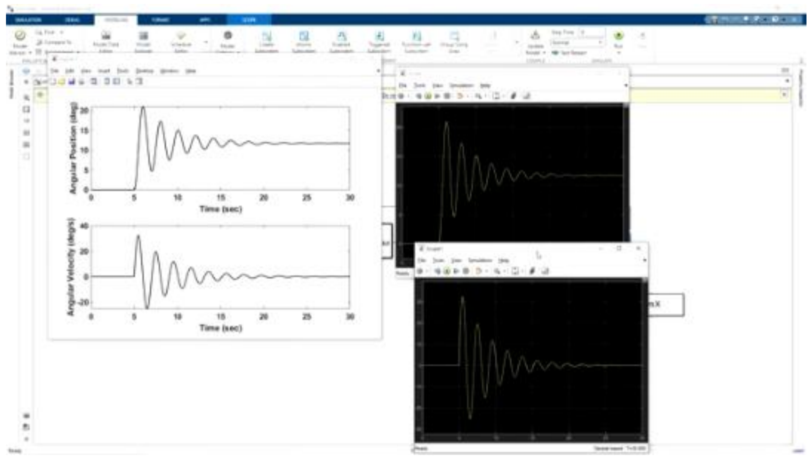
So, to do that, we need to send those variables to the workspace. We again go back to our Simulink model and type 'to workspace'. And we go back to the modeling settings. So, this output will not give me a vector. So, we need a vector of x_1 and x_2. So, to do that, just uncheck this box 'Single simulation output' and let me rename it as 'SimX'. All right. We again need to mux these signals. So, the opposite of demux is mux, M-U-X. So, I'll branch out this first signal here. I'm right-clicking the signal and then dragging and dropping here. And this is the output. Now, let me run the code directly from MATLAB and observe in the workspace. So now it has run successfully. Now we have the 'SimX' variable, but still, it is in this form. We cannot directly extract from here. So what we need to do is we write a simple code for extracting in x_1 and x_2 variables and the syntax for that is symx.data all the rows and the one column first column and the second column all the rows so now let me run this code figure one that is subplot two comma one comma one two rows one column that is first figure Similarly, two rows, one column, second figure. I have multiplied this by 180 by pi from radians to degrees. Similarly, radians to degrees.

Rest of the things remain same. Here, GCA represents get the current axis for defining the font size as 14 and font width as bold so that it is clearly visible. Now, let me run this file. Before that, let me close if there is any scope is opened here. All right.

So this is the one that I am getting the response here. Now let me check whether the similar response is observed here or no. So this is the position. And let me just bring that.

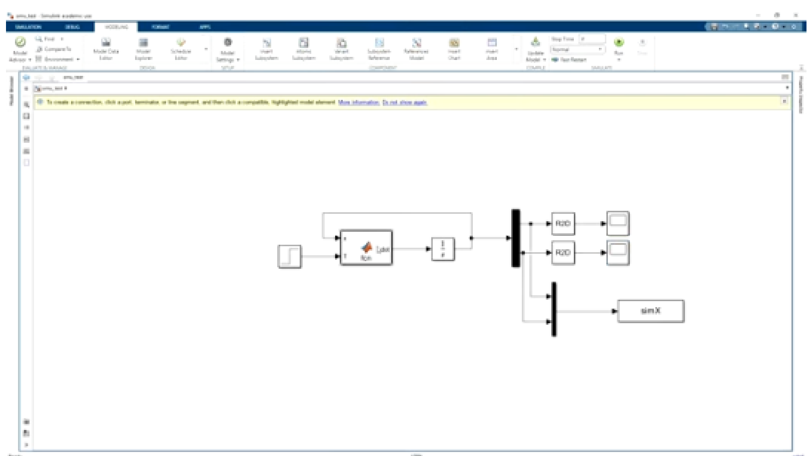
Yeah. So it is going up about 12 degrees in the angular position, and similarly, the angular velocity is going to 0, which we can see here as well. So both responses are similar. So there are different methodologies you can use. If you are interested in finding angular position or velocity, you can use either of the methodologies. All right.

(Refer Slide Time: 22:27)



Now let us keep this final value as zero. Let us see the response. Let me close this, go to MATLAB again, and run the file. See now, so now there is no response because there is no input. That is quite obvious, but still, many times students get confused. If that is the case, why do we have a response in position and velocity? Why is it not going to 0? Let me close this again.

(Refer Slide Time: 23:27)



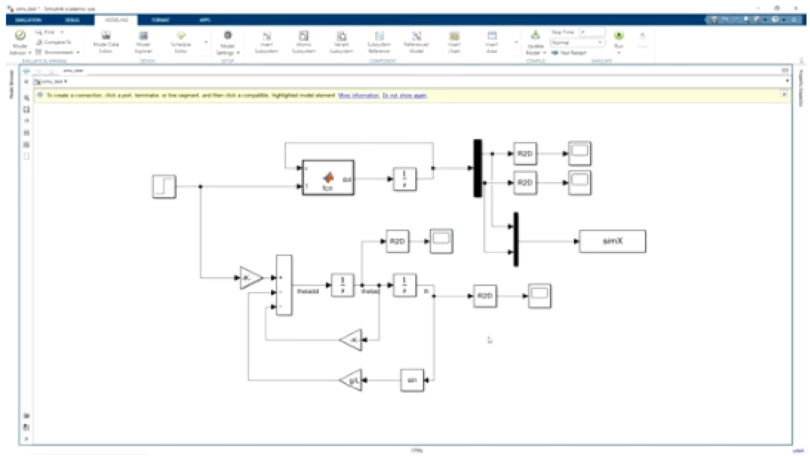
I am running the file. Yes, so here we have two responses. Here we have angular position settling around 11.8 degrees. All right, so here the response will not be zero because we have a constant input which is acting upon the pendulum. That is why we have some response. Now let us quickly obtain this similar response by another method. Remember, for the mass-spring-damper system, we have drawn a block diagram. So let us use that block diagram here. Here we have to integrate two times. So I am representing this as an integrator. One more integrator. So here the input signal is theta dot, theta double dot. This is theta dot, and this is theta. And here for $\ddot{\theta}$ theta double dot, we have three variables:

$$\frac{1}{mL^2}T - \frac{g}{L}\sin x_1 - \frac{c}{mL^2}x_2$$

So here, let me write this as plus. That will be for torque. This is minus, this is minus, since both are minus here. And consider this as again c by ml square. This is $\dot{\theta}$ c by mL square is multiplied by $\dot{\theta}$. c by mL square is multiplied by $\dot{\theta}$. $\dot{\theta}$is nothing but x_2 . All right. Plug this here. Then we have g by L, and this is multiplied by sine. What it will do is it will sign off this angle multiplied by g by L, and then this is subtracted from here. And here we have 1 by ML square. This is the gain here, which goes here, and this is the torque T. Now, let us plot this. Let us draw this block diagram directly in Simulink, all right. I'm right-clicking this integrator. We need two integrators. Let this be 0. There is no acceleration. This remains the same, x0. I'm writing some block here. We have three variables. Let me represent it by a rectangle, which is plus, minus, minus.

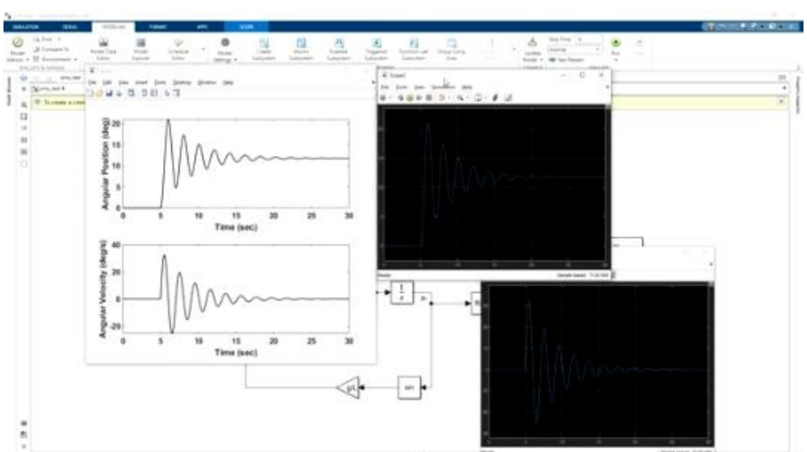
So this is nothing but theta dot, theta double dot, theta dot, and this is theta. So here we have to write gain. We need to flip this model gain format. Flip lock or control plus I. Next time, I will be using control I. Here I write c divided by ml square, m into l square. This is element-wise multiplication and then sine of theta control I. Sorry, g by L. This goes here, and this goes here. And here we have another gain, 1 by mL square. And the same torque I will provide here. Now let us directly plot scope. This is theta. We will convert it to degrees. Similarly, theta dot, that is velocity. We will convert that also to degrees. All right. I think we are more or less done. Let us run this code.

(Refer Slide Time: 29:43)



So here we have again position, angular position, and velocity. Let me go back to the Simulink model. Let us see. Are we getting a similar response? And this is the angular position, and this is the angular velocity. All right. Okay. Yeah, so we get almost a similar response with this block diagram methodology for angular position as well as angular velocity, which is going towards zero. So, you can use various methodologies to find or plot the response for y with respect to time. Next, in the next class, we will be using a state-space model and another method in Simulink. All right.

(Refer Slide Time: 30:15)



Thank you.