Advanced Aircraft Control Systems With MATLAB / Simulink

Prof. Prabhjeet Singh

Department of Aerospace Engineering

Indian Institute of Technology Kanpur

Lecture 54

SIMULINK implementation of pendulum system - continued

Hello friends, welcome back to our discussion or simulation on the pendulum system in Simulink. We will continue where we left off in the previous lecture. Now, let us go ahead with the different methodology to obtain the angular position and angular velocity using the state-space linear model. Notice I have written here 'linear' now. So, here the model is considered linear, which is a good approximation for small angles only. So, let us rewrite equation number 14. The state-space form here is written as

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

$$\dot{x}_2 = \frac{1}{mL^2}T - \frac{g}{L}x_1 - \frac{c}{mL^2}x_2$$

Notice we have x_1 here, not sine of x_1. So, what implications it will have in simulation, we will get to know (C/ML^2) into x2. So, writing in matrix form,

$$\dot{X} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{g}{L} & -\frac{c}{mL^2} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{mL^2} \end{bmatrix} T$$

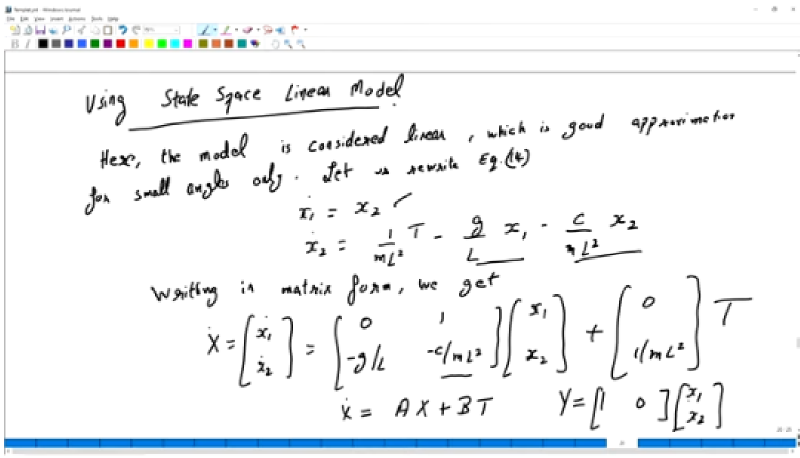
It is written as

$$\dot{X} = AX + BT$$

$$Y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Output, I am assuming only the position, anyhow D is 0.

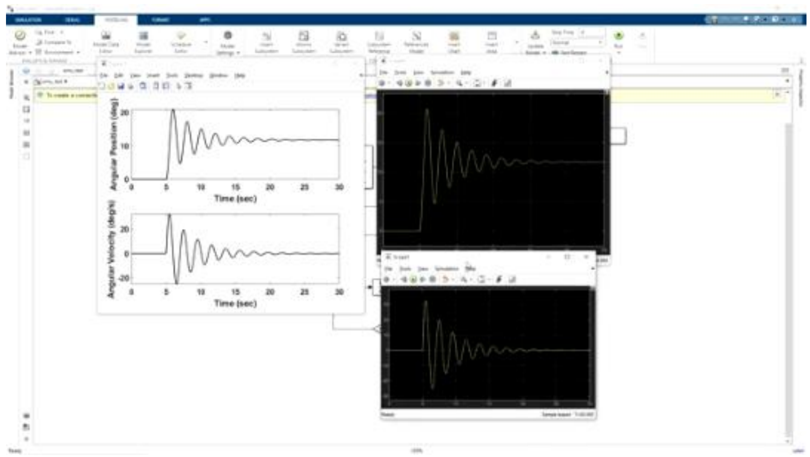
(Refer Slide Time: 03:52)



Now, let us draw a block diagram for this x well. So here we have only one integrator. Here, this is $\dot{X}$, which is yielding X. This is multiplied by $\dot{X} = AX + BT$ We will write here A. Now, remember this A is a matrix now. Here we have a summing block plus, and so, here the matrix is b, and here input is torque. So this is the block diagram for x dot equal to x plus b into t. Now, let us quickly plot this in Simulink. So till here, we have done in the previous lecture. Now, we have only one integrator. We have initial condition x naught. Writing gain k. Control I now. What is a matrix? A matrix is 0 1 0 1. We call it minus g by l minus c divided by M L square. M into L square. Now, this is a matrix here. So, it should not be an element-wise multiplication.

It has to be matrix into U because the output of this here is a vector. And this gain K will be multiplied by an input, which is a vector. Hence, it has to be matrix K into U. Then, B matrix. Here, B is given as 0, $1/ML^2$. So, this can be an element-wise matrix. Because here, the input to this block is a scalar, not a vector. So, I will just use the same torque here. One second. Here, we need to add the summing block. Now, this signal will go here, this one in this way. Now, this torque will be here, right? So now, here we have a vector. Again, we need to demux the signal so that we can easily plot x_1 and x_2, which are angular position and angular velocity. Right-clicking and simply dragging. All right. So here, we have initial conditions mentioned. This is the A matrix, and this is the P, likewise that we have done here. Now, we are ready to run our code. All right. Now, we have here Figure One. We go to our model. This is the angular position, and this is the angular velocity. Now, we again observe here that we get a similar response, similar results, irrespective of the methodology used. You can compare these plots. All right.

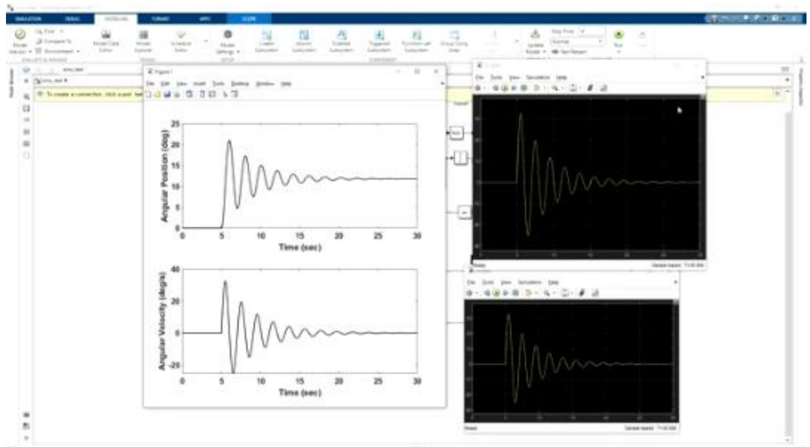
(Refer Slide Time: 10:05)



Now, let us use the same state-space block, that is, the inbuilt state-space block. So, MATLAB provides us this inbuilt state-space block. I'll just write 'state-space block.' If you double-click this, the equation is already in this format: X0 equals to X plus BU, Y equals to CX plus DU. And it will ask for the A matrix, B matrix, C, D, with the given initial conditions. So, the initial condition was X0. All right. It will not let me finish. We need to enter the matrices here. This is A. This is B, and then we have C matrix as well: 0, 1, and D 0. We have applied, okay. What we need to do here is again plug in this signal as an input to this block. All right. Let me close these figures. We'll just copy this code here.

And see if there is any problem. Control + S, run the code. Yes, so there is some issue here relative to multiple causes. Input port dimensions are being set to one. All right. We'll change this to 0, 0. Let us see now. All right. So similarly, D also needs to be changed. All right. Let me try now. Yeah, now if you plot this, this is the position, and this is the velocity. So there is some issue in velocity here. Let us see what that is. All right, so what we did here was we have written C matrix x 0 1. All right, let us write that here. So $Y = CX + DU$. Anyhow, D is 0, so this is Cx. So earlier, I had considered C as 1 and 0, but since x is a vector x_1, x_2, I had to mention the state space block 1, 0, 0, 0. So, to get the velocity vector, I need to change this C matrix to 1, 0, 0, 1. That is how, when this is multiplied by x1 and x2. So, y will be x_1 and x_2. So, this is 2 by 2, and this is 2 by 1. It will be 2 by 1: 2 rows, 1 column. Hope this is clear.

(Refer Slide Time: 14:48)



Let me change this to 1. I am saving it. Now, re-run the code. Alright. I didn't close this. Alright. Now, this is angular position. Now, we have angular velocity. Now, again, we get a similar response. Let me all right we did a little bit mistake here it should be 1 0 0 1 and rest of the things are ok so now if i run this code and check this position angular position angular velocity Now we indeed we get a similar response over here. Angular position and angular velocity. Now let us go ahead with our final methodology that is transfer function approach. We will get to know what is the limitation for using that approach. Let us proceed using transfer function approach. Now again taking Laplace transform of equation number 13 with zero initial conditions. so we have

$$S^2\theta(S) = \frac{1}{mL^2}T(S) - \frac{g}{L}\theta(S) - \frac{c}{mL^2}S\theta(S)$$

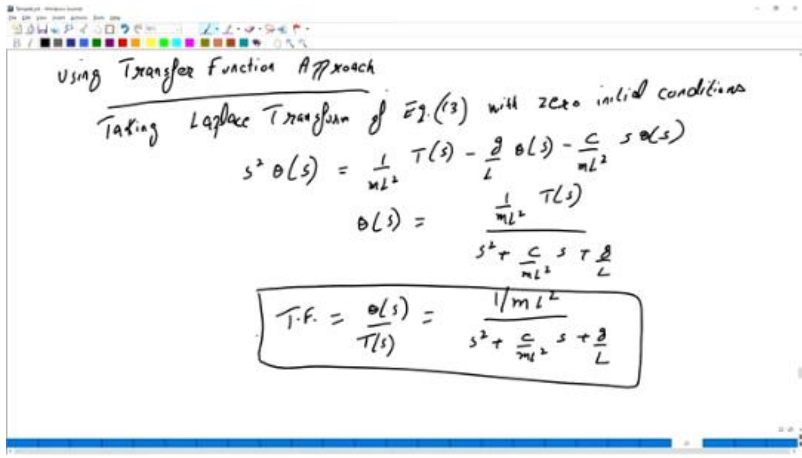
$$\theta(S) = \frac{\frac{1}{mL^2}}{S^2 + \frac{c}{mL^2}S + \frac{g}{L}}$$

So, transfer function can be written as

$$T.F. = \frac{\theta(S)}{T(S)} = \frac{\frac{1}{mL^2}}{S^2 + \frac{c}{mL^2}S + \frac{g}{L}}$$

So this is the transfer function. Now let us plot angular position and angular velocity using transfer function approach. I will write here transfer function. And we simply write what is actually in numerator coefficients. I will simply write from here.

(Refer Slide Time: 17:40)



I have already written earlier. Apply okay, like, and okay. All right, let me make it a little bit bigger. Okay, now this is a transfer function approach. We will put an input again here, which remains the same. I am not changing the input, hence expecting similar results. Radiance to degrees, and we have a scope. All right. I am saving it, and I am running the code. So, we get a similar angular position response here. How to get angular velocity? By actually doing the differentiation of this signal.

So, to do that, write the derivative of this signal, and then finally, I am copying it. All right. Control S, double-click, and run this model. Now we have velocity as well, which is going to 0. Now we have tried different methodologies: one is by using the MATLAB function, and then using this block diagram. You can write here actually by annotations. This is by MATLAB function. This is by block diagram. This is similarly state space approach even this one and this is the transfer function approach. alright so we have tried one two three four and five methodologies and we get similar results okay now let us do something different now let us see and plot with different initial conditions let me write here instead of 0 degrees I'm writing here 200 degrees So what I am actually doing here let me go to the figure all right so plus 200 degrees means it will the initial condition will be here so this will be 180 plus 20 degrees so 200 degrees. So when you drop it from here, then you expect this to go in this way and then oscillate like this. That means it will, after reaching 200 degrees or initial condition is 200 degrees, it should go further.

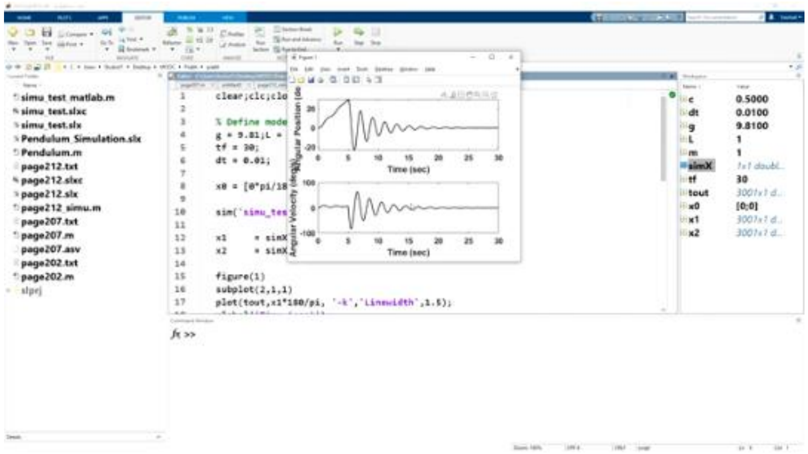
The angle should further increase towards 360 degrees and then go back and go forth. Now let us see if the model that we have done is able to predict it rightly or not. I am running this code with initial condition as 200 degrees all right and run all right So this is the first model that we have considered MATLAB function that is non-linear model. So it is currently it is rightly predicting that indeed it is going beyond 200 degrees and further later on it is settling to around 371 that is 11.2 degrees from there. Alright and the velocity is again going towards 0. Let us see now. Now this is the block diagram. All right, let us first check this. Now this is the block diagram theta. So here we have it is rightly predicting and settling towards 11.8 degrees and this was the previous response. And the other approach was a linear model $\dot{X} = AX + BU$ and if you see here It is saying this graph is saying that somehow it is going back to its 0 degrees and again oscillating back and settling down to 0. So this happened because we removed the nonlinear term sine of angle. So this is a linear equation.

This is a linear equation. linear system so as I mentioned in the note earlier that for small angles linear approximation is correct ok so linear approximation is only valid upon small angles now this 200 degrees is a obviously a huge angle it is the simulation is not able to rightly predict that it is indeed going further then finally state space approach Here also we observe the same thing. So here again we have a linear model approach which is not able to predict the correct simulation. Then finally transfer function. So what do you think the response will be here? It is still predicting the same thing. The reason, there is no methodology to enter the initial condition over here. So that is one of the drawbacks of using a transfer function. It always assumes the initial condition is 0. Now, finally, before I end this discussion on the pendulum equation, let us predict one more situation where I have considered a different input. So, when T is less than 5 seconds, then torque is a ramp input. You can consider a slope of 1, and T is greater than 5 seconds. And T is 0. So, what I mean by this is the torque is actually increasing linearly with time, and at 5 seconds, it is 0. So here, the slope is 1. I will consider here the same at 5 seconds; it is 5 Newton meters. Is 0. So, this is nothing but the input instead of the step input that we did earlier. Now, how to do this in Simulink? We use this switch block and write here the threshold limit of 5 seconds. Torque is zero after five seconds, and it is a ramp input for the first five seconds. I'll write a slope here, start time zero. All right, okay.

Now, instead of this step input, we will try with this switch block of ramp input. So, this 5 indicates time. We have time. All right. And let us see whether we are getting the similar input that we are expecting or not. We'll put a scope here. All right. I am saving the model, and let us put the initial condition back to zero just to check how it is working. All right, so you can see here. Before checking this, let us see the input. Yes, so now the input is actually increasing linearly with a slope of one up to five seconds, and then it is zero. Now, see the response of this. We again observe here that it is going, since the torque is going to zero, the angular position is coming back to its zero-degree position.

We can see here the linear variation of this torque up to 5 seconds. Let us see now the various models that we have modeled before.

(Refer Slide Time: 28:55)



This we have already checked. This should be a similar response. Yes, the same similar response, and of course we will get a similar response because the initial condition is zero and the angle is very small. So you can test with very different initial conditions. By entering different control inputs and testing the model. So by now, I think you must have gained some understanding of the Simulink model and how to use it for simulating different dynamic systems. Next week, we'll begin simulating the aircraft's six-DOF equations of motion, both in MATLAB and Simulink. Thank you.