Advanced Aircraft Control Systems With MATLAB / Simulink

Prof. Dipak K. Giri

Department of Aerospace Engineering

Indian Institute of Technology Kanpur

Lecture 16

Example of Separation Principle

This lecture will involve solving the problem and also performing the MATLAB simulation. Here, we're going to take the same example from Lecture number 14, the inverted pendulum system. So, if you remember, we have the dynamics in the dynamic state space, which was in state space form.

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 10.78 & 0 & 0 & 0 \\ -0.95 & 0 & 0 & 0 \end{bmatrix}$$

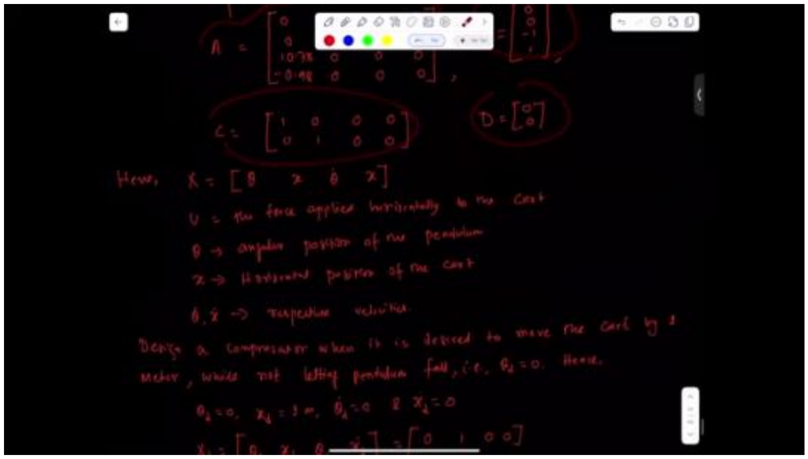
$$B = \begin{bmatrix} 0 \\ 0 \\ -1 \\ 1 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

$$D = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

The states to be controlled are $\theta \quad x \quad \dot{\theta} \quad \dot{x}$. Here, F is the force applied horizontally to the cart, θ is the angular position of the pendulum, x is the horizontal position of the cart, and $\dot{\theta}$ and $\dot{x}$ are the respective velocities of θ and x. The question is to design a compensator. The goal is to move the cart by 1 meter while not letting the pendulum fall. So, this means θ desired is 0.

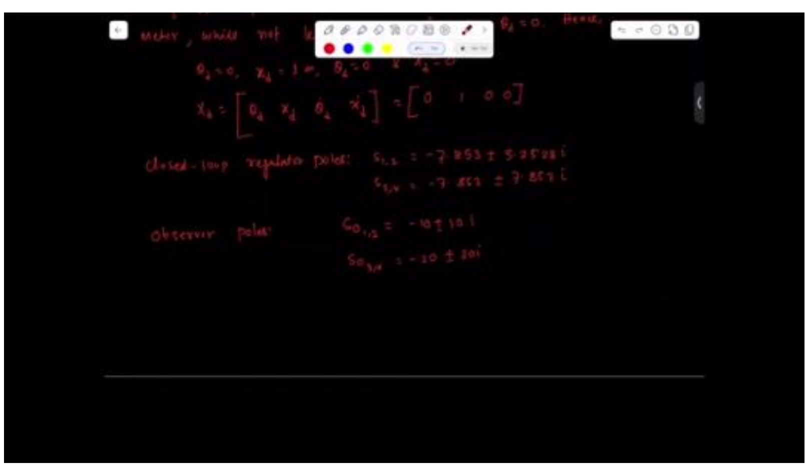
(Refer Slide Time: 00:41)



So, in this objective, we can write $\theta_d = 0$, $x_d = 1$ meter, $\dot{\theta}_d = 0$ 0, and $\dot{x}_d = 0$. So, the desired state variables can be assumed as follows. These are the desired state variables. So, this is basically a tracking problem, we can say, because it is not zero. The closed-loop regulated poles are assumed to be these, which are the regulated poles for state feedback design, control design, and these are the observer poles to be maintained.

Since the observer poles should be the first, and then the regulated poles. So, we will go step by step to find the compensator for this particular problem. So, here we are going to use the separation principle. We will separately design the observer and regulator and then combine them together. The same procedure that you have done in the earlier lecture, the separation principle, will be followed. Some steps will be the same as those in Lecture 14 for finding the control gain parameters, the same procedure. What we have followed in Lecture 14, okay? So, here first we will find the regulator control gain matrix. First, we find the regulator.

(Refer Slide Time: 02:23)



We'll have the MATLAB code also for this particular example. Regulator control gain matrix. So here we have given, we can write

$$V = [-7.853 + 3.2528i \quad -7.853 - 3.2528i \quad -7.853 + 7.853i \quad -7.853 - 7.853i]$$

$$K = acker(A, B, V)$$

$$K = [-1362 \quad -909 \quad -344 \quad -313]$$

K is the control gain matrix for the regulator if you notice here the observer poles are same what we have considered in lecture 14 right so here also we can form the L matrix

$$L = [-25149 \quad 60 \quad -183183 \quad 1810]$$

this is you can follow in lecture number 14 this is the regulator regulator gain matrix gain matrix and this is observer gain matrix. This is after vector I can say, gain vector and observed gain vector. So, now next part is since x_d is constant, If you notice that the desired state model is constant here, here you can see it is equal to 0 1 0 0. So, we can find $\dot{x}_d = 0$. So, this is also you can write it udxt, hence you can write ad equal to 0, $A_d = 0$. Algorithm for the regulator problem so now we have to find the feedforward gain matrix K_d so finding feedforward gain vector so for that as if you remember we had the last equation

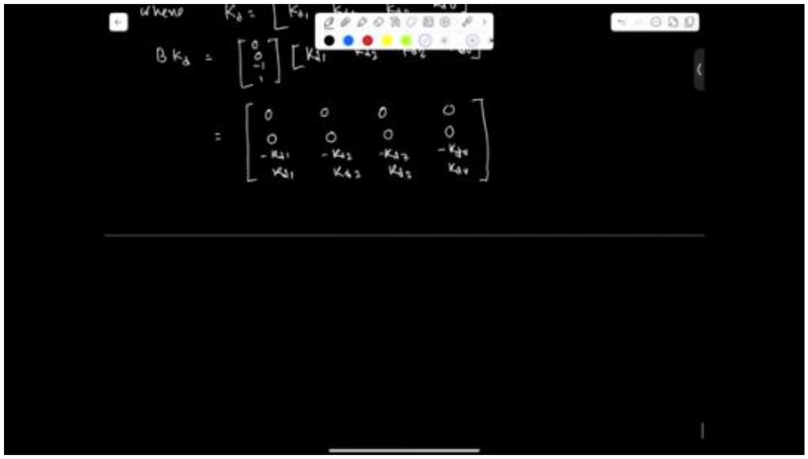
$$(A_d - A + BK_d)X_d = 0$$

That is the condition we had, right? For this $\dot{E}_0 = 0$. So, here we can write, so this expression already we had in the last lecture. So, this is the current condition for the compensator. So, here we can write

$$K_d = [K_{d1} \quad K_{d2} \quad K_{d3} \quad K_{d4}]$$

$$BK_d = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -K_{d1} & -K_{d2} & -K_{d3} & -K_{d4} \\ K_{d1} & K_{d2} & K_{d3} & K_{d4} \end{bmatrix}$$

(Refer Slide Time: 10:47)

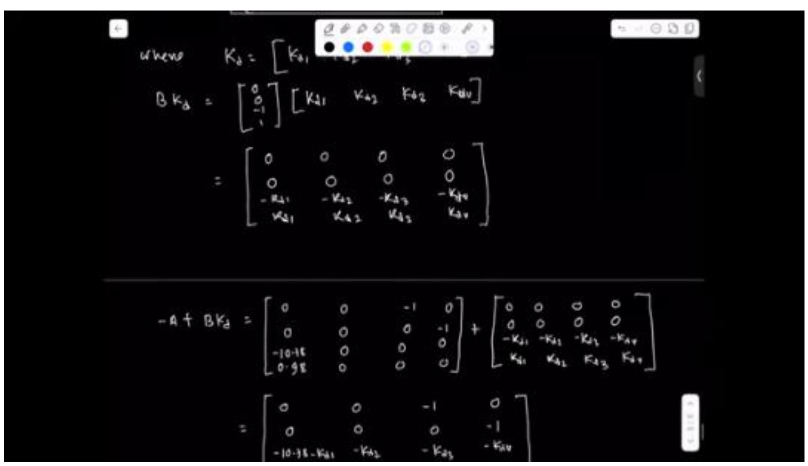


So now we will find Will find

$$-A + BK_d = \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \\ -10.78 - K_{d1} & -K_{d2} & -K_{d3} & -K_{d4} \\ 0.98 + K_{d1} & K_{d2} & K_{d3} & K_{d4} \end{bmatrix}$$

$$(-A + BK_d)X_d = \begin{bmatrix} 0 \\ 0 \\ -K_{d2} \\ K_{d2} \end{bmatrix} = 0 \dots Eq(1)$$

(Refer Slide Time: 13:36)

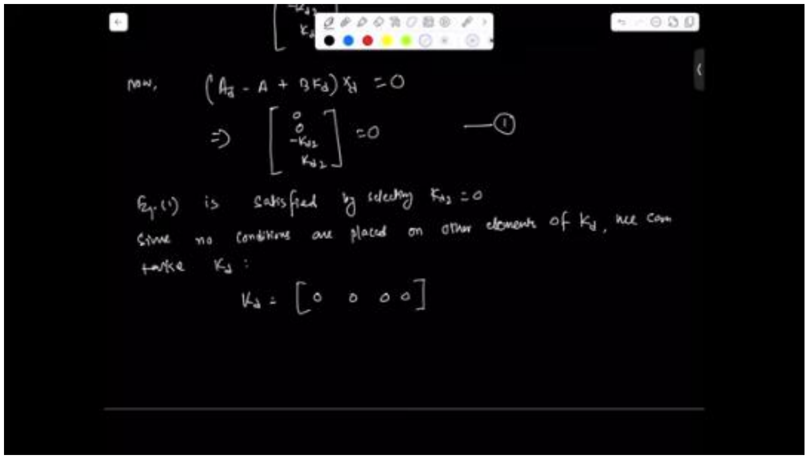


Equation 1 is satisfied by selecting value $K_{d2} = 0$. So in this case, since no conditions are placed on other elements of the K_d vector, we can take

$$K_d = [0 \quad 0 \quad 0 \quad 0]$$

Now, we will use the closed-loop state equation, which is of rank 2n, because we have the observer dynamics. observer to the state equation.

(Refer Slide Time: 16:42)



So this is very, very important. We have done this part in the last lecture. So here I'm not going to detail it because we have already found this expression in the last lecture.

$$\begin{bmatrix} \dot{X} \\ \dot{X}_0 \end{bmatrix} = \begin{bmatrix} A & -BK \\ LC & A - LC - BK \end{bmatrix} \begin{bmatrix} X \\ X_0 \end{bmatrix} + \begin{bmatrix} B(K - K_d) \\ B(K - K_d) \end{bmatrix} X_d$$

So here we can find A_{CL}, basically in MATLAB if you write in MATLAB. we can write

$$A_{CL} = [A \quad -B*K; \quad L*C \quad A - L*C - B*K];$$

$$B_{CL} = [B*(K - K_d); \quad B*(K - K_d)];$$

So if you notice here, all the terms we have found, right, all the terms are given to us in this particular section here. The only unknown we had was K and K_d, and we have found it here, and K we have found. Before this, these are here also. We have found observer gains, so kl and kd we have found. So these are the gain matrices or vectors here in this map. So now, if you find, we need to find the eigenvalues of A_{CL}, okay?

We need to find the eigenvalues of A_{CL}. This thing, if these eigenvalues are negative, then we can say this is going to zero over time. Let's now do the things. Let's check the eigenvalues of a eigenvalues. Of A_{CL}, if you go in MATLAB, you go right into MATLAB, and if you input, we'll have

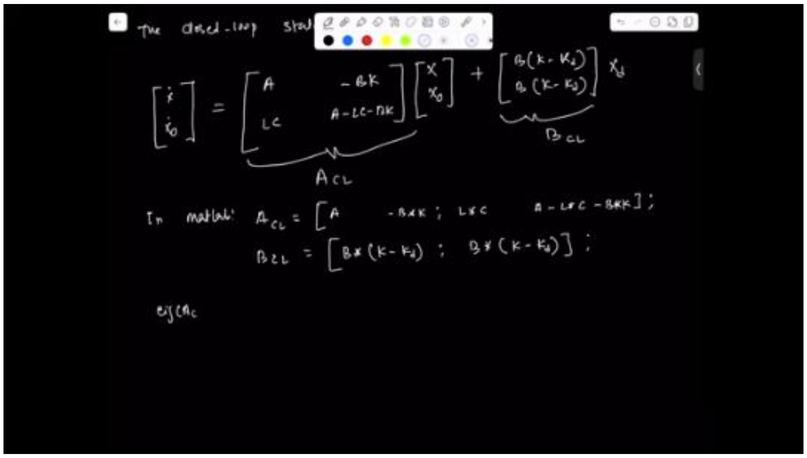
$$-20 \pm 20i \quad -10 \pm 10i \quad -7.853 \pm 7.853i \quad -7.853 \pm 3.252i$$

$$sysCL = ss(A_{CL}, B_{CL}, [c \quad zeros(1,4)], D)$$

$$[y, t, x] = lsim(sysCL, X_d, t);$$

So it means the dimension of a matrix we can say is 8 cross 8 actually. So that's why we are having 8 eigenvalues. Now we are ready to simulate the program. We have found almost all the terms here.

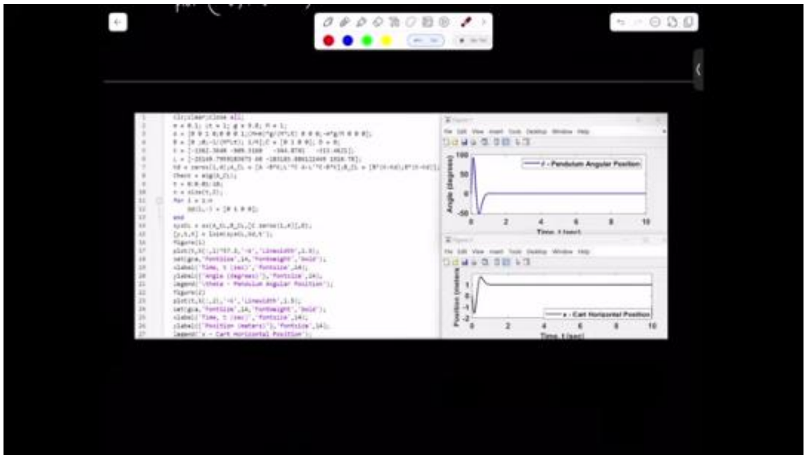
(Refer Slide Time: 20:02)



So if you simulate, we can find one thing we have to say here. Note that x vector contains the plant they are actually this expected. We are talking about this expected dynamics as well as observer dynamics. Here we can plot, we can plot and see the results. Since if you come go back to our problem, we want to see, basically, we want to see θ and x because this is the condition we are given. The compensator, when it is desired to move the cart by one meter while not letting the pendulum fall, so we are looking x, x, and θ. So, x and θ were the main results we needed. We needed to see which could track the desired value x to get and θ desired, right? So, results for θ and x mean We have x. We have a model like this: θ, x, $\dot{\theta}$, $\dot{x}$, θ, x, $\dot{\theta}$, $\dot{x}$. So, you want to see the region of θ and x in the state vector, right? So, in that case, if you want to plot in MATLAB, we can use this command: plot(t, x1 * 57.3). Why 57.3? Because we want to look in the s vector. The first term that is set is one. We are multiplying by 57.3 because we want to see the result in degrees.

So, one radian equals 57.3 degrees, and if you want to plot, this is for the cart position, and this is for the angle. Now, let us look at the result, the simulation result of this. So, this is the MATLAB code Dr. Prabhjeet has prepared. He is a very good coder. So, this is basically the whole of whatever things you have done so far. This is the result. So, if you notice, the pendulum is already falling, right? Because if you see, it is basically and 90 degrees falling. If you say 90 degrees higher, so it is 90 degrees goes and again falls. So, the design compensator is unsatisfactory for this particular problem. But the cart position has been satisfied. If you look at the cart position, it goes to one meter, but the problem is with the pendulum position by the angle.

(Refer Slide Time: 26:22)



So, this is a problem in this design of what you have done so far. So, for that, you can play on the pole. So, we have two poles that you have chosen for the regulator design. We have to play something there. So, this is the regulator part. So, we have to do something here.

We need to choose these poles with some higher values so that we can modify the result, or we can choose some new regulator poles, maybe very close to the imaginary axis. Also, we can do that. Let's look. Let's look at how we can change the poles and how we can get some bigger results for the pendulum and pendulum position. So now, what we are doing is we are trying to move by moving the regulator poles closer to the imaginary axis, we can reduce can reduce overshoot. Yeah, that is very important. So we need to, because here, if you notice, actually, we have overshoot and undershoot. So if you want to reduce this overshoot, we need to place the desired poles closer to the imaginary axis, which can increase the settling time. But if you, yeah, if you increase, if you place these poles towards the imaginary axis another problem will come. Settling time will increase. Let me write here: overshoot, but at the cost of increased settling time. So it means we can't get both the advantages. We have to

pay some cost for the other case. So if you want to reduce the overshoot, if you want to maintain this angle bias, so you need to, maybe you can place the poles towards the inverted axis. Maybe if you, for example, initially the poles are here, and you are trying to place them a little bit closer to the imaginary axis, so in that case, we can reduce the overshoot, but settling time will increase. So it means, basically, we are putting more damping into the system to actually dampen. Actually, what it helps is we reduce this overshoot and undershoot, but settling time will increase, right? That is the thing we have discussed in classical control. So now we can choose the desired poles for the regulator as

$$S = -0.7853 \pm 3.2532i$$

$$S = -0.7853 \pm 0.7853i$$

So basically, we are, if you notice here, we are just increasing and decreasing by 10 times the desired force. So now it is all in point, but if initially, some number point, right? Yeah, and if you again do the command, k equal to, if you put like so, you can find the v matrix. Sorry, you can use v equal to. We can place all the four desired poles. This four, this plus minus, you can put four poles in the v matrix, and if you use

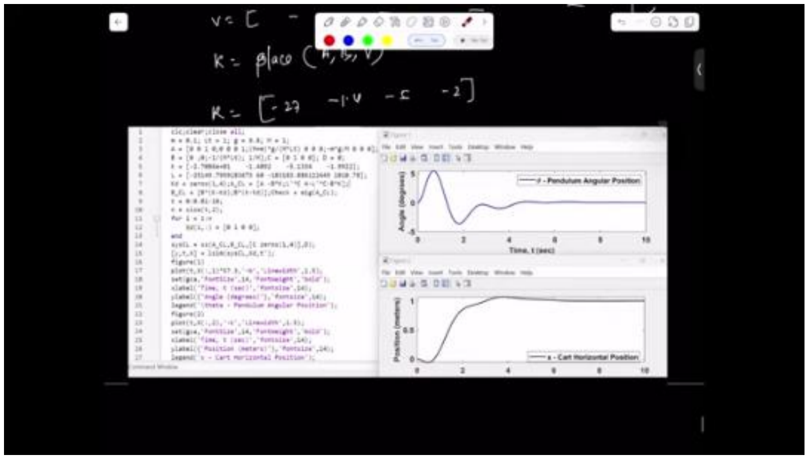
$$K = place(A, B, V)$$

$$K = [-27 \quad -1.4 \quad -5 \quad -2]$$

And if you use those K for this code or for the analysis, we have the MATLAB code for that as well. So, let us look. Yeah, we can see that now the angle is very less. So, very less, that is overshoot and undershoot is reduced.

Initially, it was close to 90 degrees, but now it is close to 5 degrees. And also, it is nicely reduced to the cart position, close to a value of 1. And here, actually, we modify the K matrix, the poles, the regulator poles, and this is the MATLAB code; you can go through it. And this is how we can modify the system response. And you can achieve our desired result.

(Refer Slide Time: 32:24)



So this is what we have; this is the compensator we have designed for the pendulum system. Now we are going to design the same concept for the aircraft system, and we will see how the different states in the aircraft can be controlled using a compensator. So, let us stop here; we will continue in the next lecture. Thank you.