Advanced Aircraft Control Systems With MATLAB / Simulink

Prof. Dipak K. Giri

Department of Aerospace Engineering

Indian Institute of Technology Kanpur

Lecture 10

Tracking Controller Design

In today's lecture, we will be discussing how we can track a time-varying function instead of a reference signal which is zero. So, in our previous example, we designed the regulator problem. In today's lecture, we will be designing a tracking controller where we will track a time-varying function as a reference signal. We will go with the first mathematical derivations of the controller. Then we will take the same example that we had in the last four lectures, but we will track a time-varying function here—basically, the desired dynamics to be tracked. So, let's start the lecture on main control design for tracking a time-varying function and also we have MATLAB simulation for validation of the concept. So, let's start the lecture here. We are assuming the desired dynamics to be tracked, i.e.,

$$\dot{X}_d = A_d X_d$$

So, here we are assuming that all the eigenvalues of A_d matrix should be negative, so it is a very stable function.

So here we are going to track the xd dynamics, okay? And the natural dynamics in state-space LTI form. We are going to consider: we already had before the system; basically,

$$\dot{X} = AX + BU + FX_n$$

So, here, if you look at the closed-loop control block diagram, this is the summing point, and here we have the controller, a control block that actually tries to find the control gain. The controller and its output go to the plant to be controlled. And this is the output, the plant X. And we are going to create a time frame function xp, instead of using 0, if I say so now. And this is feedback. So, now we can find the error dynamics.

So, here, if you write $E = X_d - X$ right. So, now you can write error dynamics. So, you have a plant, you can see the aircraft system, and we are also having noise xn in the system as well. So, error dynamics you can write. Let us write

$$E = X_d - X$$

$$\dot{E} = \dot{X}_d - \dot{X}$$

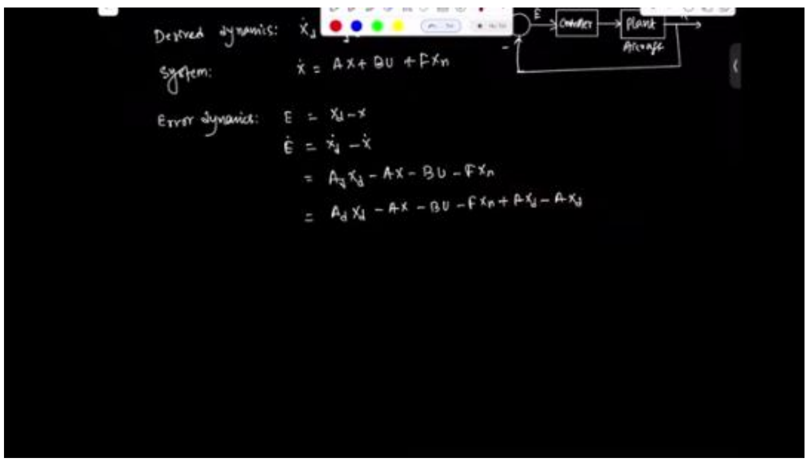
$$= AE + (A_d - A)X_d - BU - FX_n$$

So, here but for the part in the previous lecture, in the previous lecture, we had since $X_d = 0$ in the previous lecture, so u can be formed as

$$U = -KX$$

$$U = KE - K_d X_d - K_n X_n$$

(Refer Slide Time: 04:48)



This is how the control structure should be. For this particular case, I thought this in this course for taking contracts; yes, so we can write this form: so now we can come back here. So, here we can write

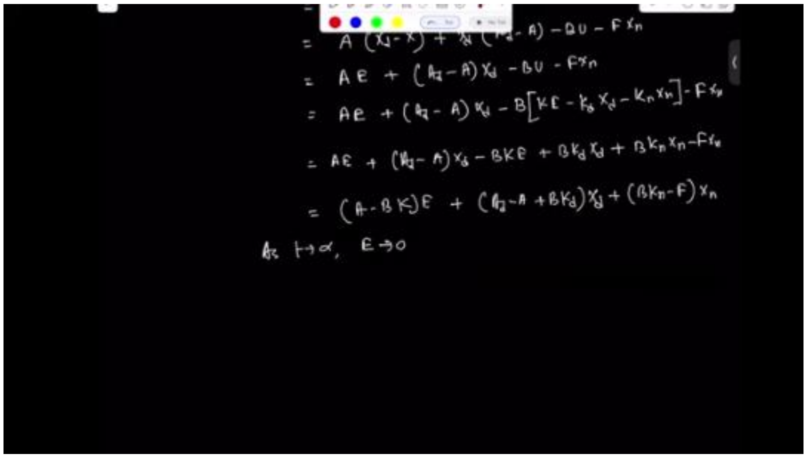
$$\dot{E} = (A - BK)E + (A_d - A + BK_d)X_d + (BK_n - F)X_n$$

So, if you notice here, we have a very interesting equation. So, in all these terms, we have some form of matrices. This is one matrix; this is one matrix; this is one matrix. If all the eigenvalues of these matrices are negative, then we can say that as t tends to infinity, E goes to zero. Right? If all the eigenvalues of the individual matrices are negative, so now let's see how we can do it. Let us find some condition on kd and kn so that, anyways, we have to design the controller for this. But the critical parts are these two parts, right? So, let's start. Let me write some stuff here: At steady state, the error either reduces to. 0 or meaning here the small value; right. So, in this case, "or" is more than means it can go to some constant value. Your steady-state means some constant value, right? So, in that case, the derivatives of e dot will be zero. We can write: at steady state,

$$0 = (A - BK)E_{ss} + (A_d - A + BK_d)X_{d,ss} + (BK_n - F)X_{n,ss}$$

$$E_{ss} = (A - BK)^{-1}\left[(A - BK_d - A_d)X_{d,ss} + (F - BK_n)X_{n,ss}\right]$$

(Refer Slide Time: 09:26)



You can write that so you see, in state feedback control, what you have learned. We can design this part right; this part we can, if you can manipulate K matrix, make the eigenvalues of the overall matrix to negative. So, in that case, it is favorable now. The only other condition this can be made zero in this part, this part, and this part are going to zero. That is very important because this is not equal to zero; otherwise, a - b cannot go into zero, otherwise, this will not satisfy the inverse property, and it will not be possible. So, even if it is practically correct in the previous equations that $A - BK$ is invertible. In this case, we can write: Since this is just A_{CL}, right? so we can write here that

$$A_{CL} = A - BK \neq 0$$

since eigenvalues are in the left half-plane. Now, for E_{ss}. to be zero, the only conditions are

$$A - BK_d - A_d = 0$$

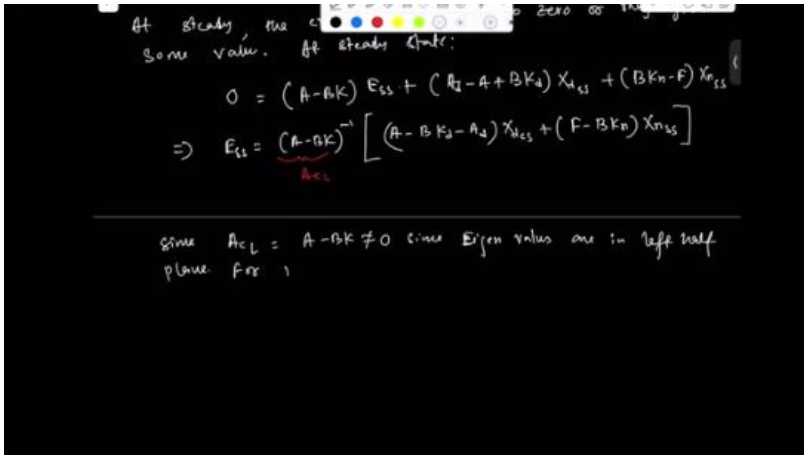
$$F - BK_n = 0$$

So, from these conditions, we are needed to find. K_d and K_n in such a way that E_{ss} goes to zero, we can write

$$K_d = B^{-1}(A - A_d)$$

$$K_n = B^{-1}F$$

(Refer Slide Time: 13:44)



So, these are the mathematical foundations to design the controller if you need to track a time-varying function. If you need to track a time-varying function. This is the procedure you need to follow. Now let's move to the example part. Example.

So here we are going to consider the same example that we have done. So, I will take this. This is the example we had in the last lecture. The same example we are going to consider here. But here, we can modify it a bit.

So this part we can modify it right up. We can design a controller that makes aircraft. Track a target or desired dynamics, causing you to write here, so what are we had skipped. We had one state, which was normal acceleration, pitch rate, and elevator deflection. Right? So, these can follow the desired dynamics here. So, we can write a desired and q desired and elevator desired. Okay, this is elevator deflection. This is elevator deflection. Okay, so we can form the desired to the .-okay

$$\begin{bmatrix} \dot{a}_d \\ \dot{q}_d \\ \dot{e}_{dd} \end{bmatrix} = \begin{bmatrix} -10.1 & 35 & 150 \\ 0.1 & -1.1 & -21 \\ 0 & 0 & -8 \end{bmatrix} \begin{bmatrix} a_d \\ q_d \\ e_{dd} \end{bmatrix}$$

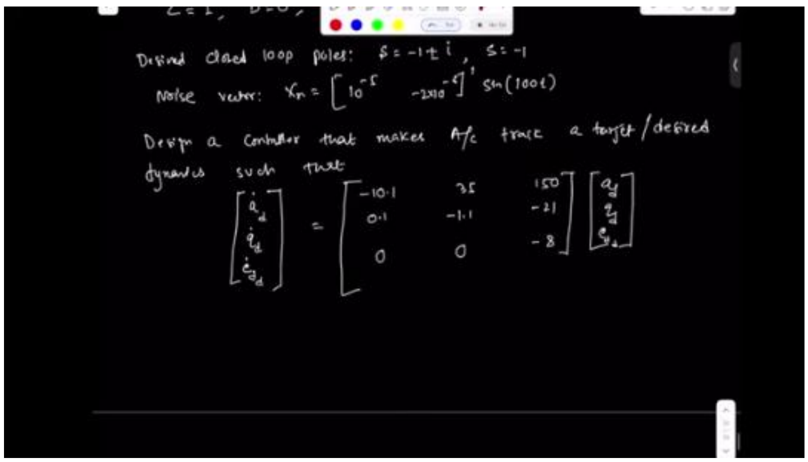
So, the desired dynamics, and one thing to notice: please remember that the dimension of the system matrix of the desired dynamics should be the same as that of the original system's dynamics; this is very important. Otherwise, we cannot take the error into account if we are taking and considering it based on that assumption, right. So now we will solve this problem. Let us first analyze the AD matrix and find the natural frequency, damping ratio, and poles in MATLAB. Okay, so for that, we can do it as follows: Here, this is the AD matrix; this is our AD matrix, right. So, if we use in MATLAB Damping, Ad, if we enter it, the output will be

$$\omega_n = 10.47 \quad 0.72 \quad 8$$

$$\xi = 1 \quad 1 \quad 1$$

$$P = -10.4 \quad -0.72 \quad -8$$

(Refer Slide Time: 18:14)



so this is the output we are getting. But if you notice here carefully, all desired poles are on the left-hand side, so we can say this system is stable, right? Now, our next part is that we will design the controller for the first part. So, now we are going to start working on this part first. This part okay? So, here we are going to consider the desired code to be the same as what is given in the example here. The same desired pole we are going to consider is this desired pole. So, the desired pole is given by

$$S = -1, \quad S = -1 \pm i$$

Now, if you remember, we can use the place command to get the control gain for this particular desired poles we had in the previous example. Please go back to it.

Go to the lecture; last lecture. We had

$$K = [0.0006 \quad -0.0244 \quad -0.8519]$$

$$A_{CL} = \begin{bmatrix} -1.526 & 43.36 & 28.28 \\ 0.22 & -1.4 & -32 \\ -0.008 & 0.34 & -0.07 \end{bmatrix}$$

$$K_n = \begin{bmatrix} \frac{-0.02}{272} & \frac{-0.1}{272} \end{bmatrix}$$

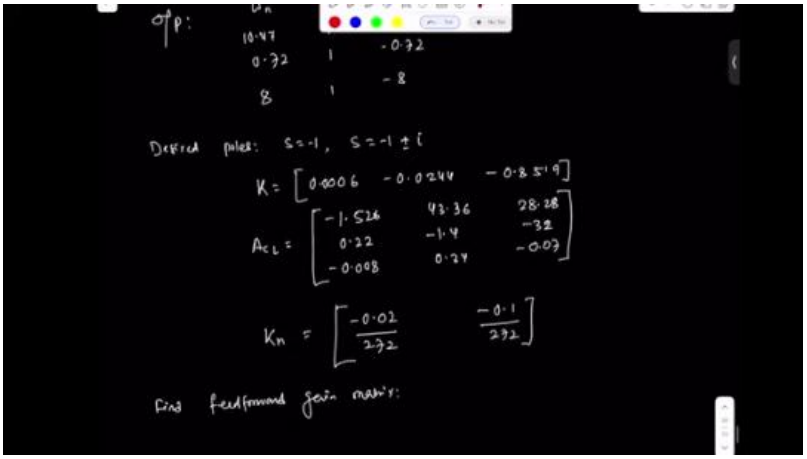
So, these are the values we had obtained in the last lecture now. here's the challenging part: the last section starts right here. This is the one we have to work on now. So let's start by figuring out how we can work on that. Here, we need to find the feedforward gain matrix. Why feed forward? Because if you notice, the controller was designed here; in the controller, one of the parameters is $K_d X_d$. So, this part can be called the feedforward control part because the controller depends on the desired dynamics X_d. So, here we can let us assume

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

k and we can find the matrix A minus may be you find difficulty but we have MATLAB problem for holes for this problem we have MATLAB solution for this problem there you get the better picture

$$A - BK_d - A_d = \begin{bmatrix} 8.4 + 272K_{d1} & 15 + 272K_{d2} & 110 + 272K_{d3} \\ 0.12 & -0.3 & -11 \\ -14K_{d1} & -14K_{d2} & -4 - 14K_{d3} \end{bmatrix}$$

(Refer Slide Time: 23:45)



Okay, so if you notice here, K_d only affect the first and third row, right, in this matrix. Moreover, we can choose in this matrix.

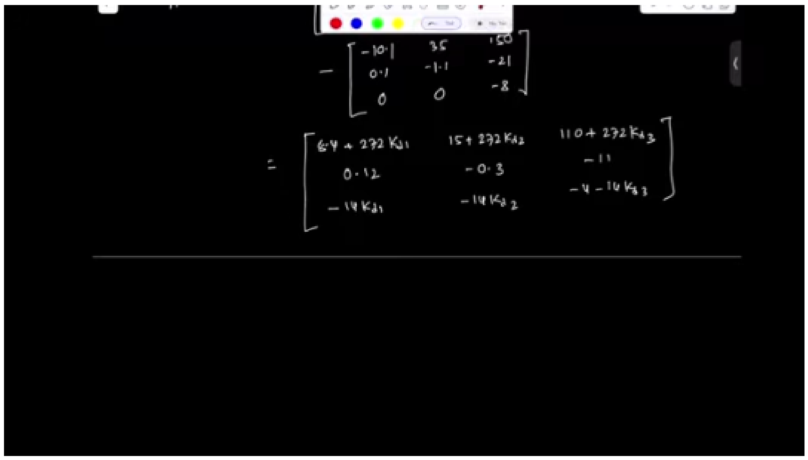
$$K_{d1} = -8.4/272$$

$$K_{d2} = -15/272$$

$$K_{d3} = -110/272$$

Right, so finally, if you substitute these values into these rows, so anyway, this row will go to zero. But this row will not have 0 terms in the last row. So, let us find it.

(Refer Slide Time: 26:18)



$$A - BK_d - A_d = \begin{bmatrix} 0 & 0 & 0 \\ 0.12 & -0.3 & -11 \\ 0.43 & 0.77 & 1.7 \end{bmatrix}$$

$$K_d = [-8.4/272 \quad -15/272 \quad -110/272]$$

Now, we'll also assume the same noise level acting in the system. So, we are also going to consider the same noise we had in the last lecture.

$$X_n = [1e^{-5} \quad -2e^{-6}]\sin(100t)$$

and also we are assuming that the initial state is

$$X_{d0} = [3 \quad 0 \quad 0]'$$

$$B_n = \begin{bmatrix} 0 & 0 \\ -0.0035 & 0.004 \\ 0.001 & 0.0051 \end{bmatrix}$$

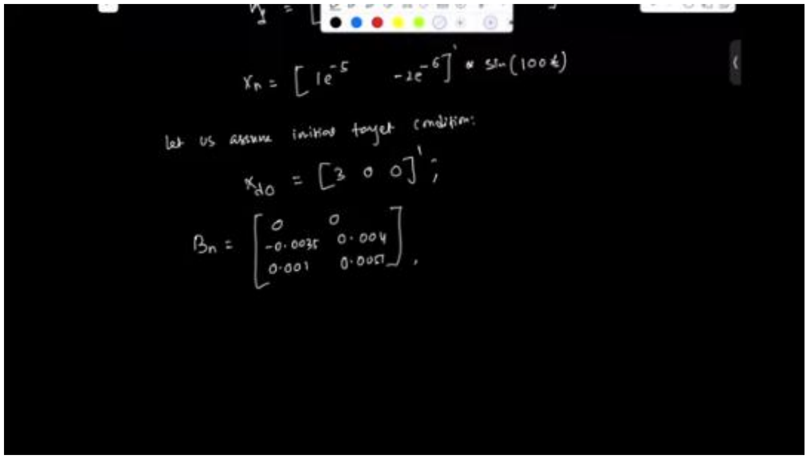
$$C = eye(3)$$

$$D = 0$$

So, these are the values given to us. Now, we will go to the MATLAB part and find so I think we have found all the parameters you require for simulating the dynamics of e dot:

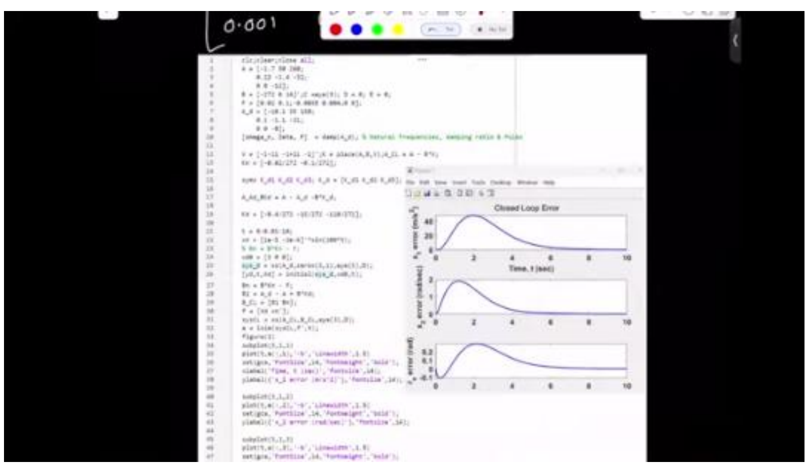
so we have all the values. In this equation, right. We found this, we found this, and we also found this. We are given this.

(Refer Slide Time: 31:10)



So, now we are also given the x and d also. So, we can simply use the MATLAB command to find the error values and their dynamics. So, let me take the MATLAB result. So, this is the MATLAB result for this particular problem. So I hope it is visible. So, here, if you notice, we are having the system matrix. So we have D matrix; we have E, D, E. A, A, D is also given to us, and also we have the uh if you can then we can find p and also i mean zeta p tan p ratio and pole of the

(Refer Slide Time: 31:40)



desired dynamics, and this is the desired eigenvalues, which will help us to find the k control gains. From this, we can find the augmented matrix, and also we have the Kn, which is going to control the noise in the system. Here, we are defining Kd. Which will be used to design as a feedforward gain matrix, and this is the AD and this is the matrix we just found. This is the A matrix. This one. This matrix. So, it is here defined. And we have the kd parameter already, and we are simulating the problem for 10 seconds with a sampling time of 0.01. This is the noise level in the system, and this is the initial value of the desired dynamics. Now we can find this as a desired system dynamic using these parameters, ad and z i and d. We can find the desired dynamics, and so these are the dynamics it is given to us. So, no problem here:

$$\dot{X}_d = A_d X_d$$

and here eye is basically and d part is already assumed to be 0, right? So from this, we can find this solution. Okay, this is the desired dynamic solution. now, we will come to the error dynamics so these are the parameter going to use in the error dynamics and this is the error dynamics we are having this sys cl closed loop error dynamics and now we are finding error so here e actually here is a vector so e is actually e_1 e_2 e_3 and this is how other things are just to modify the plot. So, if you notice in the plot, this is basically the x_1 acceleration, and this is the normal acceleration of the first plot; the second one is the pitch rate, and the third one is the elevator position, right? So, this is how we can get the Plot of this particular example: If you notice from the response, as time proceeds, we are getting the responses going to zero after 10 seconds; it is converged to zero. So, we can say that the effect of noise is not that much significant, and also we can track the desired dynamics.

Because all this plot actually is if you write here

$$x_1 = a_d - a$$

$$x_2 = q_d - q$$

$$x_3 = e_{dd} - e_d$$

So, as time proceed our error is going to zero. So, it means that the system is going to converge to the desired dynamics. So, this is how we can design the controller to track the desired dynamics, and we have done so for the longitudinal motion of the aircraft. This is a very important example.

And if you have a similar kind of system to be designed, and if you want to see how the noise is going to behave or act in the system, you can design the controller and mitigate the noise. I hope this is very important for designing an autopilot for response systems. Okay, thank you very much. We'll continue from the next lecture. Thank you.