Advanced Aircraft Control Systems With MATLAB / Simulink

Prof. Dipak K. Giri

Department of Aerospace Engineering

Indian Institute of Technology Kanpur

Lecture 09

Controller Design in the presence of noise

Hello, everyone. In today's lecture, we will be considering the noise in the plant and how we can come up with an effective controller which will help us mitigate the noise in the system. So, here we will be using the state-feedback controller, what we have done in the earlier lecture, and also we have the simulation results in MATLAB to validate our numerical analysis. So, let's start the lecture. So, today's lecture is on pole placement regulator design for plants with noise. These are the topics we are going to cover: controller design for the fighter aircraft's longitudinal motion, how we can design the controller for this particular case, longitudinal motion with noise, and also we'll have the MATLAB simulation to validate our numerical analysis.

So here I'm gonna first define some important terms: the measurement noise. The noise are two types: one is measurement noise. This is due to the imperfections in the sensors that measure the output variable. Due to imperfections in the sensors that measure Output variables. So, this is basically in the feedback loop; these values, because sensors are generally kept in the feedback loop.

So, here, if we have the block diagram, we have the summing point here, and this is the reference signal to be tracked. This is the controller. The plant to be controlled has a plant output feedback, and we have the sensor system. states actual values of the plant, which are according to the sensor. So, you can imagine this is actually measurement noise. And there is another source of noise which is called process noise. This is due to ignored dynamics when modeling plants.

So, sometimes we ignore some terms in the system to make our lives simpler and make the control synthesis easier. Some complex terms, or perhaps values that are very small in magnitude, we don't consider when designing the controller, but practically, these noises also come into play. So, this is basically what acts in the plant. This is part of the process noise. So, these are the noises we generally consider while designing controllers. So, now considering these two important noise sources in the plant's dynamics.

So, if you consider them in our state-space model, So, the state space model, including one with noise, can be written as

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

$$Y = CX + DU + EX_n \ldots Eq(1)$$

So, as we have done before, we are taking the compact form of the most critical case in this example. So, here, if you notice, this is how we are taking the "DU" part in the output equation as well as how the noise terms are going to.

affect the overall dynamics of the system; we are considering everything. So, here is the state space model of the system equation, and let us define this as equation (1). So, now we are going to make some assumptions as you have done before. So, here we assume the regulator control. So, $X_d = 0$ and $K_d = 0$ also. Since we are assuming there is a reference signal to be tracked as 0, you can correspondingly assume it is 0. So, if you go back to our earlier lecture, this is how the controller can be designed for this particular system and how these parameters come into play in the controller; we have explained it. In that controller algorithm, we are assuming this part to be zero. Okay, now, if you base the control algorithm on this consideration, we are assuming the controller we are designing:

$$U = -KX - K_n X_n \ldots Eq(2)$$

So here we are taking a new parameter in the controller, which is minus kn xn, to help mitigate the noise in the plant. This part will control the states; so, you can assume the state with the control part, and this is the part going to handle the noise in the system, okay?

So, and now, if you substitute this control algorithm into the equation, the state equation. So, let us define this as Equation number 2, OK. If you substitute this equation (2) in our state equation, the x-dot dynamics can be written as

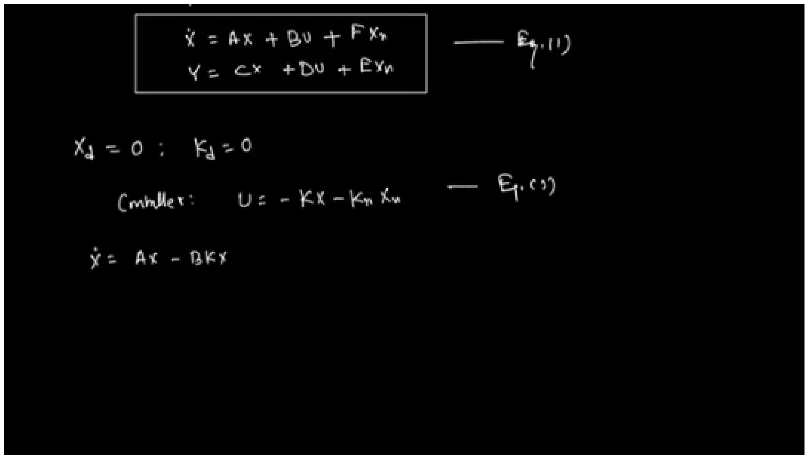
$$\dot{X} = A_{CL}X + B_n X_n$$

$$A_{CL} = A - BK$$

$$B_n = F - BK_n$$

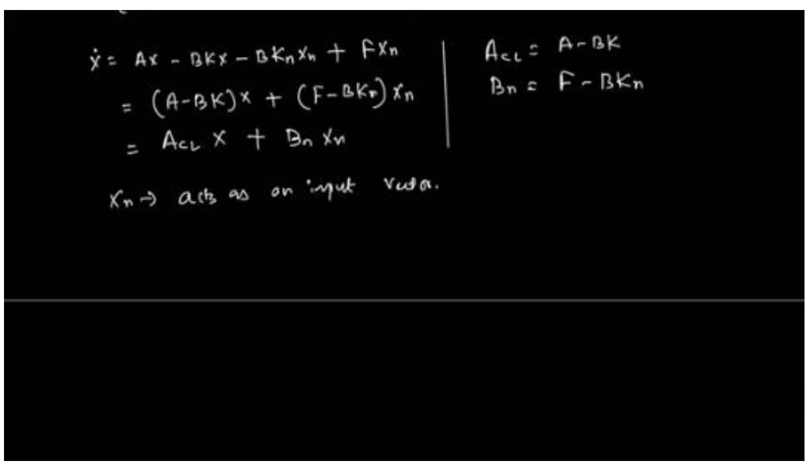
Okay, and you can also write it as In a more simplified form, so as we have done in the previous lecture, this part actually represents the augmented matrix. Alternatively, instead of writing "A - BK," we can write "A_{CL}." The closed-loop augmented matrix includes the controller part, which you will notice is also coming into play here,

(Refer Slide Time: 08:37)



So, this is the controller parameter going to handle the noise, and this is the controller parameter going to handle the regulator problem. So, here xn acts as the input vector. And practically, it is difficult to predict the values of xn, but we are having some approximation xn. For designing the noise coefficient matrix, so now, this is the overall idea how we can design the controller with noise, and how the controller can be designed. This is the controller for the particular system. The system: Now, we'll take an example to validate this problem.

(Refer Slide Time: 10:30)



This example is to define the problem. Consider a fighter aircraft whose longitudinal state Variables are normal acceleration (which is meters per second squared), pitch rate (which is radians per second), and elevator deflection. deflection, which is measured in radians, ok. Based on the deflection of the elevator, we can sum up the torque to the system.

So, while we are here assuming that input is desired. elevator deflection; this is our input to the plant. Now, the plant dynamics is in this form: so where

$$A = \begin{bmatrix} -1.7 & 50 & 260 \\ 0.22 & -1.4 & -32 \\ 0 & 0 & -12 \end{bmatrix}$$

$$B = \begin{bmatrix} -272 \\ 0 \\ 14 \end{bmatrix}$$

$$F = \begin{bmatrix} 0.02 & 0.1 \\ -0.0035 & 0.004 \\ 0 \quad 0 & \end{bmatrix}$$

This is the noise coefficient matrix and in the output equation c is given to I; I did the matrix, and d = 0, and u = 0. So, these are the values given to us. So, in the output equation, if you write it in state space form, we get

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

$$Y = CX$$

So, this is the equation over an equation for this particular example: d and u are 0. And these are pole locations; desired pole locations given to us; and these are closed. Poles given to us

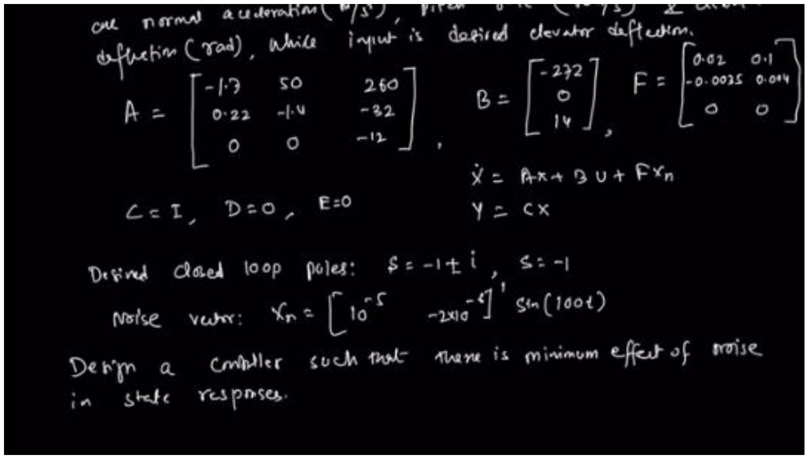
$$S = -1 \pm i, S = -1$$

So, if you notice, here is a 3-by-3 matrix. So, the characteristic equation is supposed to be of order 3, and if you derive it from the poles, you will also get a third-order characteristic equation, right. And also, the noise vector given to us,

$$X_n = [10^{-5} \quad -2 * 10^{-2}] sin100(t)$$

So, this is the noise level acting in the system. Therefore, we need to design a controller such that there is minimum effect of noise in the state responses. So, this is our objective. So, we have to design a controller such that the minimum effect of noise is on the responses. For this is our problem; now we will solve this problem, and then we will have the MATLAB code for this particular example solution. So, before you proceed to the controller design part, first, we have to check the system's properties to determine if it is controllable.

(Refer Slide Time: 16:09)



So, these are the fundamentals or basics we have to test for this particular system before you go to the control design part. So, first part is that we have to check the eigenvalues of the system. So, we can check the eigenvalues, or yeah. There is a command in MATLAB. So, I will try to use the MATLAB command for this particular example because the system is a quite big system.

So, if you go through all the derivations, it will be quite tedious. So, we are going to use the MATLAB code for this particular example. So, there is a command in MATLAB, and it is called `damp`. Then, if you use this command with some matrix A, it will provide natural frequencies, damping, and the damping ratio. The poles are also the roots of the characteristic equation, right here. For our particular example, if you write "A matrix," where your A matrix here is this matrix, and if you enter the command in MATLAB, you will have output for natural frequency, damping ratio, and poles. So we are getting four points. 1 minus 4.87, and natural frequency 1.77 minus 1 and 1.77; and the last state: 12 1 and -12. So, here, if you notice the pole locations in this case, there is one pole on the right-hand side of the S-plane. So, this is making the system unstable. So, this is an unstable pole. So, we need to design a controller because this plane is not stable.

So, we can say here, we can write S equal to 1.77 is an unstable pole. Now, we will go to the next part: system's instability check. Now, we have to check whether the system is controllable or not, and whether we can control the system or not. So, for that, we need to check the rank of the matrix system. So, we are having. here our b is given to us which is

$$B = [-272 \;\; 0 \;\; 14]'$$

$$C = eye(3)$$

$$r = rank(ctrb(A, B))$$

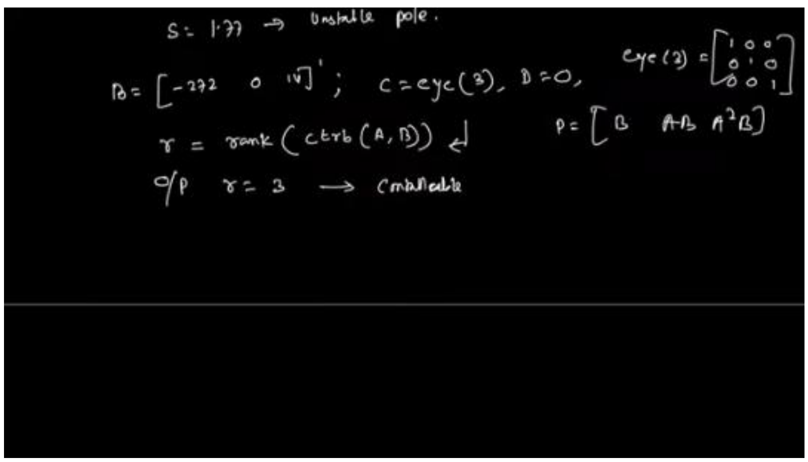
We are getting an output equal to 3, so the system is full rank and it is controllable. So, the control you can design will affect all the states in the system, right?

So, here, if you notice, this is basically CTRB AB, and it will give you the P matrix. If you remember, P matrix, you know,

$$P = [B \quad AB \quad A^2B]$$

So, automatically with this command, `CTRB` will generate this matrix. And we are finding the rank of this P matrix. Okay, that is 3. It is controllable. Now, we can use another command to find the control parameters. If you go back to our previous lecture, we had the two characteristic equations, and you can compare them. From the characteristic equation, we can find the control parameters, right? So here, you can use some commands in MATLAB to find the control matrix directly.

(Refer Slide Time: 22:08)



So, our V is given to us, desired poles at

$$V = [-1 - i, -1 + i, -1]$$

So, if you look at this pole location, these are stable because all poles are in the left-hand side of the S-plane. So, if you in the MATLAB command window, if you write

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

$$K = [0.006 \quad -0.0244 \quad -0.8519]$$

so this is the control parameter k_1, k_2, k_3 which is going to control the three states, right? Three states means these are the states: one is Our acceleration, normal acceleration, pitch rate, and elevator deflection are all right. This state can be controlled using these control gain parameters. So, now we have the control parameters and also the A and B matrices. Therefore, we can find the augmented matrix, right? So, now we will find the augmented matrix.

This is what we know:

$$A_{CL} = A - BK$$

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

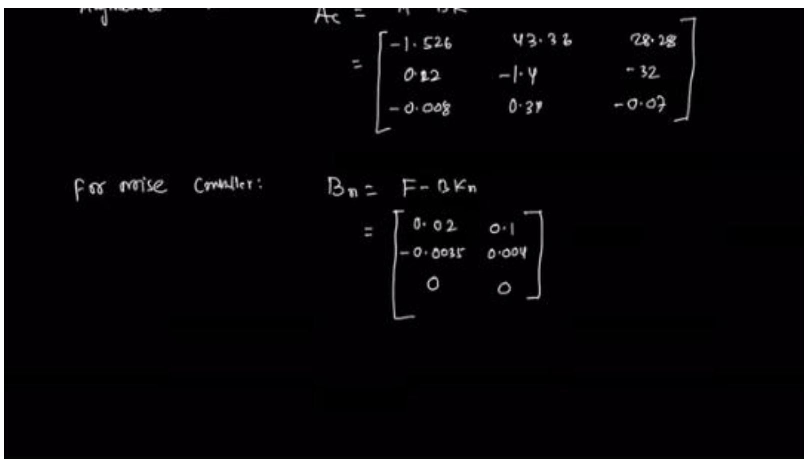
Right, this is the augmented system matrix, so we have done the first part of the controller. So, first part of the control. Now we're going to start the second part: how to control the noise. The second part: So, here, if you notice, we need to find

$$B_n = F - BK_n$$

$$= \begin{bmatrix} 0.02 + 272k_{n1} & 0.1 + 272k_{n2} \\ -0.0035 & 0.004 \\ -14k_{n1} & -14k_{n2} \end{bmatrix}$$

So, now, what we are going to do is find k_{n1} and k_{n2}.

(Refer Slide Time: 26:43)



So, what we are going to do is, we will make these parameters 0. We will make this parameter the largest parameter to 0. So, if we use these two parameters, we can find them. So, what we are going to do is make the first row zero; we can then find k n 1 we can find minus 0.02.

$$k_{n1} = \frac{-0.02}{272}$$

$$k_{n2} = \frac{-0.1}{272}$$

And considering these values: if you consider these values in this matrix, so we can find b n. So, B, and the first row will be 0. It is obvious because you already assumed it to be 0. So,

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

So, this is the B_n matrix you are getting now. If you notice carefully here, so okay, this means we are always going to have some effect noise in the closed-loop system because if you notice here, here we have some non-linear values; so, there are effects in the system.

But let's see. We'll get a better picture of how it is going to happen while we're doing the MATLAB code. So, now we have `bn` and `scl`, so we can find the overall state space model. So, let's work on this. Now we can use the MATLAB command `state space`. lti linear time-invariant systems can be designed by using the `ss` command in MATLAB. So, if you consider this, I will come back here again. So, if you notice, this system matrix is for a closed-loop system. This is basically a closed-loop system, right? Because everything here shows that k n is the controller, and based on this, we have an augmented matrix for the state and another augmented matrix for x n, right? So, now this whole equation is actually in closed loop form because the controller is already in the loop. So, now we can come up with the closed-loop state space model. So, here let us write in the MATLAB command as

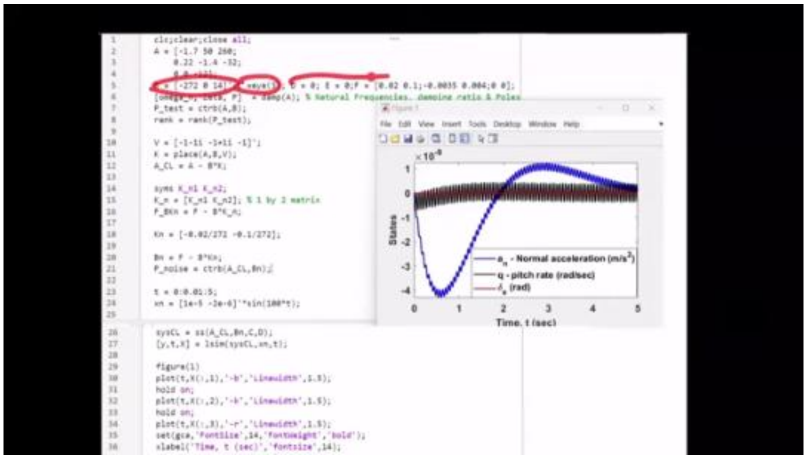
$$syscl = ss(A_{CL}, B, C, D)$$

In this particular case, our D-part is 0 and C is an identity matrix; and B and C have already been found here. So, if you use these matrices in the SS command, we can get the output. So, if you enter it in the MATLAB command, we will get an output of this quite big equation, which I am not writing here. outputs as continuous-time state space model of the closed-loop system. Okay, now, how we can see the response of the system—this is a very important part. So, we have the closed-loop system where we have the controller inside the system. Now we will see how we can get the response and whether it is stable or not because the controller is already designed. In MATLAB, if you use some commands—this command is called `lsim` in MATLAB.

So, this produces the state response from the given or measured data. So, how can we use this command? Let us work on this. So, here in the MATLAB command window, if you Okay, instead of "that," I can take the MATLAB code and result, which will be useful for you. So, what I can do is: this is the MATLAB result of this particular example. The full code is here; let's work on this. Yeah, so the full code for this particular example is given. Here, if you notice, the matrix is defined.

B matrix is defined; C matrix is defined. So, this is basically the problem that we have given in the system. So, then we have used the damp parameter, which will provide the natural frequency, damping ratio, and poles, right. Then we are checking the controllability of the system, and it is found that the system is controllable. Then we are finding the closed-loop function for B_n .

(Refer Slide Time: 34:11)



So, this is how we found it. So, this is the `bn` function. This is how we are finding. Then we are finding the system. Then again, we are checking because if you notice, we have the system x dot. We have the system

$$\dot{X} = A_{CL}X + B_nX_n$$

So now, here's what we are doing: we are checking the controllability of this system. So, this is what we are doing. Controllability of the system with noise: okay, then we are simulating the system for 0 to 5 seconds with a sampling time of 0.01. This is the noise given to us: this part, x_n, is given, and then we are finding the state space model of the overall closed-loop system. And we are using the `lsim` command here to simulate the closed-loop system. We are giving this noise to the system and finding its response over time. So, if you notice here, you can see that the noise of the system has very little effect on the states; we have three states in

and pitch rate, and elevator deflection. So, you see, the value is 10 to the power of minus 9. The effect of the noise is quite minimal in the overall response of the system. But, as a control engineer, if you can increase the supposed value in our specific case, we have chosen the desired eigenvalues (or the poles) at this level. But if you want to increase them, if you put Suppose, for example, in the present scenario these are the pole locations: for example, but if you can place them here, here, or here (sorry, or maybe here), so if you can choose your poles like this, you can get a better response. But the problem is

You need more control authority in the system. So, you should have the proper technique; you can get better results, but you need to apply more control. So, to apply this amount of control, it is quite challenging for the onboard system, because we have some control and some saturation, right? We cannot use as much as we want. There is some limit.

We have to look at those things. So, that's right. In the present scenario, we have those as the desired eigenvalues, and based on that, we have designed the controller; this is our response. So, from this example, the takeaway is how we can design a controller that helps us mitigate or try to mitigate the effects of noise in the system. So, I hope it is useful practically; how can we design an autopilot for your aircraft?

So, let us stop here. We will continue from the next lecture with some new topics. Thank you.