Advanced Aircraft Control Systems With MATLAB / Simulink

Prof. Dipak K. Giri

Department of Aerospace Engineering

Indian Institute of Technology Kanpur

Lecture 40

Backstepping Controller (Contd.)

Hello, everyone. Also, we'll continue the same topic, backstepping control. Here, we're going to have an example and also a MATLAB simulation. So, let's first define the examples we are going to solve. And also, we'll have the MATLAB simulation for the same example. Let us consider the following nonlinear system. Defined by

$$\dot{x}_1 = ax_1 + x_2$$

$$\dot{x}_2 = u \quad \ldots Eq(1)$$

There are no f_2 terms or nonlinear functions, as you have done in the previous example. So here Let us define this as equation number one. And the question is to design control u to stabilize x_1 and x_2 . Okay. So here, there is no desired state to be tracked. Simply, we will design the control to stabilize x_2 , and x_2 is going to control x_1. So we will start the problem to solve it. So here, as you know, in the backstepping control, the second state is assumed to be virtual control, as you have done in the last lecture. So the second state is assumed to be virtual control and which is going to control the x_1 dynamics. So, let's assume $x_2 = \alpha_1$with the virtual control. In this lecture, we will go a little differently to solve this problem. How can we come up with the error dynamics, and how can we use the error dynamics to improve the stability of the overall system? And considering this virtual control concept, the first subsystem can be written as the first subsystem. Yields to be

$$\dot{x}_1 = ax_1 + \alpha_1 \ldots Eq(2)$$

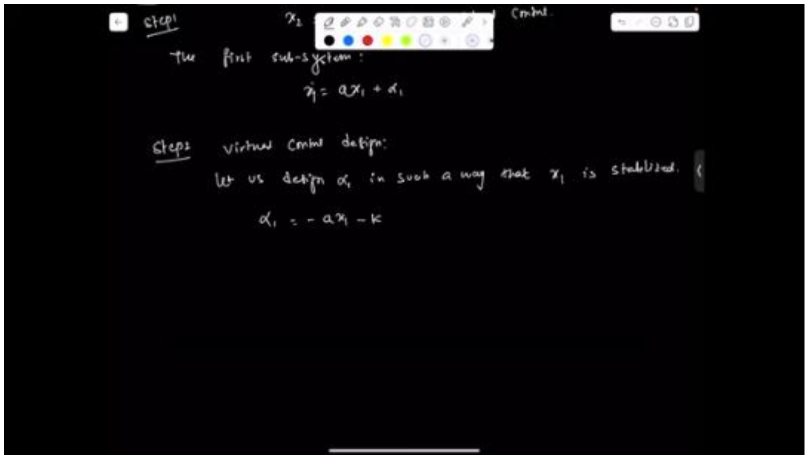
Now, step, this is our step 1. Step 2, let us design virtual control to stabilize the x_1 dynamics. Virtual control design. So here, let us design alpha1 in such a way that x_1 is stabilized. So, let's choose

$$\alpha_1 = -ax_1 - k_1$$

$$= -(a + k_1)x_1 \quad \ldots Eq(3)$$

So, if you substitute this dynamic, this equation in this equation, so x_1 dynamics should be stabilized.

(Refer Slide Time: 04:26)



Now, step three, let us define the Lyapunov function as

$$V_1 = \frac{1}{2}x_1^2$$

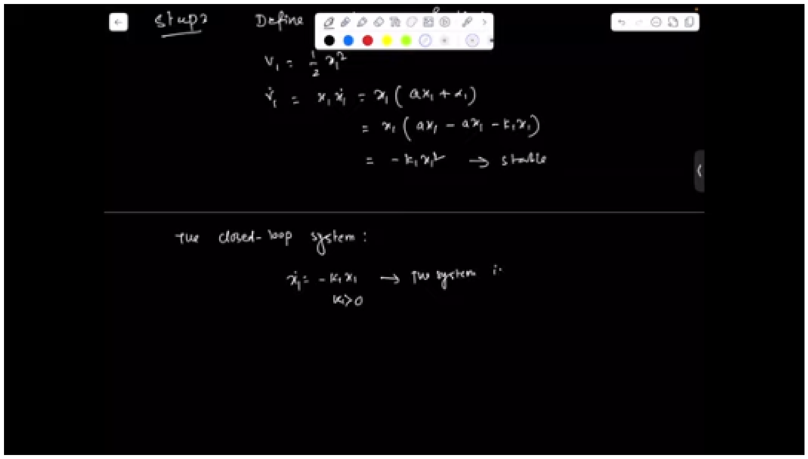
$$\dot{V}_1 = x_1\dot{x}_1 = -k_1x_1^2$$

And if you notice this $\dot{V}_1$ is negative, negative definite. We can say this is stable. Stability can be checked for this particular expression. And for this, we can write if you now substitute this α_1 in this expression, x_1 dynamics. So now, the closed-loop system will be

$$\dot{x}_1 = -k_1x_1$$

And this is, we can say here, k_1 is greater than 0. So, the system is globally asymptotically stable. So now, proceed from the second subsystem as you have done in the last lecture. The state x_2 must be equal to the virtual controller α_1.

(Refer Slide Time: 06:42)



But since they are starting from the defined initial condition, we have to design a control force for x_2 to track the partial control α_1. So here, step four: design control to force x_2 to track the virtual control α_1, right? So for that, let us define the error.

$$e = x_2 - \alpha_1 \quad \ldots Eq(4)$$

And our aim is for e to tend to 0 as t tends to infinity. So if we can do this, we can say this x p From equation 4, we can write

$$x_2 = e + \alpha_1 = e - (a + k_1)x_1 \quad \ldots Eq(5)$$

So, in this example, we are going a little differently than we did in the previous example. But conceptually, both are the same thing. I think this approach is easier than the previous one. Maybe you can find it. So, here we can write. Substituting five, in equation two.

$$\dot{x}_1 = -k_1 x_1 + e \quad \ldots Eq(6)$$

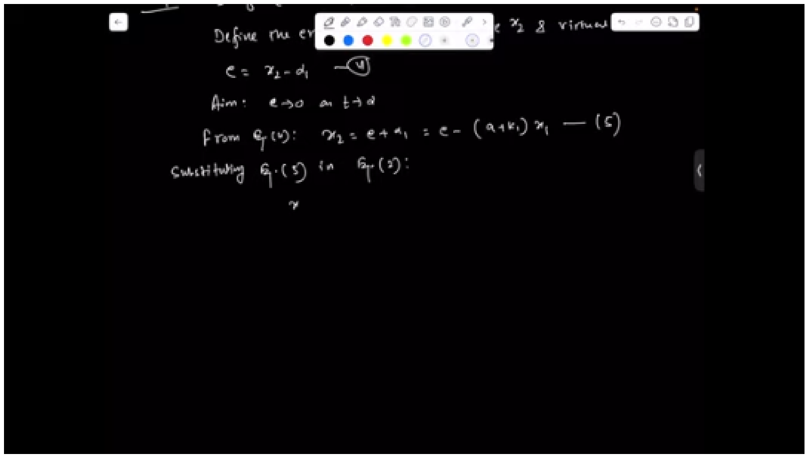
Now, taking the time derivative of equation four, taking the time derivative of Equation 4, we can write

$$\dot{e} = \dot{x}_2 - \dot{\alpha}_1$$

we can substitute $\dot{\alpha}_1$ and solving, we get

$$\dot{e} = (a + k_1)e - k_1(a + k_1)x_1 + u \quad \ldots Eq(7)$$

(Refer Slide Time: 10:20)

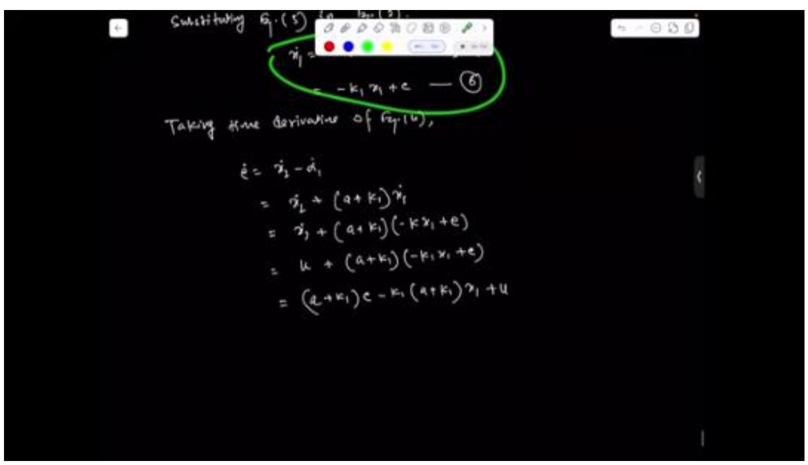


Now, this $\dot{e}$ and this $\dot{x}_1$, these two expressions and this expression will be used to derive the Lyapunov function. Okay, this is seven. Now, step five, let us choose a Lyapunov function as

$$V_2(x_1, e) = V_1 + \frac{1}{2}e_2^2$$

$$\dot{V}_2(x_1, e) = -k_1x_1^2 + e[x_1 + (a + k_1)e - k_1(a + k_1)x_1 + u] \ldots Eq(8)$$

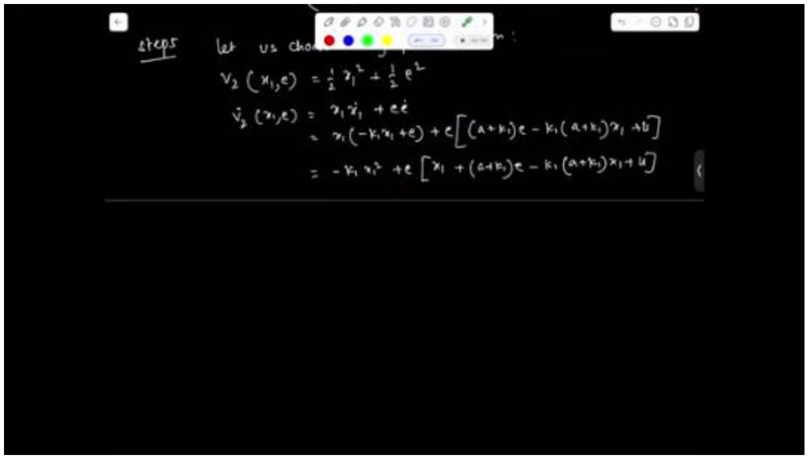
(Refer Slide Time: 13:06)



So now, in this expression, if you notice carefully, this is a stabilizing function, negative definite, right? So now, we have to choose in this expression u in such a way that all the terms are canceled, and here we'll have something $-k_2e^2$. So we have to choose u such that this whole term becomes negative definite. So let's write choose u in such a way that $\dot{V}_2 \leq 0$. So here, we are choosing u as

$$u = -x_1 - (a + k_1)e + k_1(a + k_1)x_1 - k_2 e \ \ldots Eq(9)$$

(Refer Slide Time: 15:49)



Substituting equation 9 in equation 8 and solving, we can write

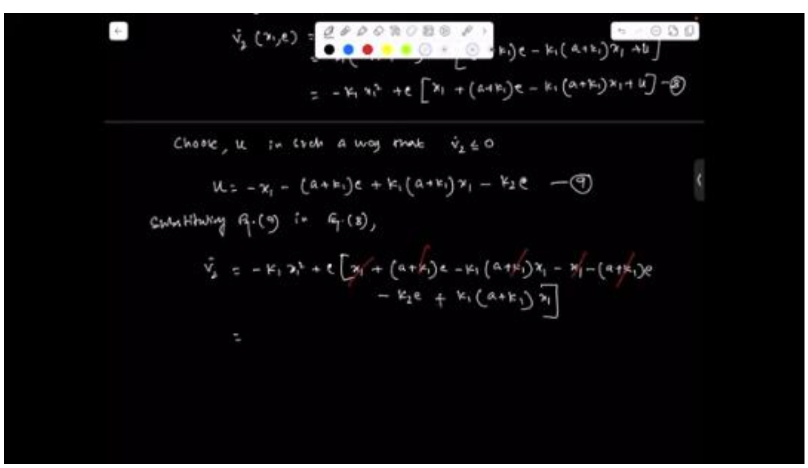
$$\dot{V}_2 = -k_1 x_1^2 - k_2 e^2$$

so this is we can write just make it a definite and hence we can write. And here also k_1 and k_2 are greater than 0 so here we can write

$$x_1(t), e(t) \rightarrow 0$$

as t tends to infinity for all initial values of $x(0)$ and $e(0)$ okay now what about x_2 here actually if you notice you are proving x_1 and e right what about the x_2 function how it is going to zero so if x_1 is and e is going to zero in that case we can write uh we can say that. We can say this whole term going to 0, right? This term going to 0.

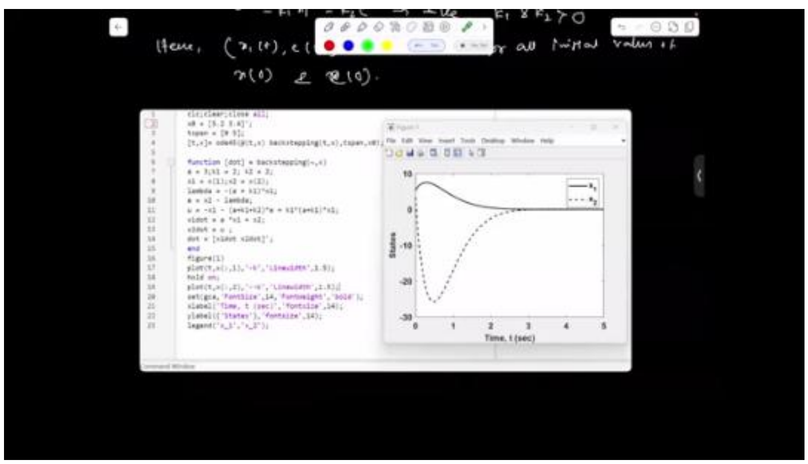
(Refer Slide Time: 18:50)



α going to 0. Is α going to 0? x_2 is going to 0. So this is how we can say all the states are going to 0. So now we can write.

Okay. So this is the overall control in this particular example. This u, this is our overall control. Now we'll have the MATLAB code for this example. And we'll see how the state is going to zero. This is the MATLAB code. Here we are simulating this. These are the initial values of x_1 and x_2, and you are simulating the program for five seconds. This is only four to five to solve the differential equation, and this is the function backstepping we have used. And here we have chosen a equal to 3, k_1 equal to 2, and k_2 equal to 2. How these values are chosen, you can refer to the previous lecture. We came up with some conditions, and this is how we are choosing and defining the variables. Here we have α_1, instead of α_1 , we are writing λ, and this λ is nothing but α_1 . This is the overall control you have obtained. For the stability analysis. This is the $\dot{x}$ dynamics. This is x_1 dynamics, the first equation. Second is the x_2 dynamics.

(Refer Slide Time: 21:15)



And these are the variables going to the function. And this is the figure. These are labels to define the figure for this particular example. And you can see that over time, the states are going to zero. And yes, the system is stable for the particular control algorithm you obtained in equation nine. This is basically stabilizing control which makes the system stable. So this is how we can write the MATLAB code for this particular example. So if you have any other dynamical system, you can easily come up with the MATLAB code. So let's stop it here. We'll continue from the next lecture with different examples.

Thank you.