

PROFESSOR: Hi. Today I'd like to talk about signals and systems again. At this point, you're probably familiar with the motivation for why we're talking about discrete linear time and variant systems, and also with a few of the representations that we're going to end up using in this course. But you're still not sure what it is that we're trying to accomplish. Or where's the part where we get to predict the future based on the fact that we are capable of manipulating these systems?

Well, we actually have to be capable of manipulating these systems. And at this point, we can describe this system as we see it, but we can't also manipulate its representation in ways that make sense to us. So the thing that I'm going to do today is talk about different system equivalences and how to take a system and solve for an expression that represents a complex system and also how if you know that some things in your system are equivalent, how you can convert between them. At that point, we should be able to talk about poles, which is how we're going to actually predict the future.

So different equivalences that I'd like to talk about. I'm first going to briefly review the facts that last time we discovered the notion of system function, right? We can take a representation of a system and abstract it away into some sort of function, where we take the input as it's given to us and then multiply it by this function and then get the output that we're interested in.

How do we deal with something more complex? I mean, y is all the way over here. And we've got multiple system functions. And I don't even know what happens here, but it doesn't have to be that scary. Let's break it down.

One of the easiest ways to approach something like this is to identify each position where you have a new signal, or if you were to sample here, you would have a new signal, and label those values appropriately. You can then start with your final output and then back solve for the values that you're interested in as a consequence of that final output.

In this particular example, y is going to be y_2 plus y_3 . y_2 is going to be y_1 times H_2 , where H_2 is some system function. And it probably is abstracting away some combination of gains, delays, and adders like this one here. y_1 is going to be x times H_1 . And Y_3 is going to be x times H_3 .

Now I've got all my expressions in terms of either y or something for which I have an equivalent expression for x . So I can do my substitutions, come up for an expression for y over x , in terms of H_1 , H_2 , and H_3 .

Here I've just made the substitutions of the equations above and factored out the x . If I wanted the system function, I would then just divide by x . And then I would have y over x is equal to this expression.

The thing I wanted to indicate is that if I wanted to abstract this away into its own box-- maybe I wanted like a big H or an H_0 or something like that-- and it represented what was happening in this top line, cascading two system functions is the functional equivalent of multiplying them together. So if I have an expression for H_1 and I have an expression for H_2 , and I want the expression that is equal to cascading H_1 and H_2 , I just multiply them together.

Likewise, if I want an expression for the linear combination of two system functions applied to an input individually, like the combination H_1 and H_2 , and H_3 , it's a summation of those two values which is expressed here. This is the same as the relationship that we reviewed in a very basic sense when we were originally doing the accumulator. The only thing I'm attempting to indicate is that, that relationship scales to an arbitrary level of complexity. So if you need to, you could shift around these values, if you can find some sort of equivalence.

Let's see what happens when H_2 is equal to H_3 . I'm going to take my operator equation. What this means is that if I wanted to rewrite this block diagram, I could do so by doing-- This is really similar to bubble pushing. If you've done 6.004 or 6.002 and have experience with logic gates, I just wanted to indicate that it's also a thing that you can do for block diagrams and system functions.

There's one more type of equivalence that I want to talk about. I call it feedback equivalence. Here's our normal accumulator rate. If I wanted to represent this feedback system as a feed forward system, what would I have to do? Well, the first time that I sampled from x , it would just be y . So right now this diagram matches for the first time step.

On the second time step, if I had an input from x from the previous time step, I would also want to account for it by putting in a delay and then summing it with the current value of x in order to get y . At the second time step, I would want access to the starting value, the value from the previous time step, and the value from the current time step.

And one more time, to exhaust the example, at the third time step, my output would be a linear combination of the starting value, the value from the first time step, the value from the second time step, and the value from the current third time step.

We'd end up doing this ad nauseum to model our feedback system. So it's difficult to do on paper, but it turns out there's a great relationship between these two equivalences and things that we already know from-- I want to say high school calculus or possibly 18.01, 18.02.

Geometric sequences. When we solved for the system function, we found an expression for our feedback system. If I wanted to find an equivalent expression using this feed forward system, I would look at this infinite summation of x terms. So if I wanted to know something about the long-term behavior of the system, in terms of this system function, I would solve for this expression and then using my knowledge of geometric sequences, in order to express the long-term behavior.

In the general sense, in this course, we're going to be looking at the unit sample response of a system. What that means is, if the only thing I ever do for input is a single value of 1 at time 0, then what does my output look like? The reason we're looking at the unit sample response is because it's (a) the simplest way to look at the long-term behavior of a discrete linear time invariant system. But the other reason (b) is -- once we have this, we can also use it to do things like to make

predictions about the long-term step response and other more complicated input signals.

In the case of the accumulator, if I input 1 at time 0, my output is going to be 1 forever more. That's reflected in the coefficient of my geometric sequence. If I want to know what my long-term response is going to look like, I can look at the coefficient of R and make a decision about whether or not I'm going to diverge or converge or do neither.

So if I put a coefficient on R , whatever p_0 converges to is what my system is going to converge to. So using my knowledge of p_0 , I can make long-term predictions about the behavior of the system. Next time I'm going to go over some general classifications of those behaviors for the system and how to more effectively use our knowledge of p_0 and how to deal with things like second order systems.