PROFESSOR: Last time, which is now two weeks ago, we started the talk about the signals and systems approach, by which I mean, think about a system by the way it transforms its input signal into an output signal. That's kind of a bizarre way of thinking about systems. I demonstrated that last time by thinking about a mass and a spring, something you have a lot of experience with, but you probably didn't use this kind of approach for thinking about it. Over this lecture and over the next lecture, what I'd like to do is show you the advantages of this kind of approach. And so for today, what I'd like to do is talk about how to think about feedback within this structure, and I'd like to also think about how you can use this structure to characterize the performance of a system in a quantitative fashion.

So first off, I want to just think about feedback. Feedback is so pervasive that you don't notice it most of the time. You use feedback in virtually everything that you do.

Here is a very simple example of driving a car. If you want to keep the car in the center of your lane, something that many people outside of Boston, at least, think is a good idea, then you are mentally doing feedback. You're constantly comparing where you are to where you'd like to be and making small adjustments to the system based on that.

When you think of even the most simple of systems, like the thermostat in a house-- I'm not talking about a cheap motel. The cheap motels don't do this. But in a real house, there's a thermostat, which regulates the temperature. That's important, because if the temperature suddenly drops, it would never do that of course, but if the temperature ever dropped, it could compensate for it.

Here's one of my favorite examples. Feedback is enormously pervasive in biology.

There's no general rules in biology, but to a first cut, everything is regulated. And in many cases, the regulation is amazing.

Here's an example from 6.021, where it's illustrating the system that your body uses to regulate glucose delivery from food sources to every cell in your body, which is crucial to your being cognizant and mobile. And the idea is that it does that with amazing precision despite the fact that eating and exercise are enormously episodic. In order for you to remain healthy and functional, you need to have something between approximately two and ten millimoles per liter of glucose in your blood at all times. Were it to go higher than that, systematically, you would develop cardiac problems and could lead to even congestive heart failure. If you were to have lower than two millimoles per liter, you would go comatose.

That's a very narrow range, two to five, especially because what we eat is so episodic, when we exercise is so episodic. It's amazing, and just to dramatize how amazing it is, how much sugar do you think circulates in your blood right now? Well, if you convert five millimoles per liter, and if you assume the average person has about three liters of blood, which is true, and if you calculate it, that comes out to 2.7 grams. That's this much. This is 2.7 grams of table sugar.

So this is how much sugar is in your blood, now. This is what's keeping you healthy. This is what's keeping you from becoming comatose. Or not. How much sugar do you think is in this?

AUDIENCE: The amount in that other cup.

PROFESSOR: Exactly. We call that the Theory of Lectures. I don't want to look like an idiot, so my problems make sense. This is the amount of sugar in a can of soda. Numerically, this is 39 grams. That's more than 13 times this much.

So when you down one of these, it's very important that the sugar gets taken out of the blood quickly, and it does. And that happens by way of a feedback system, and the feedback system is illustrated here. Basically, the feedback involves the hormone insulin, and that's why insulin deficiency is such a devastating disease.

Finally, everything we do has feedback in it. Think about how your life would be different if it didn't. Even a simple task, like removing a light bulb, would be virtually impossible except for the fact that you get feedback from everything.

Your hand's amazing. You have touch sensors, you have proprioceptive sensors, you have stress sensors on the muscles and ligaments, and they all coordinate to tell you when to stop squeezing on the light bulb so you don't break it. That's all completely amazing, and what we'd like to do, then, is think about that kind of a system, a feedback system, within the signals and systems construct.

As an example, I want to think through the WallFinder problem that you did last week in Design Lab. I'm sure you all remember that, in that problem, we were trying to move the robot to a fixed distance away from the wall, and we thought about that as a feedback system comprised of three parts, a controller, a plant, and a sensor. We wrote difference equations to characterize each of those parts, and then we figured out how to solve those difference equations to make some meaningful prediction about how the robot would work.

So just to make sure you're all on board with me, now, here's a question for you. These are the equations that describe the WallFinder problem. How many equations and how many unknowns are there? Take 20 seconds, talk to your neighbor, figure out an answer between (1) and (5). Unlike during the exam next week, you are allowed to talk.

[CLASSROOM SIDE CONVERSATIONS]

PROFESSOR: T and K are no.

[CLASSROOM SIDE CONVERSATIONS]

PROFESSOR: OK, so what's the answer, number (1), (2), (3), (4), or (5)? Everybody raise your hands. Let me see if I got the answer. OK. Come on, raise your hands. Come on, everybody vote.

If you're wrong, just blame it on your neighbor. You had a poor partner. That's the idea. Right? So you can all vote, and you don't need to worry about being wrong. And you're all wrong, so that worked out well.

OK, so I don't like the-- so the predominant answer is number (2). I don't like number (2). Can somebody think of a reason, now that you know the answer by the Theory of Lectures, the answer is not (2)? Why isn't the answer (2)? Yeah?

AUDIENCE: Oh, I was saying (5).

PROFESSOR: You were saying (5). Why did you say (5)?

AUDIENCE: I don't remember for sure, but can you substitute n for different things? Like Df of n, you could substitute n for, and if you did that you would have two equations, which is [INAUDIBLE].

PROFESSOR: [UNINTELLIGIBLE] kind of thing, you count before or after doing substitution and simplification. And I mean to count before you do simplification. Any other issues? Yeah?

AUDIENCE: Is D0 [UNINTELLIGIBLE]

PROFESSOR: Say again?

AUDIENCE: D0 [UNINTELLIGIBLE]

PROFESSOR: Is **D0_n** different or the same from **D0_n-1?** That's the key question. So I want to think about this as a system of algebraic equations, and if I do that, then there's a lot of them. So it looks like there's three equations. The problem with that approach is that there's actually three equations for every value of n.

If you think about a system of equations that you could solve with an algebra solver, you would have to treat all the n's separately. That's what we call the samples approach.

So here's a way you could solve them. You could think about what if k and t are parameters, so they're known, they're given? What if my input signal is known, say it's a unit sample signal, for example? What would I need to solve this system?

Well, I'd need to tell you the initial conditions. So in some sense, I want to consider those to be known. So my knowns kind of comprise t and k, the initial conditions for the output of the robot, and the sensor, all of the input signals because I'm telling you the input and asking you to calculate the output.

My unknowns are all of the different velocities for all values of n bigger than or equal to 0 because I didn't tell you those. All of the values of robot's output at samples n bigger than 0. All the values of the sensor output for values n bigger than 0. So I get a lot of unknowns, infinitely many, and I get a lot of equations, also infinitely many. So the thing I want you think about is, if you're thinking about solving difference equations using algebra, that's a big system of equations.

By contrast, what if you were to try to solve the system using operators? Now, how many equations and unknowns do you see? By the Theory of Lectures.

OK, raise your hand. Or talk to your neighbor so you can blame your neighbor. Talk to your neighbor. Get a good alibi.

[CLASSROOM SIDE CONVERSATIONS]

PROFESSOR: So the idea here is that if you think about operators instead, so that you look at a whole signal at a time, then each equation only specifies one relationship among signals, and there's a small number of signals. So if I think about the knowns being k, t the parameters and the signal d(i), and if I think about the unknowns being the velocity, the output, and the sensor signal, then I get three equations and three unknowns. So one of the values of thinking about the operator approach is that it just simply reduces the amount of things you need to think about. It reduces complexity.

That's what we're trying to do in this course. We're trying to think of methods that allow you to solve problems by reducing complexity. We would like, ultimately, to solve very complicated problems, and this operator approach is an approach that lets you do that.

It does a lot more than that, too. It also generates new kinds of insights. So it lets you focus on the relations, but the relation is now not quite the same as we would have expected from algebra.

Now, the relation between the input signal and the output signal is an operator. We're going to represent the operation that transforms the input to the output by this symbol, h. We'll call that the system functional. It's an operator. It's a thing that, when operated on x, gives you y.

And this is one of the main purposes of today's lecture, it's also convenient to think about h as a ratio. We like to think of it that way because, as we'll see, there's a way of thinking about h as a ratio of polynomials in R. So two ways of thinking about it. We're trying to develop a signals and systems approach for thinking about feedback.

We want to think about the input goes into a box. The box represents an operation. We will characterize that by a functional. We'll call the functional h. The functional, when applied to the input, produces the output, and what we'd like to do is infer what is the nature of that functional, and what are the properties of the system that functional represents.

OK, so I see that you're all with me. Think about the WallFinder system. Think about the equations for the various components of that system when expressed an operator form. And figure out the system functional for that system. figure out the ratio of polynomials that can be in r that is represented by h. Take 30 seconds, talk to your neighbor, figure out whether the answer is (1), (2), (3), (4), or (5).

So what's the answer -- (1), (2), (3), (4), or (5)? Come on, more voter participation. All right? Blame it on your partner. OK, virtually 100% correct.

So the idea is algebra. You solve the operator equations exactly as though they were algebraic. Here, I've started with the second equation and just done substitutions until I got rid of everything other than d(0) and d(i). So I express v in terms of ke, then I express e in terms of d(i) minus rd(0). Then I'm left with one equation that relates d(0) and d(i), which I can solve for the ratio. And the answer comes out there, which was number (3).

Point is that you can treat the operator just as though it were algebra, so that results in enormous implications. But what we want to understand is what's the relationship between that functional, that thing that we just calculated, and the behaviors. These are the kinds of behaviors that you observed with the WallFinder system. When you built the WallFinder system, depending on what you made k, you could get behaviors that were monotonic and slow, faster and oscillatory, or even faster and even more oscillatory. And what we'd like to know is, before we build it, how should we have constructed the system so that it has a desirable behavior?

And, incidentally, are these the best you can do? Or is there some other set of parameters that's lurking behind some door that we just don't know about, and if we could discover it, it would work a lot better? So the question is, given the structure of our problem, what's the most general kind of answer that we can expect? And how do we choose the best behavior out of that set of possible behaviors?

So that's what I want to think about for the rest of the hour and a half, and I want to begin by taking a step backwards and look at something simpler. The idea's going to be the same as the idea that we used when we studied Python. I want to find simple behaviors, think about that as a primitive, and combine primitives to get a more complicated behavior, so I want to use an approach that's very much PCAP. Find the most simple behavior, and then see if I can leverage that simple behavior to somehow understand more complicated things.

So let's think about this very simple system that has a feedback loop that has a delay in it and a gain of P0. What I want to do is think about what would be the response of that very simple system if the input were a unit sample. So find y, given that the input x is delta.

In order to do that, I have to start the system somehow. I will start it at rest. You've all seen already, I'm sure, that rest is the simplest assumption I can make. I'll say something at the end of the hour about how you deal with things that are not at rest.

For the time being, we'll just assume rest because that's simple.

Assume that the system is at rest, that means that the output of every delay box is 0. That's what rest means. If the output of this starts at 0, then the output of the scale by P0 is also 0. And if that's 0, and if the input is at 0 because I'm at time before 0, then the output is 0, indicated here.

So now if I step, then the input becomes 1 because delta of 0 is 1. The output of the delay box is still 0, so the first answer is 1, the 1 just propagates straight through [UNINTELLIGIBLE] box. Then I step, and the 1 that was here goes through the r and becomes 1, the 1 goes through P0 and becomes P0, but at the same time, the 1 that was at the input goes to 0 because the input is 1 only at time equals 0. So the result, then, is that after one step, the output has become P0. This propagated to 1, that became P0, add it to 0, and it became P0, so now the answer, which had been 1, is P0.

On the next step, a very similar thing happens. The P0 that was here becomes the output of the delay, gets multiplied by P0 to give you P0 squared, gets added to 0 to give you P0 squared, et cetera. The thing that I want you to see is that the output was in some sense simple. The value simply increased as P0 to the n, geometrically.

There's another way we can think about that. I just did the sample by sample approach, but the whole theme of this part of the course is the signals approach. If I think about the whole signal in one fell swoop, then I can develop an operator expression to characterize the system. The operator expression says the signal y is constructed by adding the signal x to the signal P0RY. If I solve that for the ratio of the output to input, I get 1 over (1 minus P0R).

Again, going back to the idea that I started with, that we're going to get ratios of polynomials in R now the R is in the bottom, and now I can expand that just as though it were an algebraic expression. I can expand 1 over P0R in a power series by using synthetic division. The result is very similar in structure to the result we saw in sample by sample. It consists of an ascending series in R, which means an ascending number of delays. Every time you increase the number of delays by 1, you also multiply the amplitude by P0, so this is, in fact, the same kind of result, but viewed from a signal point of view.

Finally, I want to think about it in terms of block diagrams. Same idea, I've got the same feedback system, but now I want to take advantage of this ascending series expansion that I did and think about each of the terms in that series as a signal flow path through the feedback system. So one, the first term in the ascending series, represents the path that goes directly from the input to the output, passing through no delays.

The second term in the series, P0R, represents the path that goes to the output, loops around, comes back through the adder, and then comes out. In traversing that more complicated path, you picked up 1 delay and 1 multiply by P0. Second term, two loops. Third term, three loops. Fourth term, four loops.

The idea is that the block diagram gives us a way to visualize how the answer came about. It came about by all the possible paths that lead from the input to the output. Those possible paths all differed by a delay, and that's why the decomposition was so simple, each path corresponding to a different number of delays through the system. That won't always be true from more complicated systems, but it is true for this one.

This flow diagram also lets you see something that's extremely interesting. Cyclical flow paths, which are characteristic of feedback-- feedback means the signal comes back. Cyclical flow paths require that transient inputs generate persistent outputs. They generate persistent outputs because the output at time n is not triggered by the input at time n.

It's triggered by the output at time n minus 1. It keeps going on itself. That's fundamental to feedback. There's no way of getting around that. That's what feedback is.

And it also shows why you got that funny oscillatory behavior in WallFinder. There wasn't any way around that. Feedback meant that you were looping back. That meant that there was a cycle in the signal flow paths. That means that even transient signals, signals that go away very quickly like the [INAUDIBLE] sample, generate responses that go on forever.

So that's a fundamental way of thinking about systems. Systems are either feedforward or feedback. Feedforward means that there are no cyclic paths in the system. No path in the system that take you from the input to the output has a cycle in it.

That's what acyclic means. That's what feedforward means. Acyclic, feedforward, those all have responses to transient inputs that are transient.

That contrasts with cyclic systems. A cyclic system has feedback and will have the property that transient signals can generate outputs that go on forever. OK, how many of these systems are cyclic? Easy questions. 15 seconds, talk to your neighbor.

OK, so what's the answer? How many? OK, virtually 100%. Correct, the answer's (3).

I've illustrated the cycles in red, so there's a cycle here, there's two cycles in this one, and there's a cycle here. So the idea is that, when you see a block diagram, one of the first things you want to characterize, because it's such a big difference between systems, is whether or not there's a cycle in it. If there's a cycle, then you know there's feedback. If there's feedback, then you know you have the potential to have a persistent response to even a transient signal.

OK, so if you only have one loop of the type that I started with, where we had just one loop with an R and a P0, then the question is, as you go around the loop, do the samples get bigger, or smaller, or do they stay the same? That's a fundamental characterization of how the simple feedback system works. So here, if on every cycle the amplitude of the signal diminishes by multiplication by half, that means that the response ultimately decays. Mathematically, it goes on forever, just like I said previously, but the amplitude is decaying, so practically it stops after a while. It becomes small enough that you lose track of it.

By contrast, if every time you go around the loop, you pick up amplitude, if the amplitude here were multiplied by 1.2, then it gets bigger. So the idea, then, is that you can characterize this kind of a feedback by one number. We call that number the pole.

Very mysterious word. I won't go into the origins of the word. For our purposes, it just simply means the base of the geometric sequence that characterizes the response of a system to the unit sample signal.

So here, I've showed an illustration of what can happen if p is 1/2, p is one, p is 1.2, which you can see decay, persistence, divergence. Can you characterize this system by P0? And if so, what is P0? Yes? No?

[CLASSROOM SIDE CONVERSATIONS]

PROFESSOR: Yeah, and virtually everybody's getting the right answer. The right answer's (2). So we like algebra. We like negative numbers, so we're allowed to think about poles being negative. In fact, by the end of the hour, we'll even think about poles having imaginary parts, but for the time being, this is fine. If the pole were negative, what that means is the consecutive terms in the unit sample response, the response of the system to a unit sample signal, the unit sample response, the unit sample response can alternate in sine.

OK, so this then represents all the possible behaviors that you could get from a feedback system with a single pole. If a feedback system has a single pole, the only behaviors that you can get are represented by these three cartoons. So here, this z-axis contains all possible values of P0. If P0 is bigger than 1, then the magnitude diverges, and the signal grows monotonically. If the pole is between 0 and 1, the response is also monotonic, but now it converges towards 0.

If you flip the sign, the relations are still the same, except that you now get sign alternation. So if the P0 is between 0 and minus 1, which is here, the output still converges because the magnitude of the pole is less than 1. But now the sign flips. And if the pole is below minus 1, then you get alternation, but you also get divergence.

The important thing is we started with a simple system, and we ended up with an absolutely complete characterization of it. This is everything that can happen. That's a powerful statement. When I can analyze a system, even if it's simple, and find all the possible behaviors, I have something.

If you have a simple system with a single pole, this is all that can happen. There might be offsets. There might be delays. The signal may not start until the fifth sample, but the persistent signal will either grow without bounds, the k to 0, or do one of those two with alternating sign. That's the only things that can happen, which of course, begs the question, well, what if the system's more complicated.

OK, so here's a more complicated system. This system cannot be represented by just one pole. In fact, the system's complicated enough you should think through how you would solve it. You should all be very comfortable with this sort of thing.

So if you were to think about what if I had a system like so, and I want it to be 1.6 minus 0.63. What would be the output signal at time 2 if the input were a unit sample signal? OK, as with all systems we're going to think about, we have to specify initial conditions. The simplest kind of initial conditions we could think about would be rest.

If I thought about this system at rest, then the initial outputs of the R's would be 0. That's at rest. For times less than 0, the input would be 0. 0 times 1.6 plus 0 times -0.63 plus 0 would give me 0.

Now, the clock ticks. When the clock ticks, it becomes times 0. At times 0, the input is 1. This 0 just propagated down to here, but this was 0, so nothing interesting happens at the R's. But now my output is 1.

Now, the clock ticks. What happens? When the clock ticks, this 1 propagates down here. This 0 propagates down here, but that was 0. This 1 goes to 0 because the input's only 1 at times 0.

So what's the output? 1.6. Now, the clock ticks. What happens?

Well, this 1 comes down here. This 1.6 comes down here. This 0 becomes another 0 because the input has an infinite stream of 0's after the initial time. So what's the output? Well, it's 1.6 times 1.6 plus 1 times -0.63, so the answer is number (3).

Yeah? OK. I forgot to write it up there, so the answer's in red down here. 1.6 squared minus 0.63. OK?

The point is that it's slightly more complicated to think about than the case with a single pole, and in fact, if you use that logic to simply step through all the responses, you get a response that doesn't look geometric. The geometric sequences that we looked at previously either monotonically increased, monotonically decreased towards 0, or give one of those two things and alternated. This does none of those behaviors. So the point is Freeman's an idiot. He spent all that time telling us what one pole does, and now two poles does something completely different. Right?

So the response is not geometric. The response grows and then decays. It never changes sign. It does something completely different from what we would have expected from a single pole system.

As you might expect from the Theory of Lectures, that's not the end of the story. So the idea is to now capitalize on this notion that we can think about operators as algebra. If our expressions behaved like I told you they did last lecture, if they behaved as entities upon which-- if they are isomorphic with polynomials, as I said, then there's a very cute thing we can do with this system to make it a lot simpler. The thing we can do is factor.

If we think about the operator expression to characterize this system, the thing that's different is that there's an R squared. But if R operators work just like polynomials-- you can factor polynomials. That's the factor theorem from algebra. And if I factor it, I get two things that look like first-order systems. Well, that's good.

The factored form means that I can think about this more complicated system as the cascade of two first-order systems. Well, that's pretty good. In fact, it doesn't even matter what order I put them in because, as we've seen previously, if the system started at initial rest, then you can swap things because they obey all the principles of polynomials, which include commutation.

So what we've done, then, is transform this more complicated system into the cascade of two simple systems, and that's very good. Even better, we can think about the complicated system as the sum of simpler parts, and that uses more intuition from polynomials. If we have one over a second-order polynomial, we can write it in a factored form here, but we can expand it in what we call partial fractions.

We can expand this thing in this sum, and if you think about putting this over a common denominator and working out the relationship, this difference, 4.5 over 1 minus 0.9R minus 3.5 over 1 minus 0.7R. That's precisely the same using the normal rules for polynomials. That's precisely the same as that expression. But the difference, from the point of view of thinking about systems, is enormous.

We know the answer to that one. That's the sum of the responses to two first-order systems, so we can write that symbolically this way. We can think about having a sum system that generates this term. This term is a simple system of the type of that we looked at previously that, then, gets multiplied by 4.5.

I'm just factoring again. I'm saying I've got something over something, which means that I can put something in each of two different parts of two things that I multiply together. And I can think about this as having been generated by this system, and you just add them together. The amazing thing is that that says that, despite the fact that the response looked complicated, it was in fact the sum of two geometrics. So it wasn't very different from the answer for a single pole.

What I've just done is amazing. I've just taken something that, had you studied the difference equations and had you studied the block diagrams, it would have been very hard for you to conclude that something this complicated has a response that can be written as the sum of two geometrics. By thinking about the system as a polynomial in R, it's completely trivial. It's a simple application of the rules for polynomials that you all know.

So what we've shown, then, is that this complicated system has a way of thinking about as just two of the simpler systems. The complicated response that grew and decayed, that's just the difference, really, 4.5 minus 3.5. It's the weighted difference of a part that goes 0.7 to the n, then a different part that goes 0.9 to the n.

So far, we've got to results, the n equals 1 case, the first-order polynomial in our case, the one pole case, that's trivial. It's just a geometric sequence. The response is just a geometric sequence. If it happens to be second-order, this is second-order because when you write the operator expression, the polynomial in the bottom is second-order, second-order polynomial in R.

This second-order system has a response that looks like two pieces. Each piece looks like a piece that was from a first-order system. And in fact, that idea generalizes.

If we have a system that can be represented by linear difference equation with constant coefficients that will always be true if the system was constructed out of the parts that we talked about, adders, gains, delays. If the system is constructed out of adders, gains, and delays, then it will be possible to express the system in terms of one difference equation. General form is showed here.

Y then can be constructed out of parts that are delayed versions of Y and delayed versions of X. If you do that, then you can always write the operator that expresses the ratio between the output and the input as the ratio of two polynomials. That will always be true.

So this, now, is the generalization step. We did the n equals 1 case, we did the n equals 2 case, and now we're generalizing. We will always get, for any system that can be represented by a linear difference equation with constant coefficients, we can always represent the system functional in this form.

Then just like we did in the second-order case, we can use the factor theorem to break this polynomial in the denominator into factors. That comes from the factor theorem in algebra. Then we can re-express that in terms of partial fractions.

And what I've just showed is that, in the general case, regardless of how many delays are in the system, if the system only has adders, gains, and delays, I can always express the answer as a sum of geometrics. That's interesting. That means that if I knew the bases for all of those geometric sequences, I know something about the response. The bases are things we call poles. If you knew all the poles, you'd know something very powerful about the system.

So every one of the factors corresponds to a pole, and by partial fractions, you'll get one response for each pole. The response for each pole goes like pole to the n. You know the basic shape. You don't know the constants, but you know the basic shape of the response just by knowing the poles.

We can go one more step, which makes the computation somewhat simpler. I used the factor theorem. Here, I'm using the fundamental theorem of algebra, which says that if I have a polynomial of order n, I have n roots. The poles are related to the roots of the R polynomial.

The relationship is take the functional, substitute for R -- 1 over Z. Re-express the functional as a ratio of polynomials in Z. The poles are the roots of the denominator.

So recapping, I started with a first-order system. I showed you how to get a second-order system. I showed that, in general, you can use the factor theorem to break down the response of a higher-order system into a sum of responses of first-order systems. Now, I've shown that you can use the fundamental theorem of algebra to find the poles directly, and then by knowing the poles, you know each of the behaviors, monotonic divergence, monotonic convergence, or alternating signs.

And so here is this same example that I started with, worked out by thinking about what are the poles. The poles are 0.7 and 0.9, which we see by a simple application of the fundamental theorem of algebra. OK, we got a long way by just thinking about operators as polynomials. We haven't done anything that you haven't done in high school. Polynomials are very familiar and we've made an isomorphism between systems and polynomials.

OK, make sure you're all with me. Here's a higher order system. How many of these statements are true -- 0, 1, 2, 3, 4, or 5? Talk to your neighbor, get an answer.

So how many are true -- 0, 1, 2, 3, 4, or 5? Oh, come on. Blame it on your neighbor. You weren't talking, but I didn't hear you not talking. How many are true -- 0, 1, 2, 3, 4, or 5? Raise your hands.

AUDIENCE: They can't all be true.

PROFESSOR: They can't all be true. Are they mutually contradictory?

AUDIENCE: Well, yeah. 5 --

PROFESSOR: N1 of the above, that sounds like-- OK, so you've eliminated 1. Which one's true? How many statements are true?

Looks like about 75%. Correct? What should I do? How do I figure it out? What's my first step? What do I do?

AUDIENCE: Operators.

PROFESSOR: Operators, absolutely. So turn it into operators. So take the difference equation, turn it into operators. The important thing to see is that there are three Y terms. Take them all to the same side, and I get an operator expression like that.

The ones that depend on X, there are two of them, that's represented here. The thing this is critical for determining poles is figuring out the denominator. The poles are going to come from this one.

After I get the ratio of two polynomials in R, I substitute 1 over Z for each R. So for this R, I get 1 over Z. For this R squared, I get 1 over Z squared. Then I want to turn it back into a ratio of polynomials in Z, so I have to multiply top and bottom by Z squared. And when I do that, I get this ratio of polynomials in Z.

The poles are the roots of the denominator polynomial in Z. The poles are minus 1/2 and plus 1/4. So the unit sample response converges to 0. What would be the condition that that represents?

AUDIENCE: Take the polynomial on the bottom.

PROFESSOR: Something about the polynomial on the bottom. Would all second-order systems have that property that the unit sample response would converge to 0?

AUDIENCE: [INAUDIBLE]

PROFESSOR: Louder?

AUDIENCE: Absolute value of the poles?

PROFESSOR: Absolute value of the poles has to be?

AUDIENCE: Less than 1.

PROFESSOR: Less than 1. If the magnitude of the poles is less than 1, then the response magnitude will decay with time. So that's true here, and it would be true so long as none of the poles have a magnitude exceeding 1.

There are poles at 1/2 and 1/4. No, that's not right. It's -1/2 or 1/4.

There's a pole at 1/2. No, that's not right. There's a pole at -1/2.

There are two poles. Yes, that's true. None of the above. No, that's not true. So the answer was (2).

Everybody's comfortable? We've done something very astonishing. We took an arbitrary system, and we've figured out a rule that let's us break it into the sum of geometric sequences. We can always write the response to a unit sample signal, we can always write, as a weighted sum of geometric sequences, and the number of geometric sequences in the sum is the number of poles, which is the order of the operator that operates on Y. OK, so we've done something very powerful.

There's one more thing that we have to think about, and then we have a complete picture of what's going on. Think about when you learned polynomials. One of the big shocks was that roots can be complex. What would that mean? What would it mean if we had a system whose poles were complex valued?

So first off, does such a system exist? Well, here's one. I just pulled that out of the air.

If I think about the functional, 1 over (1 minus R plus R squared) -- if I convert that into our ratio of polynomials in Z and then find the roots, I find that the roots have a complex part. The roots are 1/2 plus or minus root 3 over 2j. There's an imaginary part. So the question is what would that mean? Or is perhaps that system just meaningless?

Well, complex numbers work in algebra, and complex numbers work here, too. So the fact that a pole has a complex value in the context of signals and systems simply means that the pole is complex, that the base of the geometric sequence, that base is complex. So that means that we can still rewrite the denominator, which was 1 minus R plus R squared, we can rewrite that denominator in terms of a product of two first-order R polynomials.

The coefficients are now complex, but it still works. The algebra still works right. That has to work because that's just polynomials. That's the way polynomials behave.

Now, we can still factor it. We can still use the factor theorem. In fact, we can still use the fundamental theorem of algebra to find the poles by the Z trick. That's fine. We can still use partial fractions. All of these numbers are complex, but the math still works.

The funny thing is that it implies that the fundamental modes-- by fundamental mode, I mean the simple geometric for the case of a first-order system, more complex behaviors for higher order systems. The mode is the time response associated with a pole. So the modes are, in this case, complexes sequences. So in general, the modes look like P0 to the n. Here, my modes are simply have a complex value.

So what did I say they were? The poles were 1/2 plus or minus-- so my modes simply look that. Same thing.

The strange thing that happened was that those modes, those geometric sequences, are now have complex values. The first one up here, if I just look at the denominator, these coefficients mean that it's proportionate to the mode associated with this, which is that, which has a real part, which is the blue part, and the imaginary part, which is a red part. There were two poles, plus and minus. The other pole just flips the imaginary part.

So if I have imaginary poles, all I get is complex modes, complex geometric sequences. An easier way of thinking about that is thinking about-- so when we had a simple, real pole, we just had P0 to the n. That's easy to visualize because we just think about each time you go from 0 to 1 to 2 to 3, it goes from 1 to P0 to P0 squared to P0 cubed.

Here, when you're multiplying complex numbers, it's easier to imagine that on the complex plane. Think about the location of the point 1, think about the location of the point P0, think about the location of the point P0 squared, and in this particular case, where the pole was 1/2 plus or minus the square root of 3 over 2 times j, this would be pole to the 0. This is pole to the 1. This is pole squared, pole cubed.

As you can see when you have a complex number, the trajectory in complex space can be complicated. In this case, it's circular. The circular trajectory in the complex plane corresponds to the sinusoidal behavior in time. So there's a correlation between the way you think about the modes evolving on the complex plane and the way you think about the real and imaginary parts evolving in time.

It seems a little weird that the response should be complex. We're studying this kind of system theory primarily because we're trying to gain insight into real systems. We want to know how things like robots work.

How does the WallFinder work? What would it mean if the WallFinder went to position one plus the square root of 3 over 2j? That doesn't make sense.

So there's a little bit of a strange thing going on here. How is it that we need complex numbers to model real things? That doesn't seem to sound right. But the answer is that, if the difference equation had real coefficients, as they will for a real system-- if you think about a real system, like a bank account, the coefficients in the difference equation are real numbers, not complex numbers. If you think about the WallFinder system, the coefficients in the WallFinder system, the coefficients of the difference equations-- the coefficients of the different equations describe the WallFinder behavior were all real numbers.

Here, I'm thinking about the denominator polynomial. If we try to find the roots of a polynomial, and if we find a complex root, if the coefficients were all real, it follows that the complex conjugate of the original root is also a root. That's pretty simple, if you think about what it means to be a polynomial.

If you think about a polynomial is whatever-- so I've got 1 plus Z plus Z squared plus blah, blah, blah. 2, 3 minus 16, if all of those coefficients real, then the only way the-- If P were a root of this polynomial, then P* would have to be a root, too, because if you complex conjugate each of the Z's, it's the same thing as complex conjugating the whole thing because the coefficients are real valued. So the idea then is that if I happen to get a complex root for my system that can be described by real value coefficients, it must also be true that it's complex conjugate is a root. If that happens, the two roots co-conspire so that the modes have canceling imaginary parts.

You can prove that. I'm not worried about you being able to prove that. I just want you to understand that if you have two roots that are complex conjugates, they can conspire to have their imaginary parts cancel, and that's exactly what happens.

Here, in this example, the example that I started with, where the system was 1 over (1 minus R plus R squared), the unit sample responses showed here. You can write as the sum of sinusoidal and cosinusoidal signals, and the sum that falls out has the property that the imaginary parts cancel. It's still useful to look at the imaginary parts the same as it is when you're trying to solve polynomials. It's useful because the period of all of these signals is the same.

If I think about the period of the individual modes, if I think about the period of P0 to the n, 1/2 plus or minus root 3 over 2j to the n, the period of that signal -- I can see in the complex plane, this is the n equals 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 -- the period is 6. If I were to take the minus 1, I would go around the circle the other way. This would be zero term 1, 2, 3, 4, 5, 6, et cetera. Both of the modes, the geometric sequence associated with the two complex conjugate poles, both of those modes had the same period, they're both 6, as does the response to the real part. So you can deduce the period of the real signal by looking at the periods of the two complex signals.

So think about this system. Here, I've got a system whose responses showed here. I'm going to tell you that the response was generated by a second-order system, that is to say a system whose polynomial was second-order, whose polynomial in R was second-order.

Which of these statements is true? I want to think about the pole as being a complex number. Here, I'm showing you the complex number in polar form. It's got a magnitude and an angle, and I'd like you to figure how what must the magnitude have been, and what must the angle have been.

So what's the utility? Why did I tell you the pole in terms of it's magnitude and angle rather than telling it to you in it's Cartesian form as a real and imaginary part? What's good about thinking about magnitude and angle?

Or if I break the pole into magnitude and angle, and if I think about the mode-- the modes are always of the form P0 to the n. If I think about modes as having a magnitude and an angle, when I raise it to the n, something very special happens. What happens? You can separate it.

What is R e to the j omega raised to the n? That's the same as R to the n, e to the j n omega. It's the product of a real thing times a very simple complex thing. What's simple about the complex thing?

The magnitude is everywhere 1 e to the j, the magnitude of that term. So all of the magnitude is here, none of the magnitude is here. All of the angle is here, none of the angle is here. I've separated the magnitude and the angle.

So it's very insightful to think about poles in terms of magnitude and angle because it decouples the parts of the mode. So I can think, then, of this complicated signal that I gave you as being the product of a magnitude part and a pure angle part. From the magnitude part, I can infer something about R. R is the ratio of the nth 1 to the n minus 1-th one, so R is a lot bigger than 1/2. In fact, R in this case is 0.97.

And this is pure angle. This lets me infer something about the oscillations. In fact, I can say something about the period. The period is, here's a peak, here's a peak, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12. The period's 12.

If the period is 12, what is omega? Omega is a number, such that by the time I got up to 12 times it, I got up to 2 pi. What's omega?

AUDIENCE: Pi over 6.

PROFESSOR: 2 pi over 12. So it's about 1/2. So that's a way that I can infer the form of the answer. I ask you what pole corresponds to this behavior.

Well, the decay that comes from the real part, that comes from R to the n. The oscillation, that comes from the imaginary part, and I can figure that out by thinking about the period and thinking about that relationship. So the answer, then, is this one, R is between 1/2 and 1, and omega is about 1/2.

OK, one last-- whoops, wrong button. One last example. So what we've seen is a very powerful way of decomposing systems, so that we can always think about them in terms of poles and complex geometrics, which we will call modes, poles and modes. And that behavior works for any system in an enormous class of systems, so I want to think about one last one, which is Fibonacci sequence.

You all know the Fibonacci sequence. You've all programmed it. We started with that when we were doing Python, and we made some illustrations about the recursion and all that sort of thing by thinking about it. Now, I'm going to use signals and systems do the same problem.

So Fibonacci was interested in population growth. How many pairs of rabbits can be produced from a single pair in a year if it is suppose that every month each pair begets a new pair, from which the second month it becomes productive? OK, it's not quite the same English I would have used. From this statement, you can infer a difference equation.

I've written it in terms of X. What do you think X is? X is the input signal, and here, I'm thinking about X as something that's specifies the initial condition. This is the thing I alluded to earlier. One trick that we use to make it easy to think about initial conditions is that we embed them in the input.

So in this particular case, I'll think about the initial condition arising from a delta function. If I think about X as a delta, then the sequence of results Y0, Y1, Y2, Y3 from this difference equation, is the conventional Fibonacci sequence. It would correspond to what if you had a baby rabbit, one baby rabbit, that's the one, in generation 0, that's the delta. So the input is a way that I can specify initial conditions, and that's a very powerful way of thinking about initial conditions.

So here's the problem. I've got one set of baby rabbits at times 0. They grow up. They have baby rabbits, which grow up at the same time the parents had more baby rabbits, at which point more babies grow into bigger rabbits. And big rabbits have more babies, et cetera, et cetera, et cetera, et cetera, et cetera.

So you all know that. You all know about Fibonacci sequence. It blows up very quickly. What are the poles of the Fibonacci sequence?

The difference equation looks just like the difference equations we've looked at throughout this hour and a half. We can do it just like we did all the other problems. We write the difference equation in terms of R. We rewrite the system functional in terms of a ratio of two polynomials in R. We substitute R goes to 1 over Z, deduce a ratio of polynomials in Z, factor the denominator, find the roots of the denominator, and find that there are two poles.

The poles for the Fibonacci sequence are plus or minus the root of 5 over 2. That's curious. There's no recursion there. It's a different way of thinking about things.

There are two poles. The first pole, the plus one -- 1 plus the root of 5 over 1, corresponds to a pole whose magnitude is bigger than 1. It explodes. There it is.

The second one is a negative number. So the first one is the golden ratio, 1.618... The second one is the negative reciprocal of the golden ratio, which is -0.618... And those two numbers, amazingly, conspire so that their sum, horrendous as they are, is an integer. And in fact, that's the integer that we computed here.

So we've used Fibonacci before to think about the way you structure programs, iteration, recursion, that sort of thing. Here, by thinking about signals and systems, we can think about exactly the same problem as poles. There's no complexity in this problem. It doesn't take N or N squared or N log N or anything to compute. It's closed form.

The answer is (pole one) to the N plus (pole two) to the N. That's it. That's the answer. So what we've done is we found a whole new way of thinking about the Fibonacci sequence in terms of poles, and more than that, we found that that way of thinking about poles works for any difference equation of this type. And we found that poles, this way of thinking about systems in terms of polynomials, is a powerful abstraction that's exactly the same kind of PCAP abstraction that we used for Python.