[SQUEAKING] [RUSTLING] [CLICKING]

DAVID PERREAULT: So what I wanted to do for the next few lectures is talk a bit about control of power converters. And actually, I'm more going to talk about modeling of power converters than control of power converters because I can't really teach control theory in this class.

If you haven't had a control class yet, it's really time to go out and take one because no matter what you're going to do, you're really probably going to need to understand feedback control. But what we can do in this class is show you how one can model power converters to put them-- to get a model in the format that you can use it for feedback control and do basic control.

So let's think about a power converter. Let's start off with our friendly neighborhood boost converter. So maybe I have some input voltage VIN. I'll switch it with some switching waveform q of t. And then I'm going to have a diode. I will have a capacitor. And the converter ends here. And then somewhere, we usually show RL. And I have my input voltage. And I have my output voltage, VOUT.

Now, I usually in my examples, just by way of an aside-- I usually show a resistive load. A lot of loads are resistive. But there's no guarantee that in an actual application, your load looks like a resistor. It could look like whatever.

Maybe a lot of loads you might represent as some maybe Norton equivalent where there's some constant current component and some resistive current component. And I'm just doing the simplified case where the Norton source is 0 and the resistance is RL.

This is just a way of saying in general, you would worry about exactly what the load looks like not just in terms of how much current does it draw. But exactly what the load looked like can affect the dynamics. But for simplicity, let's just talk about resistive loads, which is a very common case.

What do we have to do in a typical power converter application? We want to usually regulate the output voltage to be something we want. So we want to make VOUT go to some reference voltage that I'll call VREF.

VREF might be fixed-- it might be I want 12 volts output all day long-- or it may be something that you dynamically vary based on some other consideration that we're not going to talk about. So it could be a fixed or a variable reference. But for the present time, let's just assume I want to get my output on my converter to be some fixed value that I would like it to be. If I'm creating a 12-volt power supply, I want it to be 12 volts.

And we have to keep the output voltage closer at the reference voltage under a wide variety of conditions. And this is where power electronics is challenging. Firstly, for variations in load-- so maybe I would have RLmin less than what RL is less than RLmax. So this would be heavy load because the lowest resistance gives me the highest power. And this might be light load.

I also need to keep VOUT near Vreference in terms of-- for variations in VIN. So maybe there's a VINmin less than VIN less than VINmax that I've got to deal with.

Why do I have to worry about those? Well, my load's not always going to be the same. Maybe this is controlling a voltage which goes to a bunch of microprocessors or other things or actuators and they're turning on and off. So the effective resistance, the effective power, I have to supply varies, even at fixed voltage.

And here, maybe I'm powered from a battery and the battery discharges over time or I'm charging or whatever. So there's going to be some input range, input voltage range, over which this converter is designed to reject that as a disturbance and just keep the output where I want it.

Now, so far, we've spent a lot of time talking about the steady-state operation. And that's usually always the first thing I look at because it tells me when things are sitting at some place, know some load resistance, some input voltage. What's it going to Do And we said for the boost converter, what I get is VOUT is equal to VIN over 1 minus D.

So this is the steady-state relationship. This is in PSS, in periodic steady state. Once we've settled down, this converter ought to do this. So if I control the duty ratio of my switch, q of t, to this value, this is what I should get for an output voltage.

So you might think, maybe I could just measure the input voltage, set a duty ratio, and get the output voltage I want, and that's the end of the day. But there's two problems with that. One this is only in periodic steady state. And it doesn't say how things will change transiently in between periodic steady-state conditions. And I really care about that because if the voltage goes-- if I decide to change the load, the voltage goes wild until it settles down. I'm probably not going to be happy.

The other thing about this is this is an idealized relationship. When we derived this, we did it by volt-seconds balanced across this inductor. I pretended the inductor had no resistance. I ignored the diode drop. I ignored a whole bunch of things which don't matter for getting an approximate relationship. But they do matter if I need to control this to 12 volts plus or minus 2%.

The model's just not good enough. I need some better way to get my converter to get where I want it to be and maintain it there under variations like this. Any questions about that?

Just to illustrate that, what I've got here is a SPICE simulation of a boost converter which is undergoing a load transient. So what's happening in this is here at some time, at actually at 1.4 milliseconds, we had a 5-ohm-- we originally had a 5-ohm load on the converter. And we put in another 5-ohm load. So it's like some new load has come in.

In this example, we're just holding the duty ratio of the converter constant. So we expect that, barring a lot of second-order effects, the output voltage should go back to the same thing in periodic steady state. But clearly, if the output voltage is going to be the same and I've got smaller net resistance on it because I have the net resistance, the power has to go up. The current has to go up.

So things have to change. And that's what we see happening. So this trace down here is the input current. So the input current was sitting on average around 12 amps. I go through some transient. And I end up around 24 amps. The output voltage had been sitting at 22 volts, goes through some transient, and I get back to 22 volts because it's running at fixed duty ratio.

So what we see here is the stuff we've skipped over so far. We know we could calculate based on periodic steady state-- say that the output capacitor voltage and the average inductor current in both conditions. But we haven't accounted for how does it transiently go between those conditions. And that's what we have to know in order to make our system happy. And so that's what we're going to focus on.

And I'd like you to notice that this response looks like a second-order system. It has a-- looks like a damped LC oscillation, sort of, which is not surprising because I've got one C and one L. And there's load on it. So energy has been taken out of it. So maybe that's not surprising. The question is, what does this thing look like?

So how would I control this thing in the real world? I'd use feedback control. So what are we going to do? We're going to say I have some reference voltage, VREF, that I want my converter to go to. And I provide that voltage reference. And I'm going to compare that voltage reference to the output voltage, VOUT.

And I'm going to generate an error between those two. I'll take that error and feed it into a high-gain compensator. Maybe I could represent that with a transfer function, GC of s. This might be, for example, a PID controller or something fancier-- depends on the topology. This is where having a control class--

But you might think what this does is if I have error, I amplify that error by some amount, either proportionally or look at the integral of the error or something like that, and generate an output, which I'm going to call Vd. What is this signal? This is usually a voltage, although I could represent it with a current. It's a voltage representing the duty ratio I want.

Now I have a voltage representing the duty ratio. I have to put this into some PWM circuit, pulse width modulation circuit, that's going to give me q of t because if I look at up there, I'm talking about in terms of duty ratio. But the actual signal going to my switch is some q of t or some amplified version of q of t from a gate drive that's turning the switch on and off at the right duty ratio.

So the input of this is some value of duty ratio I want. The output of this is some PWM waveform switching at period T with an on time that's dT. This then goes into my actual converter, which also, if you think about it, has inputs of VIN. And it's being affected by what the RL is. And then its state variables are the inductor current and the output voltage, which is also the capacitor voltage, VOUT. And that's what I'm feeding around back here.

So this is a typical closed-loop control system. If VOUT gets smaller than the reference voltage, Verror goes up. I amplify that. My duty ratio goes up. And when my duty ratio goes up, my output voltage goes up and I close the loop. And eventually, hopefully this settles to VOUT close to VREF. Any questions about that picture?

Just out of curiosity, how many people here have had either a classical or state space control class? Good fraction. Good. Let's make that fraction 100% next year because you really want to know this, no matter what you do.

This is a standard feedback system. What we need to do this-- if I put it in the wrong compensator and I put feedback, I could make this thing oscillate and go unstable. We've all seen the effects of bad feedback, like when you put your microphone too close to your speaker. And suddenly, the thing goes crazy. And you can certainly make a power converter go crazy if you're not careful.

And what we need to know in order to make this work is some model that basically says, as I'm changing my duty ratio or something else, like the load resistor, how does this dynamically vary? If I know that, then I can design a control loop for the thing.

And so what I'd like to talk about is how do we build this model, this dynamic model, instead of a periodic steady-state model. What does the dynamic model of this thing look like from variations in duty ratio, for example, to the variations in the output voltage?

And coming back to this picture, we can see that what we have here is a bunch of switching. Here's the switching ripple, the triangular ripple current in the inductor. And it goes through some oscillations. And then it settles down, and the same thing with the capacitor voltage.

Well, I'm not particularly interested in this switching ripple. Why am I not interested in it is because notice that the dynamical behavior is at a very low frequency compared to the switching ripple. And I can't really do anything about the switching ripple, anyways. In other words, that's intrinsic to the converter. I'm going to put some energy in and I'm going to process it through. I'm going to take it out.

If I'm trying to, say, vary my duty ratio on the time scale of this switching, I'm very, very fast, I'm probably going to make the converter do unhappy things. My whole notion was that at least within a cycle, duty ratio is not something that I'm swinging around wildly.

So I'm interested in the behavior of this circuit for frequencies relatively low compared to the switching frequency. So what I want to do is generate models that capture this average behavior, like these low-frequency oscillations, and ignore the switching ripple. Everybody follow that?

So let me think about this in a couple of pieces. Well, first of all, let's just talk about this PWM block. We've talked about this before. This isn't part of the average model. But essentially, what would the PWM block look like? Basically, what I would have is maybe I would generate some triangle wave. This could be a sawtooth, for example, with a period T. And I'll just generate this.

And maybe for simplicity, I'll just say this is a 1-volt signal. And I will compare that to this duty ratio signal. So here's Vd. And Vd is-- maybe it's changing. But it's changing very slowly. So here's Vd. And I'm just going to run this into a comparator. I'm going to call this VTRI. And I'll compare VTRI to Vd and generate q of t.

So all I'm going to do is-- at the beginning of each cycle, what you can see is if this is time, at the beginning of each cycle, q of t is going to go high until the triangle exceeds the V sub d, in which case I'll go low. And then I'll go high at the beginning of the next cycle. And I'll stay high until I hit the end, until I hit the comparison, and so forth-- same again in the next cycle, and so forth.

So by comparing my V sub d to a linearly rising element, I can generate something where this is-- whatever V sub d was here, that's-- represents some duty ratio. That's going to be dT at this point. And I'm going to have dT at this point, which might be different, et cetera. And each cycle, I get a duty ratio that's related to V sub d during that cycle assuming V sub d doesn't vary very quickly.

So this is how I build the PWM generator. That's this block here. That's not how I model it. That's just how I implement it. So from a mental perspective, we can worry about going from some desired duty ratio to some response or the local average duty ratio to some response where I'm assuming that Vd is only varying very slowly within any given switching cycle. And if your compensator lets it very, very quickly, you're probably going to be unhappy.

So that gives me a way to implement this. And I should say, by the way, that if I go buy a PWM chip for a typical DC-to-DC converter or controller chip, I'll usually get a reference, some kind of error amplifier, op-amp, and other things to generate the compensator. I'll get the PWM chip. And in fact, this triangle also serves as my clock. So I can set the clock frequency. I can set T. And then maybe it gives me a gate driver.

So I'll get all this stuff in one chip very often for a simple design. Of course, you can also do a lot of this stuff digitally these days, depending upon what kind of processor you use.

But this doesn't tell me-- this tells me how did I get from duty ratio to the q of t. But it doesn't tell me how I model this whole block. And that's what I'd like to talk about now.

And we said we are not interested in this high-frequency ripple, but only the low-frequency behavior. And towards that end, I'd like to introduce what's known as a local average.

So the local average operator is this. If I had some signal for any signal x of t, some time-varying waveform x of t-- could be a voltage, could be a current, whatever I wanted-- I can define a local average x bar of t is defined as being equal to 1 over T integral between t minus capital T and t of x of tau d tau. This is the local average of a signal. And what is this local average? It's just a moving average.

So if I'm at my signal time right now at time T, what I'm going to do is I'm going to look back over 1 capital T and take the average of the waveform just over that time. And then if I'm at another time, I'll look back and I'll take the average over at that time.

So if I had some time-varying waveform x of t and I said, here's t1, maybe I'd go back. And here's t1 minus capital T. Maybe I would take the average of this waveform over that time. And then if I'm at some other time t2, I'd go back one cycle. And this is t minus T2-- t2 minus capital T. And I'd look at the average over this time period, et cetera.

So at any given point, I'm going to look at what I've been doing over the last capital T and take that average. Why am I doing that? Well, because if I set capital T to be my switching period, I can look at what the average of this variable is over one switching cycle at any point in time.

And we can think of this as basically tracking the low-frequency variations and rejecting the switching ripple. And we can think about it qualitatively. And we can think about it quantitatively. Any questions about the idea of what we want to do with this operator?

So what are the properties of this operator? The first is that it's LTI, linear and time-invariant So when I say it's linear, I mean the following-- that if I took a times x of t plus b times y of t and then I took the local average of this whole signal, what I would get is a times x bar of t plus b times y bar of t.

That's nice. It's also time-invariant, by which I mean if I took a signal x and shifted it by some time t0 and then I took the local average of that shifted signal, it's equal to x bar shifted. It's equal to x bar of t shifted by t0. So the local average of the shift is the shift of the local average. That make sense to everybody?

So now why am I worrying that it's a linear time-invariant operator? That means I can actually represent this thing with a transfer function. As an operator, it's basically a filter that I can put on my circuit.

So this is just a way of describing a filter. Here's what the filter transfer function looks like. Here's the operator definition and time domain. Here's what it looks like in the frequency domain. What we can see is that at low frequencies, near DC, it has a gain of 1. It passes through the low-frequency content.

But this particular filter just happens to have zero amplitudes at the switching frequency, twice the switching frequency, three times, et cetera. So it very precisely keeps the low-frequency stuff and murders all the switching ripple.

Now, could I pick some other filter that captures this idea? Sure. But we just happened to pick this one because it's a good one for our purpose, which is to keep the low-frequency content and get rid of the switching ripple and its harmonics.

You'll also notice that the phase of this thing is basically representing-- it's a linear phase response-- basically, a delay of half a cycle. Why? Because if I'm looking now, I'm really asking, what's my signal over the last capital T? So I'm, on average, looking back half a t in duration.

If I didn't like that, I could make my local average between t minus T over 2 and t plus T over 2, capital T over 2. And then I wouldn't have any delay in it. But I also wouldn't be causal.

Now, I'm not saying we're actually going to physically build a filter. But this is an operator that lets me think about how I'd process my signals so that I could keep the low-frequency content and get rid of the switching ripple. Any questions about that?

So we're going to see how to use this kind of filter to do interesting things. There's a few ways we can use it. And they give similar results. But they're useful. But let me just, lastly, talk about a couple of other properties of this operator. And the first that's a very useful property is the differentiation property. And this is one of the reasons why we picked this particular filter.

If I took some signal dx dt-- so some time-varying signal-- I looked at its derivative, and I said, give me the local average of the derivative of this signal, It's pretty easy to show that this is the same as the derivative of x bar of t. That is, the derivative of the local average is the local average of the derivative, or the local average of the derivative is the derivative of the local average.

Why is that true? Because basically, I'm taking the derivative of this thing. I can interchange the order of differentiation and integration. I get back the same thing. And this property turns out to be a very useful one for modeling, as we'll see in a minute.

So how might I use this average model or average operator? And the way I'd like to show you to use it today is on a circuit itself. So suppose I came up and I gave you a boost converter and I said, well, gee, could I apply that average model to the boost converter itself, to the circuit?

And let's think about what that means. If I thought about any circuit, what defines its topology? What defines its topology? What's connected to what? The structure of it is really the set of KVL and KCL equations that exist for that circuit.

So any circuit has some set of KVL equations, which essentially are the sum of Vij is equal to 0 around a bunch of loops. So there's a whole bunch of equations that have this form. And likewise, there's a set of KCL equations, the sum of i sub k's, that are equal to 0.

So I could take any circuit and write down a list of KVL and KCL equations that pertain to that circuit, one for every voltage loop, one for every node. And we can go back and say-- figure out which ones of those are independent, and so forth.

Well, what happens if I came up and said, let me take a KVL or KCL equation or, more generally, the ones that represent my circuit and average those things? Well, all I would do is if I took this and I averaged it-- when I say average, I mean apply this operator-- what would I do?

I'd take the local average of each side. The local average of 0 is 0. This is just the local average of the sum, which I already said, because of linearity, is the sum of the local averages. So this gives me, if I average it, the sum of Vij local average is equal to 0. And this gives me the sum of i sub k local average, which is 0.

So what I'm really saying here is if I have some circuit like my boost converter up there, I could apply averaging to that circuit or all the KVL and KCL equations. And I could get another circuit that's average that has the same connections in it.

Why? Because for every KVL equation in this circuit, there's an identical KVL equation in this circuit. And it just has averaged voltages. And likewise, every KCL in this circuit generates a new KCL circuit just with averaged currents. So I can get a new circuit with the same topology for which averages pertain.

Why does this work? This is the-- similar to what we were saying. Remember, we talked about, well, there's average KVL and average KCL over all time from periodic steady state. Well, I can also talk about it over a cycle. If KVL holds instantaneously, then it ought to hold for the local averages. And likewise, if KCL into a circuit node holds instantaneously, it ought to hold averaged over the last cycle. So it makes sense that you should be able to do this.

And the key point here is because these equations have the same form as these equations, you get the same exact circuit back, the same exact circuit topology back. What the elements in that circuit topology do is another question. We haven't said how are voltages and currents of the elements related. All we've said is that the circuit topology stays the same. Any questions about that?

Let's start talking about circuit elements. What do I have in this circuit? If I come back and look at my boost converter, I have inductors, capacitors, resistors, and then switches and sources and switches. Well, the local average of a source is just the local average of a source. So VIN would become VIN bar.

But what about a resistor? What about my load resistor? Well, if I took my load resistor, what I really have is V-- I have i sub R and V sub R. And what I get is V sub R is equal to i sub R times R.

Well, what happens if I take the local average of that guy? I'm just going to take the local average of this equation. So I'll come over here. And I'll average the left. And I'll average the right. And I can pull R out of the equation.

So then my average element ought to just be this. I have V sub R bar, i sub R bar, and R because my relationship is simply V sub R bar is equal to i sub R bar R, which is exactly the relation of a resistor.

So that just says if I had a resistor in my original circuit and I look at the local average of the constitutive relation of the two things connecting the nodes, if it's a resistor, it just stays being a resistor. That make sense?

Well, what happens if I have an inductor? I have V sub L, i sub L. And the relationship is V sub L is equal to L di sub L dt. Well, if I average this guy, I'm going to take the local average here and the local average here. This shouldn't have been a local average. And this would give me what?

Well, the relationship is V sub L bar. Now, I can pull the L out because of linearity. And the local average of the derivative becomes the derivative of the local average. I said that was my property over here. And what I get is then-- is equal to L d dt of iL bar, which is just an inductor of value L. Does that make sense?

I could do the same thing with a capacitor. So the beautiful thing here is if I go to average my circuit, the resistors, inductors, capacitors, sources just become the local averages of themselves. Well, that's nice because what I'm suggesting is I can come to this circuit and take its local average and come up with a new circuit that does the same thing, but without switching ripple. That make sense to everybody?

Let me see how I would do this here. But in order to do it, what I haven't said is what happens to the switches. And that's a little trickier to think about. To do that, let me make something that's a little bit circuit model-friendly.

So let me come up with a different model for my boost converter. My boost converter had an input voltage, an inductor, L of that with a current iL. And then it had a capacitor, C, and a resistor, R, with an output voltage VOUT on it.

And then it had the switch in the diode. And I could represent everything with the switch in the diode. But maybe it would be easier from a modeling perspective if I just came back and modeled the switch in diode with equivalent dependent sources. And here's how I'm going to argue-- and let me-- there's a couple of ways I could do this.

Maybe the switch-- I have to worry about what's the voltage, Vx, here and the current, ix, at this node. Maybe this is Vx and this is ix. Maybe I would model this as a dependent voltage source.

So what would this voltage source be? I could write this voltage source, dependent source, as being 1 minus q times VOUT because notice this voltage, Vx, pulsates. When the switch is on, when q is high, it's 0. And when switch is off, it's equal to VOUT. So in continuous conduction mode, this is a perfectly legitimate representation of what the voltage is doing there.

So I'm replacing the switch with something that generates the same voltage, Vx, as the switch does. I'm going to replace the current diode with a dependent current source that equals id. And that would be 1 minus q times iL. When the switch is on, this current is 0 because the diode is off. When the switch is off, this dependent current source is equal to iL.

So I've got one dependent voltage source and one dependent current source, which impose the same constraints as what's happening at this node as the original switch in diode. Any questions about that? Yeah, Laurie?

STUDENT: So [INAUDIBLE] is your choice [INAUDIBLE]?

DAVID PERREAULT: No. In fact, I could have modeled this with a dependent current source and that with a dependent voltage source. It would have been totally fine. It would have done the same thing.

There's a lot of ways I could do this. And in fact, I don't even have to do it. I could operate everything I was going to say with the switches. But really, what I'm interested in is, how does the switch in the diode control the waveforms in the circuit and cause currents to be fed to the capacitor and voltages to be fed to the inductor? And this is just a simpler way to get it. Does that make sense to everybody?

So this is still a switching circuit. In fact, if I simulated this circuit and I knew q of t from some-- I ran a controller and I had some q of t, I could simulate this. And it would generate exactly this switching waveform. If I told you q of t and then q of t is the constant duty ratio here and I simulated this circuit with that q of t, I would-- and I changed the load resistance, I would get exactly that response. So as long as I'm in continuous conduction mode, this model represents that circuit.

But I want to get rid of the switching. So let me just go average this circuit. I'm going to apply that operator to that circuit. And what do I know? Well, I know that if I apply the operator to the circuit, the topology of the new circuit doesn't change. Why? Because when I apply averaging, I get the same set of KVL and KCL equations. So the topology ought to be the same.

And what that means is I can take the same circuit. And I can just replace VIN by VIN bar, iL by iL bar, VOUT by VOUT bar. And L, C, and R all stay the same because I've told you what happens to the I-V characteristics of my linear time-invariant components.

The only thing I haven't said what happens is to these two components I've drawn in pink. And so what I really have to do is take the local average of the product of VOUT and the switching waveform and iL and the switching waveform. And that's ugly because now I have the local average of a product. And I've got to go think about that a little bit.

Now, let me tell you something about the local average of products. One property that this operator doesn't have, in general, is that typically, if I have x of t y of t and I take the local average of that, in general, it's not equal to x bar of t y bar of t. You can't just arbitrarily do that.

How can I prove it to you? Suppose I had two signals that were like this. Here's x of t. Suppose x of t is just a 50% duty ratio square wave like this. Here's 0.5t. Here's t. This is x of t.

What is x bar of t for that signal? And suppose this goes on forever and this is of height 1. It's just going to be 1/2. At any given time, if I slide a window over this and I take the average, I've always got it high half the time. So x bar of t is just equal to 0.5.

Well, what happens if I made its complement, y bar of t? I'm sorry, here's y of t. y of t is exactly the complement of x of t. So what I would argue is that y bar of t is also equal to 0.5. It's just-- happens to be out of phase.

Well, what's x of t y of t? I think I heard somebody say that it's 0. One of them is 0 all the time. And that's "zip." So if I took the local average of x bar-- if I took the local average of x times y, that would have to be equal to 0. But if I took x bar y bar, that would actually be equal to 0.25.

So that's proof that in general, I can't just take the local average of the product and expect it to be the product of the local averages. That just doesn't work, generally. And I care because I have this local average of a product over here.

But imagine the reason this failed was because when y was high, x was low, and vice versa. They were both doing crazy stuff. What happens if, perhaps, one of x of t and y of t look like a constant, roughly speaking?

And what do I mean by constant? I mean that it has two properties. If x or y has, one, a small ripple-- in other words, it's not varying full-scale within the cycle-- and, two, slow variation, meaning it's not one thing in one cycle and jumping to some massive thing in another cycle, massively different in another cycle, it's just-- the two waveforms just-- one of the two just-- it may move, but it only moves a little bit-- then I can think within any given interval, one of those two is constant and then like linearity. Then I can just take the product of the local averages.

So in this case, I can have xy bar is approximately equal to x bar y bar. So the only reason this failed is because they're both flapping up and down in counterpoint. If only one of them is flapping up and down and the other is constant, at least constant-- not changing that much in a given capital T interval, I could make that approximation. Does that make sense to everybody?

So this might be true of if x or y is an inductor current or a capacitor voltage. We know that even in that example I was showing you, yeah, there's some ripple. But it's on top of a big DC offset. It's not really violating that approximation very much.

So here, maybe I could say that 1 minus q-- the product of 1 minus q times VOUT is approximately equal to 1 minus q times VOUT bar and 1 minus q times iL is approximately equal to 1 minus q bar times iL bar. Let me define d as being equal to q bar of t.

In any given cycle, if I'm assuming my duty ratio is changing slowly, it's about equal to the local average of q of t because that's what my PWM modulator over there was doing. It was giving me a duty ratio.

Now, this is precisely only true when I'm looking at capital T. And I look back. And I get d for that cycle is exactly here if I did my local average right here. But if d of t is changing only slowly, that's a pretty good model to within half a cycle.

So then I could replace this. I could say 1 minus q is equal to 1 minus d times VOUT bar. And this is equal to 1 minus d times iL bar. And I can average this circuit. And what I'm going to get is this. VIN bar iL bar-- this becomes a source 1 minus d times VOUT bar. And this becomes 1 minus d times iL bar CR VOUT bar.

This circuit has no switching. d is a continuously varying variable that my controller might generate. There's no switching left in this circuit. It does vary in time. It's a dynamic circuit. It has an L and a C. And if d changes over time, the voltages and currents in this will change. But I've killed the switching ripple, essentially applying my local average operator to the circuit. That make sense to everybody?

Let's take a look at what happens if I do that. This is a simulation of this exact circuit for the same transient, except that I'm simulating this thing. It's in that little box that says "average model."

There is a diode there. That diode is just providing a constant drop because keep in mind my approximate models didn't really include constant diode drops or anything like that. So that diode there is not doing any switching. It's providing a constant DC offset. But this is the transient you get when you change the resistance in that circuit. Here's the voltage. And here's the current. It's right up the middle.

So basically, I've got a circuit that has second-order response. And it tracks the average values of my waveforms-- gotten rid of the switching. So this circuit could be-- it's not in a controller form. But it's a circuit that tracks this whole thing without any switching. Questions about that?

Now, we're going to take this up next class. But I wanted to show you the basis for doing this. I want to say this is really useful for building control models for converters because now I have something that-- I've gotten rid of all the complicated switching stuff. But it still gives me the low-frequency dynamics that I want to deal with my controller.

I will warn you that this system is nonlinear. Why? Because you notice d is my control variable here. And it multiplies the output voltage. And it multiplies the inductor current. So I have a product of my control input and my state variables. This is not a linear system from-- in terms of going from duty ratio to output. So I still have nonlinearity I have to deal with. And that's just life. We'll talk about how to deal with that.

It's very good for doing low-frequency controller modeling. It's not useful for other things. So for example, if I asked you to calculate power dissipation of components, you cannot use this model. Why? Because what would the power dissipation in the switch be? The power dissipation of the switch would be the switch voltage times the switch current, both of which are pulsing up and down together. It violates this assumption I was talking about.

In fact, the voltage and current on the switch actually look a lot like that. So I can use this average model to do things like control dynamics. That's perfect for it. I can't use it to answer questions like dissipation or other kinds of things.

So I am well out of time today. But we will take up the notion of control modeling next class. Have a great day.