DAVID HARDT: Good morning, everyone here in Cambridge, and happy holidays to you guys in Singapore. What's the holiday? I had it on my PDA, but I've forgotten what it is. Let's see if I can tell you. Labor Day, it's Labor Day in Singapore. OK?

Somebody was asking me when Labor Day is in the US, and you think of labor, of course, it means something different than this. But do you think of labor as something that's kind of onerous and something you don't always look forward to doing. And that's, for an academic at least, that's September, when you have to go back to work. So it's in September in the US.

OK. Excuse me. My voice is shot, so you may be lucky, and I may finish early. It's never happened before but might today.

What I've been asked to talk about today is this concept of using feedback to improve the statistical performance of a process, and it's got this title, Cycle to Cycle Control. Just a little caveat, this whole lecture and the material behind it and two research projects that came out of it were spawned about eight or nine years ago, when I was preparing a lecture for this class, pre-internet transmission, that kind of stuff. And I thought, gee, what would happen if we-- we used to do a lot of stuff on feedback control in the class but for a different reason. I said, how about if we applied it here and came up with this idea of cycle to cycle control.

One of the problems was, of course, I found out that it was old material. It was actually hidden in textbooks. It was treated in a different way. So this is, let's just say, a different spin on a somewhat well-known topic but usually under the heading of advanced material on the use of feedback control in the context of a statistical process control problem.

Now, I'm going to go through a couple of slides which were from day one of the class, and to emphasize, I think what we said back then is that, in effect, there's always feedback going on. Now, most of the stuff we've done in the class so far on SPC and designed experiments has been, in the classical feedback control sense, open loop. You do something, you record what happens, but there's no necessarily algorithmic approach to saying, OK, what if it's not right?

I don't what Professor Bohning said about SPC, but SPC doesn't necessarily tell you what actions to do to correct what's wrong. What it does is it tells you the process is not following the model, the statistical model that you had. And are designed experiments and process optimization is indeed falls into the larger context of open-loop optimization. How can I run a process in effect unattended open loop and still have it be robust to things trying to deter it from its proper course?

We talked at the beginning of the term about this idea of the general process control problem, and I'm sure you remember like it was yesterday that I referred to this process control loop is the Holy Grail. Boy, if we could do that, all this other stuff wouldn't matter. Because it would mean that I would have in some way the ability to interact with a process and say here's what I want.

Here's a dimension that I want. Maybe here's a variability that I can tolerate. Here's a property strength or a reactivity or something I want. If I simply measure it and apply correctly feedback control principles, compare the two, somehow with the magic control algorithm, everything will be fine.

And feedback control, if it's successful, always seeks to make the difference between those two close to zero and as quickly as possible. So the concept of process output control or controlling the product geometry and properties was put forward. And at the time, we probably said, but it doesn't work, or it can't happen, or it's very rare.

So what motivates this particular concept is the idea that this is indeed very hard to do, and it's very hard to do for two reasons. One is you seldom can find out what's wrong with the product, until it's too late. When you think about most of the processes-- there are a few exceptions to this-- but most processes, semiconductor is a great example, but every kind of mechanical and thermal process around, basically you have to make something, put it down, oh, we're done.

OK. Let's measure it. Did it fit? Is it the right shape? Does it have the right properties? And it's too late. So if it's wrong, what do you do? OK? So what do you do? If you run the process, you pick up the product, you measure it, it's not right. What do you do? It's not a rhetorical question. It's a question. What do I do?

AUDIENCE: Throw it away. Or redo your work.

DAVID HARDT: Throw it away. Rework it. And that's what we do now right. You got pass/fail, throw it away if it's a low value added. If it's a high value added thing, like an airplane or something like that, then you do rework it. Richard, what else?

AUDIENCE: Well, if the [INAUDIBLE] is big enough, or the production is big enough, you could try to adjust your process. For example, when you have a C and C turning process, you could try to make adjustments for the true position.

DAVID HARDT: Yep. Oh, yeah. So you can do all three of those things. And what we're going to talk about today is more towards what Richard was saying, which is you learn. You take that as data and say, oh, something's not right here. And the question now is we can always take that data-- and I made it sound black or white-- it's a good part or it's a bad part.

But indeed, as you know, is it-- from the concept of process capability, is it falling where I expected it to? Is it falling outside there or is it tending there-- to there? And how quickly do I take action? Now, SPC in general is about-- you, in effect, sit back, measure the statistical performance of the process and say, OK, is this matching my model? If it's not, pull the cord. Make some changes. Something's wrong. Investigate.

What we're going to talk about today is the extreme of that, which is to say every single part that comes out we will assume should be correct. But it will be off by some amount-- based on that error, on a production cycle to cycle basis. Every time we make something, we'll measure it. Every time we measure it, we'll compare it to the target value.

And if it's wrong, we'll make an adjustment. Now, I can guarantee you there are a number of papers-- many papers, probably thousands-- hundreds, I should say-- and many textbooks that will tell you, that's crazy talk. It's nuts-- should never do that, because you'll never you'll get it better.

Now, based on what you know so far about statistical processes and the nature of these techniques that we've done, as well as a little bit on physical processes, why would you imagine the statistics community in particular would have that attitude towards this approach of what we call constant adjustment? Can you think why that's not a good idea? Yeah?

AUDIENCE: Instead of some underlying median [INAUDIBLE] process, and you continuously keep changing the mean, then you're continuously changing the balance of the median. So unless it is statistically significant outside some bounds, I guess they would object to making changes based on something that's the normal expected means,

DAVID HARDT: OK. Yeah, yeah. Richard-- same thing? Yeah, that's basically right. I'm going to state it a little bit differently. Everything could be fine, and your process is following a nice normal distribution, and here's your target-- basically says you will always have an error. It will always be wrong. You'll always be adjusting the process.

It's the concept of a dog chasing its tail. How can that possibly make it better? So they just say, don't do it. And they're actually studies out there. There's a wonderful study that was done by a professor at Harvard Business School on-- I've forgotten what the-- do you know this one? The [INAUDIBLE]?

I think it was a big basic material, basic mineral processing plant, one of these huge things with lots of different set points. And the operators had control over this thing, and they were always looking at things. And they could type the new numbers and change the change the set points in the plant, and that sort of thing, and they all had their idea of how they could make it better.

And he had this hypothesis that they were chasing their tail. They were looking at these things, and they were always adjusting the process. So one day, with the concurrence of the management, from the operator's point of view, everything was still the same, but they disconnected it so that the operators were making changes to all the set points in the machines-- nothing was actually changing.

And guess what happened. It got better. The variance went down. And so that kind of circumstantial stuff says, OK, it is crazy talk. Why should I bother to do that? But-- there's an important but here-- is that, if that's all the information you have-- if your idea is, OK, based on the error, make an adjustment, if you have any schooling in feedback control, you know that just closing the loop is not success.

In fact, closing the loop, as I was once chastised by one of my professors-- all that guarantees is that you can take a nice stable process and eventually make it go unstable. But you have to close the loop with a knowledge of what's going on and a knowledge of what the limits are. And in the end, what that means is you've got-- if you're going to have an operator making adjustments to the process based on measurements, they should have some basis-- for a basis for that. So that's what this whole thing is about is, what is that basis, and how good could it be?

OK. OK, wake up now. All right. And just to tie this back in with what we talked about in the beginning, if you look at this variability equation-- which, I don't know how much you've used that during the term-- it's not important-- but the whole idea is that what we'd like to do is change the variability, reduce the variability of a process, and we'd like to do it-- SPC does it by reducing the disturbance, design experiments does it reducing the sensitivity, and feedback control does it by trying to offset the problems.

So if there is something that's causing a mean shift, you want to push it back by making an offset change, and that's all that this is relating to. And so in effect, we get to the third level of this process control hierarchy that-- talked about on day 1, which is-- the first one is basically good housekeeping, standard operating procedures, statistical analysis, and identification of whether everything is in a state of stationary statistical control.

Second one is reducing sensitivity or increasing robustness. That's the whole designed experiment area. And then finally-- this gets it as good as it can be without active manipulation-- and then finally, we actively manipulate and see what we do. So I do think I've come to firmly believe-- even though I came to this whole thing from a feedback control can solve any problem in the world point of view-- this is the last thing you do after you've tried these far cheaper, far more robust, and far more, in some ways, in depth knowledge of the process approaches.

OK, so I have to take a moment to just review feedback control. How many of you guys have had a feedback control course? OK. Hayden, you've never had a feedback control--

AUDIENCE: Yes.

DAVID HARDT: OK. If the TA hasn't had the course, we're in trouble. And we often use this generic feedback control regulator approach, and I want to do this because I'm going to use a little bit of terminology. This is stuff we've already used in the class, but I want to look at a little bit differently. So let me just define the terms here.

Whoops-- this box right here-- that's the process. That's the manufacturing process. That's the machine, the material, all that other stuff we talked about. And the input to that is something that I can change-- temperature, pressure, a feed, a time-- something like that. The output of the process is the output of the process. It's what we care about.

Actually, I'm going to put the output of the process over here for the moment. The key thing is that the process-- there are two things that make the process not always do the same thing. In other words, if I hold the inputs the same all the time, the output still varies. We know that. We know that from SPC. You know it from data that you've taken.

Where is that other thing coming from? Well, in a control system sense, one of the sources is the disturbance. We just kind of think of it as an external factor, this D1 or D2. The classic example of that is a flight control, altitude, autopilot. What's a good disturbance to an autopilot? Well, a wind current or something like that is external to the machine. It causes it to change and causes the output of the flight control system to change.

The purpose of the controller in that case, and the purpose of the controller primarily for manufacturing, is to minimize the effective these disturbances. You'd like to make it look like they aren't there so that you're flying the plane, and wind's blowing up and down, and it still keeps going level. You're machining something, and the material gets harder or softer. You just keep going the same way.

You're etching something, and maybe your solution is losing strength, but it still etches enough-- all those sorts of things. That's a standard way of modeling things in control, and that's the way we'll do it for this approach. In fact, what happens more often than not in manufacturing-- it's captured by this crazy little arrow here, the dark one, which basically says, yeah, well, this input-output relationship between the settings on the machine and the output of the material is, in fact, never a constant.

We're always putting in different material. The machine is changing over time. This is all the stuff we talked about in the first days. So in fact, we have two types of disturbances-- these external ones and these internal ones. The problem is that, from a control system theory point of view, the external ones are easy to model and analyze. The internal ones are very uneasy to model and analyze.

So we will actually look more at this one, even though we'll pretend that we're looking at both of these. So again, the purpose of the control system is to make this output quantity stay exactly in line with what the input quantity is, which, of course, would be some sort of a desired dimension-- something like that.

Again, if any of you have studied control systems in any advanced way, we have this-- one aspect of control systems is called stochastic control systems, which is where the control theory actually assumes that the variables that you're controlling and other things have some randomness associated with it. Most control is done, of course, for deterministic quantities, but in fact, most things have these random components.

And when you do that, you find that what you're taught on that is that, well, if I use feedback, I can minimize mean shifts in the process, or mean shifts in the disturbance. And I can often reduce so-called dynamic disturbances. If things are changing from time to time, I should be able to reduce those.

So if I translate that into the lexicon of this class, that kind of means-- well, look here. Here's a distribution. Let me-- I'm drawing too many lines here, hang on. I can't-- This distribution on the left-- well, first of all, let me point out what I tried to draw here. Here you see this is your process capability window. That's where you want to be. So ideally, we're centered on that, and well within the bounds of that.

Imagine you have a process that's operating over here. So it's obviously got terrible process capability. The mean is way over here, and even if I were to send it within the target window, it would probably be outside that, and I'd have a low process capability. So wouldn't it be nice to have a controller that can take this distribution, center it, and even make it narrower?

Well, there's a suggestion that feedback control can do that, because we know we can eliminate mean errors with feedback control. The question is, what happens with the variance? And as we thought about this idea of chasing your tail-- excuse me-- how can I be measuring a random variable, particularly an uncorrelated random variable, and use that knowledge to actually reduce the degree of randomness? Keep thinking about that.

But we can analyze it, and then, of course, the proof of the pudding will be to actually do an experiment. Oops-- OK, so what do we want to look at-- in trying to understand this, what kind of disturbances do we expect to see, or how do we want to model disturbances? Well, we know so far that, to capture the process, we're going to have to model disturbances as being random. Our whole premise for the statistical models is that the process has some randomness associated with it.

And then we have things like unexpected mean shifts, which, in control lexicon, we call infrequent stepwise. And then we might have things like tool wear or just degradation of solutions and things over time, and so there's a slowly varying change of the process. And then, of course, cyclic-- things that are happening because of shift changes, time of day, other things like that.

It would be nice to deal with all those. For the moment, we're going to deal primarily with infrequent stepwise random disturbances. There are more advanced topics on this that will deal with these too, but we won't get into those today. OK, so cycle-to-cycle control is simply defined as doing the best you can with this situation that we have now. So the ideal thing is to have a full measure of how you're doing on your production.

You can always make that measurement after the cycle of production has finished. Maybe an injection molding machine is a good model for this, or something that happens kind of quick, or a metal stamping operation or high-speed forging operation-- any of these things where you can't get in there with your gauge while it's happening, unless you're crazy. And even if you did, you wouldn't-- it wouldn't tell you what you wanted to know.

Like an injection molded part-- until it's popped out-- in fact, until it's cooled off, you don't know how you did. So it's finally there, you measure it, and then you say, oh, I have to make a change. Can I make a change? And most machines will allow you to make a change somewhere. You did your designed experiments, and the whole idea of that was to say, what are the significant inputs to change significantly-- have a significant change in the output?

So we find out what those significant ones are and try to match them. So if the dimension is wrong, you say, OK, here's what I want to adjust to change it. Now, the key thing is this-- or think of it this way. Take injection molding, because I'll give you some data on that in a moment. You set it up. You run a part. You take it onto the bench. You measure it.

You say, it's not right. How soon can I make a change in the process to improve it? Next cycle-- so before I say go again, I make a little change. I increase the hold time, I decrease the pressure-- something like that. Then I say, go. And I keep doing that, and it's cycle to cycle the cycle. But within each cycle, when I say change the whole time or I say change the pressure, during the actual cycle of injection-- of melding, and injecting, and cooling the material, nothing changes.

So from the injection molding machine process point of view, it's still this fixed input process. So that's the important thing. The dynamics occurs between the cycles, not during the cycles. So what this allows us to do is treat this as a very simple digital control system, or discrete control system.

So in discrete control theory concepts, things only happen in an instantaneous period in time. Nothing happens in between. So we've got a nice match between the theory and what's actually happening. And so we execute this loop once every cycle. Now, you may be saying, jeez, do we have to measure it after every cycle? What about every 10th cycle? Maybe it's going to take too much time to make the measurement and slow the process down, and so on.

Fine-- we can put those multipliers in. But the fastest we could ever do it would be one cycle-- one measurement per cycle. Well, this is where we found out that we had reinvented the wheel-- or uninvented the wheel, or something like that. Within the semiconductor industry, there's this concept of run-by-run control, where you've got a recipe, you run the process, you make a measurement, you say it's not right-- let's change the recipe-- that sort of thing. It's given the title run-by-run control.

And I think even, if you look in the text on Montgomery-- and I don't remember what chapter it's in-- there's something in there about-- forgot the term he used-- I think he calls it feedback control. But there is some discussion of this. And there are some larger theoretical tracks around that talk about the concept of a discrete control system like this.

So what is it we're really trying to do? Well, to use, again, the terminology of the class, we would like to reduce the expected loss function or increase a measure of process capability with feedback. So ultimately, as I said, if our goal is to go, say, from-- if we have a target that looks like that and we have a process that's over here somewhere-- oops-- it's not very [INAUDIBLE] looking.

If we can make the process look more like that, we will have increased our process capability or decreased our process-- our expected loss. And we can think of that, again, with our two major goals. One is to reduce mean shift to zero, if possible, and decrease the variance.

Now, this is a bit of a-- I can't think of the word I'm trying to come up with. Anyway, it's-- we're looking to the future of what's going to happen with the results here. What if I have a process that's over here, and I'm able to recenter it without reducing the variance? Is that a good thing? Yeah. OK.

What if I have a process over here, and all I can do is recenter it-- I can recenter it, but at the cost of increasing the variance? Is that a good thing? Well, that sort of depends. I didn't tell you how much I'm going to increase it. If I increase it by a factor of 10, you'd probably say it's no good. And what if I could move it over here, center the process, and reduce the variance at the same time? Of course-- why wouldn't that be great?

So we're going to think of those as objectives. Now, we have to go through this to understand the derivation. And the punch line that I'm going to come to is, hey, we can actually do two of those three. I'm going to show you that we can send her this-- guarantee it-- I love that-- seldom get to say that. But we can guarantee that we can it, virtually under any circumstance.

And we can even tell you how long it will take to center it, how many cycles-- can always do that. And then there'll be two extremes. In one extreme, I can see it, but I will always increase the variance-- but by a well-established, bounded amount. So it's not going to be one day 20 times, another day two times-- that sort of thing. It'll tell you exactly how much it is.

In the other case-- the better one, of course-- I can center it and reduce the variance. Now, the naysayers that we talked about earlier, who would say, oh, don't adjust the process-- it'll only get worse-- we're right, in the sense that, if I just the process and center the mean, but increase the variance, by one measure of goodness, things have gotten worse. It's got more variability.

And we can see, more heuristically than theoretically, why that, in fact, has to be true. But again, if it's bounded and we know what's going on, maybe it's better. Maybe it's a lot better. OK, so how do we actually get to use this? And I'm going to go through this real quickly, because again, if you guys have seen feedback control in its basics, you probably did it for continuous systems, and that doesn't really give you that much insight into what we're going to be doing here.

And there's some-- a couple derivations here, which I will just zip through and assume, if you're interested, you can back and look at the notes. Oh yeah, before I forget-- because I don't want to wait until the punch line at the end-- there are two papers that I've put on the website that basically recite what we're talking about-- what I'm talking about here-- paper that George Su and I authored several years back.

Well, George did his master's thesis on this whole topic. He was a 2830 student the year I kind of made up this lecture. Actually, it was a whole section of the class back then. And he said, hey, that would be interesting. Can I do a master's thesis on it? I said yes, and he did, and that's-- a lot of the results I'm going to talk about today are on that.

And then we had a particular problem in our-- oh, great-- I just erased it. Well, there's a second paper that's on there that Adam [INAUDIBLE] did with me based on his PhD. And Adam was also a student in the class, also a TA in the class, and he wanted to say, well, what happens if I have a really big complicated process? Instead of having one input and one output, maybe it had 1,000 inputs and 1,000 outputs-- because we happened to have one in our lab that fit that.

And he said, maybe we can make it work for this. So he did the extension to that. And that paper is very brief version of what happens when you try to extend it beyond what I've done. But it all requires that we go through a little bit of the theory.

So what am I trying to represent with this diagram? I'll give you guys a test of your control background. What is this? Richard--

AUDIENCE: Step response?

DAVID HARDT: Yeah. This is a classical, highly simplified example of a-- the simplest dynamic system you can have. And in the lexicon of our class here, I've made a step change to an input. I instantaneously change the pressure, temperature, feed-- something like that.

And what's the output do? Well, in real systems, they don't change instantaneously. They have dynamics. And the simplest dynamics would be something like this, a simple lag. This would be a first-order linear system, and it takes four time constants to settle out. And so there's an important number here, which is what I'll call the process time constant. How long does it take the process to reach a steady state after you've made a change to it? That's going to become an important number for us.

And then we have the disturbance, which would be our randomized and that sort of thing. But why do I care about this steady state number? Why do I put this number four times the time constant of there. Why do you think-- given where we're going? We're only going to look at the process after it's done. We're only going to make a change between the processes.

I'm only going to look at the output. Think of it, we're going to talk about a sampler in just a second. Sampler is a mathematical concept, in effect, based on a delta function that says, here's a continuous number. I'm going to sample it and look at it for an infinitesimally small period of time. So I'm going to do this to the output. So I'm going to make a change, and then I'm going to sample the output. When do I want to sample the output?

AUDIENCE: After it's stabilized.

DAVID HARDT: After it's stabilized-- all right, so conceptually, if I have a process that has this speed to it, and I make sure I always sample it after it's reached this steady state-- after a stepwise change-- what does the process look like, from an input-output point of view?

Constant input gives me a constant output. I don't see any dynamics. I don't see any of this stuff in between. And can you see that then, if I looked at a lot of different inputs-- let the output all go to steady state, looked at all the sample ones-- I basically have a bunch of inputs, a bunch of outputs. And what would be the simplest relationship between those two?

You just did your designed experiments. What's the simplest relationship that you could have between a single input, a single output? Yeah-- straight line, linear fit. And that's a great first place for us to start with this model. So the nice thing is, with the cycle-to-cycle control, I can take a process that could be, in fact, a lot more complicated by this-- than this, and just say, sorry. I can reduce it down to-- I hope it's my next slide. I hope this thing will work.

I'm going to skip through all this. This diagram says that what's, in fact, maybe happening with the processes that I make a change at a particular time. It comes up to steady state, but I sample it here, here, here, and here. And all that mess in between I'm unaware of, I don't care about.

And what I can do now is reduce the whole process down to a single number. The nice thing is-- I don't need to go into this right now-- I don't even need to worry about the offset. You think of a straight line fit-- you've got the slope and the offset. In the approach that we take here with cycle to cycle, I don't even care about the offset. In run-by-run they do, but it turns out it comes out in the wash anyway.

So now all I need to know about the process is an input-output relationship between-- but it's discrete. It's between a discrete input at and instant in time and a discrete output. So that is a valid model of our process now. The only thing that's questionable is, what's this number, and is it a constant? In other words, if I run the process today, is it one number? Is it another number?

And of course, as you know, we would do a test for linearity, or maybe it should be-- maybe it would be better modeled as quadratic and other stuff like that. Without going into it, and again, invoking the linearization properties of a good control system, it doesn't matter. As long as we have a pretty good estimate of it and as long as it's not rapidly changing, we're OK.

OK. So here's, ultimately, our model. We have the process. We have some process uncertainty, and we're going to bring that in as a purely additive number-- because the model I just showed you was a deterministic model. There was no randomness in it, so it obviously doesn't capture what we care about. And I'm going to sample that output compare it to what I want, and that'll be my cycle-to-cycle controller.

Now, when you get into discrete control systems, one of the things that you can easily model is the idea of a pure delay. Obviously, if I got something that happens a given cycle, and nothing-- I don't do anything until the next cycle of time happens, I can have the concept of a pure delay as well-- something happens here, but there's no reaction until here, or here, or here.

And the theory of discrete control systems allows you to deal with delays very nicely. And we have three kinds of delays in any control system, but certainly happened here. One is measurement delay. Part comes off-- takes me some time to measure it. Because of that measurement delay, there's no way that I can make an adjustment before the next cycle. So I already have a delay based on that.

There's controller delay. Well, of course, if we're using fast computers and that sort of thing, that can be made very small, but it does take some time. And if it's an operator, a human being measuring it and then doing a little bit of pencil work on a path, there's a delay associated with that. And then there's a process delay. When can I make the machine? When is the machine next available to be changed?

So given all three of these, what do you think-- for this model, what do you think the correct model of delay is for this approach to process control? Now, let me preface this with one other comment. We have a basic unit of time here now? Other than zero, what's the shortest time interval that's possible that exists in this approach to the whole feedback control thing? The fastest I can do something--

AUDIENCE: One cycle--

DAVID HARDT: One cycle-- yeah-- one cycle, right? Because nothing exists between cycles. When something happens, the next chance I can do something on is a cycle. So this is all kind of time-independent. In terms of real time, it's just based on the cycle time of the process.

So given this, go back to that other question. There are delays involved in all discrete control system. There are delays in this control system. How do you think we should model the delays of a manufacturing process for this context of cycle-to-cycle control?

I'll give you the thought experiment. Run the injection molding machine. Put the part on the bench. Measure the part. Figure out the next control cycle based on the algorithm I'm going to come up with. Adjust the machine. Push the button. Run the machine again. That's the delay structure. How would you model that delay? Or how many delays or how many machine cycles do I need to delay the process, minimum? One? 1/2? 1/10? 10? I know I'm not making myself clear here. I can see that. Richard--

AUDIENCE: Well, the discrete points of time, where you're looking at your process that you're actually modeling, I probably arrived when your part comes out, kind of. So you should probably have a delay of one cycle.

DAVID HARDT: Exactly, exactly-- if I can measure quickly, make a decision quickly, and be ready to adjust the machine before it's time to push the button, then I have built in a one-cycle delay. The best I could ever do is-- and this is how you capture in the theory, at least. The whole idea of cycle-to-cycle control is I make a measurement. The next chance I have to intervene on the process is one cycle later. So you build in a one-cycle delay.

OK, why does this make a big difference? Well, in looking at discrete time control theory, the whole concept of a delay operator is the whole basis for it. Now, if you dig into your deep dark pasts on the control system theory, you may have remembered something called the Laplace transform, Laplace operator, and all that other business, which was a way of algebraically describing time differential equations.

What you have with a discrete control system is time difference equations, things are just sums of inputs and outputs at different intervals of time. There's an equivalent to that operator, which is called the z transform. And you're not going to do z transforms here, much as I'd love to, but basically it's an operator that says, if I-- I transform everything into this z domain, and now the concept of an advance in time by one time step can be done by having-- by pre-multiplying by this operator z with an exponent that's a number of steps ahead you want to do.

If I want to have a delay, then I just put a negative exponent up here. And so for example, if I think of this concept of-- here's the terminology is, here's the output y at instant of time i. There's no continuous variable t. It's the sample i, if you will. If I multiply that number by z to the plus 1, that's the same as saying, oh, you mean y one time step ahead? And if I multiply it by z squared, it's two time steps ahead.

And more relevant to what we're doing here, if I say, OK, here's an output at the current time i, if I pre-multiply it by this operator z to the minus 1, I'm really saying let's look at it one time step behind. So when we look at our process model, we have to modify it a little bit we say, OK, we can do this anywhere in the loop we want, but we decided to do it here because it's the easiest.

When I make a change to the input, conceptually, I don't get to see what happens to that until the cycle is over and I'm ready to do the next cycle. So we're going to model the whole thing not as a simple gain, but as-- it's got messed up in this-- but as this. Or in other words, we're going to say that the current output is equal to the prior input times that gain.

And this may seem pretty trivial to you, but this is the basis of the whole thing. It's very simple from a discrete control system point of view, but the mere fact that you say, yeah, the input leads to the output-- but a time step later is what makes the whole thing-- it's the part that makes it not work, if you don't know what you're doing, and it's the part you have to analyze to be sure what is going on.

OK, so for the moment, we've modeled the process. It's a pure gain with a one time step one-cycle delay. Here's some data from a process that's-- I think this is probably under a state of statistical control. And you know that these lines in between here-- this is done on Excel-- the lines in between there don't exist. It's dots here.

Here's a measurement, there's a measurement-- so on. They're just there to let you see it better. So that's the output of our real process. Well, how does the model I just came up with reflect this in any way? Of course, it doesn't. I haven't included the randomness at all. So what we have to do is add in some randomness, then do our sampling, and then see if we get something that looks like this.

So again, we have to come up with a model that fits what we observe in real life-- what I just showed-- you and also is amenable to our control theory. So again, as I said, here's-- we're still continuous here now. This is before we've done the sampling. We've got this continuous random variable, d of t. It's always there. It's always changing. Then we sample at our-- once every cycle, and we get our discrete variable. Now it should include a random component.

That's not exactly what it should say. Well, that's jumping ahead, but let's go ahead and do that. So now, in our cycle-to-cycle control system, we're going to add into it basically a sequence of numbers that follows a stationary normal distribution function. Well, that's no big deal. That's what you've been doing all along the term, so that's-- that shouldn't be a problem.

This is just a little bit of a connection between continuous theory and the discrete theory. A continuous or random number is, in fact, that. If you go back to our early stuff on statistics, this is really where we started from. But there's this number that's changing continuously, and it's never the same number at any time, and it's following a Gaussian distribution, and it's stationary-- or a normal distribution, I should say.

At any instant in time, there is no correlation, meaning, if I sample this distribution right now and I get a particular number, that gives absolutely no information about what's going to happen next. It does allow me eventually, with enough data, to say what the distribution is, but it tells me nothing about what's going to happen next. So it's totally uncorrelated. And if we sample that process, then we get what we've used throughout the term, is this normal identically distributed independent random sequence. Had-- do any of you use NIDI? You talked about the acronym?

AUDIENCE: We've used IIND, and--

DAVID HARDT: OK, all the other--

AUDIENCE: Other permutations--

DAVID HARDT: OK. Normal identically distributed independence-- so it means a stationary random variable with no correlation from time to time. All right, here's the algebra. And it's the key to the whole thing, but I won't go through it, except to say that, if you remember your block diagram algebra, there's a clear algebra that happens here.

And we'll just take the first step here. The output y sub i-- what this diagram means is that the output y sub i going this way is equal to-- that's a summation there-- I should have put out there-- it's equal to d sub i plus the output of this box. And the output of this box is kp times u sub i delayed by one time step. So that's all of that equation means right there.

Then we solve backwards here. We say, OK, well, there's this value right here-- the input to that happens to be equal to the controller gain times the difference of the reference and our feedback. You put it all together, and this is the equation for this-- the output of the controller.

And then here's the important thing. If I now say, all right, I'm going to make the input my command-- think of it as a dimension or something for what we're doing here-- a constant value-- 10 inches-- never going to change. And you can think of it as a constant. Oh, I'm sorry. Back up. I'm going to do that too.

For the moment, let's say that the disturbance is not random-- it's just a constant value. I'm going to use the term mu, but it's a constant value. If I do that, and put it into this equation, and solve for y sub i, and I let time go forever, so it's now steady state, if you go through the algebra, what you get is that the steady state value of the output is equal to these two quantities right here.

It's the disturbance divided by this quantity, and it's the reference multiplied by that quantity. OK-- big deal. But here's the big deal. Here's the output value in steady state. Here's the disturbance. What do I want that quantity to be, if this is a good control system? And for the moment, imagine that my reference is zero. Conceptually, that's an important thing. Imagine that my reference is zero, so the ideal output of the process is zero.

Now this says that the ratio of the output in steady state to the disturbance, which is finite-- some constant value-- is this quantity. What would I like this quantity to be? How about zero? Disturbances are bad. If this number is anything but zero, then I am not rejecting my disturbance. Let's jump ahead to say, if I can't make this zero, I can't center this distribution, because this shift is indeed that-- its mean shift.

How can I make this zero, Can't-- can I? How do I make this quantity zero? There it is right there. How do I make that quantity zero? I could do that. If I can make it an infinite gain, it would be zero-- tough thing to do. So kind of saying, it would be easy, we're not going to be able to do it. And this means that I won't be able to do this, and the naysayers were right, and all that sort of thing.

But fortunately, you'll see in a moment, there is a way we can do it. And maybe you saw this in your control system things as well. Conceptually, all I need is this game to be equal infinity when things are in steady state. That's one way to think of it.

The other way to think of it is, if you did your control systems theory, it said, steady state area equals zero-- that means I want an integral controller. And there's theoretical reasons for that. So what we can do, instead of using a simple controller-- I should back up here. What I did in here is I picked our controller, was simply to say that the controller was, if there's an error, make a purely proportional change in the set points.

Now I'm going to say let's get a little bit more sophisticated. Let's do an integral controller. And an integral controller in discrete terms is simply a running sum of the errors. If you remember back to Calculus 101, calculus-- typically, when you drive the integration, you start off with a summation, and then make the summation interval smaller and smaller until it becomes continuous.

Now we break it back the other way and say, OK, the best we can do for a discrete time integrator is to sum up errors. So this is my new controller. It simply says the output at any time, how I'm going to change the machine is the sum of all the prior errors. Again, I can turn this into recursive form. So you just say the next output is the prior output plus the next error.

And then I can do the z transform business on it. It comes out like this. And this is where it gets useful. Now my controller, instead of being a simple gain, looks like this-- the z over z minus 1, which is describing this equation. And I can do the algebra of this and see what happens. So I follow the same block diagram algebra, and the nice thing now is that z is simply a multiplier, so I can treat it like any other quantity.

If I do that and I go through all this-- so these z's cancel out. I write the equation for the disturbance. I'm assuming that r is equal to zero, so this is zero over here. Go through the block diagram algebra, get down to here, and now a couple of things are kind of interesting. But what I can say is that steady state-- we know that all of the y's in the system are the same. Everything's settled out.

And I come down here and say that, because this disturbance is a constant, these two things are the same. The disturbance at time 20,000 and time 20,000 plus 1 are the same, because it's a constant. And when you come down here, you find out that this quantity's equal to zero, regardless of the loop gain. In other words, it doesn't matter what value I pick for kc. It's always zero.

So it's just a way of showing that, with a simple integral controller, I indeed can take a finite constant disturbance and completely eliminate it in steady state-- not instantaneously, but in steady state. So now, here's the problem. Since kc doesn't matter-- that's my controller gain. That's my design. Since that doesn't matter-- so must matter to something, right? So what value do I pick for that gain of kc?

What's the effective the value of this controller gain on the performance of this system? Do you guys remember from control theory at all what the effect of a controller gain is, typically? Low gain, high gain, little gain, too high gain, too low gain-- generally, what happens as you adjust the controller gain?

AUDIENCE: Speed of the system?

DAVID HARDT: Yeah. Think of it this way. Forget about this for a moment. This controller game says it's how much I'm going to adjust the system based on the error. If I'm very conservative, and kc is really small, and I get, let's say, an error-- a modest error, I'm going to make a really modest, timid change. It's a measure of timidity, I guess. I'm going to take a timid change on the-- over here.

If I'm bold and trusting my control system-- small error here, large gain-- I make a huge change. And obviously, a huge change is going to get the system moving faster. So conceptually, heuristically-- however you want to call it-- high gains will generally lead to faster system response. Low gains will lead to slower system response.

Which ones do you think typically-- but not always-- which ones typically, do you think, are the most dangerous, with respect to things like, as I mentioned earlier, the property of control systems known as instability-- high gains or low gains?

High gains, of course-- yeah-- because imagine, tiny error here. You're almost there-- huge gain. You whack the system with a big thing, it goes way past where you wanted it to, and now you say, oh my gosh, now it's a large area. And it keeps coming back and forth, and next thing you know, you've got a system that's unstable.

OK, so to understand this, you have to understand discrete time control theory. There's some wonderful, books, and courses and other things on that. We don't have time for that here. We'll do it heuristically. We'll do it just for this system. Here's what happens.

I'm going to step through different gains. Now, I already knew the theory behind this. I knew what these numbers needed to be, but here's what you get. I got to say one of the thing. What I'm going to show you is the product of these two gains. That's what really matters, because again, these-- you can collapse these blocks together, and it's just this product, kc kp, that matters-- the so-called loop gain.

If I use a loop gain of 1/2, here's what happens. If I make a step change in the input, it goes 1, 2, 3, 4, 5, 6, 7-- maybe 8 cycles later it has reached the final value, and the error is zero, just as I expected. So OK, that's good. What happens if I increase the gain? Should get better-- well, indeed, it does. Look at this. If I increase the game to 9/10-- 1, 2, maybe 3 cycles-- it's done.

And you notice, this is really emphasizing the discrete control aspect of it. There's a change, and then who knows, and then another change, and then who knows? So 9/10 is good. 9/10 is good. Then 1.5/10 should be good. 15/10 should be good. So I go to 15/10, and what happens?

Well, it now takes two things. It took three cycles to settle here. Now we're back out to taking eight cycles to settle, and I got this oscillation. So it seems like I kind of went through-- I was getting better, and now I'm getting worse. Just to see what happens, since this is a simulation anyway, let's set the loop gain equal to 2. And what happens? The system indeed is now what we call marginally stable. And I don't have it on here, but if I set it to 2.001, it starts to diverge.

Here's the cool thing. If I set that equal to 1, how long does it take to get to the correct value? One time step-- and this is actually a pretty well-known concept in discrete time control systems-- that, if you can magically find exactly the right gain so that the loop gain is equal to 1, you indeed can get exactly there in one time step. So this is our goal.

And it says that the best control system I could have now would take a mean offset like this and take it to zero error in one time step. So this is probably the right controller for us to use. It also tells us that, if I'm too aggressive and I use a gain-- a loop gain of 2, then think of the operators now at the cement plant that I was talking about. And they were turning the knob a little bit too far, and so they were maybe here or here, and making things worse.

Or they were here and they were being too timid, and it really didn't have any effect over the time frame that they were talking about. If you knew what your process gains were and you knew where to set this, you can always guarantee that you got the maximum effect in the minimum amount of time. Also, the other thing is the operators were not using this control algorithm. They were using that control algorithm, so it would never get to zero error.

So with a fairly large gap in coverage here, what I'm purporting to show you is that, if I have a random-- if I have an output-- a process whose output looks like this, and I apply this simple discrete time controller-- which means I measure, adjust, but adjust according to this algorithm here, which basically is-- this is something I think it's important-- which basically says, OK, I'm the operator. I have to make an adjustment to the process.

Here's the adjustment I'm going to make. And the adjustment I'm going to make is whatever it was the last time plus this controller gain-- which, you're going to tell me what it is-- times the current error-- little bit different than what it might have been otherwise. You can also notice that, if I bring this over to the other side, what it's really telling the operator is the increment in the adjustment should be proportional to the air, not the absolute value of it.

But anyway, you implement that controller. And what that says is that, if I have a random output on my process stationary-- in a state of stationary statistical control, but offset from here, it tells me that at least the mean value here before will now be the mean value after. But the question is, will the distribution look like that or will it look like that?

So we have to look at the random component. This gets a lot harder to do so, I'll just kind of pop through it-- and again, more heuristically than mathematically. Our random model of this thing is that we have this normally independent-- identically distributed independent sequence for d sub i. So it's a bunch of numbers that are coming out, and they're sampled from a normal distribution.

If they are totally independent, what that means is that these-- there's no way that I can predict what the next value of the disturbance will be-- this d sub i. If there were, then I could write an equation that says, OK, there's some relation between the current value and all the prior values. You often see these so-called time series equations, or discrete time equations, or that sort of thing.

But since there isn't, then it says, sorry. It's just not equal to any function of any of the prior values. There's no way to predict what's going to happen next. So if I do that, then you can-- I think you can see then that knowing this doesn't tell me-- doesn't give me a good idea of how to adjust it to take care of what the next value is going to be.

So there's no way that knowing the prior value of the disturbance will allow me to write an equation to exactly cancel what the next one will be. So again, what I'm saying is this. Let's imagine here that I'm at time interval sub i, and I have a disturbance t sub i. And I measure that, and I say, ah! The disturbance at this instant is the value of d sub i.

What I need to do then is come up with a-- if you will, a delta u that's equal to minus d sub i for the next cycle. But of course, that's too late. I can't cancel out the disturbance that's already happened. So I need to know what d sub i plus 1 is. And then I could write this equation, and I'd cancel it out. But I can't, because I have no idea what the next one will be.

But with the controller that I've written, there's no way to separate out these two functions, so I'm going to do this anyway. I'm going to do this, knowing full well that it's not right, because it's the only way I can implement the controller. And so heuristically, it can't be good right, so it must be bad. Oops-- if you go through the theory on this, and even do simulations and things like that, indeed, you find out it's bad.

And here's the measure of badness. Here's a variance ratio. This is basically saying I've taken a random number with a distribution and with a certain variance-- and that's sigma squared-- this is before and after. Let me write that down here. This is before and this is after.

This ratio is how much the control system has increased or decreased the variance of the output. And this is a function of this loop gain that we talked about, this quantity we'd like to have at 1. OK, so what does this mean? Well, if I don't do any control-- loop gain is zero-- then I haven't increased the variance at all, and so it's at 1. That ratio's at 1.

Let's go out to our favorite number, which was 1, because it gives the best response. And it means I've increased the variance ratio to about 2.2. Or the standard deviation ratio, I think, comes out to about 1.5. So I've taken this width, and I've increased it by 50%-- nothing I can do about, nothing I can do about it.

I can run a slower process. I can say, OK, I'll take some time steps to reach steady state and maybe bring it down to 1.3, but-- so this is, in fact, a really good description of what I was telling you before. With this control system, with this type of noise, I can take a process that looks like this, I can guarantee the [INAUDIBLE] but I'm going to increase the variance by some factor, depending on what I set the gain at-- probably 50%.

Is that good or is that bad? Depends-- getting rid of mean shifts is a pretty important thing. If you think about the philosophy of Six Sigma, this is actually pretty good, because Six Sigma-- one starts with narrow distributions, allows for mean offsets. And if we say, OK, but we'll make sure you don't get mean offsets, and you already have a narrow distribution, we'll only make it 50% wider. So I don't want to sell it too much, but it's half good, it's half bad. Richard--

AUDIENCE: But if you can adjust the loop gain at 0.2, we would still arrive at our process mean, right?

DAVID HARDT: Yep. Yeah, but I'll show you an example of why that might not be a good idea, when we get to the experiments. OK, so the conclusion-- cycle-to-cycle control for an uncorrelated random disturbance can do what I said. It takes this distribution, shifts it, guarantees that it's going to be at zero mean, but it widens the distribution.

Obviously, I've drawn this so that it still looks fine and the-- and if I looked at this as cpk, or expected quality loss, the way I've drawn it, this is a winner by a lot. This has terrible process capability, and now this is really pretty good, even though it got wider-- because of the mean shift.

However, not all processes are the same, and there's a big difference in a lot of these. And so what if the process is not well described by a normally identically distributed independent random variable? Don't worry about what that says right there.

It's sometimes called the concept of history, or process history, or memory, or something like that. I think it would work well with a semiconductor process, but I know it works well with injection molding, because I'll show you in a second. Imagine I have an injection molding process, and I decide to make these changes by changing the hold time. Every time I get an error, I change the hold time.

So every time I cycle this thing, which is pretty rapid, I change the hold time. Now, the hold time determines, among other things, the temperature at which you open the mold-- how cool it is when it opens up, when you release the pressure. OK, how long do you think it takes that mold to reach equilibrium after I change the hold time? You think it reaches equilibrium after the first cycle?

Big piece of metal-- I say, hold it an extra 10 seconds so it starts to cool off. It's probably going to take several of those cycles before it cools off to its new equilibrium temperature. Does that makes sense? And then, if I say, let's open it quicker all of a sudden-- I open it quicker-- it's going to start to heat up-- it's going to take a while to heat up there.

So in fact, the process is not at steady state. At least the disturbances are not at steady state. There's a little bit of history. And we can represent this mathematically by taking a purely random independent sequence and put it through a first-order process, and you end up with an equation that shows, in fact, the next disturbance is somewhat predicted by the prior ones. There's a little bit of memory between the two, and this is where we get the concept of some process correlation.

So what happens if there's process correlation? Well, this is a lot harder to do theoretically, so we did it in simulation. I think we also have a theoretical one on this, but now look at-- here's the variance ratio, same thing as before, but now with a disturbance that has a little bit of correlation. Watch what happens. As I increase the gain from zero up to 1, what happens to the variance ratio?

It actually goes down for a while. I'm not sure these numbers are right. This one's not right. That one's not right. But anyway, it's going down. Actually, I'm sorry. That can be right. There's another factor in here. So why would the variance ratio go down? Well, it's because I now actually do get some information about what the next disturbance is going to be based on the prior one. That's essentially what correlation means.

And it's kind of cool, because it can't be much more specific than that, but it basically says, I can-- I'll show you this in a moment, but I can actually use this controller as a way of measuring correlation in the process. If the variance ratio goes down, there had to have been some correlation.

Here's the curve that I was thinking of. So you can think of it instead as sort of a continuum of correlation. Here's zero correlation, and here's a large correlation, if you will. So zero correlation is the first case we had-- independent random variable. And here's that same curve that we had before, where, at 1, it was still up there.

As I increase the amount of correlation-- in other words, as the words as the memory gets stronger and stronger-- this curve starts to come down here, where at very large values, you can see that at 1, I could have a variance ratio as low as 1/3, which means that, depending on the physics of the process, I could take a broad shift in distribution and make it narrow and centered with this approach. And this is with exactly the same controller, nothing different.

So the cool thing is that we end up with these-- it's actually a continuum, but we can say that, for a totally uncorrelated process, here's what's going to happen-- mean shift and variance increase. If we start to have some correlation in the process, and if there's some memory, some reason that the disturbance is occurring, we can start to actually get both.

So who believes this? It's just simulation. It's just math. Who believes it? Come on. Abstention's not allowed. Believers?

AUDIENCE: I don't believe in bending an injection molding.

[LAUGHS]

DAVID HARDT: Believers, agnostics? OK. Obviously, it must work, or I wouldn't say this. So George Su, a good part of his thesis was to say that the theory wasn't that-- in the end, if you'd already had the background, the theory isn't that extensive. And it does seem too good to be true, so we said, let's do an experiment, and let's try to find two extremes.

Let's find something that we think has the least amount of correlation and something that has a reasonable correlation. Let's also find something that's quick and easy to do in the lab that we have. And so we decided not to do semiconductors. We decided to do bending an injection molding.

And so George set up this experiment, which, in fact, you guys in Singapore, when you view this, will recognize, because you've done this in your summer experiments. It's an experiment we used to do in this class. You take a simple strip of sheet metal and you bend it. The input is the depth of the punch. The output is the angle-- pretty simple.

And we're going to do injection molding. This is in George's thesis. And here's the process model for bending-- pretty simple, right? The input is the displacement of that punch. The output is the angle. George did a number of tests. He put in different [INAUDIBLE]. He measured different angles-- shouldn't be the straight, but he got some good results. And so he came up with this number for a particular value of steel.

Now, keep in mind, this number will change immediately if I put different thickness material in or a different composition material, because it depends yield strengths, and stiffness, and that sort of thing. But he got a nice number for this. So all he needed from that was the slope. Now he had that slope. He could tune the loop gain, knowing that slope to be somewhere around where we wanted it to be.

So he implemented that. And this data's a little bit weird, because we hadn't, at this point, done all the theory and realized that there was only one true controller. So none of these are actually for the best controller case. So here's where a controller that is a proportional controller, no integral control, and has a gain of 1/7.

And what we want to look at was the variance ratio only, and then see-- indeed, what we got is-- here's the data beforehand. Here's the data after-hand. And our variance did go up, and it went up roughly equal to what we expected it to see. That's not one of the good ones.

Yeah, so here was one of the cooler things that we did. This Is, now where we did integral control. And we put in a disturbance, but the disturbance was not a mean shift. What we did is we took material of exactly the same thickness-- I think that's what we did here. Yeah, we did material exactly the same thickness, but a different material. I think we put it in aluminum, instead of steel.

So now the gain is wrong. So right here, everything's fine. The target is 35 degrees. And George slips in a different material. So immediately, the angle goes right down, because he's using the setting from the prior cycle. So now we've got this huge error. But what happens? By following the algorithm, in one, two, three, four time steps, its back and right on the money where it should be.

Now, he was using a game that was too low, so this took a long time to come back up. And the variance ratio was a little bit lower than we would have expected, but this is another value for this controller that I wanted to point out to, and why using higher gains is important-- if something like that came along, different material or significant changes in the material. This automatically, with no intervention on anybody's part-- not even pulling the and on cord, or having to do a statistically significant number of measurements-- just get you back on track. And it's a classic example of the strength of disturbance response for a controller.

Now, a compilation of everything that George did over all the experiments with the integral controllers is he just-- for the bending, he said, let's take a look at this quantity v, which was the expected quality loss. And he plotted it as a function of the gain, and indeed, it goes through a minimum right at the gain of 1, and then goes back up. So that was a pretty good result. And it said that, indeed, from no gain at all where the variance is very high and the offsets could be very high, we did a lot better.

AUDIENCE: Excuse me, was that curve from experiments?

DAVID HARDT: Yes, yes. Wait a second. Actually, no, you're right. I'm sorry. This was calculated. This was not from experiments. Yeah. But it suggested, again, that kc should be 1. Yes, I'm sorry. So then he did this experiment. He increased the gain to 1 for the material shift, and-- well, this is a different material shift too. This was a change in thickness.

And, again change in thickness-- it immediately goes down, and just in two time steps, it's back up to zero-- so pretty cool. So it confirms that, indeed, we can go from here to here in a predictable amount of time. The higher the gain, the faster it gets there. But what this doesn't show, but which we did have with other data-- it said, yeah, it does increase the variance about the amount we expected.

OK, so now let's go to injection molding. Well, that got more complicated right away, because the first thing you find out with injection molding is you're not sure which variables matter. You're not sure what the process gain is, so we had to do a full DOE on the thing and do a significance, test and find out what was important. And what we ended up with, through the end of analysis and some other stuff was, I think-- yeah, that hold time-- for his particular problem, hold time was the most important.

He was molding rings, simple test rings. And so we came up with a model that looked like this. Again, that term didn't matter, but it was a constant times x2, which was the hold time. This is reduced one. He started with a lot more variables to start with.

So here's proportional control-- again, one that we later abandoned. But it gives you a nice example of what-- of a bunch of things. Here's open loop, and he's making these parts and noticing they're all over the place, and they're kind of drifting. They're not even steady. And then he turns on the controller with a closed loop gain of 1/2, and the mean value is not correct, but it's constant. So it's completely eliminating these shifts that were going on, and the effect is quite dramatic.

Oh, and the other thing that's really cool is that the variance ratio was like 1/4. So you can see here that this is kind of all over the place, and now it's greatly reduced. You can actually measure the correlation of these two processes by just doing-- running it through a correlation filter.

This doesn't tell you just by looking at it, they're using that function, but this basically means that bending was indeed uncorrelated, and injection molding was-- had some correlation. There's a little bit of correlation in the bending, because the variance comes from coupons that were cut from the same sheet of sheet metal and that sort of thing.

That's cool too. That's basically more of the same thing. Here's the one I want to show you is that in injection molding integral control-- it just zooms right in. The desired value was 1.436. It just pops you right onto that. I guess the measurements have some error in, but this is in inches-- 1.438, and a variance ratio of 0.4. So this is the one example from his data of implementing what we consider the ideal controller. Simply put, it does what we expected it to do.

So from these model predictions and the assumptions that we make, we make the argument that we can do this discrete time control stuff-- that discrete time control theory will well fit what's going on. And you get this condition, where you will always center this, and you will get varying degrees of either amplification or attenuation of the variance.

But the thing I want to emphasize is that this is hardcore feedback control. But the way we implement it, the way it could be implemented is simply to have a path of paper that says, if you measure this error, do the following thing. If you measure that error, do the following thing. And you will get these results. So it's highly implementable. It could be done automated as well, but it's highly implementable.

And as I understand, in the semiconductor field, the run-by-run is-- I guess my sense was that, when it first came out, everybody said, yeah, that's great, but we don't need it. And now, as basically the needs here and the criticality of the mean shifts have gotten tighter and tighter, they're saying, oh, jeez, we do need this, and it's starting to be used. Is that correct?

AUDIENCE: Yeah.

DAVID HARDT: Yeah. I think we talked about this earlier in the term. This whole issue-- when you get to process capability, if the state of the art is-- here's the process, and here's the process capability window, show me the problem you have. There's no problem. But if this window all of a sudden looks like this, say, because you're doing 10th micron lines, whereas before, you were doing 5 micron lines, or you're doing precision machine, whereas before you were doing non-precision-- this compared to this looks pretty awful.

And if you don't know how to go from here to here, what do you do? Well, again, run-by-run or cycle-to-cycle says, well, one thing I can tell you is I can get that baby right over in the middle there. And if I can work on the variance, I can do more.

And the other thing is the correlation helps. We've kind of avoided correlation-- because it's theoretically nasty-- throughout this class, but most processes do have a little bit of correlation in them. And if I have the correlation, then indeed, I can reduce this even more, as I mentioned. And the last thing I mentioned is that we have demonstrated this as well on a couple different processes, but Adam [INAUDIBLE], a couple of years ago, extended it to a 100-by-100 problem-- so 100 inputs, 100 outputs-- because we had this whole process we've been looking at in our lab for years called flexible die-forming forming.

And we had a tool, a forming machine-- full-size forming machine made up of a lot of elements you can move around. And in this particular case, there were 100 elements. We used to have a lot more, but he had 100. And the idea is you would form something, measure it, and say there's a shape error. How do I fix it? And you'd like to do that as quickly as possible.

And with this method, you could show that, within one or two cycles, you could reduce the error in that. The extension is pretty complicated, but most of the results just to show up the same way. OK? Labor Day well spent-- all right, if you're interested in this more, as I said, there are a couple of papers that I put on the web. There are two underlying theses-- a master's thesis and a PhD thesis.

That's just the work we've done here. And then, of course, in the semiconductor literature primarily-- a little bit in the statistical control literature-- there's more into this idea of run-by-run control. And let's hear the term they use, Duane? It's in the--

AUDIENCE: There's run-to-run, run-by-run--

DAVID HARDT: Yeah, run-to-run-- but even in the general statistics community, there is a term for-- oh, they call it adjustment, something like continuous adjustment or feedback adjustment.

AUDIENCE: --engineering control--

DAVID HARDT: Yeah, engineering control-- yeah. It's used a little bit derogatorily.

AUDIENCE: [INAUDIBLE]

DAVID HARDT: Yeah-- right, right. So yeah, if you have to have any more questions about that, feel free to contact me. OK, thank you all.