Automated Insulin Delivery

Lane Desborough
Co-founder and Chief Engineer, Bigfoot Biomedical

Hi, my name is Lane Desborough. We're going to talk about something I care deeply about: automated insulin delivery. I hope that by the end of this lecture you will appreciate the extent to which cyberphysical systems affect our lives. I have been fortunate enough to be exposed to many of these systems in different industries. Hopefully you find these insights valuable. Please interrupt and ask questions / challenge me as we go. I thrive on feedback.

Disclaimer and Disclosure

- Lane Desborough is an employee of Bigfoot Biomedical, Inc.
- Bigfoot Biomedical, Inc. does not market or sell any medical devices.
- This presentation was prepared by Lane Desborough in his personal capacity. The opinions expressed in this article are the author's own and do not reflect the view of Bigfoot Biomedical, Inc.

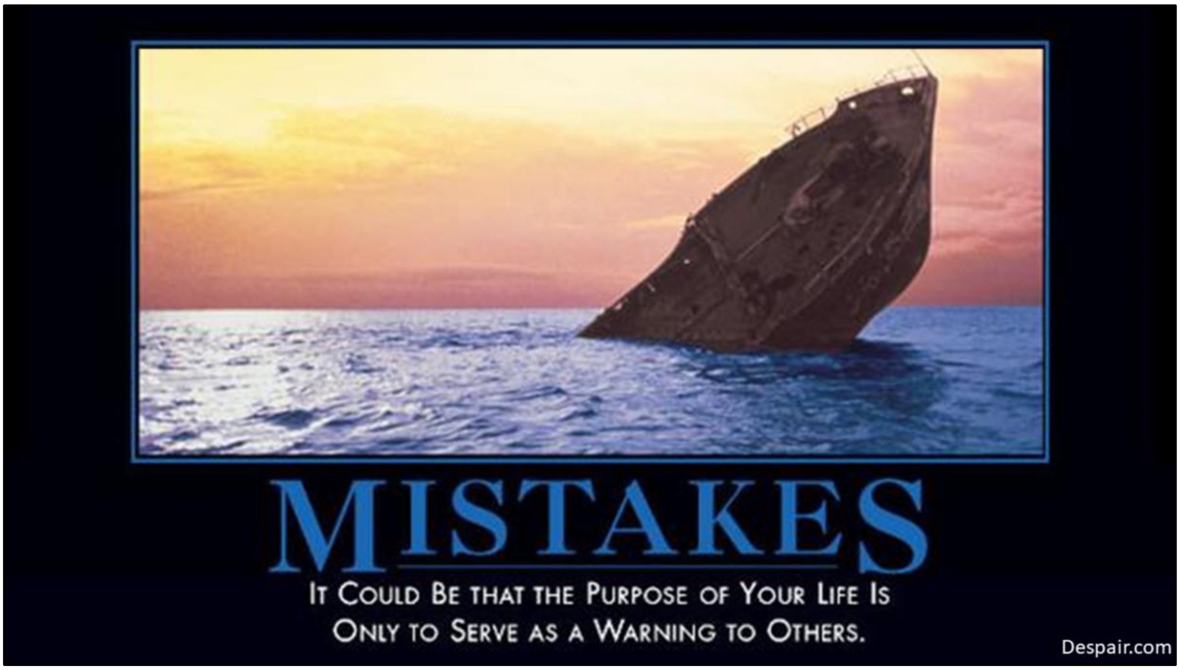
As a quick disclaimer, I am an employee of Bigfoot Biomedical but the views and opinions expressed hereafter are my own.

Lane Desborough

1992 M.Sc. Chemical Engineering (Process Control), Queen’s University, “Performance Assessment Measures for Univariate Control”

1990 B.A.Sc. Chemical Engineering with Management Sciences Option, University of Waterloo

I’m an engineer. I got my bachelor’s at the University of Waterloo and my master’s at Queen’s. I race sailboats and run for fun.



With the benefit of nearly 30 years of industrial experience, I've learned a lot of things the hard way. I've seen some scary things. I hope that through sharing some of these experiences and anecdotes, you will learn. I wish someone had done the same for me when I started my career.

What is this?

1. It cost billions of dollars to build, operate, and maintain
2. It is in some way unique and has no identical twin
3. It is one of the most complex systems in the world
4. It's but one part of a larger system
5. It deals with incredibly hazardous situations, 24x7
6. It provides something which society must have to survive
7. It employs hundreds of professionals from many disciplines
8. Its ongoing operation involves thousands of complex tasks
9. It's subject to ever changing conditions in the environment
10. It adapts and changes over a multidecade life
11. It is incredibly safe, reliable, secure, and efficient

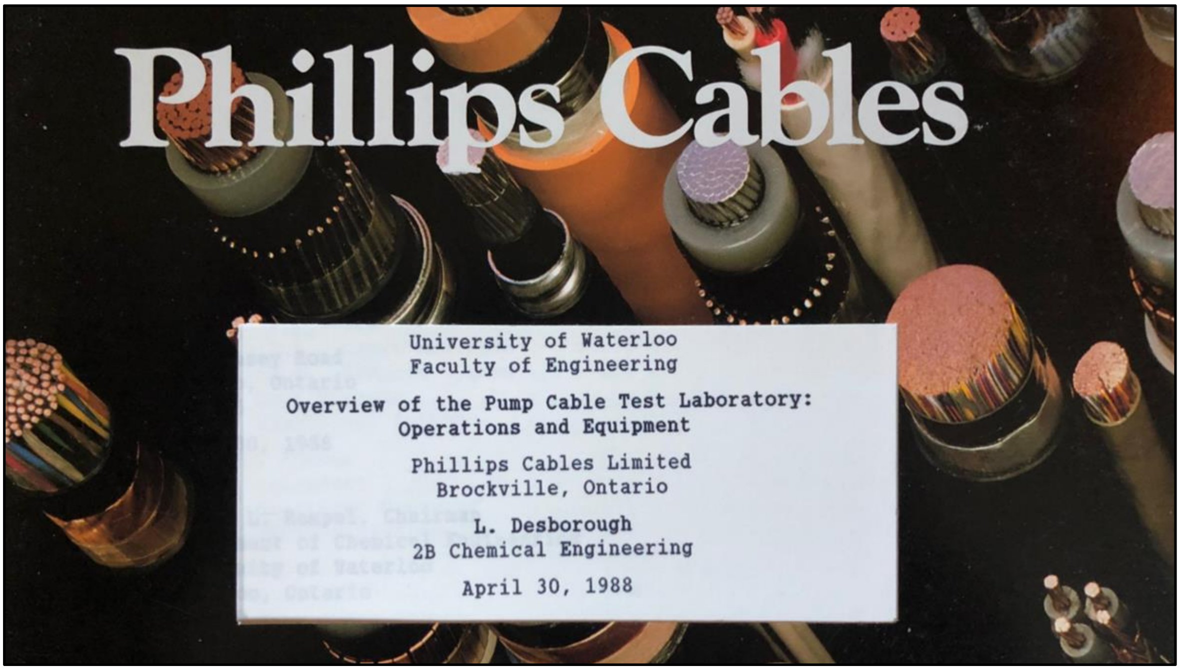
Let's start off with a quiz. What am I talking about here?

These are all examples of cyberphysical systems. They provide the needs of modern society, at low cost and great safety / reliability.

My first cyberphysical system

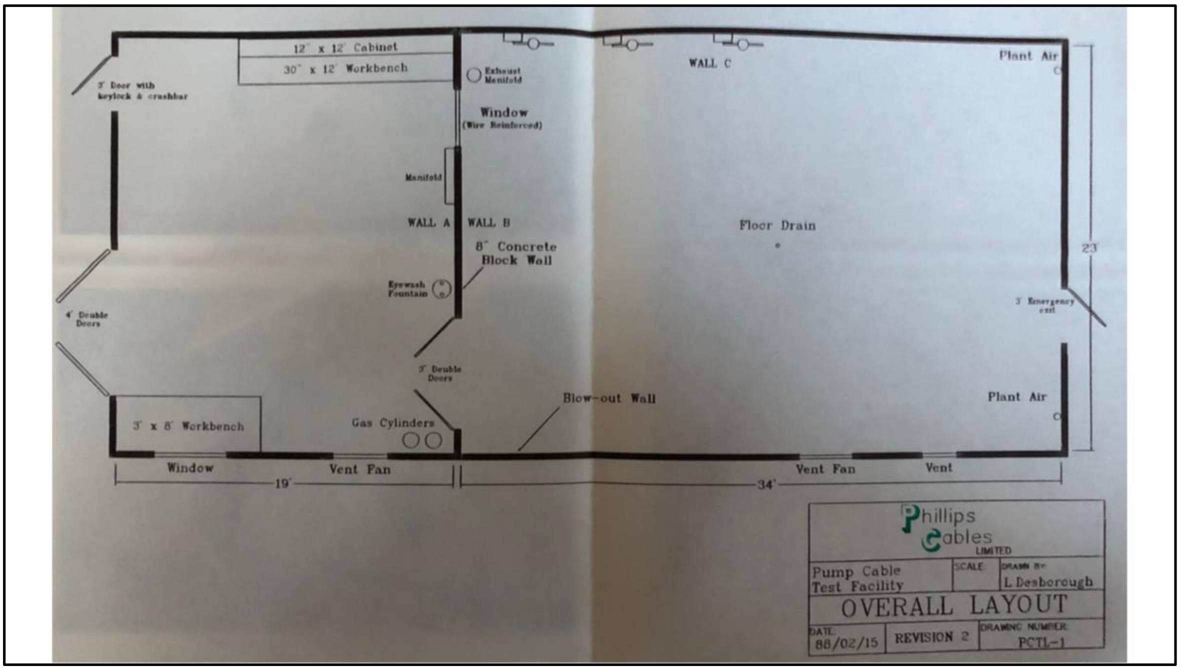
I'd like to indulge for a few minutes to reflect on a few of the systems which influenced my thinking in my early, formative years.

This is armored pump cable. It is used to power submersible pumps in oil wells far underground.



In second year at the University of Waterloo, I spent four months on a work term at Phillips Cables designing and implementing a lab to simulate the conditions at the bottom of an oil well so as to characterize the performance of this cable.

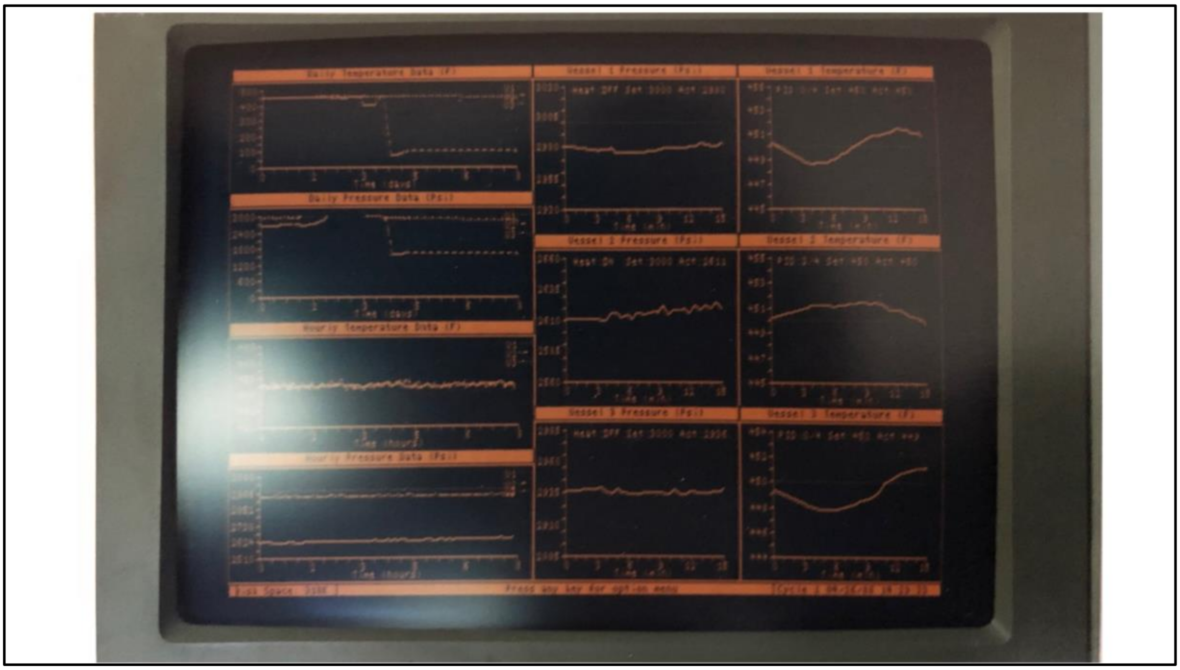
It was a smallish room in this massive plant which made all manner of medium voltage (~10,000v) power cables. On another work term I worked in a lightning simulator lab but that's a different story.



The lab comprised a room for the computers (cyber) and a room for the test apparatus (physical).

These are the downhole simulators – basically chunks of pipe we filled with oil and the pump cable samples, then heated them up to the temperature and pressure seen down hole.

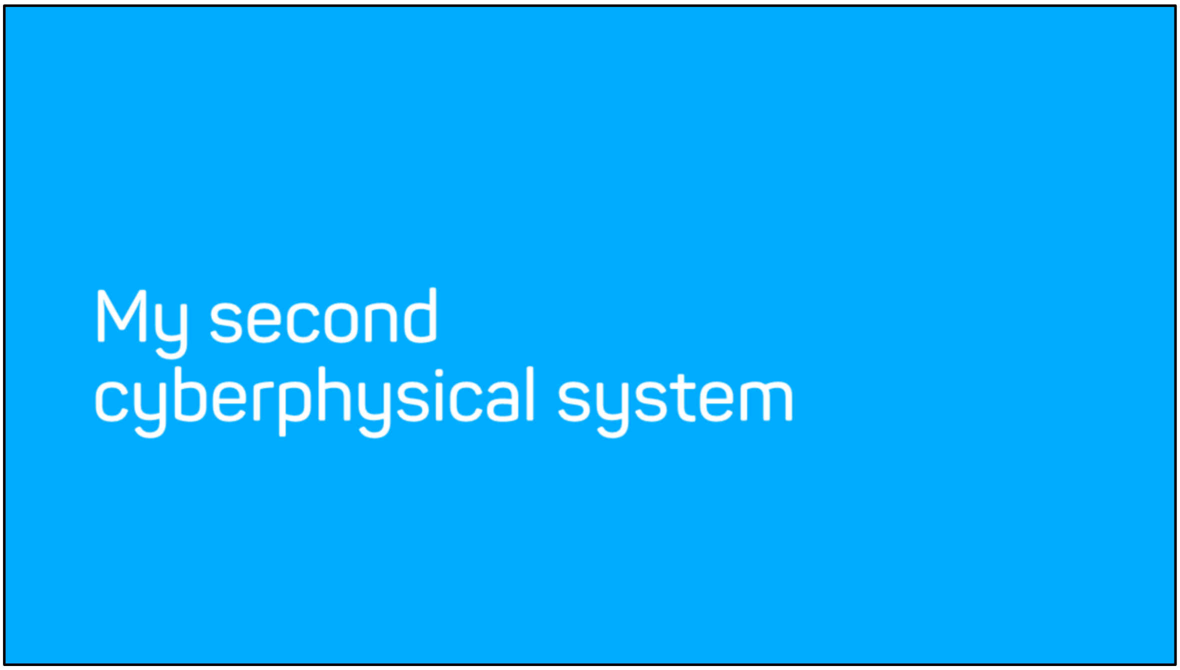
Here you can see the cutting edge printer, data acquisition, and compute platform. A $5 Raspberry PI is 100 times more powerful.



I hadn’t taken a control class yet – that would come in third year – so my boss handed me a textbook and said “I seem to remember this thing called a PID algorithm”. I used TurboPascal to build a complete SCADA (Supervisory Control and Data Acquisition) system. It used pulse-width modulation to control the electric resistive heaters which maintained pressure and temperature at setpoint.

The plant was closed 20 years ago.

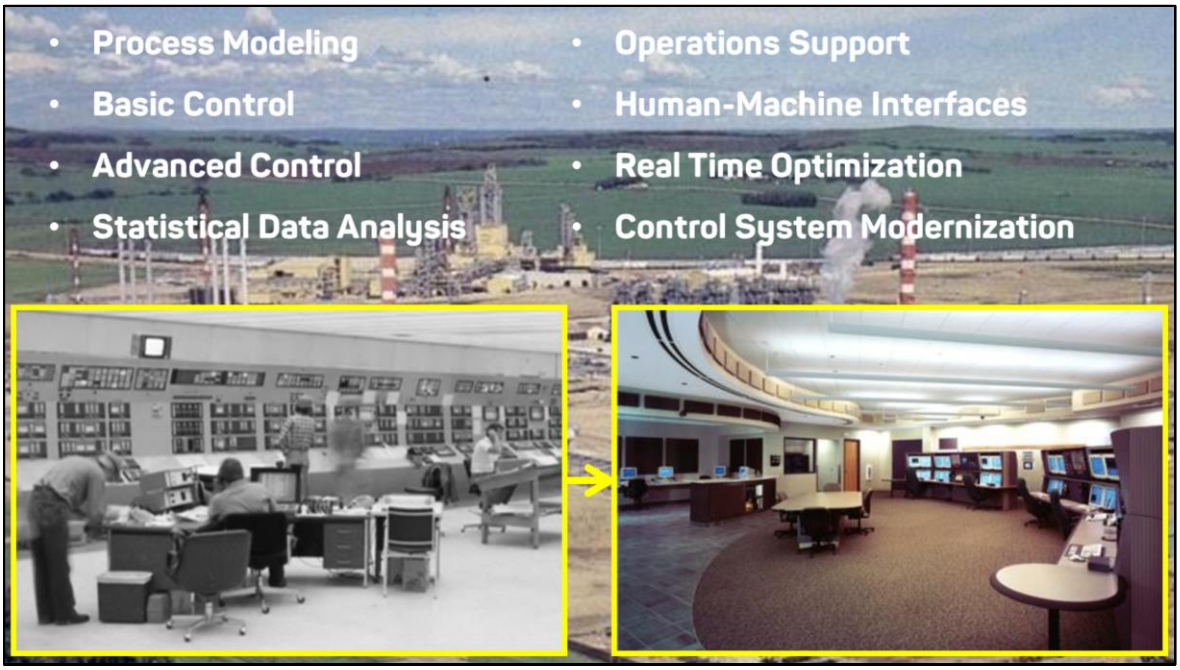
It’s now a park.



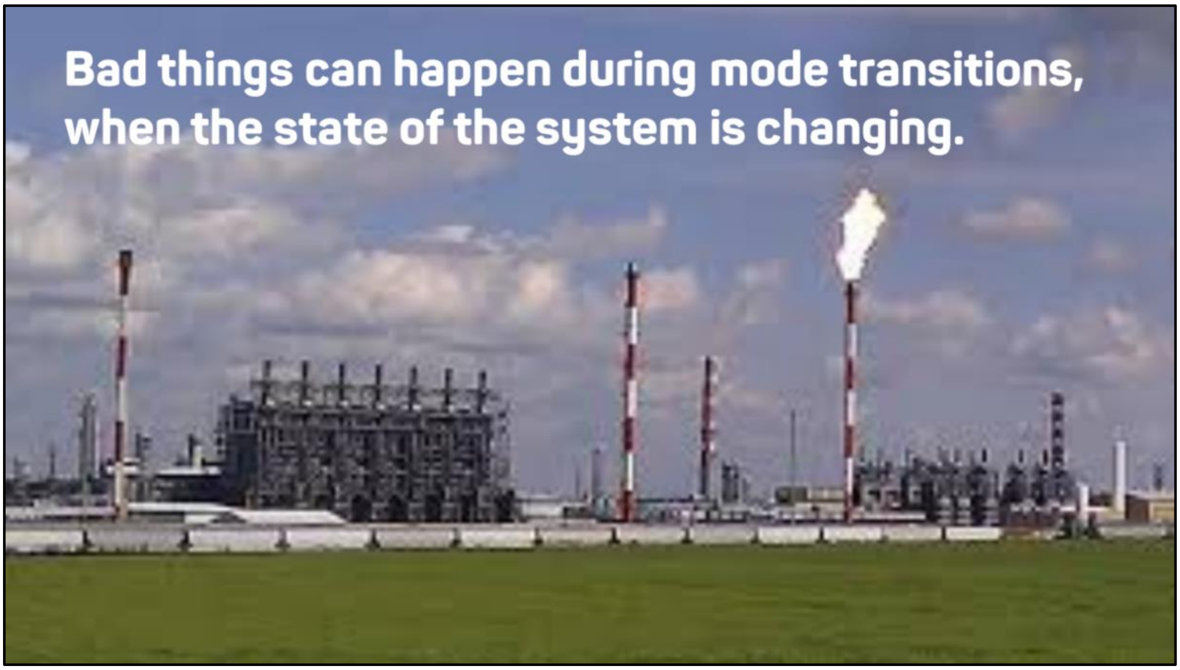
My last two undergrad workterms and my first job after grad school were at the following plant. I saw and learned things there that made a huge impact on my life.

This is the Nova Chemicals Petrochemical complex in Joffre, Alberta, Canada. With three ethylene plants, two polyethylene plants, a linear alpha olefins plant, and a hydrogen offgas plant, it's one of the largest facilities of its type in the world. It's now owned by the United Arab Emirates.

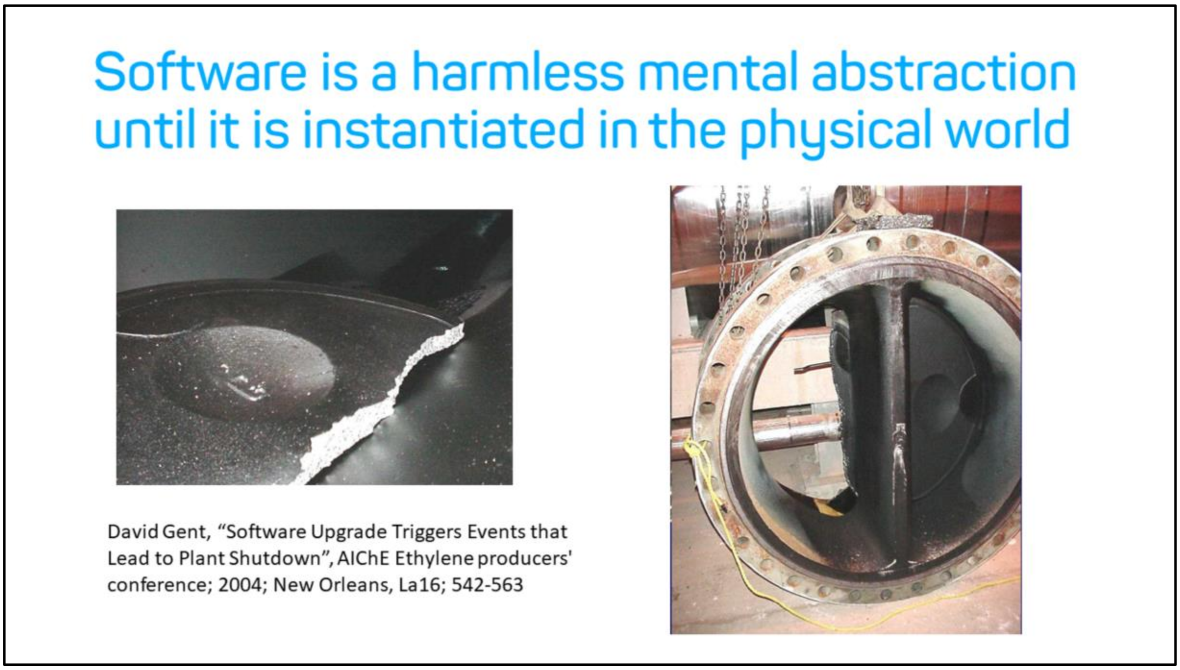
This plant is big. How big? 6 billion pounds of ethylene per year. 5000 control loops. All supervised by about 15 control room operators.



This plant is a good example of a continuous process industry facility. It was a great place for a new engineer to learn. During my time at Nova, one of the big projects I worked on was the modernization of the control system at one of the ethylene plants. I got to see industrial automation up close. In 1989 on a work term while still an undergrad, I implemented a basic controller. It was a single PI controller in a plant with thousands of other controllers in operation. It was there that I learned the majority of the work to implement a controller is not the control algorithm. It's the stuff around the control algorithm: the HAZOP (Hazard and OPerability analysis), the updating of the operator graphics, the documentation, the user training, the commissioning.



Plants like these are very dangerous. In an emergency the gas has to go somewhere so it gets rerouted to the flares through pressure relief valves. The red and white flare stacks are 250 ft tall and when the full flare is going the flame is another 250 ft on top. My most visceral memories are of an attempted re-start of the plant after a two week maintenance turnaround / massive debottlenecking project where we had also simultaneously replaced the control system and the flare system. I spent two weeks on night shift as the process control support leader. During each attempted re-start (we were flaring ~100 tons per hour of feedstock the entire time so there was immense pressure to get the plant running) I would be monitoring the control loops and tuning them on the fly. I tuned something like 100 loops. Sometimes they had been configured incorrectly (air-to-open instead of air-to-close) so they were providing positive feedback not negative feedback - a very bad thing. One night the flare blew out during an attempted re-start (the pilot on the new flare tip had a problem) and I remember being in the "blast resistant" control room wondering which way the wind was blowing the massive plume of uncombusted gas and what would happen when it found an ignition source. It re-lit off of the flare from an adjacent plant and we were fine. Bad things can happen during mode transitions, when the state of the system is changing.



In another incident, the company was reminded that software is a harmless mental abstraction until it is instantiated in the physical world.

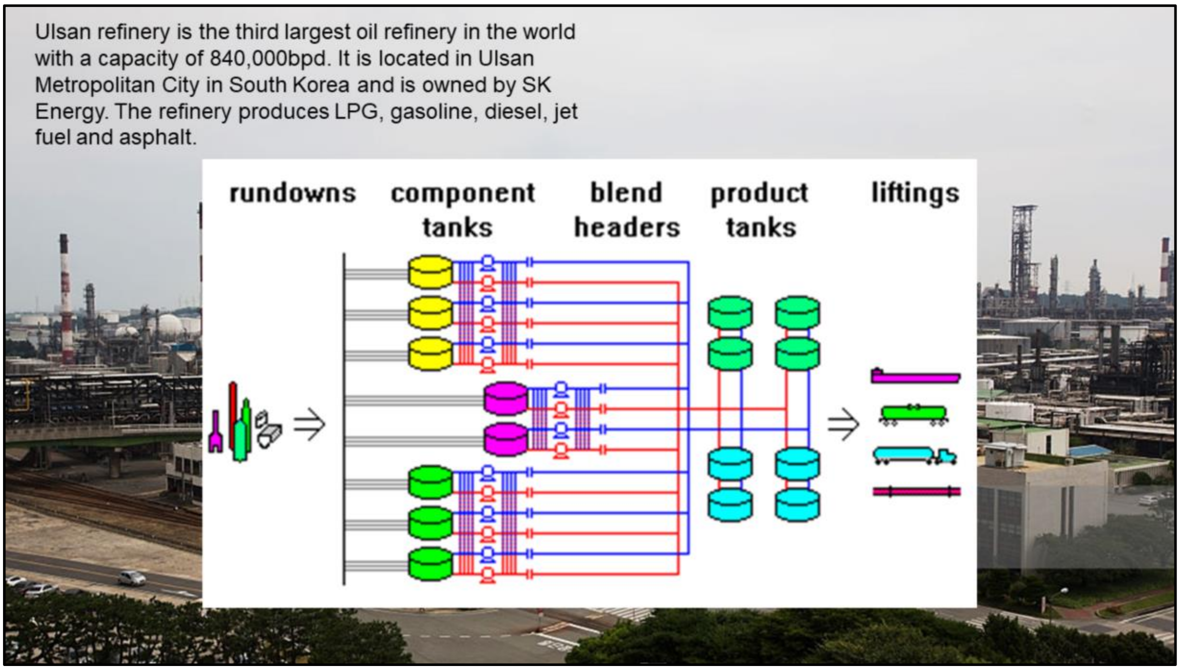
The routine upgrade of control system software resulted in the sudden closure of 82 valves during operation. My friend and fellow control engineer Mike was the one who pressed the button. The software was not new. It had over 2 million hours of operation at this and hundreds of other plants. In fact the same upgrade had been performed without incident on 7 other systems at the same plant that very morning. A dangerous exothermic reaction went from 180 degrees to over a thousand degrees in minutes, destroying the catalyst. There were many other losses from this incident. Many pieces of equipment were damaged in the plant trip, including this compressor check valve which is 42" in diameter. It took nearly two weeks to restart the plant - approximately one hundred million pounds of lost production. One hundred million pounds.

At the time of both these events I was working for Honeywell, the provider of the control system.

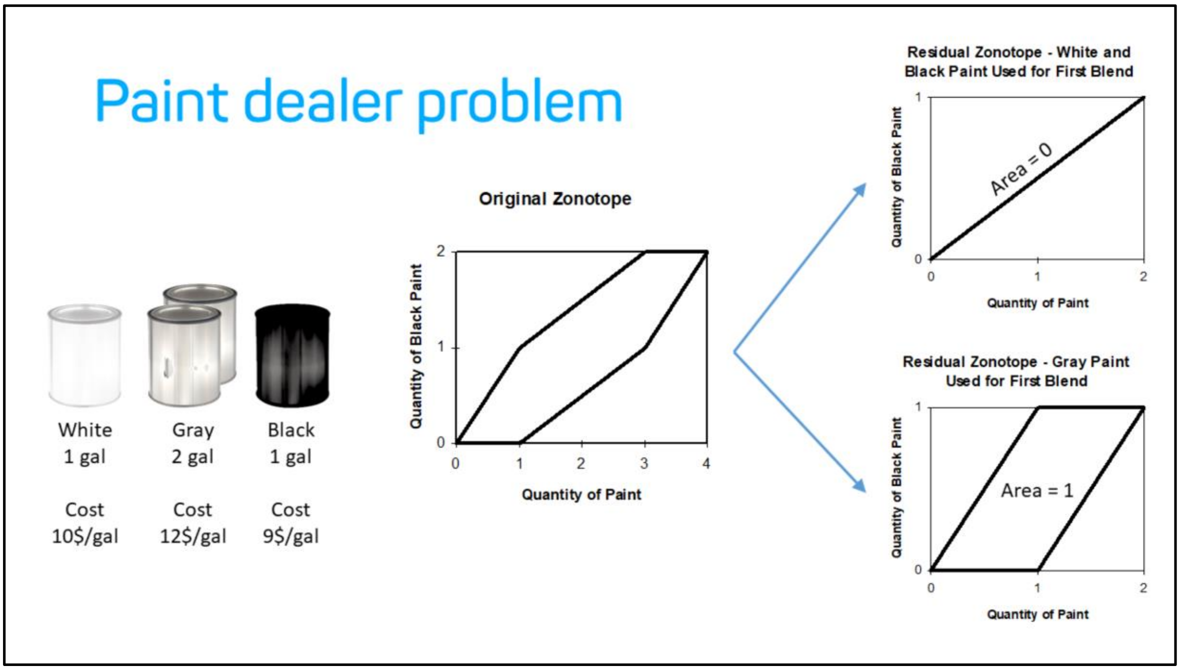
My third cyberphysical system

I promise I will get to automated insulin delivery. Just a few more motivational examples.

In 1995 I moved to Southern California and began work at Honeywell, the largest provider of automation for the continuous process industries. My first project was to install blend planning and scheduling software at what was then the largest single site oil refinery in the world, in Ulsan, Korea. My first two weeks at Honeywell were in Lyon, France, taking delivery of the software which we had licensed from the Elf, the French national oil company, and preparing for the FAT (factory acceptance test) with some very detail-oriented Korean engineers.



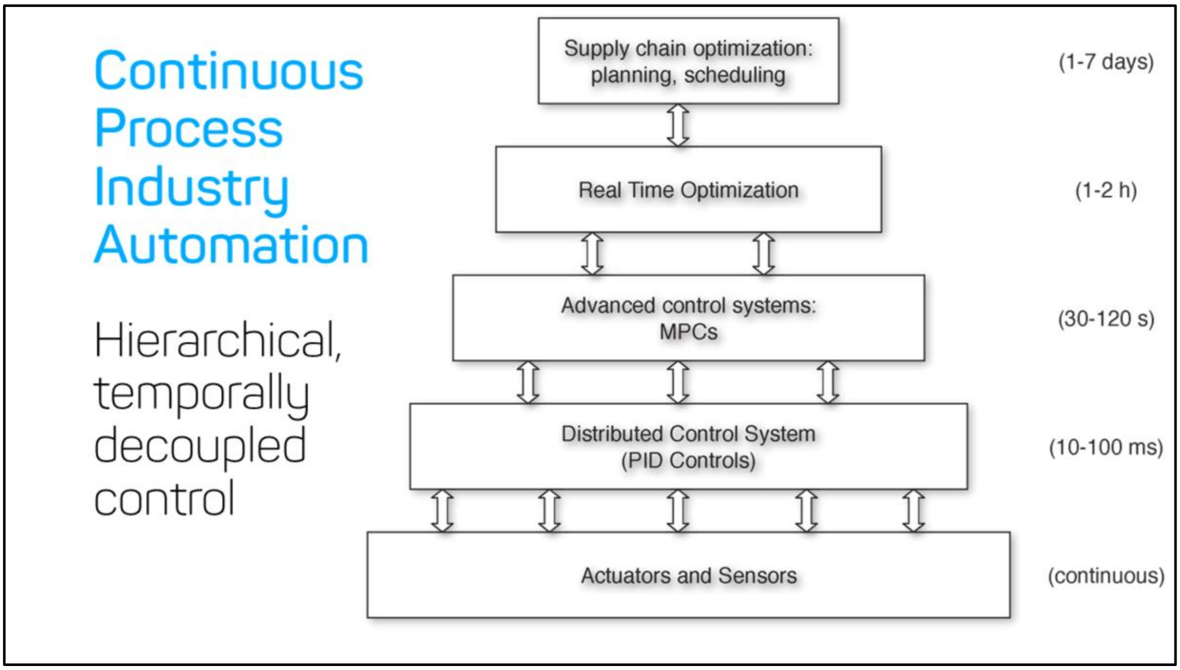
Oil refineries basically separate crude oil into components, upgrade the components in reactors, then blend the components together to make products like unleaded gasoline. As you can imagine, at the largest refinery in the world this is a complex and difficult task.



Have any of you heard of zonotopes? It turns out zonotopes are used in temporal logic packages like BREACH. We also used zonotopes to blend products in oil refineries.

Here's a simple example. Let's say you are a paint dealer and sell only "shades of gray" – white, gray, and black paint. Your inventory is shown here, along with the cost you pay your supplier. If a customer comes in to the store and asks for two gallons of gray paint, you can satisfy their product demand by selling your stock of gray paint, or you can mix the black and white paints together and give him that. Let's say that you make more money if you sell him the mix (19$ vs. 24$ cost). So you do that – you mix the white and black.

All good, right? What if the next customer comes in and asks for white or black paint? You don't have any left because you just sold them to the first customer. We used residual zonotopes to maximize the ability to satisfy unknown product demands.



The blend planning and scheduling example I just shared is part of the top layer of the control hierarchy commonly used in continuous process industry facilities.

Supply chain optimization software issues new targets infrequently to the RTO layer underneath, which issues new targets to MPC underneath, and so on.

Hierarchical, temporally decoupled control allows disturbances with different frequencies to be rejected at the appropriate layer. Care must be taken to ensure the layers are sufficiently separated to prevent “fighting”. This is called decoupling.

Through the course of my career at Nova Chemicals and Honeywell, I worked on every one of the layers.

My 4th-65th cyberphysical systems

Put on your seatbelts, here we go.

Australia, Brazil, Britain, Canada, France, Germany, Hungary, India, Japan, Korea, Malaysia, Mexico, Singapore, South Africa, United States

Ok so I'm not going to go through the remainder of my cyberphysical system experience except to say that it was a great way to see the world and meet a lot of really amazing people and see a lot of incredible things.

29

From the oil sands of Alberta

30

To the savannahs of South Africa.

31

To the jungles of Brazil

This is a long wall machine at a trona mine in Wyoming. 1700 feet underground. There is nothing but those hydraulic jacks on the left to hold the roof up.

Perhaps a hundred control rooms. I think the scariest one was on top of a reactor which was making hydrofluoric acid. To evacuate you had to get past the poison gas cloud. Or the ones in the building making uranium hexafluoride, protected by guard with submachine guns, and x-ray machines, and Geiger counters. Or the ones in “explosion resistant” buildings with doors so heavy they couldn’t be opened by humans and required pneumatic actuators. Or the one in South Korea the night we got the call that an empty North Korean minisub had been found on a beach – watch out for saboteurs.

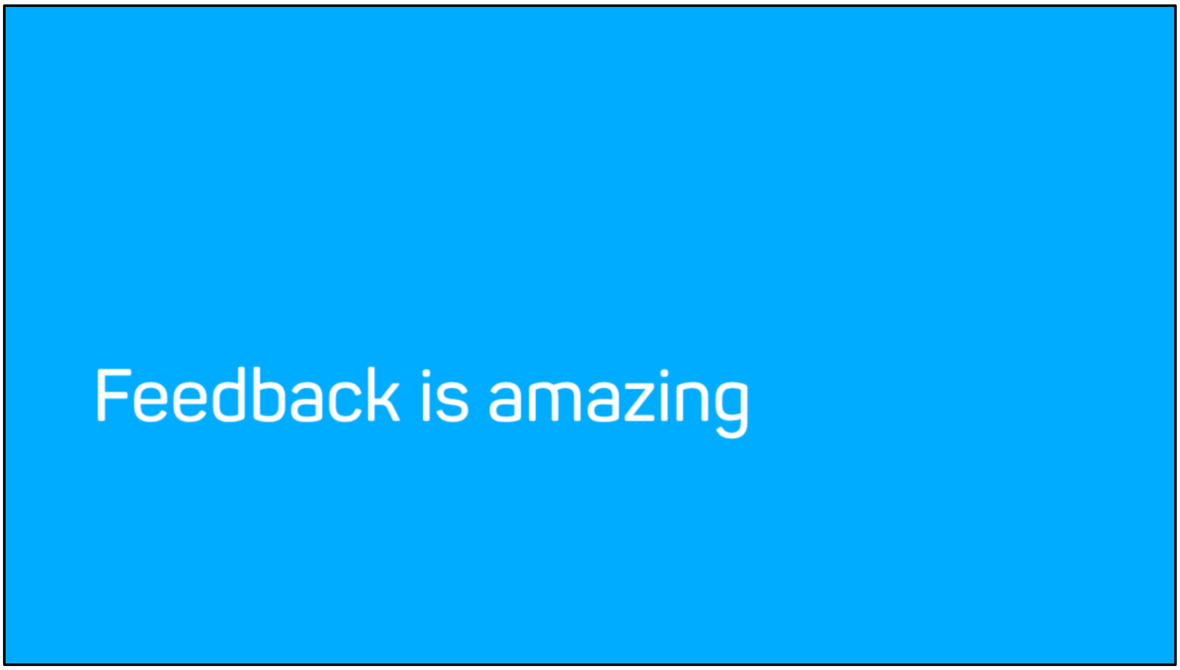
Cyberphysical systems

	USA	Canada	Korea	South Africa	Australia	France	Germany	Hungary	Japan	Malaysia	Scotland	Brazil	Grand Total
Chemical Plant	12	4	1	1			1		1	1	1		**22**
Oil Refinery	12	3	1		1	1		1					**19**
Discrete Manufacturing Plant	9	1											**10**
Mine	3	2		1								1	**7**
Paper Mill		2											**2**
Power Plant	2												**2**
Aluminum Mill		1											**1**
Pipeline	1												**1**
Steel Mill		1											**1**
Grand Total	**39**	**14**	**2**	**2**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**65**

Most of my cyberphysical experience is in chemical plants and oil refineries. I don’t regret a single minute. How many careers take you to places like this?

This the gas turbine combined cycle power plant control room at Stanford University, where they make ice in a 4 million gallon tank under a parking lot at night when the electricity rate goes down. This is one of the last facilities I visited before my life took an unexpected turn. By this point I had worked in three distinct industries: petrochemical production, industrial automation, and power generation.

https://alumni.stanford.edu/get/page/magazine/article/?article_id=32125



One think I learned across all these experiences is that feedback is amazing. All of these cyberphysical systems are using feedback control. Why? Because it (mostly) works.

The Power of Feedback

Karl Åström

Feedback has some amazing properties, it can

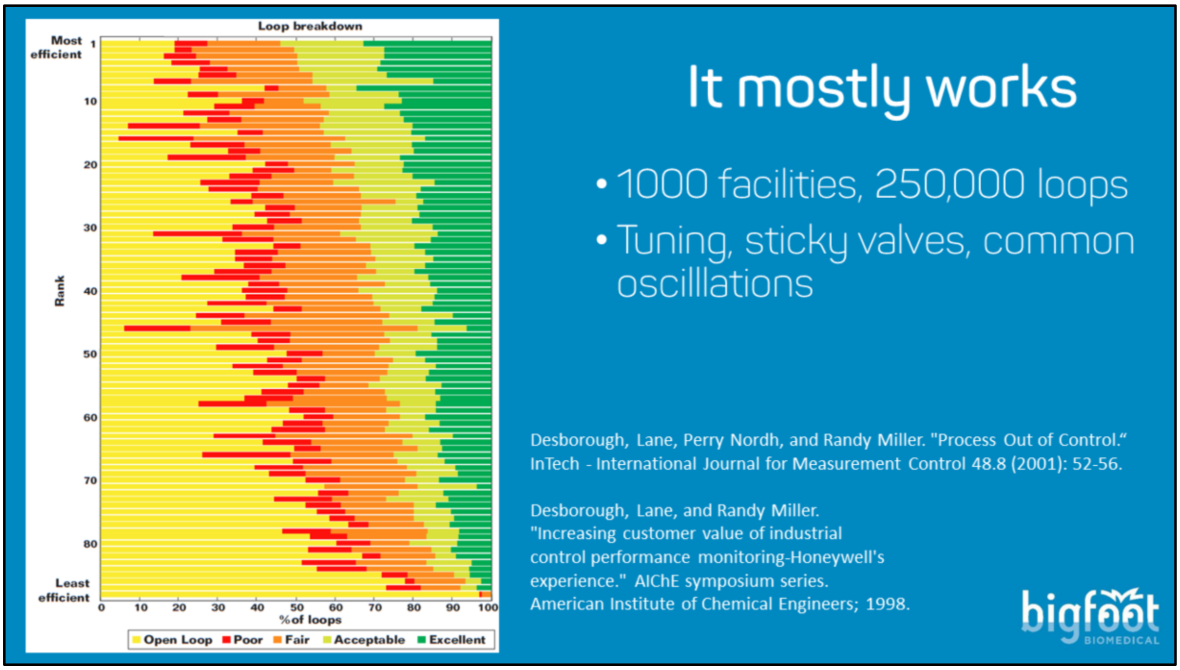
- make a system insensitive to disturbances and component variations,
- make good systems from bad components,
- stabilize an unstable system,
- create desired behavior, for example linear behavior from nonlinear components.

The major drawbacks are that

- feedback can cause instabilities
- sensor noise is fed into the system

Karl Astrom, one of the fathers of modern automation, highlights the tremendous advantages of feedback.

Provided you have a high quality sensor, and provided your tuning isn't too aggressive which leads to oscillations, feedback will reject disturbances and compensate for poor or degrading components and even do a good job accommodating changes in the process itself. With very little muss or fuss. There are billions of closed loop feedback systems in the world.

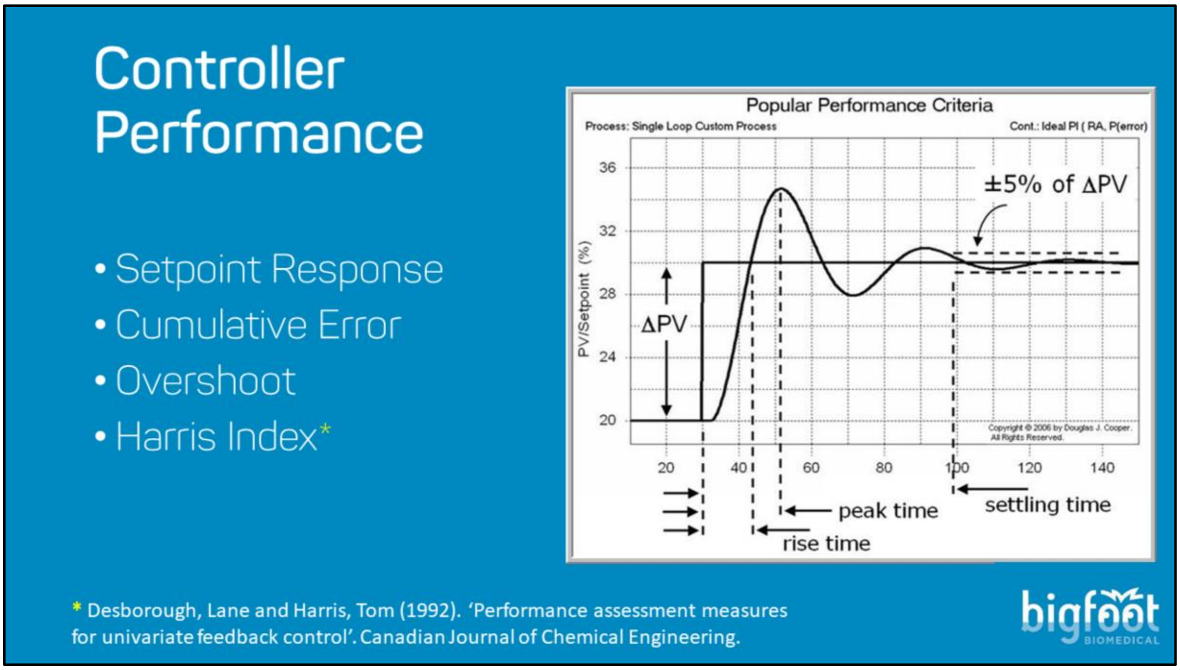


The tuning doesn't need to be spot on.

The valve may start sticking.

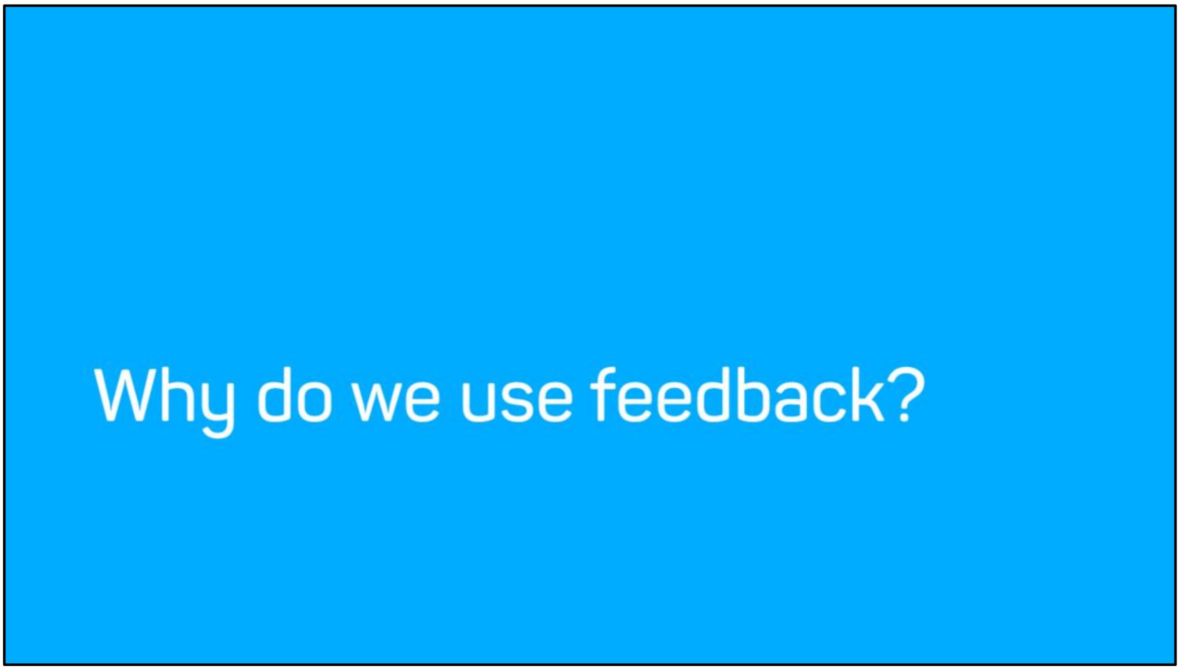
The process may change over time, such as a heat exchanger gradually fouling.

The loop keeps chugging along.



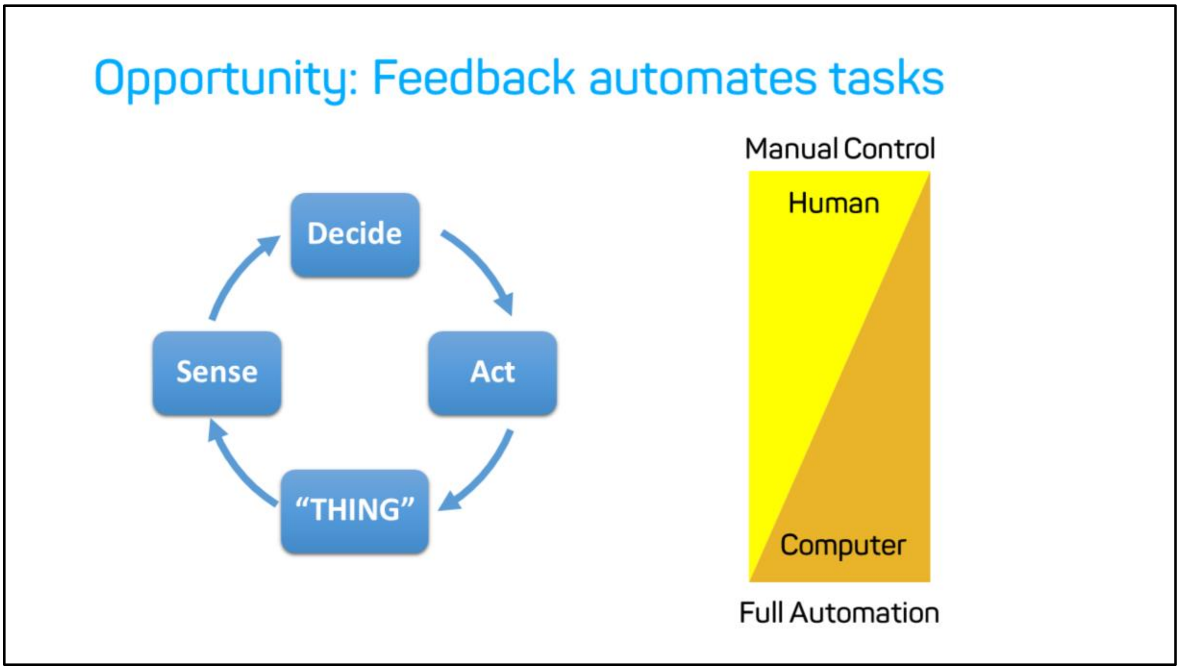
Performance standards have remained essentially unchanged for 30 years. Yes, some new measures like the Harris Index have come along, but for the most part, performance is assessed infrequently, on a case-by-case basis, using simple statistics and / or visualization of loop response to setpoint changes or disturbances.

This is in large measure due to the uniqueness of each loop – overarching performance standards don't make much sense when one loop could be controlling the basis weight of paper coming off of a paper machine and another is controlling the reflux flowrate on the overheads of a crude distillation unit in a refinery.



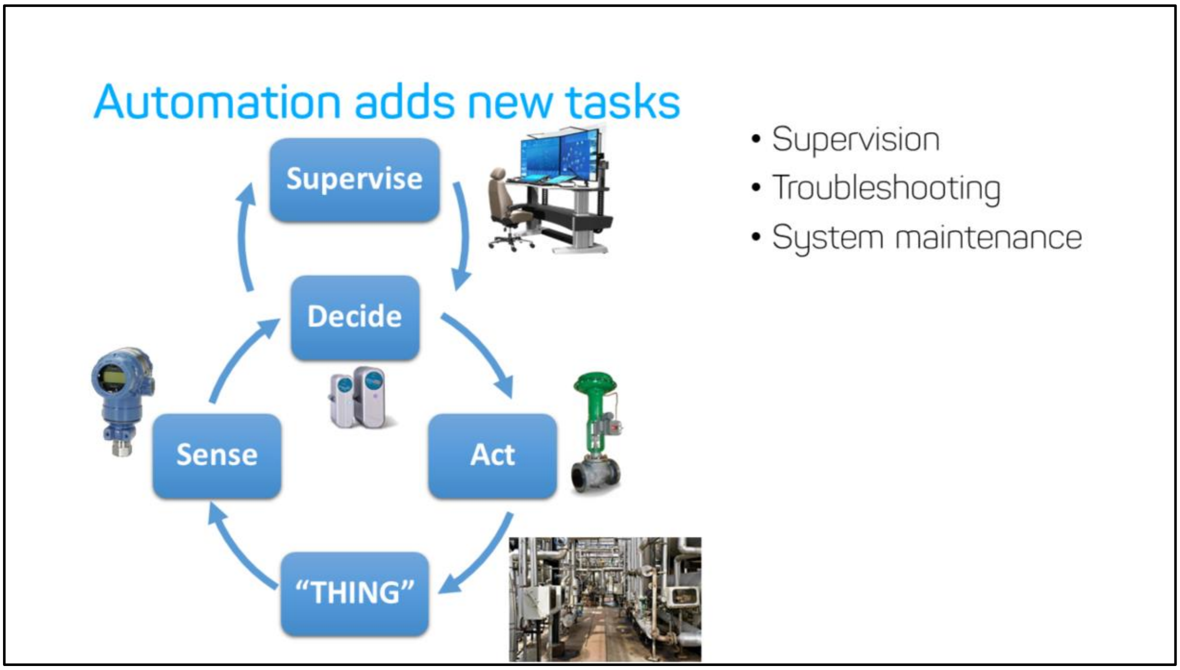
I’m about to say one of the most important things I’m gonna say. It’s something my grad school supervisor taught me 25 years ago.

My favorite way to explain control to people is that is transfers variability from a place where it hurts to a place where it doesn't hurt as much, in order to make a human's job easier. So our car's cruise control transfers variability in speed to variability in fuel consumption. Who cares if I'm using a little more or less gas, I just want to drive 65 mph. Same with these other systems. They take the variation introduced by disturbances and, through feedback, transfer it to the manipulated variable. And that makes the human's job easier.



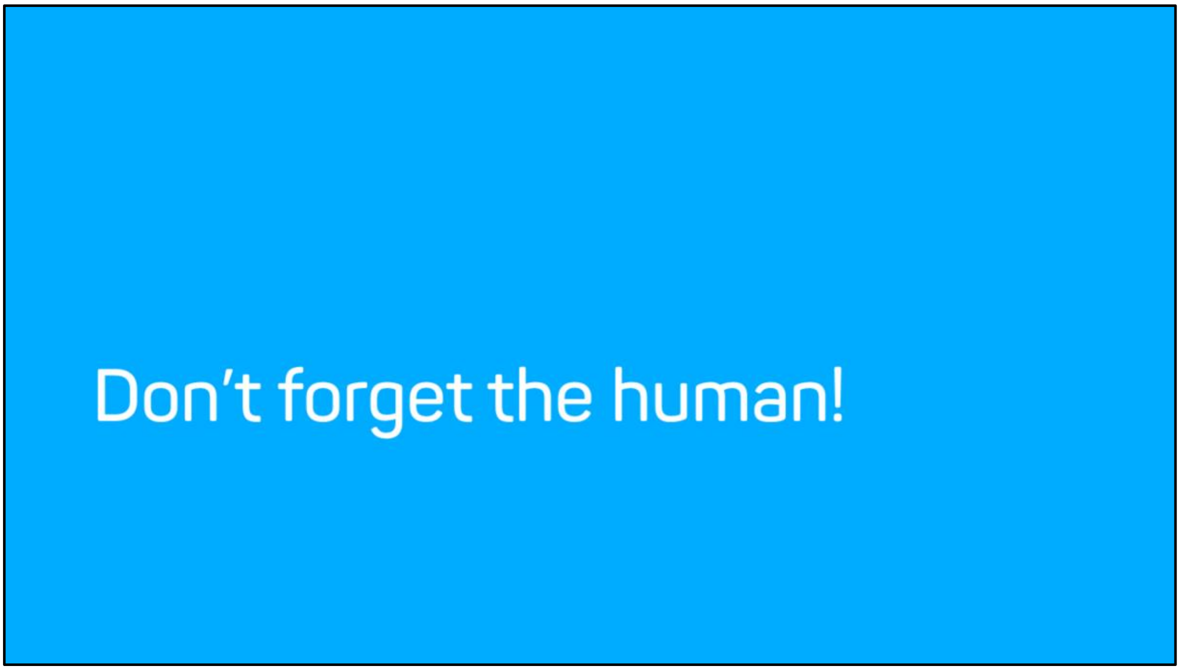
We sense something with a sensor, we decide what to do with a control algorithm, and we perform an action with an actuator or final control element, thereby affecting the thing being controlled. Sense, decide, act. Closing the loop. Feedback control.

What is often missed is that each of these tasks – sensing, deciding, acting - can be performed by a human, a computer, or a combination.

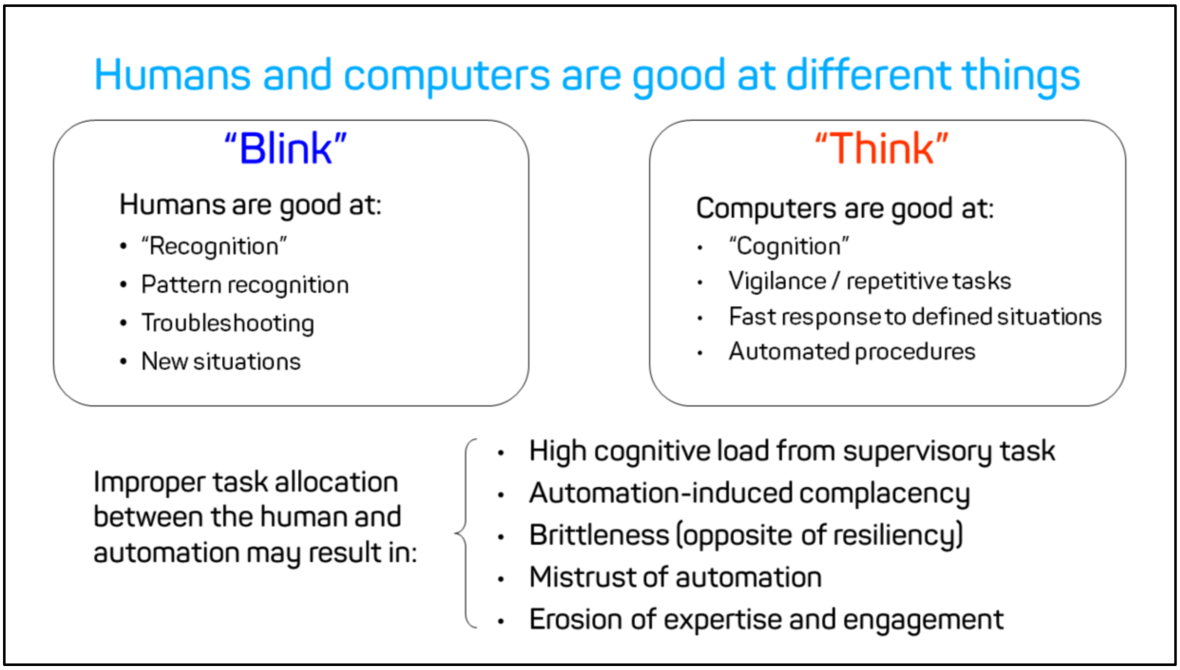


In addition, many miss the fact that new tasks are added with automation. Some of these tasks are quite difficult. Supervising the automation. Troubleshooting the automation when it has a problem. Performing maintenance on the automation.

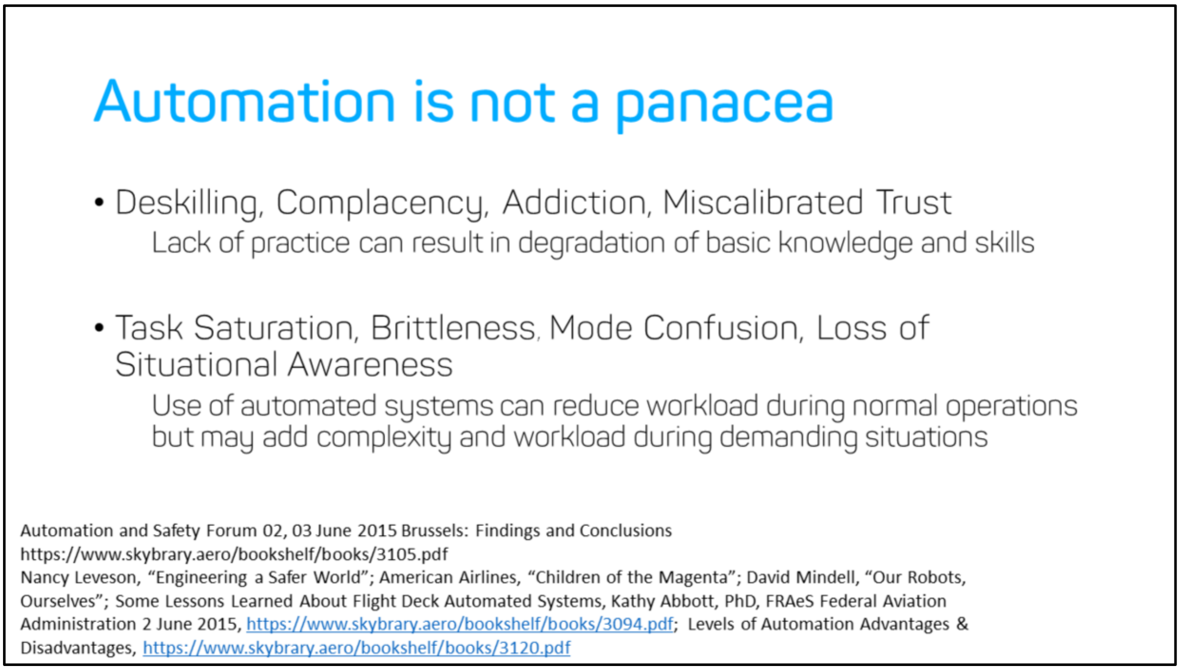
"The plant of the future will have one dog and one human. The human's job is to feed the dog, and the dog's job is to keep the human from touching the plant." I first heard this joke two decades ago. There are important reasons why there are still humans in cyberphysical systems.



It is so easy with all this exciting talk of transfer functions and cyberphysical systems to forget the human. Don’t forget the human.

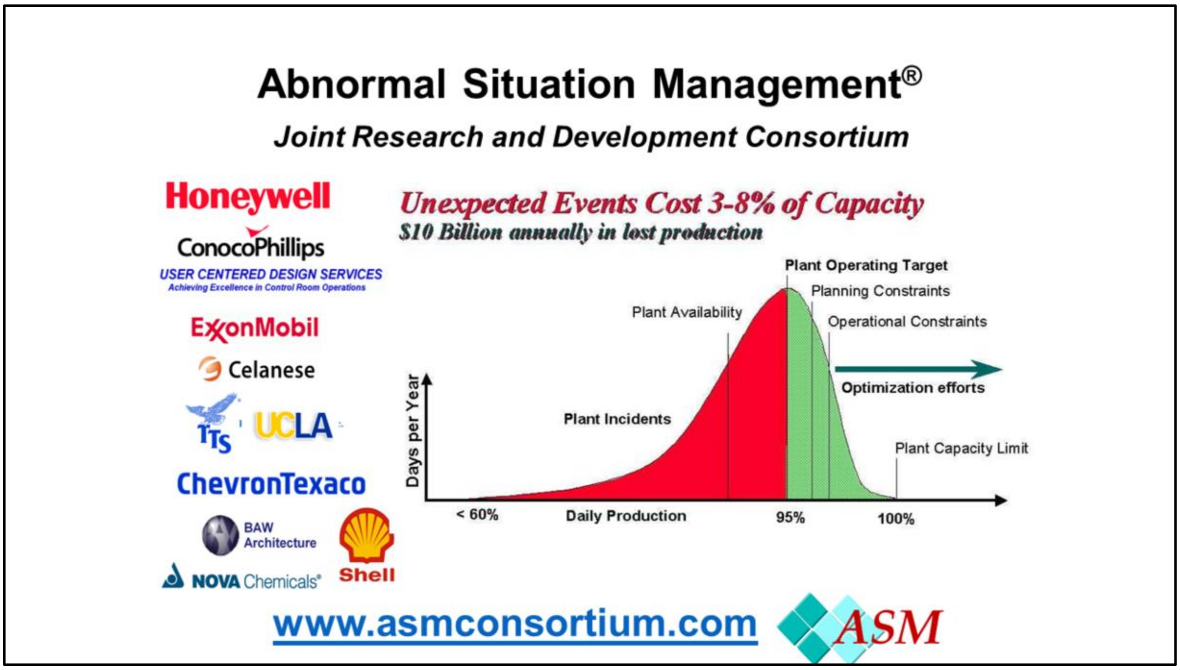


Properly allocating tasks is critically important when considering automation.
The human has information about the past, present, and future which is unavailable to the computer. The human has five senses.

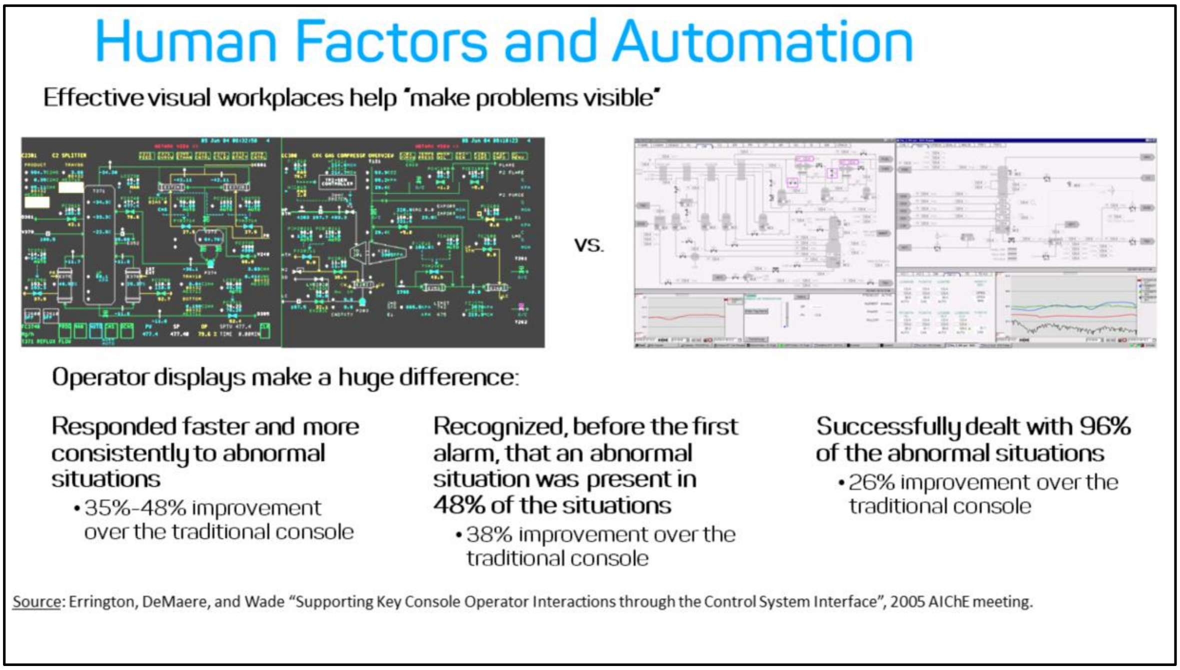


When humans are removed from the loop, bad things can happen. They become deskilled. They become complacent or even addicted to the automation to the point where they are afraid to turn it off and take over control. They may over- or under-trust the automation.

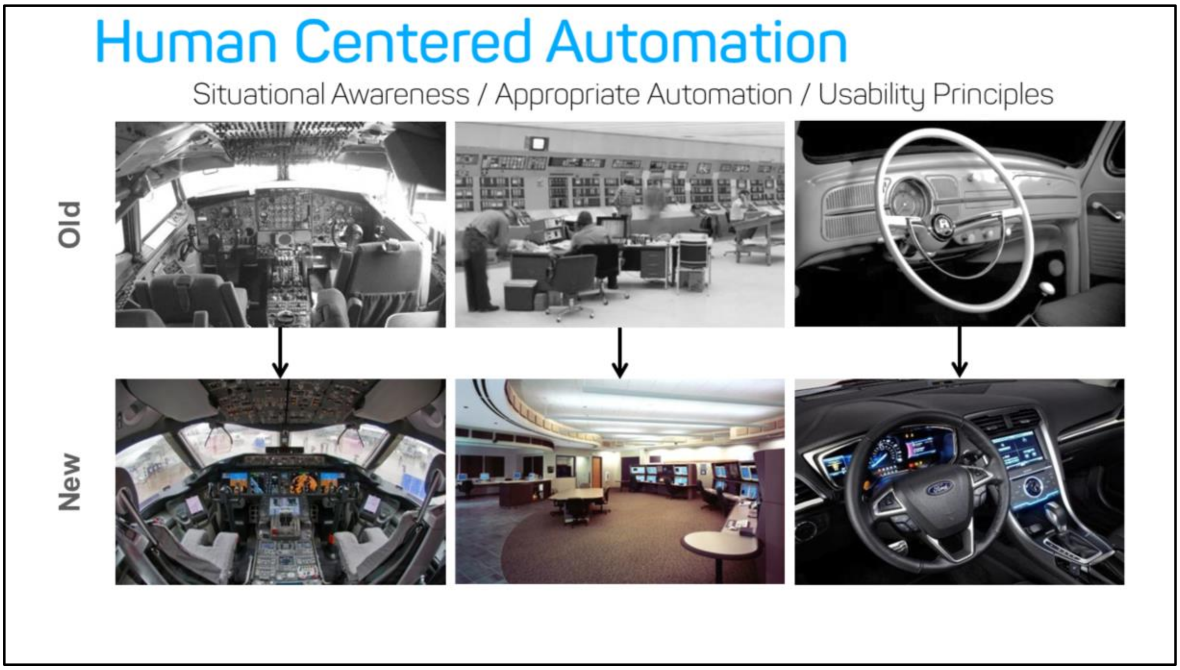
And worst of all, during critical situations they can get distracted and overwhelmed, unable to re-insert themselves into the loop and make the necessary control actions to save the day.



In the late ‘80’s incidents were happening at chemical plants and refineries. The industry investigated and determined that alarms and automation and human factors were contributing 40% of the 16B$ of annual losses accruing to the industry. So the major players got together and formed the ASM Consortium. I had the privilege, through Nova Chemicals and later Honeywell, to be part of the consortium for 15 years. This is where I developed a strong appreciation for the imperative to consider the human operator during design of alarm, display, and closed-loop automation systems.



At the same ethylene plant I've spoken so much about, my colleagues used high fidelity operator training simulators to demonstrate the benefits of implementing well-designed operator displays. The display on the right - which embody the standards and principles developed by the ASM Consortium – were markedly better than the one on the left at helping the operator respond, recognize, troubleshoot, and resolve abnormal situations.



These are examples of the transformations I've seen in areas of automation which affect my life.

The left is cockpit from a mid '70's passenger jet which required at least three people in the cockpit. Modern airliners have fewer humans and much more automation.

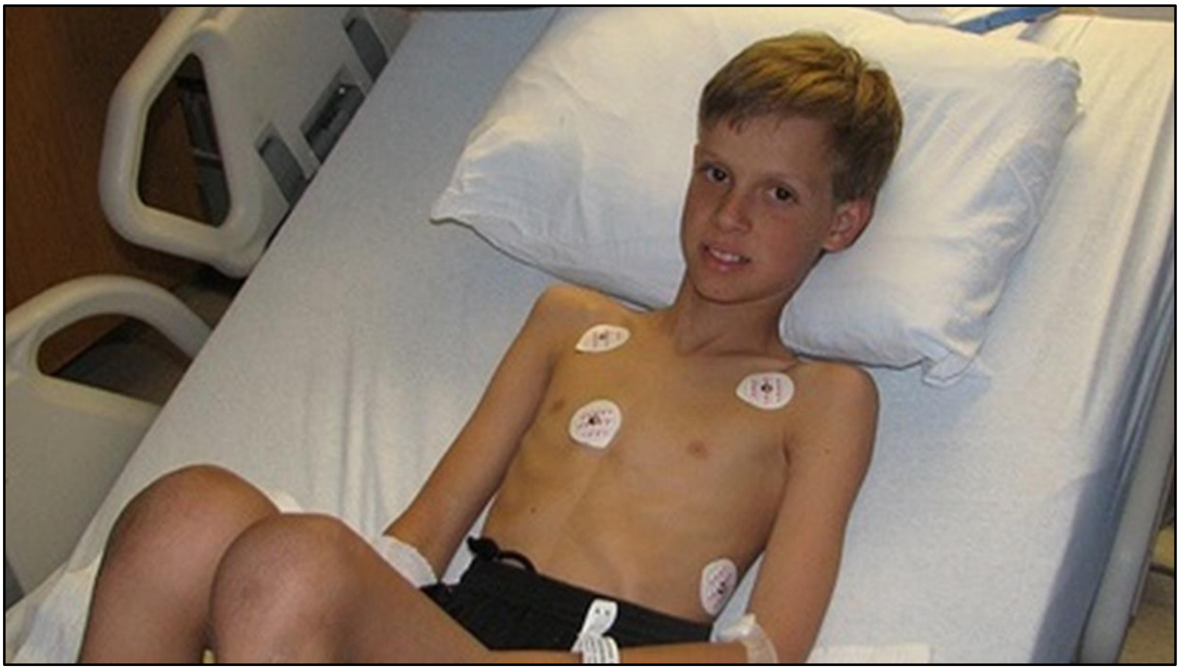
The middle shows the plant I worked in before and after the transition to digital control systems. One side effect is that operators can no longer "walk the board" to get a glance at the state of the plant. Another is the replacement of the hard-wired alarm "light boards" with easy-to-implement alarms which now regularly cause alarm floods.

The right shows the first car I ever drove – a 1967 Volkswagen Beetle with no gas gauge or radio, and my wife's Ford Fusion "magic car" with all manner of automation.

Some believe that automation has been taken too far, that taking the human out of the loop is causing new kinds of accidents. Some tasks are transferred from the human to the computer. But automation introduces new tasks, and has the potential to change the nature and severity of hazards.

My most recent cyberphysical system

Ok, sorry for the long preamble. Now I'm going to talk about automated insulin delivery systems. In the summer of 2009, I was working on the Smart Grid for GE Energy. I got a call from my wife on a Friday afternoon to tell me that our son had been diagnosed with type 1 diabetes.



With no family history, I had no idea what this meant. I knew that diabetes involved blood glucose and insulin, but that was about it.

Insulin requiring diabetes has a large unmet need

A **data driven** disease

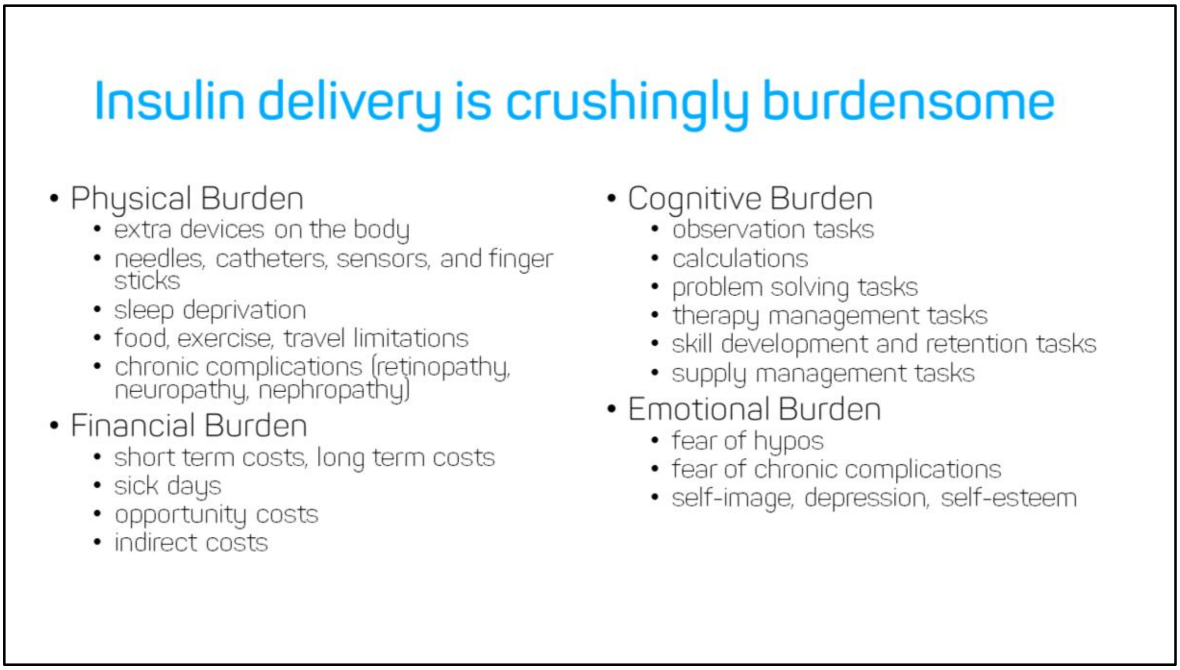
where an **average person** is required to **self-titrate**

a **dangerous medicine** on a **24x7x365** basis;

more than **66%** of them

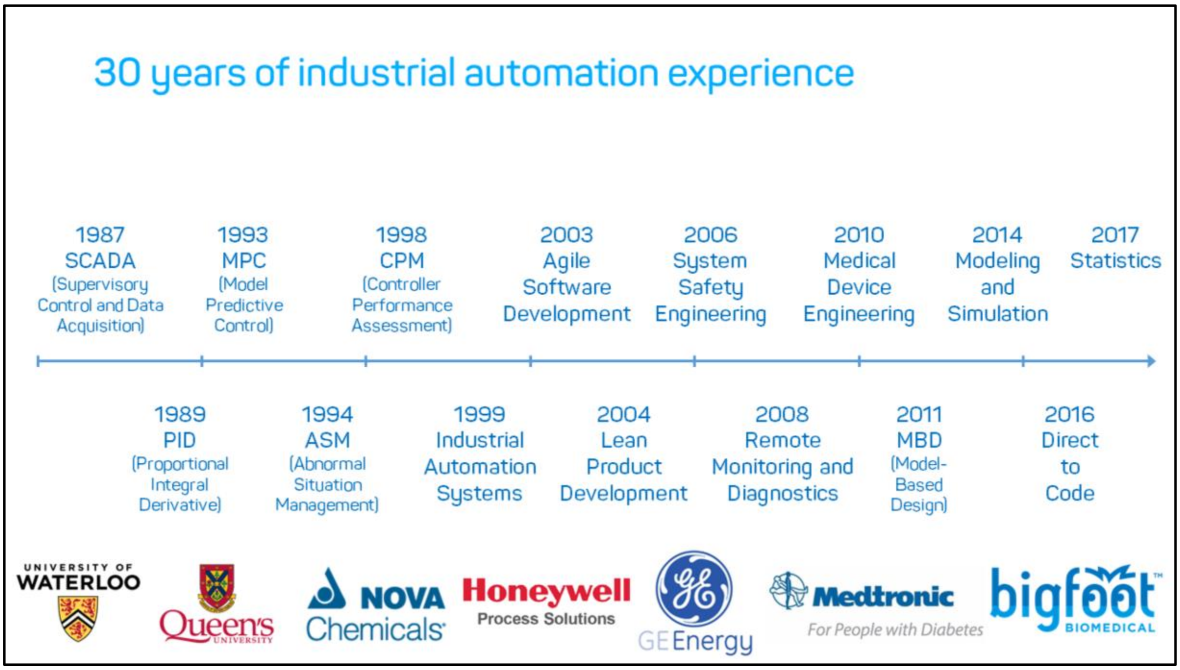
are treated by **primary care physicians**

I was soon to learn that diabetes is a big challenge.



To stay alive, people with type 1 diabetes must take insulin to lower their blood glucose because their immune system has destroyed the insulin-producing cells in the pancreas. Although insulin sustains life, delivering insulin is dangerous and crushingly burdensome. It involves the physical burden of carrying stuff around, poking yourself all the time. It involves costly drugs and devices. It is a “thinking disease” requiring plenty of complex cognitive tasks, driving a high emotional burden.

With type 1 diabetes you are only ever about 6 hours away from disaster. That’s four chances to die, every day, for the rest of your life.



It took no time for me to realize that insulin delivery was in fact a cyberphysical system control problem. All of the things I'd done up to this point in my career could be applied to type 1 diabetes.

Through a set of very fortunate circumstances, within months of my son's diagnosis I found myself at Medtronic Diabetes, leading the commercialization of their next step towards automated insulin delivery.

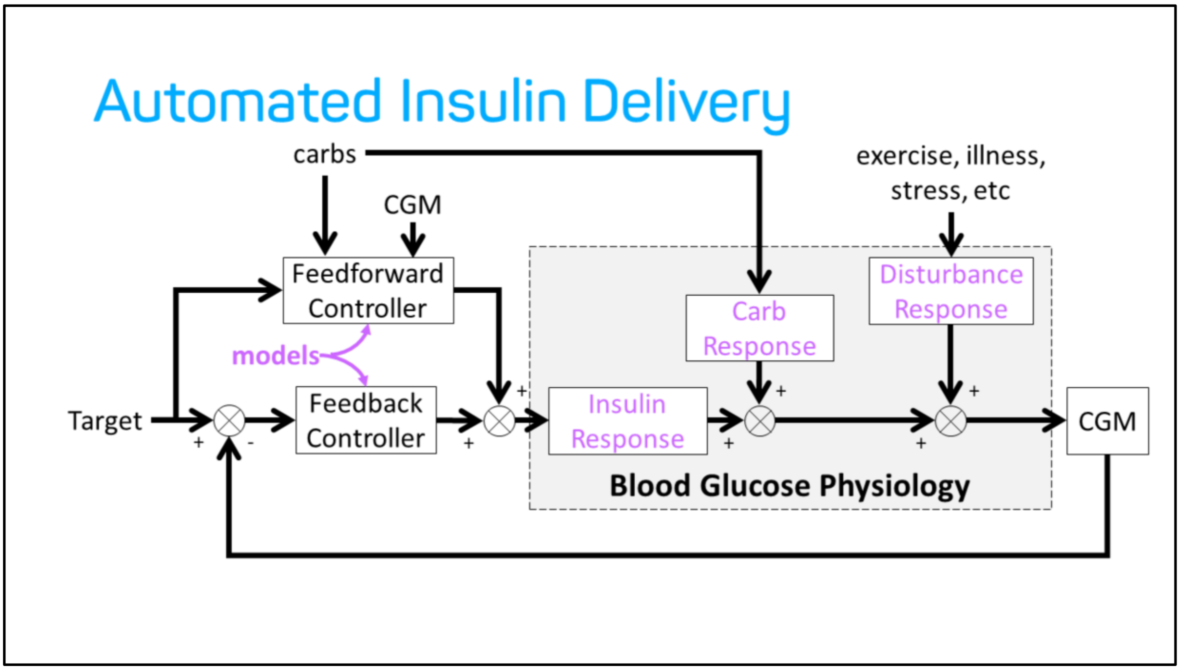
After nearly five years at Medtronic – the largest and most successful medical device company – I found myself frustrated by the pace of change. So I got together with a couple other d-dads and started Bigfoot Biomedical.

In just three years, Bigfoot has grown to seventy employees. We've raised about 90 million dollars. We built our first system and completed our first clinical trial 18 months ago. Now we're fleshing out that system to prepare for our Pivotal Trial.

Here are the main physical components of the system we are developing. On the left, we have an app which is the primary window for the user into the system. One of the big stigmas with diabetes is pulling out a syringe or a pump or a pen in public in order to interact with your disease. As a result, people will often not treat themselves in public settings – school, work, etc. And that's not good.

Our pump and continuous glucose monitor can be worn under clothes. The user interacts – gives commands, reviews status – from the app. As far as anyone else is concerned, they're just checking their email.

Another huge benefit of using the phone is that we can send things to the cloud, and we can receive software or firmware updates. No other system does this.



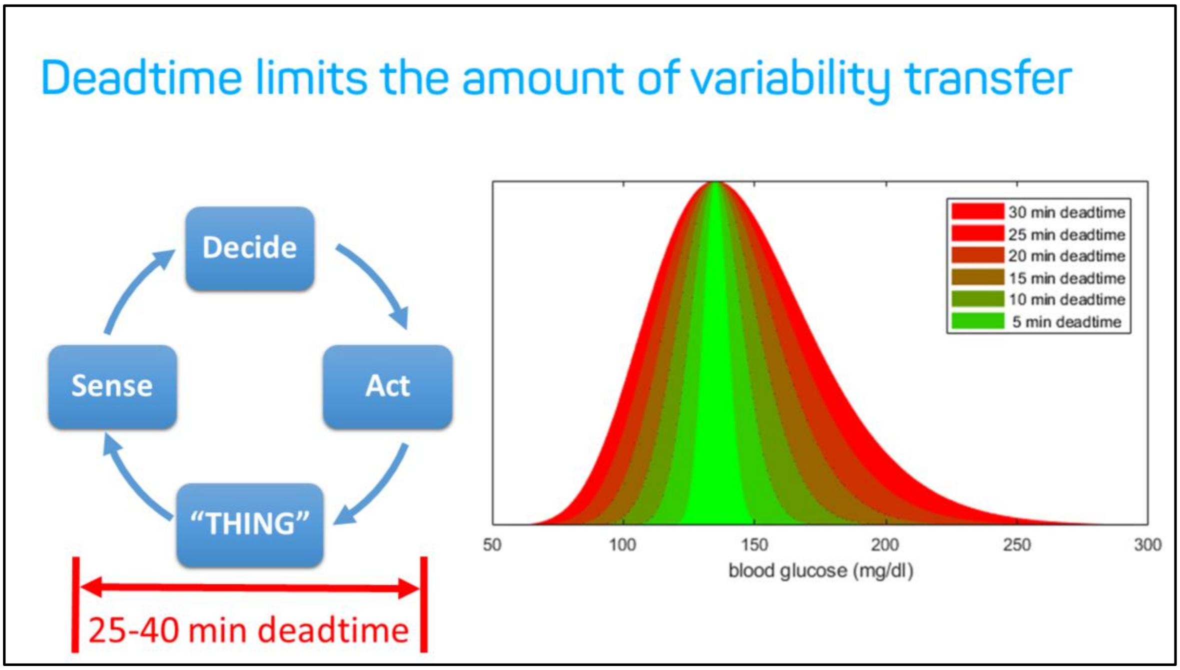
The basics of automated insulin delivery are shown in this block diagram. For reasons I'll explain shortly, best results are achieved through a combination of feedforward and feedback control. At mealtimes, the user "announces" to the system that they are eating a meal. Meal size is used to compute an appropriate amount of insulin to offset the glucose-raising effect of the meal. Then, the continuous glucose monitor (CGM) senses the blood glucose, and a feedback controller decides the appropriate profile of insulin delivery to return blood glucose to target.

Of course it is more complex than this but the basics are here. Now let's talk about some of the challenges.

Challenges

For 25 years people have been saying a "cure" for diabetes such as automated insulin delivery is only five years away. It's a receding horizon. Why haven't we closed the loop? In fact there are many challenges. I will now discuss just a few of them.

To start, you can't un-give insulin. If your system gives too much insulin, there is no automated way to overcome the effects; there is no counter-actuator to raise blood glucose. The human must intervene by consuming carbohydrates.

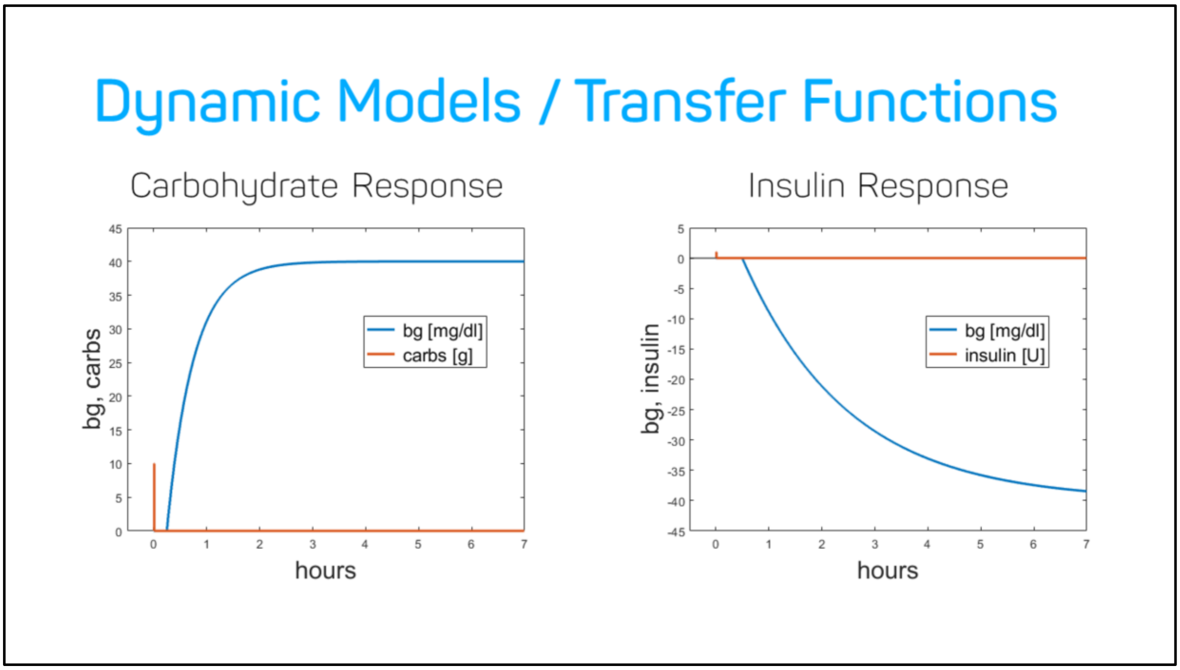


I learned 25 years ago that deadtime is the main factor which limits the achievable performance of control loop, in fact that's what my grad research was all about.

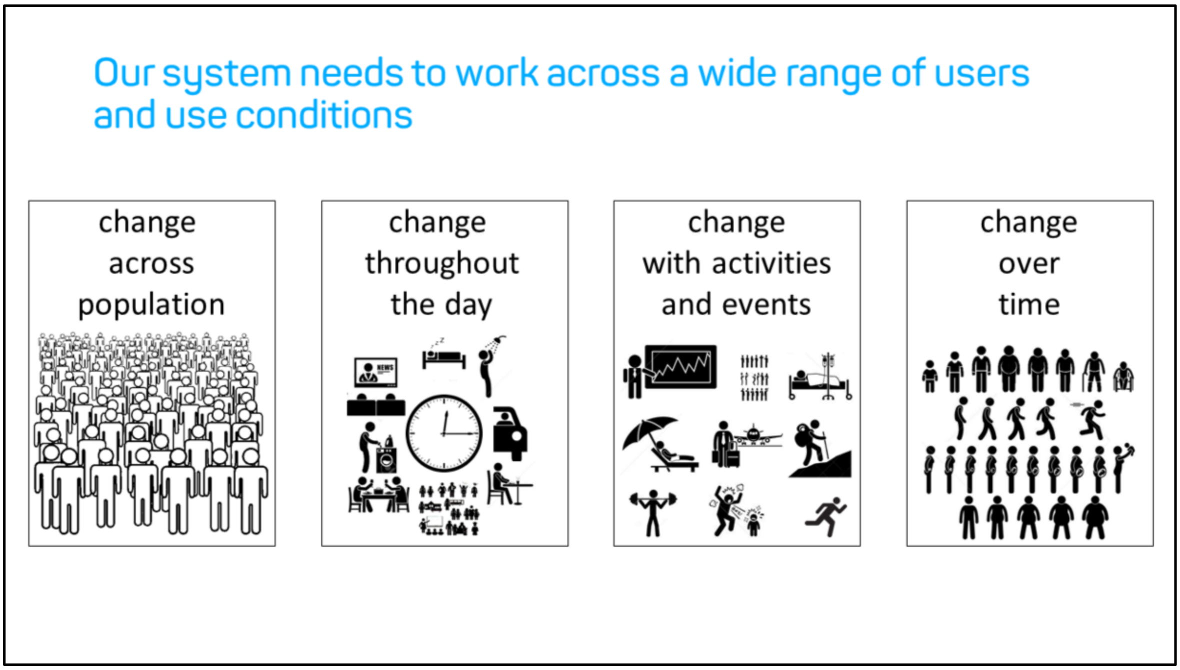
The time between when a change is made to the actuator and the effect of that change is sensed by the sensor is called the deadtime (or delay).

Unfortunately for automated insulin delivery, the subcutaneous delivery of "rapid" acting insulin has about half an hour of dead time between when it is injected, and when reaches the liver and starts to have an effect on glucose read by the CGM.

The plot on the right shows how much better control could be if we had a faster way to get insulin to the liver. In fact there is a faster way – inhalable insulin – but that can't be automated. So until such time as a faster insulin / faster pathway is developed, we will be constrained in how much variability we can transfer.



Here are the approximate dynamic responses of blood glucose to carbs and insulin, respectively. As you can see, the effect of carbs is much much faster than the effect of insulin. This is why feedforward control / meal announcement is so important for automated insulin delivery systems.

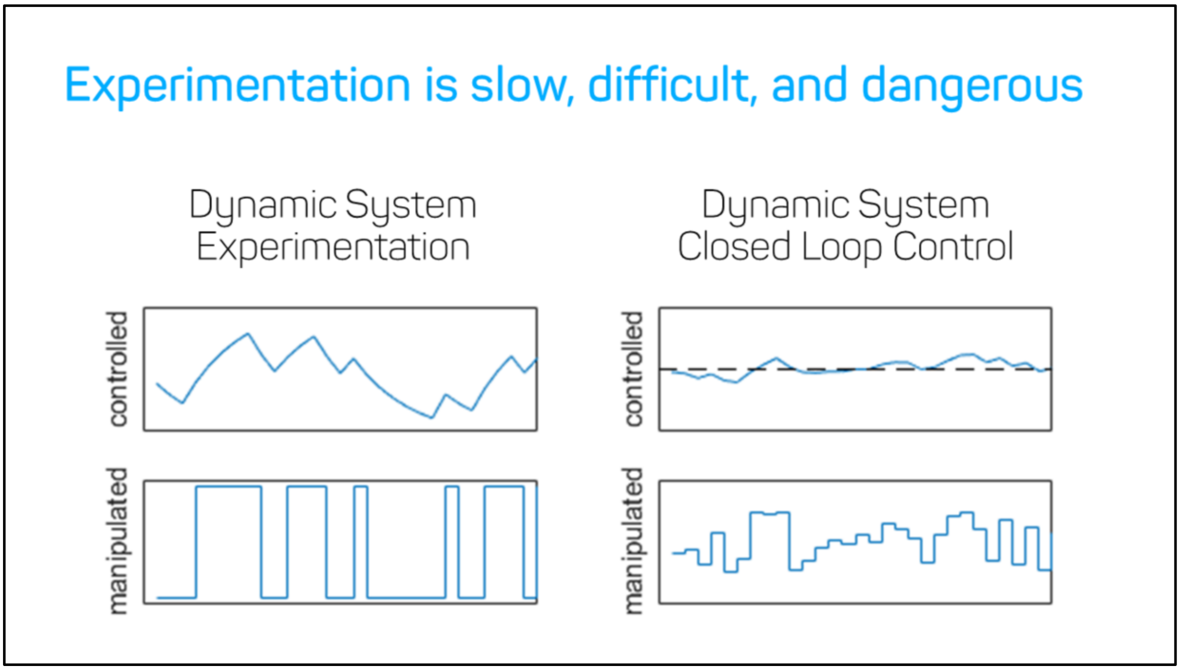


Unlike many control systems which get designed once and replicated a million times – like a hard drive head position controller – we are designing one controller that needs to work safely and effectively across a million “plants” – people with diabetes.

The system dynamics – especially the gains – vary by orders of magnitude across the population. They vary for each person through the course of the day. There are myriad activities and events which affect blood glucose – most of which we do not have sensors to measure.

Last but not least, people’s behavior and physiology change over time. Decide to run a marathon? Your insulin sensitivity will go down. Pregnant? Puberty? Sick? Your insulin sensitivity will go down.

At last count there are nearly fifty factors which contribute to blood glucose, and we can’t easily measure any of them.

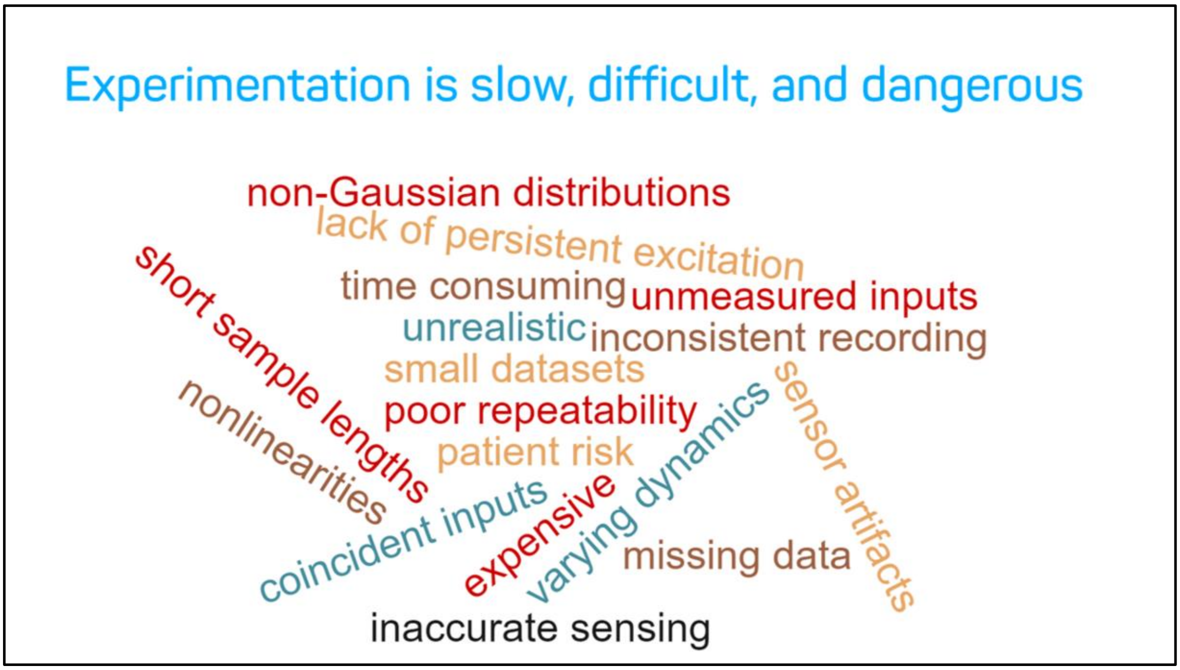


The traditional approach to developing a new controller is to perform designed experiments on the “plant” to develop models. At Honeywell, about half the million dollar budget for a typical model predictive control project was sending me out to the plant site to perform experiments. Unfortunately this is very difficult to do when the plant is a human.

The goal of experimental design for dynamic system identification is to introduce as much variation as you can, across the broadest possible range of manipulated variable actions, so as to maximally excite the controlled variable.

This is diametrically opposed to the goal of closed loop control, which is to draw as flat a line as possible in the controlled variable, i.e. keep it close to target by transferring variation to the manipulated variable.

As a result, experimentation for the purpose of dynamic system modeling is pretty challenging.



It is difficult, time consuming, expensive, and sometimes even unethical to experiment with human subjects. The data generated by these experiments is messy, hard to replicate, and has many other problems which result in being far from an optimally designed experiment. Which means the statistics coming out of these experiments often have very wide standard errors / the models are poor.

Temporal constraints

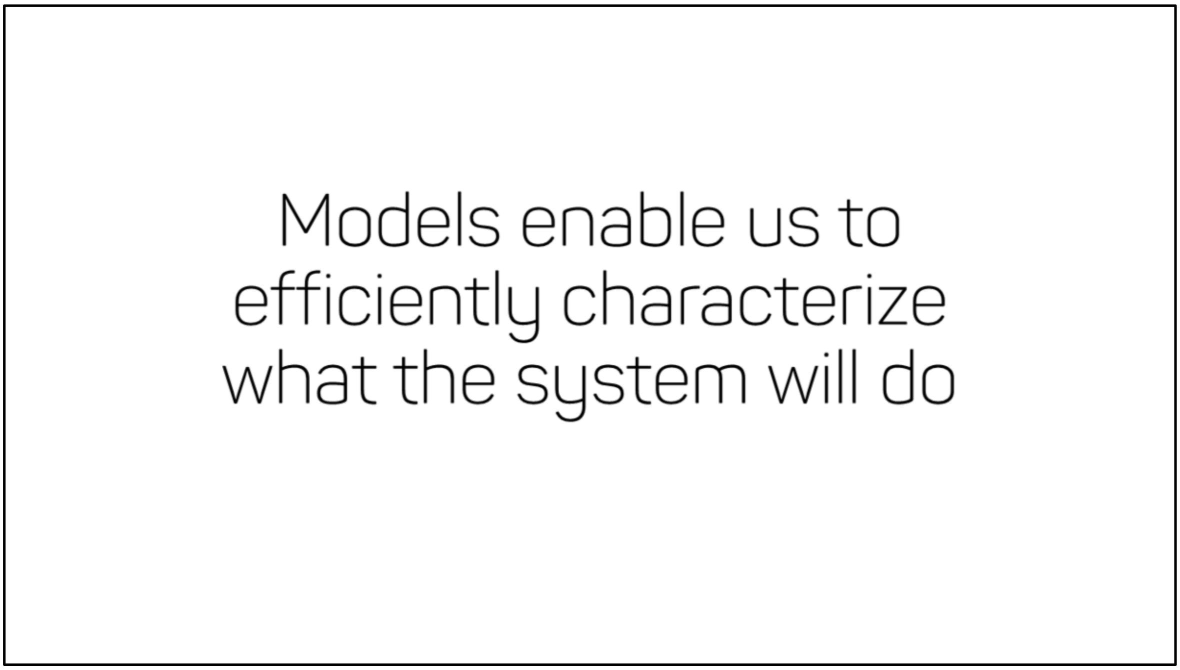
- There should not be prolonged periods of low blood glucose caused by overdelivery of insulin, which can lead to coma and death
- There should not be prolonged periods of complete absence of insulin, which can lead to DKA (diabetic ketoacidosis), high blood glucose, and death

It's like being in an airplane which can never land and needs to say between two flight levels. Too little insulin is bad, too much insulin is bad. This would be a straightforward control problem but for the challenges I've discussed. The "plant" is changing. The disturbances are unmeasured. The actuator is slow, one-way, and not 100% reliable. Many of the techniques you're learning in this course can be directly applied to designing and characterizing the behavior of an automated insulin delivery system.

Modeling and Simulation to the rescue

One of the benefits of working in so many different industries is that methods which are commonplace in one may be as yet untapped in another. I once told the president of Medtronic that working on automated insulin delivery was like getting in a time machine and going back 20 years in my career. The things I was doing 20 years ago hadn't even occurred to them to do yet.

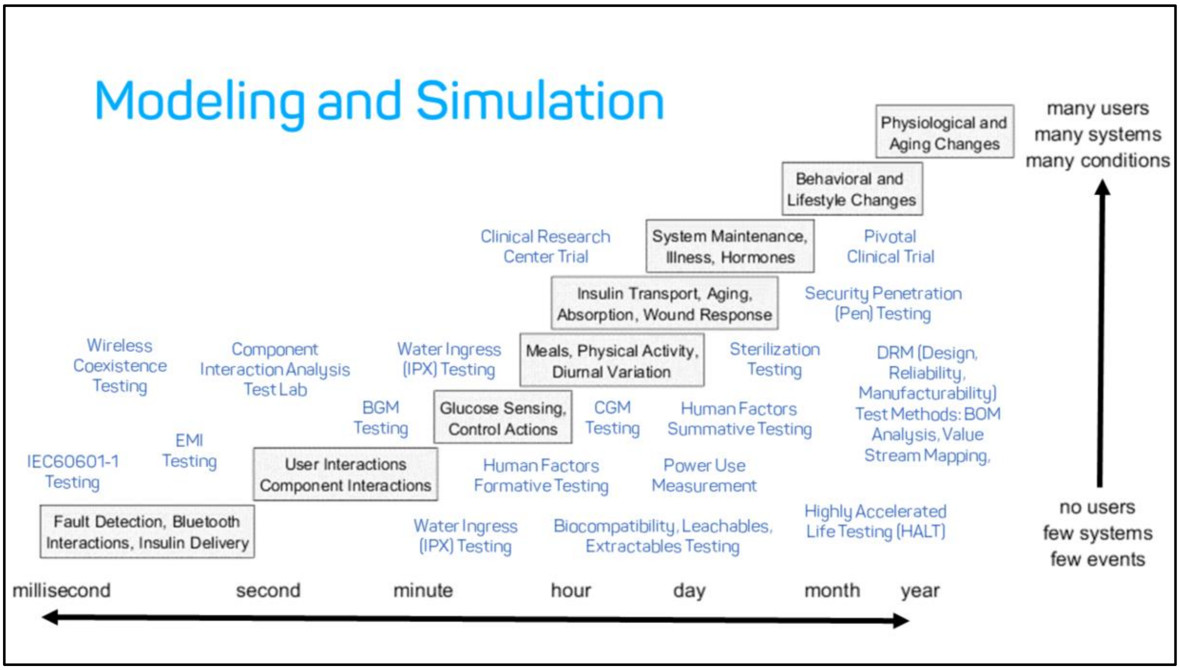
Once such method was the use of modeling and simulation. If one were building a new 5B$ ethylene plant, you wouldn't just show up in a farmer's field and start welding a bunch of pipe together. You would simulate it with a modeling tool like Aspen or Hysis. Same goes for a new aircraft, or a new car. Or, as it turns out, an automated insulin delivery system.



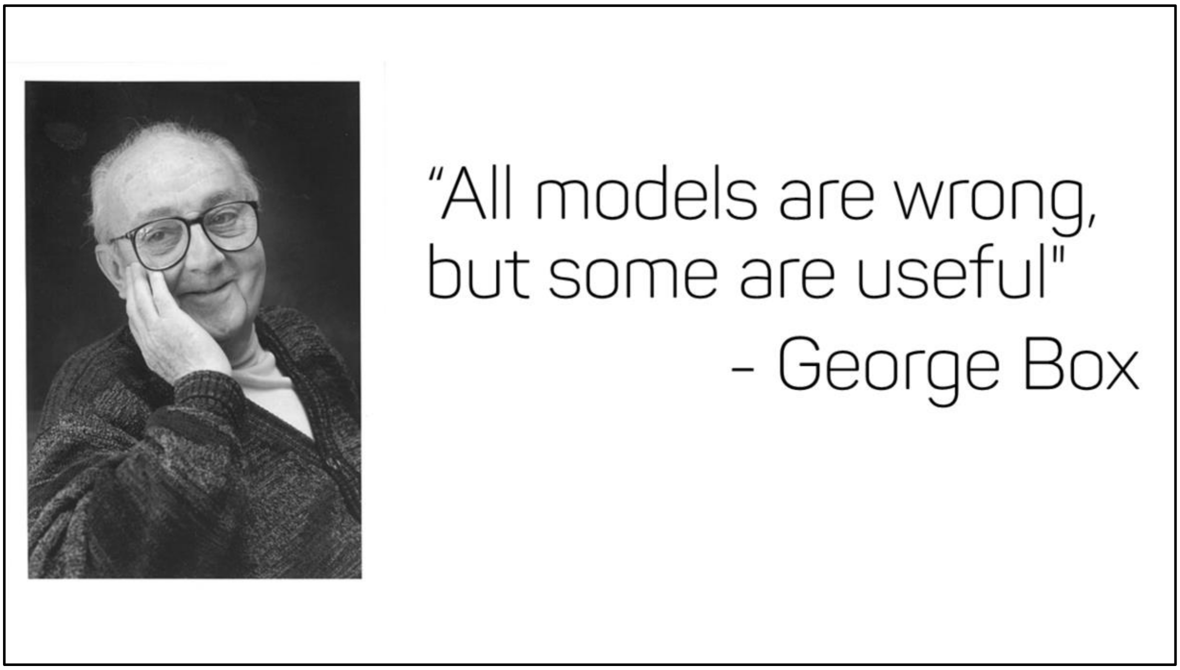
We can predict the behavior of a system before we have finished the system. How awesome is that?

Model (noun):
A simplified representation used to explain the workings of a real world system or event.

These models aren't just software models. Many types of simplified representations can be used to explain the workings of a real world thing.



All of these things are models. They are approximations of reality. They help us characterize the system we are developing, before it is released as a class 3 medical device used by thousands of people for many years. Even something like a standard "drop test" is an approximation of the reality of dropping the pump onto concrete. It is only an approximation of the use conditions / scenarios likely to be encountered in the real world. We have a particular challenge because ours is a multiscale system – things are happening over many time scales, from milliseconds to years. That makes characterization pretty difficult. Imagine if we had to use real-world testing to characterize our system – we'd never get to market. So instead, we use modeling and simulation.



Ever hear of this guy? He's the father of robust statistics and the guy who wrote the book on time series analysis, forecasting, and control. Literally. He's also my academic great grandfather. He was my grad school supervisor's supervisor's supervisor.

It's important to remember that models are a means to an end. They are not an end in and of themselves. Do only as much modeling as you need to solve the larger question at hand.

Modeling gotchas

- Purpose
  - Explanation vs. Prediction
  - Open- vs. Closed-loop
- Implementation
  - Overfitting
  - Unnecessary Complexity
- Don't forget
  - Stoichasticity
  - Sensitivity analysis
  - Verification and validation

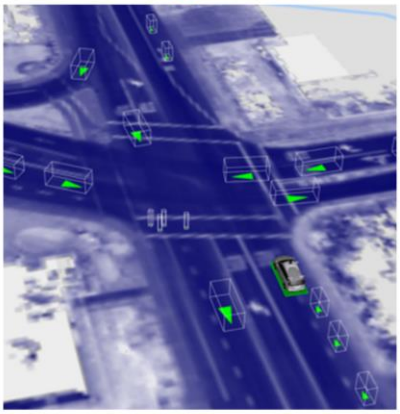
1. Law, Averill M., W. David Kelton, and W. David Kelton. *Simulation modeling and analysis*. Vol. 3. New York: McGraw-Hill, 2007.
2. Shmueli, Galit. "To explain or to predict?." *Statistical science*(2010): 289-310.
3. Saltelli, A., Tarantola, S., Campolongo, F., and Ratto, M. (2004). Sensitivity Analysis in Practice - A Guide to Assessing Scientific Models. Wiley.
4. MacGregor, John F., Thomas J. Harris, and J. D. Wright. "Duality between the control of processes subject to randomly occurring deterministic disturbances and ARIMA stochastic disturbances." *Technometrics* 26.4 (1984): 389-397.
5. Sterman, John D. "A skeptic's guide to computer models." *Managing a nation: The microcomputer software catalog* 2 (1991): 209-229.
6. Sargent, Robert G. "Verification and validation of simulation models."*Simulation Conference (WSC), Proceedings of the 2010 Winter*. IEEE, 2010.
7. Hjalmarsson, H. (2005). From experiment design to closed-loop control. Automatica, 41(3), 393-438.
8. Roy, C. J., & Oberkampf, W. L. (2011). A comprehensive framework for verification, validation, and uncertainty quantification in scientific computing. Computer methods in applied mechanics and engineering, 200(25), 2131-2144.
9. Pianosi, F., Beven, K., Freer, J., Hall, J. W., Rougier, J., Stephenson, D. B., & Wagener, T. (2016). Sensitivity analysis of environmental models: A systematic review with practical workflow. Environmental Modelling & Software, 79, 214-232.

It's important to note that modeling is not a silver bullet. In fact there are many mistakes which can be made which will cause the outputs of your modeling effort to be incorrect.

Many of the models you have been exposed to in academia are "toy" models to explain a concept. These should not be confused with the models required for making decisions with very large consequences in a commercial / industrial setting.

Modeling is one of the most difficult tasks I do. I've been doing it for 25 years, and I'm still learning. So please, approach your modeling efforts with caution and humility. And stand on the shoulders of giants. Check out the references I've listed here.

Modeling and Simulation

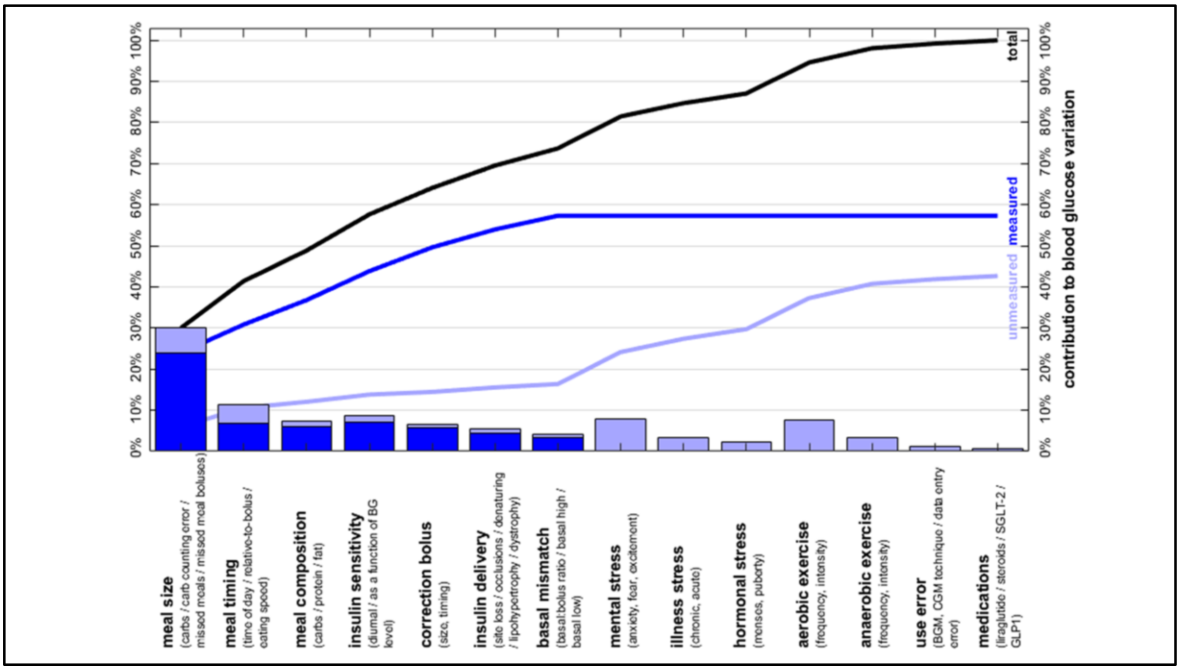


Alexis C. Madrigal, "Inside Waymo's Secret World for Training Self-Driving Cars", The Atlantic, Aug 23, 2017

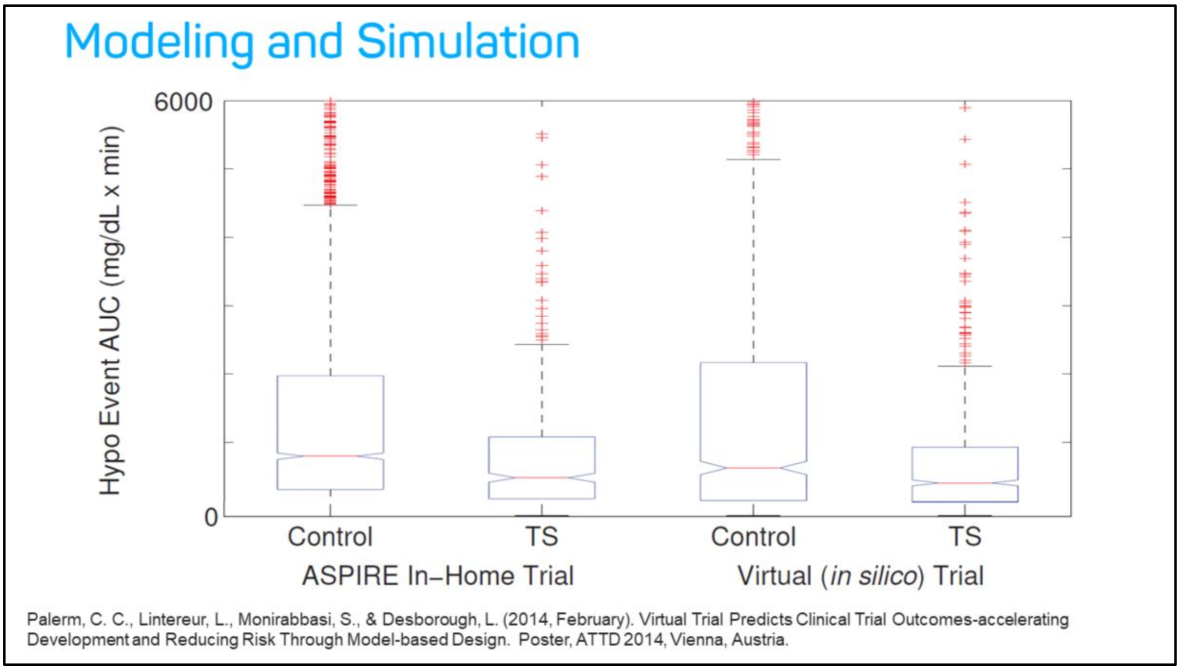
- Variance reduction techniques
- Monte Carlo simulation
- Population sampling
- Complex and challenging scenarios
  - Simultaneous faults
  - Risky behaviors
  - Challenging responses

Michael DeKort, "Autonomous Levels 4 and 5 will never be reached without Simulation", June 20, 2017

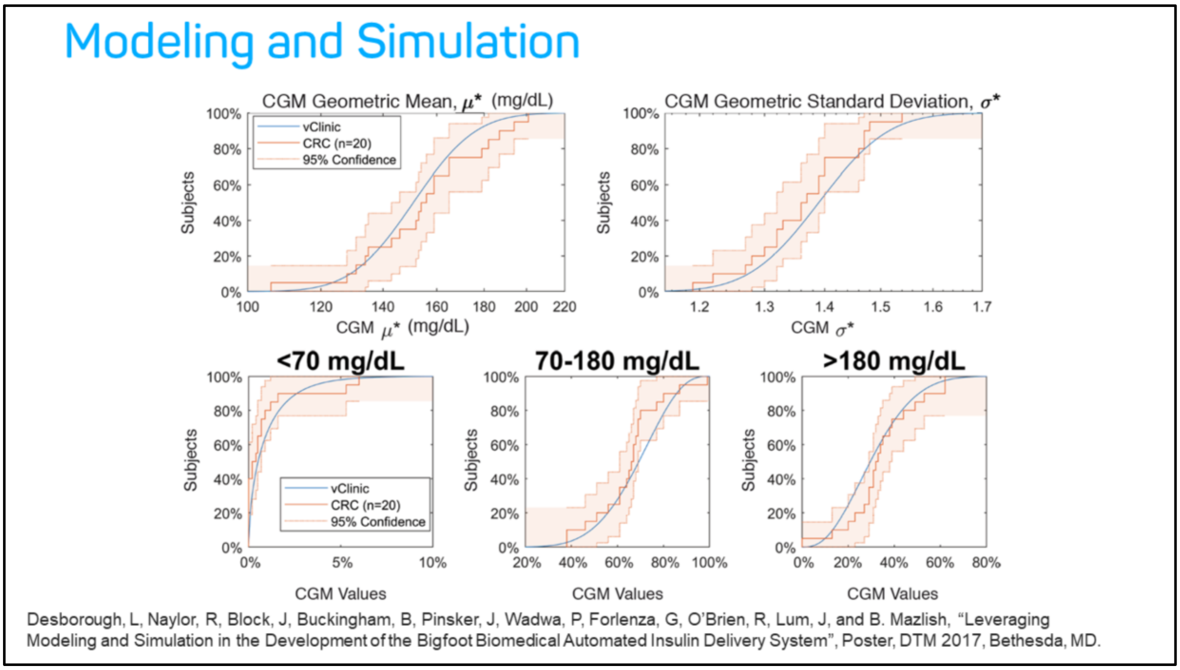
Autonomous vehicles have been making the news lately because of some unfortunate incidents involving "shadow driving"; collecting data in the real world. Companies like Toyota and Google / Waymo are now placing a heavier emphasis on modeling and simulation.



When it comes to automated insulin delivery, the myriad factors affecting blood glucose must be characterized. This way, we can design control algorithms which work across the wide spectrum of intended use scenarios. Here is a semi-quantitative assessment of the major factors affecting blood glucose. This is how we prioritize and focus our modeling and simulation efforts.



Modeling and simulation work. Here are results from some work we did at Medtronic demonstrating the strong concordance between the results of the "In-Home Trial" (left) and the Virtual Trial (right). What this doesn't show is that the former took years and millions of dollars, while the latter took days and a few pennies worth of electricity.



More recently, we showed the concordance between the predicted and actual results for Bigfoot's first clinical trial, which concluded 18 months ago.

Modeling and Simulation

With Modeling and Simulation:

- Rapidly evaluate multiple algorithm candidates and parameters
- Simulate performance of closed loop algorithms in a larger more varied population
- Inform design of clinical trial protocols, predict outcomes
- Predict performance over months or years of use

These are some of the benefits of modeling and simulation.

Modeling and Simulation

With Modeling and Simulation:

- Rapidly evaluate multiple algorithm candidates and parameters
- Simulate performance of closed loop algorithms in a larger more varied population
- Inform design of clinical trial protocols, predict outcomes
- Predict performance over months or years of use

Plus:

- Perform experiments in ways not possible or safe to do in in-vivo clinical trials
- No IRB or exclusion criteria are necessary
- No recruitment bias

But wait, there’s more!

Modeling and Simulation

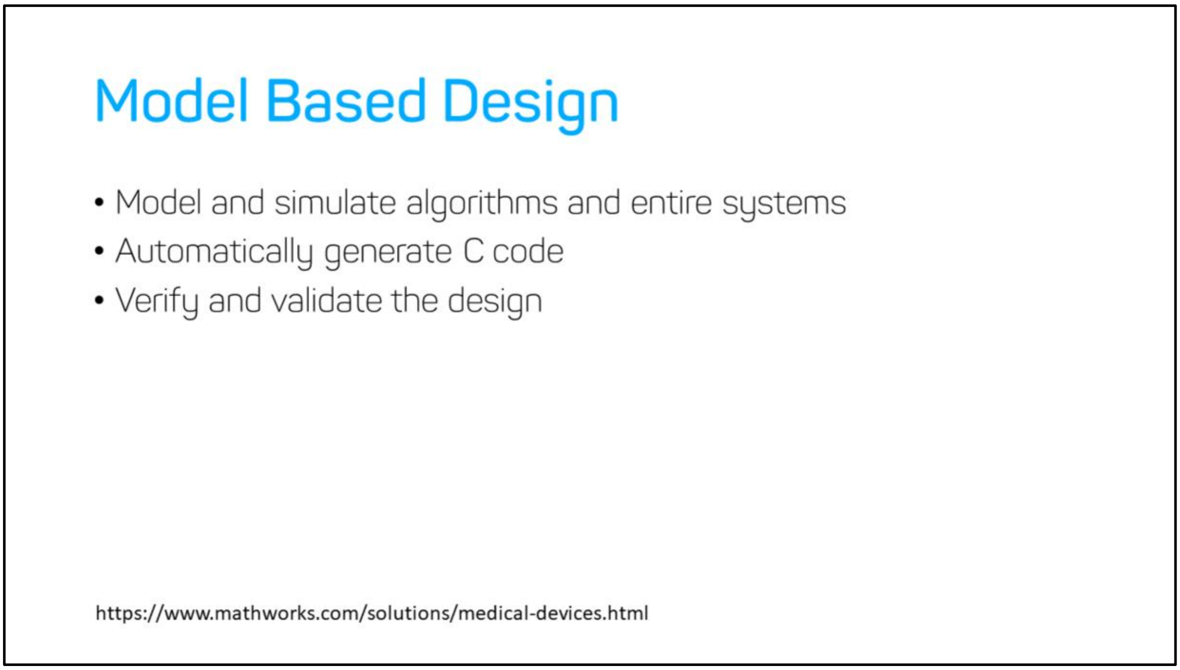
With Modeling and Simulation:

- Rapidly evaluate multiple algorithm candidates and parameters
- Simulate performance of closed loop algorithms in a larger more varied population
- Inform design of clinical trial protocols, predict outcomes
- Predict performance over months or years of use

Plus:

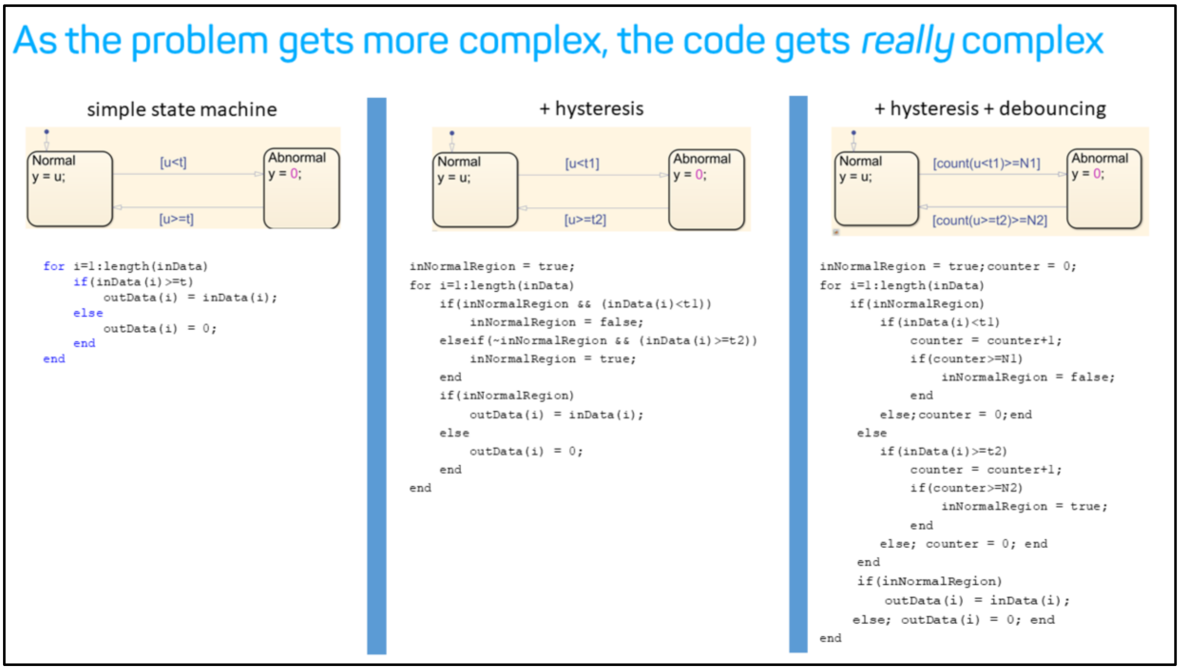
- Perform experiments in ways not possible or safe to do in in-vivo clinical trials
- No IRB or exclusion criteria are necessary
- No recruitment bias
- 4,000,000 times faster and less expensive than real-time (~1 cent per simulated contact-day vs. ~$1,500 per contact-hour)

Oh, and small detail, modeling and simulation is 4,000,000 faster and less expensive.



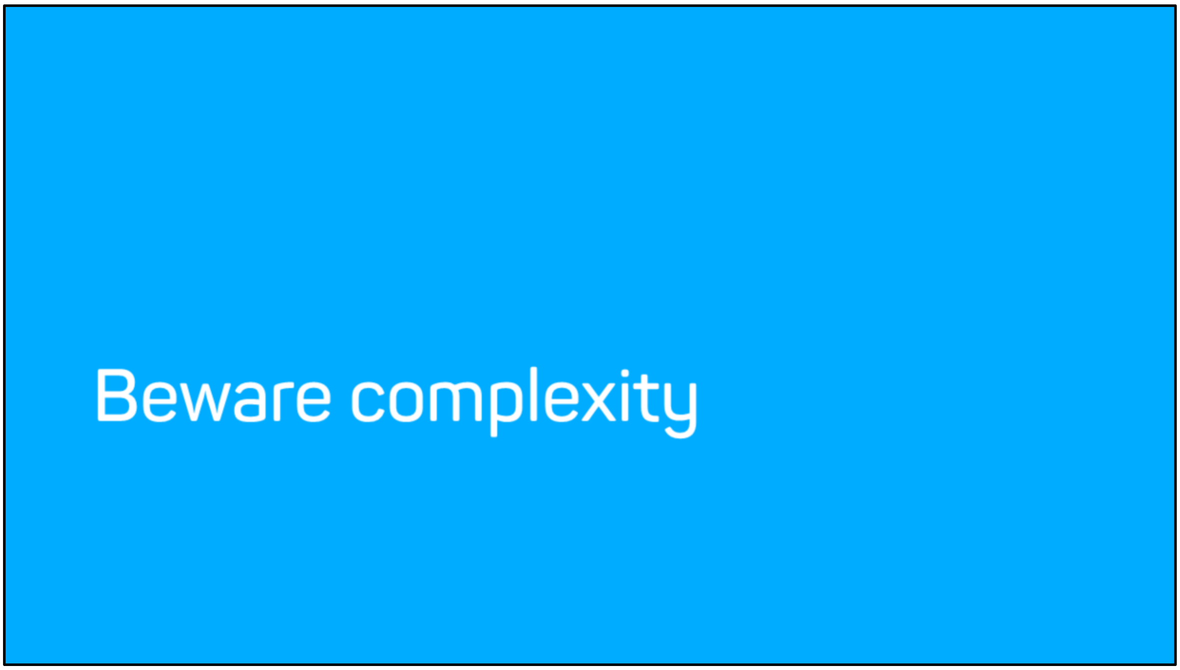
Finally, once the models have been developed and the control algorithms have been designed, they can be reused during the actual development. Using Model Based Design, we can take the Matlab / Simulink models and autogenerate C-code which gets compiled directly into the firmware of the device.

Other tools can be used to verify and validate the design.

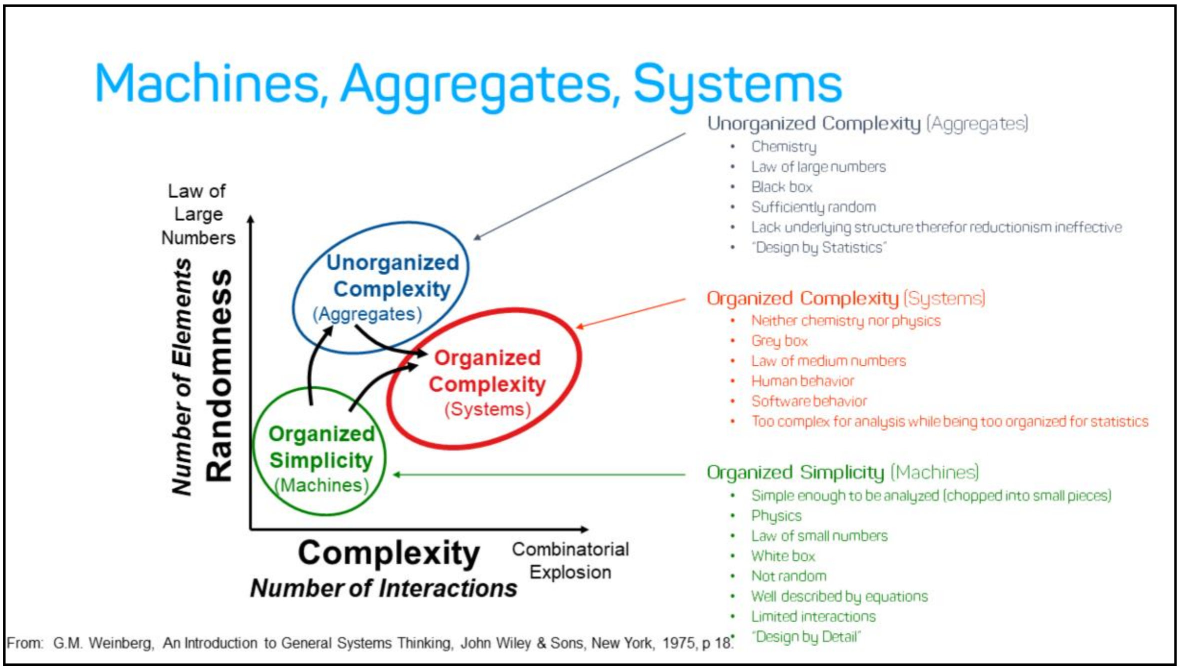


Model Based Design tools have other benefits. For instance Stateflow from Mathworks allows complexity to be abstracted in a visual representation. Imagine trying to debug a problem with the nested ifs of a complex state machine.

Bigfoot joins many other companies in embracing the benefits of Model Based Design.

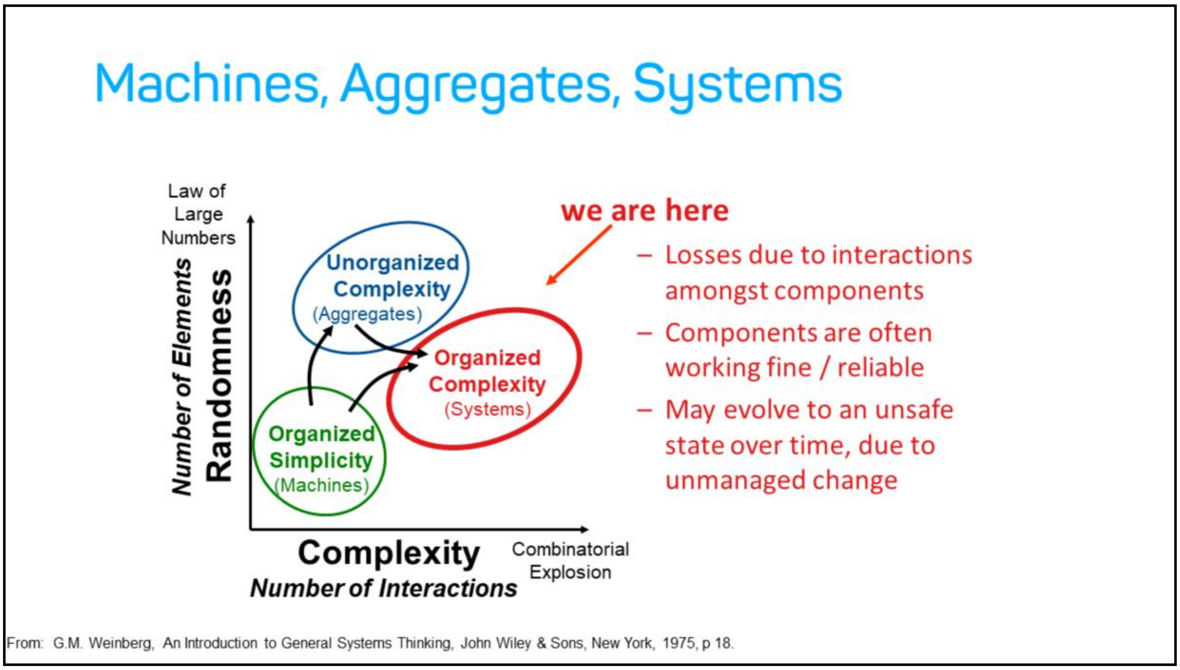


These systems are different from those which came before.



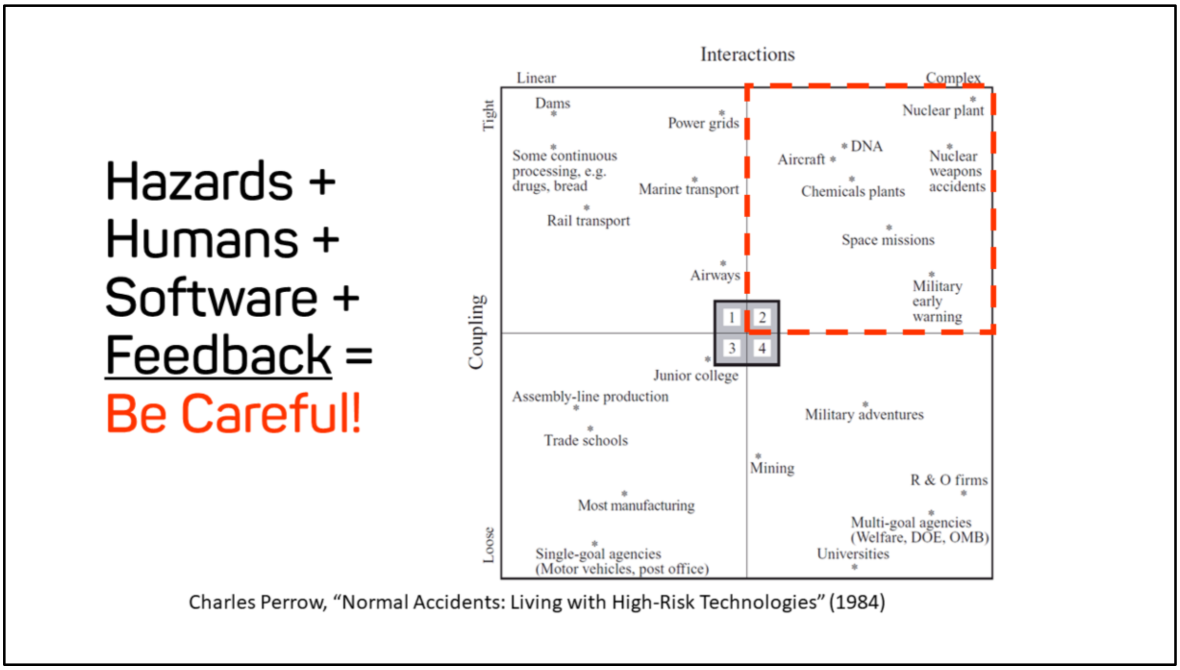
What do these systems have in common? They have humans and software. They are complex, with many interactions.

They are too organized for statistics and are not truly random. You can't examine a small part of the system and draw conclusions about the whole. They have feedback or component interactions which defy easy analysis.



These accidents are complex processes. Traditional event chain models cannot describe this process adequately.

Software is not bound by the constraints of the physical world. This is its biggest benefit and its biggest curse. Since it is a mental abstraction, it can do neither harm nor good until it is instantiated in a physical system with humans and hardware. Software is not predictable. It does not fail the same way hardware fails. It's not like the spring wears out in the for loop after a million cycles. Humans have certain biases, behaviors and cognitive limits which affect the systems they are a part of. For instance we constantly experiment with systems in order to understand their behavior. When this experimentation produces success, we ignore it. When there is an accident, we call it human error.



Charles Perrow, "Normal Accidents: Living with High-Risk Technologies" (1984)

The addition of feedback moves you into a new regime; a regime where the industries I've been discussing have a great deal of experience. I worked for GE Energy on the smart grid, integrating the entire North American electrical grid. In 2003, overgrown vegetation triggered a cascading set of failures amongst tightly integrated systems, blacking out over 50 million people in the northeast.

Like the other examples I've shared, there was not one "root cause". A software bug in the GE Energy Management System stalled the alarm system and deprived operators of important information.

It is so tempting to "blame the dead human" when losses occur but that's not the right approach. I am instead a huge fan of the work of Dr. Nancy Leveson at MIT. Her outstanding book "Engineering a Safer World" is available for free download.

This is a hospital infusion pump. Pretty simple, right?

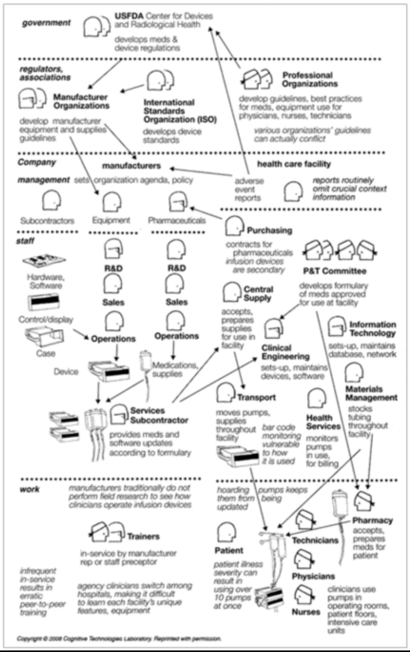
An infusion pump is a system

"An infusion "device," the most widely used information technology (IT) in health care, is actually an interdependent network of relationships. It's a socio-technical system that spans all who develop, supply, and use the result, from the level of the care provider or manufacturer organization, to associations and regulators, to government. This requires a different approach to safety: at the systems, not the device, level."

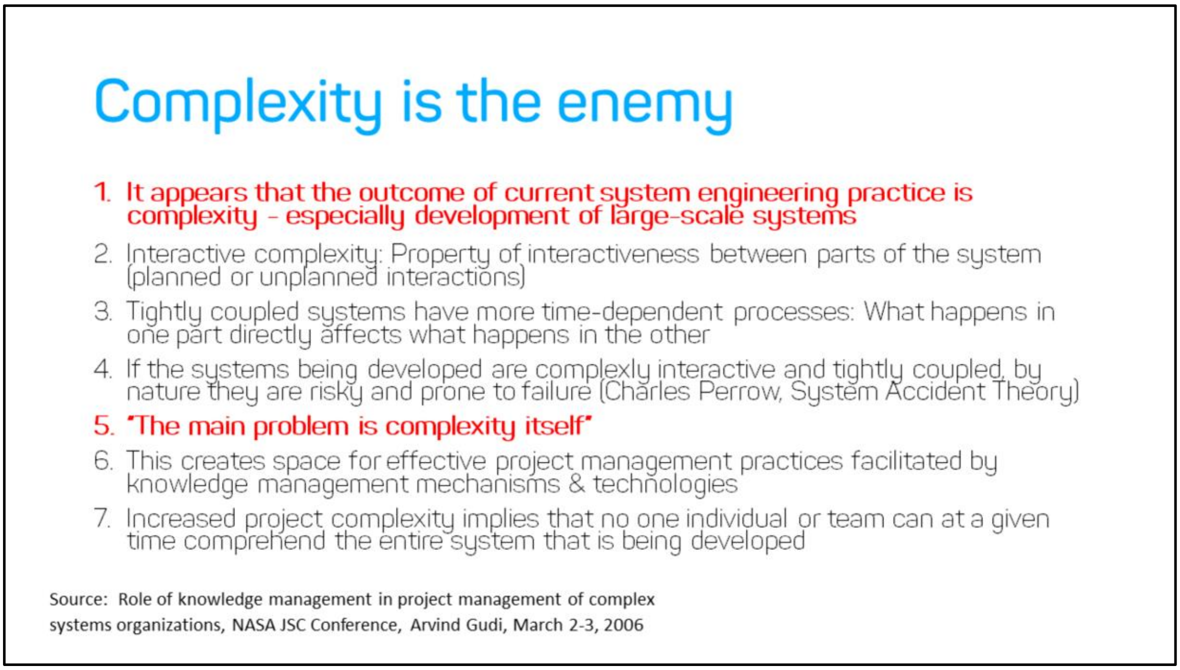
– Christopher Nemeth

Sources:

- Nemeth C, "The Safety of Medical Devices Perspective", http://www.webmm.ahrq.gov/perspective.aspx?perspectiveId=104
- Nemeth C, Cook R. The infusion device as a source of resilience. In Nemeth C, Hollnagel E, Dekker F, Dekker S, eds. Resilience Engineering Perspectives, 2. Farnham, UK: Ashgate Publishing; 2009. ISBN: 9780754675204. Preparation and Restoration; vol 2.



Not really. Even a "simple" infusion pump is just one part of an enormously complex system. Know your system boundaries. Consider the emergent properties.

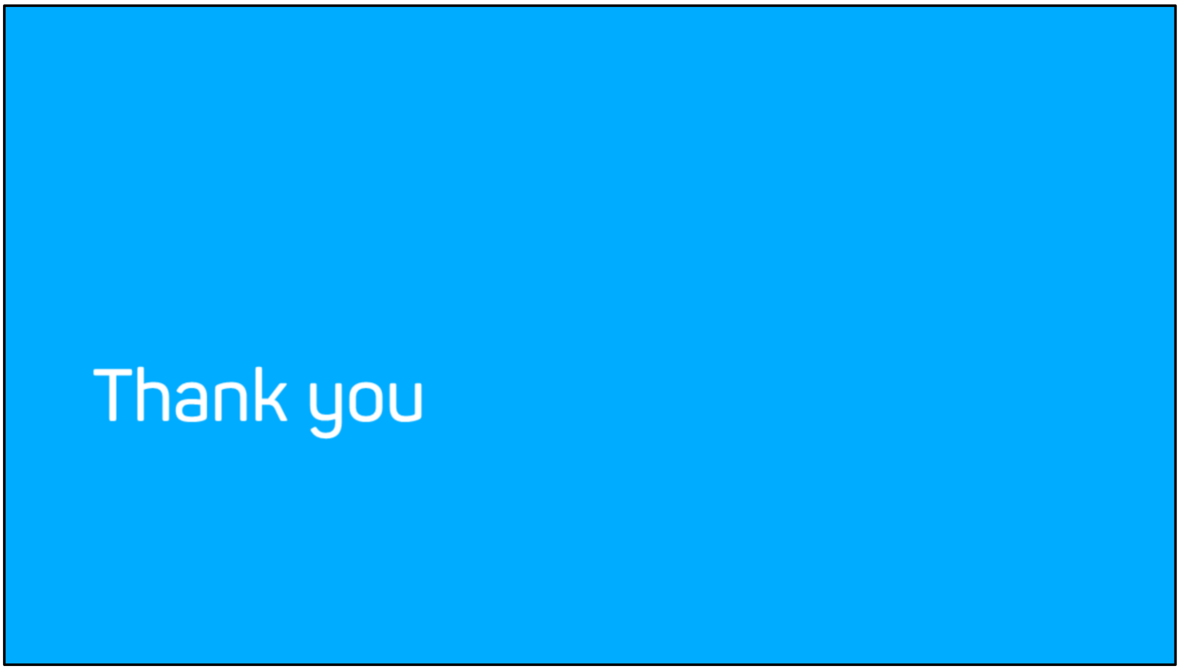


I can't emphasize enough the perils of complexity. There is the inherent or domain complexity of the problem you are trying to solve. Can't do anything about that. What you can control is the accidental complexity – the unnecessary complexity added as you attempted to solve the problem. At every step consider the complexity you're adding and ask "is this really necessary"? Because if it's not, you're just adding technical debt. And like all debt, if too much is incurred then you eventually go bankrupt.

"The future is already here, it just hasn't been evenly distributed yet" – William Gibson

This quote has been attributed to William Gibson, author of Neuromancer and coiner of the phrase "cyberspace". If there's one thing I've learned, it's that I have no original thoughts. How arrogant would I have to be to think that out of nearly 8 billion people on the planet, I was the first to think a particular thought.

So I don't even try. I use Google. I assume that somebody, somewhere, perhaps in a different industry or academic discipline has solved a similar problem. I seek out people from different companies. That's how I found Jyo. That's why I'm here today.



Thank you.

Notes

- Best way to contact me: Ldesborough@bigfootbiomedical.com
- More about designing safe cyberphysical systems
  - https://aamiblog.org/2013/07/30/lane-desborough-the-value-of-simplicity-in-a-complex-world/
  - http://psas.scripts.mit.edu/home/get_pdf.php?name=2-1-Desborough-Using-STPA-in-the-Development-of-an-Artificial-Pancreas.pdf
- More about Bigfoot:
  - https://www.mathworks.com/videos/developing-an-artificial-pancreas-using-model-based-design-1481554386617.html slides and video
  - https://www.bigfootbiomedical.com/vision/
  - https://www.bigfootbiomedical.com/podcasts/
  - https://www.bigfootbiomedical.com/news/

References

- The Systems Bible - Gall
- Engineering a Safer World - Leveson
- Product Development Flow – Reinertsen
- Our Robots, Ourselves - Mindell
- A list of my books (200+), rated: https://www.goodreads.com/review/list/1114014

A model is not reality. - Rechtin's heuristics for system architecting

Complex systems exhibit unexpected behavior. - The Systems Bible

Every once in a while you have to go back and see what the real world is telling you. [Harry Hillaker, 1993] - Rechtin's heuristics for system architecting

Overengineering: Spending resources making a project more robust and complex than is needed

Just because it worked in the past there's no guarantee that it will work now or in the future. (Kenneth L. Cureton, 1991) - Rechtin's heuristics for system architecting

Reality is more complex than it seems. - The Systems Bible

The Kantian Hypothesis (Know-Nothing Theorem): Large complex systems are beyond human capacity to evaluate. - The Systems Bible

Magic numbers: Including unexplained numbers in algorithms

Choose the elements so that they are as independent as possible; that is, elements with low external complexity (low coupling) and high internal complexity (high cohesion). - Rechtin's heuristics for system architecting

Don’t assume that the original statement of the problem is necessarily the best, or even the right, one.   - Rechtin's heuristics for system architecting

New systems mean new problems. - The Systems Bible

In order to understand anything, you must not try to understand everything. (Aristotle, 4th cent. B.C.) - Rechtin's heuristics for system architecting

Errors are most frequent during the requirements and design activities and are the more expensive the later they are removed. - Boehm' Law

Plan to scrap the first system: You will anyway. - The Systems Bible

If you can’t explain it in five minutes, either you don’t understand it or it doesn’t work. (Darcy McGinn, 1992 from David Jones) - Rechtin's heuristics for system architecting

Individual developer performance varies considerably. - Sackman' Law

The Fundamental Theorem: New systems generate new problems.
- The Systems Bible

The total behavior of large systems cannot be predicted. - The Systems Bible

Premature optimization: Coding early-on for perceived efficiency, sacrificing good design, maintainability, and sometimes even real-world efficiency

Simplify, combine, and eliminate. (Suzaki, 1987) - Rechtin's heuristics for system architecting

Big ball of mud: A system with no recognizable structure

Kaplan's Law of the Instrument - Give a small boy a hammer and he will find that everything he encounters needs pounding.

"A complex system that works is invariably found to have evolved from a simple system that works"
– John Gaule

Everything is a system. - The Systems Bible

A complex system designed from scratch never works and cannot be patched up to make it work. You have to start over, beginning with a working simple system. - The Systems Bible

"The future is already here. It just hasn't been evenly distributed yet" - William Gibson

A structure is stable if cohesion is strong and coupling low. - Constantine' Law

The first line of defense against complexity is simplicity of design.
- Rechtin's heuristics for system architecting

A system is no better than its sensory organs. - The Systems Bible

Unbounded limits on element behavior may be a trap in unexpected scenarios. [Bernard Kuchta, 1989] - Rechtin's heuristics for system architecting

Build reality checks into model-driven development. [Larry Dumas, 1989] - Rechtin's heuristics for system architecting

Simplify. Simplify. Simplify. - Rechtin's heuristics for system architecting

High quality, reliable systems are produced by high quality architecting, engineering, design, and manufacture, not by inspection, test, and rework. - Rechtin's heuristics for system architecting

An evolving system increases its complexity, unless work is done to reduce it. - Lehman' Law

If things are acting very strangely, consider that you may be in a feedback situation. - The Systems Bible

Complicated systems produce complicated responses to problems. - The Systems Bible

Relationships among the elements are what give systems their added value. - Rechtin's heuristics for system architecting

"All other things being equal, the simplest solution is the best." - Occam's razor

If the politics don't fly, the hardware never will. (Brenda Forman, 1990) - Rechtin's heuristics for system architecting

Technical problems become political problems. - Rechtin's heuristics for system architecting

If you can’t analyze it, don’t build it. - Rechtin's heuristics for system architecting

Stockdale Paradox: "You must never confuse faith that you will prevail in the end – which you can never afford to lose – with the discipline to confront the most brutal facts of your current reality, whatever they might be."

In introducing technological and social change, how you do it is often more important than what you do.   - Rechtin's heuristics for system architecting

Lava flow: Retaining undesirable (redundant or low-quality) code because removing it is too expensive or has unpredictable consequences[5][6]

"Simplicity is an acquired taste. Mankind, left free, instinctively complicates life." - Katherine F. Gerould

Hierarchical structures reduce complexity. - Simon' Law

Connect-the-Dots Principle: There must be a traceable connection from business strategy to each enterprise architecture decision. - Dana Bredemeyer

Accidental complexity:
Introducing unnecessary
complexity into a solution

The value of models depends on the view taken, but none is best for all purposes. - Davis' Law

Without data, discussions produce more heat than light - Edwards Deming

Performance, cost, and schedule cannot be specified independently. At least one of the three must depend on the others.
- Rechtin's heuristics for system architecting

A combination of different V&V methods outperforms any single method alone. - Hetzel–Myers' Law

Minimalist Architecture Principle: Make only those decisions that have to be made at this level of scope to achieve the business strategy and meet the architecture objectives and vision.
- Dana Bredemeyer

Decisions With Teeth Principle: Only include decisions in your Enterprise Architecture that you, and the governance organization, are willing and able to enforce. - Dana Bredemeyer

In architecting a new [software] program all the serious mistakes are made in the first day. [Spinrad, 1988]   - Rechtin's heuristics for system architecting

Development effort is a (non-linear) function of product size. - Boehm' Law

"Simplicity is the ultimate sophistication" - Leonardo da Vinci

Choose a configuration with minimal communications between the subsystems. (computer networks) - Rechtin's heuristics for system architecting

"Any intelligent fool can make things bigger, more complex, and more violent. It takes a touch of genius -- and a lot of courage -- to move in the opposite direction."
- Albert Einstein

Sinclair’s Law - If a man's paycheck depends on his not understanding something, you can rely upon his not understanding it

Requirement deficiencies are the prime source of project failures. - Glass' Law

"If you don't have time to do it right, when will you have time to do it over?" – John Wooden