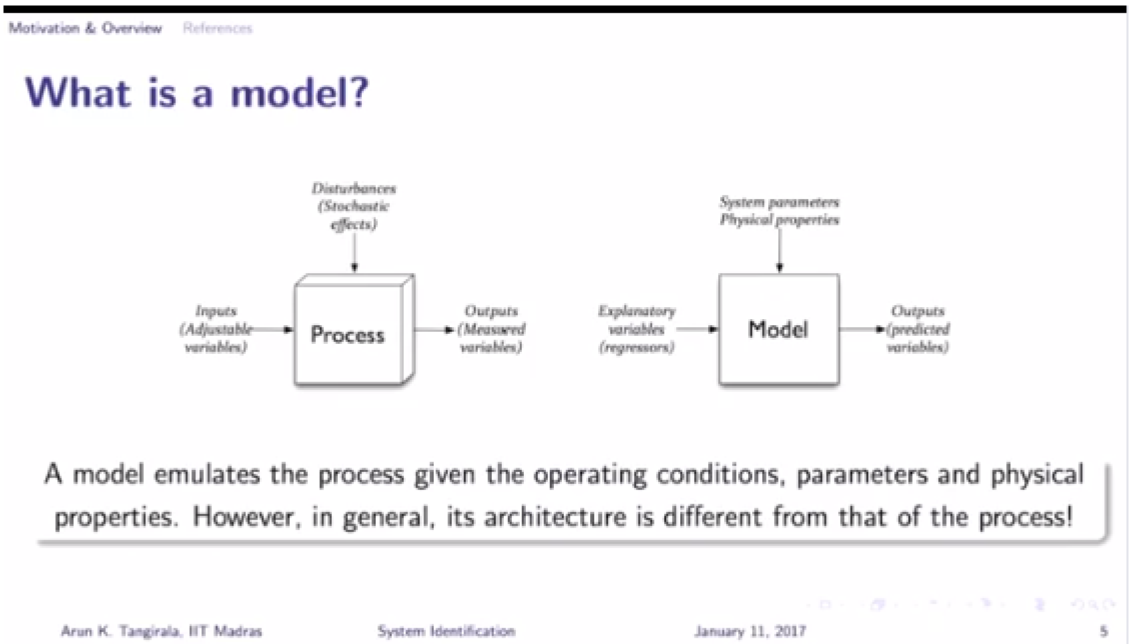
CH5230: System Identification

Motivation and Overview 2

Earlier we have talked about a model, it's a mathematical abstraction of a process. But I want to highlight a few other aspects of what a model is because many among us is not clear as to what a model is when you compare it with a process. So I have given you -- I'm showing you the schematic of a generic process and a generic model. If you look at this carefully, the process receives some physical inputs and gives out some response which we call as outputs and in this course of response it also experiences disturbances, right?

(Refer Slide Time: 1:00)



So, if you think of the air conditioner, there are inputs, so there is a coolant flow that's actually being manipulated and then the temperature is the response, you can say that's the output of interest. But the air conditioner also experiences disturbances when somebody opens a door or the ambient temperature is changing and so on. So those are the kinds of disturbances that the process can experience. Now, if I were to build a model for the air conditioner, what kind of a model would I build? Or what would be the inputs to the model? What to expect on the model? Again, the input to the model would be the same coolant flow and the model is supposed to predict the room temperature, right? So in this -- in that example, the inputs to the process and inputs to the model are same. However, there are other things that I have to feed to the model which are the system parameters sometimes.

If it's a first principles model, I have to supply the physical properties, the system parameters, the size of the room and so on. Those are considered from an algorithmic viewpoint, from a computational viewpoint, these are the inputs that the user has to supply to the model, so that the final prediction can be computed. So in general -- and let me now -- and in this example that we just discussed, the inputs to the process and the model seem to be identical, but they are not. Why? Let's take that a bit more in depth. We just said that if I take an air conditioner, the input that I'm interested in is the coolant flow, because I have to adjust it and the output is the room temperature, right? That's a process that I'm looking at. Great. But when I build a model, typically, I would like to build what is known as a dynamic model. What's a dynamic model? What's the difference between a steady state and a dynamic model? I'm hearing some standard answers which are not right. So what's the difference between a steady state model and a dynamic model? Any idea?

In a dynamic model is you're given the state that any instant you can find out for a state at any other instant in time.

Okay. What would be the steady state model? Steady state model also I can compute that, right? No? And what you've just earlier said for dynamic model, that's also not completely right. What distinguishes between a steady state and a dynamic model?

It takes in the transient characteristics of the [3:45 inaudible], the dynamic model.

Okay, so it tries to explain the transient -- what do we mean by transient characteristics? Tell me, explain what is meant by transient characteristics.

How does the rate of change of the instant [4:10 inaudible] something like that.

Okay. And how is it different from a steady state model? So I'll write two equations and you have to tell me because models after all going to be mathematical for us. I'll write two different model equations and you can, you have to tell me which one corresponds to transient and which one corresponds to a static one. So let's denote the input by u and output by y. This is a notation that any way would follow in this course. I thought I'll not write equations but nevertheless this should not be counted as an equation. So let's say I'm writing a discrete time model because these are the kind of models that we are going to develop unlike continuous time models that you're used to. This is a K here, corresponds to an instant in time. And it corresponds to a specific sampling instant in time. And I'll explain the notation a bit more in detail later on. So you understand this is one equation that relates the input to the output. A1 and b1 are some model parameters. And then, I have another equation, let us say, then I'm going to write one more because two you may be able to answer easily, three, we'll see if you're able to answer this. Once you've answered all these three, I'll write one more. Anyway, so now you can tell me which of this equations here, so let's call this equation one, model two and model three. Which of this you would call as a static model?

Second.

Second one. Unanimously. Is there a doubt? Why? Why does it qualify to be called a steady state model? Any idea?

System has settled. It doesn't require any, I mean, whatever past has given it doesn't affect have the present.

Okay. Any other reason why it's called a steady state model or the static mode? To the response of the system, as per equation two, it says the response to the system is only dependent on the present conditions. Okay. In modeling terminology, you can say it's a memory less kind of system. It doesn't really depend what has happened in the past has no bearing on the response of the system, okay. And you would develop these kinds of models for processes that have reached steady state, obviously. Which one would you call as a dynamic model?

First one.

First one. Why? Why would you call that? When I -- if you look at the equation carefully, when a change in input occurs, the output starts to change, does it change immediately?

Delayed.

There's a delay, right? Good. But after the delay, it starts to change, but it continues to change even though the input is held fixed, right? Here if I hold the input, does output change? No. Whereas here, when I give -- introduce a change and input and hold it, the output continues to change for a while and then settles down, depending on the value of a1, right? But then, we say that that phase in which the output is changing, the response is changing until it reaches, if there is a steady state, we call that as a transient phase. So this model is there for explaining what happens between two changes in the input, right? So, therefore, we call that as a dynamic. So, a dynamic model explains what happens between two changes in an input or one, when the input goes from one steady state to another steady state. Whereas, the second one only tells me what happens at steady state. That's it. It is it has no information on what happens in between. What did this poor thing that you've not quailed it to be one, either static or dynamic? What's wrong with that third one? Is there something wrong? Isn't -- it's neither a steady state not a dynamic model.

Steady state.

But you didn't say, you just said only two when I asked for steady state. You left it out. Are you convinced that the third one is a steady state model or you have some element of doubt? It is also a steady state model, but with a delay. Once the delay has elapsed, it's at steady state, that's all. So it's called a static plus delay model. Whereas model two is a static model and model one is a dynamic model, of course, there is a delay that that doesn't matter. It is still a dynamic model. Now as I promised, I said I'll write a fourth one. What do you think of this model? What do you think? Is it a steady state model or a dynamic model?

Steady state.

Steady state? Seriously? Why? Why do you call it as a steady state model? It doesn't tell you what happens between two changes in the input. Do you want to revise your answer? Come on, man, it's not yet lunchtime. Quickly. Few seconds. Do you want to revise your, nobody wants to revise it? You're all sure? [Pakka] [11:28 inaudible]? It's steady state model. I was hoping at least one would revise the answer. What do you think?

Steady state.

Steady state. Why?

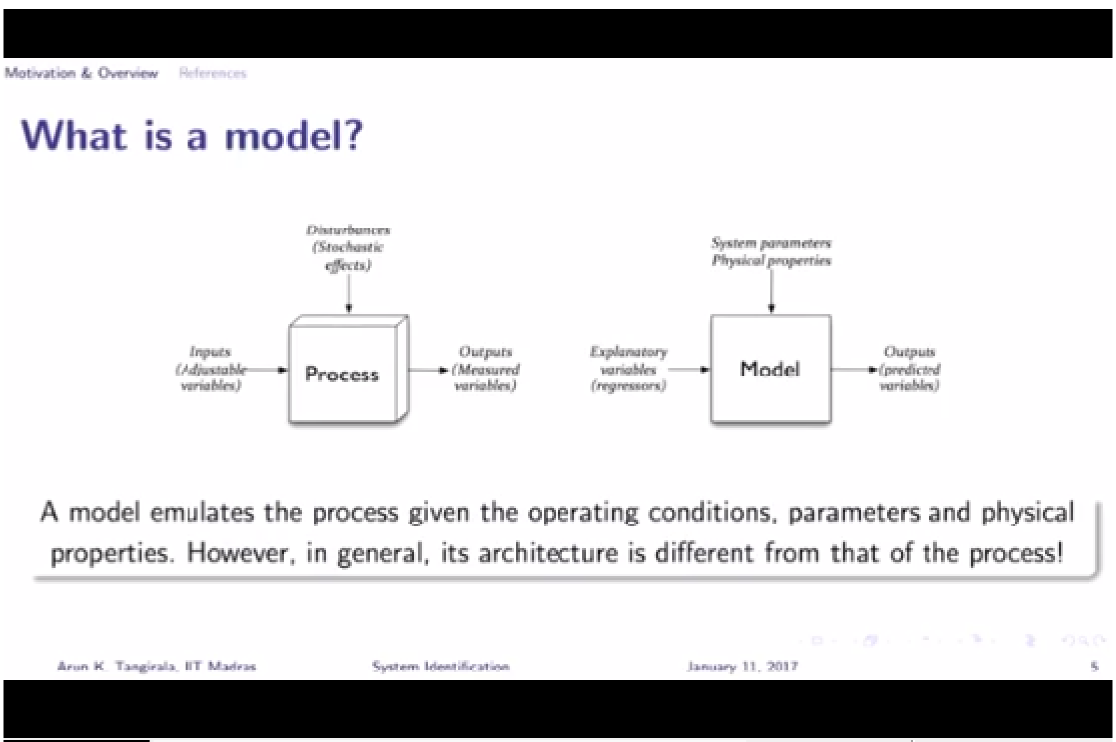
Because the--

Whatever is the input, the output is fixed. I mean, it won't change.

Why? There is a past input, right? So just to make things more clear, let us say, I mean, simpler for you, so that don't get entangled in these delays. So let's say introduce a change in the input. You don't see the output changing? It just reaches a steady state instantly. If you were to simulate this model and as a user you have to supply the input, right? Let's say the system is at steady state. Now you introduce a change in the input and hold it fixed. How do you think the model -- output would look like, the response would look like? Will it reach a steady state? It won't change or not? It will change. This is a kind of dynamic model as well. This is a kind of dynamic model. We are used to only looking at these kinds of models, right? Whereas here, this is a different kind of dynamics.

In fact, later on we'll learn that we'll be able to write this model in this form, maybe of on a higher order but you should, you should be able to cast, recast that model. All you have to ask yourself is, whether this model is capable of explaining or describing the transients when I change the input, from one steady state to another state. That we call as a step change. You just introduce a change and hold it there. Change in the input. And now ask if the model is able to capture the transients or explain. Model two will not be able to do that, right?

(Refer Slide Time: 13:45)



It's just describing the steady state characteristics of the process. So, there are these distinctions.

Now, why this has been brought up in the context of the slide that we have there from a model viewpoint, so let us look at this model here, particularly, the dynamic model. If you look at it from the model viewpoint, let us rewrite this in a slightly different way, so that it's computationally friendly, right? Because this is a recursive equation that you would use in a computation in an algorithm. If I look at it from that viewpoint, what the model is receiving, if I were to draw the block diagram, here, it requires not only the past input, but also the past response, correct? In order to compute or predict what happens at k.

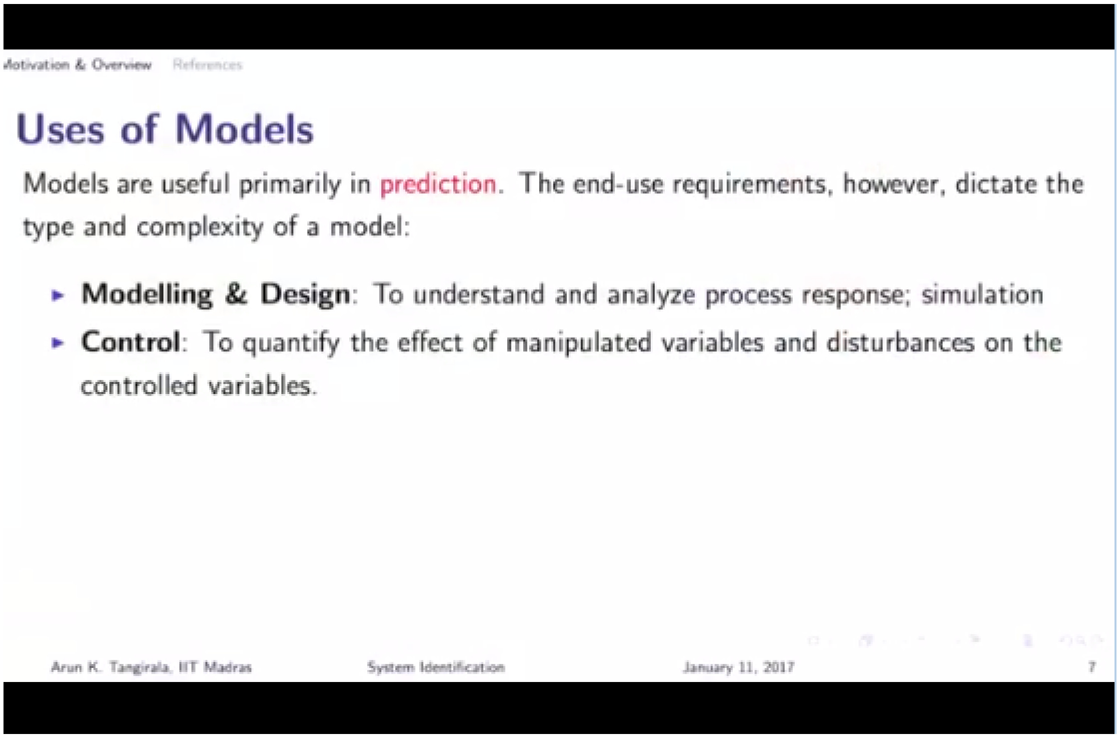
That's quite different from how the process behaves. That is, if you look at from the process as a physical setup, do you find a channel which has y k minus 1?

Suppose this is the model for your air conditioner, do you find a pipe or you know a channel into the air conditioner previous room temperature label. Do you find that? You don't find that. All you see is that there is a flow, physically there is a channel for the coolant flow and then temperature of course, you know, is a response, is an effect that you feel and you're sensing it. That architecture is different from a model architecture, where the inputs to the model are no longer necessarily the physical inputs. The inputs to the model are in general known as explanatory variables. Whatever it takes to explain the output will go into the model. In this example, u k minus 1 and y k minus 1 are considered explanatory variables. They need not tally with the physical architecture.

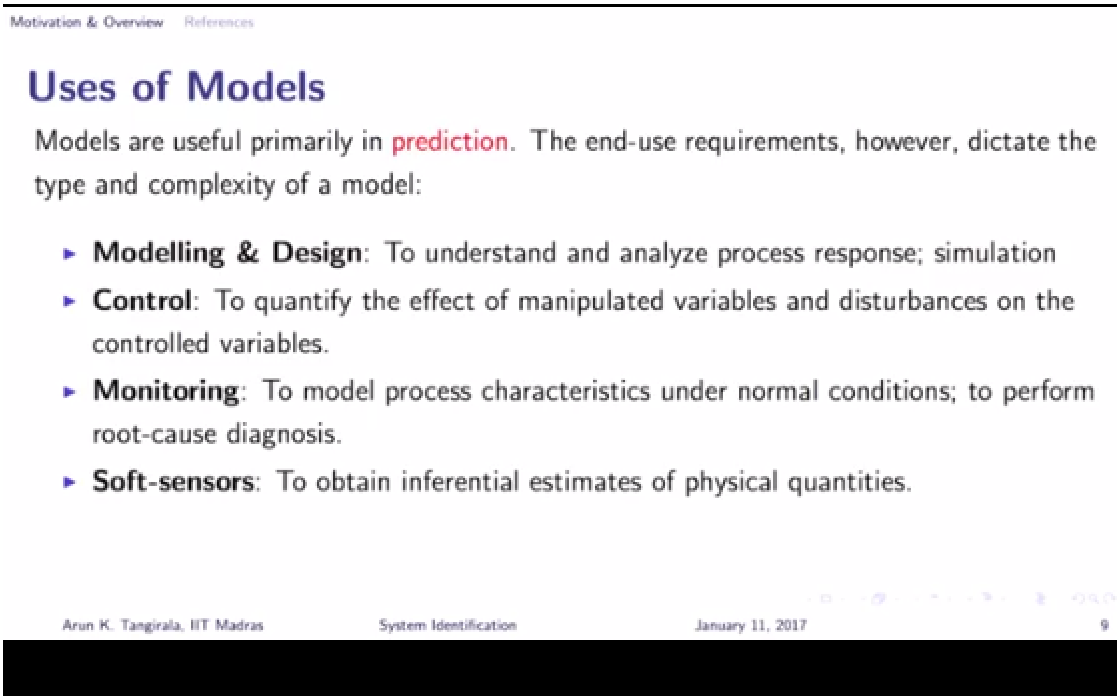
In fact, in a larger example, you would have many other variables being fed to the model, their past, not only the immediate past, but maybe past two observations, past 10 observations, depending on what kind of a model it is and so on but those are the things that go into a model. So don't think that the model architecture and the process architecture are identical. And this is important, because when you simulate, all of this has to be taken into account, when you're simulating, drawing a block diagram, or even coding, you have to understand and you should be comfortable with simulating a model. And writing the equation this way tells you what has to be fed to the model for it to speak out. So you have to you have to -- you cannot simply say, "Oh, look, the process receives only the input, I'm only going to give the input to the model." That's not the case. It needs the past information as well, at least for one step ahead prediction. So that is something to keep in mind.

Very quickly, let's go through the uses of models before we just talk about comparison between first principles and empirical that we were talking earlier. We have already remarked models are used in prediction primarily, but there are other applications as well. Typically, the kind of model and the complexity of the model is determined by the end use. So what we mean by end use is, for example, you may be looking at simply design of a process. You may be developing a model to design a process. And in design, generally, one works with steady state models. If you have taken a course on process design, you should be aware that you would have dealt with steady state, what is the throughput and so on, no. Generally, you don't worry about transient characteristics, right? Even if you were to talk to a car maker, only in some respects the car manufacturer will worry about the transient parts. Of course, you know, fine tuning the engine and so on. But there are other aspects of the design which only worries about the steady state characteristics.

(Refer Slide Time: 18:03)



Whereas in control, you need a model that describes the dynamics of the process, because control is all about shaping the transient, right? In air conditioners, in cars, in flights, everywhere, in the human body we have controllers. And what is the job of this controller? To shape the transient. And if the controller has to shape the transient, you need first a model. You need to tell it how the process would respond under transient conditions, and of course, control is not a human being, in most of the automated processes, it's a machine, it's a device. So, in order to program and design and program a controller, you first need a dynamic model and then you apply controller design algorithms to design the controller. So you see, compared to design, controller requires a different kind of a model. Moreover, if you look at the kind of models that are used in control, you don't have to develop very rigorous models as compared to other kinds of applications where you may need more rigor, right? You may need more accurate models. In feedback control, approximate models work wonders, because they say feedback is forgiving. So anyway, the feedback will take care of the errors that I made in modeling. On the other hand, in process monitoring, which is an alternative term to fault detection and so on, abnormal situation management, you would build different kinds of models. You may build a model coming out of pattern recognition or classification exercise, or you may build a dynamic model. Those models look different and the kind of requirements are different you want, in fault detection, you not only detect faults, but you also diagnose the source of the fault. And for diagnosis, you need a model that is causal that means it only relates causal variables. Whereas, in soft sensing and this is a classic example of the distinction that can be used to distinguish between empirical modeling and your causal modeling.



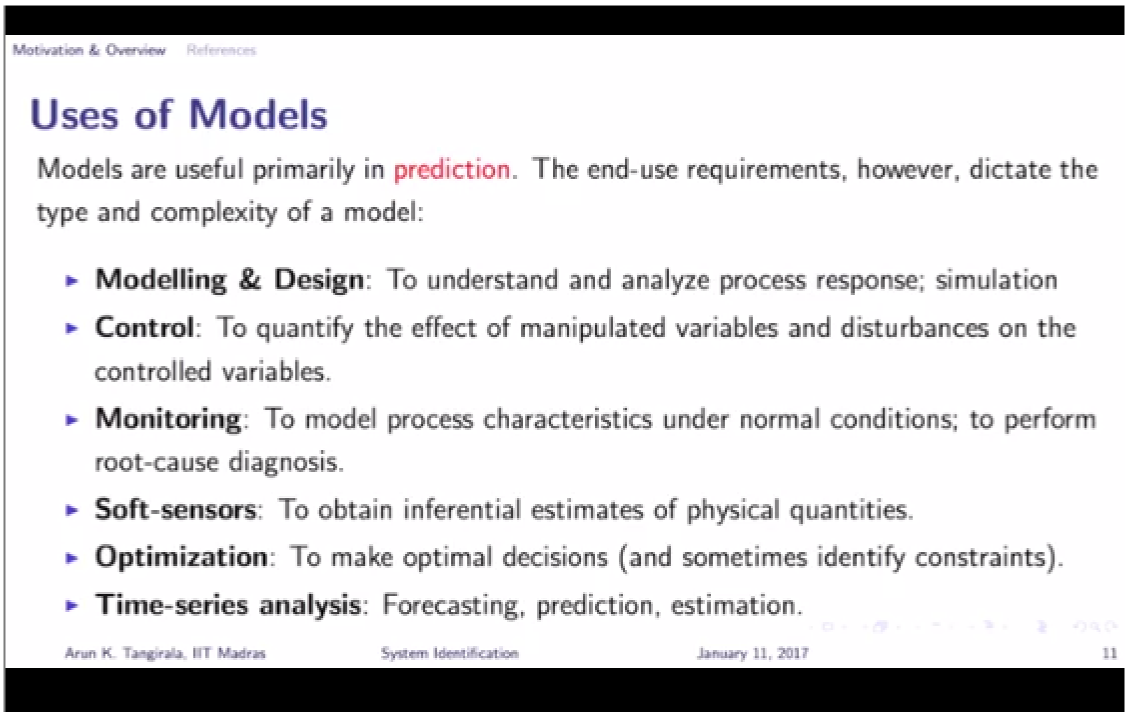
What is a soft sensor? A classic example is that of a blind person, right? You know, we know when for blind people, they cannot see. What is other alternative they have? They rely on sounds, right? You see blind people using their sticks to tap and then hear. The sound -- based on the sound, they kind of figure out where they are, that's because their hardware sensor, which are the eyes are not functional. So they have to infer the position based on an alternative measurement, which is the sound. That's a classic example of soft sensing.

Do you see this in process engineering? Yes, you see that. For example, if you take cement, one of the most important aspects of cement is cement fineness. How fine the cement is, right? Unfortunately, there is no sensor even today that can measure the fineness of cement online. So what do you do? And what do they do? They take -- they collect samples of cement and take it to a lab where the measurements are made offline with the help of analytical instruments. Is it possible for me to measure cement online? Yes, but through inferential sensing. I have other variables. I have the sound of the mill, I have the fresh feed flow, there are so many other variables that are being measured frequently. I can rely on all of those and construct an inference of the cement fineness. When you do that, what you have developed is a virtual or a soft sensor. These are becoming increasingly popular and we use that. We use it all the time.

Suppose I want to know if the bucket is full when I'm filling a bucket, without looking at the bucket. What do I rely on? On the sound that's coming in, right? The sound keeps changing, so I can kind of infer whether the bucket is half-full or full or near empty, and so on. So that's also soft sensing. What you're doing in soft sensing is building a model between so-called informative variables and a predictive variable. Those informative variables need not be the cause, need not be the cause. If I look at the bucket filling example, if I think of it, the sound that is coming out is actually an effect. It's not a cause for the bucket to be full. The cause is actually the tab being open, right. But I'm relying on the effect to infer the cause. And that's what distinguishes soft sensing from classical system identification. But that's also the -- if you look at the principles of soft sensing, the model development and so on that also they rely on system identification principles.

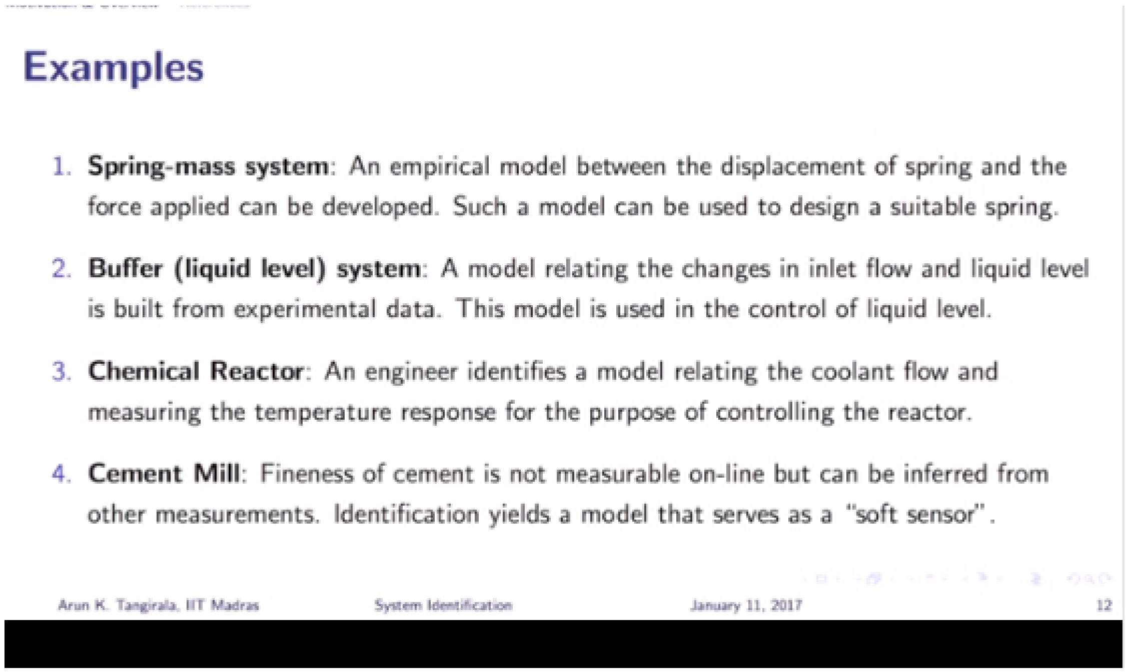
And then, of course, you have optimization where you may use dynamic or steady state models and then we talked about time series analysis where the kind of models that you build are not necessarily input output models, they are primarily stochastic models, which assumes some fictitious input sensor.

(Refer Slide Time: 23:15)



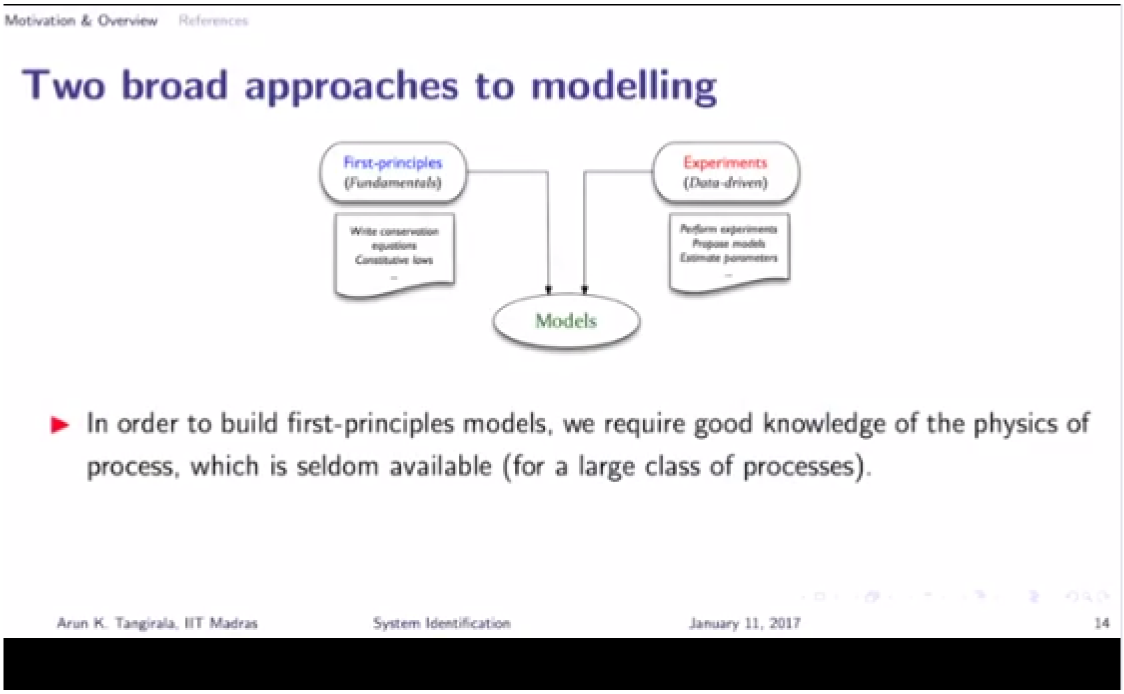
So there are a number of users of models in different applications and so on.

(Refer Slide Time: 23:20)

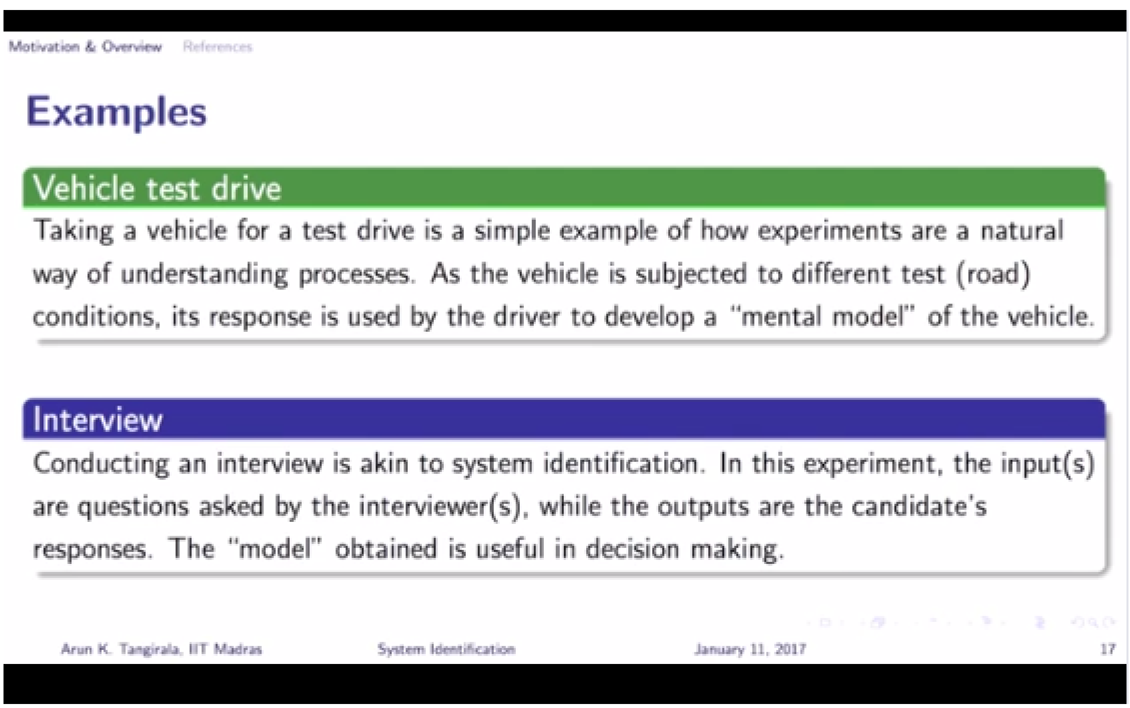


And there are so many examples that one can give. I've already talked about the cement mill example. We will study later on an example in detail, which is of modeling a liquid level system. It is a most innocuous system. You don't have to be a chemical engineer to understand liquid level system I mean, everybody uses it today, whether you're using, if you're using a geezer or if you're using a flush tank you are actually using liquid level system. So you see it every day, every morning, you need that. So we will go through a detailed case study on how to build an empirical model for a simple system such as a liquid level system. It will give us a lot of insights into what a typical identification exercise would contain. So I just want to conclude today's class with a comparison between first principles and empirical modeling.

(Refer Slide Time: 24:11)



Just one or two minutes, and then we'll continue this discussion tomorrow, where I'll talk about the course contents and so on. So we have talked about this two different approaches and I just want to go past these two, one example we've already discussed. The other example we'll bring up in a later session.



We have compared first principles and empirical models in some respects. Somebody said, "Well, it's going to take me a long time to solve first principles models," which is correct, because they may be computationally very unfriendly, because you're going to solve a bunch of nonlinear ODEs, PDEs and so on. Whereas that may not be the case with empirical models, but the foremost reason as to why we turn to data-driven modeling is because many processes out there are too complicated to understand from a first principles, from a fundamental viewpoint. And therefore, if data is available, I would like to turn to the data and ask it to speak out and to tell me what is happening underneath the process.

Now, one of the key differences between first principles and empirical modeling is, in first principles models, the models are -- structures are very rigid. You don't have any say. None of us have a say on what the model should look like. Whatever equations come out of the laws that you apply, those are the equations we have to live with. Maybe some approximation, but by and large, they dictate. Whereas in empirical modeling, there's a lot of flexibility. Now that doesn't mean you can fit any model, but the possibilities are enormous, which means there's a lot of flexibility, as a result of which there is also freedom and responsibility that one has to exercise. You can't build any model and say, "Look, you said there's a lot of freedom, I'll give you some model." That cannot be the case. And the other thing that is often pointed out as a difference is, first principles models are rigorous. Whereas empirical models are not so rigorous, it is true and not true and not true.

First principles models are good for a wide variety of operating conditions because you're developing from physics or, you know, laws, whereas empirical models are being built from data. So think of this example. This is the interview example that I had brought up earlier. When I'm conducting an interview, I ask a few questions and the candidate responds, right? I can only access the capability or the caliber of the candidate with respect to the questions that I have asked. Suppose I've asked questions on quantum mechanics. Now, I want to decide whether this candidate is capable of singing. Okay, that's a wide extrapolation, right?

I cannot infer such things because my questions do not contain -- they are not gone ventured into those operating conditions, of singing operating conditions. So data-driven modeling has enormous limitations when it comes to extrapolation. Now having said that, it is unfair to say that because your data has not ventured -- your experiment did not venture into the operating condition. If you want this model that you're building to predict in some other operating condition, you have to have that in your data set. Because you're training the model, the model is as good as the data. If your data quality is poor, model is poor. If your data is good, model is good. If the model is good -- if the data has covered a wide-range of operating conditions, your model is also going to be good. So data is the food for identification and therefore it's extremely important in identification to have good quality data if you want good model. Right now, we need food for the stomach, so we'll stop here.