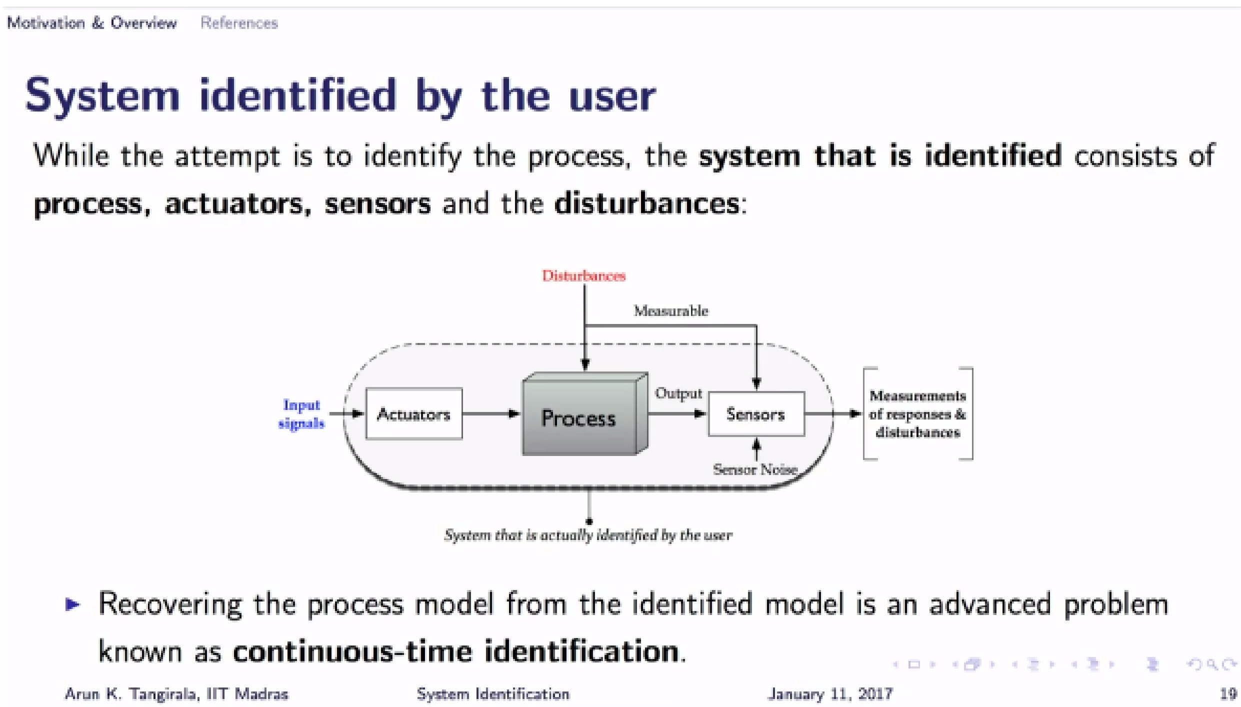
CH5230: System Identification

Journey into Identification 1

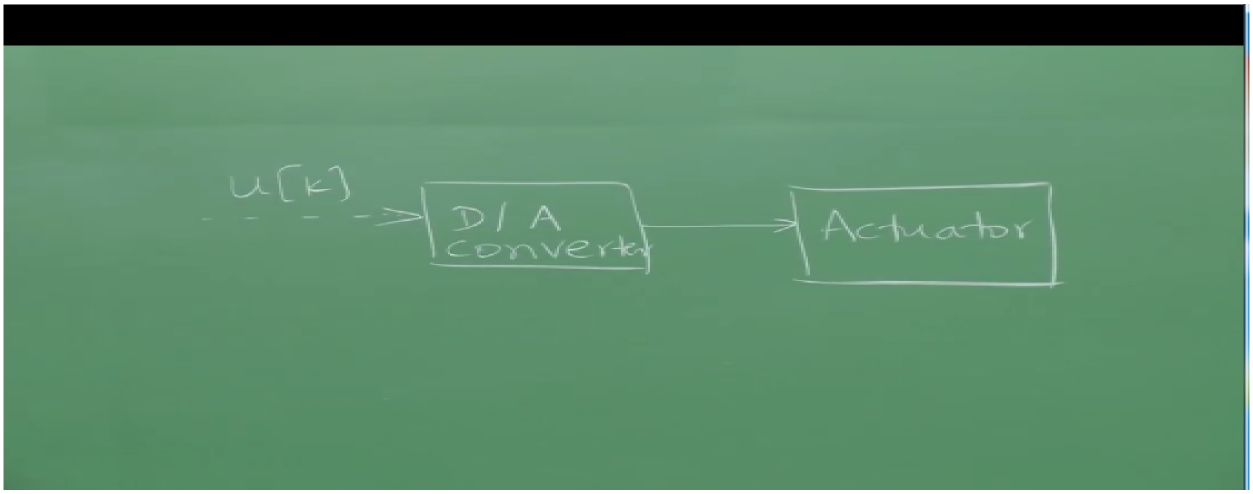
Before we get going today, just wanted to make a small correction to what I have said yesterday when we were talking about some important aspects of identification. If you recall, we spoke of actuators, and sensors, and so on. And at some point in the lecture, I had mentioned that the role of the actuator is to convert the discrete time inputs applied by the user into an approximate continuous time input. In fact, that is not strictly correct, because there are two elements missing in this schematic. So typically, what happens is, which of course, a schematic of, a full schematic of which we'll see later on. But let me denote the discrete time input with the dashed line.

(Refer Slide Time 1:11)



Typically there is a D to A converter, which then sends this command to the actuator. So, the actuator does not receive the discrete time input. It didn't-- I kind of, omitted this, a very important aspect yesterday.

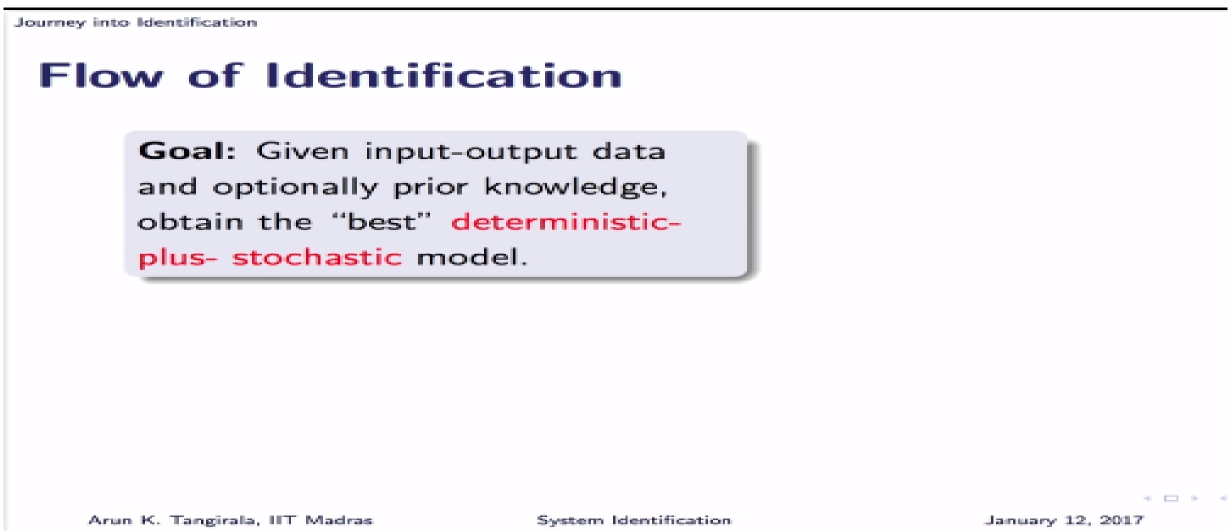
(Refer Slide Time 1:28)



But all the other points that we discussed hold, the role of the D to A converter is to convert this discrete time input to an approximate continuous time input. The D to A stands for digital to analog. Well, when we say digital, somehow we always think of discrete time. But strictly speaking what digital means is that the amplitude have been digitized. You can still have a continuous time digital signal. In principle, you can have that. But by and large, since we talk of digitization for discrete time signals, somehow this word digital has been used interchangeably with discrete time. So here, you should understand when we use the word digital, we are referring to actually discrete time signal and likewise for analog as well.
Analog refers to the values that a signal takes, that is amplitudes, whether they are continuous or discrete. And you can have an analog continuous-time signal, as well as, an analog discrete time signal, because you can still have a discrete time signal with continuous valued amplitudes. And once again, like in the digital case, the word analog has been used more or less interchangeably with the continuous time, kind of, signals. So, once again here, you should therefore, use this word analog-- view this word analog as a substitute for continuous-time. And then the actuator will convert this electrical signal to a physical signal, which then excites a process.

So, what I want to say is, yesterday when I talked about this actuator, the role of this actuator is not to convert the discrete time signal to the continuous-time signal. Strictly speaking, its role is to convert this electrical signal to a physical signal which then excites the process. And likewise on the sensor side, although I put a block named sensor there, there are quite a few elements within that sensor. So you have a sampler, and then you have quantizer, which are not seen there. And then you have the real sensor also, right? So there is a sampler, then there is hardware, there is a physical element, for example, thermocouple used for measuring temperature. And then, you have a quantizer because ultimately, the signal that's coming out of the sampler and the hardware sensor has to be quantized in order for us to store in a hard drive. Maybe it's a computer hard drive or a flash disk or whatever it is, ultimately, storage of signals requires quantization. So those elements are missing but they are not so much relevant at this moment. At a later stage, when we learn how to connect this continuous-time process to the discrete time process that we identify, at that point in time, we will bring in all these elements, and then learn how to build a model that maps the continuous-time process to the discrete time process. But at this moment, it is not necessary. I just wanted to make this small correction to what I have said yesterday. All the other aspects remain intact. So now, getting to today's lecture, let me now take you through a journey into identification. I can start off with the theory, particularly the linear systems theory, but I don't want to do it straight away without giving you a feel of what you should expect to see in identification. And in fact, most of this material-- I think, all of this material has been taken from chapter 2 of the text. And I believe that this is a good way of introducing the subject rather than straight away plunging into the theory so that, you also know what constitutes a typical identification exercise, what you should expect to see when you sit down to estimate parameters, and particularly, what you should expect to see, in terms of, effects of noise. What can noise, actually, what role noise plays in identification, and how it can impact your ability to estimate parameters, or estimate model and so on. Alright? Of course, then it'll be subsequently useful to you also to make a decision, whether you want to continue with this course or not. So, let's begin with this journey here. And the first thing that we want to know is, what is the goal of an identification exercise? And what is a typical flow of an entire identification exercise? So, our goal is very clear. I'm given input-output data and I am going to estimate the model, that's it. It's very simple, and typically, the shorter the statement of the goal is, the deeper is the under underlying theory and everything else associated with it.

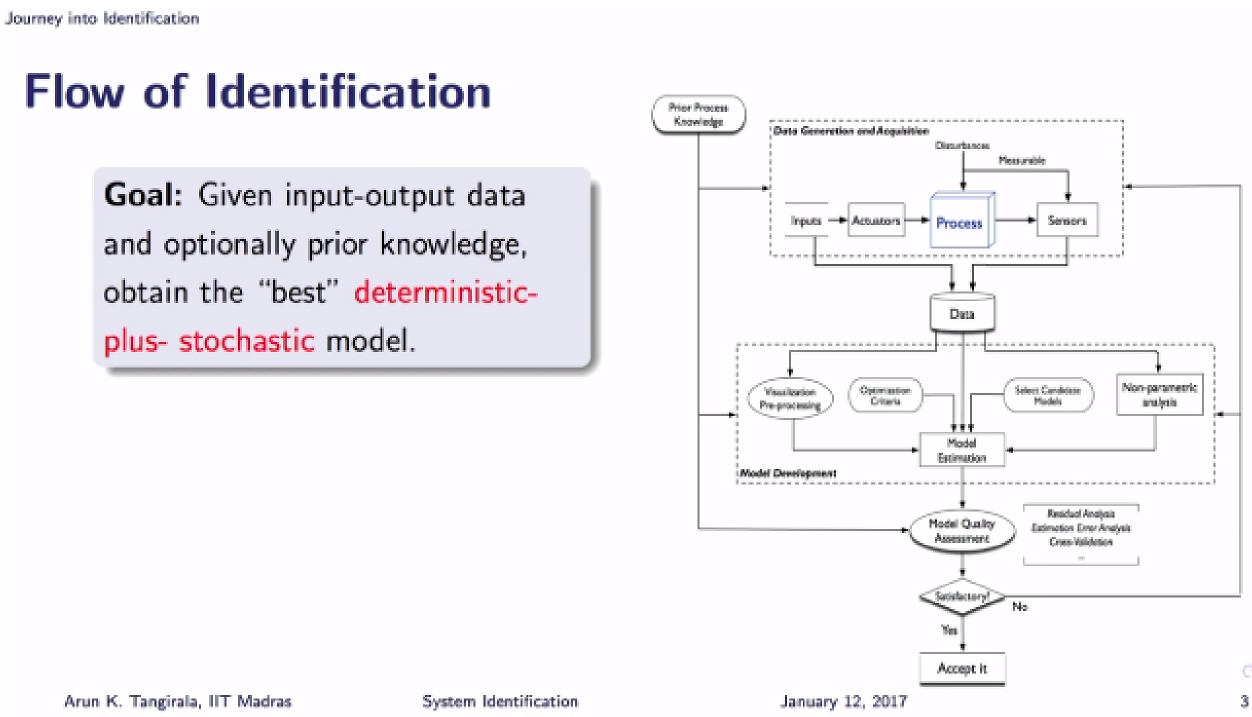
(Refer Slide Time 6:56)



Right? So, there is an inverse relationship between the depth of the exercise and the statement of the problem. As simple as it sounds, yesterday we discussed quite a few salient aspects; we said there is this problem of input design, then there is this question of what sampling rate should I choose.

More importantly, let's say somebody has done that experiment for me and now I have to decide what models to build, what are the decisions that I have to make during model building? How do I validate my model? There are so many questions that are begging answers in order to realize this goal. So typically, you would see a flow in identification and this flow is not necessarily sequential. Let me tell you upfront. There is definitely some sequence, but it's also iterative in nature. You may have to go back and forth. And I don't know how well you can see this schematic, but let me go over it a bit in detail. At the top, that is, in the first stage of identification, you have data acquisition. Essentially, you can think of this as, maybe three or four stages depending on how you can view the identification workflow. The first stage in identification is always data acquisition. Either you do it or someone else does it. Remember, data is food for identification, without data there is no system identification. At this stage of data acquisition, there are a number of questions that have to be answered. Some of which we have already mentioned yesterday.

(Refer Slide Time 8:40)



For instance, what input should I use in my experiment? What should be the input design? How frequently should I change the input? Should I use a step input? Should I use sinusoidal? Or should I use some arbitrary signal, a chirp like signal, or some pulse like signal, what are the possible inputs? And does it matter really, or the theory that I'm going to rely on for parameter estimation, does it impose some restrictions on the class of inputs that I can use? So, there are so many questions that have to be answered, and all these questions constitute the sub-branch of System Identification know as input design. And this is a very [hard 9:22] area. It is still evolving. We don't have complete answers to this question of input design. But partially some questions have been answered, at least, for linear time invariant systems, which is the class of systems that we are going to consider. That's the scope of this course. I've not talked about the scope and notation yet, but we'll do that before we get into formal discussions. But, I'm just telling you upfront that it's the LTI, linear time invariant systems that we are going to look at largely. And for such systems, these questions have been, at least, answered to a great deal. But there are so many more other classes or systems for which still it's not known what is the optimal input that I should give.

Then comes the question of choosing the sampling interval and placing the sensors. So on the output side; you have these questions to answer. How do I actually choose the sampling time? What are the criteria? What should I think of? Should I sample too fast or can I relax and sample? What are the constraints that

I face? Are there any issues with the sampling too fast? And in answering these questions, typically there are two considerations. One is process dynamics. If I observe a process too slowly, what happens? Suppose I observe too slowly, or just in a slow, its relative time, then I'll miss out on the dynamics, right? There may be some key features of the dynamics that I may miss out. There is a peak in the response and I may miss that out.

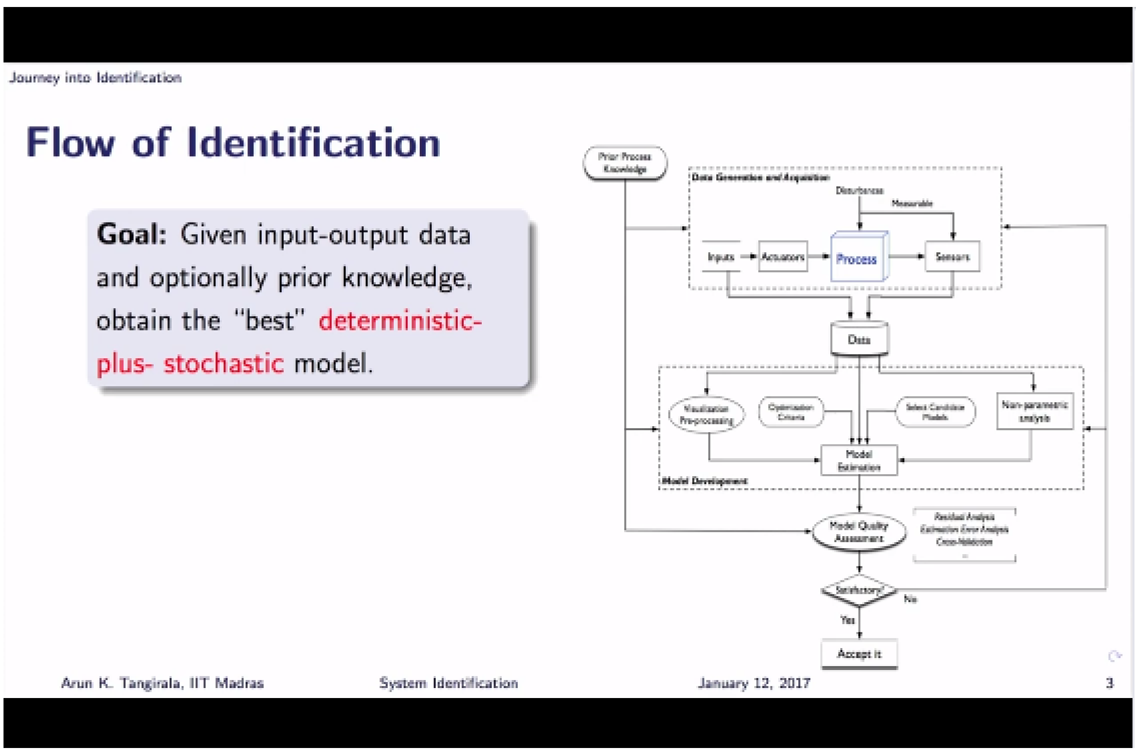
That's one consideration. At one end of your sampling, you have this issue of missing out. If I sample slowly, then I'm going to miss out on the process dynamics. If I sample fast or too fast, what we mean by sample too fast is, suppose I'm standing at a railway station and typically you're waiting for the train to arrive, you've gone to receive someone, and you're just constantly checking if the train is going to arrive or you are going to board the train. The train will arrive regardless of whether use a sample fast or slow. It's going to take its own sweet time to come, right? But what happens when I sample too fast, I get disappointed, because I'm expecting the train to arrive every time I'm seeing my expectations increase. But that's got to do with the psychology of the mind. Here there is no mind but there is something else that we have to really mind about. Which is, noise. If I sample too fast, I may end up getting more noise than actual dynamics. This is what many of the television channels end up doing to us, right? Fortunately, I don't have a TV at home. But the TV channels, they feed on any event that occurs in any part of the world, and then they divide the screen into two or three portions, and then there is a, you know, a screen dedicated to the happening of the event and it's just put through a for-loop, so it just keep on looping through this event. And they keep creating news. In reality, there is not much happening every second there. Whatever supposed to happen has happened, and then the next significant event may happen only an hour or two or depending on the situation. But because the news channel wants to keep you engaged, keep you hooked onto that channel, it just keeps talking something or the other all the time, and contacting this person who is about to drown or something else is happening and they keep creating news. And you think, "My god, so many things are happening there." Most of it is noise. So at that point, you should think of news as a short word for nuisance. Okay?

It's just an abbreviation of that. Now, that is a fact, also in signal processing, in data driven, in these kinds of experiments where you are collecting data. The actual process is not changing as fast as you are observing or the scales are not commensurate. So, the end result is you bring in more noise into the data and that places a burden on the estimation algorithm. Because the estimator is supposed to be distinguishing between process dynamics and noise in a clever way, but it also has its limitations. And we quantify this, in terms of, what is known as a signal to noise ratio. I'll show you an example shortly. What is the impact of signal to noise ratio on parameter estimates? I don't know how many of you have heard this term called signal to noise ratio, but it's a measure of the extent of certainty in the data, determinism in the data to the extent of stochasticity, randomness in the data. Obviously, it makes a huge difference. Our goal is to get to the deterministic part of the process. But this noise is inevitable. And we should design our experiment including choice of sensors, the way we perform the experiment, everything should be done so as to minimize the effects of noise. But when we make wrong decisions such as fast sampling intervals and so on,fast sampling rates, then we do end up defeating the purpose of that. We end up getting more noise. Now that's one aspect of data acquisition.

The other aspect is choosing the location of the sensors, number of sensors. In many applications, we may have to-- we may not have the luxury of placing the sensor anywhere I want and many other applications I have, and maybe one sensor may not suffice. I may have to place actually many sensors depending on the kind of application that I'm looking at. So there is this problem of sensor network placement. And remember, every sensor gets you some information about the process. Question is, whether the number of sensors that you have obtained or you have placed, is actually getting you new information or just getting repeated information? Remember, each sensor is going to be expensive. It's going to have a bearing on your finances. You have to invest money to buy this sensor and you also have to maintain it.

So one has to be careful and that's the case in industry. They're pretty careful when it comes to making decisions on how many sensors to buy, and place, and so on. So there is a sub-branch of, again, system identification you can say, which has got to do with sensor network placement and so on. That, again, of course, we won't discuss in this course. So, let us say all of this has been done and data has been acquired. Now, you are really at this stage where data has been obtained by someone or by you, and the next important goal is to obtain the model of the process.

(Refer Slide Time 16:34)



At this stage, when you're presented with a data, in many situations, you may have some prior knowledge about the process. What we mean by prior knowledge is, you may know for example, that the process is linear. It's linear or nonlinear, or that it's first order or second order, some knowledge you may have, or that, some parameters are partly known, some are not known, or you may have some idea of the structure of the model or you may not know anything about the model. Typically, we assume that we do not know anything. In black box modeling, anything in a sense, of course, I should know what is the process from which this data has been acquired? It's not that I wouldn't know whether the data is coming from an atmospheric process or an engineering process.

That I would know. But the kind of knowledge that we are talking about is that helps you in your modeling. That we'll assume is not available. What is input-output delay? What is the order? Whether there are any peculiarities about the process such as inverse response and so on. Assume that none of this is known. In black box modeling, that's the case. Therefore, it's very important to spend some time in doing what is known as exploratory analysis. So, this is typical of any data analysis, there is this first stage called exploratory analysis. You're simply exploring the data. You are in an unknown territory and you're trying to figure out what is there in there, right? So you don't make too many assumptions, and the methods that also you employ for this exploratory analysis, do not make too many assumptions. They have to make some assumptions but they don't make too many. That constitutes-- so that has two parts to it. One is visualization, your own analysis of the data without relying on any mathematical method. Believe me, that's a very important stage in data analysis. Please, do not consider the machine as a complete replacement for the human being. And I hope that never happens and I think it will not happen. Because the brain is far more intelligent than we can ever imagine. And a lot of inferences that we can draw from visual analysis can take enormous time to code. You think of it this way, when I give you a signal, suppose you're looking for peaks in the signal, right?

Visually I can pick the peak with quickly, very easily. There may be some subjectivity but we are very good at picking where the peaks are. Now, try writing an algorithm that does peak detection. It's a very nicely studied problem, but still it's evolving. There’s still some open ended problems. But finding a sound algorithm and coding it, you will realize that there are so many challenges in this peak detection problem, when you have to automate, you just have to write a code, you want to replace the human factor there with the machine. That problem just of peak detection requires enormous amount of coding. Therefore, you should not undermine the power of visual analysis. Spend time, make friends with the data, understand what your data contains, whether it contains anomalies, any trends, anything that you can quickly extract visually should be noted down. So spend some time in generating different plots that give you a lot of insights into the data. And then only get into the algorithms or the mathematics that will help you extract more juice out of the data.

But the first stage should always be your personal attention to the data. This is what unfortunately, even in hospitals, many of the doctors do not practice. When a patient walks into the clinic, typically, the doctor-- I'd still remember one stage when I took my mother to a doctor. She was complaining of some shoulder and neck pain. So he said, "Okay, please describe your problems. I have a look up table in my mind. Shoulder pain? Okay, this tablet. Neck pain? This tablet. You may go." There was no clinical examination of the patient at all, which is not correct at all. A doctor should clinically examine because, the patient himself or herself may not know what the actual problems are? What you feel is not necessarily the entire truth to it. Moreover, patient may only report the symptoms, but the doctor has to clinically examine the patient to figure out more. That's exactly what as a data--