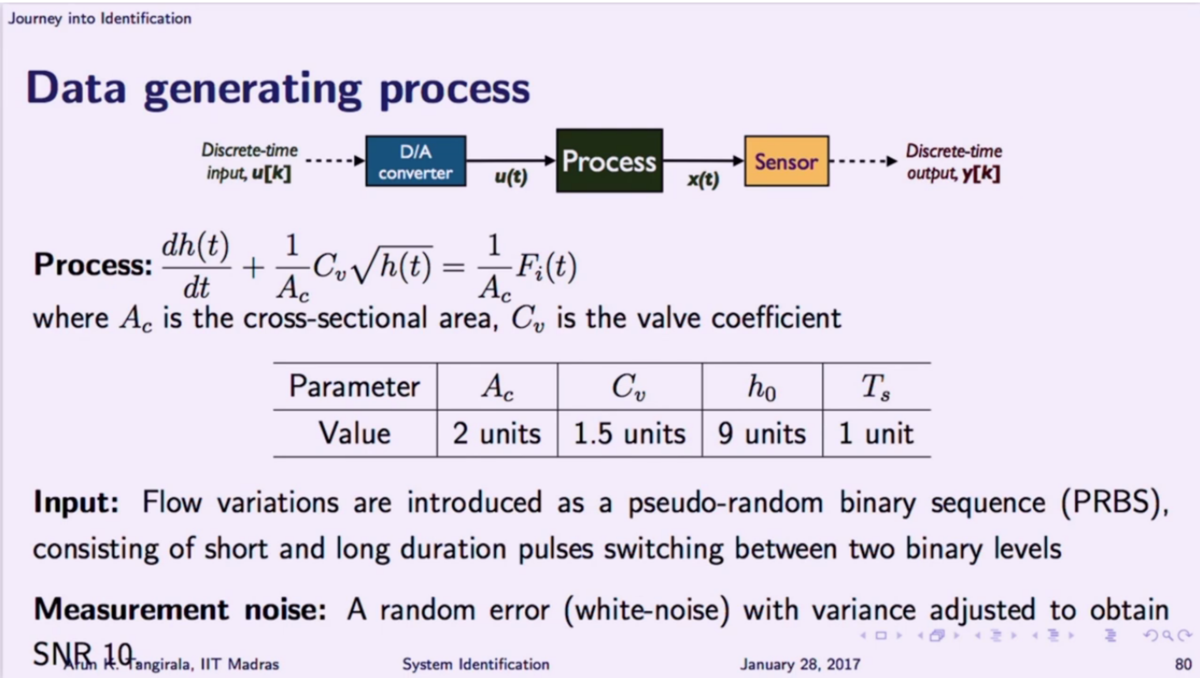
CH5230: System Identification

Journey into Identification 18

What we'll do is of course, as I promise today will start of on the theory, but I thought it would be anti-climax if I do not disclose the full details of the process that I had simulated for the liquid level example. So here, this is slide that explains or kind of gives you the details on what process was used to generate data for the liquid level case study. As I've already told you the process is a first-order, described by first-order nonlinear OD. And in this equation A c is a cross-sectional area. C v is the so called valve coefficient. For those of you are not familiar with the basic fluid mechanics.

(Refer Slide Time: 01:17)

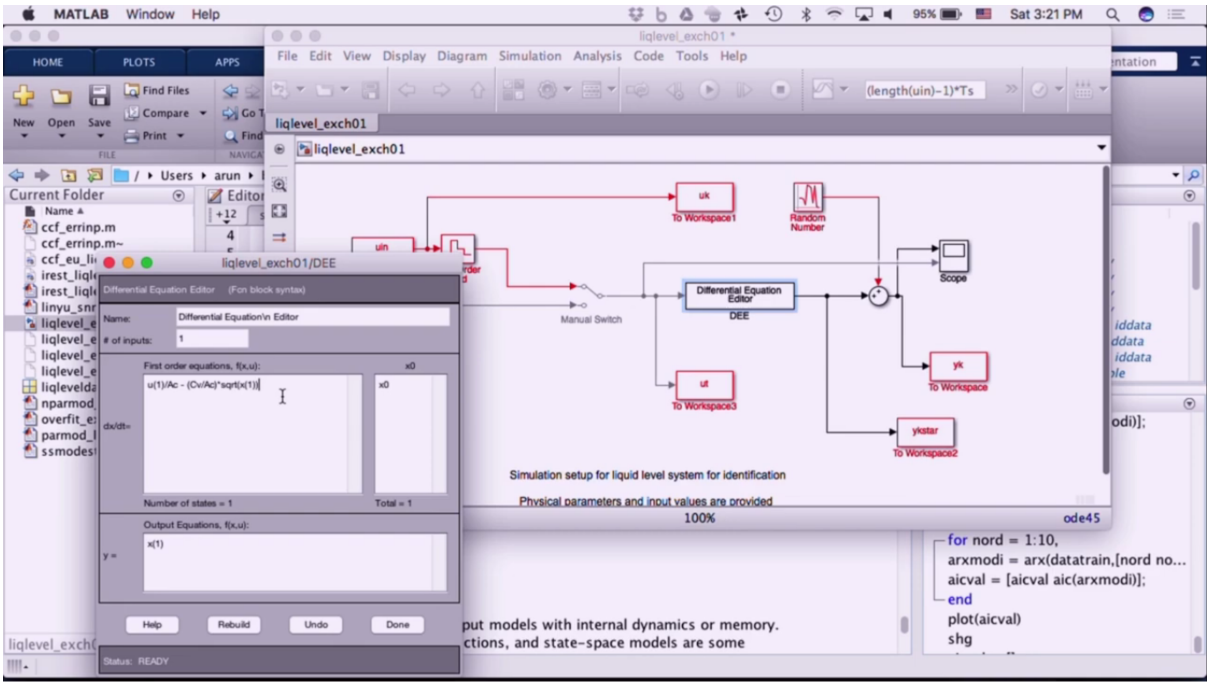


Generally for a flow that is coming out of a pressure head, you can actually model the flow as a function of the square root of the head, liquid level head that you have. In other words, say, F out is modelled as some constant time square root of h and that's how this term enters the differential equation. So C v is called a valve coefficient, assuming that there is an outlet valve and it signifies the resistance. Larger the C v, lower is the resistance. Which means for a given height, larger the C v more the flow and because more the flow, lesser is the resistance at the outlet. And in fact, one can draw parallels with other systems, what I mean by the other systems in other disciplines. For example, you can consider an RLC circuit in electrical engineering systems as a good analogy of a liquid level or this kind of a flow system.

Where the equivalent of cross-sectional area would be the capacitance there, because the cross-sectional area is the capacity of the time and the flow, analogy of flow would be the current in the circuit. And what would be the analogy of the liquid level? The potential the voltage. Potential difference across the terminals of the circuit. And then what about C v? Inverse of the resistance, right? We have already said that C v is a valve coefficient. It is inversely proportional to the resistance offered by the valve at the outlet. So simple an RC circuit itself is sufficient. In other words, by going through this example, you have also understood how to develop a model for an RC circuit. Although in an RC circuit, you may not see the square root type of nonlinearity. Typically you assume an ohm's law, right? The drop across the resistor. But here, it's a mildly nonlinear, anyway. So this is a continuous time process that we have used. And remember we said we use a D to A converter which is typically a zero order hold device. In Simulink you can drag ZOH block from the Simulink library. And hopefully most of you have already looked at the Simulink diagram that is available on the website. It shows you the constitution of the block diagram. So let me actually show you. This is how it should look like. You'll see that in place of the liquid level system I have the differential equation editor. If you double click on this, you would see the differential equation written here. Many students do not know the availability of such a block in the Simulink library.

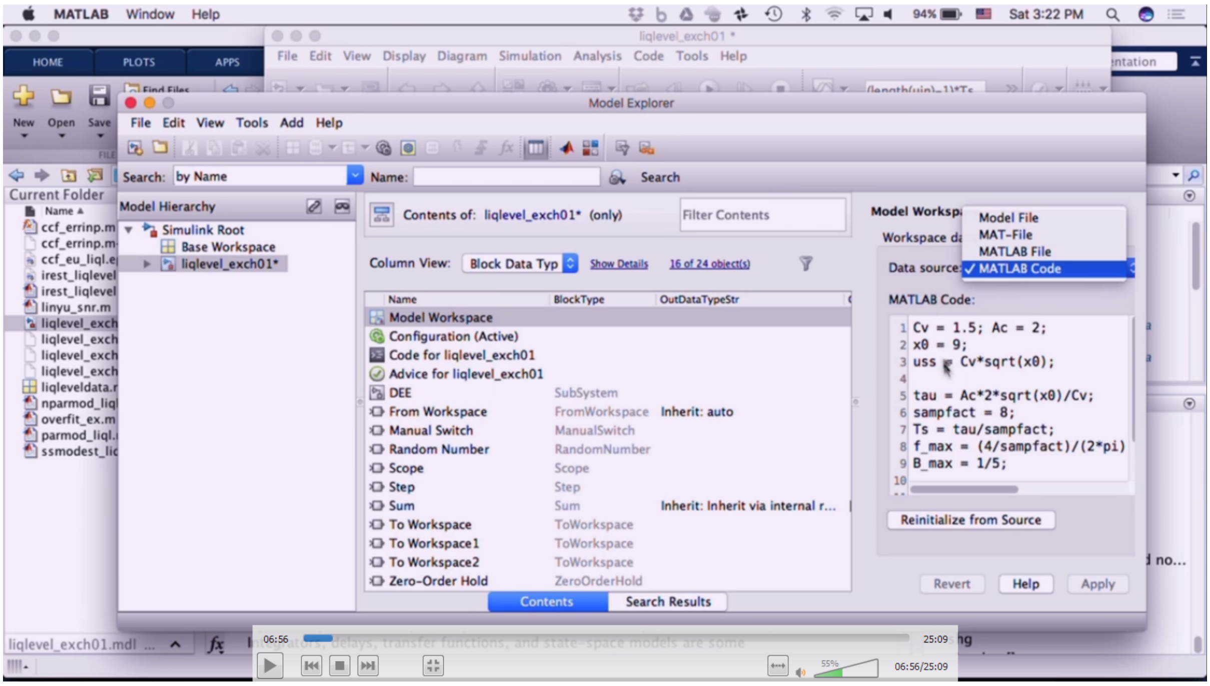
The way you bring up this block is first by typing DEE at the MATLAB command prompt, right? And that would bring up a Simulink window with the DE block and you can drag it into your model. What you see in this window is called as a Simulink model. Where you are connecting blocks of different types together. All of them described by some mathematical equations or some statistics. And therefore Simulink offers a very powerful platform for simulating any kind of system for which you know already the math. And then of course, what you see here is a zero order hold. This block is supplying the input that I have designed using PRBS. And, of course, if you go back to the differential equation editor, I don't know how well you can see from the back, but there are this symbol C v, A c and so on. And these symbols have been defined within the Simulink model.

(Refer Slide Time: 05:22)



So in case you want to know, where it is in case you want to play around with this increase the cross-sectional area or change the valve coefficient and so on. There are many ways of accessing this. One of the simplest ways is to go to this view and go to model explorer, right? And that will bring up this window for you and in this window, go to model workspace. A few years ago, I don't know how many years ago. But a few years ago this facility was not there. This provision of declaring the symbolic variables within the Simulink model was not available. So in those days what you had to do is, you will have to declare those variables in the MATLAB workspace. Or preload them from some MATLAB file. Essentially those variables had to be declared in MATLAB workspace. But a few years ago, this scenario changed and this is very nice because now these variables are not globally available to you. They are just available to you and for that Simulink model. And that's very nice. And here, you can assign values, you can include commands here. You can do all kinds of things, all you have to tell is what you're doing? Here I'm clearly telling is, telling Simulink that this is a MATLAB Code that has to be executed. Each time the model is run. Okay. So you can do all your calculations here that are required for the model to run. Or you can link it to a MATLAB File, if you have written it up. And or you can give it as a MAT-File and so on. I prefer the MATLAB Code, because then this model can be supplied as a single MDL file.

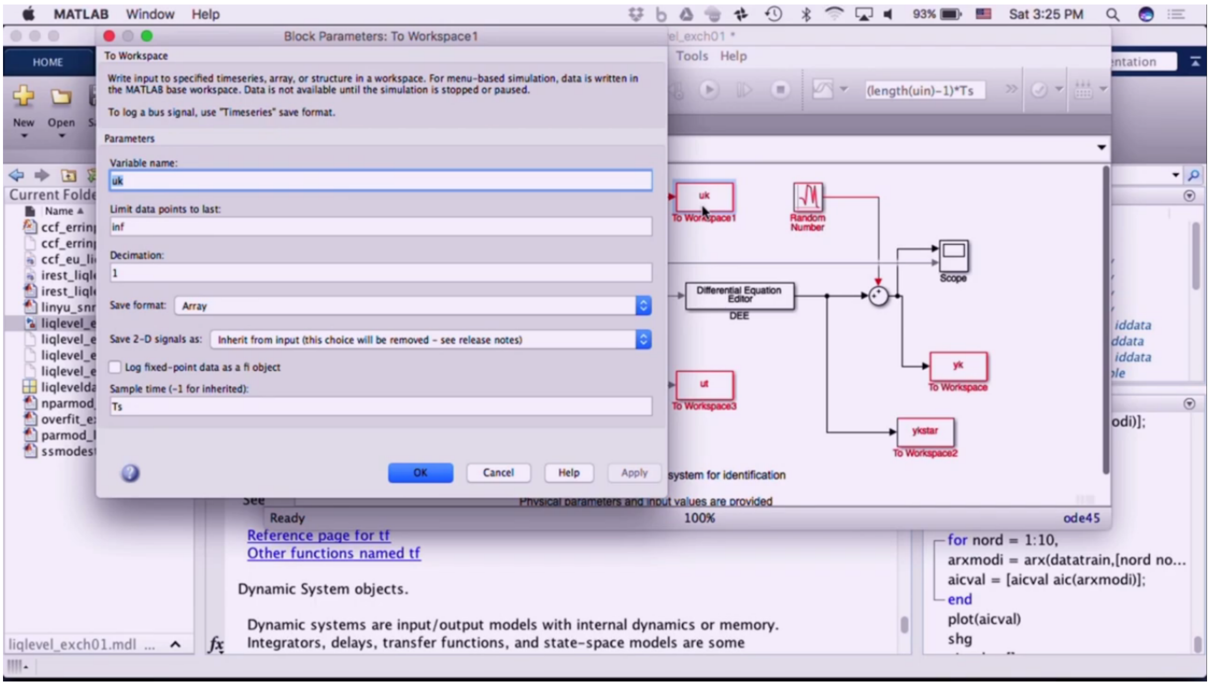
(Refer Slide Time: 07:12)



All the Simulink model files have a dot MDL extension. On the other hand if I say, a MATLAB-File then I have to supply to whoever is using it the MATLAB-File as well. Okay. So you can see here several MATLAB comments, pertaining to the variables that we use in the Simulink model. Please take a look at this. So let me take you back here. There are so many other things that you can actually play around here. But I have set even this top time and I have even included the input design. I don't know, if you noticed let me actually bring that back. So you see the ID input command in MATLAB system identification tool box allows you to generate a PRBS a pseudo random binary sequence.

And I specified how many observations you could play around with this you can even change the type of the input that you are using and so on. So, almost everything that is required for the execution of the model is contained in that model workspace. And I've also had, I'm not sure if it's the same. This provision is also there in the model that I posted. I will have to go back and check. But basically what I have done here is if I want to generate a step response. Remember I showed you a step response early on preliminary experiment. I could do that and I can use the switch in Simulink to switch between these two inputs. So, that I don't have to add and delete and so on. And of course, I'm collecting all the signals in these respective variables and in any of these blocks a typical two workspace block will look like this where you specify the variable name. You specify the sampling interval and there is also a decimation. Okay.

(Refer Slide Time: 09:17)



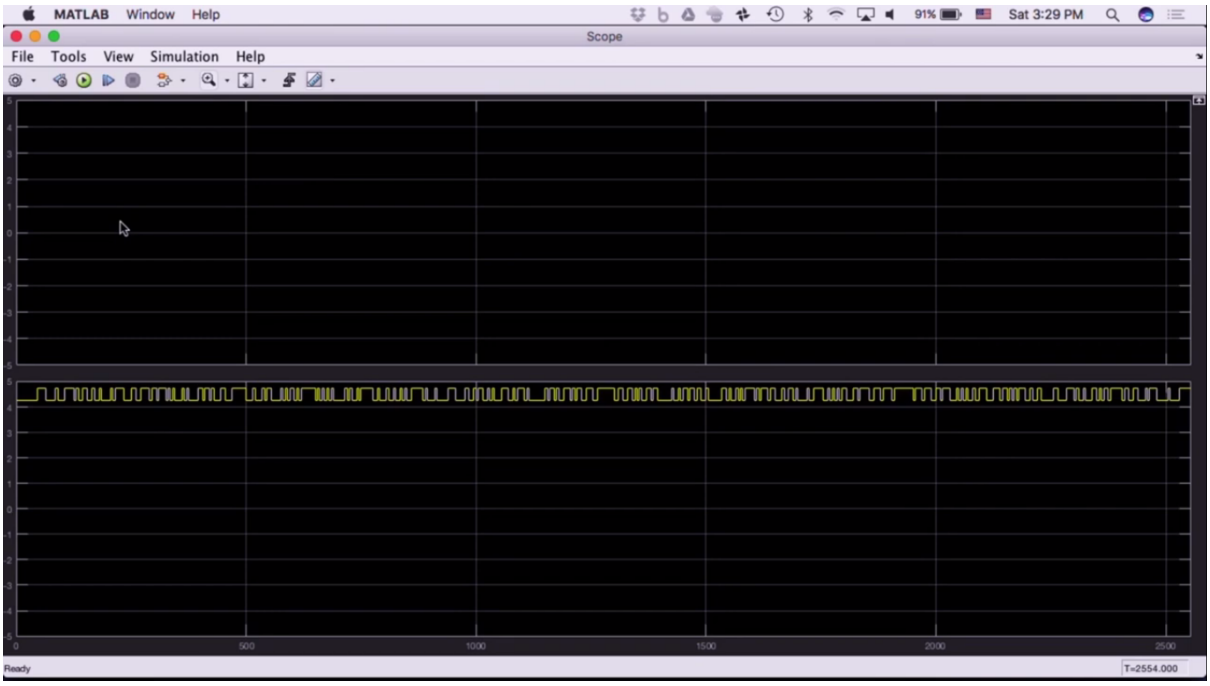
Now there is a difference between sampling interval and decimation. I don't know how many how many of you actually understand how Simulink works. Simulink essentially, you know, when you're simulating a dynamic system. You are actually solving ODs. That's all. That's what you're doing. I mean if you are simulating a continuous time dynamic system. Of course, if you are simulating a discrete time linear dynamic system then that is a different equation. But since we are simulating a continuous time dynamic system here, you should know that what Simulink is actually doing is, it is performing numerical integration. And we have discussed about numerical integration, right? How do numerical integration schemes work? They use a fixed set, fixed step or a variable step scheme. And you can choose this solver by default Simulink chooses what is known as an ODE45solver.

And this ODE45 is a variable step numerical, implies a variable step numerical integration method. If you do not specify the sampling time, let's say here there is a sample time. At the bottom here, right? I don't know how clearly you can see from there but there is, this is corresponds to the sample time field. And I have given here this Ts and I declared the Ts in the MATLAB Code that I showed you earlier. All right. I could say minus 1 here. Instead of Ts if I say minus 1 what it does is, it collects the data as and when the numerical integration is performed. Which means the data obtained is at the time instance that Simulink chooses two numerical integrate? Which may not be on a regular grade? Because it's using a variable step-size, right? Whereas here, I am demanding that the values of the signal be obtained at specific time instance in time, right? That is at Ts, 2Ts, 3Ts, and so on.

In addition to that there is a decimation. Now this decimation serves as a down sampler. This is different from sampling. So, right now, the decimation is set to 1 which means no down sampling is performed. I could have Ts and I could have decimation too. Which means it would integrate the response at 0, 2Ts, I mean it will give you, it may integrate at 0, Ts, 2Ts, and so on. But it would store only at 0, 2Ts, 4Ts, and so on. Typically that would not be required in any of our simulations. 99.99% of the times we wouldn't even look at this, but just for your own information I'm giving you this description here. All right.

And there is a format in which you can save these variables. Typically I would like to choose array, but you have the option of saving it to a structure. I don't know how many of you are familiar with MATLABS structure format. But it's a very powerful way of storing an object, because you can have multiple fields, when you say structure it will actually store the values and the time instance and so on at which it has been updated, but array is good. And you had the option structured with time or just simply structure and so on. And then there are of course, other things which will not worry about. That's it. And then there is a scope here. Which gives you the display of whatever signal that you want to see?

(Refer Slide Time: 13:01)



Now, although I say that's it. There is a very important thing here, which is the source of this course and everything, which is the random number generator. Which serves as the source of noise that we are adding to the measurement? Now, what is a block diagram tell you? This by the way this random number generator is going to generate white noise sequence. A Gaussian white noise sequence what we mean by Gaussian white noise sequence is, it's going to generate a bunch, a sequence of numbers falling out of a Gaussian distribution. With 0 mean, of course, you can change the mean and variance as you specify. But the most important thing is these numbers are going to be uncorrelated. That means there is no linear model that can predict, not improve your prediction, the prediction of the sequence beyond its average. So given any amount of past the prediction won't improve theoretically. And here, there is a sample time. You want to make sure that the sample time is the same that you have set everywhere. And you can see that there are these objects that are coloured, right? What do you think those colours are? What are they signifying those red colours? Yes. No. There are only two colours, right? Red and black. What is the difference between those channels or those blocks?

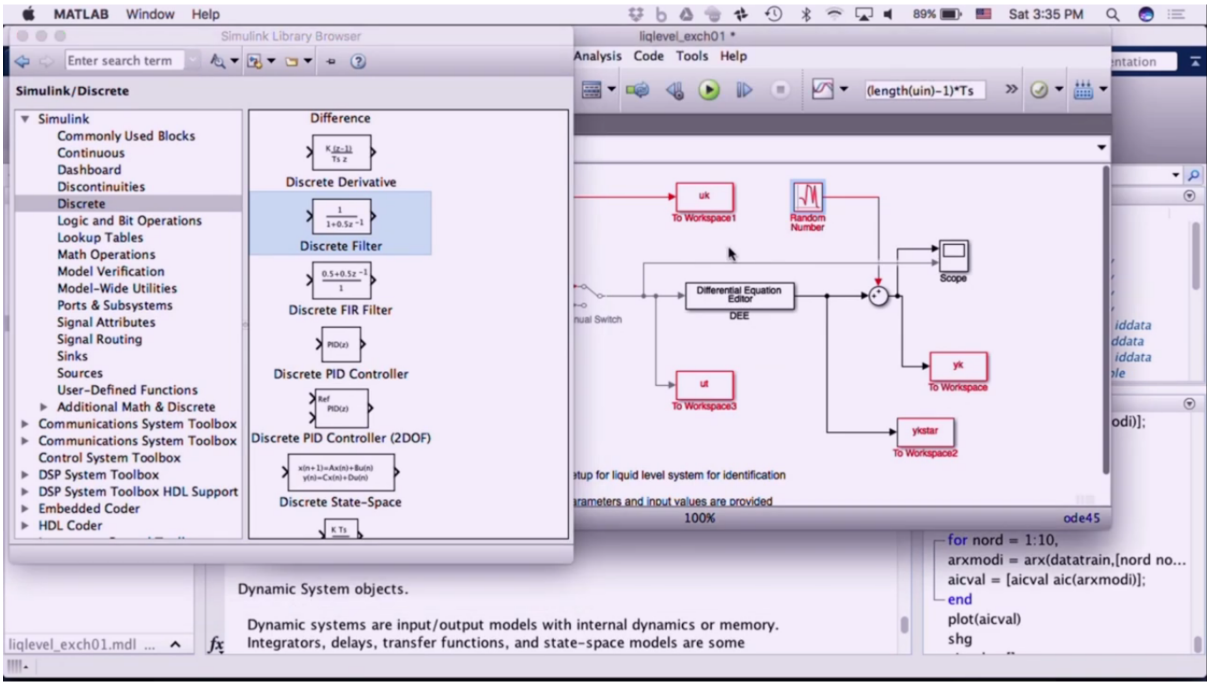
No. No. Why workspace, two workspace is also in red, right? I am sorry for the font size, but colours I am sure you can see. So you can see all these blocks are in the red, right? And where are these ones are in black or greyish black. The difference is that all these blocks that have a red colour, red is not so relevant but they are coloured, tells you that they're operating in discrete time. That means there is a sampling interval associated with them. Those blocks are going to be triggered only every Ts that you have specified. Whereas this differential equation that we have keyed in is continues time. That's going to working continues time.

When that is also kind of cheating, because it's not actually working in continues time, but it is being treated as a continuous time object and numerical integration will be applied there, right? So you can actually come up with these colours by going to I think display, Signals and Ports. Gives changing with every 2 years versions, but there is somewhere you can actually say sample time colours. Previously it used to be a sample time colours within a feature, but though there is an existing. So if I take off, if I switch off the colours when we are gone but I would prefer this to be on so that I clearly know, which ones and of course, the choice is yours depending on your preference and the mood in the day you can change the colours, but that's okay.

All right. So I personally feel that Simulink I'm not advertising for Math Works here, but I've been using Simulink, I should say for the last 20 years now. And it has remained one of the finest products that you will see for graphically simulating dynamic systems. Okay. So it offers a fantastic GUI. Any day I would prefer to use Simulink to simulate other than MATLAB. You can do all that you do in the Simulink model with MATLAB as well, but it requires a lot of typing and a lot of bookkeeping I have to imagine and then connect them and so on. But obviously we would like to do it this way because the interface is so simplifying and it just removes all the clutter that you would have in your brain, if you were to do it from command line. Okay. So this is the EK in our simulation. And you can clearly see I'm adding white noise directly to the response, right?

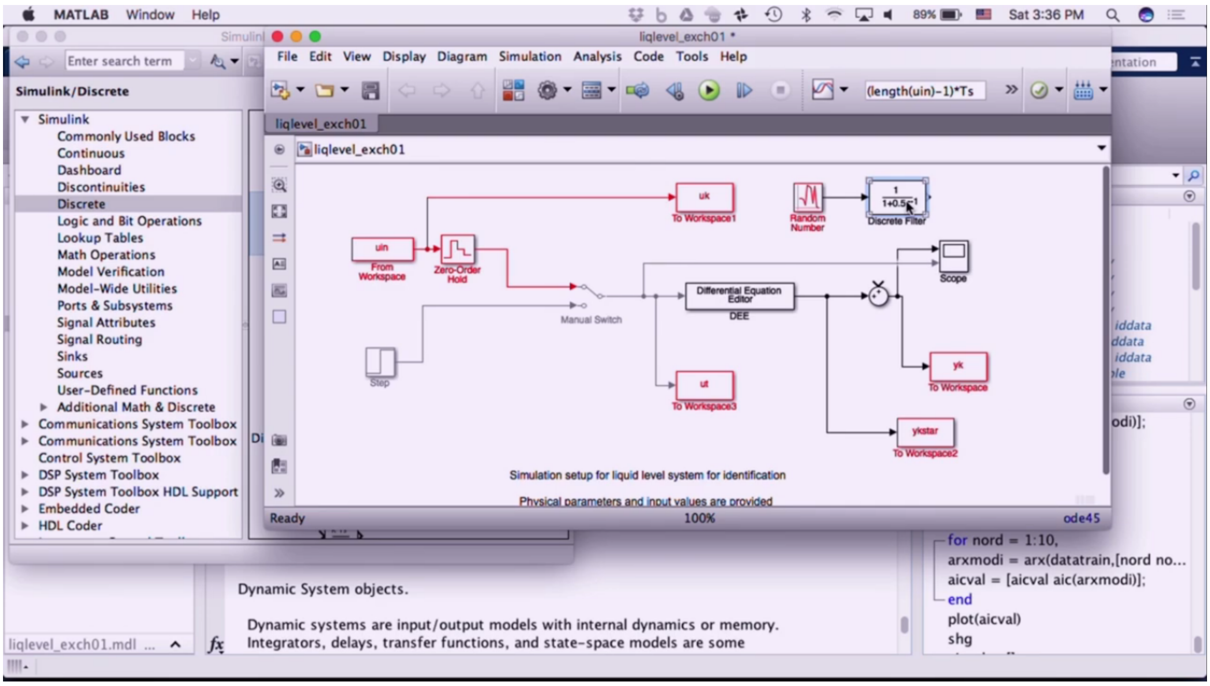
Which means the model that I have simulated the data that I have generated has come from truth plus white noise? It is not because of that, that the output error like somebody said, oh, yeah, no, no, it's cheating you probably generated data coming from an output error process and therefore OE model what?" No. That is why I've given you a question in the assignment. The last question in the assignment for you is to actually have this white noise passing through a filter. You can get a filter from again you can invoke the library. So go to library browser on the tools and get a discrete time filter, right? So if you go to discrete here. Are you able to see, at least, you know, in terms of blocks are you're having some difficult. So under tools you bring up the library browser. And then a go to discrete block set and under the discrete block set you would see this discrete filter. And I've given you the instructions in the question. Simply drag this here to your model that you have.

(Refer Slide Time: 19:23)



Did it come here? Not yet. Here. And you can place it along the path here. You can actually disconnect this. And now, there are many beautiful ways of connecting it, the only thing that's remaining is voice command. Okay. That is yet to come. But otherwise there are so many different ways of connecting it like. I don't know, I have not tried, maybe you phase. You keep a block very close to another block they get connected after a while. Because it become friends and they just cannot resist and so on. So that's also possible. I'm not right, but at least I know two or three different ways in which you can connect.

(Refer Slide Time: 20:03)



The simplest is to connect the output to the input. Okay. That as remain for long time. But you never know right, I mean with the increasing technology and so on anything is possible. All right. So you can now connect the output of the filter. So know what's happening is I'm not adding white noise, but I'm adding what is known as coloured noise. Think of it this way there is a white paper and I'm placing a filter, like in many dashing hotels they do this right. There is a tube light nice white light like this and then you add a colour paper it gives you a feeling of coloured light. That's exactly what it is. And I have given you what kind of filter to use and so on. All those instructions are given. Simulate the data. I'm sorry. Simulate the model, generate the data. Go through the same steps as we have gone through. That is fit an output error model and see if you are still able to estimate the deterministic part correctly, right? What will, of course, happen is that the output error model will not meet the noise model requirements.

See there are two checks that we conduct. One is for the adequacy of G. And the other one is the adequacy of H noise model. To check if we have obtained a good deterministic model, what do we do? We examined the cross correlation between the residuals and the input, right? Because if there are any remnant input effects in the residual that means G is calling for some refinement. And then there is another check that we perform to see if the noise model is adequate which is by looking at the auto correlation of the residuals. Only in these two tests are, the model passes these two tests, we come to the conclusion that the G and H that we have identified are good working models. You will find in this case with the output error structure. It will pass the test for G, but failed the test for H, because why? Why will it fail the test for H?

Yeah, because in the output error model. I am ignoring this filter, I'm assuming white noise enters and corrupts the response directly. Whereas that's not the case, so there is a mismatch. And that would be reflected in the autocorrelation, right? And then, of course, you have to sit and fit that model, but I'm not asking you to do that in the question. I'm just asking you to see whether the output error structure still works in this case. You understand the purpose of that question. There is a mismatch between the noise dynamics, but yet the output at a structure is suited. It is not I mean, the main properties that of the structure, but then there is also an estimation algorithm will not be able to estimation algorithm as of now. The idea is for you to understand the consequences of choosing a model structure. On finding or estimating the correct G, so-called correct G and H. Okay.

The values that are reported in the slide are the same values that I have used in the MATLAB Code that I showed you in the model workspace. Please feel free to play around and so on. One of the, I say I've mentioned this earlier also shortcomings of working with empirical or black box models is they are not scalable. You understand what I mean? Suppose I want to, I have a model that I've identified for these conditions. That is let's say, for the cross-sectional area of two units. Now I want to know what would be the response if I change the cross-sectional area to4 let us say. Are 1 or 3, it's not possible to use this black box model that we have identified to test that. Whether it whereas a first principles model will allow you to do that, because first principles model is predominantly symbolic.

All you have to do is plug in the values and then simulate. So that is one of the areas also hot areas of this how do you actually scale up or scale down black box models? And it's not so easy. Unless of course, you know the relation between the parameters of your black box model and the physical parameters that you are going to change in your design. Understand. Okay. So with that we will formally close the liquid level case study. Hopefully it has actually it has given you a lot of insights and given you a feel of what is encountered in a typical identification journey. Please do go through the bunch of questions that I have at the end of chapter two in my book, because there are…