

ELEC ENG 4CL4 – CONTROL SYSTEM DESIGN

Lab #1: MATLAB/Simulink simulation of continuous casting

Objectives:

To gain experience in simulating a control system (controller + plant) within MATLAB/Simulink.

To explore the trade-offs between system response speed and susceptibility to measurement noise in a high-gain feedback loop.

Assessment:

Your grade for this lab will be based on your ability to create a Simulink model of the continuous casting system described below and on your reporting of the results obtained with the model. The total grade will be out of 20 points. The written report will be worth 15 points, and the remaining 5 points will be based on your demonstration of your Simulink model to the TA and/or a pre-lab quiz on the theory required for this lab.

Clearly label all plots and their axes (points for style will be deducted otherwise)

DON'T COPY OTHERS BLINDLY!!

Please attend the lab section to which you have been assigned.

You should complete this lab with one lab partner. If there are an odd number of students, then one group of three will be created by the TA.

You may choose to complete the lab assignment partially or entirely in your own time (in groups preferably of two students but definitely no larger than three). However, if you choose to do this, you must show up at the start of your scheduled lab time to give the TA a brief demonstration of your MATLAB/Simulink code and model.

Each pair of students should complete one lab report together, which is to be submitted one week from the date of the lab.

Description of the continuous casting control system:

The continuous casting control system (controller + plant model) is shown in Fig. 1 below. This system is described more fully in the lecture notes and in Chapter 2 of the Goodwin et al. textbook—you should read these before attending the lab class. Additionally, an interactive Java demo of this system is available via the textbook web site or the CD that comes with the textbook—it would also be helpful to play around with this demo before attending the lab class.

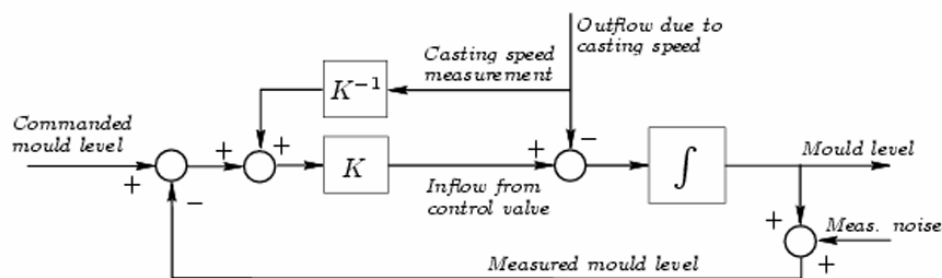


Figure 1 Continuous casting control system (controller + plant model)

1. Creating a Simulink model from Figure 1

- a. Start Matlab
- b. From the Matlab menus select “File> New> Model” to open a new Simulink model window.
- c. From the Simulink model window menus select “View> Library browser”, to open a library of Simulink blocks that can be dragged and dropped onto your model window.
- d. You will need the following blocks to simulate the system shown in Fig. 1:
 - i. A generic “Signal Generator” block for the *commanded mould level* can be found in the “Sources” section of the library, as can a “Constant” block for the *outflow due to casting speed* and a “Band-Limited White Noise” block for the *measurement noise*.
 - ii. “Gain” blocks for the *feedforward* (K^{-1}) and *feedback* (K) gains can be found in the “Math Operations” section of the library, as can “Sum” blocks for the various *summations* and *subtractions* performed on different signals.
 - iii. An “Integrator” block for the *integrator* \int (part of the plant model) can be found in the “Continuous” section of the library.
 - iv. A “Scope” block for viewing the *commanded mould level* and the actual *mould level* can be found in the “Sinks” section of the library, as can “To Workspace” blocks for sending the *commanded mould level* and the actual *mould level* to the MATLAB workspace for further plotting and numerical analysis. Note that two signals can be plotted on the same axis of the Scope by multiplexing them (using the “Mux” block found in the “Signal Routing” section of the library) and then passing the multiplexed signal into the “Scope” block.
- e. After you have drag-and-dropped all the necessary blocks onto your model window, you can change the orientation of any block by right clicking on it and selecting “Format> Rotate block” or “Format> Flip block” from the pop-up menu.
- f. The output port of a particular block (indicated by a > pointing *out of* the block) can be connected to the input port of another block (indicated by a > pointing *into* the block) by left-clicking on the output port and dragging the pointer to the output port while keeping the left mouse button held down. A short-cut for doing this operation is to left-click on the block providing the output port and to then, *while holding down the Cntrl-key*, left-click on the block with the input port.
- g. Connections between blocks can be adjusted by selecting them with a left-click and then dragging them while holding down the left mouse button. Connections can also be selected and then deleted by hitting the “Delete” key.
- h. A new branch from an existing connection can be created by left-clicking on the input port to which you want this branch to go and dragging the pointer to some desired branch point along the existing connection.

2. Setting model and simulation parameters

- a. Double-clicking on a block opens up a “Block Parameters” window, which can be used to adjust the parameters of that particular block. You will need to set the following block parameters:
 - i. Set the “Signal Generator” block parameter “Frequency” to “F” Hertz. You can then set the desired frequency of the command signal from the MATLAB command window by typing, for example, “F = 0.25” and hitting the Enter/Return key before you run a particular simulation.
 - ii. For the “Gain” blocks, set the *feedforward* block “Gain” parameter to “1/K” and the *feedback* “Gain” parameter to “K”. You can then set the desired gain of the system from the MATLAB command window by typing, for example, “K = 1” and hitting the Enter/Return key before you run a particular simulation.
 - iii. Instructions for setting the signs (i.e., pluses or minuses) for the “Sum” block are given when you double-click on the block to set its parameters.
 - iv. Set the “Noise power” parameter of the “Band-Limited White Noise” block to “Pn”. You can then set the desired measurement noise power from the MATLAB command window by typing, for example, “Pn = 0.02” and hitting the Enter/Return key before you run a particular simulation.
 - v. For the “To Workspace” blocks, connect one to the output of the “Signal Generator” block and set that “To Workspace” block’s “Variable Name” parameter to “hstar” and its “Save format” parameter to “Array”. Connect the other “To Workspace” block to the output of the “Integrator” block and set its “Variable Name” to “ht” and its “Save format” parameter to “Array”. After you have run a particular simulation, the variable “hstar” in the MATLAB workspace will contain the *commanded mould level* and the variable “ht” will contain the actual *mould level*. Simulink will place a third variable “tout” in the MATLAB workspace, which contains the time instances corresponding to the samples of “hstar” and “ht”.
- b. From the model window menus, select “Simulation> Simulation parameters...” Change the simulation stop time from “10.0” to “40.0” and then click on the “Workspace I/O” tab and uncheck the “Limit data points to last” checkbox.

3. Running simulations and viewing and saving results

- a. Save the Simulink model as “caster_mmm_nnn.mdl”, where mmm and nnn are the student numbers of the lab partners creating this model. If you are working on one of the ECE lab computers in T13, you should save your model and any data or figures you want to save in the “z:\” drive. (At the end of the lab session, you should save your files to a floppy disk or USB drive, or email them to yourself.)
- b. Before running each simulation, you will need to enter the desired parameter values for the feedback gain “K”, the measurement noise power “Pn” and the command mould level frequency “F” from the MATLAB command window, e.g.:

```
>> K = 1, Pn = 0.02, F = 0.25 [Enter]
```

where >> indicates the MATLAB command prompt and [Enter] indicates pressing the Enter/Return key.

- c. Now you are ready to run a simulation by selecting “Simulation> Start” from the Simulink model window menus or by pressing the ► icon on the Simulink model window toolbar.
- d. If you’ve done everything correctly, then the simulation should run without returning any error messages. If an error message window pops up, then read the error message(s) and try to work out what you’ve done wrong. If you cannot find the problem, ask the TA.
- e. You can now look at the simulation results by double-clicking on the “Scope” block. A “Scope” figure window should pop-up showing the *commanded mould level* and actual *mould level* (as a function of time) as two different coloured lines. If you click on the “Autoscale” icon of the “Scope” window toolbar (it looks like a pair of binoculars), the axes will be rescaled to fit the simulation data.
- f. To do any numerical analysis, create plots for your report or save the simulation results, you will need to use the MATLAB command window. For example:

```
>> figure [Enter]
```

```
>> plot(tout,hstar,'--',tout,ht,'-') [Enter]
```

will open a new figure window and plot the *commanded mould level* as a dashed line and the actual *mould level* as a solid line (versus time) together on a single axis, and:

```
>> RMSE = sqrt(mean((ht-hstar).^2)) [Enter]
```

will calculate the root mean squared error (RMSE) between the *commanded mould level* and the actual *mould level*.

Plots can be saved as MATLAB figure (.fig) files and can be copied and pasted into word processor (e.g., MS Word) documents for your report. Results can be saved as MATLAB data (.mat) files—you will need to give a unique filename for each simulation result that you want to save.

4. The report

- a. Your report should include a brief description of the continuous casting system and a figure showing the schematic of your Simulink model—you can copy and paste the schematic from your Simulink model window directly to a word processor document.
- b. Next you should provide some example figures and RMSE values for simulations run with different parameters, give brief written descriptions for each simulation. Start with $K = 1$, $P_n = 0$ (i.e., with no measurement noise) and $F = 0.25$. Describe the behaviour of the actual *mould level* compared to the *commanded mould level*.
- c. What happens if you increase K to 6?
- d. Now add some measurement noise by setting $P_n = 0.02$. How well does the control system handle this amount of measurement noise? Does the actual *mould level* sometimes overshoot the maximum and minimum *commanded mould levels*—this could correspond to the mould dangerously overflowing or emptying out!
- e. What if you increase P_n to 0.08? Try some different values of K between 0 and 6? What value appears to give the best trade-off between obtaining good response speed and accuracy and avoiding overshoots of the maximum and minimum *commanded mould levels*?
- f. Open the “Signal Generator” block parameters and change the signal type from “sine” to “square”, save the model, and repeat the above simulations (b to e) with this new signal type. Describe how the system behaviour with the square-wave signal is similar to and different from the system response with the sine-wave signal for each set of parameters.