Multivariable Control (ME-422) - Exercise session 4

Prof. G. Ferrari Trecate

1 Model Order Reduction: Theory

Consider a discrete time-invariant state-space model, given by:

$$G : \begin{cases} x^+ &= Ax + Bu \quad , \quad x(0) = x_0, \\ y &= Cx + Du, \end{cases} \tag{1}$$

where $x \in \mathbb{R}^n, u \in \mathbb{R}^m, y \in \mathbb{R}^p$. The *model order* is the dimension of x, denoted by n, and is a measure of the state-space model's complexity. It is more challenging to analyze, understand, and simulate higher-order models, i.e., models with a larger n. It is also easier to design a controller for a simpler model of a lower order. Therefore, approximating a high-order model with a reduced-order one while preserving some pivotal model characteristics can make working with it more convenient. This technique is called *model order reduction.*

A reduced order system is a state-space system G_r:

$$G_r : \begin{cases} z^+ &= A_r z + B_r u \quad , \quad z(0) = z_0, \\ y_r &= C_r z + D_r u, \end{cases} \tag{2}$$

such that $z \in \mathbb{R}^r$, where $r < n$, and $y_r \in \mathbb{R}^p$. Note that the same input u is applied to both G and G_r.

A good reduced-model, G_r, produces trajectories y_r which are *close* to the trajectories of the original model, y. Depending on the criterion used to evaluate the resemblance between G and G_r, several MOD methods have been developed. In this exercise, we study two such methods, both done in two steps:

Step 1: Project the state-space coordinates x into $\overline{x}$ using $x = T\,\overline{x}$. That is, find a suitable invertible matrix $T \in \mathbb{R}^{n\times n}$, and transform the state-space model

$$\overline{A} = T^{-1}\,A\,T = \begin{bmatrix} \overline{A}_{11}, \overline{A}_{12} \\ \overline{A}_{21}, \overline{A}_{22} \end{bmatrix},\ \overline{A}_{11} \in \mathbb{R}^{r\times r}, \qquad \overline{B} = T^{-1}\,B = \begin{bmatrix} \overline{B}_1 \\ \overline{B}_2 \end{bmatrix},\ \overline{B}_1 \in \mathbb{R}^{r\times m},$$

$$\overline{C} = C\,T = \begin{bmatrix} \overline{C}_1, \overline{C}_2 \end{bmatrix},\ \overline{C}_1 \in \mathbb{R}^{p\times r}, \qquad \overline{D} = D.$$

Selecting the transformation T depends on the employed method.

Step 2: Partition the state-space coordinates into $\overline{x} = (\overline{x}_1, \overline{x}_2)$, where $\overline{x}_1 \in \mathbb{R}^r$ and obtain G_r by selecting $A_r = \overline{A}_{11}, B_r = \overline{B}_1, C_r = \overline{C}_1, D_r = D$. Accordingly, the new states are $z = \overline{x}_1$.

Method 1: dominant pole approximation This approach focuses on producing a reduced-order model G_r that resembles the steady-state behavior of G. Recall that state and output at time step k of a stable model can be expressed as a weighted sum of system's modes. For eigenvalues close to the origin, the corresponding mode quickly goes to 0 as the time increases. Therefore, these eigenvalues primarily affect the transient response and have a negligible effect on the steady-state behavior.

The dominant pole approximation method assumes that the slowest mode(s) of the system dominates the response and accordingly, ignores faster mode(s). When all eigenvalues are real, a model of reduced order r is obtained by projecting the state space on the subspace spanned by the eigenvectors corresponding to the r slowest (dominant) eigenvalues. For simplicity, we do not discuss the case for complex poles.

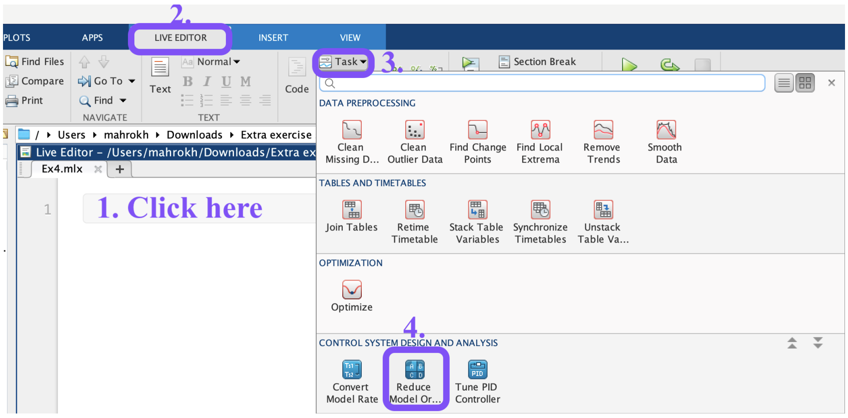

Figure 1: Starting the `Reduce Model Order` tool. Click on an empty code section in your live script. Then, from the top bar, select `LIVE EDITOR` and click on `Task`. Under `CONTROL SYSTEM DESIGN AND ANALYSIS`, double click on `Reduce Model Order`. If you do not see this field, make sure that the `CONTROL SYSTEM DESIGN AND ANALYSIS` library is installed correctly. The tool will start in the empty code section that you have selected.

Method 2: balanced truncation Balanced truncation aims at obtaining G_r which is close to G in terms of the amount of energy transferred from the input into the output. To achieve this, states of G with relatively small energy contribution are removed. Balanced truncation identifies state coordinate directions that are i) very hard to excite with finite amounts of input energy, and ii) do not result in much energy in the output. For checking (i) and (ii), all states must be **controllable** and **observable**.

2 Exercises

We will use the `Reduce Model Order` tool in Matlab for experimenting with the introduced order reduction techniques. In this section, we solve a series of exercises to familiarize ourselves with dominant pole approximation and balanced truncation functionalities of this tool. Download the data file `Ex4-data` from Moodle. Open a new live script and load the data using the command `load('Ex4-data')`. The file contains two discrete-time state-space models, `G1` and `G2`. System matrices can be accessed through `G1.A`, `G1.B`, $\cdots$ or viewed by double clicking on `G1` or `G2` in Matlab Workspace.

1. Find the number of states, inputs, and outputs for `G1` and `G2`.

2. Calculate eigenvalues of `G1` and `G2` in Matlab. Are `G1` and `G2` stable? Categorize the eigenvalues as being fast or slow.

3. Follow instructions in Fig. 1 to launch the tool and set fields according to Fig. 2. The `Mode Selection` method retains the most effective modes in a frequency region you specify. To target the slower part of the response, set the lower frequency bound to a very small value. The upper frequency bound distinguishes between slow and fast modes. Use the values in Fig. 2 to obtain a first-order approximation of `G1`, named `G1DP`. Compare the impulse response of the full and reduced order models, `G1` and `G1DP`.

4. According to the theory, what is the eigenvalue of the first-order approximated system, `G1DP`? Find the eigenvalue of `G1DP` using Matlab and check if the result matches your expectation.

5. In a new live script section, start another `Reduce Model Order` tool and load `G2` in it. Set the frequency bounds to get a first-order system and call it `G2DP`. Compare the impulse response to *exercise 3*. First-order reduction approximates which system, `G1` or `G2`, better? Why?

6. In part 5, slowly extend the frequency upper bound to increase the order of the reduced system (you can check the order in the legend of the impulse response). Observe that eigenvalues of `G2` will

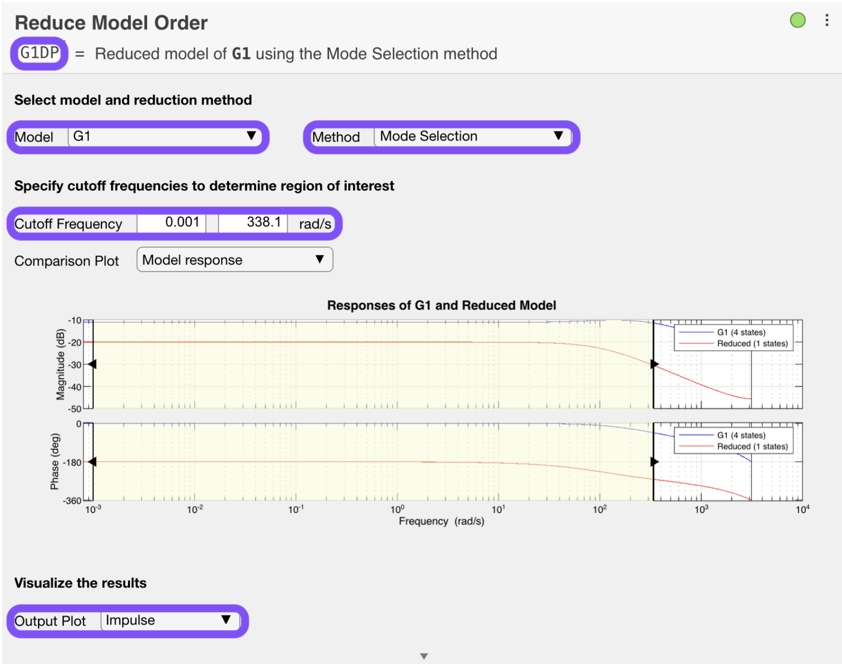

Figure 2: Setting the fields for dominant pole approximation. Select the full-order model and set the method to `Mode Selection`. Name the reduced-order model `G1DP`. Specify the frequency range as $[0.001, 338.1]$. The reduced-order system will appear in your Matlab Workspace with the indicated name, `G1DP`, and the impulse response of `G1` and `G1DP` will be plotted. Note that changing the fields, for instance, the frequency range, overwrites `G1DP`. So, you should either change the reduced-order model name or copy `G1DP` in another Workspace variable before changing the fields if you wish to keep `G1DP`.

get involved in the reduced system, `G2DP`, in order of magnitude. Describe the effect of extending the frequency upper bound on the impulse response.

Next, we perform balanced truncation, which is applicable to controllable and observable systems.

7. Check controllability and observability of `G1` and `G2`.
8. Set up balanced truncation according to Fig. 3. Pick `G1` as the full-order model and call the reduced model `G1BT`. Start with setting `Reduced Order` to 1 and increase it gradually one at a time. Monitor the error reduction in the impulse response.
9. Obtain a first-order reduced model of `G2` through balanced truncation and call it `G2BT`. Plot the impulse responses of `G2BT` and compare it to `G2DP` obtained in question 5. Which method, dominant pole approximation or balanced truncation, performs better? Why?

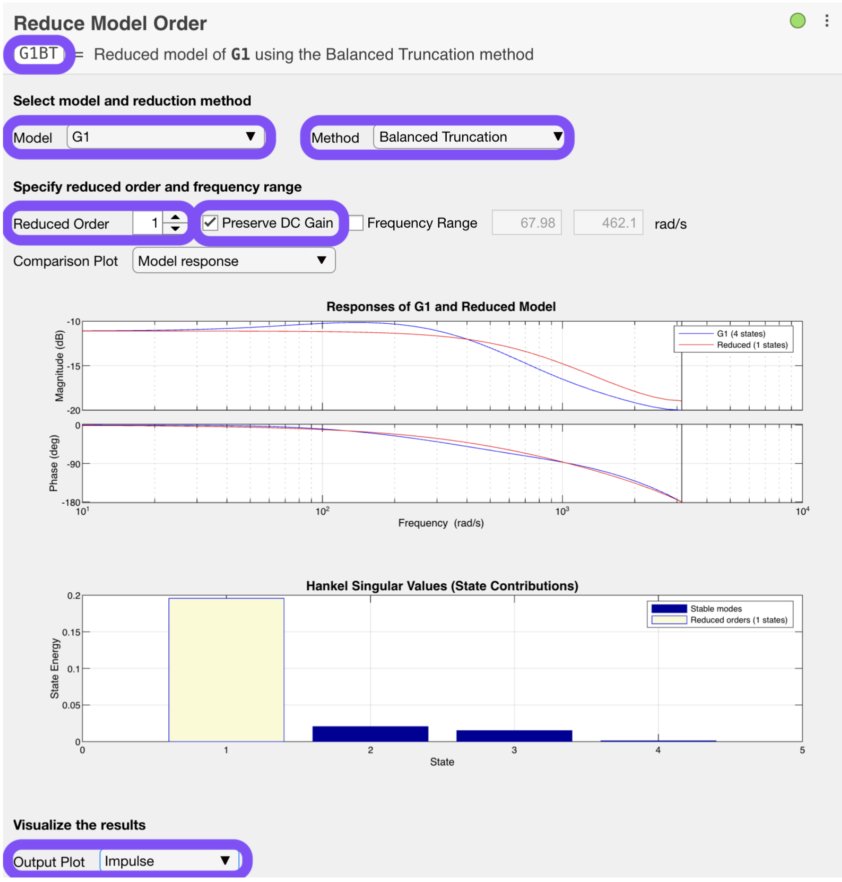

Figure 3: Setting the fields for balanced truncation. Select the full-order model and set the method to `Balanced Truncation`. Name the reduced-order model `G1BT` and set the reduced order to 1. Indicate to preserve the DC gain. Select impulse as the output plot. The reduced-order system will appear in your Matlab Workspace with the name `G1BT`, and the impulse response of `G1` and `G1BT` will be plotted.