MCE 484
Sample System Identification Toolbox Session

This document serves as a computer lab guide and a summary of the main operations to be followed to arrive at a transfer function estimate.

1. **Starting the toolbox:** Type `ident` at the Matlab prompt.

2. **Checking the data:** In the lab session, download the file `chirpdata00.mat` and place it on your working directory. Make sure the current Matlab directory is the same as your working directory. Examine the data by plotting input and output against time. Determine the sampling interval by taking the difference of consecutive time points: `>>Ts=t(2)-t(1)`. Check the length of the data: `length(t)`. Note that long data records (hundred thousand points or more per variable) will be handled poorly in conventional PCs and are probably unnecessary. Data resampling (thinning out) may be needed in some cases).

3. **Importing the data:** Use the import function (time domain) and enter the input and output names `u` and `y` in the dialog box. Name the data set and enter a starting time of 0 and a sampling interval of `Ts`. You should see the data set as one of the boxes in the Data View section.

4. **Displaying the data:** Use the Time Plot and Data Spectra check boxes to see the data in the time and frequency domains. Close the 2 windows after inspecting the data.

5. **Pre-Processing: Remove means** It is often necessary to refer the data to a zero level. Use Preprocess/Remove means to achieve this. A new data record appears, with a different color (if you used the default name `mydata`, the modified data will be `mydatad`). Drag the new data to the Working Data box.

6. **Pre-Processing: Filter** Filtering is usually required when working with experimental data. You should have previously conducted FFT analysis to determine the range of frequencies where the noise is most significant. Other reasons to filter are to limit the bandwidth of the model and to accomodate low sample rates. In the lab example, FFT analysis reveals that the noise is broadband (not concentrated at a single frequency). Our criteria for filtering are the sampling rate (as a rule of thumb, the sampling frequency should be at least 5 times the desired model bandwidth). With $f_s = 1$ kHz, our model should be accurate up to 200 Hz. This should also be the frequency used to filter the data (low-pass). Use Preprocess/Filter, switch

to Hz under Options (if desired) and enter a pass band of 0.01 to 200 Hz or draw a window with the mouse. Then press Filter and Insert. You should see a new data set appearing in Data Views. Drag the filtered data to the Working Data and Validation Data boxes. You may close the Filter window.

7. **Estimating a Transfer Function:** Use Estimate/Parametric Linear Model. There are several options (templates) for discrete transfer functions and other types of models. Here we use the Output Error structure. Select OE. The Orders box shows the selected orders (zeroes, poles and lags for the noise). This is an iterative process guided by a quality-of-fit score. Try the default (2 2 1), press Estimate and wait for the process to complete (a data set will appear in the Model Views section).

8. **Viewing the Quality of Fit:** Press Model Output. A window will appear showing the original and predicted outputs along with a score. Judge the quality of the fit visually and by score. Keep in mind that scores higher than 60 are usually good fits when assessed visually.

9. **Viewing Model Properties:** You can display a pole-zero map to check for stability (in discrete transfer functions, all poles must fall in a circle of radius 1 centered at the origin. The zeroes don't count for stability). You can also see the Bode plot and the step response.

10. **Iterating:** Guided by system physics when available, iterate on the number of zeroes and poles until you obtain a good compromise between score and model order. Sometimes changing the filtering range has a large impact. Re-do the filtering starting with the mean-free data and use a frequency range from 0.01 to 20 Hz.

11. **Exporting the model :** When satisfied with the model, export it to the workspace. By dragging the model data set to the Workspace box. If needed, save your session for future reference. The file will have extension `sid` and it will contain the data sets (no need to import).

12. **Converting to continuous time:** After exporting, the workspace will contain a variable (`oe221` for instance). This is a discrete-time transfer function. To convert it to continuous time, use `sysCT=d2c(oe221)`.

13. **Evaluating the results:** Sometimes, the continuous-time model contains poles or zeroes which are well-beyond the intended bandwidth. A reduction is performed using the pole dominance criterion studied in MCE441. Special care needs to be placed in maintaining the DC gain of the original and reduced models ($G(0)$ value).

If necessary, the reduced transfer function can be placed in a Simulink model and driven with the same input used for the system identification process. The output produced by the reduced continuous-time transfer function should be close to the output data used for system identification.