MATLAB based GUI for post-processing experimental data of superelastic tyres: User Guide

Jiho Yang
M.Sc. student in Computational Science and Engineering
Department of Informatics
Chair of Scientific Computing
Technical University of Munich

July 1, 2016

Abstract

This document provides the manual for using MATLAB based Graphical User Interface (GUI) for post-processing experimental data, and some of its theoretical details. This work has been conducted as a student job (HiWi) under Dipl.-Ing Seungyong Oh from Chair of Materials Handling, Material Flow, and Logistics, Department of Mechanical Engineering, Technical University of Munich, for post-processing experimental data of superelastic tyres. Since this GUI has been developed specifically for a particular problem, by no means it holds functionalities as a generalised data post-processing tool, although it can be used as a template for extending it for other applications.

1 Introduction

There are 3 modules for each problem case: Hydropulser, Trommel, and Forklift. Each module has different functionalities implemented, although many of them overlap. All of these modules essentially aim to do the following tasks: 1. Data import, 2. Plotting of the data, 3. Data processing (e.g.

Damping coefficient computation, etc), 4. Data export.

Please note that not all the details of functions used are described here. For further details, one should refer to the actual code. However, some details of the theories/methodologies implemented for data processing are briefly described in the document.

2 Hydropulser Module

Within the Hydropulser folder, please use the codes inside "Single file multiple variable calibration" directory.

2.1 Outline of functionalities

1. Data import (.txt file for Experimental data, and .dat file for Simulation data)

2. Plotting of data for both Experimental and Simulation data

3. Simulation data calibration (shift in x and y axis, and scale in y axis) for comparison with Experimental data

4. Reset calibrated simulation data to its original values

2.2 Manual

1. Run faml_Hydropulser_GUI.m

2. Import experimental data using "Browse" button next to "Experimental Results". This will import the experimental data and plot the following: Force vs Time, Displacement vs Time, and Displacement vs Force in both GUI environment and figures.

3. Import simulation data using "Browse" button next to "Simulation Results". The imported data file should hold the following data (in column order): 1. Time, 2. Force, 3. Displacement, 4. Some data, 5. Some data. "Some data" is user specified when generating simulation data. It will then generate the same plots as the experimental data on different plot panels.

4. Import up to 4 simulation data files for comparing multiple simulation data set with experimental data. The number of simulation data files is strictly limited (although if one wishes to import more data GUI will allow it, but will only calibrate the first 4 simulation dataset. Newly imported simulation data will be plotted on the same plot panels with corresponding legends. These dataset will be denoted as T1-4.

5. Select the simulation dataset (from T1-4) one wishes to calibrate (please make sure to select only one at a time - this applies to all the other cases).

6. Select the plot to be compared (Force vs Time, Displacement vs Time, or Displacement vs Force)

7. Select the operation (shift in y, x, or scale in y)

8. Insert the calibration value

9. Using "Calibrate" button, perform calibration. This will calibrate the data accordingy and regenerate the calibrated simulation data on the same plot panels as experimental data. The net calibration value for each operation (shift in y, x or scale in y) will be displayed in the legends. When shifting in x axis, in particular, it is recommended to use small calibration value to avoid the calibrated simulation data plot going out of the domain (it will generate error message).

10. If one wishes to reset the calibrated simulation data, push "Reset" button. This will reset all the calibrated dataset (T1-4) and plot the original simulation dataset. Please note that this operation resets all the dataset (i.e. it won't reset just one data set, say, T1).

11. If one wishes to bring new simulation dataset, the GUI needs to be reopened. Unfortunately, reset functionality for imported dataset is not currently implemented

3 Trommel Module

3.1 Outline of functionalities

1. Data import (.asc file for Experimental data, and .dat file for Simulation data)
2. Plotting of data for both Experimental and Simulation data
3. Calibrate displacement to obtain deflection
4. Compute damping coefficient (from previously computed deflection - experimental) for user specified velocity and force values
5. Plot damping coefficient vs velocity
6. Export data (curvature of exponentially decaying vibration - delta, velocity, and damping coefficient) for user specified force value
7. Calibrate deflection data from simulation to compare with previously computed deflection (experimental)
8. Reset calibrated simulation data to its original values

3.2 Manual

1. Run faml_Trommel_GUI.m
2. Import experimental data using "Browse" button next to "Experimental Data". This will import the experimental data and plot the following: 4th Sensor Displacement vs Time, 5th Sensor Displacement vs Time, and 6th Acceleration vs Time in both GUI environment and figures
3. Insert the offset and correction factor value, and press "Calibrate" button. This will compute and plot Deflection vs Time.
4. Insert velocity and force values, and press "Calculate Damping" to calculate damping coefficient. This will take the average of each excitation cycle of deflection data (computed in the previous step), compute damping coefficient, and plot the followings: Averaged deflection, Peak

values, and exponential curve regenerated from the damping coefficient. The data computed will be displayed in the table

5. Repeat Step 2 - 4 until you get sufficient amount data for a given force (wheel load)

6. Plot Damping coefficient vs Velocity graph by using “Plot Damping”. Make sure to plot the graph once you have all the data needed for a given force (otherwise you may need to restart the program). If you accidently computed damping coefficient with wrong velocity and force values, use “Remove last row” button to remove this. You may also use “Reset” button to clean up the computed data. Also, do not close the figure with this plot (currently, this is figure 7) since you will need it for plotting next graphs.

7. Repeat Step 2 - 6 until you get all the Damping coefficient vs Velocity graphs

8. Once you have all the damping coefficients for a given force value, use “Export Data” button to export the computed data. The following column convention is set (in column order): 1. Delta, 2. Velocity, 3. Damping coefficient. The data will be stored in the folder named “Exported Data_MATLAB” in the same directory. The name of the file will be: TP_F_⟨force value⟩N.txt

9. Import simulation data using “Browse” button next to “Simulation Data”. This will import the deflection data obtained from the simulation.

10. Calibrate simulation data for comparing it with experimental data following the same procedure as Step 6 - 10 in Section 2.2

3.3 Some technical details

Deflection calculation

Deflection (experimental) is calculated using the following equation:

$$Deflection = (Displacement + Offset) \times Calibration\ Factor$$

Damping coefficient calculation

Damping is calculated by assuming that the oscillatory motion follows an exponential function:

$$y = ce^{-\delta t}$$

where y is the deflection, c is some system related constant, δ is the curvature, and t is time. By taking natural logs, this expression is linearised:

$$\ln(y) = -\delta t + \ln(c)$$

By taking the peak values the averaged deflection cycle and its corresponding time values, this linear equation can be achieved by some least-square curve-fitting method (*polyfit* was used here). It must be noted, if one wishes to extend the code in any way, that when taking the reverse log of this linearised equation (e.g. for plotting), it is best to avoid using *polyval* since the *polyfit* - *polyval* combination causes numerical instability (pre-formulated mathematical expression is used in this GUI). Once the curvature is computed, damping coefficient is computed with the following equation:

$$d = 2\delta m$$

where d is the damping coefficient, and m is mass.

4 Forklift Module

4.1 Outline of functionalities

1. Data import (.asc file for Experimental data, and .dat file for Simulation data)

2. Plotting of data for both Experimental and Simulation data

3. Data filtering and quadrature (acceleration to displacement)

4. Simulation data calibration (shift in x and y axis, and scale in y axis) for comparison with Experimental data

5. Reset calibrated simulation data to its original values

6. Export data (displacement vs time)

4.2 Manual

1. Run faml_Forklift_GUI

2. Import experimental data using "Browse" button next to "Experimental Data". This will import the experimental data, filter acceleration data, and compute displacement from acceleration data. Then the following plots are generated: 1st Sensor Acceleration (filtered) vs Time, 2nd Sensor Acceleration (filtered vs Time, 4th Sensor Velocity vs Time, 1st Sensor Displacement (computed) vs Time, and 2nd Sensor Displacement (computed) vs Time in both GUI environment and figures

3. If one's not satisfied with the quality of filtered acceleration value, insert high pass and low pass cut off frequency values (1 and 30 are set as default, respectively) and then press "Update". This will filter acceleration data and compute displacement with new cut off frequencies

4. Import simulation data using "Browse" button next to "Simulation Data". The simulation data to be imported must have the following data available (in column order: 1. Time, 2. 1st Sensor Acceleration, 3. 2nd Sensor Acceleration, 4. 1st Sensor Displacement, and 5. 2nd Sensor Displacement. This will then import the simulation data and plot the following plots on the same plot panels as experimental ones: 1st Sensor Acceleration vs Time, 2nd Sensor Acceleration vs Time, 1st Sensor Displacement vs Time, and 2nd Sensor Displacement vs Time

5. Calibrate simulation data for compairing it with experimental data as previously described

6. Again, "Reset" function is available to clean up the calibrated simulation data

7. Insert force, velocity, and threshold values, and use "Export displacement" button to export the computed displacement (experimental) data. This will generate a folder named "Exported Data_MATLAB" in the same directory. The following file name convention is used: Exp_Stapler_f_$\langle force\ value\rangle$_v_$\langle velocity\ value\rangle$_h_$\langle threshold\ value\rangle$.dat

4.3 Some technical details

Two main aspects are covered in this section: Acceleration data filtering and displacement computation. Both of them are implemented in "faml_acc2disp.m", so please refer to this function for the actual code lines.

Acceleration filtering

Low pass butterworth filter is used for acceleration filtering. Constant sampling rate is computed from the time values and low pass cut off frequency is specified by the user (default is set as 1). "butter" MATLAB built-in function is used for this task. The GUI lacks in computation of correct cut-off frequency, but is left as a manual job for the user.

Displacement computation

Cumulative trapezoidal quadrature rule is used for displacement computation using "cumtrapz" MATLAB built-in function. Due to low frequency errors generated from quadrature, high pass filters are applied after each quadrature step. Again, butterworth filter is used for this and high pass cut off frequency can be specified by the user (default is set as 30). In summary, following steps are taken for displacement computation:

1. Low pass filtering on acceleration
2. Apply cumulative trapezoidal quadrature rule to acceleration data: velocity values are obtained
3. High pass filtering on computed velocity data
4. Apply cumulative trapezoidal quadrature rule to velocity data: displacement values are obtained
5. High pass filtering on computed displacement data