

# Slosh Testing

---

This repository includes the following MATLAB scripts and functions.

## Scripts:

---

- `calibration.m` : Finds the K96 and K66 matrices to convert sensor voltages to system's forces and torques. This is done using the LabVIEW data collected from the 3 three-axis force sensors after performing the static system calibration of the system. For more information, see the 'calibrationREADME' file.
- `postprocessing.m` : This script reads LabVIEW CSVs containing the 9 voltages corresponding to the three axis of the three Kistler force sensors and uses the K96 matrix to calculate the net forces and torques of the system. Then it reads the empty tank data and finds the slosh forces caused by the liquid. For more information, see the 'postprocessREADME' file.

## Functions

---

- `diagramcreate.m` : Plots a top-down view of the position of the force sensors on the triangular plate and shows the 3 forces acting on each sensor. It also shows the non-calibrated net forces and torques of the system. It takes the following inputs:
  - (Array) **fdata**: Array including the forces from the force sensors in the format [Fx1,Fx2,Fx3,Fy1,Fy2,Fy3,Fz1,Fz2,Fz3]
  - (Vector) **startpts**: Start points of the load steps.
  - (Vector) **endpts**: End points of the load steps.
- `filter1.m` : Filters force data using a Type II Chebyshev filter and returns the filtered data. It takes the following arguments:
  - (Float) **cutoff**: Cutoff frequency of the filter in Hz.
  - (Int) **order**: The order of the filter.
  - (Int) **fs**: Sampling rate in Hz.
  - (Vector) **dataVec** and the detrended data as arguments.
- `fitline.m` : Creates a linear fit of the s-sensor force data and Kistler sensors force data and returns the slope as K coefficients. It takes the following arguments:
  - (Vector) **x**: S-Sensor force data.
  - (Vector) **y**: Kistler sensors force data.
  - (Vector) **weights**: Weights for the fit.
  - (Boolean) **plotBool**: Whether to create plots of the linear fit.
- `getCalibratedLoadsK96.m` : Returns the time array, and an array with the 6 calibrated loads (Fx, Fy, Fz, Tx, Ty, Tz) using the 9 sensor forces (which it filters using `filter1.m`) and the K96 matrix. It takes the

following arguments:

- (Array) **K96**: 9x6 Matrix obtained with `calibration.m`.
- (String) **filePath**: Path to the CSV file obtained with LabVIEW.
- (Vector) **filterParams**: Parameters of the filter in a vector ([Sample Rate in Hz, Cutoff Frequency in Hz, Order of the filter]).
- (Float) **sf**: Scaling factor to apply to the 9 forces before using them with K96 (if necessary).
- **getThFreqAccDoubleAmpFill.m** : Returns a vector containing the target frequency in Hz, acceleration in g's, double amplitude in mm, and tank fill percentage. This is achieved by reading an Excel spreadsheet named "FileNames.xlsx" that contains the tabulated names of the tests according to these parameters. It takes the following arguments:
  - (String) **fileName**: Name of the test (e.g., "test124").
  - (cell) **fileNamesCell**: Cell containing the data read from "FileNames.xlsx".
  - (Boolean) **logParams**: Whether to print to the console the parameters of each file that is being processed (true/false).
- **findStartEndCycles.m** : Returns the locations of the peaks found in a dataset. It takes the following arguments:
  - (Array) **ftArray**: Array of the calibrated forces and torques obtained with `getCalibratedLoadsK96.m`.
  - (Float) **thFreq**: Target frequency of the test in Hz, obtained with `getThFreqAccDoubleAmpFill.m`.
  - (Int) **sr**: Sample rate in Hz.
  - (Boolean) **plotCycles**: Whether to plot the dataset showing the position of the peaks.
- **getSingleEmptyTankCycle.m** : Returns an array including datapoints for a single cycle of an empty tank log. It takes the following arguments:
  - (Array) **K96**: 9x6 Matrix obtained with `calibration.m`.
  - (Int) **dataPtsPerCycle**: Number of datapoints that a single cycle should have.
  - (Float) **thFreq**: Target frequency of the test in Hz, obtained with `getThFreqAccDoubleAmpFill.m`.
  - (String) **emptyTest**: Path to the CSV file of the empty tank test. This string is formed using the parameters obtained from `getThFreqAccDoubleAmpFill.m`.
  - (Vector) **filterParams**: Parameters of the filter in a vector ([Sample Rate in Hz, Cutoff Frequency in Hz, Order of the filter]).
- **reshapeEmptyData.m** : Returns the empty tank data array trimmed to the same size of the filled tank data array, and with its peaks aligned. It takes the following parameters:
  - (Array) **emptyData**: Array containing the empty tank single cycle data repeated to be greater or equal in size to the filled tank data.
  - (Array) **filledData**: Array containing the filled tank data. When used in `postprocessing.m`, this parameter is set to the 'Steady State' loads, which is x% of the total cycles that are considered to be in steady state.
  - (Float) **theFreq**: Target frequency of the test in Hz, obtained with `getThFreqAccDoubleAmpFill.m`.
  - (Int) **sr**: Sample rate in Hz.
- **createTFplots.m** : Creates and/or saves a tiled layout of the calibrated forces and torques. It takes the following parameters.

- (Vector) **tArray**: Time vector in seconds.
- (Array) **ftArray**: Array of the calibrated forces and torques.
- (Int) **sr**: Sampling rate in Hz.
- (Int) **percentageToPlot**: Percentage of the total data to trim the x axis of the plot (0-100).
- (Int) **testN**: Test number.
- (String) **testType**: String for formatting the plot title depending on what is being plotted (e.g., "System", "Slosh").
- (Vector) **loadsToPlot**: Vector including the loads to include in the layout (where [1,2,3,4,5,6] corresponds to [Fx,Fy,Fz,Tx,Ty,Tz]). (E.g., If plotting all 6: loadsToPlot=[1,2,3,4,5,6], if plotting only Fx and Ty: loadsToPlot=[1,5]).
- (Boolean) **plotBool**: Whether to create a window displaying the loads. If looping for several tests, it's recommended to set it to 'false'.
- (Boolean) **saveBool**: Whether save the figure as a 'jpeg' image to storage. If set to 'true', it saves the images to './Plots/'.
- **processAcceleration.m**: Returns and creates/saves a plot of the acceleration. It takes the following parameters.
  - (String) **csv\_file\_path**: Path to the CSV file containing the time and acceleration data.
  - (Vector) **filterParams**: Parameters of the filter in a vector ([Sample Rate in Hz, Cutoff Frequency in Hz, Order of the filter]).
  - (Int) **testN**: Test number.
  - (Boolean) **plotBool**: Whether to create a window displaying the plot. If looping for several tests, it's recommended to set it to 'false'.
  - (Boolean) **saveBool**: Whether save the figure as a 'jpeg' image to storage. If set to 'true', it saves the images to './Plots/'.