

# Slosh Testing

---

This repository includes the following MATLAB scripts and functions:

- `main.m`: Finds the K matrix to go from sensor voltages to system's forces and torques. This is done after performing the static system calibration of the system using LabVIEW data collected from the 3 three-axis force sensors.
- `sample_postprocess.m`: This script reads LabVIEW CSVs containing the 9 voltages corresponding to the three axis of the three Kistler force sensors, and uses the K matrix to calculate the net forces and torques of the system.
- `diagramcreate.m`: This is a MATLAB function that plots a top-down view of 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 X input (S-Sensor force data), Y input (Kistler sensors force data), start points and endpoints of the load steps.
- `filter1.m`: This is a MATLAB function that filters force data using a Type II Chebyshev filter. It takes the cutoff frequency (in Hz), the order of the filter, sampling rate (Hz), and the detrended data as arguments.
- `fitline.m`: This is a MATLAB function that 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 X input (S-Sensor force data), Y input (Kistler sensors force data), the weights, and a boolean that determines if the function should (or not) create a plot of the linear fit.

## `main.m`

This script processes the static system calibration data obtained with LabVIEW to calculate the K-matrix.

### Section 1: Set parameters

This section is used to set parameters and flags for the script to use in the following sections.

1. If using the script to test a single CSV, set `singleFileBool` to `true`. Otherwise, set it to `false`.
2. If `singleFileBool` was set to true, set `single_idx` to the desired CSV index according to:

```
[1,2,3,4,5,6,7,8,9,10,11,12] = [Fx+,Fx-,Fy+,Fy-,Fz+,Fz-,Tx+,Tx-,Ty+,Ty-,Tz+,Tz-]
```

3. Set `prefixfilename` to the directory where the CSV files are located. The default value assumes that it's 'FinalData', which is inside the same directory as this script. If this is not the case, change this variable's value.
4. Type the names of the CSV files in the `csvarray` array according the order mentioned in step 2.

If running the script for a single file, name the file in the correct index.

5. The `caseflag` array contains 12 booleans (for the 12 tests performed) that determine if the tests were done by using low-high steps `caseflag=0` (achieved by using weights), or high-low steps `caseflag=1` (achieved by using the crane).

6. The `calsensorflag` array contains 12 values (for the 12 tests performed) that determine the s-sensor calibration number that was used. Use **1** for sensor 1, **2** for sensor 2, **3** if both sensors were used for torque, and **4** if both sensors were used for force.
7. The `sf` variable is a 9x12 matrix containing the scaling factor values for the x, y, z components for the 3 sensors over the 12 experiments. The row follows the following order: [Fx1,Fx2,Fx3,Fy1,Fy2,Fy3,Fz1,Fz2,Fz3]

By default, it's a matrix of 1's. **ONLY** change this if the scaling factors entered in LabVIEW didn't match the values of the charge amplifiers during data collection.

8. The lines following `sf` and before **Section 2** contain constants and don't need to be changed.

## Section 2: Read CSV files

This section is used to read the CSV files and store the 9 forces obtained by the three 3-axis force sensors in `Ftablefs` and the calibration forces recorded by the s-sensors in `FtableCS`. If `singleFileBool` was set to `true`, it will only iterate once for the CSV file corresponding to `single_idx`.

## Section 3: Plot for Unknown Starting/Ending Points

This section is used to plot the data from 3-axis force sensors and from the calibration sensors. This is used to manually select the starting and ending points of the lows and highs of the load steps. Run this section if the start and end points for a CSV file are unknown. The plot generated can be used to manually pick these points. Pick a range of ~1000 points. Write these points in the next section.

## Section 4: Start and End Points for All Steps

This section is used to record the start and end points for all steps for all the 12 files. If running the script for a single file, write down these points for the corresponding file index.

## Section 5: Check Start and End Points

This section will plot 9 subplots (for each axis of each sensor) with vertical lines showing the start and end points to check if the selection of points was correct.

## Section 6: Create Diagram

This section creates a figure that shows a top down view of the plate with the force sensors locations, and displays the corresponding x, y and z forces for each sensor, and it calculates the net forces and torques about the center. This is helpful to analyze if the data recorded for calibration was performed properly by looking at the forces and torques and comparing them to what would be expected. For instance, when doing Fx+ calibration, the net Fx should be positive. And forces on other axis should be (ideally) < 5% of Fx.

## Section 7: K-Matrix Creation Section

This section can only be run if all 12 tests have been performed, meaning that there are 12 CSV files, and all 12 start/end points have been found. A 9x6 K matrix will be created at the end. Save this matrix as "K96.mat".

## Section 8: Get Matrix Inverse

This section will reshape the matrix found in the previous section into a 6x6 matrix using the equations for `Vfx_Ks`, `Vfy_Ks`, `Vfz_Ks`, `Vtx_Ks`, `Vty_Ks`, and `Vtz_Ks`. This 6x6 matrix will then be inverted to be used to find the 'calibrated' loads. Save this matrix as "K66inv.mat".

### postprocess.m

This script reads a CSV file containing the 9-column force data obtained from slosh experiments, detrends and filters this data, and calculates the net forces and torques of the system using the K matrix obtained from `main.m`.

1. Set `sr` to the sampling rate (in Hz).
2. Set the filter parameters: Cutoff frequency (Hz) using the `cutoff_f` variable, and the order of the filter using the `filt_order` parameter.
3. Set `csv_file_name` to the path to the CSV file.
4. Set `percentageToPlot` to the desired percentage (of the time axis) to plot. This will plot half of that amount before the middle point, and the other half after. For example, if 10% is selected and there's 100 data points, it will plot from 45-55.
5. Run the **Using K66** or the **Using K96** section (or both) to generate a tiled figure including 6 plots corresponding to Fx, Fy, Fz, Tx, Ty and Tz.

### Using K66

This section uses the 6x6 K matrix to find the net forces and torques of the system. It does so by finding Vfx, Vfy, Vfz, Vtx, Vty, and Vtz using Fx1,Fx2,Fx3,Fy1,Fy2,Fy3,Fz1,Fz2,Fz3. It then uses the 6 voltages and the 6x6 inverse K matrix to find the 3 net forces and torques of the system.

### Using K66

This section uses the 9x6 K matrix to find the net forces and torques of the system. It does so by making use of the " MATLAB operator to solve the system of linear equations including the 6x1 Force and Torque vector, 9x6 K matrix, and 9x1 Forces (voltages) vector using the least squares method.