postprocess.m

This script uses the K96 matrix obtained from `calibration.m` to process the CSV files obtained from LabVIEW during the slosh experiments. It creates plots for the forces and torques of the system, and of the liquid only.

Parameters:

- (Int) **percentageToPlot**: Percentage of the total data to trim the x axis of the plot (0-100).
- (Int) **ssPercentage**: Percentage of what is considered to be the steady state portion of the entire cycles.
- (Boolean) **logParamsBool**: Whether to print to the console the parameters of each file that is being processed (true/false).
- (Boolean) **plotBoolSystem**: Whether to create a window displaying the loads of the system. If looping for several tests, it's recommended to set it to 'false'.
- (Boolean) **plotBoolSlosh**: Whether to create a window displaying the loads caused by the liquid. If looping for several tests, it's recommended to set it to 'false'.
- (Boolean) **plotBoolAcc**: Whether to create a window displaying the plot of the accelerometer. If looping for several tests, it's recommended to set it to 'false'.
- (Boolean) **saveBoolSystem**: Whether save the figure of the loads of the system as a 'jpeg' image to storage. If set to 'true', it saves the images to './Plots/'.
- (Boolean) **saveBoolSlosh**: Whether save the figure of the loads caused by the liquid as a 'jpeg' image to storage. If set to 'true', it saves the images to './Plots/'.
- (Boolean) **saveBoolAcc**: Whether save the plot of the accelerometer as a 'jpeg' image to storage. If set to 'true', it saves the images to './Plots/'.
- (Vector) **loadsToPlotSystem**: Vector including the loads to include in the plot of the system loads (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]).
- (Vector) **loadsToPlotSlosh**: Vector including the loads to include in the plot of the loads caused by the liquid (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]).
- (Int) **sr**: Sampling rate in Hz.
- (Float) **cutoff_f**: Cutoff frequency of the filter in Hz.
- (Int) **filt_order**: Order of the filter.
- (Array) **K96**: 9x6 matrix obtained from `calibration.m` . Change the path to this file if needed.
- (Cell) **fileNamesCell**: Cell containing the data read from "Filenames.xlsx". Change the path to this file if needed.
- (String) **empty_files_path**: Path to where the empty tank CSVs are.

- (Array) **ma**: Moment arms for the three sensors. No need to change unless the force sensors are mounted elsewhere in the system.
- (Array) **cyclesPerFreq**: Array relating the frequency of the test (in Hz) to the number of cycles performed at that frequency.
- (Vector) **filterParams**: Vector including the sampling rate, cutoff frequency, and order of the filter.
- (Vector) **grayTestNumbers**: Vector including the test numbers to not process.

Processing

- Change the for loop range to iterate over the desired range of tests.
- The following will iterate for all tests.
  - The if statement inside the for loop will make the script ignore the CSVs in `grayTestNumbers`.
  - The script uses `readtable` to import force data from the CSVs.
  - This force data is then passed to `getCalibratedLoadsK96` where it's detrended, filtered, and converted to a 6 column array containing Fx,Fy,Fz,Tx,Ty,Tz. This is stored in `ftArray`. This function also stores the time array into `tStamps`.
  - The `createFTplots` function is then used to plot/save this data.
  - Using `getThFreqAccDoubleAmpFill`, the target frequency, acceleration, double amplitude, and fill percentage are stored in `thFreq`, `thAcc`, `thDoubleAmp`, and `thFill`.
  - The CSV file name for the empty tank log is obtained using `empty_files_path`, `thFreq`, and `thAcc`. It is stored in `empty_file`.
  - The location of the peaks of `ftArray` are obtained using the `findStartEndCycles` function. This provides information of when the cycles begin and when they end.
  - Using the location of the beginning and the end of the cycles, the location of the beginning and end of the 'steady state' cycles is obtained by making use of `ssPercentage`. These indexes are saved in `ssStart` and `ssEnd`, and the steady state portion of the data is saved in `ssLoads`.
  - The duration of a single cycle (in number of data points) is obtained using the sample rate and the target frequency. This value is stored in `singleCycleDuration`.
  - Using `getSimpleEmptyTankCycle`, a single cycle of the empty tank data is stored in `singleEmptyCycle`.
  - This data is then used to create a repeated array of greater length than `ssLoads`. This is then modified to match the length of `ssLoads`, and to also match the peaks. This is stored in `reshapedData`.
  - The liquid forces are obtained by subtracting `reshapedData` from `ssLoads`.
  - The `createFTplots` function is then used to plot/save this data.
  - The `processAcceleration` function is finally used to read, detrend, and filter the acceleration to plot it against time. The function stores the detrended and filtered acceleration to the `filtered_acc` variable.