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. The following directories need to contain the following files:

- **'Slosh Data' directory:** CSV files from the experiments
- **'EmptyTankFiles' directory:** CSV files from the empty-tank experiments
- **'Position' directory:** Position txt files.

A file named 'Filenames.xlsx' containing the testnames tabulated against frequency, fill percentage and acceleration needs to exist inside the main directory.

Important:

Some loads from the sensors in the CSV files aren't oriented correctly since the mounting of the three sensors is not the same for the 3 of them. The function `getCalibratedLoadsK96.m` takes care of orienting these loads correctly by multiplying individual columns by -1 (line 16). However, before starting postprocessing, it is important to verify that the orientations are correct. To verify this, plot Fx1, Fx2, and Fx3 (from the CSV file) together and verify that they are oriented correctly. The same applies for Fy and Fz. If they are not oriented correctly, edit line 16 accordingly.

Parameters:

- (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).
- (Int) **sr**: Sampling rate 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.
- (Array) **K66inv**: Inverse 6x6 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 the empty tank CSVs.
- (String) **pos_files_path**: Path to the position txt files.
- (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) **grayTestNumbers**: Vector including the test numbers to not process.
- (Vector) **cutoffFreqs**: Cutoff frequencies (in Hz) for the filter related to the experiment frequencies [0.1, 0.5, 1, 2, 3, 4, 5, 6, 7 , 8, 9, 10] also in Hz.
- (Vector) **filterParams**: Vector including the sampling rate, cutoff frequency, and order of the filter.

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` .
  - Using `getThFreqAccDoubleAmpFill` , the target frequency, acceleration, double amplitude, and fill percentage are stored in `thFreq` , `thAcc` , `thDoubleAmp` , and `thFill` .
  - Based on `thFreq` , the cutoff frequency is assigned based on `cutoffFreqs` .
  - The position and time vector of the linear encoder are store in `posVector` and `tVectorPos` , respectively.
  - `getCalibratedLoadsK96` is used to read the CSV file with the experimental data. Inside this function, this data is detrended, filtered, and converted to a 6 column array (using K96) containing Fx,Fy,Fz,Tx,Ty,Tz. This is stored in `ftArray` . This function also stores the time array into `tStamps` , and the acceleration vector to `accVector` . If the CSV file has no acceleration column, it will set `accVector` to an empty string ("").
  - The CSV file name for the empty tank log is obtained using `empty_files_path` , `thFreq` , and `thAcc` . It is stored in `empty_file` .
  - `getSSFilledTankLoads` uses the 6-column array obtained from `getCalibratedLoadsK96` , trims it according to `ssPercentage` , and stores it in `ssLoads` .
  - Using `getEmptyCyclesAmplitudes` , the amplitudes of the empty-tank datasets are obtained.
  - These amplitudes are then used to create sinousoidal datasets corresponding to the frequency of the experiment using `getEmptyTankLoads.m` .
  - `reshapeEmptyData.m` is used to trim the empty-tank dataset to match `ssLoads` . This data is stored in `reshapedData` .
  - The liquid forces are obtained by subtracting `reshapedData` from `ssLoads` . This is stored in `sloshResults` .
  - Finally, `createReportPlots` creates 14 plots per test and stores them as png images in './Plots/Report/{testName}'. These images are then used to create the report with the `docBuilder.py` Python script.