```
%{
This file takes in the struct containing all the datasets
and performs FFT to find the threshold frequency for the filer we will
 use.
It is only used for that, never again.
Provides good frequency domain plot that visually justifies the choice
 for
the cutoff frequency used in the low pass filter.
%}

% extract time column (x) and one feature (y)
current_dataset = raw_data(3).RA;
x = table2array(current_dataset(1:280,1));

% ---------------------------------------------
% Loop through all of the raw_data
% ---------------------------------------------
% array containing the names of the folders. These names will match
 the
% field names in the struct
folders = ["LGW","RA","RD","SiS","StS"];

Y = 0;
% loop through each of the folders
for ff = 1 : length(folders)
    for kk = 1 : length(raw_data)
%         fprintf("\nkk is %i, ff is %i\n", kk, ff)
        current_dataset = raw_data(kk).(folders{ff});
        % Filter the data
        % loop through the columns in the single dataset
        for ii = 1 : width(current_dataset)
            % obtain the relevant column
            colm = table2array(current_dataset(1:280,ii));
            % ignore timestamp columns
            avg = abs(nanmean(colm));
            % if columns is NOT a timestamp one
            if avg < 1000
                if Y == 0
                    Y = colm;
                else
                    % Y is the total of the readings in all columns
 which will be used
                    % for FFT
                    Y = Y + colm;
                end
            end
        end
    end
end


% ---------------------------------------------
```

```
% Loop through a single dataset
% ------------------------------------------------
%{
Y = 0;
% loop through the columns in the single dataset
for ii = 1 : width(current_dataset)
    % obtain the relevant column
    colm = table2array(current_dataset(:,ii));
    % ignore timestamp columns
    avg = abs(nanmean(colm));
    % if columns is NOT a timestamp one
    if avg < 1000
        % Y is the total of the readings in all columns which will be
 used
        % for FFT
        Y = Y + colm;
    end
end
%}

% ------------------------------------------------

% plot a single reading
% plot(x, y, 'r')

% plot the fft
nfft = length(Y);           % length of time domain signal
nfft2 = 2^nextpow2(nfft);   % length of signal in power of 2
ff = fft(Y, nfft2);
fff = ff(1:nfft2/2);
plot(abs(fff))
```

Published with MATLAB® R2020a