
```

%{
This function imports all the datasets and deletes all timestamp
columns
apart from the very first one in each dataset and any columns with
values of 0.

Returns
- `rawData`      -> a 1x12 struct with 5 fields ("LGW","RA","RD","SiS"
and "StS")
%}

function [rawData] = importDatasets()
    % initialisations
    previous_is_timestamp = false;
    columnsToDelete = ("yes");

    % array containing the names of the folders. These names will
match the
% field names in the struct
folders = ["LGW","RA","RD","SiS","StS"];
% loop through each of the folders
for ff = 1 : length(folders)
    % specify the folder we are interested in during this loop
    myFolder = folders(ff);
    % Get a list of all .dat files in the folder
    filePattern = fullfile(myFolder, '*.dat');
    theFiles = dir(filePattern);
    % We now want to loop through all the files in the current
folder of
% interest
    for kk = 1 : length(theFiles)
        baseFileName = theFiles(kk).name;
        fullFileName = fullfile(myFolder, baseFileName);
        fprintf(1, 'Now reading %s\n', fullFileName);
        % read the .dat file into a table variable
        dataset = readtable(fullFileName, "ReadVariableNames",true);
        % read the .dat file also into a cell variable to allow
retrieval of
        % column details such as data type and units (seconds,
milliseconds, etc)
        rawDataset = readcell(fullFileName);

        % -----
        % loop through each column in the dataset to remove those
with synchronisation time vectors
        % -----
        for col = 1 : width(dataset)
            thisColumn = dataset(:, col); % Extract this one column
into its own variable.
            avg = abs(nanmean(thisColumn{: ,end}));
            % should delete columns with an average this high as it
means

```

```

        % values in them are a timestamp. Except for a single
column for
        % each IMU which we keep
        if (col>1) && ((avg > 1000) || avg == 0)
            columnsToDelete(end+1) =
dataset.Properties.VariableNames{col};
        end
    end
    % Before moving on to the next .dat file, we want to delete
    % all the columns in 'columnsToDelete' from this file
    if length(columnsToDelete) > 1
        % delete the "yes" element from the array
        columnsToDelete(1) = [];
        % delete the relevant columns from the dataset
        dataset = removevars(dataset,columnsToDelete);
    end
    % restart the columnsToDelete list
    columnsToDelete = ("yes");
    % -----
    % append the contents of the .dat file to the struct under
the
        % relevant field name and number
        % -----
        rawData(kk).(folders{ff}) = dataset;
    end
end
end

```

Published with MATLAB® R2020a