

Data Analysis Tool

TOOL OVERVIEW

Data Logging Format

Data is logged in a text file in the following format:

```
#Profile = ./Profiles/GAUIquadPID.profile
#UAV Constants      P      I      D      Trim
#Pitch              -1980      0      -409      0
#Roll                0      0      0      0
#Yaw                 0      0      0      0
#Altitude            0      0      0      0
#Longitude           0      0      0      0
#Latitude            0      0      0      0
#Orientation:       OptiTrack-VRPN
#Position:          OptiTrack-VRPN
#Direction:         OptiTrack-VRPN
#Communication:     6-Channel Trainer
#Control:           None
```

Experiment configuration

Data headers

```
#Time  Lat Lon Alt Roll Pitch Yaw PIDpitchIn PIDpitchOut setThrottle matrixInPitch Motor_1 Motor_3
0      +0.1306 +0.3377 +1.4709 +0.28 +3.48 -0.03 0.00 2606.05 0.00 434.34 0 1300
7      +0.1306 +0.3377 +1.4709 +0.28 +3.48 -0.03 0.00 2606.05 0.00 434.34 0 1300
11     +0.1303 +0.3380 +1.4709 +0.30 +3.53 +0.00 0.00 155.69 0.00 25.95 0 325
```

Data

Components

DataAnalysis	← script
params	← structure
expData	← structure
parse_log()	← function
plot_data()	← function
plot_separate()	← function
plot_multi()	← function
plot_sub()	← function

How to Use the Tool

1. Open up the `DataAnalysis` script
2. Set the options for the analysis
3. Run the `DataAnalysis` script
4. If needed, use the plotting functions for extra plots

IMPORTANT: The data header of time values in the log file must be named "Time"

DataAnalysis Script

This is the top level script that user will first interface with

Layout

- Set analysis configuration options
- Parse logged data
- Plot data, if required

```
%% DO NOT MODIFY %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fpath = '/local/ucart/Documents/MicroCART_17-18/groundStation/logs/';
if(isempty(fname))
    [fname, fpath] = uigetfile('.txt','Select log file', fpath);
end

% storing file options in the structure
params.file.name = fname;           % file name only
params.file.path = fpath;           % file path only
params.file.pathName = [fpath fname]; % file path with file name

% storing plotting parameters in the structure
params.plotting.plot = plot;
params.plotting.separatePlot = separatePlot;
params.plotting.multiPlot = multiPlot;
params.plotting.subPlot = subPlot;
params.plotting.clearFigs = clearFigs;

% storing plotting data in the structure
params.plotting.separateData = separateData;
params.plotting.multiData = multiData;
params.plotting.subData = subData;

% storing colors for plotting in the struture
params.plotting.color = color;
params.plotting.marker = marker;
params.plotting.style = style;
params.plotting.backgnd = backgnd;

% saving params into a file
%save params params;

% DO NOT MODIFY - END
```

Analysis Configuration Options

`params`

All the analysis configuration options set in the `DataAnalysis` script are stored in the `params` structure

Parameters overview

- Log file name
- Plotting
 - Quantities to plot
 - Type of plotting
 - Separate plots for multiple quantities
 - Single plot with super-imposed quantities
 - Sub-plots of multiple quantities
- Units for common quantities

Analysis Configuration Options

`params.file.name`

- Name of the file

`params.file.path`

- Path of the file with a trailing file name

If the user knows the file name and the log file is in the current directory

- the user can set the file name in the `fname` variable

else

- the user can leave the value of `fname` blank
- an explorer window will pop up to locate the file to be parsed

Analysis Configuration Options

`params.time.units`

- Unit of time to be displayed while plotting

`params.time.scaler`

- Value by which the time vector in the log file needs to be multiplied by to obtain the time in the unit mentioned in `params.time.units`

`params.angle.units`

- Unit in which the angles have been recorded

Analysis Configuration Options

`params.plotting.plot`

- Switch to generate plots

`params.plotting.separate`

- Switch to generate individual plots

`params.plotting.separateData`

- Cell array of headers to be plotted individually. If empty, all headers are plotted on separate plots.

`params.plotting.multi`

- Switch to generate a plot with multiple data in the single plot

`params.plotting.multiData`

- Cell array of headers to be plotted together on the single plot. If empty, the `params.plotting.separateData` vector is used.

`params.plotting.subPlots`

- Switch to generate 2x1 subplots

`params.plotting.subPlotsData`

- Cell array of headers be plotted in the sub plots. If empty, the `params.plotting.multiData` vector is used.

parse_log() Function

```
function [loggedData] = parse_log(filename, params)
```

filename

- Either the name of the log file OR the path to the log file

params

- The analysis configuration options

loggedData

- A structure that contains the data contained in the log file
- This is the `expData` specifically

This function parses the data stored in the log file and returns a structure containing the data

expData Structure

This structure will store all of the logged data in the following format:

expData

- <header-name>
 - name
 - data
 - units

Example:

expData

- Pitch
 - name = "Pitch"
 - data = <array of data logged for Pitch>
 - units = "rad"

To access an attribute of any logged quantity, use the following:

`expData.<header-name>.<attribute>`

`expData.Pitch.data`

plot_data() Function

```
function plot_data(expData, plotParams)
```

expData

- The structure containing the data in the log file

plotParams

- The plotting options that were set in the DataAnalysis script
- This is `params.plotting` specifically

This function plots the data according to the parameters mentioned in `plotParams`.

This function will most likely be used in the `DataAnalysis` script only.

plot_separate() Function

```
function plot_separate(expData, varargin)
```

expData

- The structure containing the data in the log file

varargin

- The name of headers to be plotted on the same plot with optional **plot formatting** options

This function plots separate plots for the data headers mentioned.

Example: `plot_multi(expData, 'Pitch', 'go', 'Roll', 'Yaw', 'rs')`

plot_multi() Function

```
function plot_multi(expData, numOfHeaders, varargin)
```

expData

- The structure containing the data in the log file

numOfHeaders

- number of data sets to be plotted on the same plot

varargin

- The name of headers to be plotted on the same plot with optional **plot formatting** options

This function is used to plot multiple headers on the same plot.

Example: `plot_multi(expData, 3, 'Pitch', 'go', 'Roll', 'Yaw')`

plot_sub() Function

```
function plot_sub(expData, varargin)
```

expData

- The structure containing the data in the log file

varargin

- The data headers to be plotted in the sub plots along with optional **plot formatting** options

This function generates 2x1 sub plots of data headers specified

Example: `plot_sub(expData, 'Pitch', 'go', 'Roll')`

Plot Formatting Options

These are character strings like those that are used with MATLAB's plot function.

The character string made from one element from any or all the following 3 columns:

b	blue	.	point	-	solid
g	green	o	circle	:	dotted
r	red	x	x-mark	-.	dashdot
c	cyan	+	plus	--	dashed
m	magenta	*	star	(none)	no line
y	yellow	s	square		
k	black	d	diamond		
w	white	v	triangle (down)		
		^	triangle (up)		
		<	triangle (left)		
		>	triangle (right)		
		p	pentagram		
		h	hexagram		

Examples: `'go'`, `'rs'`, `'--rs'`

MAKE SURE YOU FORMAT DATA HEADERS CORRECTLY

```
% SEPARATEDATA  
% write names of the data headers that you want to plot using  
% seperatePlots. If this is empty but "plot" switch is 1, all the columns  
% will be plotted using seperatePlots.  
separateData = {'Pitch', 'Roll', 'Yaw'};
```

Plotting will not work if format is wrong.

Examples

Log File Used

Using sampleLogFile.txt

The log file contains the following headers:

```
#Time   Lat Lon Alt Roll   Pitch   Yaw PIDpitchIn  PIDpitchOut setThrottle matrixInPitch  Motor_1 Motor_3
```

EXAMPLE 1: Separate plots only

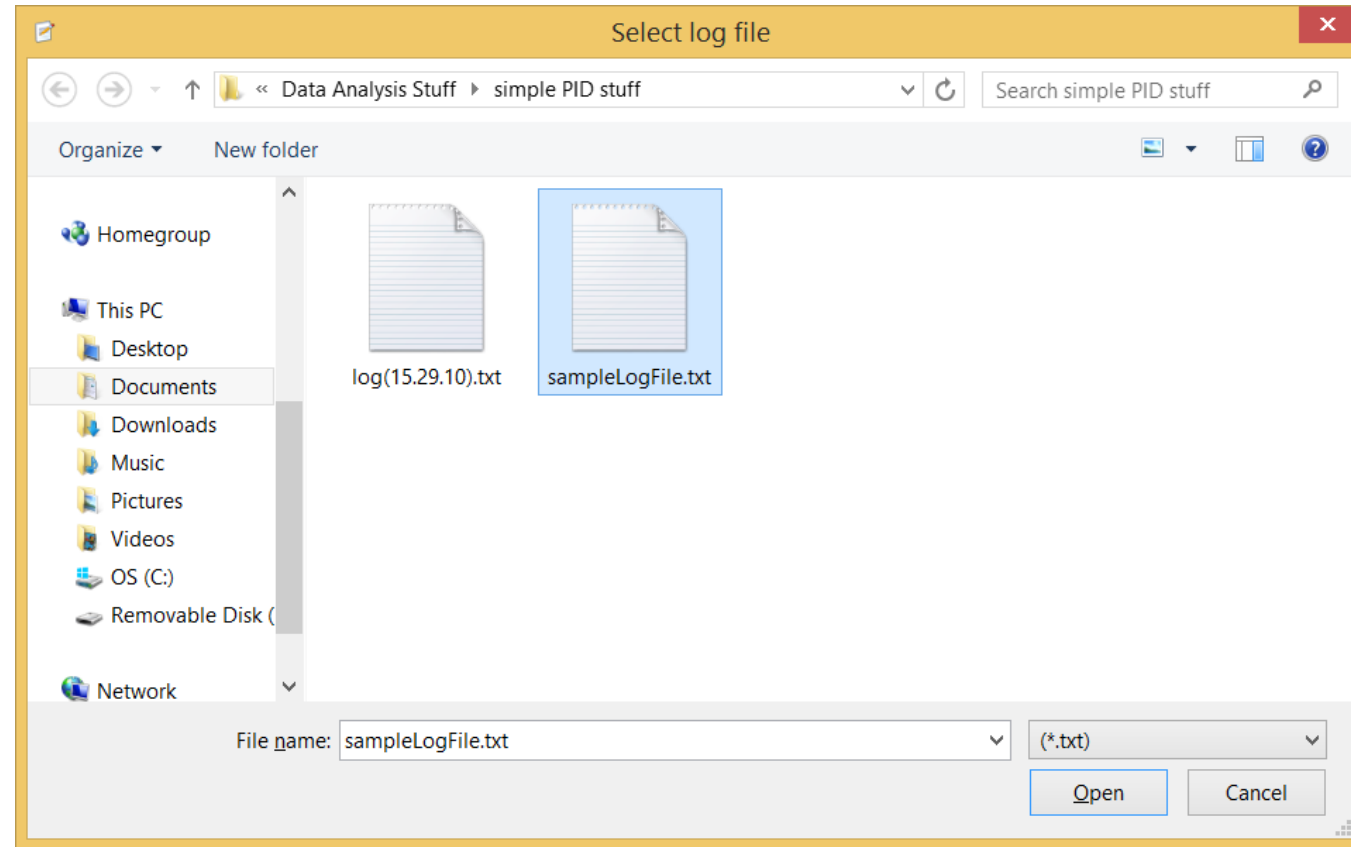
Analysis Configuration Options

```
% PLOTTING SWITCHES - set them to 0 or 1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
plot          = 1;    % to choose plotting
separatePlot  = 1;    % to generate separatePlots
multiPlot     = 0;    % to generate multiPlot
subPlot       = 0;    % to generate subPlots
clearFigs     = 0;    % to close all the plots (needed only by the GUI)

% DATA TO PLOT %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% MAKE SURE EACH DATA SECTION FOLLOWS A FORMAT LIKE THIS OR PLOTTING WILL
```

After running this script →

Select File



Generated params Structure

```
params  
file  
time  
angle  
plotting
```

```
params.file  
name: 'simpleLogFile.txt'  
path: 'C:\Users\Rohit\Documents\...'
```

```
params.time  
units: 's'  
scaler: 1e-03
```

```
params.angle  
units: 'deg'
```

```
params.plotting  
plot: 1  
separate: 1  
multi: 0  
subPlots: 0  
separateData: {'Pitch' 'Roll' 'Yaw'}  
multiData: {}  
subPlotsData: {}
```

Generated expData Structure

expData

Time

Lat

Lon

Alt

Roll

Pitch

Yaw

PIDpitchIn

PIDpitchOut

setThrottle

matrixInPitch

Motor_1

Motor_3

expData.Time

data: [733x1 double]

units: 's'

type: 'Time'

expData.Pitch

data: [733x1 double]

units: 'deg'

type: 'Angle'

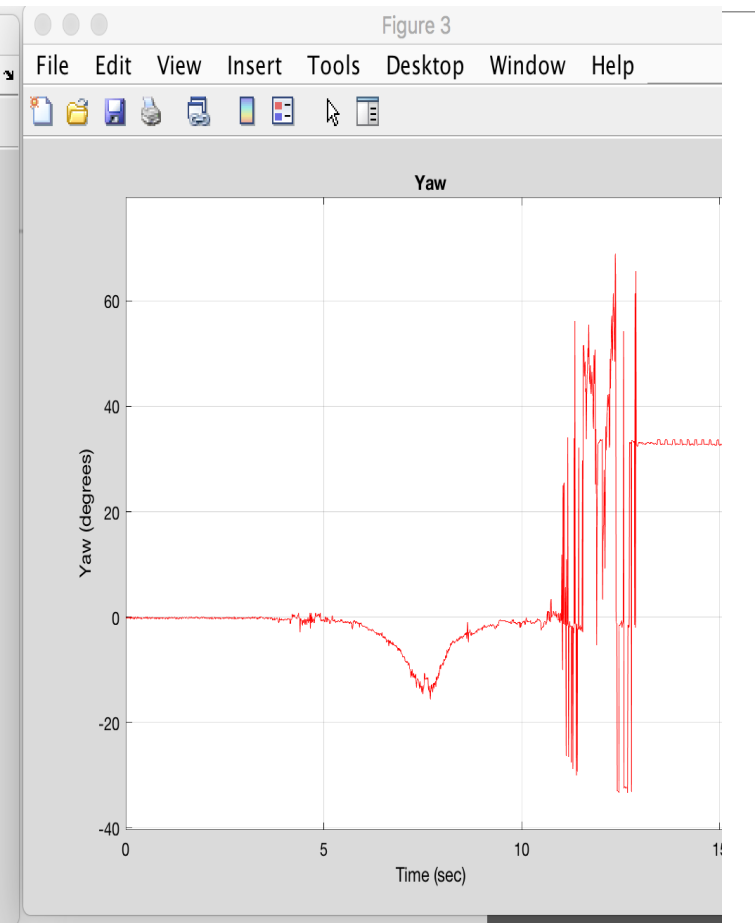
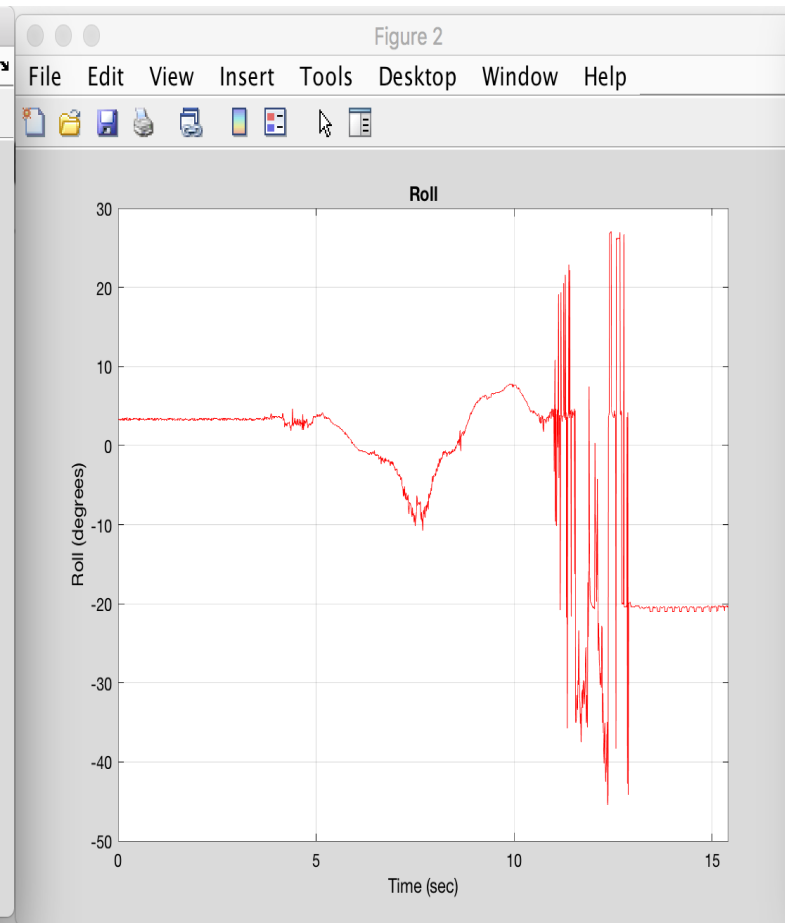
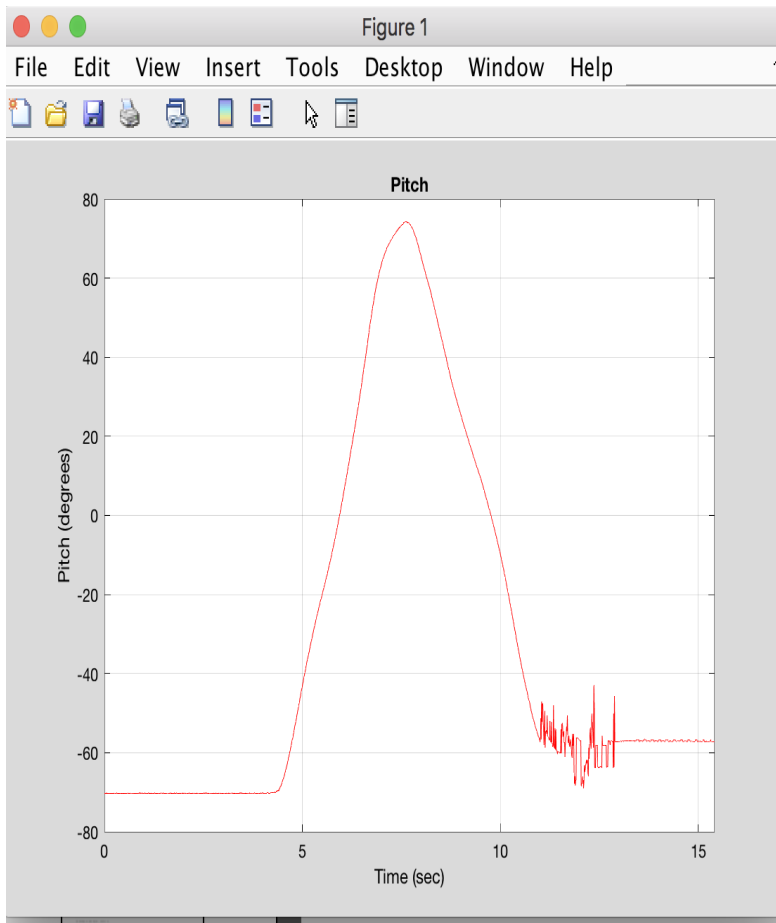
expData.setThrottle

data: [733x1 double]

units: ''

type: 'Other'

Generated Plots

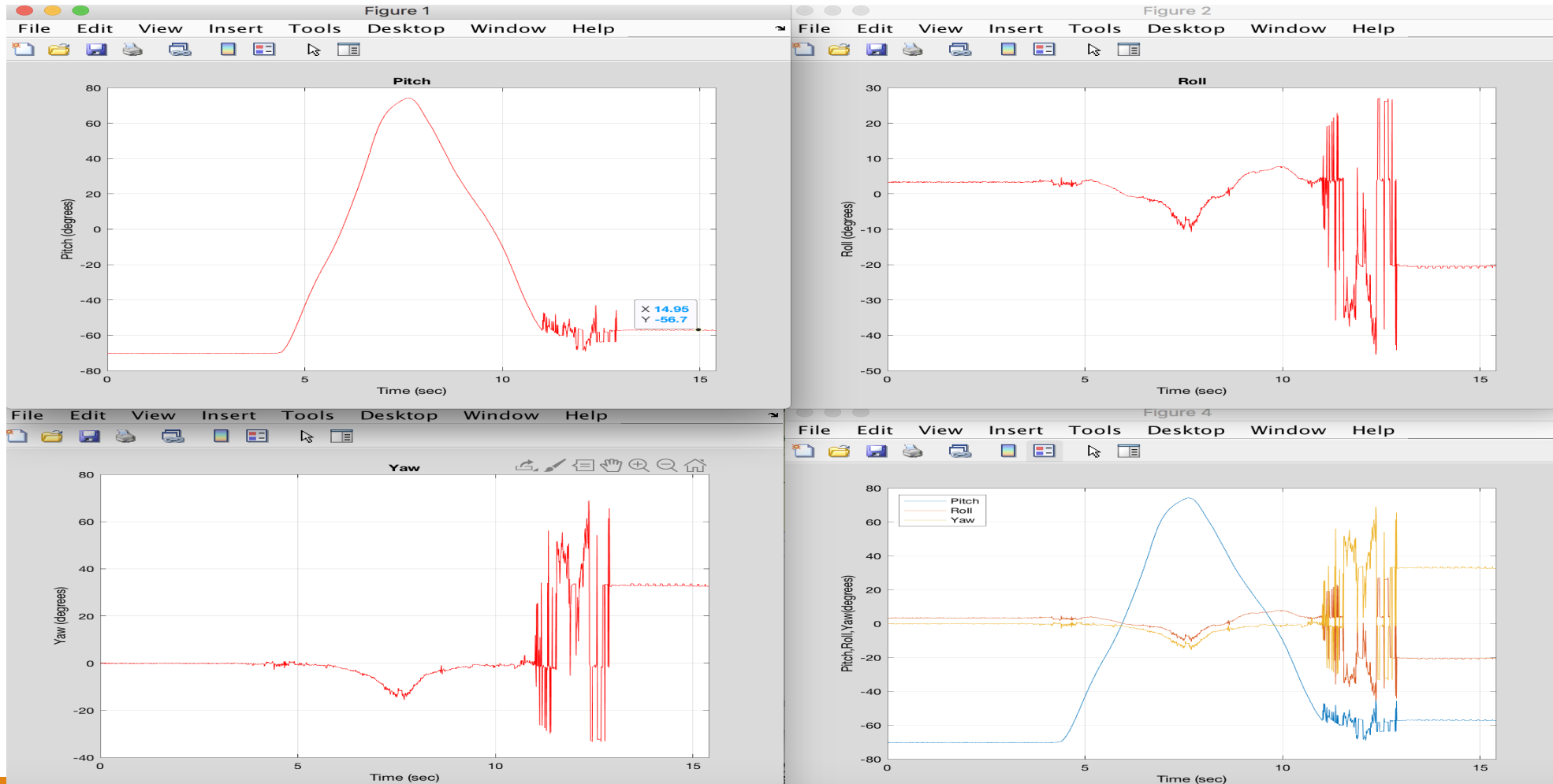


EXAMPLE 2: Separate and Multi Together

Analysis Configuration Options

```
% PLOTTING SWITCHES - set them to 0 or 1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
plot          = 1;    % to choose plotting
separatePlot  = 1;    % to generate separatePlots
multiPlot     = 1;    % to generate multiPlot
subPlot       = 0;    % to generate subPlots
clearFigs     = 0;    % to close all the plots (needed only by the GUI)
```

Plots Generated

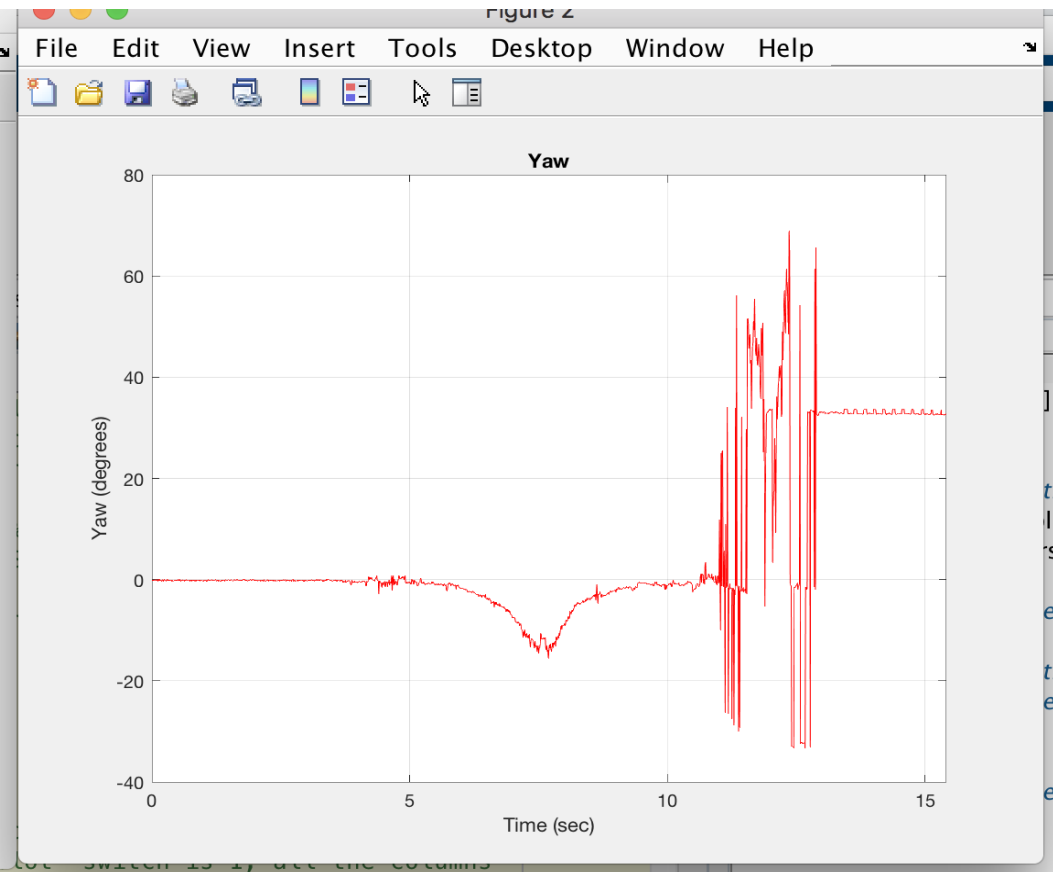
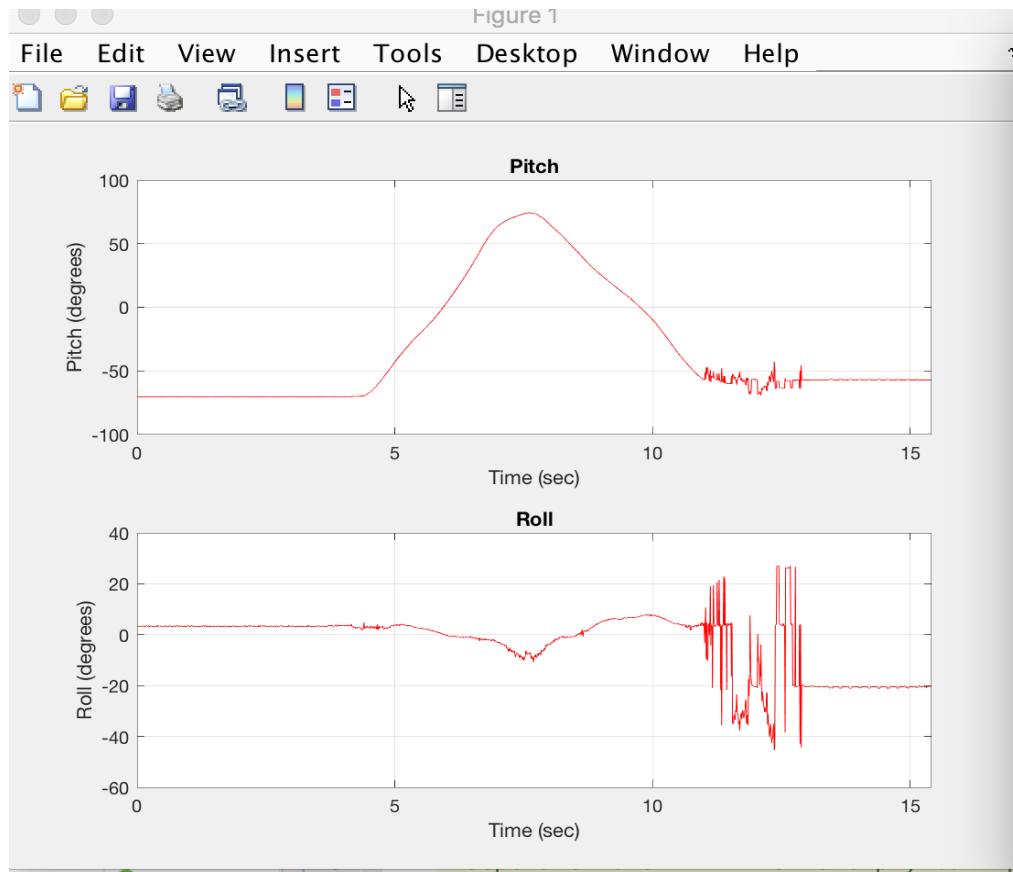


EXAMPLE 3: Subplots

Independently Using Plot Functions

```
plot_sub(expData, 'Pitch', 'go', 'Roll');
```

Plot Generated



Potential New Additions

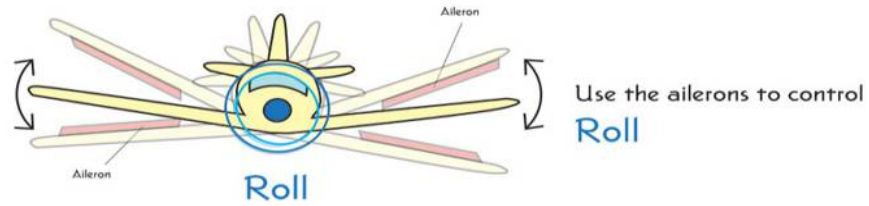
Add the following functions to the tool

- `plot_separate_vectors()`
- `plot_multi_vectors()`
- `plot_sub_vectors()`

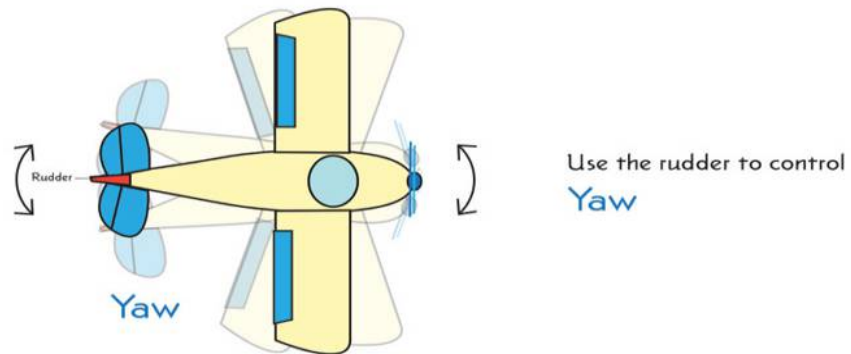
The current plotting functions allow only data from the data structure, `expData`, to be used. If any data needs to be modified (scaled, etc.), the original data in the data structure needs to be modified the plotting functions are to be used.

These functions will take in variable names instead of header names to allow for plotting of data that is not in the `expData` structure.

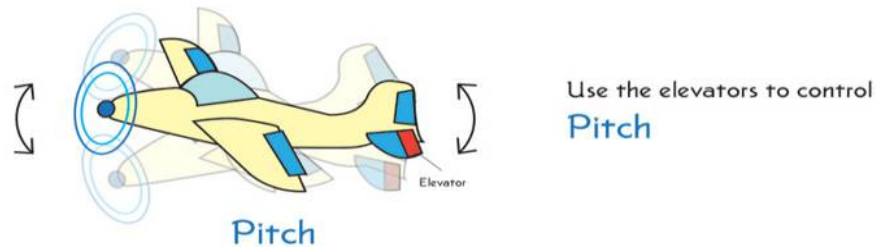
Quick Guide to Interpreting Graphs Results



Roll-Rotation around front to back axis



Yaw-Rotation around vertical axis

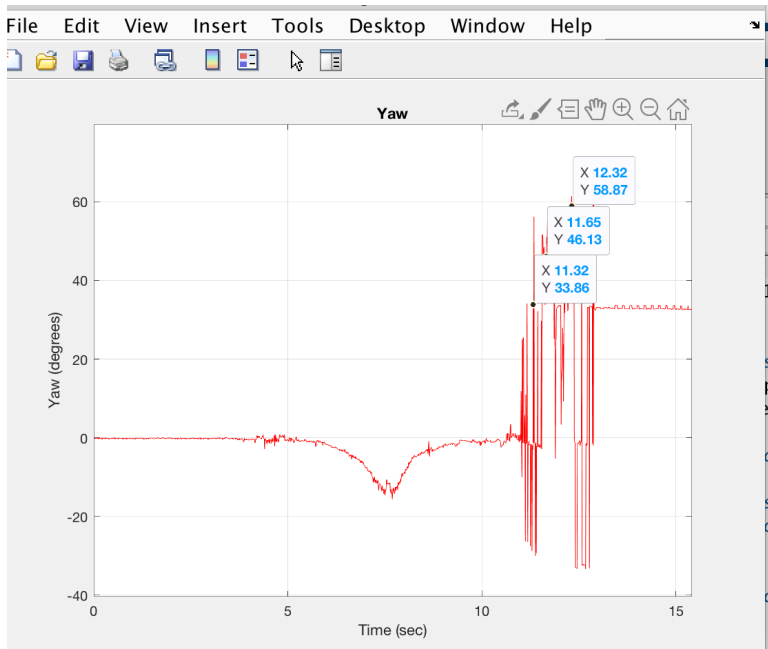
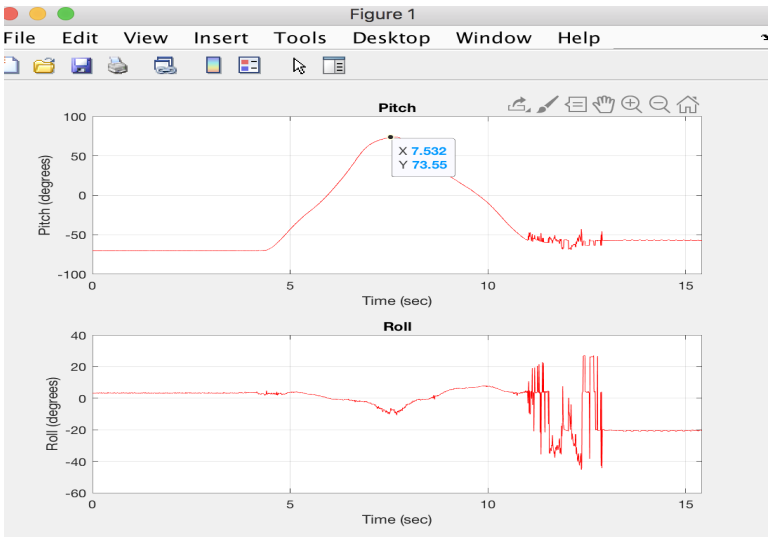


Pitch-Rotation around side to side axis

Image source: <https://howthingsfly.si.edu/flight-dynamics/roll-pitch-and-yaw>

Getting Precise Values

As simple as clicking any point on the line



Can even have multiple points selected