
px4tools Documentation

Release 0.4.9

James Goppert

May 18, 2018

Contents

1	Module: logsysid	3
2	Module: analysis	5
3	Module: mapping	7
4	Indices and tables	9
	Python Module Index	11

Contents:

CHAPTER 1

Module: logsysid

Analyze a PX4 log to perform sysid and control design.

```
px4tools.logsysid.attitude_control_design(name, y, u, rolling_mean_window=100,  
                                           do_plot=False, verbose=False, d_tc=0.008)
```

Do sysid and control design for roll/pitch rate loops. :param name: :param y: output :param u: input :param rolling_mean_window: number of samples in rolling mean :param do_plot: controls plotting :param verbose: show debug output :param d_tc: derivative time constant :return: (gain, closed loop transfer function)

```
px4tools.logsysid.attitude_sysid(y_acc, u_mix, verbose=False)
```

roll/pitch system id assuming delay/gain model :param y_acc: (roll or pitch acceleration) :param u_mix: (roll or pitch acceleration command) :param verbose: show debug output :return: (G_ol, delay, k)

G_ol: open loop plant delay: time delay (sec) k: gain

```
px4tools.logsysid.calculate_fitness(k, delay, y, u, dt)
```

Find how well the function fits the data

```
px4tools.logsysid.control_design(raw_data, do_plot=False, rolling_mean_window=100, ver-  
                                bose=False)
```

Design a PID controller from log file. :param raw_data: data frame :param do_plot: controls plotting :param rolling_mean_window: number of rolling mean samples :param verbose: show debug output :return: (gain ordered dict, local variables)

```
px4tools.logsysid.control_design_uhog(raw_data, do_plot=False,  
                                       rolling_mean_window=100, verbose=False)
```

Design a PID controller from log file. TODO, debug this :param raw_data: data frame :param do_plot: controls plotting :param rolling_mean_window: number of rolling mean samples :param verbose: show debug output :return: (gain ordered dict, local variables)

```
px4tools.logsysid.delay_and_gain_fit_fun(x, y, u, dt)
```

Fitness function for delay_and_gain_sysid :param x: state (k, delay) :param y: output :param u: input :param dt: period (sec) :return: the fitness cost

```
px4tools.logsysid.delay_and_gain_sysid(y, u, verbose=False)
```

Finds gain and time delay to best fit output y to input u :param y: output :param u: input :param verbose: show debug output :return: (k, delay)

`px4tools.logsysid.lqr_ofb_con(K, R, Q, X, ss_o)`

Constraint for LQR output feedback optimization. This asserts that all eigenvalues are negative so that the system is stable. @K gain matrix @Q process noise covariance matrix @X initial state covariance matrix @ss_o open loop state space system @return constraint

`px4tools.logsysid.lqr_ofb_cost(K, R, Q, X, ss_o)`

Cost for LQR output feedback optimization. @K gain matrix @Q process noise covariance matrix @X initial state covariance matrix @ss_o open loop state space system @return cost

`px4tools.logsysid.lqr_ofb_design(K_guess, ss_o, verbose=False)`

LQR output feedback controller design. @K_guess initial stabilizing gains @ss_o open loop state space system @return gain matrix

`px4tools.logsysid.lqr_ofb_jac(K, R, Q, X, ss_o)`

Jacobian for LQR Output feedback optimization. TODO: might be an error here, doesn't not help optim

`px4tools.logsysid.pid_design(G, K_guess, d_tc, verbose=False, use_P=True, use_I=True, use_D=True)`

Parameters

- **G** – transfer function
- **K_guess** – gain matrix guess
- **d_tc** – time constant for derivative
- **verbose** – show debug output
- **use_P** – use p gain in design
- **use_I** – use i gain in design
- **use_D** – use d gain in design

Returns (K, G_comp, Gc_comp) K: gain matrix G_comp: open loop compensated plant Gc_comp: closed loop compensated plant

`px4tools.logsysid.plot_delay_and_gain_fit(k, delay, y, u, dt=0.001)`

Plot the delay and gain fit vs the actual output.

`px4tools.logsysid.plot_loops(name, G_ol, G_cl)`

Plot loops :param name: Name of axis :param G_ol: open loop transfer function :param G_cl: closed loop transfer function

`px4tools.logsysid.setup_data(df)`

Resample a dataframe at 1 ms to prep for sysid. :param df: pandas DataFrame of px4log :return: (df_rs, dt) resample dataframe and period (1 ms)

CHAPTER 2

Module: analysis

Analysis of px4 logs

`px4tools.analysis.all_new_sample(df)`

`px4tools.analysis.alt_analysis(data, min_alt=None, max_alt=None)`
Altitude analysis.

`px4tools.analysis.background_flight_modes(data)`
Overlays a background color for each flight mode. Can be called to style a graph.

`px4tools.analysis.filter_finite(data)`

`px4tools.analysis.find_lpe_gains(df, printing=False)`

`px4tools.analysis.find_meas_period(series)`

`px4tools.analysis.get_auto_data(data)`
Extract auto data.

`px4tools.analysis.get_float_data(dataframe)`
Get float data out of dataframe.

`px4tools.analysis.isfloatarray(cell)`
Convert cell to float if possible.

`px4tools.analysis.new_sample(series)`

`px4tools.analysis.octa_cox_data_to_ss(data)`
Extracts state space model data from octa cox log.

`px4tools.analysis.plot_attitude_loops(data)`
Plot attitude loops.

`px4tools.analysis.plot_attitude_rate_loops(data)`
Plot attitude rate control loops.

`px4tools.analysis.plot_control_loops(data)`
Plot all control loops.

`px4tools.analysis.plot_faults(data)`

`px4tools.analysis.plot_modes(data)`

`px4tools.analysis.plot_position_loops(data)`
Plot position loops.

`px4tools.analysis.plot_timeouts(data)`

`px4tools.analysis.plot_velocity_loops(data)`
Plot velocity loops.

`px4tools.analysis.pos_analysis(data)`
Analyze position.

`px4tools.analysis.process_all(data_frame, project_lat_lon=True, lpe_health=True)`

`px4tools.analysis.process_data(data)`

`px4tools.analysis.process_lpe_health(data)`

`px4tools.analysis.set_time_series(data)`
Set data to use time series

`px4tools.analysis.statistics(df, keys=None, plot=False)`

CHAPTER 3

Module: mapping

CHAPTER 4

Indices and tables

- `genindex`
- `modindex`
- `search`

p

`px4tools.analysis`, 5
`px4tools.logsysid`, 3

A

all_new_sample() (in module px4tools.analysis), 5
alt_analysis() (in module px4tools.analysis), 5
attitude_control_design() (in module px4tools.logsysid), 3
attitude_sysid() (in module px4tools.logsysid), 3

B

background_flight_modes() (in module px4tools.analysis), 5

C

calculate_fitness() (in module px4tools.logsysid), 3
control_design() (in module px4tools.logsysid), 3
control_design_olog() (in module px4tools.logsysid), 3

D

delay_and_gain_fit_fun() (in module px4tools.logsysid), 3
delay_and_gain_sysid() (in module px4tools.logsysid), 3

F

filter_finite() (in module px4tools.analysis), 5
find_lpe_gains() (in module px4tools.analysis), 5
find_meas_period() (in module px4tools.analysis), 5

G

get_auto_data() (in module px4tools.analysis), 5
get_float_data() (in module px4tools.analysis), 5

I

isfloatarray() (in module px4tools.analysis), 5

L

lqr_ofb_con() (in module px4tools.logsysid), 3
lqr_ofb_cost() (in module px4tools.logsysid), 4
lqr_ofb_design() (in module px4tools.logsysid), 4
lqr_ofb_jac() (in module px4tools.logsysid), 4

N

new_sample() (in module px4tools.analysis), 5

O

octa_cox_data_to_ss() (in module px4tools.analysis), 5

P

pid_design() (in module px4tools.logsysid), 4
plot_attitude_loops() (in module px4tools.analysis), 5
plot_attitude_rate_loops() (in module px4tools.analysis), 5
plot_control_loops() (in module px4tools.analysis), 5
plot_delay_and_gain_fit() (in module px4tools.logsysid), 4
plot_faults() (in module px4tools.analysis), 5
plot_loops() (in module px4tools.logsysid), 4
plot_modes() (in module px4tools.analysis), 6
plot_position_loops() (in module px4tools.analysis), 6
plot_timeouts() (in module px4tools.analysis), 6
plot_velocity_loops() (in module px4tools.analysis), 6
pos_analysis() (in module px4tools.analysis), 6
process_all() (in module px4tools.analysis), 6
process_data() (in module px4tools.analysis), 6
process_lpe_health() (in module px4tools.analysis), 6
px4tools.analysis (module), 5
px4tools.logsysid (module), 3

S

set_time_series() (in module px4tools.analysis), 6
setup_data() (in module px4tools.logsysid), 4
statistics() (in module px4tools.analysis), 6