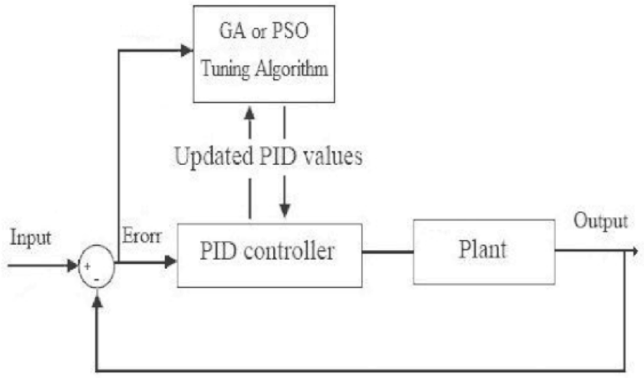
Technical Documentation

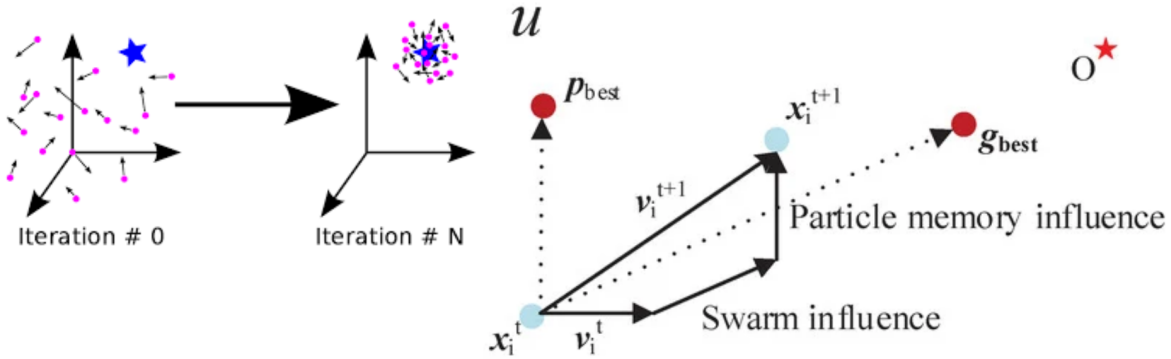
Irene Grace Karot Polson
Guide: Rohit Gupta

Particle swarm and Genetic algorithm are optimisation techniques that could be used for PID Tuning.



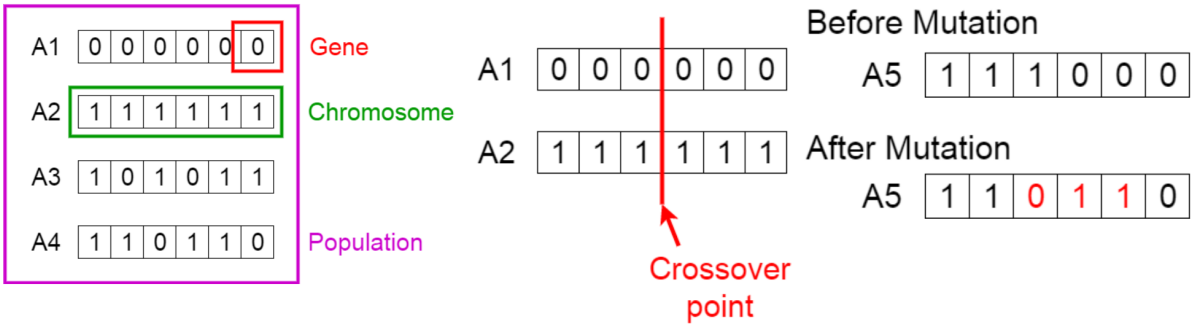
Particle Swarm Optimization

- These particles are moved around in the search-space in a desired manner.
- The movements of the particles are guided by their own best known position in the search-space as well as the entire swarm's best known position.



Genetic Algorithm

- Based on Darwin's evolution theory.
- Natural selection ---> Fittest individuals ---->Crossover---> Mutation



PSA v/s GA

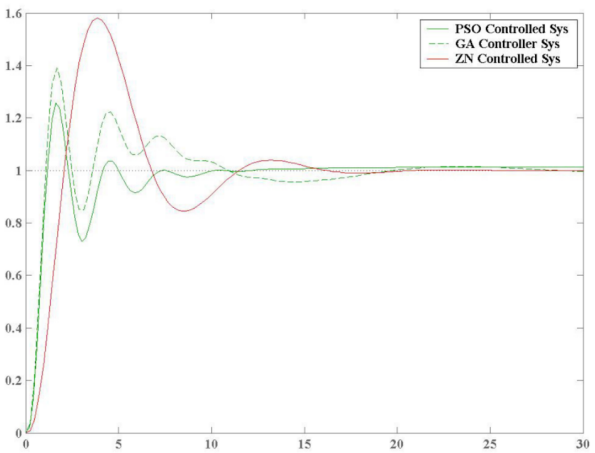
Particle Swarm	Genetic Algorithm
• Small number of parameters(3) and correspondingly lower number of iterations. • The convergence to the global optimum is slow.	• 6 parameters, more adjustments to find the better optimizer. • Works good for noisy environments. • Genetic algorithms do not scale well with complexity. • Could converge to a local minimum.

PSA v/s GA

$$G(s)=\frac{1}{s^4+6\cdot s^3+11\cdot s^2+6\cdot s}$$

The system is stable.

	PSO	GA	ZN
Rise Time	1.2000	1.2000	2.1000
% Overshoot	24.345	28.5908	58.1561
Settling Time	10.0	20.4000	19.8000



Performance criteria were improved by 35% for the GA and 52% for the PSO from ZN.

PSA v/s GA - Optimal Bus Timetabling

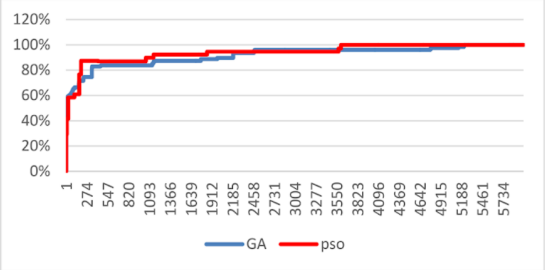


Figure 5. Comparison of GA and PSO Accuracy Charts

Table 1. Comparison of GA and PSO Performance

	Small	Medium	Big
Variable	2	18	100
Constrain	6	38	202
Data	Generate	Real	Generate
Iteration	600	6000	60000

	Small		Medium		Big	
Result	Average Accuracy	Best Iteration	Average Accuracy	Best Iteration	Average Accuracy	Best Iteration
GA	100%	49	99%	5221	93%	56743
PSO	100%	52	100%	3654	100%	44711

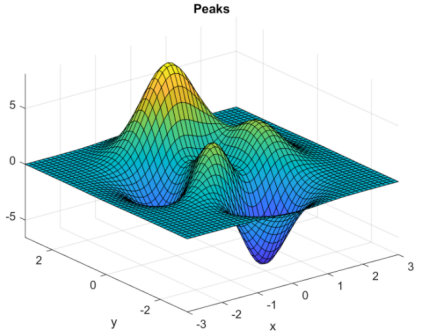
Table 2. Summary of Comparative Results

Indicator	GA	PSO
Complexity	$O(n^2)$	$O(n)$
Accuracy	Solutions in large variables and constraints result in feasible solutions that are near optimal	Resolution produces the optimal solution
Iterate	The more variables and constraints, the more iterations required. In general, it takes much more than the PSO	The more variables and constraints, the more iterations required. Generally, it takes fewer iterations of the GA
Additional techniques used	It takes the addition of chromosome extermination techniques in certain generations in order to approach the optimum solution	No additional techniques

Genetic Algorithm : Peaks

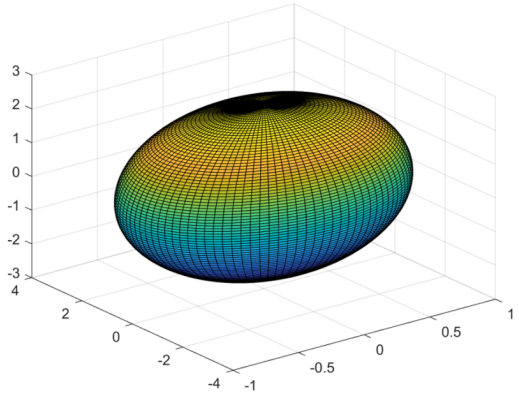
Applied genetic algorithm optimisation on MATLAB's example function PEAKS.

z = 3*(1-x).^2.*exp(-(x.^2) - (y+1).^2) - 10*(x/5 - x.^3 - y.^5).*exp(-x.^2-y.^2) - 1/3*exp(-(x+1).^2 - y.^2)



Generation	f-count	Best f(x)
1	1060	-6.55112
2	2100	-6.55113
3	3140	-6.55113
4	4180	-6.55113
5	5220	-6.55113

Genetic Algorithm : Ellipsoid



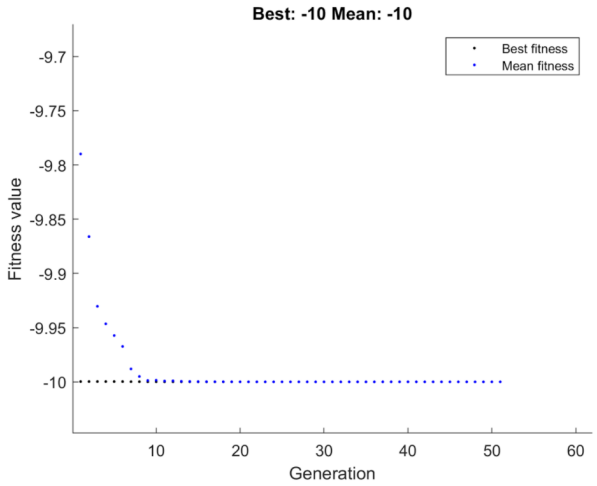
Took the lower half :
z = -c*sqrt(100-(x(1)^2/a)-(x(2)^2/b));

This is constraint as function is only available in a ellipse on XY plane.

nonlcon: @ellipseConstraint , pass this with arguments.

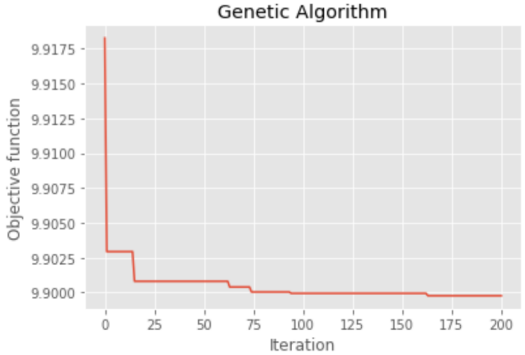
$$\frac{x^2}{6^2} + \frac{y^2}{8^2} + \frac{z^2}{6^2} = 1$$

Genetic Algorithm : Ellipsoid



Generation	Func-count	Best f(x)	Mean f(x)
1	100	-10	-9.79
2	147	-10	-9.866
3	194	-10	-9.93
4	241	-10	-9.947
5	288	-10	-9.957
6	335	-10	-9.967
7	382	-10	-9.988
8	429	-10	-9.995
9	476	-10	-9.999
10	523	-10	-9.998
11	570	-10	-9.999
12	617	-10	-9.999
13	664	-10	-10
14	711	-10	-10
15	758	-10	-10

Genetic Algorithm : Python



'max_num_iteration': None
'population_size': 100
'mutation_probability': 0.1
'elit_ratio': 0.01
'crossover_probability': 0.5
'crossover_type': 'uniform'
'max_iteration_without_improv': None

PID tuning of a Spring-mass damper system with Genetic Algorithm.

We defined the system with mass m = 1kg, damping constant b = 20 N/m/s and Spring constant k = 100N/m.

G(s) = Gp * Gc

= (Kp + Kds + Ki/s) * 1 / (ms^2+bs+k)

= (Kp + Kds + Ki/s) * 1 / (s^2+20s+100)

$Y_{CL} = G_P * G_C * E(s)$

Steps on MATLAB

- Define a GA
- For each particle in each generation,
  - We apply step input
  - Find the response
  - Calculate the objective function
  - Pass it back to GA
- The GA picks the best performing responses for the next generation.

Initially the objective function I had defined was an LQR, this minimises the error as well as input required to achieve that (optimal control).

Objective Function

$$J(K) = (1 - e^{-\beta})(M_p + E_{ss}) + e^{-\beta}(t_s - t_r)$$

$$J = w_1 \times t_u + w_2 \times u(t) + w_3 \times e(t)$$

$$\text{cost} = (1-e^{-\beta})(M_p + e_{ss}) + e^{-\beta} t_s \, .$$

Some J of this sort needs to picked according to our requirement.

Main features to consider : Max Overshoot, Settling time, Rise time, Steady state error.

Deciding on MSE

It is clear to not use IAE due to its poor performance for delays and MSE works best as well so it is safe to choose MSE.

Table 2. Performance indices of tuning methods and GA methods.

Delay	Ziegler-Nichols					Iterative Method					Optimized by MSE	Optimized by IAE	Optimized by ISE	Optimized by ITAE	Optimized by ITSE
	MSE	IAE	ISE	ITAE	ITSE	MSE	IAE	ISE	ITAE	ITSE					
0.01	0.000973	2.992285	1.461115	0.073894	0.020615	0.000927	2.730597	1.391095	0.061045	0.014833	0.00081	1.623776	0.840777	0.011395	0.005388
0.025	0.002626	6.927413	3.942144	0.432876	0.124957	0.002313	6.300772	3.471926	0.391845	0.089213	0.001884	3.701027	2.827366	0.218351	0.038659
0.05	0.004988	12.98842	7.487622	1.580628	0.452425	0.004376	11.6831	6.567769	1.452929	0.318439	0.003708	7.09663	5.57464	0.301924	0.155625
0.075	0.007197	18.53609	10.80309	3.249557	0.929874	0.006334	16.58511	9.507315	2.979748	0.642704	0.005508	10.75044	8.266998	0.754851	0.350289
0.1	0.009284	23.69959	13.93484	5.304512	1.517501	0.008226	20.99624	12.34746	4.801577	1.035597	0.0073	13.86607	10.95693	1.526731	0.622276
0.25	0.020422	48.31448	30.65349	20.82159	6.345217	0.019092	40.69953	28.65714	15.58065	4.507248	0.017983	34.18455	26.99306	7.69427	3.854827
0.5	0.037119	76.26794	55.71523	45.56416	17.73724	0.037603	71.60299	56.44149	37.68745	16.79849	0.035612	67.45696	53.45713	30.00848	15.15463
0.75	0.05416	108.6372	81.29453	91.98153	36.10692	0.057392	123.8553	86.14606	146.8766	42.31687	0.053071	100.0041	79.65463	63.58332	33.54903
1	0.072117	146.0676	108.247	174.114	65.18471	0.077665	176.087	116.5754	298.2008	81.12556	0.070379	134.0671	105.6145	111.3298	58.69364
Av.	0.02321	49.38123	34.83767	38.12475	14.26883	0.02377	52.28229	35.6784	56.44808	16.31655	0.021806	41.41674	32.68734	23.93657	12.4916

GA based PID tuning with MSE

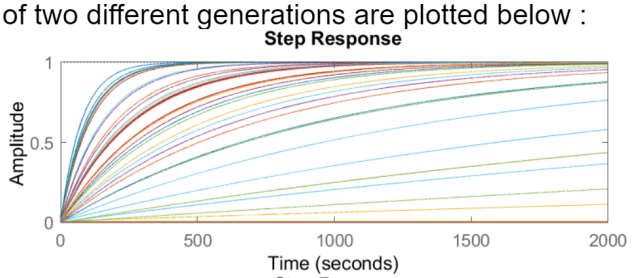
Genetic algorithm parameters :

Initial Population Size : 50
Max Generations : 100
Crossover Probability : 0.6
Mutation Probability : 0.02
Elite ratio : 0.1*PopSize
Objective function : MSE over 2000 time steps.

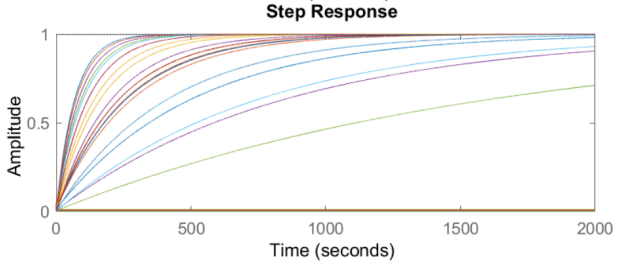
GA Plots

All the responses of two different generations are plotted below :

Generation 1 :

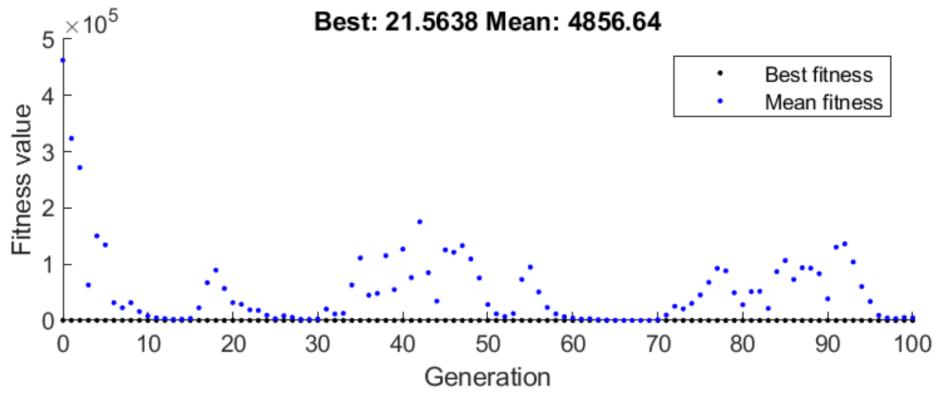


Generation 25 :



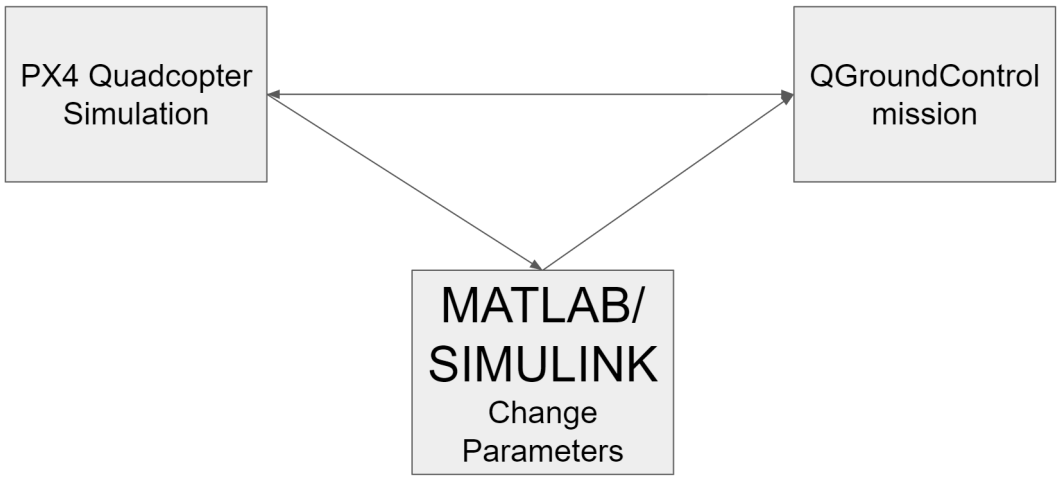
GA Plots

The average value of the objective function and the least value of the objective function in each generation is plotted as below :

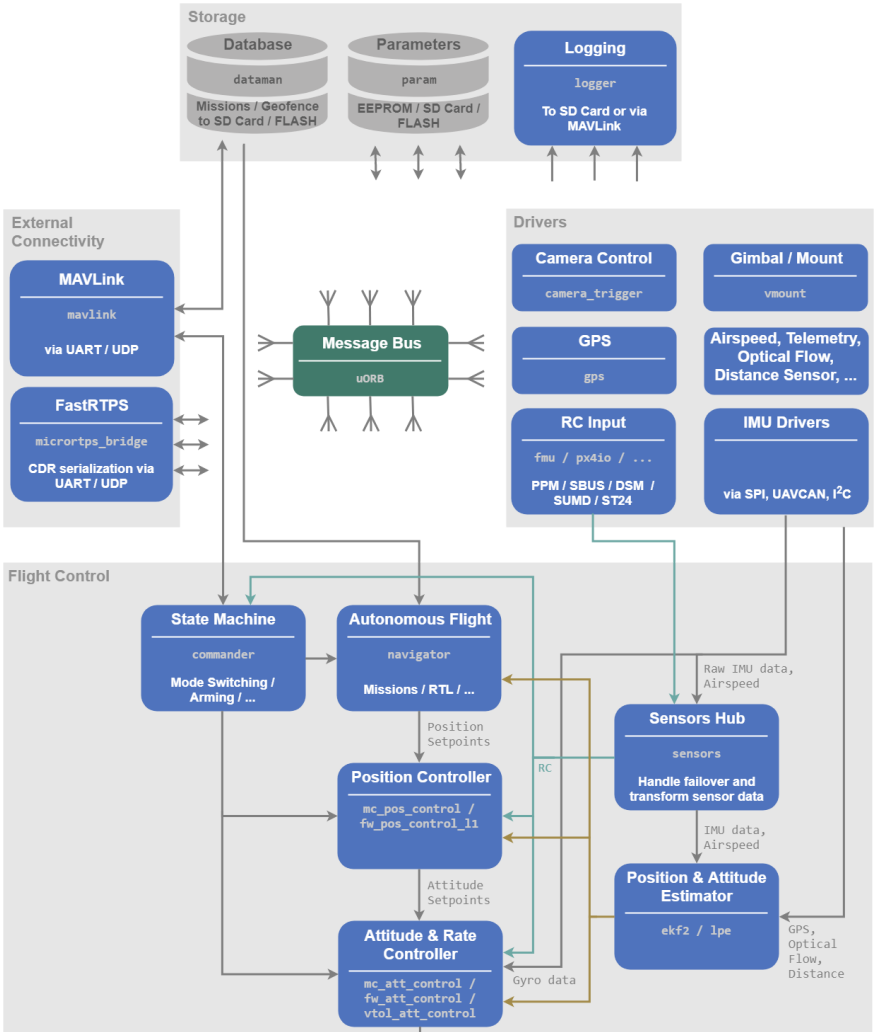


Integrating MATLAB and PX4 Together :

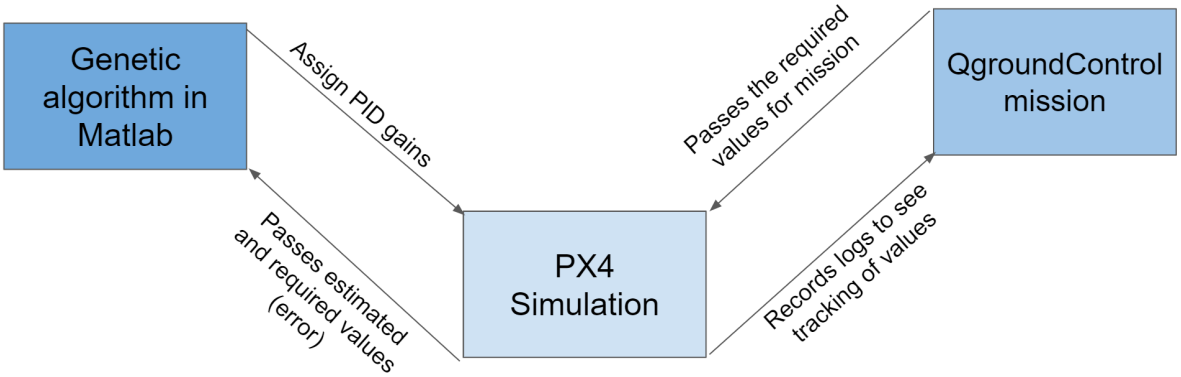
Objective



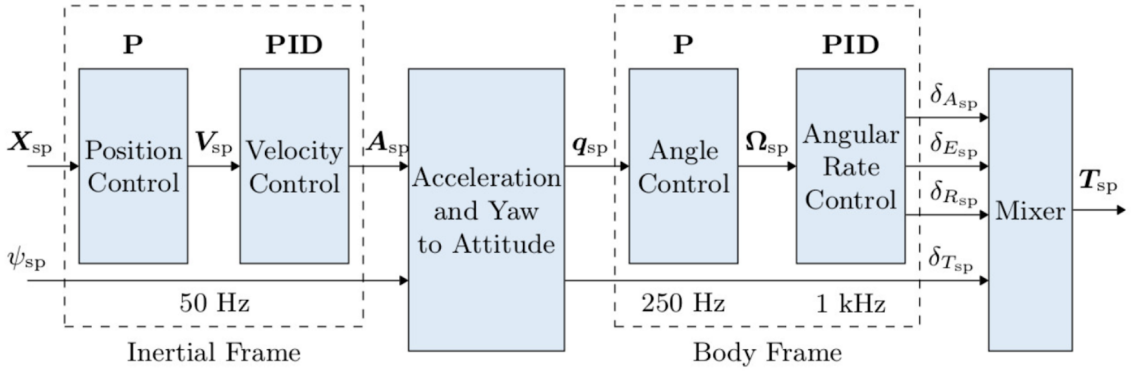
PX4 Messages and their Packaging :



Working of Program

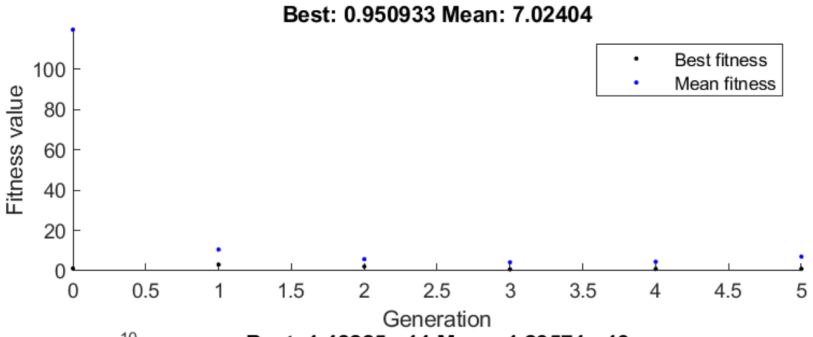


Automatically tuning the PID parameters in Attitude Rate controller to an optimal value.

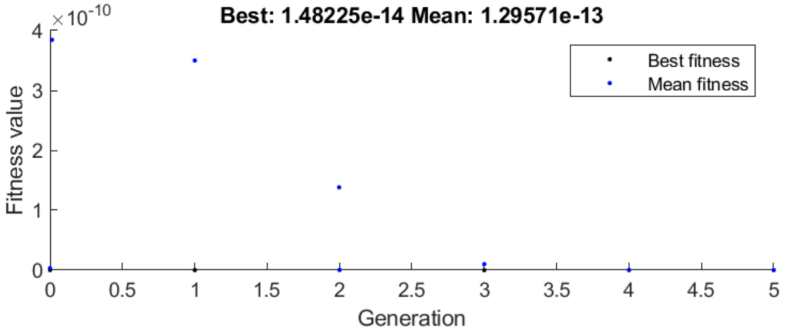


GA Plot :

Mission Flight Mode

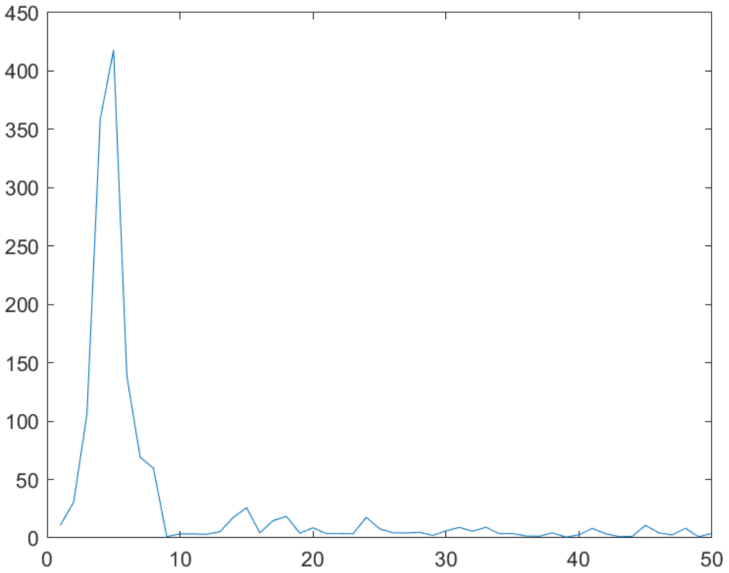


Manual Flight Mode

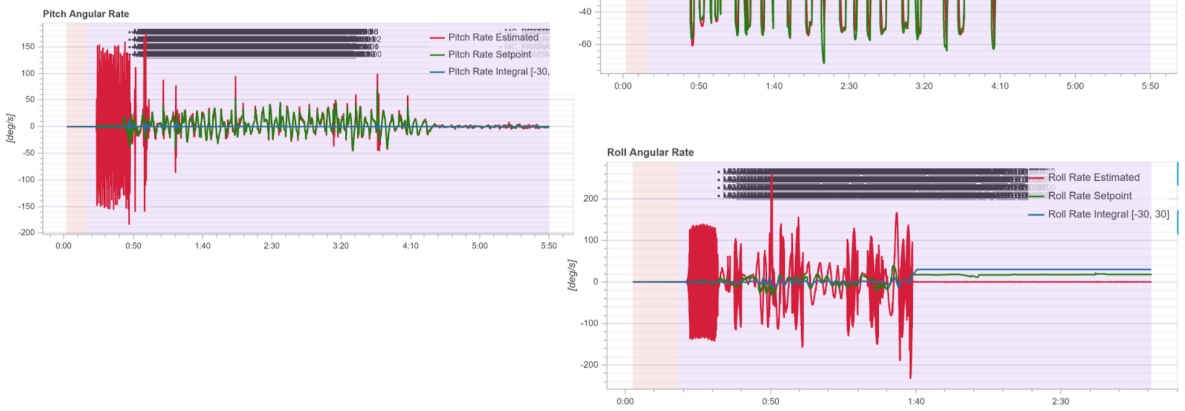


Fitness Plot :

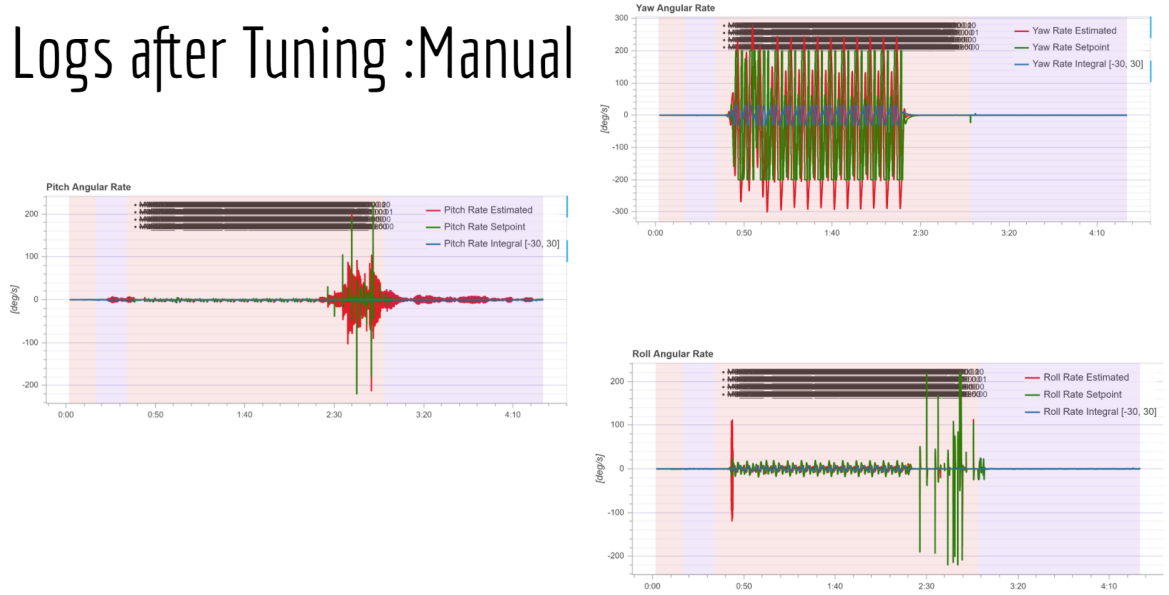
At the end, we assign least fitness PID gains to the system.



Logs during Tuning :Mission



Logs after Tuning :Manual



3 Problems in Dynamics and Controls

Known : u(t), G Find y(t)	Known : G, y(t) Find u(t)	Known : u(t), y(t) Find G

System Identification

It is a form of machine learning.
From past data, we are forming a model that would work for a general case(prediction) for dynamical systems.
Hubble Space Telescope and even the International Space Station were modelled using system identification.

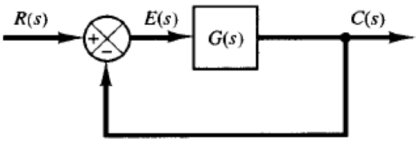
System Identification v/s Model Reduction

Model reduction is reducing the computational complexity of mathematical models.

Here we now the model of our system (long and high dimensional) and we reduce the variables to only ones we require.

Example: Navier - stokes equation reduction.

Full-State Models with Control



Methods of System Identification

- Black Box
  Based on relationship between known input and corresponding output, you infer the system.

- White Box
  You figure out some of the differential equations that govern your system and use system identification to