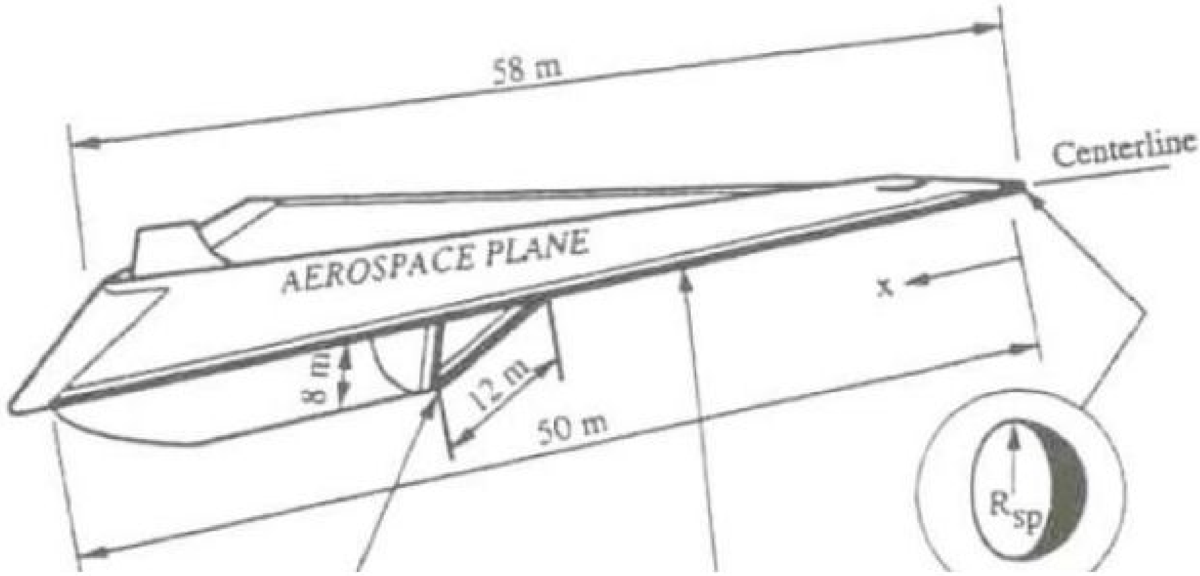
Optimization of Aerospace Plane Ascent Trajectory

SPC 504 - Project Report

Table of Contents

1 Problem Definition

The problem is to optimize the ascent trajectory of the Aerospace Plane until it reaches a certain orbit while satisfying the constraints.



2 System Model

Given the following characteristics:

$I_{sp} = 450\ s$, $C_q = 5.79 \times 10^{-8} \frac{w/cm^2}{(kg/m^3)(m/s)^3}$, $S = 860\ m^2$, $g0 = 9.81$

2.1 States Equations

The model has 6 states: $r, \gamma, V, m, \theta, Q$ and they are defined by the following equations:

$$\dot{r} = V \sin\gamma$$

$$\dot{\gamma} = \frac{L}{mV} + \left(\frac{V}{r} - \frac{g}{V}\right)\cos\gamma$$

$$\dot{V} = \frac{T - D}{m} - g\sin\gamma$$

$$\dot{m} = \frac{-T}{I_{sp} g_0}$$

$$\dot{\theta} = \frac{V\cos\gamma}{r}$$

$$\dot{Q} = C_q\sqrt{\rho}V^3$$

Where:

$$D = \tfrac{1}{2}\, C_D\, \rho\, S\, V^2\ ,\ C_D = 0.080505 - 0.03026\, C_L + 0.086495\, {C_L}^2$$

$$L = \tfrac{1}{2}\, C_L\, \rho\, S\, V^2\ ,\ C_L = -\,0.0041065 + 0.933\,\alpha + 0.854\,\alpha^2$$

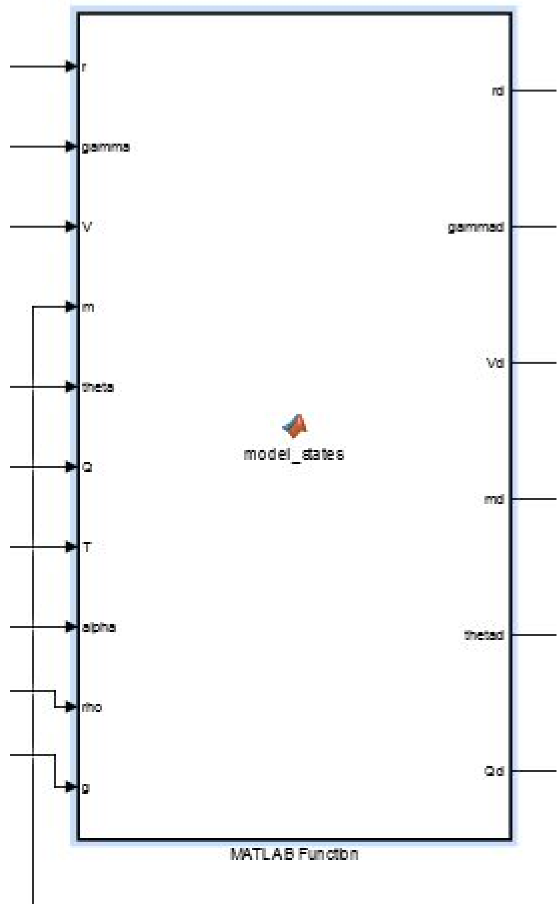
The 6 output states are being integrated and input back to the system given the following initial conditions:

$$r_{initial} = 20000 + 6378000 = 6398000$$

$$V_{initial} = 5\ mach\ =\ 295.1 \times 5 = 1475.5\ \tfrac{m}{s}$$

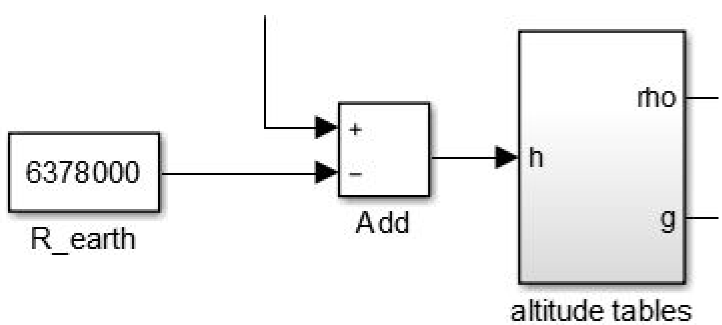
$$m_0 = 0.9 \times 454000 = 408600\ kg$$

$$\gamma_{initial} = 0.088\ rad$$



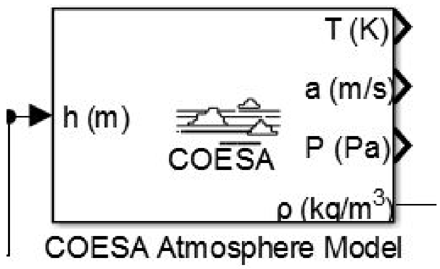
2.2 Calculating ρ and g from Altitude Tables

ρ and g are needed in the calculations and are entered as inputs to the model from a curve fitting model. The output state r is being integrated and then we subtract the radius of the earth from it to calculate our current altitude.



From this altitude, we can calculate ρ and g.

We assume g to be constant with value 9.8 during the whole trajectory, and we get the density values from aerospace toolbox.



2.3 Input Thrust and Angle of Attack

Our inputs to the model are T and α so we add them as inputs in our model along with the 6 states and ρ and g.

We define both of them as a timeseries, consisting of ten points. This time series interpolates all the inner spaces so we will have continuous values for T and α with time.

3 Constraints

3.1 Values

Heat rate per unit area: $\dot{Q}_{max} = 100 \frac{w}{cm^2}$

Heat per unit area: $Q_{max} = 350000 \frac{J}{cm^2}$

Load factor: $n_{max} = 2g$

Final velocity equals orbital velocity: $V_{orb} = 7860\ m/s$

Final altitude: $h_{final} = 75000\ m$

Final time: $t_{final} > 500\ s$

Maximum angle of attack: $\alpha_{max} = 11.5\ deg$

Maximum thrust: $T_{max} = m0\ g$

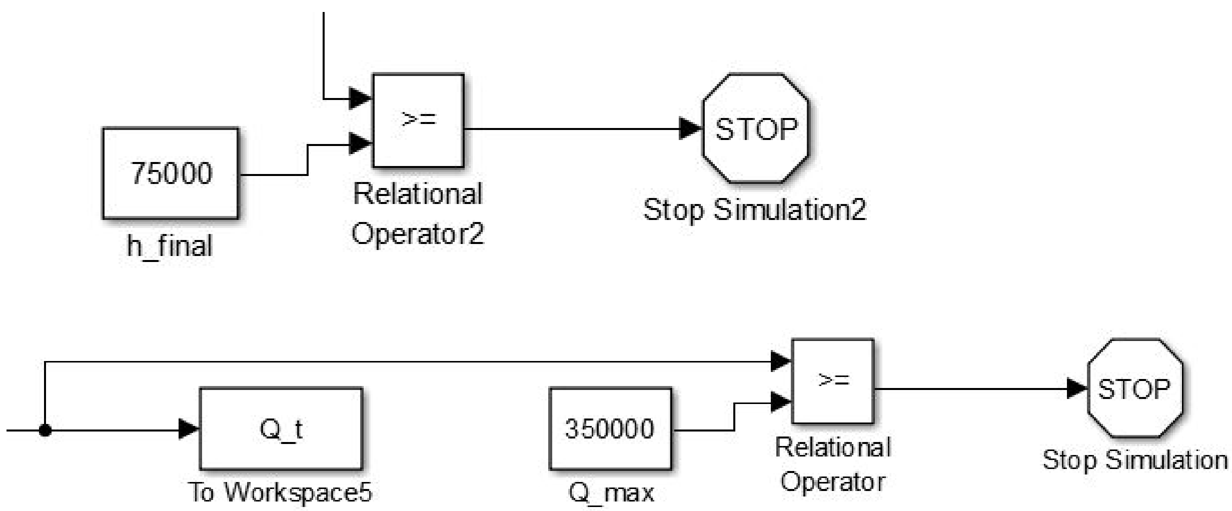
Maximum predefined dynamic pressure: $q_{max} = 20\ kPa$

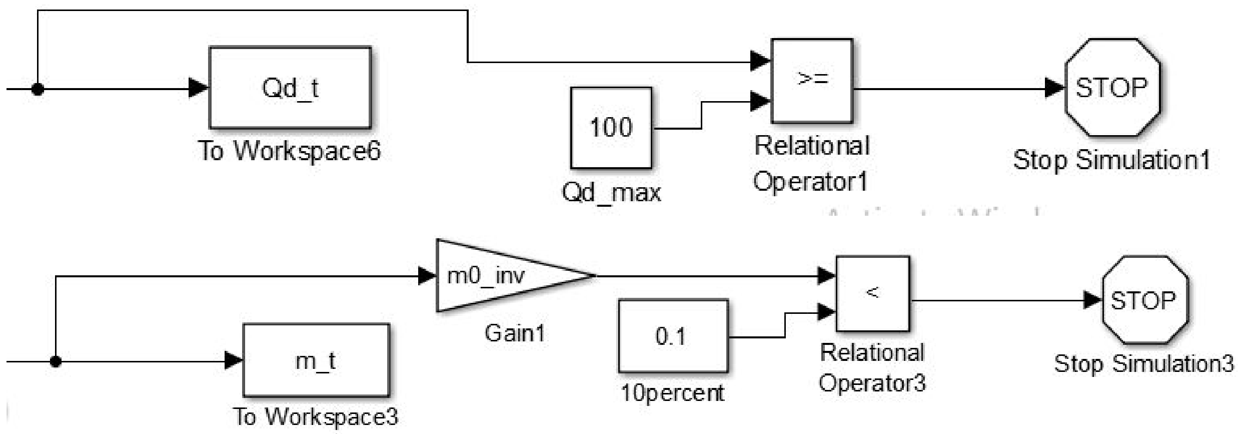
3.2 Implementation

Those constraints were taken into account differently:

Simulink Stop Condition

$h, Q, \dot{Q}, m$ were best constrained in the simulink model itself as shown below, so that the simulation stops once they exceed the maximum value or go below the minimum value.





Optimization Boundaries

Both α and T were constrained using the optimization toolbox boundaries.

Cost Function Constraints

Constraints on states like mass, radius, velocity, heat dissipation, heat dissipation rate and time are to be taken into consideration in the cost function.

4 Optimization

4.1 Inputs

My optimization function has an input of a vector of length 10. The first five points represent angle of attack values in radian, while the second five points represent thrust values divided by the initial mass. However we rescaled those inputs and made the range of the angle of attack inputs to be from -1 to 1 and that of the thrust to be from 0 to 1. Which means that all input values are multiplied by the maximum value of that variable in the code.

The time is specified to be from 0 to 900 seconds.

4.2 Cost Function

After normalizing the scale of our variables, testing many weight combinations, and taking into account the rest of our constraints, we choose the cost function to be:

$$J = -10\frac{m_{end}}{m_0} + 2500\frac{\int(\left|V_t - V_{final}\right| \cdot \tau^4)}{V_{final}} + 200\frac{\int(\left|(r_t - r_{earth}) - h_{final}\right| \cdot \tau^4)}{h_{final}} - 18000\frac{t_{end}}{t_{min}} + 20(3.5\frac{max(Q_t)}{Q_{max}} + \frac{max(\dot{Q}_t)}{\dot{Q}_{max}})$$

Where τ is the time from 0 to 1 normalized

```
tao=round(linspace(0,1,length(V_t)).^7,3);
```

4.3 Run

Files

You need to have five files in your path: “model63.slx”, “opt63_run.m”, “opt63.m”, “opt63_test.m”, and “gaOuput.m”. The first step is to run “opt63_run.m” file in order to declare and/or initialize the global variables. The second step is to launch “optimtool” and to specify “@opt63” to be the objective function.

After the optimization finishes we can use “opt63_test” to plot our results.

Boundaries

Lower bound is: [-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 0.1020 0.1020 0.1020 0.1020 0.1020 0.1020 0.1020 0.1020 0.1020 0.1020 0.1020]

Upper bound is: [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]

5 Results

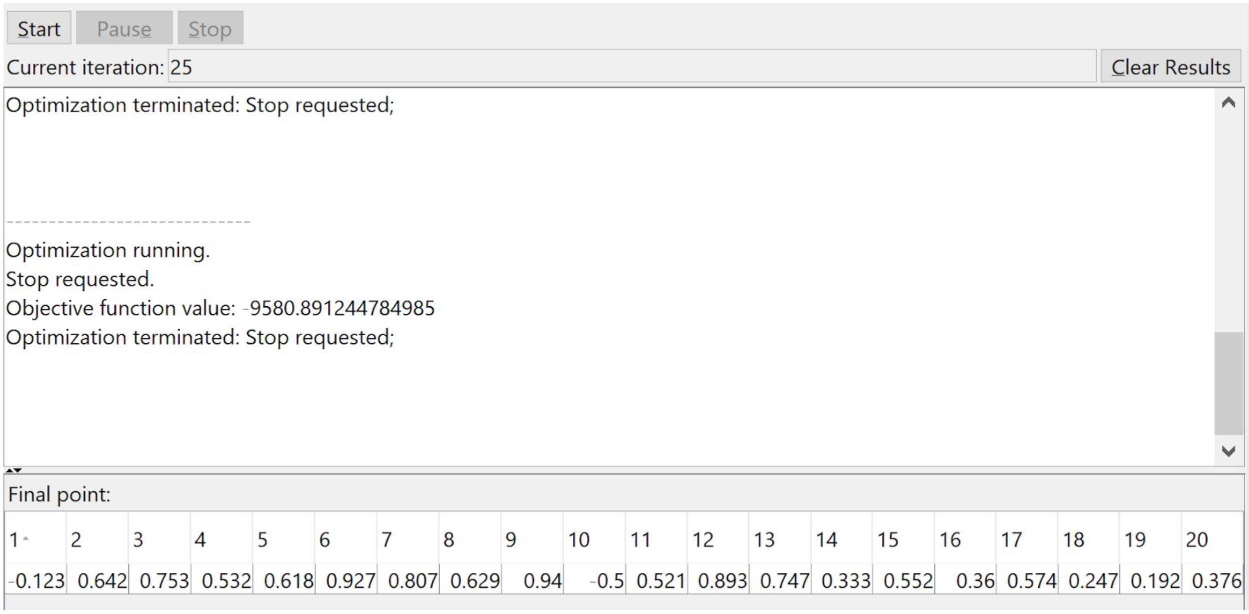
5.1 Optimizer Run

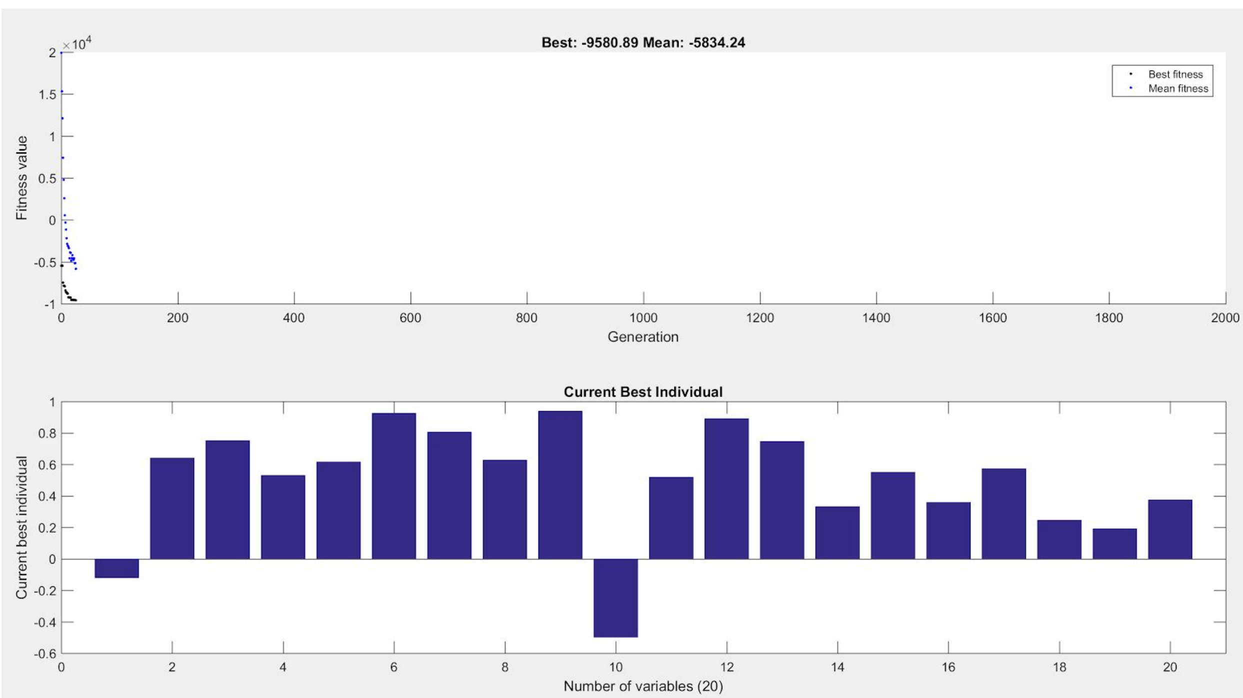
At first I tried to optimize using “fmincon” but this was not working properly. So I ended up using “ga - Genetic Algorithm”, with number of variables of 20.

We can also use “@gaOutput” as an output function for the genetic algorithm so that it saves all the input values so we can check them later on.

5.2 Optimization Tool Results

Function Value Plot





Final Point

x =

Columns 1 through 15

-0.1932	0.6425	0.6812	0.5320	0.9966	0.7960	0.8070	0.6294	-0.0599	0.4998	0.5208	0.8928	0.7475	0.3335	0.5518
-0.1932	0.6425	0.6812	0.5320	0.9966	0.7960	0.8070	0.6294	-0.0599	0.4998	0.5208	0.8928	0.7475	0.3335	0.5518
-0.1932	0.6425	0.6812	0.5320	0.9966	0.7960	0.8070	0.6294	-0.0599	0.4998	0.5208	0.8928	0.7475	0.3335	0.5518
-0.1932	0.6425	0.9835	0.5320	0.9966	0.7960	0.8070	0.6294	0.4596	-0.5002	0.5208	0.8928	0.7475	0.3335	0.4924
-0.1932	0.8023	0.9796	0.5320	0.9966	0.7960	0.8070	0.6294	0.4596	-0.5002	0.5208	0.8928	0.7475	0.3335	0.4924
-0.1932	0.8023	0.9796	0.5320	0.9966	0.7960	0.8070	0.6294	0.4596	-0.5002	0.5208	0.8928	0.7475	0.3335	0.4924
-0.1488	0.6425	0.7528	0.5320	0.9966	0.9274	0.8070	0.6294	0.8637	-0.5002	0.5208	0.8129	0.7475	0.3335	0.6360
-0.1231	0.6425	0.6812	0.5320	0.6491	0.7960	0.4895	0.6150	0.9401	0.4998	0.5208	0.8928	0.7475	0.3335	0.5518
-0.1488	0.8023	0.6812	0.5320	0.9966	0.9274	0.8070	0.6294	0.8637	-0.5002	0.5208	0.8928	0.7475	0.3335	0.4924
-0.1488	0.8023	0.6812	0.5320	0.9966	0.9274	0.8070	0.6294	0.8637	-0.5002	0.5208	0.8928	0.7475	0.3335	0.4924
-0.1488	0.8023	0.7528	0.5320	0.9966	0.9274	0.8070	0.6294	0.8637	-0.5002	0.5208	0.8928	0.7475	0.3335	0.4924
-0.1488	0.8023	0.6812	0.5320	0.6491	0.9274	0.8070	0.6294	0.8637	-0.5002	0.5208	0.8928	0.7475	0.3335	0.4924
-0.1488	0.8023	0.7528	0.5320	0.9966	0.9274	0.8070	0.6294	0.8637	0.4998	0.5208	0.8928	0.7475	0.3335	0.4924
-0.1488	0.8023	0.7528	0.5320	0.9966	0.9274	0.8070	0.6294	0.8637	0.4998	0.5208	0.8928	0.7475	0.3335	0.4924
-0.1488	0.8023	0.7528	0.5320	0.9966	0.9274	0.8070	0.6294	0.9401	0.4998	0.5208	0.8928	0.7475	0.3335	0.4924
-0.1231	0.6425	0.6812	0.5320	0.6491	0.7960	0.4895	0.6150	0.9401	-0.5002	0.5208	0.8928	0.7475	0.3335	0.4924
-0.1231	0.6425	0.6812	0.5320	0.6491	0.7960	0.4895	0.6150	0.9401	-0.5002	0.5208	0.8928	0.7475	0.3335	0.4924
-0.1231	0.6425	0.7528	0.5320	0.6491	0.9274	0.8070	0.6294	0.9401	-0.5002	0.5208	0.8928	0.7475	0.3335	0.5518
-0.1231	0.6425	0.7528	0.5320	0.6491	0.9274	0.8070	0.6294	0.9401	-0.5002	0.5208	0.8928	0.7475	0.3335	0.5518
-0.1231	0.6425	0.7528	0.5320	0.6491	0.9274	0.8070	0.6294	0.9401	-0.5002	0.5208	0.8928	0.7475	0.3335	0.5518
-0.1231	0.6425	0.7528	0.5320	0.6491	0.9274	0.8070	0.6294	0.9401	-0.5002	0.5208	0.8928	0.7475	0.3335	0.5518
-0.1231	0.6425	0.7528	0.5320	0.6491	0.9274	0.8070	0.6294	0.9401	-0.5002	0.5208	0.8928	0.7475	0.3335	0.5518
-0.1231	0.6425	0.7528	0.5320	0.6647	0.8116	0.8070	0.7544	0.9401	-0.5002	0.5208	0.8928	0.7475	0.3335	0.5518
-0.1231	0.6425	0.7528	0.5320	0.6647	0.8116	0.8070	0.7544	0.9401	-0.5002	0.5208	0.8928	0.7475	0.3335	0.5518
-0.1231	0.6425	0.7528	0.5320	0.6647	0.8116	0.8070	0.7544	0.9401	-0.5002	0.5208	0.8928	0.7475	0.3335	0.5518
-0.1231	0.6425	0.7528	0.5320	0.6178	0.9274	0.8070	0.6294	0.9401	-0.5002	0.5208	0.8928	0.7475	0.3335	0.5518
-0.1231	0.6425	0.7528	0.5320	0.6178	0.9274	0.8070	0.6294	0.9401	-0.5002	0.5208	0.8928	0.7475	0.3335	0.5518

Columns 16 through 20

0.3603	0.6954	0.2936	0.1922	0.6046
0.3603	0.6954	0.2936	0.1922	0.6046
0.3603	0.6954	0.2936	0.1922	0.6046
0.3603	0.6954	0.2467	0.1922	0.3757
0.3603	0.5742	0.2467	0.1922	0.1624
0.3603	0.5742	0.2467	0.1922	0.1624
0.1873	0.6954	0.2467	0.1922	0.3757
0.3603	0.3745	0.2936	0.1922	0.4124
0.3603	0.5742	0.2467	0.1922	0.1624
0.3603	0.5742	0.2467	0.1922	0.1624
0.3603	0.5742	0.2467	0.1922	0.3757
0.3603	0.5742	0.2467	0.1922	0.1624
0.3603	0.5742	0.2936	0.1922	0.3757
0.3603	0.5742	0.2936	0.1922	0.3757
0.3603	0.5742	0.2936	0.1922	0.3757
0.3603	0.5742	0.2936	0.1922	0.4124
0.3603	0.5742	0.2936	0.1922	0.4124
0.3603	0.5742	0.2467	0.1922	0.3757
0.3603	0.5742	0.2467	0.1922	0.3757
0.3603	0.5742	0.2467	0.1922	0.3757
0.3603	0.5742	0.2467	0.1922	0.3757
0.3603	0.5742	0.2467	0.1922	0.3757
0.3603	0.5742	0.2467	0.1922	0.3757
0.3603	0.5742	0.2467	0.1922	0.3757
0.3603	0.5742	0.2467	0.1922	0.3757
0.3603	0.5742	0.2467	0.1922	0.3757
0.3603	0.5742	0.2467	0.1922	0.3757

5.3 Results

Final Values

```
>> opt63_test(x(end,:))
Qt_end =
   3.2585e+04
Qdt_end =
  100.0000
mt_end =
   4.3442e+04
Vt_end =
   6.6052e+03
ht_end =
   7.4807e+04
tt_end =
  780.9340
```

$Q_{end} = 32585$

$\dot{Q}_{end} = 100$

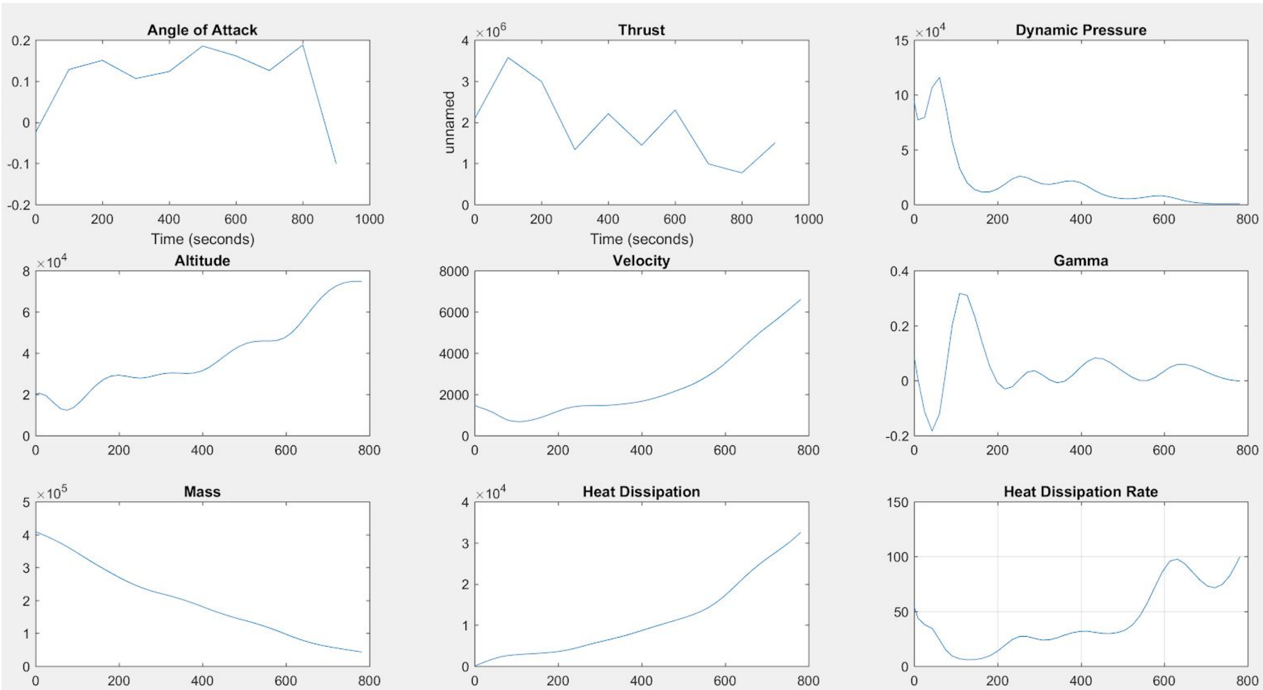
$m_{end} = 43442$

$V_{end} = 6605.2$ where $V_{final} = 7860$

$h_{end} = 74807$ where $h_{final} = 75000$

$t_{end} = 780.934$

Graphs



6 Issues Summary

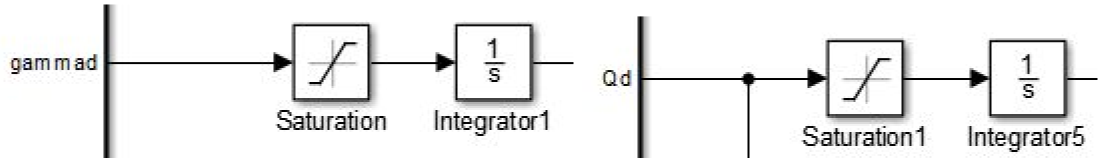
At first the optimizer was not working properly and resulted in the occurrence of this error many times:

> fmincon stopped because the size of the current step is less than the default value of the step size tolerance and constraints are satisfied to within the default value of the constraint tolerance.

In order to solve this issue, the following steps were taken:

1- Use genetic algorithm instead of "fmincon". This meant that it searched the whole space and did not require a start point.

2- Add saturation blocks on both gamma and Q outputs to limit their values. The gamma upper value was limited to "0.05" and that of Q was limited to "5000" as follows:



3- Set the simulink solver to “automatic”.

4- Use “try - catch” statement so that if the solution failed at any tried point, the code will replace it with a large number to the cost function.

```
try sim('model62.slx');
catch
    J=100000;
    return
end
```

5- In order to make the solution faster, I eliminated the thetadot equation as it was not used. So the system is only 5 states.