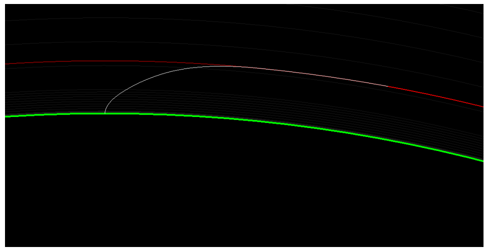

Optimizing the ascent trajectory for an orbital class launch vehicle

Final project in SI1336

Erik Weilow

January 19, 2019

1 Introduction

The goal of this project is to compare the fuel consumption for the launch of a multi-stage orbital class launch vehicle into low Earth orbit, from an equatorial and a polar launch site. To do this, the project formulates an approximate model of the conditions encountered for the launch vehicle during ascent as well as a simple control algorithm. To be able to draw conclusions from the results gained, the control algorithm must be optimized. This will be done using a randomized optimization algorithm. The project is thus split into four major parts: the equations describing the state of the rocket, the physics model, the ascent guidance algorithm as well as the optimization of control parameters.

2 Equations of the rocket

A rocket works on the principle of momentum conservation. Mass is ejected at a rate $\dot{m}$ (mass flow) through the engines at a velocity $v_e = I_{sp}/g_0$, producing a thrust $F = \dot{m} \cdot v_e = \dot{m} \cdot I_{sp}/g_0$. If we can calculate these parameters at all times, we can simulate a rocket.

To complicate things further, we need to consider that the rocket has several stages. This can be solved by letting the state of the rocket be described with position, velocity and *expended mass* (m_e):

$$S = (\vec{r}, \vec{v}, m) = (x, y, z, v_x, v_y, v_z, m_e)$$

where m_e is the integral of exhaust mass flow (ejected mass) over time:

$$m_e(t) = \int_0^t \dot{m}(S, t) dt$$

The reason for including expended mass in the model is that it lets us derive the currently active stage, as well as the necessary variables (thrust, mass and mass flow) of the rocket as a function of used propellant.

2.1 Input parameters

Consider each stage of the rocket as collection of parameters:

Parameter	Description
$m_{prop,i}$	Propellant mass of stage
$m_{dry,i}$	Dry mass of stage (without propellant)
$F_{sl,i}$	Thrust at sea level pressure
$F_{vac,i}$	Thrust in vacuum pressure
$I_{sp,sl,i}$	Specific impulse at sea level
$I_{sp,vac,i}$	Specific impulse in vacuum pressure

such that each stage $i = 0, 1, 2...$ can be written

$$s_i = (m_{prop,i}, m_{dry,i}, F_{sl,i}, F_{vac,i}, I_{sp,sl,i}, I_{sp,vac,i})$$

Furthermore, a rocket typically has an aerodynamic shell (fairing) that is deployed around the edge of space (100km). This is modelled simply by the parameters

Parameter	Description
m_i	Mass of fairing
$h_{deployment,i}$	Altitude of deployment

such that each fairing $i = 0, 1, 2...$ can be written

$$f_i = (m_i, h_{deployment,i})$$

Lastly, the payload of the rocket is modelled as a mass $m_{payload}$.

2.2 Derived parameters

To run simulations of defined by the parameters in the previous section, we need to derive the current mass of the entire rocket. This is not as simple as subtracting m_e from the initial mass, as we want to model staging. Instead consider parameters driven by the currently active stage s_{active}.

2.2.1 Mass

If the propellant of stages before s_i is

$$m'_{prop,i} = \sum_{j=0}^{i-1} m_{prop,j}$$

then the currently active stage fulfill the criteria

$$m'_{prop,active} < m_e(t) < m'_{prop,active} + m_{prop,active}$$

As we are only considering a two stage launch vehicle, the active stage can be defined by

$$s_{active} = \begin{cases} s_0 & : m_e < m_{prop,0} \\ s_1 & : 0 < m_e - m_{prop,0} < m_{prop,1} \end{cases}$$

If the mass of stages after the stage s_i is

$$m'_{stages,i} = \sum_{j=i+1} m_{prop,j} + m_{dry,j}$$

This defines the current mass of stages, when s_i is active, as

$$m_{stagemass,i} = \left(m_e(t) - m'_{prop,i}\right) + m_{dry,i} + m'_{stages,i}$$

If h_{max} is the maximum altitude reached up until time t, then the mass of fairings currently on the rocket can be described by the sum

$$m_{fairings}(h_{max}) = \sum_{i=0} m_i(1 - H(h_{max} - h_{deployment,i}))$$

Thus, the total instantaneous mass for an active stage s_i is

$$m_i = m_{stagemass,i} + m_{fairings}(h_{max}) + m_{payload}$$

2.2.2 Thrust

Since the thrust of a rocket propulsion system is linear in the pressure difference between the exhaust and surrounding pressure, it is assumed in our model that the thrust F changes linearly between F_{sea} and F_{vacuum} with pressure p.

For a given active stage s_i, if pressure is written as a function of altitude h, thrust can be written as

$$F_i(p(h)) = F_{sl,i} + (F_{sl,i} - F_{vac,i})\,\frac{p(h)}{p(0)}$$

This only holds if the stage has remaining fuel ($m_e - m_{prop,i} > 0$) otherwise

$$F_i(p(h)) = 0$$

2.2.3 Mass flow

Mass flow is necessary to find $m_e(t)$, and is given in a general form by

$$\dot{m} = \frac{F}{g_0 I_{sp}}$$

If we assume mass flow to be constant through the propulsion system, then I_{sp} must share the same linear behaviour in pressure as thrust does. Thus, for an active stage s_i, let

$$I_{sp,i}(p(h)) = I_{sp,sl,i} + (I_{sp,sl,i} - I_{sp,vac,i}) \frac{p(h)}{p(0)}$$

This gives the mass flow

$$\dot{m}_i(p(h)) = \frac{F_i(p(h))}{g_0 I_{sp,i}(p(h))}$$

2.2.4 Summarization

To tie the equations together, we define the functions

$$m(m_e) = m_i, \quad F(m_e, p(h)) = F_i(p(h)), \quad \dot{m}(m_e, p(h)) = \dot{m}_i(p(h))$$

where the active stage s_i is derived from a given m_e.

3 Physics model

To simulate the ascent of the rocket, a model of the physics involved is required.

3.1 Coordinate system

The simulation uses two coordinate systems, one cartesian and one kinematic. The cartesian system $\hat{x}, \hat{y}, \hat{z}$ is originated in the starting location and oriented such that $\hat{x}$ points towards the wanted direction of orbit, $\hat{y}$ points radially up, and $\hat{z} = \hat{x} \times \hat{y}$.

The kinematic system $\hat{r}, \hat{t}, \hat{z}$ follows the rocket and is oriented such that $\hat{r}$ is the normalized radial vector, $\hat{z}$ is the same as in the cartesian system, and the tangential vector is $\hat{t} = \hat{r} \times \hat{z}$. In a circular orbit, $\hat{t}$ is parallel to velocity $\vec{v}$.

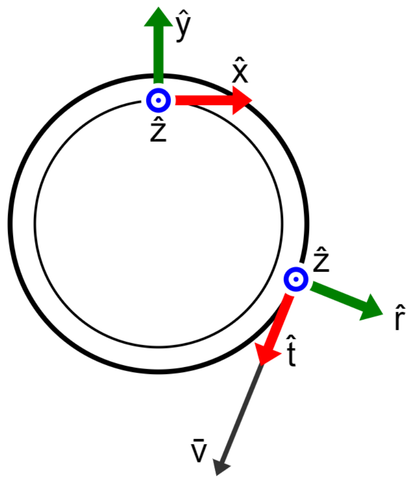

Figure 1: The instantaneous coordinates

3.2 Atmosphere

Produces $\rho(r)$

3.3 Forces

In this model, it assumed that three forces are acting on the rocket: thrust T, aerodynamic drag D and gravity G.

3.3.1 Gravity - G

Gravity is modelled based on the Newtonian formulation, resulting in a force

$$\vec{G}(\vec{r}, m_e) = -m(m_e)\frac{\mu}{r^2}\hat{r}$$

where r is the distance to the center of Earth from the rocket, and $\mu \approx 3.986 \cdot 10^{14} m^3 s^2$ is the standard gravitational parameter.

3.3.2 Aerodynamic drag - D

To model aerodynamic drag it first is assumed that the atmosphere, independently of radius, moves at a velocity:

$$\vec{v}_{atm}(\vec{r}) = v_{surf} \cdot \hat{t}$$

This allows the definition of the wind-relative velocity

$$\vec{v}_{atm,rel}(\vec{r},\vec{v}) = \vec{v} - \vec{v}_{atm}(\vec{r})$$

Under the assumption that the the drag equation holds for the entirety of the ascent, then

$$\vec{D}(\vec{r},\vec{v},m_e) = -\frac{m(m_e) \cdot C_d \cdot A \cdot \rho(r)}{2} |\vec{v}_{atm,rel}(\vec{r},\vec{v})|^2 \hat{v}$$

3.3.3 Thrust - T

To abstract the guidance algorithms from the physics model, it is assumed that guidance controls throttle as

$$\eta = \eta(\vec{r},\vec{v},m_e,t) \in [0,1]$$

and angle of thrust (AoT) from the vertical $\hat{r}$ as

$$\theta = \theta(\vec{r},\vec{v},m_e,t) \in [0,\pi]$$

Throttle and AoT interact such that

$$\vec{T}(\vec{r},\vec{v},m_e,t) = \big(\cos(\theta) \cdot \hat{r} + \sin(\theta) \cdot \hat{t}\big) \cdot F(m_e,P) \cdot \eta$$

3.4 Differential equation

The combination of gravity, aerodynamic drag, and thrust give the equation

$$\vec{a}(\vec{r},\vec{v},m_e) = \frac{1}{m_i} \left(G(\vec{r},m_e) + D(\vec{r},\vec{v},m_e) + T(\vec{r},\vec{v},m_e,t)\right)$$

4 Ascent guidance

The ascent guidance implemented in the simulation can be categorized into five phases:

- Liftoff
- Kickpitch
- Gravity turn
- Orbital insertion

- Terminal guidance

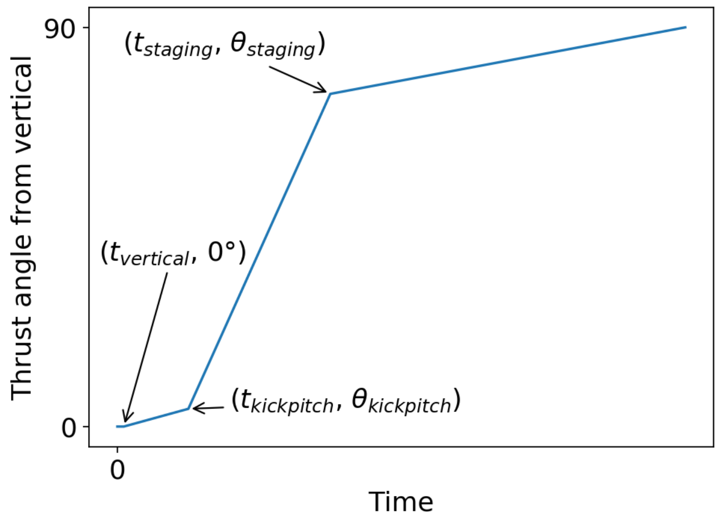

Figure 2: Typical thrust angle from vertical (intentionally without units)

4.1 Liftoff $(0 < t < t_{vertical})$

During this phase, the trajectory is entirely vertical - $\theta = 0$ and $\eta = 1$.

4.2 Kickpitch $(t_{vertical} < t < t_{kickpitch})$

During this phase, the trajectory starts pitching towards the horizon slightly. The angle θ is linearly interpolated between 0 and the defining parameter $\theta_{kickpitch} = 4°$ according to

$$\theta(t) = \theta_{kickpitch} \frac{t - t_{vertical}}{t_{kickpitch} - t_{vertical}}$$

4.3 Gravity turn $(\dot{m} t_{kickpitch} < m_e(t) < m_{prop,0})$

During the time between the end of kickpitch and the time of staging (when the expended mass is equal to the propellant in the first stage), the angle θ is linearly interpolated between $\theta_{kickpitch}$ and the defining parameter $\theta_{staging}$

according to

$$\theta(t) = \theta_{kickpitch} + (\theta_{staging} - \theta_{kickpitch})\,\frac{m_e(t) - \dot{m}t_{kickpitch}}{m_{prop,0} - \dot{m}t_{kickpitch}}$$

4.4 Orbital insertion ($m_{prop,0} < m_e(t) < m_{prop,0} + m_{prop,1}$)

This phase lasts until terminal guidance is triggered. The angle θ is interpolated from $\theta_{staging}$ to 0 according to

$$\theta(t) = \theta_{staging}\left(1 - \frac{m_e(t) - m_{prop,0}}{m_{prop,1}}\right)$$

4.5 Terminal guidance

Terminal guidance is necessary to achieve a circular orbit within this simulation and previously mentioned ascent guidance parameters. It consists of a PI-controller that attempts to cancel out any vertical velocity.

The conditions for entering terminal guidance is:

- $h > g_{ap} - 55$
- OR:
- $h > 100$ and $|\vec{v} \cdot \hat{r}| < 50$

Within terminal guidance, θ is controlled with the regulator

$$\theta = 5 \cdot e(t) + 5\int_{t_{triggered}}^{t} e(t')dt'$$

where $t_{triggered}$ is the time at which terminal guidance was triggered, and

$$e(t) = \vec{v} \cdot \hat{r}$$

Furthermore, throttle is controlled by

$$\eta = \begin{cases} 0 & : \quad r_{pe} > g_{pe} - 2km, \quad g_{ap} - 2km < r_{ap} < g_{ap} + 3km \\ 0 & : \quad r_{pe} > g_{pe} - 2km, |\vec{v} \cdot \hat{r}| < 15km \\ 0.05 & : \quad r_{pe} > g_{pe} - 10km \\ 0.25 & : \quad r_{pe} > 0km \\ 0.5 & : \quad r_{pe} > -100km \\ 1 & : \quad otherwise \end{cases}$$

If $\eta = 0$, the simulation ends in orbit.

5 Integration

The integration was done using a 4th order Runge-Kutta integrator on expended mass m_e, velocity $\vec{v}$ and position $\vec{r}$. This choice was made, because even though RK4 is not stable in terms of energy, it is simple to implement and produces low errors for the relatively short time spans simulated during a rocket launch (in the order of 500 seconds). Relatively few large time steps are required to get a stable simulation, which is very useful when running an optimization algorithm, compared to other methods.

The choice of time step for the optimization step is detailed in the next section, but the final simulation is run in scaled real-time: such that we do N steps per frame $\Delta t_f = 1/60s$, resulting in a time step of

$$\Delta t = 1/n\Delta t_f = 1/(60N)s$$

The simulation is then run as an animation with speed-up n, resulting in $n \cdot N$ time steps taken per frame. Going much higher, up to 10x more steps per frame, would be computationally possible but introduces numerical inaccuracies. Being able to run the simulation in real-time means that it could also, in theory, be run on a real launch vehicle.

6 Optimization of parameters

To optimize parameters, an error function has to be established. From introduction, we know that the goal is to reach a specific periapsis (g_{pe}) and apoapsis (g_{ap}). Let the periapsis from each simulation be r_{pe}, and the periapsis be r_{ap}. As the terminal guidance algorithm works by eliminating vertical velocity, after reaching maximum altitude, we also want to make sure to use the maximum altitude reached (h_{max}). To make sure the terminal guidance algorithm works well, we also want to use the altitude at which orbit is reached (h). Since the project goal is to look at fuel usage for equatorial versus polar launch, it would be good to also use the fuel remaining after orbit insertion (m_{final}).

This lets us define the error function

$$E(r_{pe}, r_{ap}, g_{pe}, g_{ap}, h, h_{max}, m_{final}) = \frac{(r_{ap} - g_{ap})^2}{(10)^2} + \frac{(r_{pe} - g_{pe})^2}{(10)^2} + \frac{(h_{max} - g_{pe})^2}{(10)^2} + \frac{(h - g_{pe})^2}{(10)^2} + m_{final}$$

This error function is used to score control algorithm parameters using a randomized optimization method, in which the ascent control parameters described below are randomly perturbated from the currently best choice of parameters using specific random radii and scaling, such that for each one: $parameter_{i+1} = parameter_i + scale \cdot \text{random}(-radii, radii)$

Parameter	Random radius
$t_{vertical}$	1
$t_{kickpitch}$	4
$\theta_{staging}$	2
a_{max}	0.1

For every perturbation, a simulation is run at a specific timestep Δt. If the result is scored lower than the currently best score, the currently best choice of parameters are set to the perturbated ones. If the result is not scored lower, the perturbation is discarded. This is done for different sets of time steps and scaling factors, progressively going to a finer time step, until either a max count of attempts have been reached or a minimum score is achieved according to the following sequence:

Maximum score	Maximum runs	Δt	Random scale
10	10000	1	30
5	10000	1	15
5	10000	0.5	4
2	10000	0.5	2
1	10000	0.2	1
0.5	10000	0.1	0.5

The parameters that is considered best at the end of optimization is then used in a more accurate simulation to produce results.

7 Results

We'll start by looking at how well the guidance algorithm worked. Three goal orbits was optimized in the program, and we'll look at both how close it guided the rocket into the goal orbit as well as how much propellant was consumed.

7.1 Orbit insertion accuracy

		Achieved	
Surface motion	Goal altitude	Perigee	Apogee
0 m/s	170 km	168.21 km	169.76 km
460 m/s	170 km	168.17 km	169.49 km
0 m/s	220 km	218.87 km	219.39 km
460 m/s	220 km	218.93 km	223.90 km
0 m/s	275 km	274.85 km	274.88 km
460 m/s	275 km	273.93 km	274.38 km

It's clear that the algorithm works astonishingly well, with insertions just a few km in difference from the goal. It also works well for both equatorial and polar launch.

7.2 Propellant consumption

		Propellant	
Surface motion	Goal altitude	Consumed	Remaining
0 m/s	170 km	511909 kg	6490 kg
460 m/s	170 km	510103 kg	8297 kg
0 m/s	220 km	516129 kg	2271 kg
460 m/s	220 km	512725 kg	5675 kg
0 m/s	275 km	516911 kg	1489 kg
460 m/s	275 km	515460 kg	2940 kg

The propellant consumption was highest for the highest orbit goal, a reasonable expectation, but the difference between equatorial and polar launch was the lowest for the 220 km orbit goal. This is further shown in the following table, comparing propellant usage for equatorial launch relative to polar launch:

	Propellant	
Goal altitude	Consumed	Remaining
170 km	99.64%	127.8 %
220 km	99.34%	250.0 %
275 km	99.72%	197.4 %

8 Discussion

By the results achieved, it's clear that orbital launch from the equator is better than launch from the poles. It is however interesting that the fuel consumption, for the orbit insertions simulated, is the most equal for a target orbit of 170 km. This is most likely not related to the physics of the problem, but rather an effect of the way that the trajectory is created and optimized.

One must be careful just looking at the results however, as one might say that it is not by a huge margin that equatorial launch is better, but that's where the tyranny of the rocket equation comes into play. This can be shown by introducing Tsiolkovsky's rocket equation

$$\Delta v = v_{exhaust} \ln \frac{m_{final}}{m_{initial}}$$

Solving for $m_{initial}$ gives

$$m_{initial} = m_{final} \exp\left(-\frac{\Delta v}{v_{exhaust}}\right)$$

If the goal is to accelerate 10000 kg of payload into a new velocity of 6 km/s with the second stage defined in the simulations, then reducing the fuel consumption by 2000 kg allows the usage of 10000 kg less fuel.

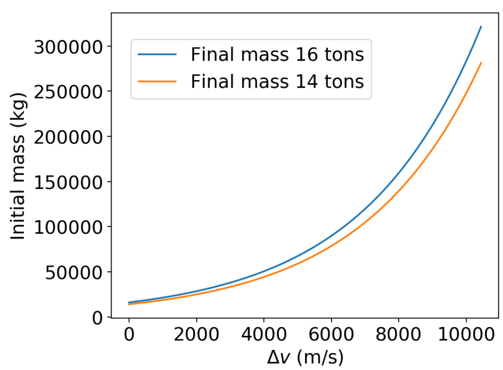

Figure 3: The initial mass necessary for the $v_{exhaust}$ used in simulation, for different final masses

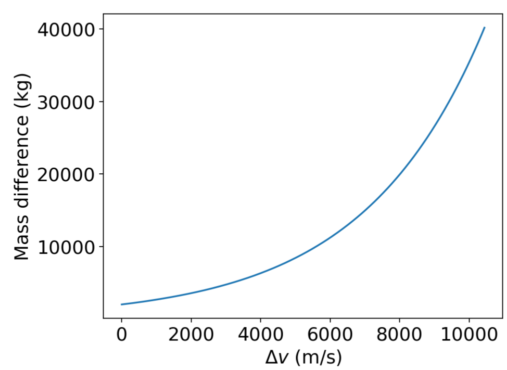

Figure 4: The difference in initial mass for final masses of 14 and 16 tons

It was also interesting to see that the *very* simple guidance algorithm was able to guide the rocket into an orbit very close to the goal. It is definitely just scraping the surface of what is possible.

A Raw data

A.1 170 km orbit

	Goal		Achieved	
Surface motion	Perigee	Apogee	Perigee	Apogee
0 m/s	170 km	170 km	168.21 km	169.76 km
460 m/s	170 km	170 km	168.17 km	169.49 km

		Propellant	
Surface motion	Payload	Remaining	Consumed
0 m/s	10000 kg	6490.91 kg	511909 kg
460 m/s	10000 kg	8297.18 kg	510103 kg
		127.82%	99.64%

A.2 220 km orbit

	Goal		Achieved	
Surface motion	Perigee	Apogee	Perigee	Apogee
0 m/s	220 km	220 km	218.87 km	219.39 km
460 m/s	220 km	220 km	218.93 km	223.90 km

		Propellant	
Surface motion	Payload	Remaining	Consumed
0 m/s	10000 kg	2271.10 kg	516129 kg
460 m/s	10000 kg	5675.03 kg	512725 kg
		250%	99.34%

A.3 275 km orbit

	Goal		Achieved	
Surface motion	Perigee	Apogee	Perigee	Apogee
0 m/s	275 km	275 km	274.85 km	274.88 km
460 m/s	275 km	275 km	273.93 km	274.38 km

		Propellant	
Surface motion	Payload	Remaining	Consumed
0 m/s	10000 kg	1489.10 kg	516911 kg
460 m/s	10000 kg	2939.79 kg	515460 kg
		197.4%	99.72%