Ontario Tech University
Faculty of Science, Computer Science

Project Group: Rocket Simulation April 6, 2024

Final Group Simulation Project
Topic: Rocket Simulation

by

Malhar Singh Paul Gabriel Katigbak

Introduction

This project aims to examine a simplified version of a rocket simulation. While an exact replica of a rocket simulation would be out of the scope of this course, a simplified version still addresses much of the physics behind rocket simulations. This simulation helps us understand rocket simulation basics without the need for expensive resources. This simulation involves physics laws such as gravity (which changes with distance from the planet's surface), Newton's laws of motion, rigid-body dynamics, optional air resistance (simplified) and a simplified version of thrust and torque. Thrust and torque are simplified since we are ignoring things like fuel consumption, mass flow rate and exhaust velocity of the engine. There is no specific conclusion for this project, but instead, a more general simulation to test out different scenarios which can be further improved.

1 Requirements

Scope: Build an accurate and visually appealing simulation based on the course material.
Problem statement: Effectively implement a Rocket Simulation to analyze the physics involved analytically and visually observe the rocket.
Additionally, important parameters such as linear and angular forces in 3D were considered as a basis for the simulation to get accurate results.

2 Modelling

Figure 1 represents the Free-body diagram of the rocket within this simulation. As seen from this figure, for 3D rigid body rocket simulations, many forces need to be split into their x, y, and z components (since this is visually only 2D, we just have x, y). The forces in this figure (when thrust is applied on an orientation): the force of gravity is always acting down, thrust and drag forces being split into the x (drag(x) opposing thrust(x)) and y (gravity and drag(y) opposing thrust(y)) components and torque to tilt the rocket.

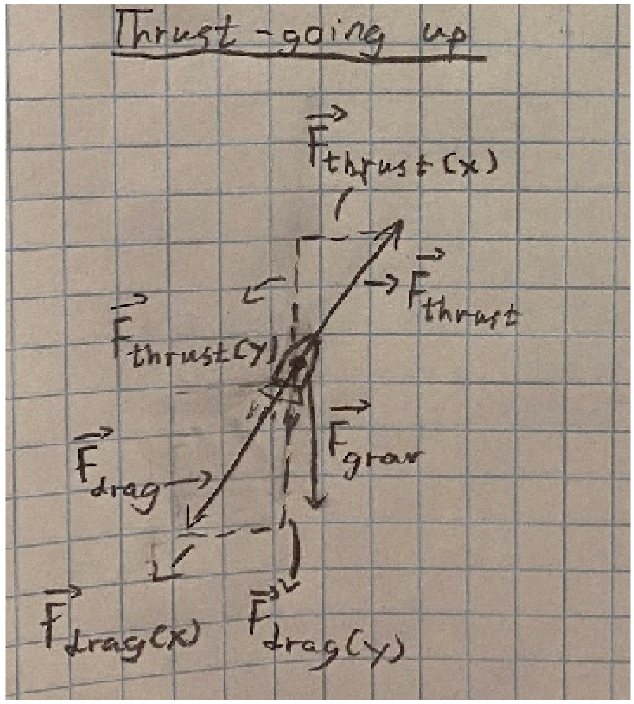


Figure 1: Free Body Diagram of Rocket Simulation

Forces in this simulation for most accurate results:

$$F_x = F_{thrust(x)} \qquad F_y = F_{thrust(y)} - F_{grav}$$

Where:

$F_{drag} = -kv$: k = drag coefficient, v = velocity (in respective direction)

$F_{thrust} = a$: a = constant acceleration (simulates thrust (500 for best results))

$F_{grav} = \frac{-G \cdot M \cdot m}{(r+R)^2}$: G = gravitational constant, M = mass of the planet (Earth in this case), m = mass of rocket, (R + r) = distance between the center of the planet and the rocket.

For additional diagrams, please check out the appendix section for Figure 8.

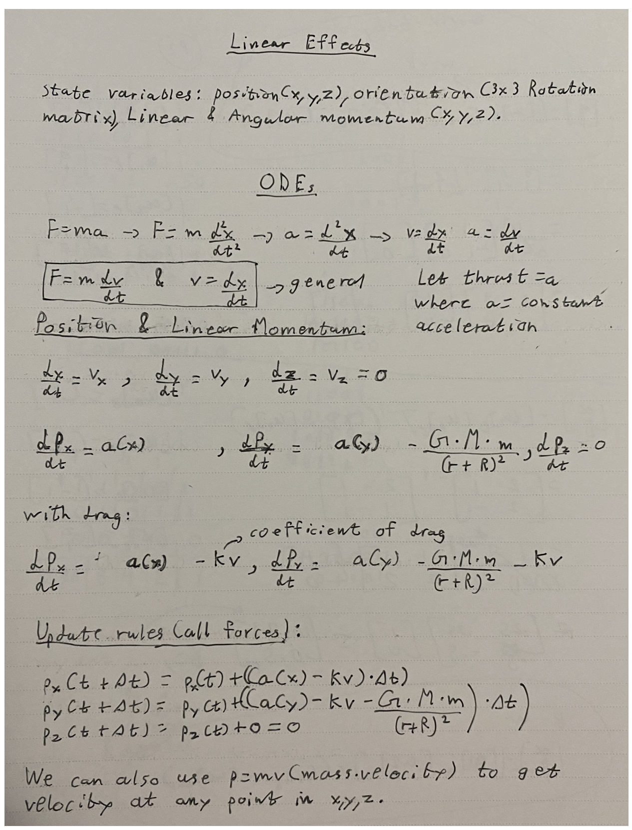
3 Data/Variables

Although there is no need for data collection for this specific simulation, we can approximate some values based on real-world situations such as rocket specifications, planet mass, radius, and more. Additionally, based on the forces from modelling the simulation, we can extract the variables needed to implement and validate the rocket simulation.

Inputs (variables the user selects before the start of the simulation): rocket mass (5000 for good results), thrust force (500 for best results), gravitational constant ($-6.67430 \cdot 10^{-11}$), Earth's mass ($5.972 \cdot 10^{24}$), and Earth's radius ($6.3781 \cdot 10^{6}$).

State variables for the rigid body (rocket): position, orientation, linear momentum, and angular momentum. Although the code works in 3D, the visuals only move the rocket in 2D, therefore, the z direction is ignored. We can now examine the differential equations related to each of these state variables.

4 Implementation and Validation



Linear Effects

State variables: position (x, y, z), orientation (3x3 Rotation matrix), Linear & Angular momentum (x, y, z).

ODEs

$F = ma \rightarrow F = m\frac{d^2x}{dt^2} \rightarrow a = \frac{d^2x}{dt} \rightarrow v = \frac{dx}{dt} \quad a = \frac{dv}{dt}$

$\boxed{F = m\frac{dv}{dt} \quad \& \quad v = \frac{dx}{dt}} \rightarrow$ general

Let thrust = a where a = constant acceleration

Position & Linear Momentum:

$\frac{dx}{dt} = v_x, \quad \frac{dy}{dt} = v_y, \quad \frac{dz}{dt} = v_z = 0$

$\frac{dP_x}{dt} = a(x), \quad \frac{dP_y}{dt} = a(y) - \frac{G \cdot M \cdot m}{(r+R)^2}, \quad \frac{dP_z}{dt} = 0$

with drag:

$\frac{dP_x}{dt} = a(x) - \overset{\text{coefficient of drag}}{k}v, \quad \frac{dP_y}{dt} = a(y) - \frac{G \cdot M \cdot m}{(r+R)^2} - kv$

Update rules (all forces):

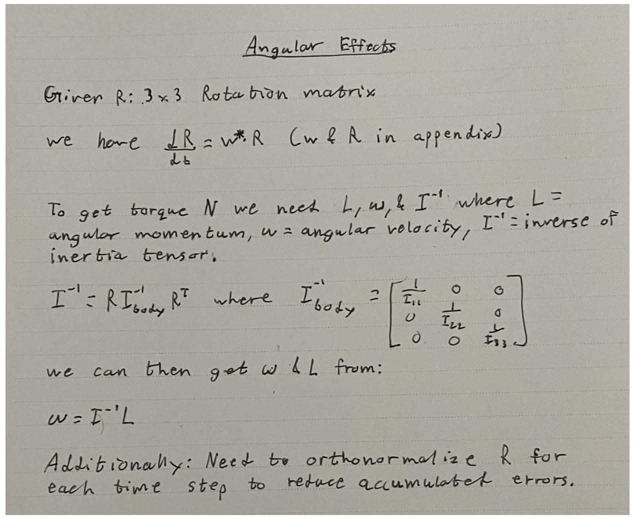
$p_x(t + \Delta t) = p_x(t) + ((a(x) - kv) \cdot \Delta t)$

$p_y(t + \Delta t) = p_y(t) + \left(\left(a(y) - kv - \frac{G \cdot M \cdot m}{(r+R)^2}\right) \cdot \Delta t\right)$

$p_z(t + \Delta t) = p_z(t) + 0 = 0$

We can also use p=mv (mass·velocity) to get velocity at any point in x, y, z.

Figure 2: Linear effects in Rigid Body Dynamics



Angular Effects

Given R: 3×3 Rotation matrix

we have $\frac{dR}{dt} = w^{*} R$ (w & R in appendix)

To get torque N we need $L, \omega, \& I^{-1}$ where L = angular momentum, ω = angular velocity, I^{-1} = inverse of inertia tensor.

$$I^{-1} = R I_{body}^{-1} R^T \quad \text{where} \quad I_{body}^{-1} = \begin{bmatrix} \frac{1}{I_{11}} & 0 & 0 \\ 0 & \frac{1}{I_{22}} & 0 \\ 0 & 0 & \frac{1}{I_{33}} \end{bmatrix}$$

we can then get ω & L from:

$$\omega = I^{-1} L$$

Additionally: Need to orthonormalize R for each time step to reduce accumulated errors.

Figure 3: Angular effects in Rigid Body Dynamics

For more information, see 11a and 11b

5 Execution

To execute the simulation, we made the rocket simulation to be both interactive and scientifically accurate as much as possible. By implementing intuitive controls mapped to the UP, LEFT, and RIGHT keys, users could actively pilot the rocket, experiencing firsthand the dynamics of space travel.

We tested scenarios that simulated real-world challenges, such as navigating Earth's gravitational pull, drag and more. These scenarios provided a realistic experience and valuable insights into the practical and theoretical concepts for rocket simulations.

Precision was important in ensuring the accuracy of our simulation. We conducted rigorous parameter adjustments, fine-tuning critical variables like mass, acceleration, momentum, and various forces to align with established physics principles. This meticulous calibration ensured that the simulation yielded results consistent with the theoretical expected values.

To enhance engagement, we incorporated dynamic visual elements. Graphical representations, including velocity and distance, were integrated into the interface. not only helped visualize complex concepts but also fostered a deeper understanding of rocket dynamics.

In summary, our execution strategy encompassed a comprehensive approach that combined interactive controls, scenario-based exploration, precise parameter tuning, and visual feedback. This approach not only supported our experiential learning but also highlighted our commitment to delivering a comprehensive simulation.

6 Results

The implementation of the code resulted in an acceptable simulation of a rocket launching from Earth while being affected with all the forces accounted for. The program was able to simulate a rocket with rigid body parameters along with user control.

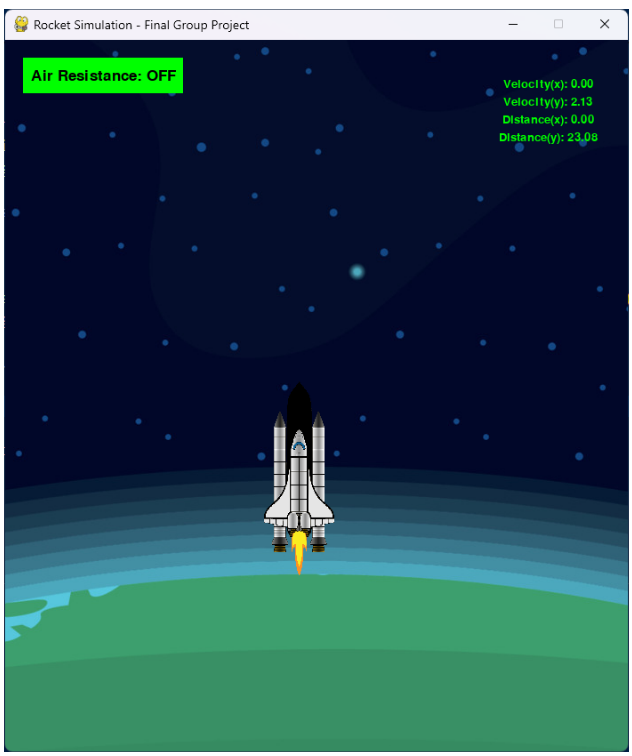


Figure 4: Rocket with thrust in only y-dir and no air resistance

Results from the simulation:

Figure 4 represents a rocket moving up (+ve velocity) due to thrust in the y-direction with no air resistance. Since there is no air resistance and the gravity force will continue to decrease with y distance, it is observed that the rocket accelerates. Figure 5 indicates

Figure 5: Rocket with thrust in only y-dir and air resistance

that when air resistance (drag) is enabled, the velocity of the rocket in the y direction is significantly slower compared to the no drag scenario while still being around the same distance as Figure 4.

Now the more complex scenarios: Figure 6 shows just how complex the forces get when working in multiple dimensions. When thrust is applied in more than one axis, the forces are split into their x, y and z components (since this is a 2D simulation we only show x and y, however, z is still being calculated at each step).

Another interesting observation was that with air resistance, the velocity in the y direction eventually reaches terminal velocity, which matches mathematical concepts since drag is proportional to velocity. To confirm, Figure 7 shows how the rocket's velocity and distance change over time (no air resistance). These figures are mathematically accurate as ini-

Figure 6: Forces in x and y axis

tially velocity linearly increases due to thrust being applied, then linearly decreases due to gravity pulling the rocket back down. Also, the distance is a nice parabola due to no drag or random forces (atmospheric changes or others).

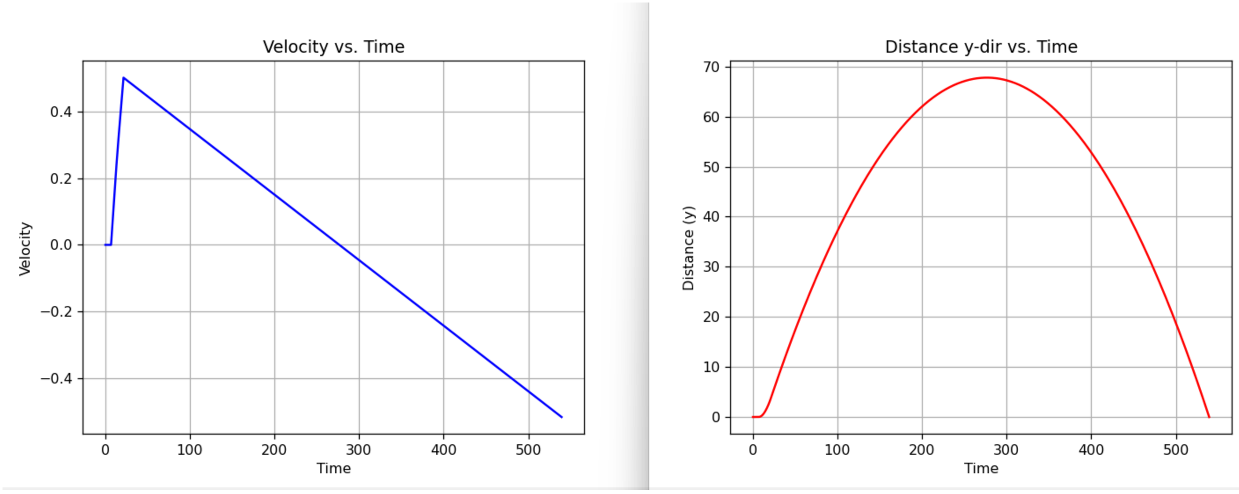


Figure 7: Graphs of Velocity and Distance in y direction

7 Conclusions/Limitations

In conclusion, we explored rocket simulation to understand rigid body dynamics more efficiently. Our goal was to make a simple yet detailed simulation that captures how rockets move. We started by figuring out what we needed for the simulation to work accurately which led us to use mathematical equations we learned throughout the course to create a model that demonstrates the rigid body dynamics of rockets.

Ultimately, we made a simulation where rockets launch from Earth and react to gravity and thrust. We could also control the rocket to see how it moved. However, we had to keep things simple by ignoring some details like fuel use and complex air effects. Going outside the boundaries also presented a significant challenge since space is vast and it was difficult to put into scale the enormity of the display map.

Even though our simulation isn't perfect, it's a good start. It helps us learn about rockets and space travel in the scientific aspect and helps solidify our understanding of modelling and running simulations.

Overall, this project taught us a lot about how simulations can be so versatile when wanting to recreate a general or specific scenario. In the future, this simulation can be made much more realistic and informative with the help of more advanced calculations.

References

OpenAI. (2024). ChatGPT (3.5) [Large language model]. ChatGPT
background Image Link Rocket Image Link
Explosion Image Link Thrust Sound Link
Explosion Sound Link

A Appendix

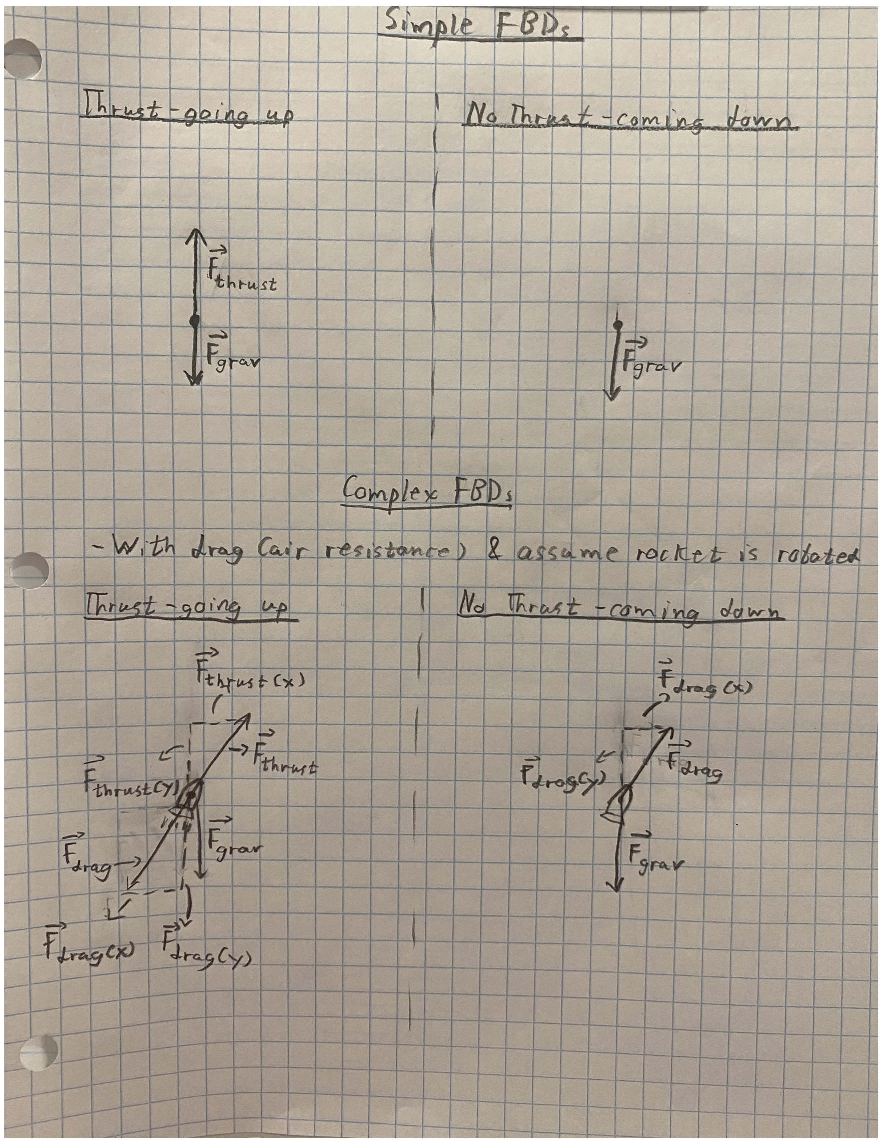


Figure 8: Free Body Diagrams

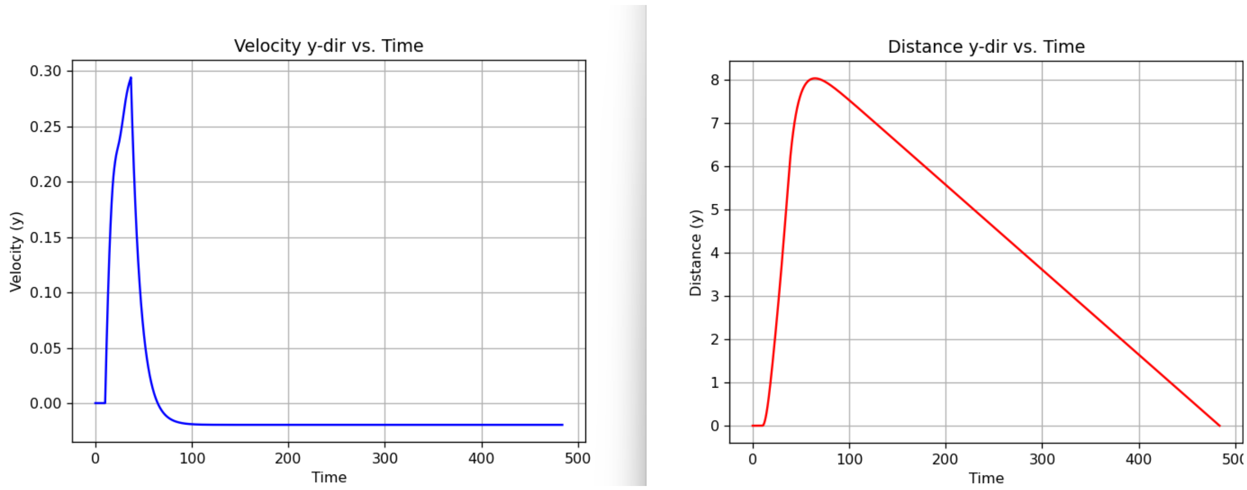


Figure 9: Graph with Drag enabled

```
Velocity at t = 1: 0.009803600531485547
```

$$f(F, m) = \frac{F}{m}$$

$$v(a, t) = at$$

$$f(500, 5000) = 0.1$$

$$v(0.1, 1) = 0.1$$

(a) Velocity Proof

$$\frac{\frac{\left(-6.67430(10)^{-11} \cdot 5.972(10)^{24} \cdot 5000\right)}{\left(100000 + 6.371(10)^{6}\right)^{2}}}{5000} = -9.51881142215$$

$$\frac{\frac{\left(-6.67430(10)^{-11} \cdot 5.972(10)^{24} \cdot 5000\right)}{\left(0 + 6.371(10)^{6}\right)^{2}}}{5000} = -9.81997342622$$

```
Gravity Force at 0.0: -9.819973426224685
Gravity Force at 100027.61: -9.518730200538103
```

(b) Gravity Proof

Proof of accurate simulation:

These calculations show that our simulation and the theoretical values are very close, verifying the accuracy of the simulation

1. **Mass** (m): Kilograms (kg)
2. **Gravitational Constant** (G): $m^3 \cdot kg^{-1} \cdot s^{-2}$
3. **Earth Mass** (M): Kilograms (kg)
4. **Earth Radius** (R): Meters (m)
5. **Distance** (r): Meters (m)
6. **Force** (F): Newtons (N)
7. **Velocity** (v): Meters per second (m/s)
8. **Torque**: Newton-meters (Nm)

(a) Units (OpenAI, 2024)

State variables

Position	$\mathbf{x}$	1 by 3 vector
Orientation	R	3 by 3 rotation matrix
Linear Momentum	$\mathbf{P}$	1 by 3 vector
Angular Momentum	$\mathbf{L}$	1 by 3 vector

Constants

Mass	m	scalar
Inertia tensor	I_{body}	3 by 3 matrix (in body frame)

Derived quantities

Linear velocity	v	1 by 3 vector
Angular velocity	ω	1 by 3 vector
Inertia tensor	I^{-1}	3 by 3 matrix (in world frame)

Total force	$\mathbf{F}$	1 by 3 vector
Total torque	$\mathbf{T}$	1 by 3 vector

$$\omega^* = \begin{bmatrix} 0 & -\omega_x & \omega_y \\ \omega_z & 0 & -\omega_x \\ \omega_y & \omega_z & 0 \end{bmatrix}$$

(b) Additional information (Faisal Qureshi, 2024)