TEKNOFEST 2022

ROCKET COMPETITION

MEDIUM ALTITUDE CATEGORY

CDR FLIGHT SIMULATION REPORT (FSR)

Altinbas University

EVA X

TABLE OF CONTENTS

Kinematic and Dynamic Equations

Total force acting on rocket is equal to mass times acceleration

$$\sum \vec{F} = m\vec{a} = \frac{d\left(m\vec{V}\right)}{dt}$$

Forces acting on the rocket are gravity, thrust and drag forces (lift force is neglected).

$\vec{F}_{gr}$: Gravity force is caused by gravitational acceleration, direction is always to the centre of Earth

$\vec{F}_{imp}$: Thrust force is generated by rocket engine, same direction with rocket motion (no angle of attack)

$\vec{F}_{drag}$: Drag force is caused by motion of the rocket in air, inverse direction of motion

$$-\dot{m}_f = \frac{dm}{dt} \tag{1}$$

$$\sum \vec{F} = \vec{F}_{gr} + \vec{F}_{imp} + \vec{F}_{drag} = m\vec{g} + \dot{m}_f\vec{V}_e + \frac{1}{2}\rho\vec{V}^2 AC_D = \frac{d\left(m\vec{V}\right)}{dt}$$
$$= \begin{bmatrix} m\vec{a}_x \\ m\vec{a}_y \\ m\vec{a}_z \end{bmatrix} = \begin{bmatrix} \vec{F}_{gr,x} + \vec{F}_{imp,x} + \vec{F}_{drag,x} \\ \vec{F}_{gr,y} + \vec{F}_{imp,y} + \vec{F}_{drag,y} \\ \vec{F}_{gr,z} + \vec{F}_{imp,z} + \vec{F}_{drag,z} \end{bmatrix} \tag{2}$$

m : Mass of rocket (kg)

$\vec{g}$: Gravitational acceleration (m/s^2)

$\dot{m}_f$: Mass flow rate of fuel of rocket (kg/s)

$\vec{V}_e$: Exit velocity of fuel gases (m/s)

ρ : Density of air (kg/m^3)

$\vec{V}$: Total velocity of the rocket (m/s)

A : Cross-sectional area of the rocket (m^2)

C_D : Drag coefficient

Thrust data are provided by rocket engine manufacturer. In this report, $\vec{F}_{imp}$ data are already known thus thrust force will be indicated implicitly ($\vec{F}_{imp}$). Equation (1) and (2) comprises a partial non-linear differential equation system. A numerical method can be used to solve such system. In this report, Euler method has been used for solution.

If equation (1) and (2) discretized using the Euler method

$$-\dot{m}_f = \frac{m_{i+1} - m_i}{\Delta t} \Longrightarrow m_{i+1} = m_i - \dot{m}_f \Delta t \tag{3}$$

$$\begin{bmatrix} \vec{a}_{x,i+1} \\ \vec{a}_{y,i+1} \\ \vec{a}_{z,i+1} \end{bmatrix} = \begin{bmatrix} \frac{1}{m_{i+1}} & 0 & 0 \\ 0 & \frac{1}{m_{i+1}} & 0 \\ 0 & 0 & \frac{1}{m_{i+1}} \end{bmatrix} \begin{bmatrix} 0 + \vec{F}_{imp,i} \cdot cos\alpha_i + \vec{F}_{drag,i} \cdot cos\alpha_i \\ 0 + 0 + 0 \\ mg + \vec{F}_{impi} \cdot sin\alpha_i + \vec{F}_{drag,i} \cdot sin\alpha_i \end{bmatrix} \tag{4}$$

i : Time step

Δt : Time step interval (s)

α : Flight angle $(radian)$

Acceleration values at $(i + 1)$th time step can be obtained by solving the algebraic equations (3) and (4). To find velocity and position values at $(i + 1)$th time step, equations (5) and (6) can be used.

$$\begin{bmatrix} \vec{v}_{x,i+1} \\ \vec{v}_{y,i+1} \\ \vec{v}_{z,i+1} \end{bmatrix} = \begin{bmatrix} \vec{a}_{x,i+1} \\ \vec{a}_{y,i+1} \\ \vec{a}_{z,i+1} \end{bmatrix} \Delta t + \begin{bmatrix} v_i \\ v_i \\ v_i \end{bmatrix} \tag{5}$$

$$\begin{bmatrix} \vec{x}_{i+1} \\ \vec{y}_{i+1} \\ \vec{z}_{i+1} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} \vec{a}_{x,i+1} \\ \vec{a}_{y,i+1} \\ \vec{a}_{z,i+1} \end{bmatrix} \Delta t^2 + \begin{bmatrix} \vec{v}_{x,i+1} \\ \vec{v}_{y,i+1} \\ \vec{v}_{z,i+1} \end{bmatrix} \Delta t + \begin{bmatrix} \vec{x}_i \\ \vec{y}_i \\ \vec{z}_i \end{bmatrix} \tag{6}$$

In here $\vec{v}$, indicates velocity vectors and $\vec{x}$, $\vec{y}$, $\vec{z}$ indicate position vectors accordingly.

Atmosphere Model

The atmosphere model to be used in the application has been created with reference to the "U.S. Standard Atmosphere Model, 1976" **[2]**. Constants to be used for the model can be found in Table 1.

Table 1: Constants table

Symbol	Value	Explanation
M	$28.97\ kg/k \cdot mol$	Molar mass of air
R^*	$8.31432\ J/K \cdot mol$	Gas constant

Air has been accepted as an ideal gas in the calculations, so the ideal gas equation will be used.

$$P = \frac{\rho R^* T}{M} \tag{7}$$

The lower layers of the atmosphere are in hydrostatic equilibrium. Therefore, the pressure change is related to the height change only.

$$dP = -g\rho dz \tag{8}$$

If equations (7) and (8) are combined:

$$\frac{dP}{P} = -\frac{gM}{R^* T} dz \tag{9}$$

Solving equation (9)

$$\int_{P_0}^{P} \frac{dP'}{P'} = -\int_{0}^{z} \frac{gM}{R^*(T_0 - L \cdot z')} dz' \tag{10}$$

$$P = P_0 \left(1 - \frac{Lz}{T_0}\right)^{\frac{gM}{RL}} \tag{11}$$

In addition, the atmospheric temperature varies linearly between 0-10 km.

$$T = T_0 - Lz \tag{12}$$

Using equations (7), (11) and (12), density function can be obtained.

$$\rho = P_0 \cdot M \cdot \frac{\left(\frac{1 - Lz)}{T_0}\right)^{\frac{gM}{RL}}}{R \cdot (T_0 - Lz)} \tag{13}$$

ρ	: Air density (kg/m^3)
P_0	: Sea level standard atmospheric pressure (Pa)
T_0	: Sea level standard temperature (K)
g	: Earth-surface gravitational acceleration (m/s^2)
L	: Temperature lapse rate 0.0065 K/m
R	: Ideal (universal) gas constant 8.31446 J/(mol·K)
M	: Molar mass of dry air 0.0289652 kg/mol
z	: Altitude (m)

Engine Model

- Time-dependent thrust force model:

A time dependent thrust force model was created by interpolating the thrust data received over Openrocket. Interpolation was performed using linear interpolation on matlab.

- Time dependent mass consumed (fuel spent mass) model

The formula $I_{sp} = \frac{T}{\dot{m}_f g}$ was used to calculate the mass thrown out over time.

(I_{sp}= Specific impulse, T= Thrust force, $\dot{m}_f$ =Mass flow rate of fuel, g= Gravitational acceleration)

The thrust and time values obtained from Openrocket. They are firstly interpolated and then processed with the specific impulse value of our rocket's engine, and the mass flow rate per unit time is calculated.

The code used to calculate the time-dependent consumed mass:

```
function [mass,itki_1] = rocket_mass_thrust()
itki = [our trust force values];
time = [our time values];
time_1 = 0:0.01:4.27;
itki_1 = interp1(time,itki,time_1);
Isp  = 197.6;
g    = 9.801;
m_debi= itki_1/(Isp*g);
mass = 26.813;

for i = 1:length(time_1)-1
    mass(i+1) = mass(i) - (m_debi(i+1)+m_debi(i)).*(time_1(i+1)-time_1(i))/2;
end
```

- Thrust force – Time graph

  The thrust force with respect to time graph is given in Figure 1.

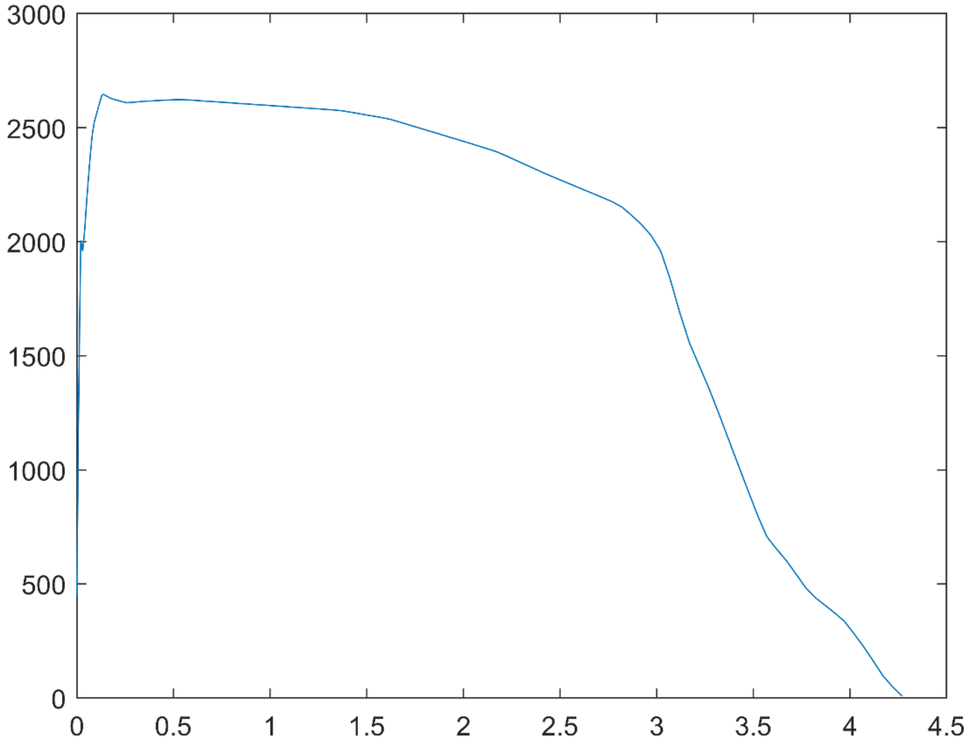

Figure 1. Thrust force-time graph

- Thrown mass – Time graph

  The thrown mass with respect to time graph is given in Figure 2.

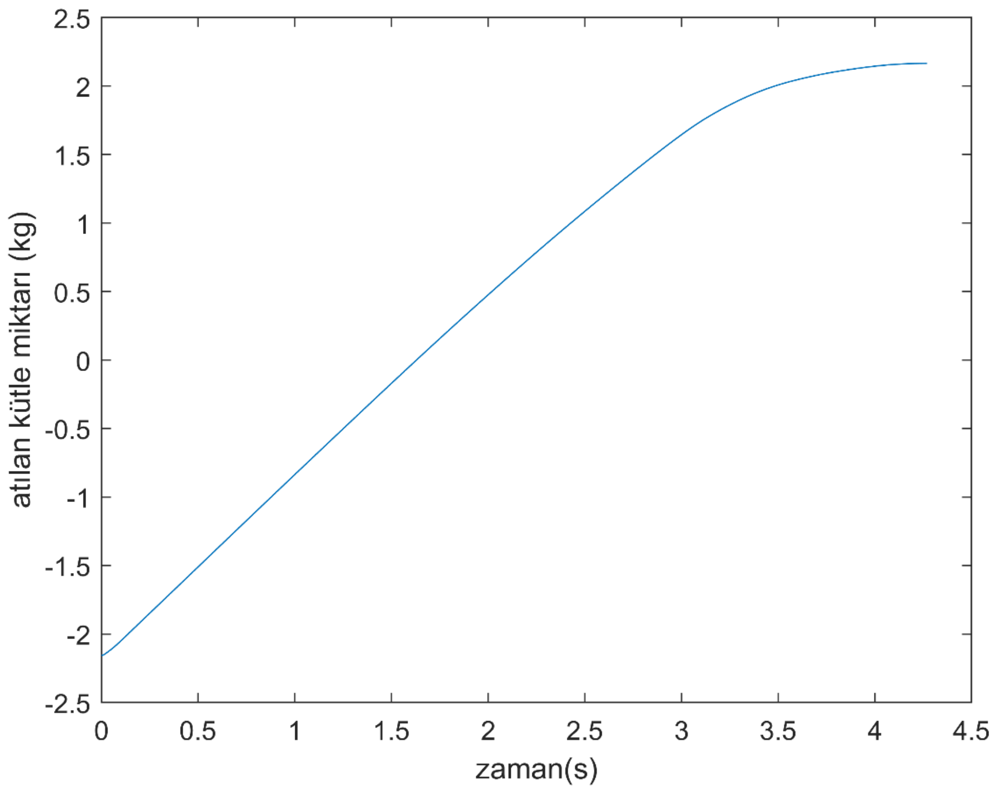

Figure 2. Thrown mass-time graph

Aerodynamic Model

Drag coefficient data are obtained from ANSYS analysis regard to different Mach numbers and then a curve is fitted.

- Mach-C_d Graph:

  Mach number with respect to Cd value graph is given in Figure 3.

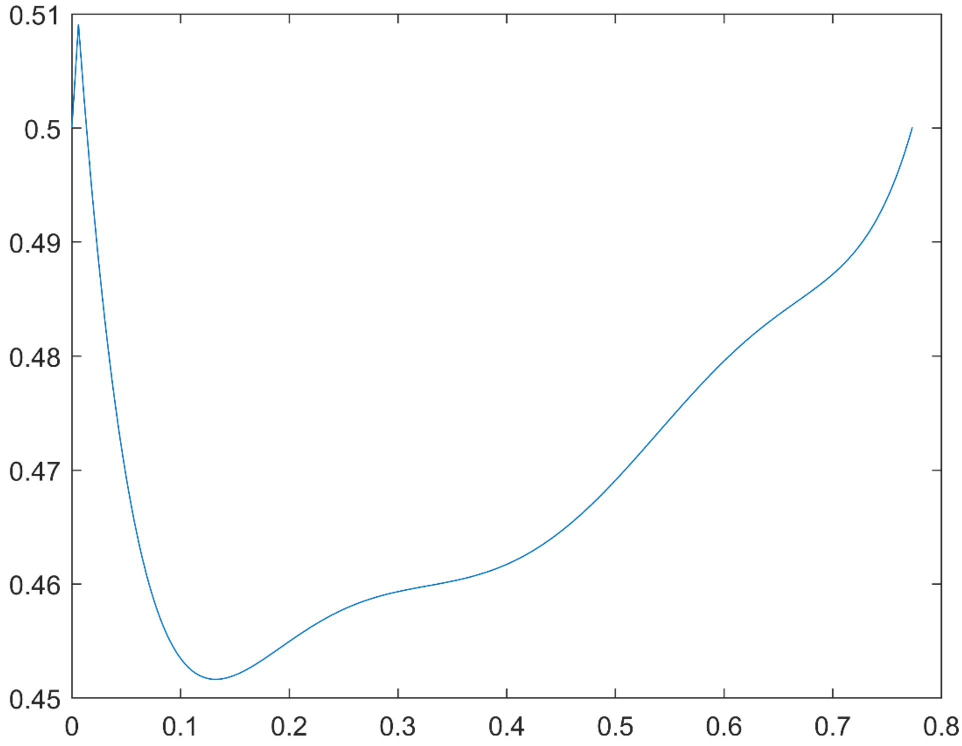

Figure 3. Mach number- C_d graph

Simulation Structure

Flight simulation was performed using Matlab programming language. Euler Numerical Integration method was used as the solution method. Initially, sea level atmosphere air information, gravitational acceleration (9.801 m/s^2), time interval (0.01 s), initial flight angle (85 degrees), projected area of rocket, and specific impulse value are given as input to the simulation code. After that for each step aerodynamic drag coefficient (Cd), thrust value are given to calculate the forces.

The flight simulation code is divided into 2 main blocks. These are the operations from the starting position until the rocket runs out of fuel and from rocket runs out of fuel until the peak.

First of all, the mass of the rocket was calculated according to the specific impulse, thrust value and gravitational acceleration using the Euler method. Then, acceleration, velocity, position, density, resultant velocity, drag force and flight path angle were calculated (These calculations were made according to the formulas written in the Kinematic and Dynamic Equations section).

The same calculations are performed from the beginning to the peak with the updated values thanks to the Euler method.

Our simulation codes prepared on Matlab:

```
clear
clc

k = 1.4; % Specific heat ratio of air

% Initial Conditions
g      = 9.801;              % Gravitational acceleration (m/s^2)
dt     = 1e-2;               % Delta_t (s)
alpha  = 85*pi/180;          % Launch angle (degree), from x direction -z direction
pos    = [0;0;0];            % Initial position vectors (m)
vel    = [0.1743;0;1.9924];  % Initial velocity vectors (m/s)
velo   = sqrt(vel(1,1)^2+vel(2,1)^2+vel(3,1)^2);
acc    = [0;0;0];            % Initial acceleration vectors (m/s^2)
A      = 0.013273;           % Cross-sectional area of the rocket(m^2)
rho    = zeros(1);           % Density
Mach   = zeros(1);           % Mach number
dyn_pres= zeros(1);          % Dynamic pressure
C_D    = 0.5*ones(1);

mass_thrust_data = 'rocket_mass_thrust_tekno';
switch mass_thrust_data
   case {'rocket_mass_thrust_evax'} , [mass,thrust]=rocket_mass_thrust_evax(); %C_D =
35.282*Mach(n+1)^6-91.301*Mach(n+1)^5+92.48*Mach(n+1)^4-
46.293*Mach(n+1)^3+11.955*Mach(n+1)^2-1.4575*Mach(n+1)+0.5176;
   case {'rocket_mass_thrust_tekno'} , [mass,thrust]=rocket_mass_thrust_tekno(); %C_D =
1.0663*Mach(n+1)^4-2.45*Mach(n+1)^3+2.1104*Mach(n+1)^2-0.89*Mach(n+1)+0.5195;
end

%Flight variables
fd = zeros(1,1);          % Drag force (N)
n  = 1;                   % Started from 4 since rocket starts movement at t = 0.04 s
```

```
%% Until Motor BurnOut
% From launchrod leaving until burnout
while n*dt<4.27
    m = mass(n);
    fi= thrust(n);

[Mass,Force]            = mass_force(m,fd(n),fi,alpha(n));

acc(:,n)                = sum((Mass*Force),2);
vel(:,n+1)              = vel(:,n) + acc(:,n)*dt;
velo(n+1)               = sqrt(vel(1,n+1)^2+vel(2,n+1)^2+vel(3,n+1)^2);
pos(:,n+1)              = pos(:,n) + vel(:,n+1)*dt + 0.5*acc(:,n)*dt^2;

[rho(n+1),Mach(n+1)]    = density_mach(velo(n+1),pos(3,n+1));
C_D(n+1)                = 1.0663*Mach(n+1)^4-2.45*Mach(n+1)^3+2.1104*Mach(n+1)^2-
0.89*Mach(n+1)+0.5195;
fd(n+1)                 = 0.5*C_D(n+1)*rho(n+1)*velo(n+1)^2*A;
dyn_pres(n+1)           = 0.5*rho(n+1)*velo(n+1)^2;
alpha(n+1)              = atan(vel(3,n)/vel(1,n)); % New flight angle (radian)

n = n+1;
end

%% After Motor Burnout
% From burnout to apogee
while pos(3,n)>pos(3,n-1)

m = 24.622; %Constant mass (kg)

[Mass,Force]            = mass_force(m,fd(n),fi,alpha(n));

acc(:,n)                = sum((Mass*Force),2);
vel(:,n+1)              = vel(:,n) + acc(:,n)*dt;
velo(n+1)               = sqrt(vel(1,n+1)^2+vel(2,n+1)^2+vel(3,n+1)^2);
```

```
pos(:,n+1)              = pos(:,n) + vel(:,n+1)*dt + 0.5*acc(:,n)*dt^2;

[rho(n+1),Mach(n+1)]    = density_mach(velo(n+1),pos(3,n+1));
C_D(n+1)                = 1.0663*Mach(n+1)^4-2.45*Mach(n+1)^3+2.1104*Mach(n+1)^2-
0.89*Mach(n+1)+0.5195;
fd(n+1)                 = 0.5*C_D(n+1)*rho(n+1)*velo(n+1)^2*A;
dyn_pres(n+1)           = 0.5*rho(n+1)*velo(n+1)^2;
alpha(n+1)              = atan(vel(3,n)/vel(1,n)); % New flight angle (radian)

n = n+1;
end

%% Results
time = 0:dt:(n-1)*dt;

figure(1)
plot(time(1:428),thrust)
xlabel 'zaman(s)'
ylabel 'itki (N)'
figure(2)
plot(time(1:428),26.813-mass)
xlabel 'zaman(s)'
ylabel 'atılan kütle miktarı (kg)'
figure(3)
plot(pos(1,:),pos(3,:))
xlabel 'yatay mesafe (m)'
ylabel 'dikey mesafe (m)'
figure(4)
plot(time,alpha*180/pi)
xlabel 'zaman (s)'
ylabel 'uçuş yolu açısı (\circ)'
figure(5)
plot(time,Mach)
xlabel 'zaman(s)'
```

```
ylabel 'Mach sayısı'
figure(6)
plot(time,dyn_pres)
xlabel 'zaman (s)'
ylabel 'dinamik basınç (kg/ms^2)'
figure(7)
plot(time,-vel(3,:))
xlabel 'zaman (s)'
ylabel 'dikey hız (m/s)'
```

Validation of Simulation

Output Table:

The outputs of our simulation with respect to values given by Teknofest are given in Table 1.

Table 2 Outputs of Teknofest's values

	Value
Max Mach Number [-]	1.0909
Peak Position [m]	[1231.6, 0, 5122.1]
Peak Velocity (resultant) [m/s]	35.70 m/s
Peak Mach Number [-]	0.1115
Peak Time [s]	31.87 s

- Trajectory graph (altitude-range):

The trajectory graph of the simulation given in Figure 4.

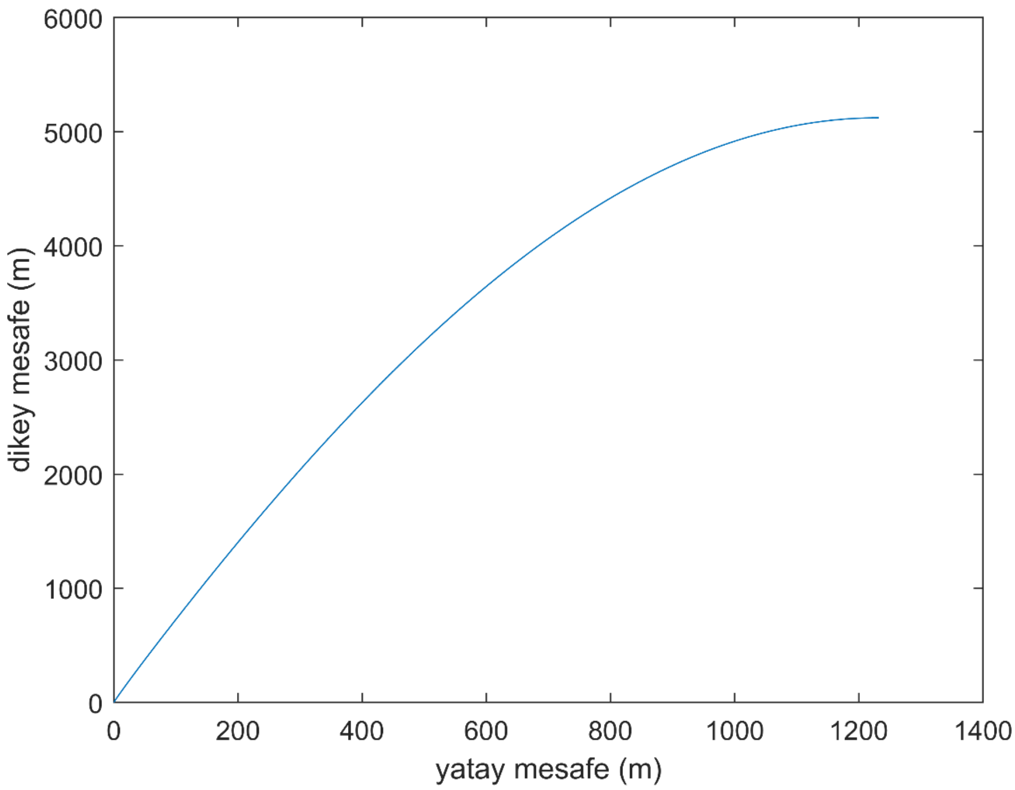

Figure 4. Trajectory graph

Simulation Results

Output and comparison table:

The outputs of our simulation and its comparison with Openrocket's values for out rocket are given in Table 2.

Table 3 Simulation output and their comparison with Openrocket's values

	OpenRocket Value (a)	**Simulation Value (b)**	**Percentage Difference (b-a)/a*100**
Max Mach Number [-]	0.8353	0.8394	0.49
Peak Position [m]	[625,0,4329]	[1013.3,0,4576]	[62,0,5.7]
Peak Velocity (resultant) [m/s]	280	280.52	0.17
Peak Mach Number [-]	0.075	0.1248	66.4
Peak Time [s]	26.2	28.05	7.06

<u>Possible reasons for the percentage differences:</u> Performing our simulation in Order 2 causes us to ignore many forces on our rocket. Therefore, since the stability of our rocket changes, there are differences according to the openrocket values. In addition, the fact that we do not add forces such as wind effect to our simulation creates serious differences in the trajectory of our rocket.

- Trajectory graph (altitude-range)
  The trajectory graph of the simulation is given in Figure 5.

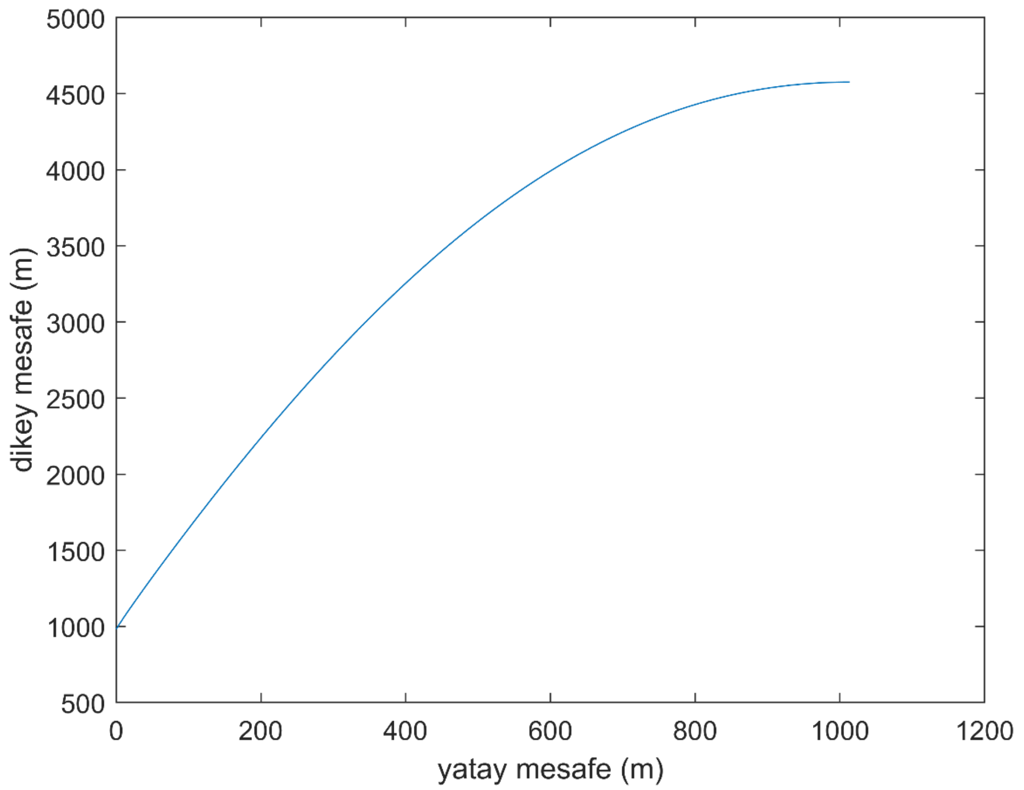

Figure 5. Trajectory graph

- Flight path angle-time graph

  The flight path angle with respect to time graph is given in Figure 6.

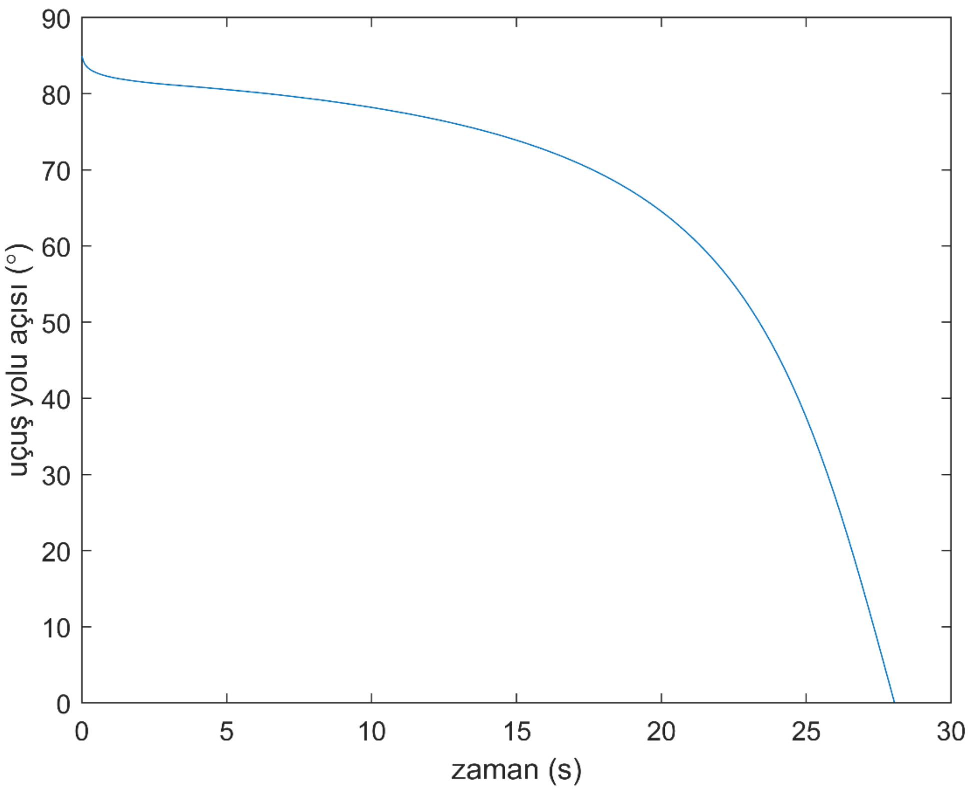

Figure 6. Flight path angle-time graph

- Mach – Time graph

  The mach number with respect to time graph is given in Figure 7.

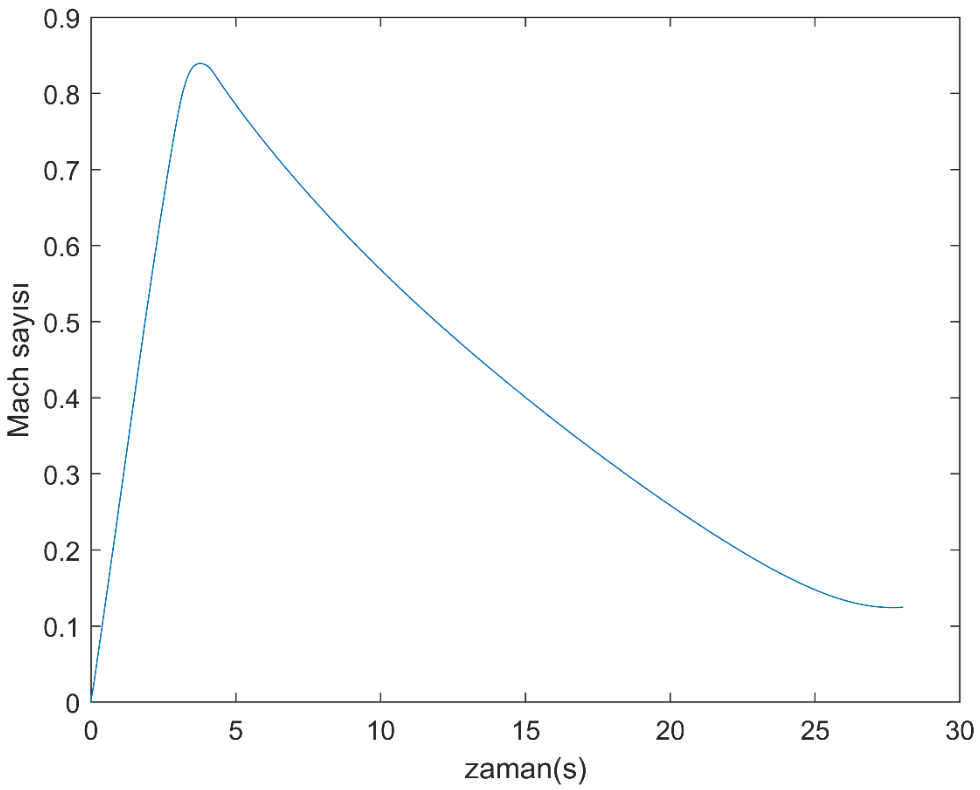

Figure 7. Mach number-time graph

- Maximum mach height:

  0.84

- Dynamic pressure – Time graph

  The dynamic pressure with respect to time graph is given in Figure 8.

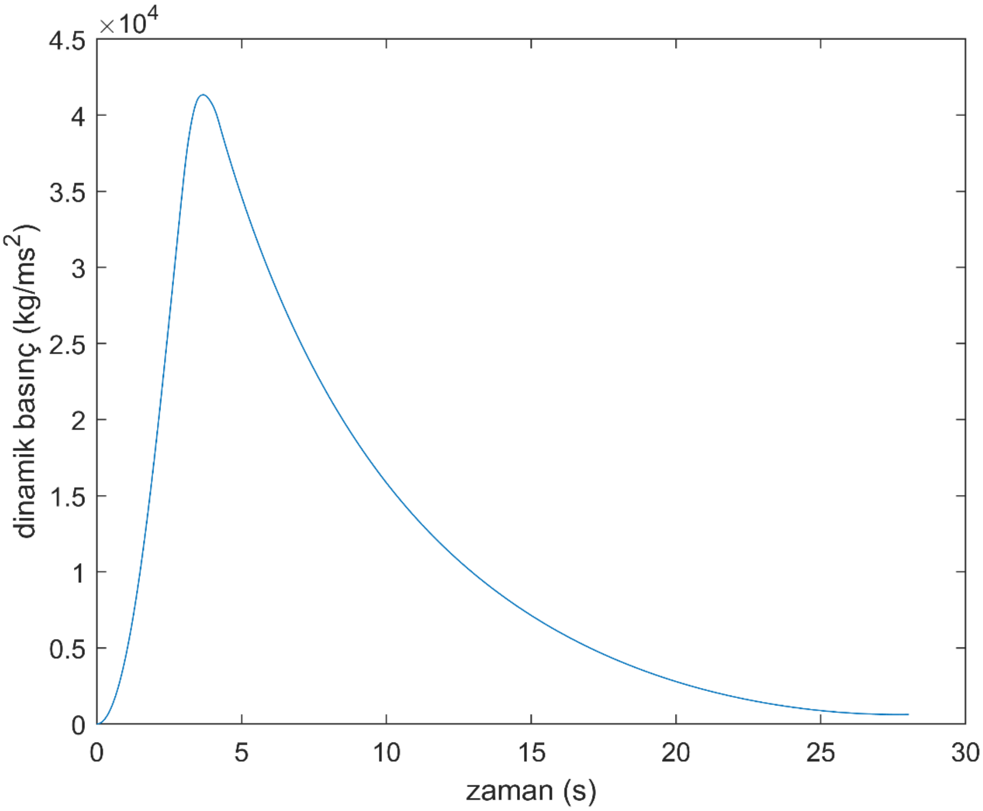

Figure 8. Dynamic pressure -time graph

- Maximum dynamic pressure height:

41446.34 kg/m^2

- Importance of maximum dynamic pressure for rocket:

The point where the speed of the rocket is maximum is the point where the dynamic pressure is maximum. It is important for the structural design of the vehicle because aerodynamic pressure and load are related.

- Vertical climb velocity (-Z axis velocity) – Time graph

  The with respect to time graph is given in Figure 9.

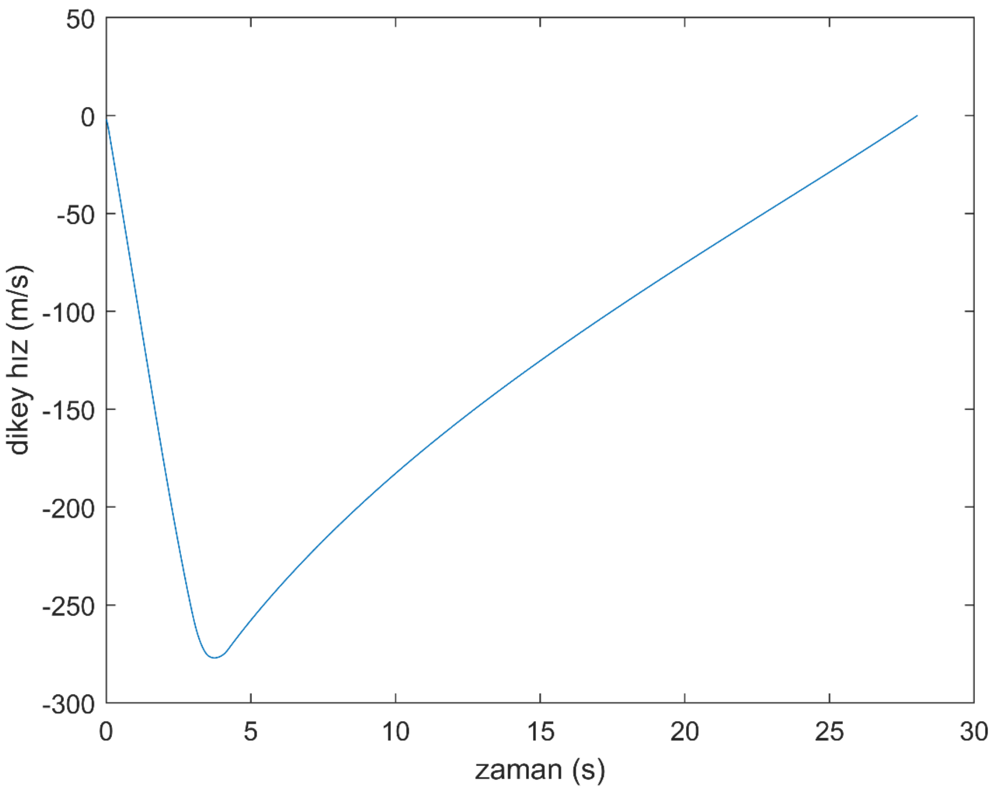

Figure 9. -Z axis velocity -time graph

References:

[1] Wikipedia contributors. (2022, April 29). Density of air. Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=Density_of_air&oldid=1085300509

[2] (N.d.). Noaa.Gov. Retrieved May 4, 2022, from https://www.ngdc.noaa.gov/stp/space-weather/online-publications/miscellaneous/us-standard-atmosphere-1976/us-standard-atmosphere_st76-1562_noaa.pdf