Table of Contents

Initial conditions, measurements, variables, etc.

```
%units (m,kg)
clc
clear all
close all
g_0 = 9.8;
m_nosecone = 4.535;
d_rocket_nosecone = .10551;
l_nosecone = .5588;
m_payload = 11.34;
d_rocket_payload = 0.152;
l_payload = 1.021;
m_engine = 27.2;
d_rocket_engine = d_rocket_payload;
l_engine =  1.374;
a_fin = .088; %one fin

%Diameter of the parachute
a_chute_nc = (0.9144.^2)/4 * pi;%36 inches
a_chute_pl = (1.2192.^2)/4 * pi;%48 inches
a_chute_eng =(2.1336.^2)/4 * pi;%84 inches

%coefficient of drag for the parachutes
%Parachute drag based on circular parachute, from Knacke 5-25
cd_parachute = 2.2;
cd_cylinder = 1;

%Calculates the areas of each part of the rocket
a_rocket_nosecone = l_nosecone * d_rocket_nosecone;
a_rocket_payload = l_nosecone * d_rocket_nosecone;
a_rocket_engine = l_nosecone * d_rocket_nosecone + 2 * a_fin;
a_rocket = a_rocket_nosecone + a_rocket_payload + a_rocket_engine;
```

Density profile of the atmosphere (0-13000m)

```
T_0 = 288.16; %sea level temperature (kelvin)
rho_0 = 1.225; %sea level density (kg/m^3)
a = -0.0065; %lapse rate (K/m)
R = 287.05; %gas constant (J/kg*K)
den = [];
```

```
for h = 1:11000
    T = T_0 + a*h;
    den(h) = rho_0 * (T/T_0)^((-g_0/(a*R)) - 1 );
end
T_isot = 217; %isothermal temp (K)
for h = 11001:13000
    den(h) = den(11000) * (exp(1))^((-g_0/(T_isot*R))* (h-11000));
end
```

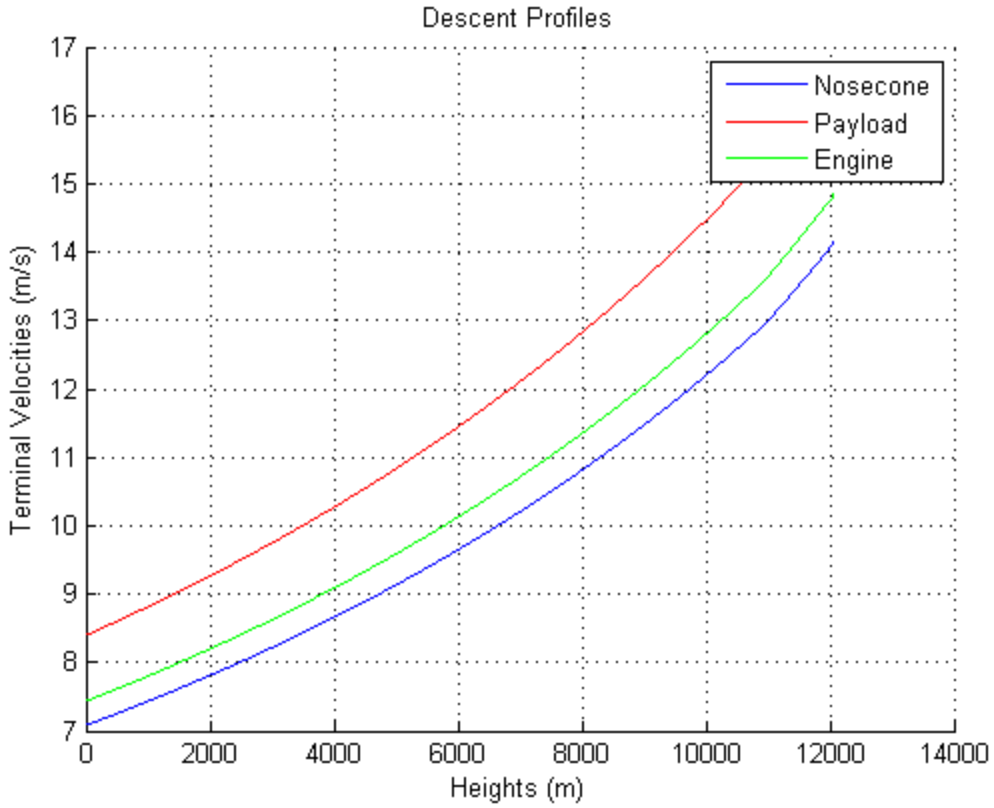
Calculations of the Decent Profile

```
dt = .1;
n_ns = 0;
v_i = 0;
h_i = 12192; %(40000 ft)
v_curr = v_i;
h_curr = h_i;
heights_nc = [];
velocities_nc = [];

while (h_curr > 10)

    heights_nc = [heights_nc h_curr];
    velocities_nc = [velocities_nc v_curr];

    F_g = m_nosecone * g_0;
    F_drag = .5 * den(round(h_curr)) * v_curr^2 * a_chute_nc * cd_parachute;
    F_curr = F_drag - F_g;
    dv = (F_curr/m_nosecone)*dt;
    v_curr = v_i + dv;
    dh = v_i * dt;
    v_i = v_curr;
    h_curr = h_i + dh;
    h_i = h_curr;
    n_ns = n_ns + 1;%counting the iterations for time

end
%calculating decent profile for payload section
v_i = 0;
h_i = 12192; %(40000 ft)
v_curr = v_i;
h_curr = h_i;
velocities_pl = [];
heights_pl = [];
n_pl = 0;

while (h_curr > 10)

    heights_pl = [heights_pl h_curr];
    velocities_pl = [velocities_pl v_curr];

    F_g = m_payload * g_0;
```

```
    F_drag = .5 * den(round(h_curr)) * v_curr^2 * a_chute_pl * cd_parachute;
    F_curr = F_drag - F_g;
    dv = (F_curr/m_payload)*dt;
    v_curr = v_i + dv;
    dh = v_i * dt;
    v_i = v_curr;
    h_curr = h_i + dh;
    h_i = h_curr;
    n_pl = n_pl + 1;

end

%calculating decent profile for engine section
v_i = 0;
h_i = 12192; %(40000 ft)
v_curr = v_i;
h_curr = h_i;
heights_eng = [];
velocities_eng = [];
n_eng = 0;

while (h_curr > 10)

    heights_eng = [heights_eng h_curr];
    velocities_eng = [velocities_eng v_curr];

    F_g = m_engine * g_0;
    F_drag = .5 * den(round(h_curr)) * v_curr^2 * a_chute_eng * cd_parachute;
    F_curr = F_drag - F_g;
    dv = (F_curr/ m_engine )*dt;
    v_curr = v_i + dv;
    dh = v_i * dt;
    v_i = v_curr;
    h_curr = h_i + dh;
    h_i = h_curr;
    n_eng = n_eng + 1;

end

velocity_nc = velocities_nc(end)
velocity_pl = velocities_pl(end)
velocity_eng = velocities_eng(end)
%in seconds (each iteration represents .1 seconds, so divide by 10 to get
%seconds)
time_ns = n_ns/10
time_pl = n_pl/10
time_eng = n_eng/10


termVel_nc = velocities_nc(80:end);
hold on
%velocities are negative, so multiplied by -1 to make positive
plot(heights_nc(80:end), termVel_nc .* -1);
plot(heights_pl(80:end), velocities_pl(80:end) .* -1, 'r');
```

```
plot(heights_eng(80:end), velocities_eng(80:end) .* -1, 'g');
legend('Nosecone', 'Payload', 'Engine');
xlabel('Heights (m)');
ylabel('Terminal Velocities (m/s)');
title('Descent Profiles');
grid on;
```

```
velocity_nc =

   -7.0913


velocity_pl =

   -8.4105


velocity_eng =

   -7.4430


time_ns =

   1.2674e+03


time_pl =

   1.0689e+03


time_eng =

   1.2076e+03
```



Published with MATLAB® R2013a