ASEN 2012 Project 2 - Group Portion Final Rocket Configuration

Table of Contents

Setup 1
Simulate! 1
Simulation 2
Extraction 3
Plotting 3


```
By: Ian Faber and Jack Walsh
SID: 108577813, 10040714
Started: 12/5/21, 18:30
Finished: 12/7/21, 23:38

Runs an optimized bottle rocket simulation subject to a set of initial
parameters through 3 different phases of flight: water thrust, air
thrust, and ballistic flight. This specific script utilizes a custom
ODE45 EOM function and rewrites the initial parameters of coefficient
of drag, initial launch angle, initial water volume, initial air
pressure based on the trends we found in our simulation scripts.
```

Setup

```
% Housekeeping
clc; clear; close all;

% Get all constants from the const structure
const = getConst();

target = 85;

% Change parameters to optimized values
const.Cdrag = 0.2;
const.thetaInit = 45;
const.VWaterInit = 0.001;
const.PGageInit = 50;
```

Simulate!

```
% Difference of Bottle and initial water volumes
VAirInit = const.Vbottle - const.VWaterInit;

% Need absolute pressure of air, also convert psi to Pa
PAirInit = (const.PGageInit+const.PAmb)*6894.76;
```

```
% Calculate rho w/ Ideal Gas EOS
rhoAirInit = (PAirInit)/(const.R*const.TAirInit);

% Calculate initial masses
mAirInit = rhoAirInit*VAirInit;
mWaterInit = const.rhoWater*const.VWaterInit;
mRocketInit = const.mBottle + mAirInit + mWaterInit;

% Calculate initial x and z velocities
vx0 = const.vInit*cosd(const.thetaInit);
vz0 = const.vInit*sind(const.thetaInit);

% Format the initial conditions vector, and by extension variables to
% integrate
X0 = [const.xInit; const.zInit; vx0; vz0; mRocketInit; mAirInit;
 VAirInit];

% Define events worthy of stopping integration
options = odeset('Events',@phase);
```

Simulation

```
% Integrate! Solves for the trajectory of the rocket by integrating
 the
% variables in X0 over tspan according to the derivative information
% contained in rocketEOM. Also stops integration according to
 "options," a
% predefined set of stopping conditions
[time, state, timePhases, ~, ~] =
 ode45(@(t,state)rocketEOM(t,state,const), const.tspan, X0, options);

% Extract intermediate variables from rocketEOM for debugging,
 particularly
% weight, drag, thrust, and air pressure. Found this approach on the
 MATLAB
% forums.
[~,gravCell, dragCell, thrustCell, PairCell] =
 cellfun(@(t,state)rocketEOM(t,state.',const), num2cell(time),
 num2cell(state,2), 'uni', 0);

%Allocate space for intermediate variables
gravity = zeros(length(time),1);
drag = zeros(length(time),1);
thrust = zeros(length(time),1);
Pair = zeros(length(time),1);

% Extract intermediate variables from their cells
for i = 1:length(time)
    gravity(i) = norm(gravCell{i});
    drag(i) = norm(dragCell{i});
    thrust(i) = norm(thrustCell{i});
    Pair(i) = norm(PairCell{i});
end
```

Extraction

```
% Extract variables of interest
rocketX = state(:,1);
rocketZ = state(:,2);
rocketVx = state(:,3);
rocketVz = state(:,4);
rocketM = state(:,5);
rocketMair = state(:,6);
rocketV = state(:,7);

% Find maximum values of interest
maxRange = max(rocketX)
maxHeight = max(rocketZ)
maxVx = max(rocketVx)
maxVy = max(rocketVz)
maxThrust = max(thrust);
```

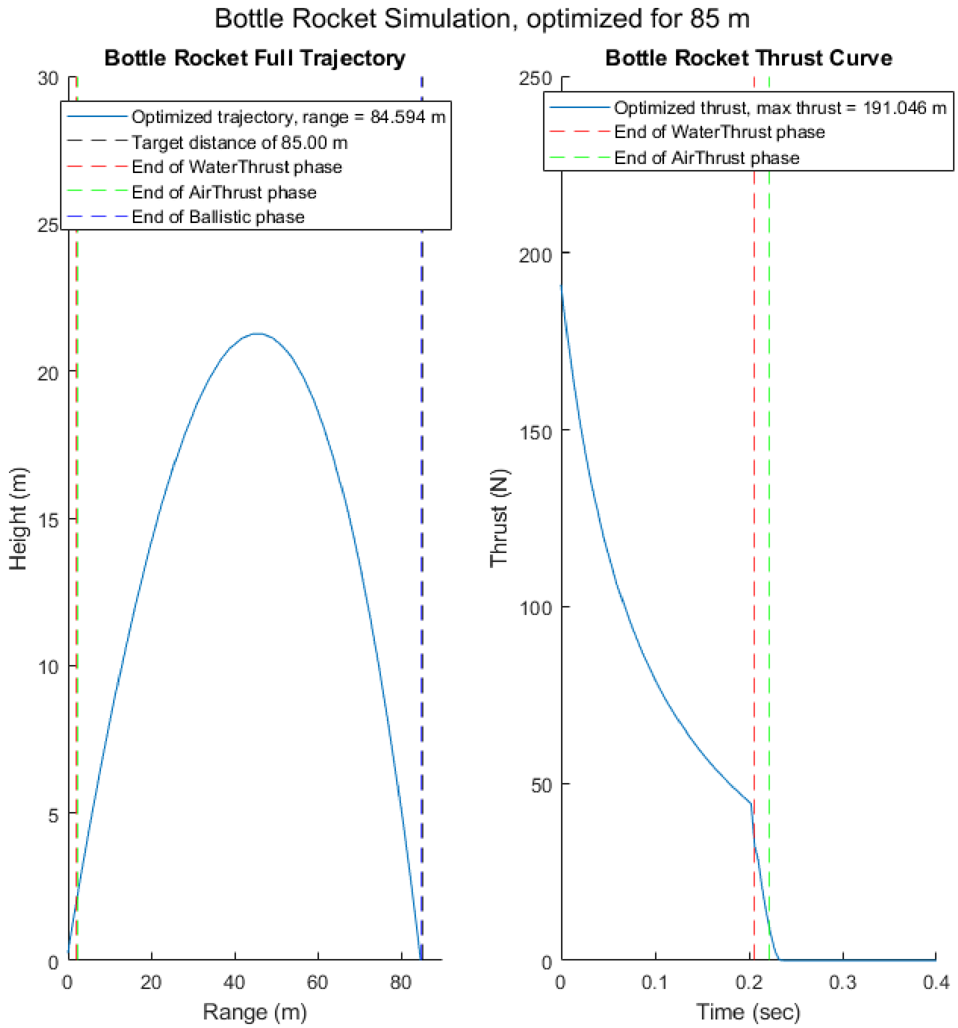
```
maxRange =

   84.5941


maxHeight =

   21.2554


maxVx =

   26.2184


maxVy =

   21.0934
```

Plotting

```
f = figure();
f.Position = [100 100 740 740]; % Start at (100, 100), end at (100 +
 740, 100 + 740)
sgtitle("Bottle Rocket Simulation, optimized for 85 m")

% Trajectory
subplot(1,2,1)
hold on;

title("Bottle Rocket Full Trajectory");

label = strings(1,5);
```

```
label(2) = sprintf("Target distance of %.2f m", target);
label(3) = sprintf("End of WaterThrust phase");
label(4) = sprintf("End of AirThrust phase");
label(5) = sprintf("End of Ballistic phase");

plots = zeros(1,5);

plots(1) = plot(rocketX, rocketZ);
label(1) = sprintf("Optimized trajectory, range = %.3f m",
 max(rocketX));

t1 = find((time >= 0.95*timePhases(1)) & (time <=
 1.05*timePhases(1)));
t2 = find((time >= 0.95*timePhases(2)) & (time <=
 1.05*timePhases(2)));
t3 = find(time == time(end));

plots(2) = xline(85, 'k--');
plots(3) = xline(rocketX(t1(1)),'r--');
plots(4) = xline(rocketX(t2(1)),'g--');
plots(5) = xline(rocketX(t3(1)),'b--');

xlim([0, 90]);
ylim([0, 30]);
xlabel("Range (m)");
ylabel("Height (m)");
legend(plots, label, 'Location', 'best');
hold off;


% Thrust
subplot(1,2,2)
hold on;

title("Bottle Rocket Thrust Curve");

label = strings(1,3);
label(2) = sprintf("End of WaterThrust phase");
label(3) = sprintf("End of AirThrust phase");

plots = zeros(1,3);

plots(1) = plot(time, thrust);
label(1) = sprintf("Optimized thrust, max thrust = %.3f m",
 max(thrust));

t1 = find((time >= 0.99*timePhases(1)) & (time <=
 1.01*timePhases(1)));
t2 = find((time >= 0.99*timePhases(2)) & (time <=
 1.01*timePhases(2)));

plots(2) = xline(time(t1(end)),'r--');
plots(3) = xline(time(t2(end)),'g--');
```

```
xlim([0 0.4]);
ylim([0 250]);
xlabel("Time (sec)");
ylabel("Thrust (N)");
legend(plots, label, 'Location', 'best');
hold off;
```



Published with MATLAB® R2021a