Celal Bayar University Journal of Science

Simulating Rocket Trajectory Using JSBSim and Optimal Thrust Profile for Maximizing Altitude of a Sounding Rocket

Sohayb Abdulkerim[1*]

[1] Gaziantep University, 27310, Şehitkamil, Gaziantep, Türkiye
*sohayb.abdulkarim@southwales.ac.uk
*Orcid: 0000-0002-3448-9129

Received: 21 April 2022
Accepted: 1 December 2022
DOI: 10.18466/cbayarfbe.1107088

Abstract

Solid rocket boosters are commonly used for launching sounding rockets due to their simplicity and power-ness. The shape and geometry of the propellant grain determine the thrust-time profile which has a significant effect on rocket performance. In practical application, the thrust profile has three typical curves; regressive, neutral, and progressive. A great deal of studies has been focused on optimizing the trajectory based on various state variables in which the profile of the thrust-time curve was not among those variables. In this research, design variables were the thrust profile, the object function was maximizing the altitude subjected to constraints of a fixed amount of fuel. The trajectory was found by solving the equations of motion. For comparison purposes, the trajectory was also found using JSBSim, an open-source flight dynamic simulator. In the results of the optimization process, the input thrust-time curve was evolved into an unusual shape, the letter "V" shaped. In this type of profile, the thrust curve starts regressively until reaches zero value at the middle of the burning time and then continues progressively until the end. This behavior can be satisfied only by two-stage boosters. Thus, these results show that two-stage boosters perform better than single-stage. The improvement is obtained by consuming the solid propellant more efficiently allowing fewer energy losses. This reason is added to the fact that two-stage boosters allow reducing the total masses due to the cassette separation.

Keywords: Flight Simulator, JSBSim, Optimization, Sounding Rocket, Thrust-Time Profile

1. Introduction

The main purpose of a sounding rocket is to deliver the weather monitoring instruments to a certain altitude and then return them to the ground, which is normally launched by a solid rocket booster (SRB). The SRB mainly consists of a grain, or a charge, and a nozzle. The grain part is burned to convert the chemical energy of the fuel into thermal one by producing gasses at high pressure and temperature that flow through the nozzle. The nozzle converts the potential thermal energy of the gasses into kinetic energy, which produces the necessary thrust based on action and reaction principles.

The shape and the geometry of the grain affect the burning surface and hence the rate of gases production versus time, which in turn defines the value of the internal pressure. The pressure determines the level of the generated thrust and its variation over time. Therefore, the resulting Thrust-Time Curve (TTC) depends on the shape and the geometry of the grain. In general, there are three typical TTCs; progressive, neutral, and regressive as shown in **Fig. 1** [1].

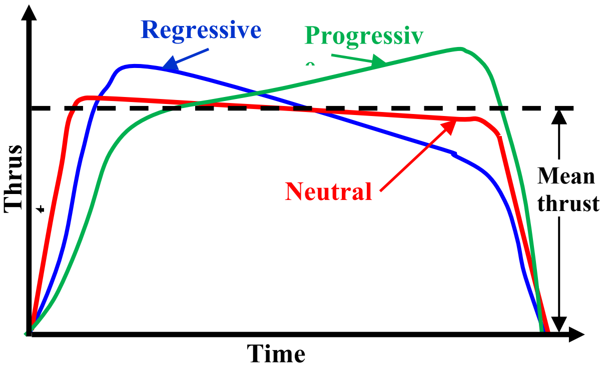


Fig. 1. Typical thrust-time curves [1].

Although the thrust in the three curves starts from zero and requires some time to reach a considerable value, It behave differently in the remaining time. The thrust in the regressive curve reduces gradually after reaching a maximum value. Conversely, the progressive thrust continues to increase until the burn-out. In contrast, the neutral maintains the thrust value almost close to the mean value. in the progressive and the regressive thrust, it is possible to obtain high thrust which is good for gaining high acceleration. on the other side, high thrusts cause structural stresses higher than average compared to the neutral thrust [1].

Also, the total mass during the burning stage varies accordingly. Since the acceleration of the rocket depends on the acting thrust and total mass, including the mass of the grain, the profile of the TTC has a significant impact on the motion of the rocket, in terms of velocity and altitude. However, understanding this impact becomes more complex when considering drag forces that resist the motion of the rocket and depend on the resulting velocity and air density, which depends on the resulting altitude.

Therefore, it is still unclear what the optimum shape of the TTC is to produce the highest altitude for a given amount of fuel. Obviously, an optimization problem needs to be first formulated and then solved.

In the literature, there is a great deal of research solving the optimization problem for the rocket performance based on certain parameters, such as the outer diameter of the chamber, thickness of grain, expansion ratio, controller, material, geometry, and center of mass. For example, exit design Mach, the nozzle throat area, and the burning angle of the grain were optimized by Navarrete et. Al. [2]; Rocket and the grain geometries were the design variable for three stages thrust hybrid rocket motor by Xiao [3]; the overall geometries of a hybrid rocket were also optimized in addition to others factors in [4]. Maximizing the mass of the payload and minimizing the initial mass in [5]. Sang-Hyeon optimized the Mach number of the nozzle to achieve the maximum altitude [6]. Most research above only considered the thrust as a design variable. However, the profile of the TTC is not among those parameters. Therefore, this project aims to propose a methodology to find the profile of the TTC to achieve the optimum rocket performance, in terms of the highest apex that the rocket can reach using the same amount of fuel.

Therefore, at the first stage, the modeling process must be carried out either with a three degree of freedom (DOF) or 6DOF model. The 6DOF model is necessary only when problems like stability, flexibility, and control need to be investigated. In addition, the 6DOF model needs high computational resources. While a simplified model using 3DOF is more efficient for performance evaluation like trajectories. This type of modeling assumes the rocket is a point-mass object moving in a plain. This model was implemented by several studies such as [3, 5], and [7-9]. To use this model, the Equation of Motion (EOM) was derived and then coded using MATLAB.

In the second stage, the modeling verification, then formulating and solving the optimization problem were carried out. For validation purposes, the numerical results of the MATLAB code were compared with the results of the JSBSim, which is a high fidelity, Flight Dynamics Model (FDM), open-source package. Literature review shows several studies relied on JSBSim for simulation of air vehicles. JSBSim has been commonly in use for a variety of configurations, which has been in development voluntarily by a number of developers since 1997 [10-13], it is capable of modeling 6DOF air-vehicle of various types, controlled or uncontrolled systems. The JSBSim was used in the NACA report to verify several check cases of 6DOF flight Vehicle Simulation [14]. Examples of aircraft trim algorithm using JSBSim was also reported by [15], modeling of a mini-UAV by [16], a reinforcement learning environment by [17], evaluating flying UAV by [18], validating MATLAB/Simulink AeroSim Blockset by [19], experimenting flight quality by [20], and much other application such as [19-21].

2. Theoretical background

Since the aim of this project is to improve the performance of sounding rocket without increasing the amount of fuel, by solving an optimization problem, the value of apex, the highest reached altitude, is set as an objective function constrained by maintaining the same amount of fuel. While the profile of the thrust curve is set as a state variable. The maximum value of the internal pressure of the SRB was restricted by limiting the peaks value of the thrust-time curve too. Therefore, to solve the optimization problem, the time's functions of thrust and fuel mass need to be identified; and light model of the rocket motion had to be established, and the optimization problem had to be formulated. As will be described in the consequence sections.

2.1 Modelling Thrust

The thrust curve is normally identified using tabulated data of large number of records. Using this format as design variables it requires high computational resource. However, the number of variables can be reduced by performing a curve-fitting process. Fourier Series, due to its high flexibility in accommodating any curve shape, was selected for the curve fitting process. Therefore, the thrust, F(t), at time, t, is be approximated using Fourier series as follows [23]:

$$F(t) = a_o + \sum_{i=1}^{n} a_i \cos(i.\alpha.t) + \sum_{1}^{n} b_i \sin(i.\alpha.t), \tag{2.1}$$

where, n is number of terms which is greatly affects the accuracy. a_i, b_i, and α are fourier coefficients which are normally defined using the curve fitting process. To calculate the mass of the rocket at any instant of time, t, the fuel mass flow is firstly determined as follows:

$$\dot{m}(t) = \frac{F(t)}{g.I_{sp}}, \qquad \textbf{(2.2)}$$

where, I_{sp} is the spcific impluse in [sec], and g is gravity acceleration [m/sec].
By integrating mass flow rate, the fuel mass function at time, t is calculated as follows:

$$m(t) = m_o - \int_0^t \dot{m}(t)dt, \qquad \textbf{(2.3)}$$

where, m_o is the initial mass of fuel.

2.2 Modelling the Rocket Motion

In general, the rocket motion is very complex and is affected by many parameters. For simplification purposes, several assumptions are considered in the derivation process of the EOM) as follows: 1- The earth is flat rather than curved, this leads to neglect of centrifugal forces due to moving on a curved path. 2- The earth is non-rotating objects rather than rotating, which means ignoring the effects of Coriolis's forces, 3- The earth's gravity has a constant and approximate value of 9.81 m/s^2. 4-The total mass of the rocket is assumed to be constant, and a point mass is located at the gravity center. These assumptions allow deriving the EOM by using Newton's second law in the longitudinal plane and by inspecting the free body diagram as shown in **Fig. 2**.

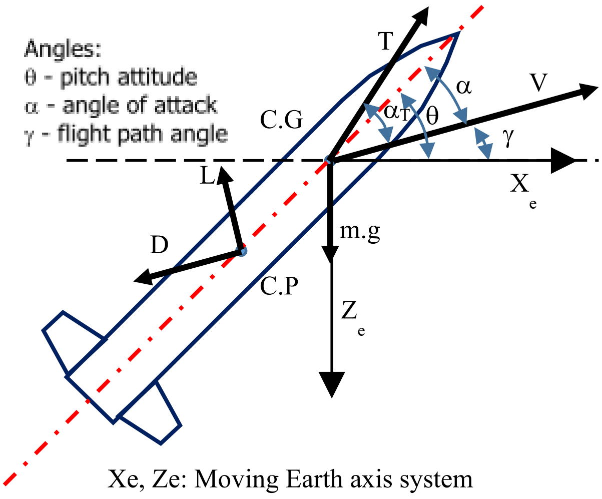


Fig. 2. Free body diagram of a rocket in a plane.

In the velocity-parallel direction, Newton's second law gives [8]:

$$m.\frac{dV}{dt} = T.\cos(\alpha_T) - D - m.g.\sin(\gamma), \qquad \textbf{(2.4)}$$

while in the velocity-perpendicular direction, it gives,

$$m.V.\frac{d\gamma}{dt} = T.\sin(\alpha_T) + L - m.g.\cos(\gamma), \qquad \textbf{(2.5)}$$

where, $D = q.S_{ref}.CD$, is the aerodynamic drag, $L = qS_{ref}.CL,$ is the aerodynamic lift, S_{ref} is the reference area, $CD \ and \ CL$, is the drag and lift coefficients, respectively, q is the dynamic pressure, which is propertional to the air density. The air density in this project was calculated using the International Standard Atmosphere (ISA) Model. In addition to kinematic equations, the rate of climb is given by

$$\dot{H} = V.\sin(\gamma) \qquad \textbf{(2.6)}$$

And the horizontal velocity component,

$$\dot{R} = V.\cos(\gamma) \qquad \textbf{(2.7)}$$

Skin-friction drag components are approximated based on Reynolds number which determining the flow nature. For laminar flow the friction drag coefficient is approximated using Blasius Solution [24] given as follow

$$C_f = \frac{0.664}{\sqrt{Re}} \qquad \textbf{(2.8)}$$

while For turbulent flow, it is approximated using the Prandtl's One-Seventh-Power Law [24] as follow

$$C_f = \frac{0.027}{Re^{\frac{1}{7}}} \qquad \textbf{(2.9)}$$

$$D_{skin} = C_f.0.5.\rho.V^2.S_{wet} \qquad \textbf{(2.10)}$$

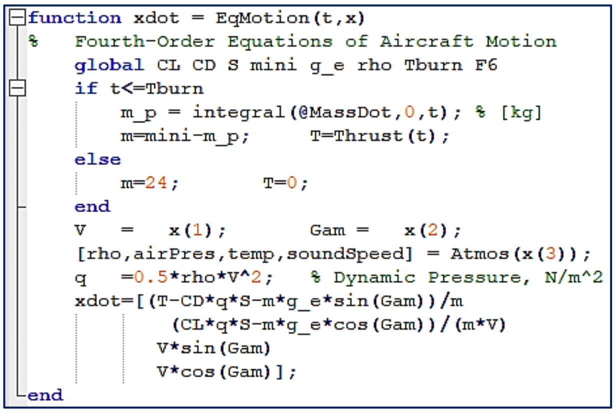
where, $S_{wet},$ is the wet area of the body.
These equations, (2.4), (2.5), (2.6), and (2.7) are describing an approximate motion of the rocket in the longitudinal plane being a point-mass object, which can be rewritten using vector format as follows,

$$\dot{X} = \begin{bmatrix} \dot{V} \\ \dot{\gamma} \\ \dot{h} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \frac{T}{m}.\cos\,(a_T) - C_D q\frac{S_{ref}}{m} - g\ sin\gamma \\ \left(C_L q\frac{S_{ref}}{m} - g\ cos\gamma + \frac{T}{m}.sin(a_T)\right)\Big/V \\ V\ sin\gamma \\ V\ cos\gamma \end{bmatrix}$$

$$\textbf{(2.11)}$$

This equation was coded using MATLAB as shown in

Fig. 3. The state vector contains four state variables: velocity, path angle, altitude, and horizontal position. The state format allows easily to implement ode45, the MATLAB built-in function, which is dedicated to solving partial differential equations. By using this function, it is possible to obtain an approximate rocket's trajectory at low computational resources.



```
function xdot = EqMotion(t,x)
%   Fourth-Order Equations of Aircraft Motion
    global CL CD S mini g_e rho Tburn F6
    if t<=Tburn
        m_p = integral(@MassDot,0,t); % [kg]
        m=mini-m_p;     T=Thrust(t);
    else
        m=24;       T=0;
    end
    V   =   x(1);       Gam =   x(2);
    [rho,airPres,temp,soundSpeed] = Atmos(x(3));
    q   =0.5*rho*V^2;   % Dynamic Pressure, N/m^2
    xdot=[(T-CD*q*S-m*g_e*sin(Gam))/m
            (CL*q*S-m*g_e*cos(Gam))/(m*V)
           V*sin(Gam)
           V*cos(Gam)];
end
```

Fig. 3. MATLAB code defining the EOM

2.2.1 Optimization Problem Formulation

The optimization problem can be formulated as follow: The objective function is maximizing the rocket apogee, the highest altitude of the path, the apex point of the trajectory. Since, the MATLAB-built-in-function, **fmincon**, was used, the objective function is converted to minimizing the inverse of the apogee altitude as follows

$$Minimize\ \left(\frac{1}{h_{max}}\right)\ , \qquad \textbf{(2.12)}$$

where, h_{max} is the highest value of the path (apogee altitude).

The objective function is subject to a set of nonlinear inequality constraints on the thrust function, F(t), using Eq. (2.1), this is necessary to avoid peaks in the internal pressure of the chamber as follows:

$$F(t) > 0; \quad 0 \le t \le t_b\ , \qquad \textbf{(2.13)}$$

where, *t* is the time variable, t_b is the burning time, and it is subject to nonlinear equality constraint, Eq. (2.14), which keep the total amount of fuel is unchanged, as follows:

$$\int_0^{t_b} \dot{m}(t)dt = M_{fuel}, \qquad \textbf{(2.14)}$$

where, M_{fuel} is the fuel mass,

The Fourier series coefficients, that used in calculating thrust, F(t), are also constrained by upper and lower limits as follows: The lower limits were $a_i = -1E5;\ b_i = -1E5;\ i = 0..6;$ while the upper limits were $a_i = +1E5;\ b_i = +1E5;$ where $i = 0..6$.

2.3 Case Study

All necessary input data of the a sounding rocket was found at textbooks of [23,25] as it is summarized as shown **Table 1**.

Table 1. Numerical data of a case study of a sounding rocket [23,25].

Total mass	34.68 [Kg]
Fuel Mass	10.84 [Kg]
Isp, spcific impulse	196.50 [sec]
Burning time	6.16 [sec]
Body diameter	131.0 [mm]
Total length	3000.0 [mm]
Aerodynamic Drag Coefficient, CD	0.24059
Aerodynamic Lift Coefficient, CL	0.0000
The thrust curve	Shown in Fig. 5

The analysis was started, firstly, by finding the thrust curve, then calculating the trajectory by two methods the in-house code using MATLAB and using the open-source package, JSBSim, then results-verification was carried out by comparing the two results. Finally, the optimization scheme on the trajectory using the first method was performed as it will be explained in the next sub sections.

2.3.1 Thrust-Time Curve

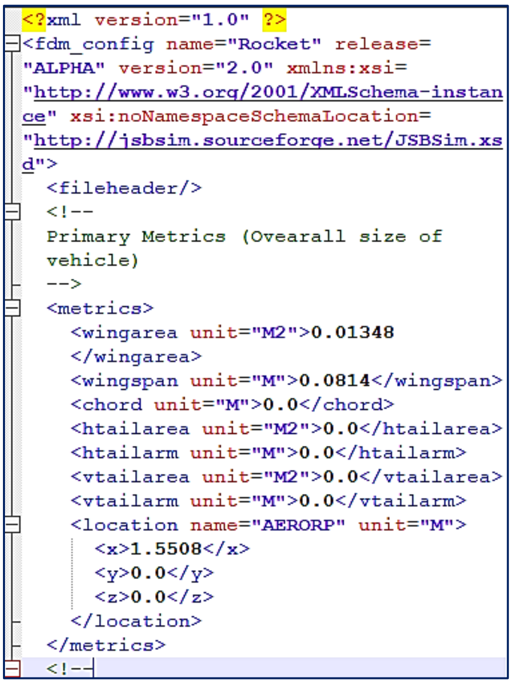
The curve fitting process of the thrust-time was carried out by MATLAB using Fourier series, Eq. (2.1), and using Nonlinear Least Squares (NLS), the procedure was followed in [25]. To show the accuracy of the process, the resulting curve along with the original discrete data are plotted as shown in **Fig. 5** which shows an adequate agreement. Thus, using this technique, the discrete thrust function was determined using continuous function with far fewer number of variables, 15 terms rather than 63 discrete points.

2.3.2 Rocket Trajectory and Verification

The EOM, Eq. (2.11), were coded using MATLAB function, as shown in

Fig. 3, involving, **ode45**, Runk Kutta method, to march the solutions in the time domain using a time step of dt=0.001 [sec] for given initial conditions. To avoid singularity, the initial velocity was given a very low value, the pitch attitude was given at 89.99 [deg]. For results verification, an open-source package, JSBSim which is available on the GitHub platform, was used to solve the same problem too. All necessary input data, as listed in **Table 1**, were also figured out again using separate XML (eXtensible Markup Language) formatted files suitable for JSBSim. Four XML files were prepared, as follows:

The first XML file, as shown **Fig. 4**, included geometries, inertia characteristics, aerodynamics drag defined by the value of CD only, mass fuel flux, and the location and the orientation of the power system.



```
<?xml version="1.0" ?>
<fdm_config name="Rocket" release=
"ALPHA" version="2.0" xmlns:xsi=
"http://www.w3.org/2001/XMLSchema-instan
ce" xsi:noNamespaceSchemaLocation=
"http://jsbsim.sourceforge.net/JSBSim.xs
d">
  <fileheader/>
  <!--
  Primary Metrics (Ovearall size of
  vehicle)
  -->
  <metrics>
    <wingarea unit="M2">0.01348
    </wingarea>
    <wingspan unit="M">0.0814</wingspan>
    <chord unit="M">0.0</chord>
    <htailarea unit="M2">0.0</htailarea>
    <htailarm unit="M">0.0</htailarm>
    <vtailarea unit="M2">0.0</vtailarea>
    <vtailarm unit="M">0.0</vtailarm>
    <location name="AERORP" unit="M">
      <x>1.5508</x>
      <y>0.0</y>
      <z>0.0</z>
    </location>
  </metrics>
  <!--
```

Fig. 4. XML code defining major main parameters.

The second file included the specific impulse and the thrust curve using tabulated data of the thrust curve which consists of two columns, the first column is the burned fuel while the second column is the thrust. The third file presented the nozzle features. For sake of simplicity, the nozzle exit area was set to zero in order to neglect the altitude dependency of the thrust. The fourth file includes the initial conditions which were set to zeros, only the pitch angle was set to 89.99 [deg] in order to obtain a vertical takeoff.

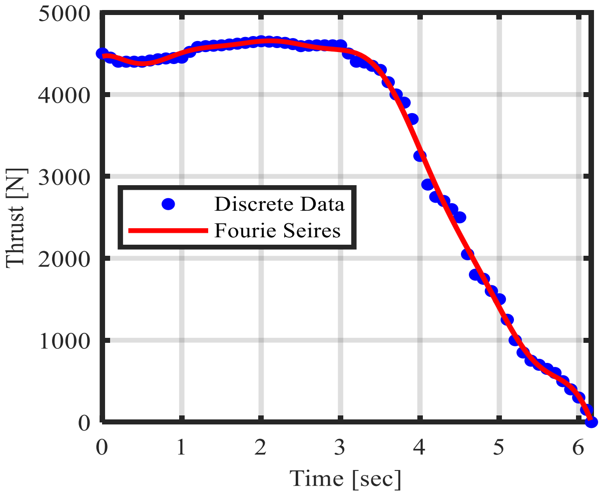


Fig. 5. Thrust curve using discrete data and using FS.

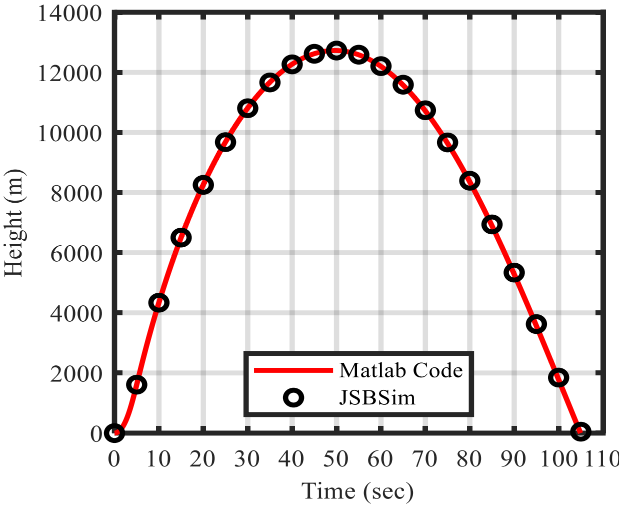


Fig. 6. Comparing the MATLAB result with JSBSim open source

Thus, having the input files ready, it was possible to use a batch file to run the simulator and obtain a CSV file of the results which included time series of the resulting trajectory at a predefined frequency which is plotted along with the MATLAB code results, as shown in **Fig. 6**. It is worth mentioning that the skin friction drag is not included in the MATLAB code since it is not determined by JSBSim too. Clearly, the comparison of the results shows an excellent agreement between the two methodologies giving verification of the adopted methodology of calculation of the rocket trajectory and can be used for the optimizer as explained in the next section.

2.4 Optimization Results

The input data is taken from the same case study in the verification section, **Table 1**. The MATLAB built-in functions, 'fmincon' involving Sequential quadratic programming (SQP) algorithm. It is worth mentioning that the apex trajectory was inverted so that the minimum value was sought rather than the maximum. The optimization code was executed in two stages. The original TTC, the first resulted TTC, and the final resulted TTC were plotted on the same chart as shown in Fig. 7. In the first stage, the thrust-time curve was evolved into a horizontal line, which means a neutral thrust profile. In the second stage, the curve evolved into unusual behavior which shows two segments. The first segment starts from the beginning of ignition and ends at the instant of 3.2 [sec] of time (middle of the burning time; the total burning time is 6.16 sec); the second segment starts from the middle and ends with ending the burning process. While in the first segment, the thrust is decreasing to zero (regressive), in the second segment, the thrust is increasing from zero, (progressive).

Although the three curves require the same amount of solid propellant, 10.8 [kg], the performance is not the same. To show the differences, the three corresponding trajectories were plotted on the same set of axes as shown in Fig. 8. The figure shows that the apex point is 9256 [m] in the original TTC. While it is increased to 9351 [m] in the second TTC; and increased further to 9588 [m] in the final TTC.

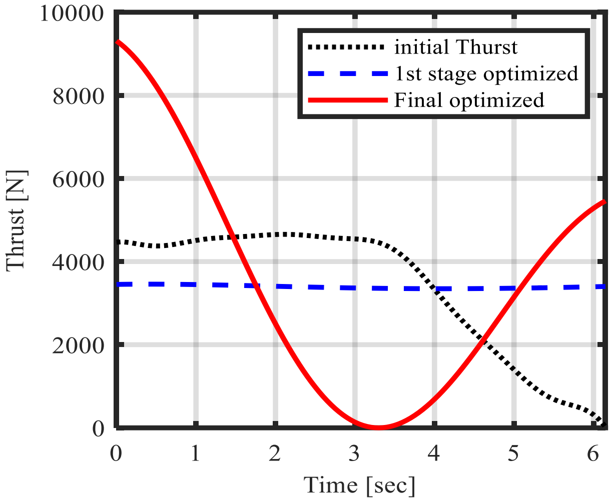


Fig. 7. Comparing resulted thrust curve with the initial thrust curve.

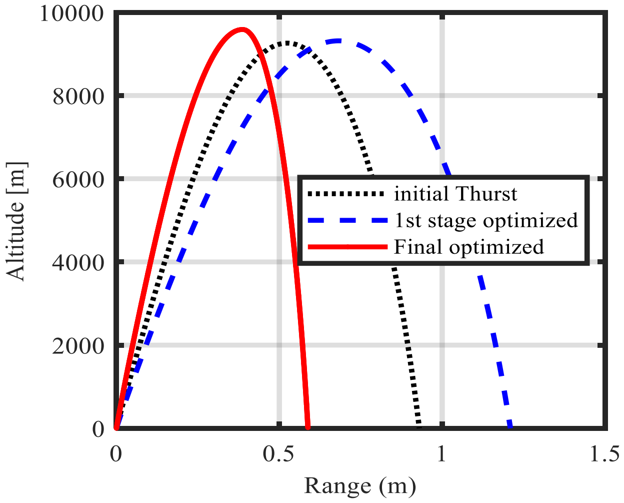


Fig. 8. Comparing trajectories of three cases of thrust curves.

3. Discussion

By inspecting the resulting trajectories from the optimization process, they show that the optimization process has improved the value of the apex point by an increment of 2.5%. It was noticed that in spite of neutral TTCs were selected many times as an input. The resulted curves have a shape of the letter “V”. Factually, the resulted curves are split into two segments. in another word, the total propellant is equally divided into two and burned respectively.

The interpretation for that division of the propellant results can be understood from the fact that the losses of the rocket kinetic energy are due to the action of drag forces which, in turn, become less as the air-vehicle gains higher altitude (less air density). The energy losses become less when flying at a lower speed since the lost energy is proportional to speed multiplied by drag force which, in turn, is proportional to speed squared. Thereby, the energy losses are proportional to the speed cubed. As a result, the energy losses are strongly dominated by speed and altitude.

Thus, the optimum rocket requires to fly at low speed, but it requires to reaches high altitude as fast as possible. Therefore, the optimum consumption of the propellant is to be burned on two durations, in the first, the altitude is gained with minimum possible losses, while saving the rest of the charges to fly more efficiently at higher altitudes.

Therefore, dividing the charges into two parts is a good strategy, in the first part, the rocket gains the highest possible altitude, where the air density is less, and maintains low speeds by regressive thrust, while the rockets use the rest part of the charge to fly more efficiently (less drag, less loses) using progressive thrust in which increases as the rocket rises to higher altitude reducing the air density and alleviating the effect of accelerating on the energy losses. In practical application the progressive thrust can be satisfied by igniting the internal surface of a cylindrical grain; the outer surface of it gives the regressive one. Dropping the TTCs to zero in the middle of burning time can be achieved only if the burning process is fully stopped. In Fact, unlike liquid propellants or bi-propellants, solid propellants cannot be stopped and restarted, after ignition. Therefore, single-stage boosters cannot perform the resulting TTC. Alternatively, two stages booster allows two segments of thrust-curve and can be separated by zero values for any duration of time. This result leads to conclude that a single-stage booster is not the best choice, while the two-stage boosters result in a higher altitude of the apex at least for the tested cases in this project.

Accordingly, multi-stage boosters are commonly used in practical applications because they enable the total masses of the rockets to be reduced by releasing the cassettes which are reflected directly in a noticeable improvement in apogee altitudes. However, the results of this project give additional justification for the feasibility of using multi-stage boosters. Additionally, the promising results give to the importance of relying on optimization techniques to tune all design parameters for every single design case.

Additionally, the resulting TTC shows smoothness. This means that it is possible to produce in practical application by adjusting the geometry of the grain. This smoothness of the resulted curves was possible due to using Fourier Series to model the thrust curves.

Comparing with similar optimization in literature, Navarrete-Martin used Hopsan software for modelling and used the case thickness as a design variable. The highest apex was found at optimum thickness [2].
While Okninski optimized single-stage thrust curve to reduce the total mass [26]. In his example the mass reduced from 515 kg to 360 kg.

4. Conclusions

In this project, the simulation of the sounding rocket, powered by a solid propellant, was modeled using the mass-point assumption. The simulation process was carried out twice, using MATLAB code and using JSBSim, the open-source package. The results showed good agreements. The full details of files necessary for JSBSim are attached in the appendix for tutorial purposes.

In the next stage, to find the best thrust-time curve yielding the highest possible apex for a vertical lunching, an optimization problem was formulated and solved. The objective function was the height value of the apex subjected to nonlinear constraints and some bounds allowing practical limits to be respected. The constraints on the thrust curve were imposed so that the amount of fuel, the burning time, and the total impulse are maintained; while the upper bounds were real positive numbers limiting the peaks of the thrust, and the lower bounds were zero values. The state variables were set to be a function determining the TTC.

In order to reduce the number of state variables, the TTC function was modeled using Fourier's series. The optimization process was carried out using a MATLAB built-in function, fmincon. The code was able to evolve the input of an arbitrary thrust curve into the letter V-shaped TTC. The optimum TTC has regression, zero, and progression behavior; the corresponding value of the apex has improved by 2.5%. The thrust according to the resulted profile requires the burning process of the propellants to be fully stopped and then restarted after the first ignition. This is only possible if the booster is made in two stages. However, in practical application, multi-stage boosters are common for a different reason which is due to reducing the mass of the booster's cassette, because of the separation. Furthermore, the result of this study adds a new justification of the effectiveness of using two-stage boosters in obtaining higher apogee. Additionally, this study emphasizes the possibility of using optimization algorithms to tune any design parameter for optimum performance.

Author's Contributions

Sohayb Abdulkerim: Drafted and wrote the manuscript, performed the experiment and result analysis.

Ethics

There are no ethical issues after the publication of this manuscript.

References

[1]. A.F. El-Sayed, Rocket Propulsion, *Fundamentals of Aircraft and Rocket Propulsion*; 2016. 907–991. https://doi.org/10.1007/978-1-4471-6796-9_11.

[2]. L. Navarrete-Martin, P. Krus. 2018. Sounding Rockets: Analysis, simulation and optimization of a solid propellant motor using Hopsan, *Transportation Research Procedia*; 29:255–267. doi:10.1016/j.trpro.2018.02.023.

[3]. M. Xiao, H. Zhu, Z. Du, Y. Gao, H. Tian, G. Cai. 2021. Design optimization of velocity-controlled cruise vehicle propelled by throttleable hybrid rocket motor, *Aerospace Science and Technology*; 115: 106784. doi:10.1016/j.ast.2021.106784.

[4]. P. WANG, H. TIAN, H. ZHU, G. CAI. 2020. Multi-disciplinary design optimization with fuzzy uncertainties and its application in hybrid rocket motor powered launch vehicle, *Chinese Journal of Aeronautics;* 33: 1454–1467. doi:10.1016/J.CJA.2019.11.002.

[5]. Simultaneous optimization of staging and trajectory of launch vehicles using two different approaches | *Elsevier Enhanced Reader*, (n.d.). https://reader.elsevier.com/reader/sd/pii/S1270963811001064?token=638D630A7AC21B69A939E91C6715B6E94EA0850CB7BB840A6CBB6AB2479418E4E5BF8C03833007F9EA31C10F104CF918&originRegion=eu-west-1&originCreation=20210914151137 (accessed September 14, 2021).

[6]. S.H. Lee. 2017. Optimal nozzle Mach number for maximizing altitude of sounding rocket, *Aerospace Science and Technology*; 74:104–111. doi:10.1016/J.AST.2017.12.019.

[7]. K. Chiba, M. Kanazaki, T. Shimada. 2017. Simple control of oxidizer flux for efficient extinction–reignition on a single-stage hybrid rocket, *Aerospace Science and Technology;* 71:109–118. doi:10.1016/J.AST.2017.09.017.

[8]. H. Zhou, X. Wang, Y. Bai, N. Cui. 2017. Ascent phase trajectory optimization for vehicle with multi-combined cycle engine based on improved particle swarm optimization, *Acta Astronautica*; 140:156–165. doi:10.1016/J.ACTAASTRO.2017.08.024.

[9]. I.M. Rossi. 1996. An analysis of first-order singular thrust-arcs in rocket trajectory optimization, *Acta Astronautica*; 39:417–422. doi:10.1016/S0094-5765(96)00105-1.

[10]. J.S. Berndt, An open source, platform-independent, flight dynamics model in C++, n.d , *JSBSim*, 2008.

[11]. J.S. Berndt, JSBSim: An Open Source Flight Dynamics Model in C++, in: *AIAA Modeling Simulation Technologies Conference and Exhibit*, the American Institute of Aeronautics and Astronautics, Providence, Rhode Island, 2004: p. 27.

[12]. J.S. Berndt, A. De Marco, Progress on and Usage of the Open Source Flight Dynamics Model Software Library, JSBSim, 2009.

[13]. O. Cereceda, A Simplified Manual of the JSBSim Open-Source Software FDM for Fixed-Wing UAV Applications *TECHNICAL REPORT*, n.d.

[14]. D.G. Murri, / Nesc, E.B. Jackson, Check-Cases for Verification of 6-Degree-of-Freedom Flight Vehicle Simulations Appendices, 2015. http://www.sti.nasa.gov (accessed April 6, 2021).

[15]. A. De Marco, E.L. Duke, J.S. Berndt, *A General Solution to the Aircraft Trim Problem*, 2007.

[16]. T. Vogeltanz, R. Jašek, *JSBSim Library for Flight Dynamics Modelling of a mini-UAV*, (2015). doi:10.1063/1.4912770.

[17]. R. Titze, (PDF) Working Paper: Configuration and use of the flight dynamic model - *JSBSim - as a reinforcement learning environment*., (2021). doi:10.13140/RG.2.2.23839.69286.

[18]. J.P. Kim, D.L. Kunz, Flying qualities evaluation of an unmanned aircraft using *JSBSim, AIAA Atmospheric Flight Mechanics Conference* 2016-Janua (2016) 1–16. doi:10.2514/6.2016-3542.

[19]. O.C. Cantarelo, L. Rolland, S. O'Young, Validation discussion of an Unmanned Aerial Vehicle (UAV) using *JSBSim Flight Dynamics Model compared to MATLAB/Simulink AeroSim Blockset*, 2016.

[20]. V. Boisselle, G. Destefanis, A. de Marco, B. Adams, Signature-based detection of behavioural deviations in flight simulators - Experiments on Flight Gear and JSBSim, *PeerJ*. 4 (2016). doi:10.7287/peerj.preprints.2670v1.

[21]. A. Mairaj, A.I. Baba, A.Y. Javaid. 2019. Application specific drone simulators: Recent advances and challenges, *Simulation Modeling and Practice Theory*; 94:100–117. doi:10.1016/J.SIMPAT.2019.01.004.

[22]. F. Nicolosi, A. De Marco, V. Sabetta, P. Della Vecchia. 2018. Roll performance assessment of a light aircraft: Flight simulations and flight tests, *Aerospace Science and Technology*; 76:471–483. doi:10.1016/J.AST.2018.01.041.

[23]. S.D. Heister, W.E. Anderson, T.L. Pourpoint, R.J. Cassady, Rocket Propulsion, *Cambridge University Press*, 2019. doi:10.1017/9781108381376.

[24]. N.D. Katopodes, Boundary-Layer Flow, *Free-Surface Flow*, *Elsevier* (2019) 652–708. doi:10.1016/B978-0-12-815489-2.00009-5.

[25]. A. Tewari, Atmospheric and Space Flight Dynamics: Modeling and Simulation with MATLAB and Simulink, 1st ed., Birkhäuser Basel, 2007.

[26]. A. Okninski. 2017. Multidisciplinary optimisation of single-stage sounding rockets using solid propulsion, *Aerospace Science and Technology*; 71: 412–419. https://doi.org/10.1016/J.AST.2017.09.039.