Mühendislik Bilimleri ve Tasarım Dergisi
13(3), 817-828, 2025
e-ISSN: 1308-6693

Journal of Engineering Sciences and Design
DOI: 10.21923/jesd.1653290

Araştırma Makalesi Research Article

TRAJECTORY MODELING FOR HIGH-POWER ROCKETS USING MATLAB/SIMULINK

Mohammad AFSHAR[1], Muhammed Fatih ÇAKIR[1], Meral BAYRAKTAR[1*]

[1]Yıldız Technical University, Faculty of Mechanical Engineering, Department of Mechanical Engineering, İstanbul, Türkiye

Keywords

Rocket,
Trajectory Modelling,
MATLAB/Simulink.

Abstract

In this study, the trajectory of a high-power rocket was obtained from a model created using MATLAB/Simulink. The non-linear aerodynamic and thrust systems and the variable mass due to fuel consumption are the basic components of this model. The model has been validated by the data collected during the launch of a real rocket. In the case of a need to add new subsystems, the model provides easy integration through its modular and flexible structure. The model has been verified by comparing the simulation results for the altitude and orbit of the rocket with the values of real rocket launch data, as well as through the Rocketpy open source trajectory prediction tool. In order to discuss the effect of mass effect on flight performance, different mass scenarios were performed. Additionally, the velocity of the rocket, the forces acting on the rocket during flight, and its position over time from launch to landing are presented.

MATLAB/SIMULINK İLE YÜKSEK GÜÇLÜ ROKETLER İÇİN YÖRÜNGE MODELLEMESİ

Anahtar Kelimeler

Roket,
Yörünge Modellemesi,
MATLAB/Simulink.

Öz

Bu çalışmada yüksek güçlü bir roketin yörüngesi MATLAB/Simulink kullanılarak oluşturulan bir model üzerinden elde edilmiştir. Modelin temel bileşenlerini, doğrusal olmayan aerodinamik ve itki sistemleri ile yakıt tüketimine bağlı değişken kütle oluşturmaktadır. Model gerçek bir roketin fırlatılması esnasında elde edilen verilerle doğrulanmış olup, modüler ve esnek yapısı sayesinde ihtiyaç duyulduğunda ek alt sistemlerin kolaylıkla entegre edilmesine olanak sağlamaktadır. MATLAB/Simulink modelinin ulaşacağı irtifa ve yörünge için gerçekleştirilen simülasyon sonuçları, gerçek bir roketin fırlatılması esnasında ve bunun yanında RocketPy açık kaynak yörünge tahmin aracı vasıtası ile elde edilen değerlerle karşılaştırılarak doğrulanmıştır. Farklı kütle senaryoları içeren simülasyonlar gerçekleştirilerek uçuş performansları incelenmiştir. Bunun yanında roketin hızı, uçuş esnasında rokete etkiyen kuvvetler ve fırlatılmasından inişine kadar geçen süre boyunca zamana bağlı konumu sunulmuştur.

Alıntı / Cite

Afsar, M., Çakır, M.F., Bayraktar, M., (2025). Trajectory Modeling for High-Power Rockets Using Matlab/Simulink, Journal of Engineering Sciences and Design, 13(3), 817-828.

Yazar Kimliği / Author ID (ORCID Number)	Makale Süreci / Article Process	
M. Afsar 0009-0008-9665-0521	**Başvuru Tarihi / Submission Date**	07.03.2025
M.F. Çakır 0000-0002-7459-5014	**Revizyon Tarihi / Revision Date**	14.06.2025
M. Bayraktar 0000-0002-8779-0476	**Kabul Tarihi / Accepted Date**	31.07.2025
	Yayım Tarihi / Published Date	30.09.2025

[*] İlgili yazar / Corresponding author: mbarut@yildiz.edu.tr, +90 0212 383 28 38

TRAJECTORY MODELING FOR HIGH-POWER ROCKETS USING MATLAB/SIMULINK

Mohammad AFSHAR, Muhammed Fatih ÇAKIR, Meral BAYRAKTAR[1†]

[1]Yıldız Technical University, Faculty of Mechanical Engineering, Department of Mechanical Engineering, İstanbul, Türkiye

Highlights

- Model provides a user-friendly platform for rocket trajectory simulation and analysis.
- Simulation results offer insights into rocket dynamics, performance, and safety parameters.
- The modular framework enables customization and expansion for future research and applications.

Purpose and Scope

Precise rocket trajectory forecasting is crucial in amateur rocketry; nonetheless, currently available open-source simulation tools frequently pose difficulties because of their programming demands and restricted customization choices. This paper presents a modular framework designed to overcome these limitations by providing a more user-friendly platform for trajectory analysis.

Design/Methodology/Approach

A six-degree-of-freedom (6DOF) system model was created to mimic rocket flight dynamics, integrating aerodynamic, propulsion, and environmental influences during the flight. The simulation tool, created in MATLAB/Simulink, allows for the visualization of the model's elements without requiring extensive programming knowledge. To verify the model, simulation outcomes were contrasted with experimental flight data, showcasing its efficiency in analyzing rocket dynamics and adapting to various design configurations.

Findings

The model demonstrates significant alignment with experimental flight data in the ascent phase, precisely forecasting apogee and reflecting the non-linear change in mass resulting from propellant use. Furthermore, the simulation highlights the considerable impact of parachute size on the descent path and ultimate landing location, offering an important understanding of recovery system efficiency.

Research Limitations/Implications

Upcoming studies will concentrate on incorporating extra subsystems, like guidance, navigation, and control (GNC), which are frequently limited due to security issues. Additional improvements involve optimizing propulsion models to integrate variable thrust, changes in specific impulse, and nozzle expansion effects, thereby providing a greater understanding of rocket launch dynamics. Broadening validation to various rocket configurations will enhance the model's reliability even more. Furthermore, exploring different landing technologies aside from parachutes may allow for improved control over landing accuracy. Integrating three-dimensional geometric modeling could enhance accuracy by offering a more complete depiction of aerodynamic and structural features.

Practical Implications

The model has practical uses that span various elements of rocket design and performance evaluation. Through the simulation of different flight scenarios, engineers can understand how mass distribution, engine performance, and parachute deployment influence the overall trajectory. These results can guide design choices, resulting in more effective and dependable rocket systems. Additionally, the model aids in evaluating possible safety hazards, including the closeness of landing sites and descent speeds, thus enhancing mission safety.

Originality

The framework's modular, block-oriented design in MATLAB/Simulink makes it user-friendly for those with minimal programming skills. This method aids in verifying parachute deployment algorithms and allows for the assessment of aerodynamic characteristics at various altitudes. Additionally, the model enables the investigation of different deployment algorithms and technologies, expanding its use in amateur and educational rocketry studies.

[†] İlgili yazar / Corresponding author: mbarut@yildiz.edu.tr, +90 0212 383 28 38

1. Introduction

Studying the rocket flight dynamics is very important for rocket safety, reliability, and its mission. The fruitful outputs on this subject enable trajectory control, reaching the rocket to its orbit or destination. Up to now, the researchers have been working on this subject to understand the rocket dynamics, optimize fuel efficiency, structural design, payload capacity, reduce design and operating costs, and enhance mission success. An accurate prediction of rocket flight dynamics is a challenging research area in aerospace engineering since it is very complex to analyze because of its continuously changing mass, forces with time and atmospheric conditions.

The studies on the rocket flight dynamics go back to the 19th century (Moore, 1813). Tsiolkovskiy, 1903 made great contributions to rocket science (Tsiolkovsky, 1903). With the 21st century, the studies on the subject have been accelerated. For example, Ohkami et al., 2004 presented a concept of a horizontal take-off/landing rocket plane with drop tanks, which will take off from the sea using a high-speed assist system. The important parameters obtained from wind tunnel measurements and the basic motion equations, as well as the longitudinal and lateral dynamics and kinematical equations for the model, were reported. Experimental studies are difficult to perform and time-consuming; therefore, numerical and computational methods have been frequently employed. As such, a novel method for a flight control system using non-linear inverse dynamics (NID) was developed by Lane et al., 1988. It was reported that the NID method enables an accurate and robust control of the flight envelope. The conducted numerical studies demonstrate the system's effectiveness in maneuvering, stability and preventing stall. It was concluded that the NID-based controllers enhance safety, performance, and pilot workload in flight. Goh et al., 2008 explored mathematically complex treatments of singular control problems for a deeper understanding of optimal control strategies in aerospace. While powerful for optimization, these control methods are less accessible and harder to integrate into rapid modular builds. A range of approaches and tools has been proposed for rocket and aerospace simulation. Simplified analytical models, such as the two-dimensional (2D) gravity turn analysis by Campos and Gil, 2020, offer foundational insights but lack full 3D rotational detail needed for more sophisticated predictions. Ceotto et al., 2021 provided an open-source, Python-based environment with a user-friendly interface. It was reported that its code-centric configuration may limit accessibility for users without programming backgrounds. Cremaschi et al., 2018 employed a commonly used platform called ASTOS, which offers powerful capabilities for trajectory analysis and optimization. Since ASTOS is a commercial product, its high cost makes it inaccessible to many students and academic institutions. This limitation reduces its suitability for widespread educational use, particularly in undergraduate courses or independent research projects where budget constraints are common. Beyond rocket-specific tools, RotorPy, as reported by Folk et al., 2023 simulates six-degree-of-freedom (6-DOF) multirotor dynamics in Python, demonstrating the benefits of interactive simulation environments. Similarly, AsaPy (Dantas et al., 2023) offers a customizable Python library focused on aerospace simulation post-processing and data-driven analysis, but it does not directly address 6-DOF trajectory modeling. Classic aerospace texts written by Tewari, 2007 and Zipfel, 2014 provide essential theoretical foundations—yet they do not include deployable software tools for hands-on trajectory simulation, limiting their instructional structure.

This study introduces a modular and educationally oriented MATLAB/Simulink-based simulation platform for comprehensive rocket trajectory analysis, designed to bridge the gap between theoretical flight dynamics and accessible simulation tools. Addressing the limitations of existing computational tools that often require advanced programming or lack flexibility, this proposed 6-DOF rocket trajectory model blends theoretical rigor with a visual, modular environment. Its user-friendly, block-based architecture supports easy subdivision of subsystems, parameter tuning, and direct visual feedback, making it ideal for teaching, exploratory research, and rapid prototyping. Unlike "black box" software, this model allows users to directly interact with all subsystems within Simulink, enhancing conceptual understanding and supporting experiential learning, with its validity demonstrated through comparisons with experimental flight data and RocketPy output.

This work aims to provide a didactically valuable tool that enhances conceptual understanding of rocket flight dynamics through simulation, and to contribute a flexible, modular modeling environment that can support both education and early-stage design efforts. Validation of the model is conducted through comparison with experimental flight data and existing simulation outputs from RocketPy, demonstrating its reliability and educational value.

2. Material and Method

The mathematical section details the mathematical framework of the 6-DOF rocket trajectory simulation developed in MATLAB/Simulink. The model shows the key forces acting on the rocket (aerodynamic, propulsive, and gravitational) and their influence on both translational and rotational motion. The present section presents a short overview of the developed mathematical model, aerodynamic forces and moments, properties of the atmosphere and details of the simulation.

2.1. Mathematical Model

The translational equations of motion for an aerospace vehicle are derived directly from Newton's Second Law of Motion. As shown in Eq. 1 the translational equations can be expressed as (Zipfel, 2014):

$$m^B\left[\frac{dv_B^I}{dt}\right]^I = [\bar{T}]^{BI}\left[f_{a,p}\right]^B + m^B[\bar{T}]^{GI}[g]^B \tag{1}$$

Where $\boldsymbol{m^B}$ is the mass of the vehicle, v_B^I is the linear velocity of the center of mass relative to the inertial frame. The $\mathbf{F}_p$, and $\mathbf{F}_a$, represent the external forces, including the propulsive force and the gravitational force ($m^B g^B$), respectively. The propulsive forces and aerodynamic forces are typically expressed in body-fixed coordinates, while the gravitational force is provided in geographic coordinates. To reconcile these coordinate systems, two transformation matrices are employed, enabling the formulation of the differential equations required for simulation in MATLAB. Rocket propulsion involves continuous ejection of fuel particles, resulting in a variable mass system (Zipfel, 2014).

$$F = \dot{m}v_e = I_{sp}\dot{m}g_0 \tag{2}$$

where I_{sp}, m, v_e are specific impulse, structural mass and velocity, respectively. Taking the time derivative of this equation and assuming v_e is constant, yields the instantaneous rate of change of velocity due to mass variation. This formulation underpins the variable-mass dynamics used in the 6-DOF simulation. For many aerospace simulations, the Flat-Earth Approximation is adopted (Zipfel, 2014). This simplifies a planar representation of the Earth's surface, unwrapping the longitude-latitude grid into a Cartesian plane. The Earth-fixed frame is treated as inertial, and Newton's laws are applied accordingly. While this assumption is common in aircraft and missile simulations, it simplifies the dynamics for computational efficiency without significant loss of accuracy for short-range trajectories.

$$\left[\frac{dv_B^E}{dt}\right]^B = -[\Omega^{BE}]^B[v_B^E]^B + \frac{1}{m}\left[f_{a,p}\right]^B + [T]^{BL}[g]^L \tag{3}$$

The Custom Variable Mass 6-DOF (Quaternion) block models the rotational and translational dynamics using quaternion representation. The quaternion approach (Eqs. 4-5) avoids singularities associated with Euler angles, providing a robust framework for handling large angular displacements. The differential equations for the quaternion components are integrated as:

$$\begin{bmatrix}\dot{q}_0\\ \dot{q}_1\\ \dot{q}_2\\ \dot{q}_3\end{bmatrix} = \frac{1}{2} * \begin{bmatrix}0 & -p & -q & -r\\ p & 0 & r & -q\\ q & -r & 0 & p\\ r & q & -p & 0\end{bmatrix}\begin{bmatrix}q_0\\ q_1\\ q_2\\ q_3\end{bmatrix} + K\varepsilon\begin{bmatrix}q_0\\ q_1\\ q_2\\ q_3\end{bmatrix} \tag{4}$$

$$\varepsilon = 1 - (q_0^2 + q_1^2 + q_2^2 + q_3^2) \tag{5}$$

The mathematical equations of the system have been solved by forming a Simulink model in MATLAB, as detailed in Simulation Process section.

2.1.1 Aerodynamic Forces and Moments

The aerodynamic forces and moments are among the most challenging aspects of atmospheric flight mechanics to model accurately. Their dependence on the state variables, such as vehicle shape, size, velocity, and atmospheric properties, is typically expressed through empirical or semi-empirical models. The Aerodynamic block computes the aerodynamic forces and moments of CG (center of gravity). The center of gravity and the center of pressure are assumed to be in body axes. While this block can output forces and moments in the stability axes, the blocks in the equations of motion library of MATLAB/Simulink are currently designed to accept forces and moments in either the body or wind axes only (Stevens, 1992). Let the angle of attack be α and ß, the sideslip so to yield the net rotation from body to wind axes:

$$C_{w\leftarrow b} = \begin{bmatrix} cos\alpha cos\beta & sin\beta & sin\alpha cos\beta \\ -cos\alpha sin\beta & cos\beta & -sin\alpha sin\beta \\ -sin\alpha & 0 & vos\alpha \end{bmatrix} \tag{6}$$

2.1.2 Atmospheric properties

The atmosphere's physical properties, such as temperature, pressure and density are of paramount importance in the design and analysis of aerospace vehicle trajectories. Temperature influences the density and the density determines the aerodynamic force components acting on a vehicle. Temperature also governs the speed of sound, influencing Mach number and aerodynamic heating. The Earth's atmospheric temperature profile varies with altitude, influencing vehicle performance. Pressure modulates the thrust generated by rocket engines, particularly during ascent through varying altitudes. The atmosphere is divided into several distinct layers, each with unique characteristics. The troposphere (0–11 km) can be characterized by a decreasing temperature gradient with altitude. Temperature and pressure gradient can be calculated with the following equations (Eq.7-8) concerning altitude (Zipfel, 2014).

$$T = 288.15 - 0.0065H \tag{7}$$

$$P = 101325\left(\frac{T}{288.15}\right)^{5.2559} \tag{8}$$

2.1.3 Gravitational force

Gravity, a fundamental force, remains a topic of philosophical curiosity and scientific study. From an engineering perspective, Isaac Newton's Law of Universal Gravitation provides a precise mathematical framework for calculating gravitational effects. The gravitational force between two masses separated by a distance is given by:

$$F = G\frac{m_1 m_2}{r^2} \tag{9}$$

Aerodynamic moments and forces are modeled for aeroballistics axes and then converted to body axes. Propulsive forces are modeled in body axes. Figure 1 summarizes the simulation approach in a flowchart where the flow between each equation and subsystem can be seen.

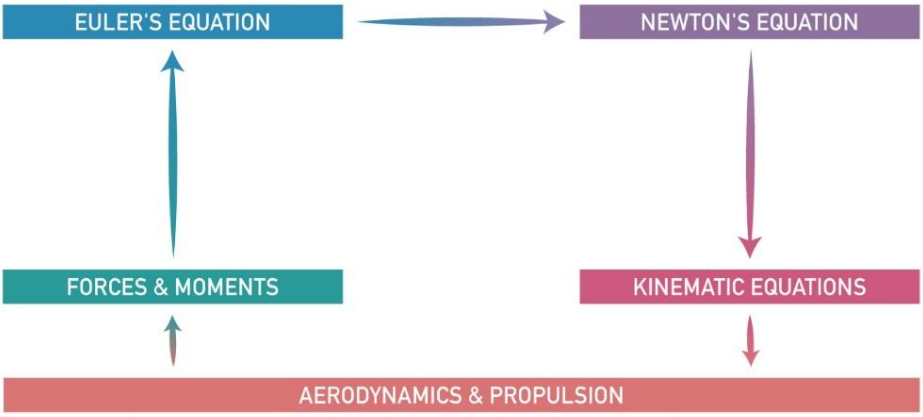


Figure 1. Flow Chart of the Model (Adapted from Zipfel, 2014)

2.1.4 Simulation Process

All simulations are performed with MATLAB/Simulink to analyze the dynamic behavior of the physical system. MATLAB/Simulink is chosen because it enables the fast, modular, and flexible construction of complex systems owing to its visual modeling capabilities. This is particularly advantageous for systems like rockets, which involve multiple components and time-varying parameters. The use of Simulink not only accelerates the model development process but also simplifies validation procedures. Based on the flowchart presented in Figure 1, the corresponding Simulink model is also illustrated in Figure 2.

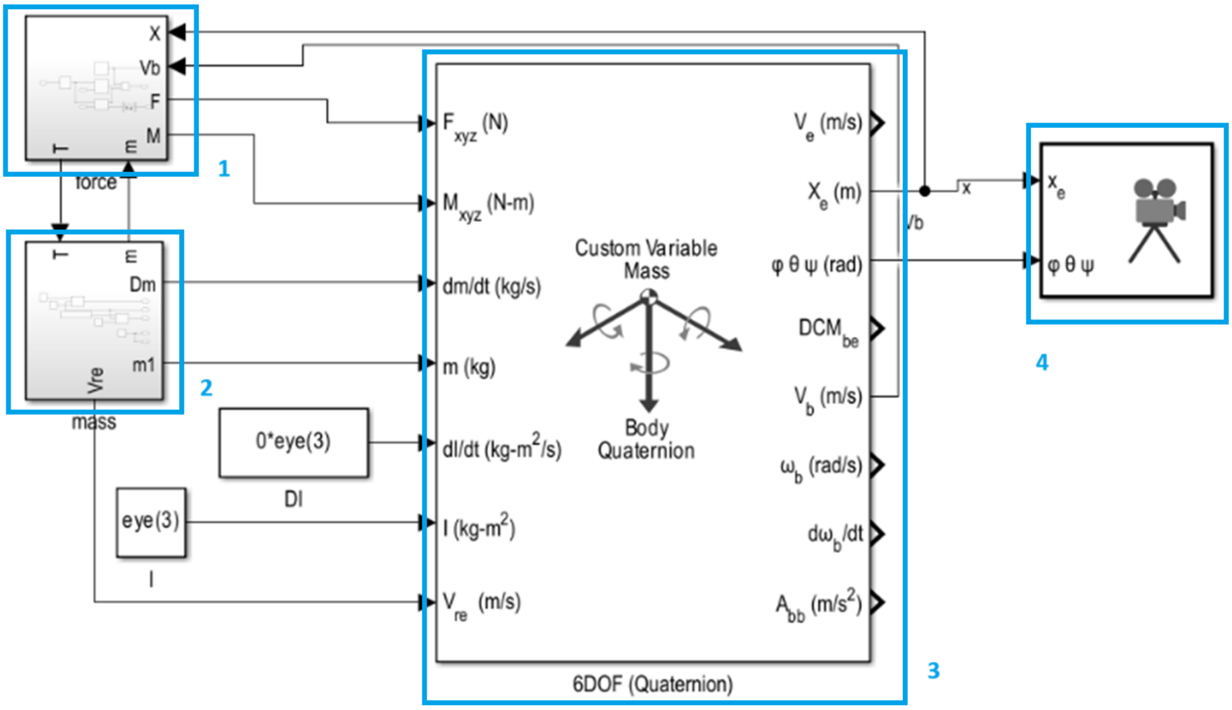


Figure 2. 6-DOF MATLAB/Simulink Model of Rocket Trajectory (For explanations, please see Table 1).

During the model development, simplicity and clarity were prioritized. This structure enables end users to model and test complex systems such as rocket dynamics in a practical and systematic manner. The model's main components and simulation critical elements are described in Table 1.

Table 1. Explanation of Blocks in MATLAB/Simulink Model

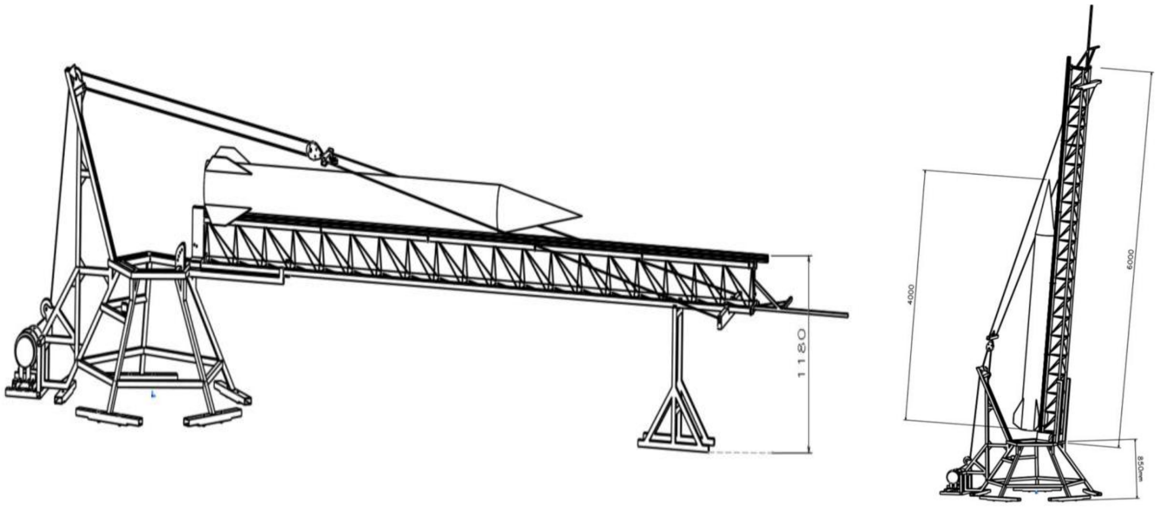
Block no	Block name	Description of the block
1	Force and moment calculation	Computes the total force and moment acting on the rocket by combining thrust, gravitational force, and aerodynamic effects
2	Change in Mass	Represents the time-varying mass of the rocket due to propellant consumption. Includes the exhaust direction and mass flow rate
3	6DOF (Quaternion)	Solves the six degrees of freedom motion of the rocket. Calculates position, velocity, orientation, and angular motion based on force and moment inputs
4	Graphical display	Presents the simulation outputs visually, enabling observation of the system's dynamic behavior.

3. Field Studies and Validation of the Model

The credibility of the present numerical study is presented by comparing the outputs of the 6-DOF trajectory model generated by RocketPy with the field measurements. The primary objective of this comparison is to validate the accuracy and robustness of the developed model.

3.1 Launch of the rocket

Field measurements were done during a test conducted as a part of a rocketry competition called TEKNOFEST in Türkiye in 2023. The rocket was launched from a test rig provided by the organizers (Figure 3). It must be noted that due to the nature of the competition, which involved a single, integrated launch of the complete rocket system, a traditional, iterative experimental setup with controlled variations in parameters was not feasible.



(a) **(b)**

Figure 3. Launch Platform **a)** Horizontal Position, **b)** Vertical Position, (Teknofest 2023)

The rocket's specifications, including dimensions, mass, and engine characteristics, are detailed in Table 2. This single flight provided a comprehensive dataset of the rocket's flight profile, which served as the baseline for model validation.

Table 2. Rocket Specifications

Rocket specifications	Details
Diameter (m)	0.15
Length (m)	2.5
Motor	M2020
Stability margin	2
Incline velocity (m/s)	20

3.2 Model Validation and Comparison with RocketPy

The performance of the developed 6-DOF MATLAB model was rigorously evaluated by comparing its predictions with both data obtained as a result of the launch and the results from RocketPy, a widely used open-source rocket trajectory simulation tool. As shown in Figure 4, the 6-DOF MATLAB model exhibited significantly closer agreement with the experimental data, particularly in predicting altitude and capturing the nuances of the flight dynamics, than RocketPy. The RocketPy model achieves a smooth descent in about 220 seconds, starting from the highest initial altitude of about 3000 m. The MATLAB model simulation reached peak height around 2900 m whilst the experimental data follow a similar pattern, but starting from a slightly lower level, reaching a peak at approximately 2850 m. They both completed their descent within 215 seconds, showing fluctuations that represent variability thought to be due to real flight conditions. In general, whilst high orbit predictions were made by RocketPy, the MATLAB model and experimental data provided closer results.

This improved accuracy can be attributed to the model's ability to incorporate customizable mass profiles and detailed aerodynamic properties, providing a more faithful representation of the physical system. This validation underscores the precision and reliability of the developed 6-DOF MATLAB model for practical aerospace trajectory simulations.

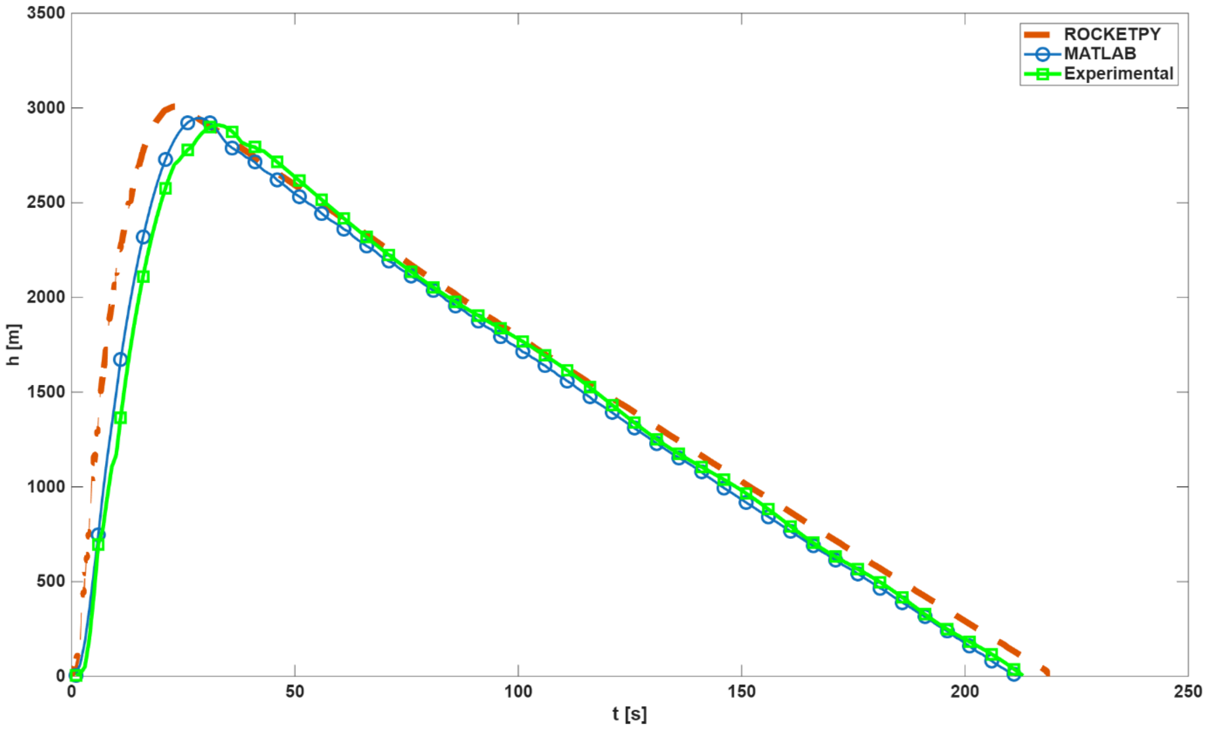


Figure 4. Time-altitude comparison between simulation, RocketPy, and experimental data

3.3 Simulation Scenarios

To demonstrate the versatility of the simulation framework, four distinct scenarios (as given in Table 3) were simulated in the study. In the first scenario, the rocket's mass remained constant throughout the flight. This served as a baseline case, illustrating an idealized ballistic trajectory in the absence of mass variation. In the second scenario, a linear decrease in mass over time was simulated to approximate a simplified model of propellant consumption. Compared to the constant-mass case, this resulted in a slightly lower apogee due to the decreasing thrust-to-weight ratio. A third scenario was examined in which a custom mass reduction profile derived from the solid rocket motor's burn rate curve provided by the manufacturer was applied. This more realistic mass profile led to a further improvement in apogee compared to the linear mass decrement scenario, highlighting the importance of accurate mass modeling. The final scenario combined the custom mass reduction profile with a parachute deployment event at apogee. This simulation captured the transition from ballistic ascent to controlled descent, demonstrating the model's ability to replicate complex descent dynamics.

Table 3. The Criterion That Differentiates the Scenarios.

Scenarios	Criteria
1st scenario	Constant mass
2nd scenario	Linear mass decrement
3rd scenario	Custom mass decrement
4th scenario	Custom mass decrement with parachute deployment

The altitude profiles for all four scenarios are shown in Figure 5, clearly illustrating the effects of mass variation and parachute deployment on the flight trajectory. From the figure, the apogee altitudes are estimated as 4950 m, 4550 m, 4150 m, and 2850 m for the first, second, third, and fourth scenarios, respectively. As expected, the highest apogee is observed in the constant-mass scenario.

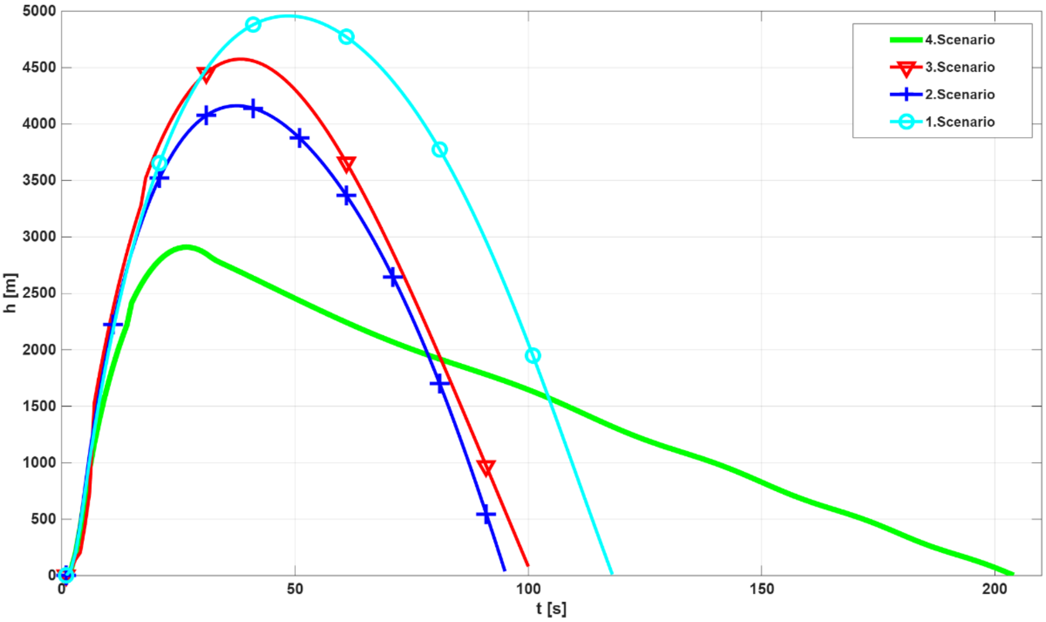


Figure 5. The Altitude Profiles for Four Scenarios

4. Result and Discussion

The developed 6-DOF trajectory simulation model successfully demonstrates its ability to accurately predict the rocket's flight dynamics across various simulated scenarios. Its flexibility in handling customizable mass reduction profiles and incorporating complex descent dynamics, such as parachute deployment, significantly enhances its applicability to a wide range of mission designs. Validation against experimental flight data further confirms the model's reliability, establishing its potential as a powerful tool for trajectory analysis and optimization.

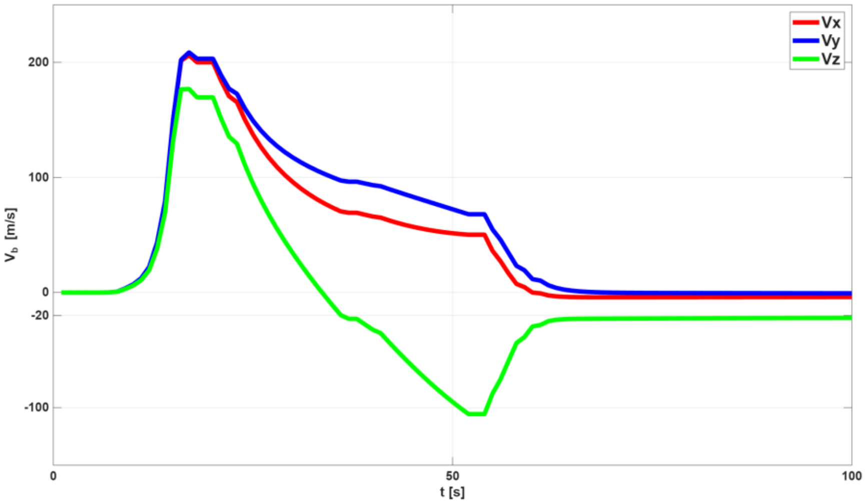


Figure 6. Velocity of the Rocket Concerning Body Frame

Figure 6 provides a detailed analysis of the rocket's velocity components in the body frame. During ascent, the velocities along all three axes (X, Y, and Z) increase rapidly, corresponding to the powered flight phase. After engine burnout, the Z-velocity (vertical) reaches its peak at apogee and then decreases under the influence of gravity. Parachute deployment is reflected in the velocity profiles: the X and Y velocities rapidly converge toward zero, indicating minimal horizontal drift during descent, while the Z-velocity stabilizes at a constant negative value, representing a controlled descent rate. This steady-state Z-velocity highlights the effectiveness of the parachute in achieving a stable and controlled descent.

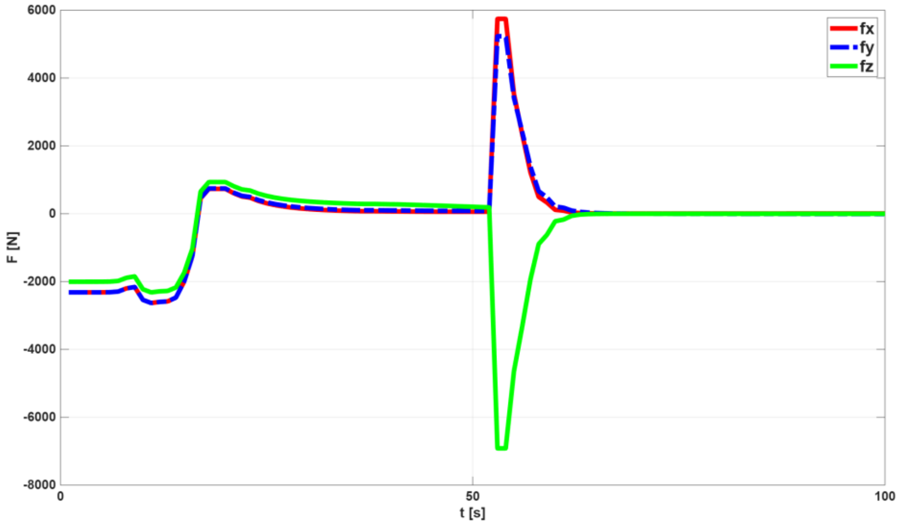


Figure 7.Forces acting on Rocket during Flight in all Dimensions

Figure 7 illustrates the key force dynamics acting on the rocket throughout its flight. The initial phase is marked by a rapid increase in net force due to engine ignition and thrust generation. After engine burnout, the net force decreases as thrust ceases, marking the transition to ballistic flight. A sharp spike in drag force indicates parachute deployment, clearly demonstrating the model's ability to accurately capture the dynamic forces acting on the rocket during distinct flight phases.

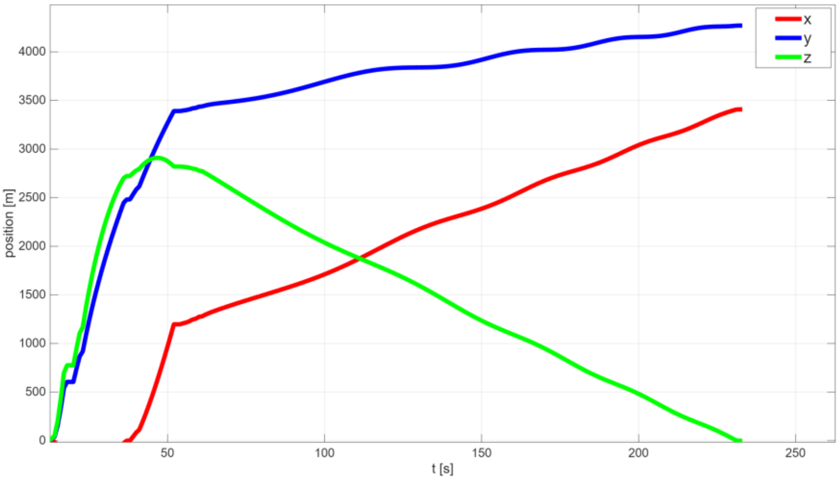


Figure 8. Position of the Rocket from the Launching Site Concerning Time

Figure 8 depicts the rocket's trajectory relative to the launch site. The Z-position (altitude) increases rapidly during the powered ascent phase, reaching a maximum at apogee, and then gradually decreases during the descent phase. The X and Y positions, representing lateral displacement, show minimal deviation during ascent. Importantly, any lateral drift is effectively arrested after parachute deployment, with the X and Y positions stabilizing near zero. This confirms the model's accuracy in simulating a controlled descent with minimal horizontal movement, further validating its ability to predict the rocket's final landing location. This is a valuable result for field researchers, as it helps estimate the maximum distance the rocket may travel after launches, especially important when only a single launch is possible.

4. Conclusion

The results of the proposed 6-DOF rocket trajectory model align well with expected flight behavior, confirming the model's accuracy and reliability. Its performance was validated against experimental flight data, and the simulation successfully captured non-linear changes in mass due to propellant consumption, as well as the dynamic effects of parachute deployment. The model offers a clear advantage in terms of transparency and modularity. Unlike RocketPy, which requires scripting in Python, the Simulink-based structure allows users to interact with individual components through a visual, block-oriented interface. This makes it particularly useful for educational environments and early-stage design work. Compared to commercial tools like ASTOS, which offer advanced optimization features but are expensive and not readily accessible to students, this model is free, customizable, and highly adaptable. One of the key strengths of this model is its flexibility. All parameters, such as thrust curves, drag coefficients, and parachute deployment triggers, can be adjusted independently. This enables engineers and researchers to simulate various mission conditions and assess the impact of each variable on the flight path. The ability to modify both ascent and descent phases also supports broader mission planning and failure mode analysis. However, like most simulation models, some limitations exist. The current implementation assumes ideal environmental conditions (e.g., no wind or turbulence), and actuator dynamics are not yet included. Future work may integrate environmental models, sensor feedback, or hardware-in-the-loop testing. A graphical user interface (GUI) could also be developed to further lower the barrier to use for students and non-specialists. In summary, the validated 6DOF MATLAB Simulink model presented in this study offers a powerful, flexible, and accessible platform for rocket trajectory simulation. Its educational utility, combined with its technical rigor, makes it a valuable tool for both academic and professional applications.

Acknowledgement

This work was supported by Yildiz Technical University Machine Technologies Club Yildiz Rocket Team.

Conflict of Interest

The authors declared no conflict of interest.

References

Abba, S., 2018. Modeling Rocket Flight Trajectory in Workshop: Teaching Computation in the Sciences Using MATLAB, Carleton College, USA. https://www. researchgate.net/publication/347711217_Modeling_Rocket_Flight_Trajectory.

Aksen, U., Aslan, A. R., Göker, Ü.D., 2024. Development of a Sensitivity Analysis Tool for the Trajectory of Multistage Launch Vehicles. Turkish Journal of Engineering, 8 (2), 254-264.

Barale, M., 2024. Precise Trajectory Calculation for Launchers: A MATLAB–Simulink Modeling Approach.

Campbell, T.A., Seufert, S.T., Reis, R.C., Brewer, J.C., Limberger T.R., Whelan, C. E., Okutsu, M., 2016. Model Rocket Projects for Aerospace Engineering Course: Simulation of Flight Trajectories. In 54th AIAA Aerospace Sciences Meeting, page 1577.

Campos, L.M.P.C., Gil, P.J.S., 2020. The Two-Point Boundary-Value Problem for Rocket Trajectories. Aerospace, 7(9). https://doi.org/10.3390/aerospace7090131.

Ceotto, G.H., Schmitt, R.N., Alves, G.F., Pezente, L.A., Carmo, B.S., 2021. RocketPy: Six degree-of-freedom Rocket Trajectory Simulator. Journal of Aerospace Engineering, 34(6), 04021093.

Ceotto, J.C.F., dos Santos Brandão, P.S., Marques, G.A., Lapa, C.M.F., 2021. RocketPy: Next-generation Trajectory Simulation for High-power Rockets.

Chowdhury, S., Pitot de La Beaujardiere, J., Brooks, M., Roberts, L., 2011. An Integrated Six degree-of-freedom Trajectory Simulator for Hybrid Sounding Rockets. In 49th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition.

Cremaschi, F., Weikert, S., Schäff, S., Wiegand, A., 2018. ASTOS, A reconfigurable Software for Design of Mega Constellations, Operation of Flying Laptop and End-of-life Disposal. 15th International Conference on Space Operations.

Dantas, J.P.A., Silva, S.R., Gomes, V.C.F., Costa, F.S.V., Teixeira, C.A., (2024). AsaPy: A Python Library for Aerospace Simulation Analysis.

Do Nascimento Júnior, D.S.P., Gurgel, J.F., Reis, M., Wehmann, C.F., 2022. Development of a Model Rocket Trajectory Simulation Tool with Python. Revista Brasileira de Física Tecnológica Aplicada, 9(2).

Folk, S., Paulos, J., Kumar, V., 2023. RotorPy: A Python-based Multirotor Simulator with Aerodynamics for Education and Research.

Goh, B.S., 2008. Optimal Singular Rocket and Aircraft Trajectories. Chinese Control and Decision Conference, 1531-1536.

Hergert, J., Brock, J., Stern, E., Wilder, M.C., Bogdanoff, D.W., 2017. Free-Flight Trajectory Simulation of the ADEPT Sounding Rocket Test Using US3D. 35th AIAA Applied Aerodynamics Conference.

Lane, S.H., Stengel, R.F., 1988. Flight Control Design Using Non-linear Inverse Dynamics, Automatica, 24(4), 471-483, https://doi.org/10.1016/0005-1098(88)90092-1.

Majstrenko, O.V., Prokopenko, V.V., Makeiev, V.I., Ivanyk, E.G., 2020. Analytical Methods of Calculation of Powered and Passive Trajectory of Reactive and Rocket-assisted Projectiles.

MBC Campos, L., Gil, P.J., 2020. The Two-point Boundary-value Problem for Rocket Trajectories. Aerospace, 7(9), 131.

Moore, W., 1813. A Treatise on the Motion of Rockets, to which is added An Essay on Naval Gunnery and Theory and Practice, G. and S. Robinson, London.

Ohkami, Y., Ogasawara, Y., Ogawa, M., Itoga, C., Sakamoto, T., Nagao, Y., 2004. A Proof-of-Concept Approach for Horizontal Take-Off/Landing Rocket Plane, IFAC Proceedings, 37 (6), 1025-1030, https://doi.org/10.1016/S1474-6670(17)32314-5.

Ożóg, R., Jacewicz, M., Głębocki, R., 2020. Modified Trajectory Tracking Guidance for Artillery Rocket. Journal of Theoretical and Applied Mechanics, 58(3).

Stevens, B. L., Lewis, E L., 1992. Aircraft Control and Simulation, John Wiley & Sons, New York.

Tewari, A. 2007. Atmospheric and space flight dynamics. Birkhũser Boston.

Tsiolkovsky, K. E., 1903. Exploration of Outer Space by Means of Rocket Devices (A. N. Trans., Trans.) [Original work published in Russian].

Wang, Y., Xu, M., An, X., Xu, Z., Xu, W., Quan, E., 2021. Numerical Study on the Trajectory of a Long-range Flexible Rocket with Large Slenderness Ratio. Aerospace Science and Technology, 117, 106959.

Zipfel, P.H., 2014. Modeling and Simulation of Aerospace Vehicle Dynamics. 10.2514/4.862182.

https://cdn.teknofest.org/media/upload/userFormUpload/EK2F%C4%B1rlatma_Rampas%C4%B1_%C3%96zellikler_gWPxQ.pdf

TEKNOFEST, 2023: https://www.teknofest.org/tr/yarismalar/roket-yarismasi/