Metallic Braking and GUI Development for Dynamic Evaluation

Internship Report

Submitted to:

Terminal Ballistics Research Laboratory (TBRL)

Defense Research Development Organization (DRDO)

Under the guidance of

Mr. Subhash Chander (Sc 'F') and Mr. Rohit Khanna (Sc 'C')

RTRS, TBRL

Submitted by:

VIJAY (Roll No. 12116018)

Department of Mechanical Engineering

National Institute of Technology, Kurukshetra

Jan 2024 – June 2024

DECLARATION

I declare that this project report titled "**Metallic Braking and GUI Development for Dynamic Evaluation**" submitted in partial fulfillment of the degree of **B. Tech in Mechanical Engineering** at National Institute of Technology, Kurukshetra is a record of original work carried out by me under the supervision of Mr. Rohit Khanna (Sc-C) and has not formed the basis for the award of any other degree or diploma, in this or any other Institution or University In keeping with the ethical practice in reporting scientific information, due acknowledgements have been made wherever the findings of others have been cited.

To the best of my knowledge, this project report has not been submitted, either partially or in its entirety, elsewhere in any other university or institution for the award of an internship certificate.

Date: June 2024 Vijay

CERTIFICATE

This is to certify that Vijay, a student of B. Tech in Mechanical Engineering from the National Institute of Technology, Kurukshetra, has successfully completed the project titled "Metallic Braking and GUI Development for Dynamic Evaluation" at the Rail Track Rocket Sled (RTRS) facility, Terminal Ballistics Research Laboratory (TBRL), Defense Research and Development Organization (DRDO).

Throughout the project, Vijay demonstrated technical competence, and problem-solving abilities. His commitment to excellence and innovation significantly contributed to the project's success. The project involved developing metallic braking systems and designing a Graphical User Interface (GUI) for dynamic testing on the Rail Track Rocket Sled (RTRS).

Rohit Khanna (Scientist 'C')
RTRS, TBRL

Subhash Chander (Scientist 'F')
RTRS, TBRL

Acknowledgement

I wish to extend my sincere appreciation to Dr. M Raghavendra Rao, Director, TBRL, Chandigarh, for granting me the opportunity to undertake my internship at TBRL.

My heartfelt gratitude goes to my supervisor, Mr. Sandeep Malik (Scientist 'G’, Joint Director, RTRS) and Mr. Subhash Chander (Scientist 'F'), and my mentor Mr. Rohit Khanna (Scientist 'C'), for their unwavering support, invaluable guidance, and timely assistance throughout my internship. Despite their demanding schedules and departmental commitments, they generously dedicated their time to provide me with direction and knowledge, ensuring my progress on the right trajectory.

Additionally, I am thankful to the staff of the RTRS department at TBRL for their assistance.

Furthermore, I express my profound gratitude to Professor Satnam Singh of the Mechanical Engineering Department at the National Institute of Technology, Kurukshetra, for his significant contributions to the completion of my project titled "Metallic Braking and GUI Development for Dynamic Evaluation."

Lastly, I would like to affirm that this project was completed solely by myself.

Vijay

ABSTRACT

The "Metallic Braking" project aimed to enhance braking efficacy through wedge-based frictional mechanisms, augmenting normal reaction to the ground for increased frictional force. Analysis focused on the forces acting on the wedges, utilizing various calculations. Challenges arose in reconciling ideal dynamic evaluation conditions with real-world scenarios, potentially compromising braking effectiveness. Initial results revealed suboptimal braking outcomes compared to expectations.

The "GUI Development for Dynamic Testing and Evaluation" project focused on creating a MATLAB-based interface to generate output including maximum velocity curves, acceleration, deceleration, and loading profiles. The GUI was developed using MATLAB with assistance from reference materials and online resources. Challenges included implementing multi-window functionalities and the absence of completed projects for testing. However, initial results indicate satisfactory performance, though comprehensive testing is pending.

Methodologies encompassed thorough analysis of forces on the wedges for the metallic braking project, while the GUI development project relied on MATLAB programming techniques. Ideal conditions for dynamic evaluation were considered in both projects, though disparities with real-world scenarios introduced complexities. Future efforts in both projects will focus on refining designs and conducting comprehensive validation tests to ensure improved performance under diverse operating conditions.

LIST OF FIGURES

LIST OF CURVES

TABLE OF CONTENTS

CHAPTER-1

INTRODUCTION

1.1 About the Organization

Development and Defense Research Organization (DRDO)

Fig 1.1: DRDO-LOGO[1]

The Defence Research and Development Organisation (DRDO) is the principal agency within the Ministry of Defence of the Government of India, entrusted with military research and development, and is based in Delhi, India. The Defence Science Organisation was founded in 1958 by the amalgamation of the Technical Development Establishment and the Directorate of Technical Development and Production of the Indian Ordnance Factories. Following that, in 1979, the Defence Research and Development Service (DRDS) was established as a service of Group 'A' Officers / Scientists directly under the administrative jurisdiction of the Ministry of Defence.

The Defence Research and Development Organisation (DRDO) is the R&D wing of the Ministry of Defence, Government of India, with a vision to empower India with cutting-edge defense technologies and a mission to achieve self-reliance in critical defence technologies and systems, while equipping our armed forces with state-of-the-art weapon systems and equipment in accordance with requirements laid out by the three Services. The DRDO's pursuit of self-reliance and successful indigenous development and production of strategic systems and platforms such as Tejas light combat aircraft have given India's military in research and technology, particularly in military technologies.

HISTORY

The Defence Research and Development Organisation (DRDO) was formed in 1958 by the merger of the Indian Army's existing Technical Development Establishments (TDEs) and the Directorate of Technical Development & Production (DTDP) with the Defence Science Organisation (DSO). DRDO was a modest organisation with only ten institutions or laboratories at the time. It has expanded in a variety of ways over the years, including the number of laboratories, achievements, and prominence.

In the 1960s, the DRDO launched initiative Indigo, its first major initiative in surface-to-air missiles (SAM). Indigo was terminated in subsequent years after failing to achieve full success. In the 1970s, Project Indigo and Project Devil collaborated to build short-range SAM and ICBMs. The Prithvi missile was later developed as part of the Integrated Guided Missile Development Programme (IGMDP) in the 1980s. Between the early 1980s and 2007, the Indian Ministry of Defence funded the development of a widerange of missiles, including the Agni missile, Prithvi ballistic missile, Akash missile, Trishul missile, and Nag Missile.

In the 1960s, the DRDO launched initiative Indigo, its first major initiative in surface-to-air missiles (SAM). Indigo was terminated in subsequent years after failing to achieve full success. In the 1970s, Project Indigo and Project Devil collaborated to build short-range SAM and ICBMs. The Prithvi missile was later developed as part of the Integrated Guided Missile Development Programme (IGMDP) in the 1980s. Between the early 1980s and 2007, the Indian Ministry of Defence funded the development of a widerange of missiles, including the Agni missile, Prithvi ballistic missile, Akash missile, Trishul missile, and Nag Missile.

VISION

Providing the country with cutting-edge indigenous defense and security technology and systems. The DRDO is determined to make the country strong and self-sufficient in research and technology, particularly in military technologies.

MISSION

1. Design and develop cutting-edge sensors, weapon systems, platforms, and allied equipment for the land, air, sea, space, and cyber domains of defense and security.

2. Facilitate the development and implementation of Systems and Technologiesdeveloped within the Department's R&D ecosystem.

3. Offer technology solutions to the Services in order to improve combat effectiveness.

4. Collaborate to nurture and strengthen defense R&D competence in Indian industry, science and technology (S&T) institutions, and academia.

5. Infrastructure and test and evaluation facilities development; design certification; skill development and human resource strengthening.[1]

Terminal Ballistics Research Laboratory (TBRL)

TBRL is a vital DRDO lab located in Chandigarh that is actively involved in the development, production, processing, and characterization of various high explosive compositions, blast, lethality, and fragmentation studies of warheads, shells, and other ammunitions, captive flight testing of bombs, missiles, and airborne systems, ballistics evaluation of various protective systems such as body armor, vehicle armor, and helmets against small arm ammunition, and so on.

Standardization Testing and Quality Certification Services (STQC), Department of Information Technology (DIT), Government of India, has accredited the laboratory to the International Quality Management Systems Standard ISO 9001:2000. In 2014, it was upgraded to ISO 9001:2008.

HISTORICAL BACKGROUND

Following the end of World War II, new generation weapons and warheads were introduced into military forces around the world. To keep up with these new technological breakthroughs, there was a great need to establish an indigenous foundation for establishing specialized instrumented facilities and range technologies required for the evolution of data required for the design, development, and assessment of warheads and weapon systems.

Terminal Ballistics Research Laboratory, Chandigarh was established in 1961 as one of the Department of Defence Research & Development armament research laboratories, with the goal of providing facilities for applied research and technology development in the fields of high explosives processing, detonics and shock dynamics, blast & damage, immunity, lethality & fragmentation, defeat of armour, and performance evaluation of warheads & other armament systems.

The Terminal Ballistics Research Laboratory (TBRL) was envisioned in 1961 as one of the Department of Defence Research & Development's new weaponry research institutes.

The laboratory became fully operational in 1967 and was publicly launched by the then-Defense Minister in January 1968. While the main laboratory is located in Chandigarh, the shooting range, which spans 5000 acres, is located 22 kilometres away in Ramgarh, Haryana.

VISION

TBRL envisions self-reliance in the development of warhead technologies and provides cutting-edge diagnostics tools for assessing the terminal effects of armaments systems.

AREA OF WORK

- Armour performance against bullets and immunity profiles.
- Ground shock, blast damage, fragmentation, and lethality research.
- Making safety templates for various weapons.
- Underwater detonics and pressure wave propagation research.
- Forming, cladding, and welding with explosives.
- Different high explosive compositions are developed, manufactured, processed, and characterised.
- Impact and penetration experiments, as well as material characterization at high strain rates.
- Explosive driven magnetic flux compression technology for producing high energy electrical pulse power.
- Bombs, missiles, and airborne systems are subjected to captive flight testing.

- Ballistics testing of various protective systems against small arm ammunition, such as body armour, vehicle armour, and helmets.

MISSION

- Create warhead-related technologies and products.
- Provides cutting-edge diagnostics for testing and evaluating weapons systems
- Create a strong technology basis in the country for arms by providing the necessary infrastructure and devoted quality workforce.
- Technologies
- High energy electrical pulse power generation using explosively induced magnetic flux compression
- Material dynamic shock compression and detonation wave shaping
- Shaped charges and EFP technology are being used to defeat armoured and naval targets.
- Small-arm shooting ranges with baffles
- Small-arms ammunition for low-intensity warfare
- Electronic Fuzzes Enabling Technologies [2]

Rail Track Rocket Sled (RTRS)

Historical Background and Establishment:

The Rail Track Rocket Sled (RTRS) facility has its roots in the defense research and development initiatives of India. Established in 1988 under the auspices of the Defence Research and Development Organisation (DRDO), the facility represents a significant milestone in the country's pursuit of advanced testing capabilities in the aerospace and defense sectors.

The decision to establish the RTRS facility was driven by the growing need for a dedicated testing infrastructure to support the development and evaluation of armament, missile systems, aerospace technologies, and other defense-related projects. As India sought to enhance its indigenous defense capabilities, the establishment of such a facility was deemed crucial for conducting rigorous testing and validation of critical defense systems.

Located at the Terminal Ballistic Research Lab (TBRL) in Haryana, India, the RTRS facility was strategically positioned to benefit from the expertise and resources available within the DRDO network. Over the years, the facility has played a pivotal role in supporting various defense programs and initiatives, contributing to the advancement of national security objectives.

With a rich historical background rooted in India's defense research and development efforts, the RTRS facility continues to serve as a cornerstone of the country's defense testing infrastructure, driving innovation and excellence in the field of aerospace and defense technology.

VISION

To be a leading global provider of advanced rocket sled based high-speed rail track test facilities, facilitating cutting-edge research and development in aerospace, armament, and ballistic systems.

MISSION

Our mission is to lead the forefront of innovation and excellence in high-speed sled track testing techniques, pushing the boundaries of accuracy and versatility to meet the dynamic and evolving demands of development and research within the aerospace and defense sectors. We are committed to providing state-of-the-art facilities and unparalleled expertise, ensuring that every aspect of our testing processes is optimized for precision and reliability. From advanced aircraft munitions to cutting-edge hypersonic vehicles, our capabilities are tailored to support the diverse and complex needs of our clients.

Fig 1.2: Rail Track Rocket Sled [3]

Rail Track Specifications, Noteworthy Features, and Achievements

Specifications:

- The Rail Track Rocket Sled (RTRS) facility features a supersonic penta-rail supporting five rail lines.
- The track is precision-aligned to ensure accurate testing conditions and withstand high loads.
- Over the years, the track length has been extended to 4 kilometres.
- Multiple lines of wider gauges are available on the track to accommodate various payloads.

Special Points:

Precision Alignment: The track is precision-aligned to ensure accurate testing conditions, contributing to the reliability and effectiveness of testing operations.

Extended Track Length: The extension of the track length to 4 kilometres provides ample space for testing operations and accommodates various payloads effectively.

Supersonic Penta-Rail: The supersonic penta-rail, supporting five rail lines, enhances the durability and reliability of the facility, ensuring safe and efficient testing operation

Fig 1.3: Penta-Rail Track [3]

Achievements:

Conducting Numerous Trials: RTRS has conducted a significant number of trials, encompassing both recovery and non-recovery scenarios, for a diverse array of payloads.

Impressive Velocity Range: Achieved velocities ranging from low subsonic to supersonic speeds, with a maximum velocity of Mach 2 in non-recovery trials.

Contribution to National Defence: RTRS has played a crucial role in supporting India's defense capabilities by providing essential testing infrastructure for armament, missile systems, aerospace technologies, and other defense-related projects.

Continuous Progress: With testing operations conducted year-round, RTRS demonstrates its commitment to meeting the evolving needs of defense research and development, driving continuous progress in the field [4]

Fig 1.4: Gaganyaan Trial at RTRS [5]

1.2 Introduction to Projects

Metallic Braking Project

The Metallic Braking project represents a pioneering approach to friction-based braking systems, specifically designed to enhance braking efficiency and safety in various applications. This project centers around the innovative utilization of wedges in friction, introducing a novel mechanism to augment braking performance.

Project Objectives: The primary objective of the Metallic Braking project is to develop and implement an braking system that maximizes braking effectiveness while ensuring stability and reliability. Through the strategic application of wedges and the manipulation of frictional forces, this project aims to create a braking mechanism capable of delivering superior performance across diverse scenarios.

Project Scope: The scope of the Metallic Braking project encompasses the design, prototyping, and testing of a specialized braking system. Central to this scope is the integration of two distinct wedges, each serving a unique function within the braking process. The project will explore various configurations and mechanisms to optimize the interaction between these wedges and the braking surface.

MATLAB GUI Project for Dynamic Testing Trials

The MATLAB GUI project for dynamic testing trials represents a critical advancement in the domain of defense research and development, specifically tailored to support the testing and estimation processes within the Defence Research and Development Organisation (DRDO). Developed as part of collaborative efforts with DRDO, this project aims to provide a sophisticated toolset for analyzing and estimating crucial parameters during dynamic testing trials, enabling researchers and engineers to make informed decisions effectively**.**

Project Objectives: The primary objective of the MATLAB GUI project is to design and implement an intuitive interface that facilitates the estimation of various parameters essential for dynamic testing trials conducted by DRDO. By leveraging MATLAB's powerful computational capabilities and user-friendly interface design tools, the project aims to streamline the process of inputting trial parameters, conducting simulations, and analyzing results to derive actionable insights**.**

Project Scope: The scope of the MATLAB GUI project encompasses the development of a comprehensive software solution tailored to the specific requirements of dynamic testing trials within DRDO. Key components of the project include designing an intuitive user interface, integrating advanced algorithms for simulation and estimation, and providing visualization tools to represent trial results effectively**.**

1.3 Introduction to Software's Used

Introduction to MATLAB and GUI Development

MATLAB, a powerful computational software package, is widely recognized for its versatility in numerical analysis, algorithm development, and data visualization. One of the key features that sets MATLAB apart is its capability to develop Graphical User Interfaces

(GUIs), which provide an intuitive and interactive platform for users to interact with MATLAB-based applications. In this introduction, we delve into the fundamentals of MATLAB and explore the intricacies of GUI development using this robust platform.

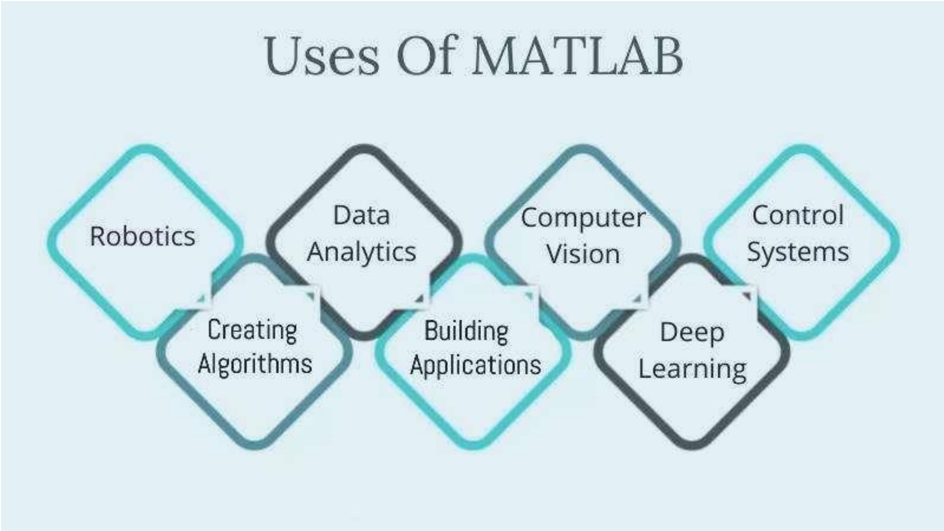


Fig 1.5: Uses Of MATLAB

MATLAB: A Versatile Computational Environment MATLAB

MATLAB, short for Matrix Laboratory, serves as a comprehensive environment for numerical computation, visualization, and programming. Its extensive library of built-in functions and toolboxes facilitates a wide range of applications, spanning engineering, science, finance, and beyond. MATLAB's interactive nature, coupled with its intuitive syntax, makes it a preferred choice for researchers, engineers, and educators worldwide.

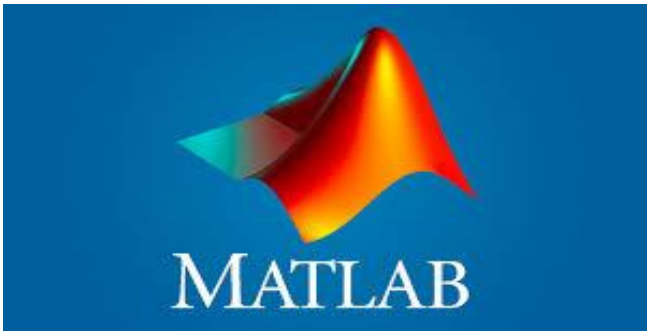


Fig 1.6 MATLAB Logo

Benefits of GUIs in MATLAB Applications

Integrating GUIs into MATLAB applications offers several benefits, including enhanced usability, improved workflow efficiency, and simplified data analysis. GUIs provide users with a familiar interface to interact with complex algorithms and data, reducing the learning curve and facilitating rapid exploration and experimentation.

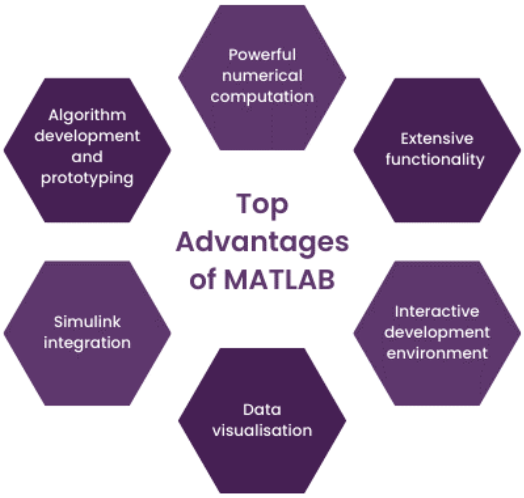


Fig 1.7: Advantages of MATLAB

Conclusion:

Empowering Innovation with MATLAB GUIs In conclusion, MATLAB's robust computational capabilities, coupled with its intuitive GUI development tools, empower developers to create sophisticated applications that address a diverse range of challenges across various domains.

CHAPTER-2

LITERATURE SURVEY: METALLIC BRAKING

2.1 REVIEW OF BRAKING SYSTEMS AND THEIR MECHANISM

Braking systems play a crucial role in ensuring safety and control in various dynamic tests, especially in rocket sled tests. This review explores different braking mechanisms employed in rocket sled testing, highlighting their functionality, advantages, and limitations.

1. Brake and Decelerate Water Scoop

Rocket sled test is an important ground dynamic test equipment. It is the rocket engine as a power device, along the high-precision rail high-speed slide of the ground test equipment. The rocket sled can simulate various environmental conditions according to the designed speed and acceleration. The rocket sled uses specially designed brakes for braking recovery. Rocket sled uses water brake braking mode.

Brake on water scoop is to use water scoop to change the inlet volume of water scoop in a trough of certain width and depth, and to absorb kinetic energy through the flow resistance to achieve the effect of deceleration and interception. This way is generally used in the deceleration brake section. It provides a relatively constant braking force. It causes the block to decelerate in the uniform deceleration state. It has good deceleration effect and low cost, and is widely used in high speed rocket skid rail test at home and abroad.

When the water brake device passes through the water tank between the tracks, the kinetic energy of the sled car is converted into the kinetic energy of water by using the kinetic energy conversion principle. The Rocket Sled's water brakes enter the water at a very high speed. The water entry method belongs to high-speed impact water entry. The impact force is bigger when entering water. When entering water, there are some risks in the structure of sled cars. It is difficult to study the structural safety of water brake when entering water. To some extent, this limits the application range of water brake.

2. Sand Deceleration

When the sand decelerates, the solid probe is attached to the rocket sled. The probe is placed in the sand between the two slides. The brake force is generated by the resistance of sand to the probe, which is similar to the deceleration of water brake.

3. Damping Partition Buffer Deceleration

This method employs damping plates made of polyethylene plastic and foam board to absorb kinetic energy upon impact, suitable for low-speed sled operations due to its low cost.

4. Drag Damping Deceleration

Tow damping deceleration is a nylon tow on the track. The tow ends are not fixed. When the sled goes through the tow, it pulls the tow together. This results in an increase in the mass and upwind area of the sled car, thereby reducing the speed and deceleration distance of the sled car.

5. Drag Parachute Deceleration

Drag parachute brake is the drag parachute hook on the rocket block. When braking, the drag parachute opens. Use air resistance to slow down and stop the block. There are two kinds of drag parachute brake: parachute brake and parachute brake. The parachute brake is to place the drag parachute at a designated spot beside the track. The parachute hook on the body of the rocket block caught the parachute line as it passed. Drag the parachute out of the barrel. Because the parachute brake is used to pull the drag parachute out of the parachute barrel. The speed of the line and the umbrella increases from zero to the same speed of the block when they are pulled out of the barrel.

There's a lot of impact overload on the line and the umbrella. The higher the speed of the block, the greater the mass of the cord and the parachute, and the greater the impact load on the pull out. This puts a strict limit on the size and braking speed of the umbrella.

6. Drag Plate Deceleration

Deceleration plate, as the name implies, the main function is to decelerate. Speed retarders were first used in aircraft. In an aircraft equipped with speed plates, there are usually two to four speed plates arranged on the fuselage or wing. When the aircraft is flying, the deceleration plate is close to the aircraft body in the closed position. The outer surface of the speed reducer is part of the streamline of the aircraft. When the plane lands, it needs to increase drag. The speed reducer is hydraulically powered to open at an Angle. The upwind area of the aircraft is increased and the aircraft flow profile is destroyed. Air forms the effect of increasing drag and turbulence. Cause the plane to slow down sharply. The greater the flight speed, the better the drag increase effect of deceleration plate. This is how the rocket sled decelerates.

Drag plate deceleration technology is to increase the headwind area of the rocket sled, so as to increase the air resistance of the rocket sled, to achieve the sled car deceleration.

7. Reverse Rocket Deceleration

The application of recoil rocket in space field has been very mature. The simplest is the recovery of the spacecraft. Fire the reverse rocket while near the ground, allowing the capsule to land gently.

Currently, the complex and powerful backrocket is a recovery from Musk's SpaceX's Falcon 9 rocket. This method has high accuracy and reliability .Reverse rocket deceleration is when the block needs to brake the reverse rocket ignition. Because

the thrust direction is opposite to the movement direction of the block to achieve braking. Its main disadvantage is high cost and heavy weight increase of the block, so it is less used.

8. Stop Cable Brake Deceleration

The principle of the rope brake is that the tackle is fitted with a hook. Hook a nylon plus wire rope cable that is horizontally arranged between two rails and symmetrical on the left and right sides. The cable is connected to a piston inside a conical cylinder filled with water. The piston moves in the direction of decreasing diameter. The flow of water around the piston creates hydrodynamic resistance. Hydrodynamic resistance consumes the kinetic energy of the block. This type of braking force is often used in the end of the block brake.[6]

Conclusion

Each braking mechanism offers unique advantages and challenges, catering to different speed ranges and test requirements in rocket sled testing. Understanding these mechanisms is crucial for optimizing safety and efficiency in dynamic testing environments.

2.2 The Role of Wedges in Metallic Braking

Wedges

A wedge is an indispensable and elegantly simple machine, typically constructed from wood or metal, characterized by a triangular or trapezoidal cross-section. It plays a crucial role in a variety of applications, commonly utilized for making fine adjustments in the position of an object—such as tightening fits or keys for shafts—and for separating wooden pieces. Furthermore, wedges can be employed to lift heavy weights or alter their positions when necessary, demonstrating their versatility and effectiveness in mechanical tasks (as illustrated in the figure below).

The wedge is one of the six classical simple machines, alongside the lever, pulley, wheel and axle, screw, and inclined plane. These fundamental devices form the basis of many complex mechanical systems. Common examples of wedges include knives, axes, and chisels, which are primarily used for cutting. Their utility spans numerous fields, including construction, engineering, automotive, and medical applications, highlighting their broad significance.

The design of a wedge features sloping sides known as faces or surfaces, and a sharp end referred to as the edge or apex. When a wedge is inserted or forced between two objects or surfaces, it generates a perpendicular force at its inclined sides. This action results in either the separation of the objects or the insertion of the wedge, showcasing the mechanical advantage provided by this simple machine.

Wedges apply force in a manner that translates a small input force into a larger output force, facilitating tasks that would otherwise require significantly more effort. This principle underpins their widespread use in both everyday tools and specialized industrial applications. By converting force direction and amplifying the applied force, wedges make it possible to perform tasks such as splitting, lifting, and securing with greater efficiency and precision. [7-9]

The free body Diagram of wedge is shown below:

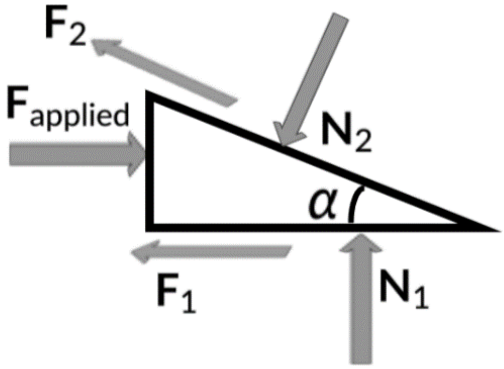


Fig 2.1: Various Forces on a Wedge

Where,

F1=Friction Force on the ground.

F2=Friction Force on the surface of wedge.

N1=Normal Reaction on the ground.

N2=Normal Reaction on the surface of wedge.

$F_{applied}$=Force applied.

α=Wedge Angle

CHAPTER 3

METHODOLOGY

3.1 Wedge-Enhanced Metallic Braking

Assumptions

- Coefficient of friction (μ) between the Wedges is 0.15 and 0.35 between lower Wedge and sliding surface.
- The thrust line of the retrorocket must coincide with the centre of gravity (c.g.) of the wedge on which the retro rocket force will be transferred.
- Proper contact must be ensured between the braking surfaces, particularly between the wedges and the surface they're braking against.
- At the time of braking, the acceleration considered for calculating reaction forces on the slippers is instantaneous. The mean acceleration is considered for the instant of time.

Principle of Metallic Braking

Wedges have the effect of allowing users to create very large normal forces to move objects with relatively small input forces. Therefore, the friction forces in wedge systems also tend to be very large.

THEORY

In the proposed braking system, two wedges are stacked, with their surfaces facing each other. Initially, a noticeable gap exists between the wedges, and they are not in contact. The lower wedge is designed to slide freely, while the upper wedge remains fixed or attached to the body of the system.

When the brakes need to be applied, the motion of the body initiates the movement of the lower wedge. As the body moves forward, it pushes against the lower wedge, causing it to slide towards the upper wedge. This sliding motion brings the surfaces of the wedges into contact with each other.

As the lower wedge continues to move against the upper wedge, friction is generated between their surfaces. This frictional force opposes the direction of the sliding motion, contributing to the deceleration or halting of the body's movement.

Simultaneously, as the lower wedge presses against the upper wedge, the normal force between them increases. This increase in normal force, resulting from the wedge geometry and the force applied by the moving body, enhances the frictional force between the wedges and the surfaces they interact with. Consequently, this amplification of friction aids in the braking process by increasing the resistance to motion between the wedges and the body.

Therefore, the interaction of wedges leads to increased frictional force between the wedges and the body, ultimately enhancing the braking effectiveness of the system. [10-13]

3.2 Friction Amplification Calculations for Dual Wedge Mechanisms.

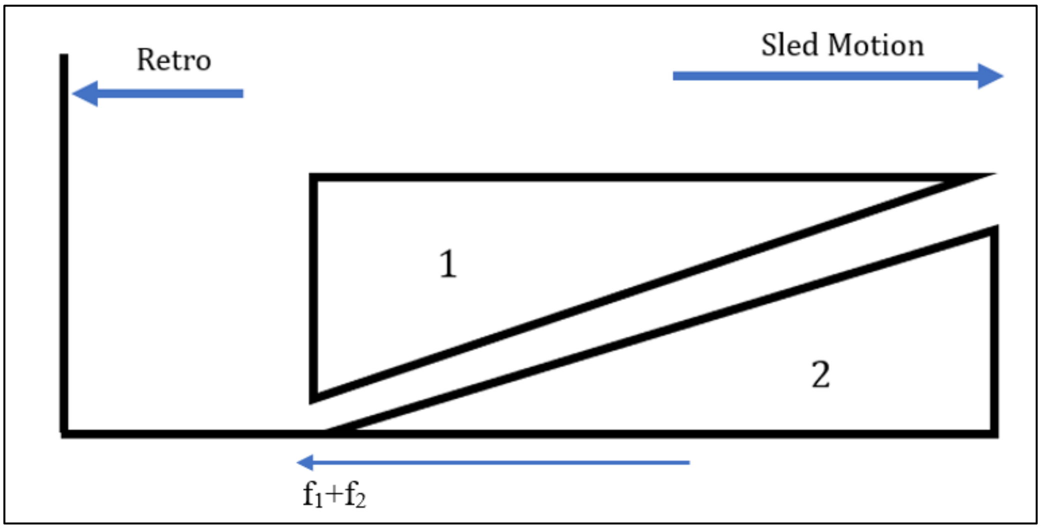


Fig 3.1: Mechanism of Wedges in Metallic Braking

$$\text{Net Friction Force} = f_1 + f_2$$

f_1=Friction Force due to Wedge.

f_2=Friction Force due to slippers.

Note:

Mass, m = 260kg

Retro Force, F = 2.4 Tonnes = 23544N

Wedge angle, θ = 5°

Coefficient of Friction between wedges, μ = 0.15

Coefficient of Friction between track and wedge, μ' = 0.35

Free body diagram of upper wedge:

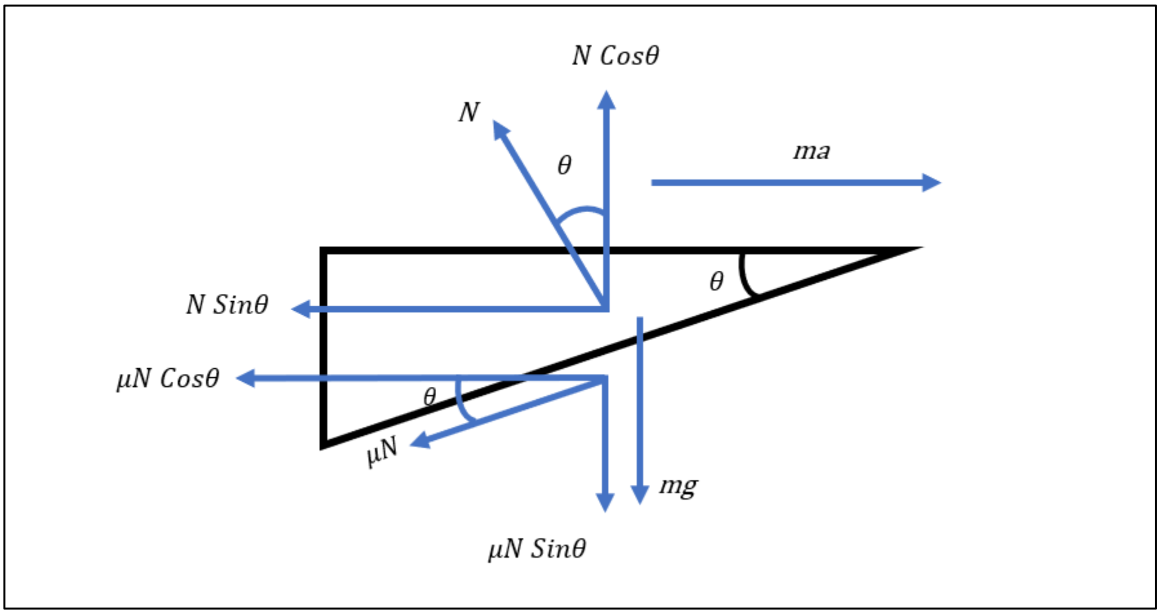


Fig 3.2: FBD of Upper Wedge

Free body diagram of lower wedge:

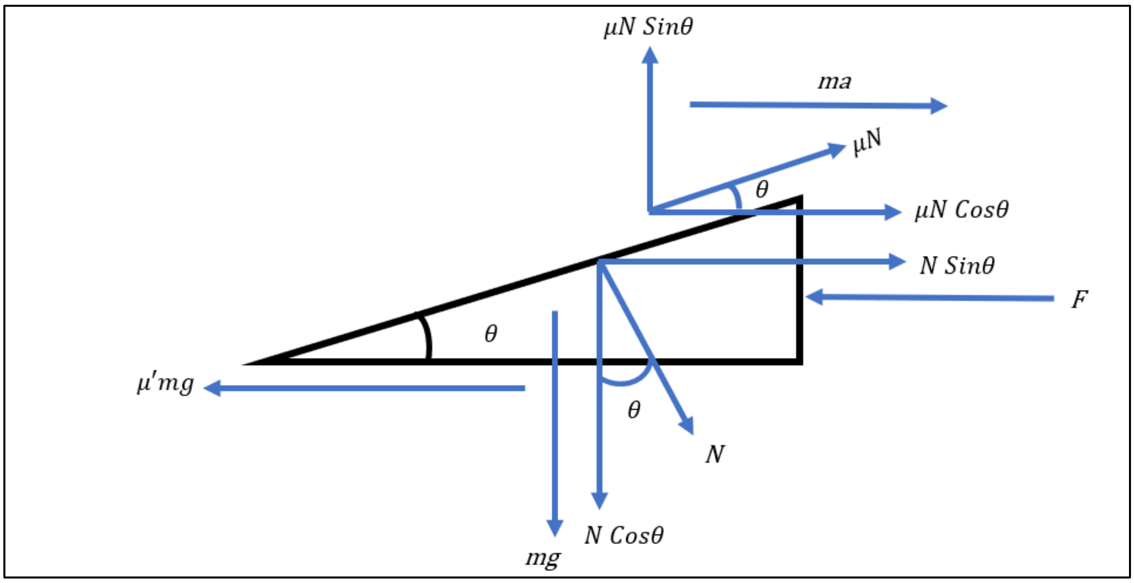


Fig 3.3: FBD of Lower Wedge

Considering Equilibrium of Forces,

$\sum F_x = 0;$

$\sum F_y=0;$

In x-direction (for lower wedge):

$$\sum Fx = \mu N\ \mathrm{Cos}\theta\ +\ N\ \mathrm{Sin}\theta + ma - \mu' mg - F$$

$$\mu N\ \mathrm{Cos}\theta\ +\ N\ \mathrm{Sin}\theta + ma - \mu' mg - F = 0$$

$$N = (F + \mu' mg - ma)/(\mu\ \mathrm{Cos}\theta\ +\ \mathrm{Sin}\theta)$$

Putting all the values,

$$N = (23544 + 0.35 * 260 * 9.81 - 260 * (-8.0910)/(0.15\ \mathrm{Cos}5°\ +\ \mathrm{Sin}5°)$$

$$Normal, N = 111933.3258N$$

In y-direction (for lower wedge):

N'=Resultant Normal Reaction in y-direction

$$N' = N\ Cos\theta + mg - \mu N\ Sin\theta$$

$$N' = 103289.3693 * Cos5° + 260 * 9.81 - 0.15 * 103289.363 * Sin5°$$

$$N' = 112594.6409N$$

$$Frictional\ Force, f_1 = \mu' * N'$$

$$f_1 = 0.35 * 112594.6409 = 39408.80N$$

$$\boldsymbol{f_1 = 3.941\ \mathrm{Tonnes}}$$

System Assumption for Reaction Calculations

Simply Supported System Assumption:

To analyse the forces within the system, it is assumed to be simply supported. This means that the system allows for vertical reactions at the support points (slippers and ground).

This assumption simplifies the calculation of reaction forces at the contact points, providing a clear understanding of the forces in the braking mechanism.

Calculation of Reaction at the Slippers

Identifying Support Points:

The four slippers that support the lower wedge are considered as support points.

Free Body Diagram:

Draw a free body diagram, indicating the applied force from the body, the frictional force, and the normal reaction forces at the slippers and the ground. Note that thrust will come into play when the retro rocket is fired. The gravitational force (mg) acts downward due to the sled's weight.

Equilibrium Equations:

Use the principles of static equilibrium to set up equations:

Sum of vertical forces ($\Sigma Fy = 0$): The vertical forces include the applied force from the body and the reaction forces at the slippers and ground.

Sum of horizontal forces ($\Sigma Fx = 0$): The horizontal forces include the frictional force between the wedges.

Sum of moments about a point ($\Sigma M = 0$): The moments involve the forces applied at various points on the wedges.

Solving for Reactions:

Solve these equations to find the reaction forces at the slippers, and at the ground.

Reactions at slippers before firing the retro rockets

Before firing the retro rockets, let's break down the situation with the sled traveling at a velocity of 99 m/s:

The length between the two reaction points is denoted as l, the vertical distance between the center of gravity (CG) and the track is y, the horizontal distance between the center of gravity and the track is x, and a is the acceleration of the body at the time of braking.

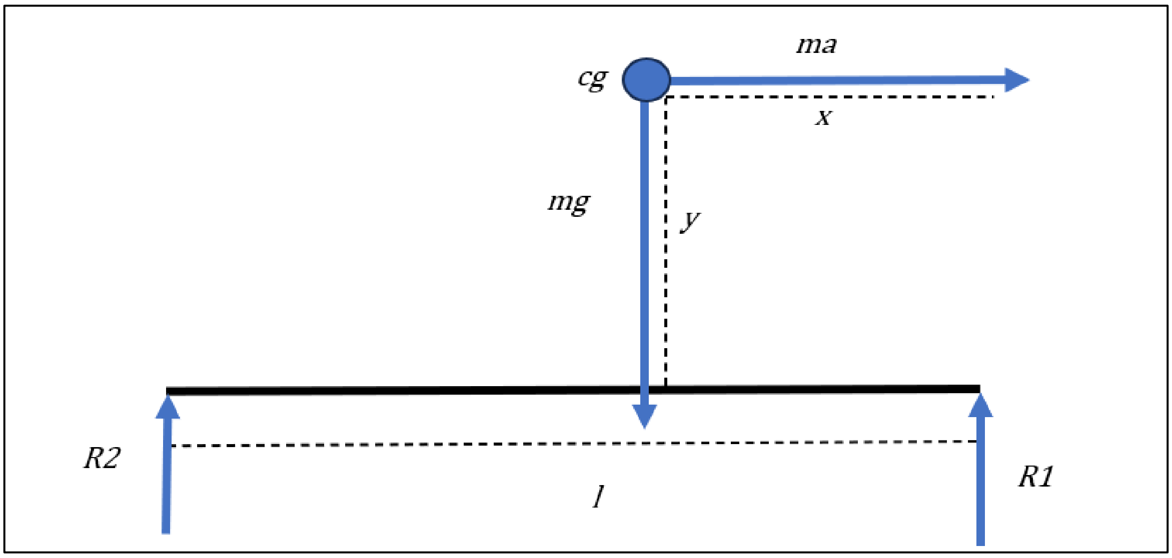


Fig 3.4: Forces at Slippers before firing Retrorocket

Here:

x = 300mm y = 400mm

l = 500mm a=-8.0910m/sec^2

Resolving forces in x and y directions,

In y-direction:

$$R1+R2=mg$$

Taking Moments about R1,

$$(R2 * l) + (-mg * x) + (ma * y) = 0$$

$$(R2 * 0.5) + (-260 * 9.81 * 0.3) + (260 * (-8.0910) * 0.4) = 0$$

$$R2 = 3213.288N$$

Therefore,

$$R1 = mg - R2 = -662.688N$$

Reactions at slippers at the time retro rockets firing

At the time retro rockets firing, velocity = 98m/s (say)

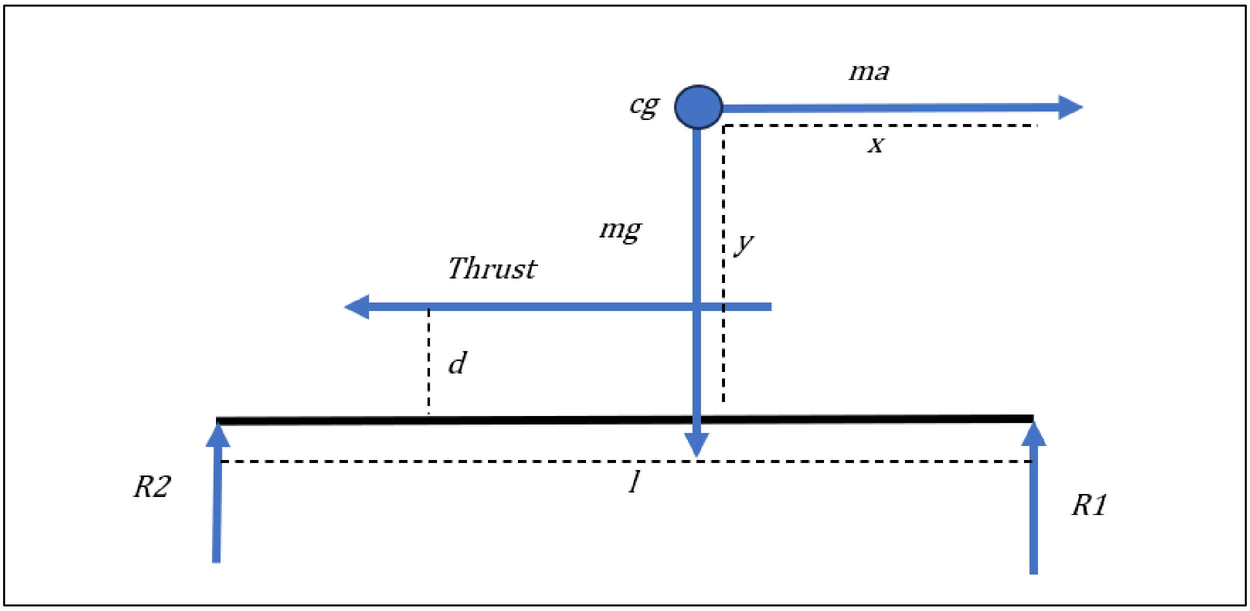


Fig 3.5: Forces at Slippers just after firing Retrorocket

Here:

x = 300mm y = 400mm

l = 500mm d=43mm

a=-7.8655m/sec^2

Resolving forces in x and y directions,

In y-direction:

$$R1+R2=mg$$

Taking Moments about R1,

$$(R2 * l) + (-mg * x) + (ma * y) + (-thrust * d) = 0$$

$$(R2 * 0.5) + (-260 * 9.81 * 0.3) + (260 * (-7.8655) * 0.4) - (23544 * 0.043) = 0$$

$$R2=5153.624 \text{ N}$$

Therefore,

$$R1 = mg - R2 = -2603.024N$$

Net Reaction, R=2550.6 N

$$f_2=\mu * R=0.3 * 2550.6=892.71 \text{ N}$$

$$\boldsymbol{f_2=0.089271\text{Tonnes}}$$

Total Frictional Force= $f_1 + f_2$ =39408.1243 +892.71=40300.8334 N

Total Braking Force, F_B= (Total Frictional Force) + (Retrorocket Thrust)

Total Braking Force, F_B = 23544+40300.8334 =63844N

$$\boxed{\mathbf{F_B=6.3844Tonnes}}$$

The net force of 6.3844 tonnes (metric tons) in metallic braking from slippers and wedge action indicates the total braking force applied to stop or slow down the sled.

This braking force is essential for safely controlling and stopping the sled's movement, demonstrating the effectiveness of both slippers and wedge action in braking systems.

CHAPTER-4

RESULTS AND ANALYSIS: METALLIC BRAKING

• **Study Investigation**: The study investigated the efficacy of metallic braking systems by examining the impact of force amplification. Starting from an initial force of 2.4 tonnes, the system was enhanced to amplify this force to 6.3844 tonnes. This significant increase was aimed at improving overall braking performance and efficiency.

• **Braking Capability Enhancement**: This force amplification led to a substantial improvement in braking capability. As a direct consequence, the study found that the number of retrorockets required for effective braking could be reduced from 17 to just 6. This reduction indicates a more efficient use of resources and a streamlined braking system.

• **Static Equilibrium Conditions**: It's crucial to understand that these results were derived from idealized static equilibrium conditions. In such scenarios, variables remain constant and predictable, providing a controlled environment to assess the braking system's performance. However, these conditions do not fully capture the complexities and variabilities of real-world dynamic scenarios.

• **Dynamic Conditions Predictions**: When predicting the performance in dynamic conditions, where factors such as speed variations, environmental influences, and real-time adjustments come into play, the expected reduction in the number of rockets is likely to be around 10. This adjustment accounts for the less predictable nature of dynamic braking scenarios, providing a more realistic estimate of the braking system's efficiency in practical applications.

Future Scope:

- **Dynamic Testing**: Conduct thorough dynamic testing to validate braking performance under real-world conditions. Utilize simulations to model stress scenarios and identify potential weaknesses before physical tests.
- **Design Refinement**: Continuously refine braking components using advanced CAD tools and material science innovations. Simulations help test design iterations and materials, ensuring optimal configurations for real-world applications.
- **Control Systems Integration**: Integrate advanced control systems and sensors for real-time braking adjustments, enhancing performance in varying conditions. Use simulations to fine-tune these systems, ensuring robust and adaptive algorithms.
- **Alternative Mechanisms**: Explore and test alternative braking mechanisms, like regenerative or magnetic systems, to complement traditional metallic brakes. Simulations evaluate their effectiveness and integration with existing systems.
- **Advanced Simulations**: Utilize advanced simulations to predict and analyze the performance of braking systems under various conditions. These simulations provide valuable insights, reduce the need for extensive physical prototypes, and help identify and mitigate potential issues early in the development process.

CHAPTER 5
LITERATURE REVIEW: GUI DEVELOPMENT

5.1 Review of GUI Development Tools and Techniques

MATLAB, known for its robust computational capabilities, also offers a range of tools and techniques for GUI (Graphical User Interface) development. Here's a review of some key aspects:

1. **GUIDE (MATLAB's GUI Development Environment)**:

   MATLAB's GUIDE is a convenient drag-and-drop tool for designing GUIs. It allows users to create interactive interfaces by placing components such as buttons, sliders, text boxes, etc., onto a canvas.

   Pros: Easy to use, especially for beginners. Integration with MATLAB's workspace makes it seamless to link GUI components with MATLAB functions.

   Cons: Limited customization options compared to manually coding GUIs. GUIs created with GUIDE can sometimes lead to messy generated code, making it challenging to maintain for larger projects.

2. **Programmatic GUI Development**:

   MATLAB provides extensive support for programmatic GUI development using MATLAB's built-in functions and objects. This approach involves writing MATLAB code to create and manipulate GUI components.

   Pros: Offers greater control and flexibility over GUI design and behavior. Well-suited for complex GUIs and larger projects.

   Cons: Steeper learning curve compared to using GUIDE. Requires proficiency in MATLAB programming.

3. **Handle Graphics**:

   MATLAB's Handle Graphics system forms the backbone of GUI development. It allows for the creation, manipulation, and customization of graphical objects within MATLAB.

   Pros: Provides low-level access to GUI components, enabling precise control over appearance and behaviour.

   Cons: Requires understanding of MATLAB's graphics objects and their properties, which may be daunting for beginners.

4. **App Designer**:

   App Designer is MATLAB's modern GUI development environment introduced in newer versions. It combines the ease of GUIDE with the flexibility of programmatic development.

Pros: Offers a streamlined interface for designing and coding GUIs. Allows for both visual design and programming within the same environment.

Cons: Limited to newer MATLAB versions. While it addresses some of the shortcomings of GUIDE, it may still generate verbose code in certain cases.

5. **Third-Party Tools and Libraries**:

   MATLAB also supports integration with third-party tools and libraries for GUI development, such as Java Swing or web-based interfaces using MATLAB Web App Server.

   Pros: Enables leveraging external libraries and frameworks for specialized GUI requirements.

   Cons: Integration may require additional setup and learning curve for interfacing with MATLAB.

   Overall, MATLAB provides a range of tools and techniques for GUI development, catering to both beginners and advanced users. The choice between GUIDE, programmatic development, or App Designer depends on the specific requirements, familiarity with MATLAB, and preferences for design versus control.

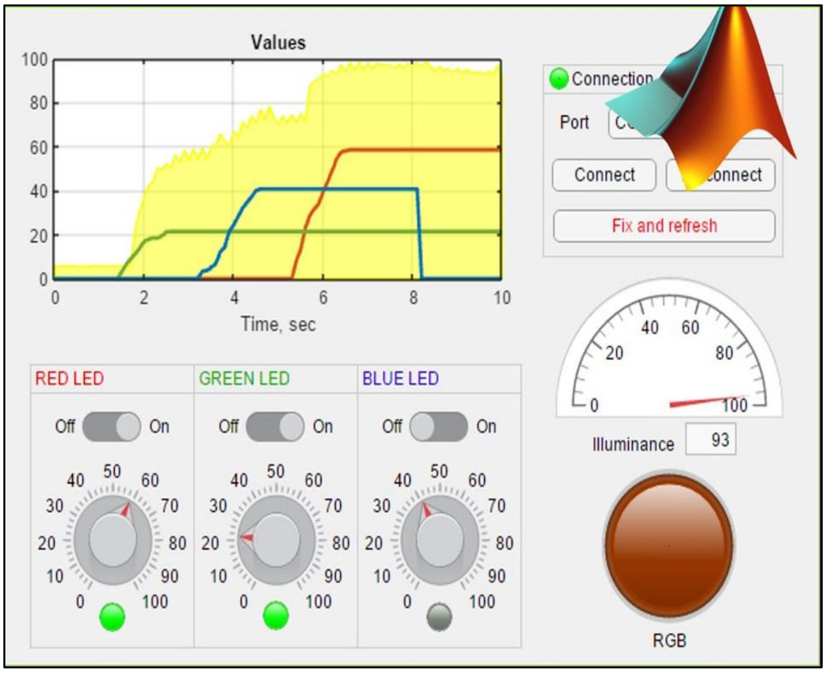


Fig 5.1: Example of GUI in MATLAB

5.2 Context and Significance of Interface Development

Graphical User Interface (GUI) development holds significant importance in modern software applications across various domains. Here's the context and significance of GUI development:

1. **Enhanced User Experience**: GUIs serve as the primary means for users to interact with software applications. A well-designed GUI can significantly enhance the user experience by

providing intuitive controls, clear navigation, and visually appealing interfaces. This usability factor is crucial for user satisfaction and retention.

2. **Accessibility and Inclusivity**: GUIs play a vital role in making software accessible to users with diverse abilities and backgrounds. Features like customizable font sizes, color schemes, and keyboard shortcuts can improve accessibility for users with disabilities, ensuring inclusivity in software usage.
3. **Increased Productivity**: Intuitive GUIs can streamline complex workflows and tasks, leading to increased productivity for users. By presenting information in a visually organized manner and offering convenient tools and functionalities, GUIs can help users accomplish tasks more efficiently.
4. **Data Visualization and Analysis**: GUIs are often used to visualize data and present analytical results in a comprehensible format. Interactive plots, charts, and dashboards enable users to explore data dynamically, gain insights, and make informed decisions. This is particularly relevant in fields like data science, engineering, and finance.
5. **Rapid Prototyping and Iterative Development**: GUI development tools like App Designer facilitate rapid prototyping and iterative development cycles. Developers can quickly create mockups and prototypes of GUIs, gather feedback from users, and iterate on designs to refine the user interface and functionality.
6. **Cross-Platform Compatibility**: GUI frameworks and development tools often support cross-platform deployment, allowing applications to run seamlessly on different operating systems and devices. This versatility ensures broader reach and accessibility for software applications across various platforms.
7. **Integration with Backend Systems**: GUIs serve as the frontend interface for users to interact with backend systems and databases. Integration with backend services, APIs, and databases enables GUI applications to retrieve and manipulate data, perform computations, and execute complex tasks seamlessly.
8. **Customization and Personalization**: GUIs can be tailored to suit the specific needs and preferences of users. Customization options such as themes, layouts, and settings allow users to personalize their experience, fostering a sense of ownership and engagement with the software application.[14]

CHAPTER 6

DESIGINING THE INTERFACE

6.1 Mathematical Formulations and Equations

Introduction to Rocket Dynamics in Impact Testing: Understanding Forces and Motion

In the domain of impact testing, where a payload is propelled by a rocket to gain velocity before impact, comprehending the underlying forces and motion is paramount. This endeavour necessitates a thorough examination of Newtonian mechanics and the interplay of various forces acting upon the system.

1. Newton's Second Law:

Newton's second law serves as the cornerstone of our analysis, asserting that the net force acting on an object is proportional to its acceleration, defined as the rate of change of velocity with respect to time. Mathematically, this relationship is expressed as:

$$\mathbf{F_{net}=ma}$$

where:

- F_{net} represents the net force acting on the object,
- m denotes the mass of the object, and
- a signifies the resulting acceleration.

2. Forces in Impact Testing:

In the context of impact testing with a rocket-propelled body, several forces come into play:

a. **Thrust (F_T):** This force is generated by the rocket engine and propels the body forward. It is directed along the positive x-axis and contributes positively to the net force.

b. **Drag (F_d):** As the body accelerates through the surrounding medium, typically air, it experiences resistance known as drag. The magnitude of drag force is proportional to the square of the velocity and can be calculated using the formula:

$$\mathbf{F_d=1/2*C_d*\rho*A*v^{\wedge}2}$$

where:

- ρ is the density of the fluid (e.g., air),
- A is the reference area (such as the cross-sectional area of the body),
- C_d is the drag coefficient, and
- v is the velocity of the body relative to the fluid.

c. **Friction (F_f):** Frictional forces arise due to the contact between the body and its surroundings. In the context of impact testing, friction might manifest as surface friction between the body and the ground or other surfaces. The frictional force can be calculated using:

$$\mathbf{F_f = \mu N}$$

where:

- μ represents the coefficient of friction, and
- N is the normal force exerted on the body.

Fig 6.1: Forces Acting on a Rocket

Net Force and Acceleration:

The net force (F_{net}) acting on the rocket-propelled body is the vector sum of the thrust, drag, and frictional forces:

$$\mathbf{F_{net} = F_T - F_D - F_f}$$

The resulting acceleration (a) of the body is determined by Newton's second law, incorporating the net force and the mass of the body.

$$\mathbf{ma = F_T - F_D - F_f}$$

If N number of Rockets are used, Then

$$\textbf{Total Mass, } \mathbf{m = N * m_{en} + m_b + m_p}$$

$$\mathbf{ma = F_T * N - F_D - F_f}$$

where:

- m_{en} represent the mass of Rocket engine,
- m_b denotes the mass of the body, and
- m_p signifies the mass of payload.

Acceleration (a) Equation:

As previously discussed, acceleration is determined by Newton's second law:

$$\mathbf{a = F_{net} / m}$$

Velocity (v) Equation:

The velocity of the rocket-propelled body can be determined using the equation of motion:

$$v=u+a\Delta t$$

where:

- v is the final velocity,
- u is the initial velocity,
- a is the acceleration, and
- t is the time elapsed.

Position (x) Equation:

To find the position of the rocket-propelled body as a function of time, we employ the kinematic equation:

$$\Delta x=u\Delta t+1/2a\Delta t^2$$

where:

- x is the position,
- u is the initial velocity,
- a is the acceleration, and
- t is the time elapsed.

This equation describes the displacement of the body from its initial position as a result of its initial velocity and the acceleration experienced over time. By integrating the acceleration equation or utilizing other kinematic equations, we can derive expressions for the position of the body at various points in time during its motion.

6.2 Designing Interface Components and their Functions [14-18]

In this guide, we'll walk through the process of creating a MATLAB app step by step, starting with the basics of opening a new app and adding components to its interface.

Step-1: Creating a new Blank app

Launch MATLAB on your computer. You can typically find it in your list of installed applications or search for it in your system's search bar.

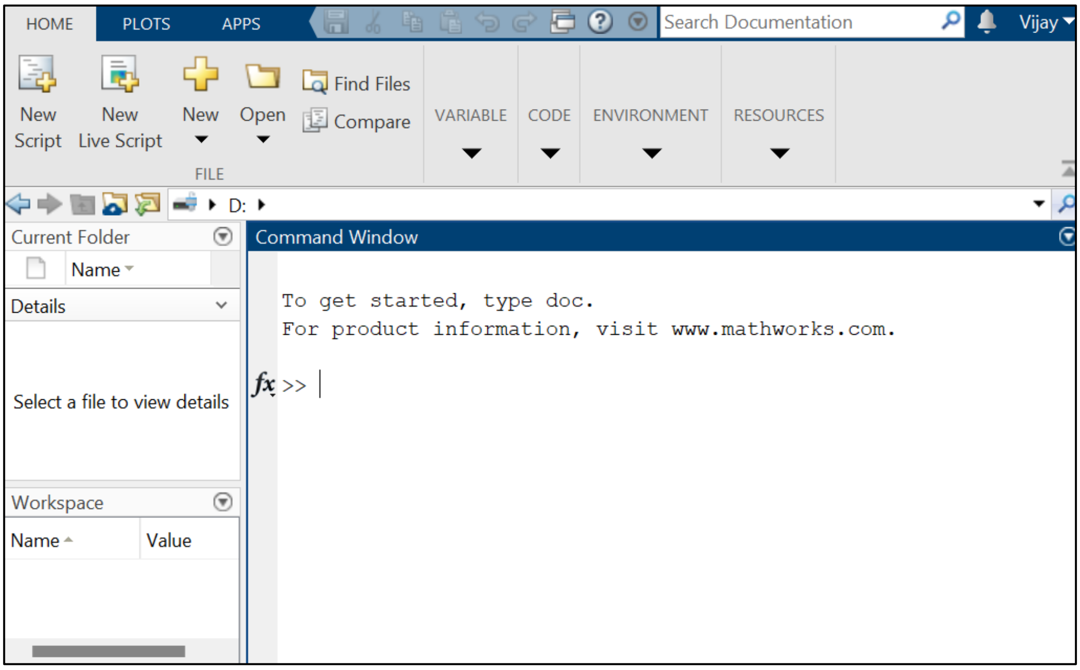


Fig 6.2: Illustrated Setup Guide for Creating a MATLAB app-1

Once MATLAB is open, navigate to the "Home" tab and locate the “New” section.

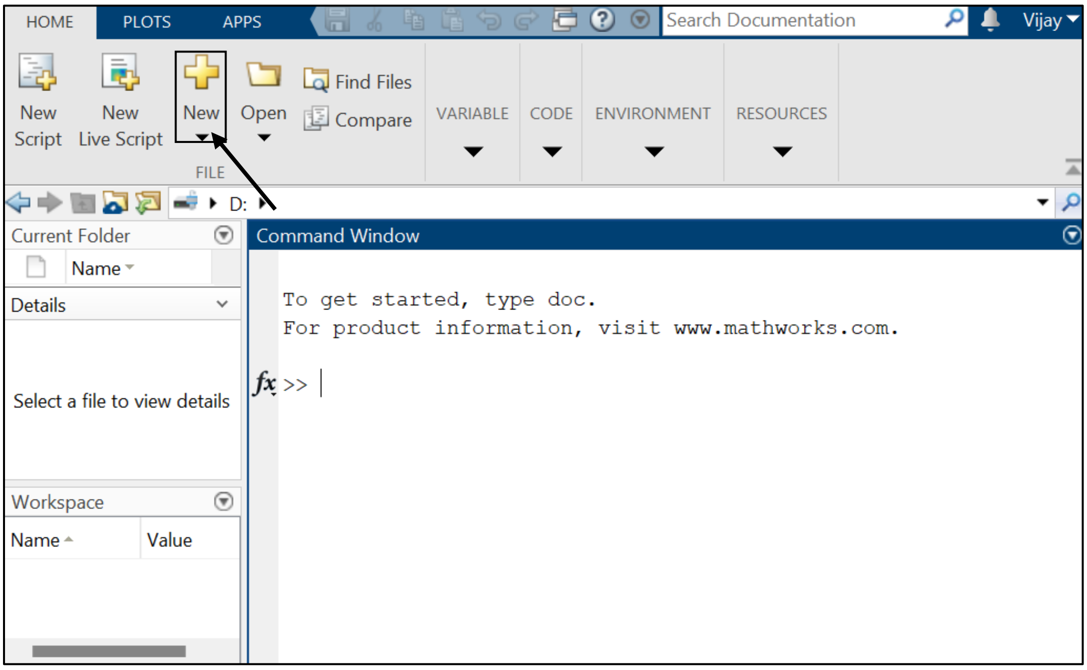


Fig 6.3: Illustrated Setup Guide for Creating a MATLAB app-2

Click on "New" to reveal a dropdown menu with various options and Select “App”.

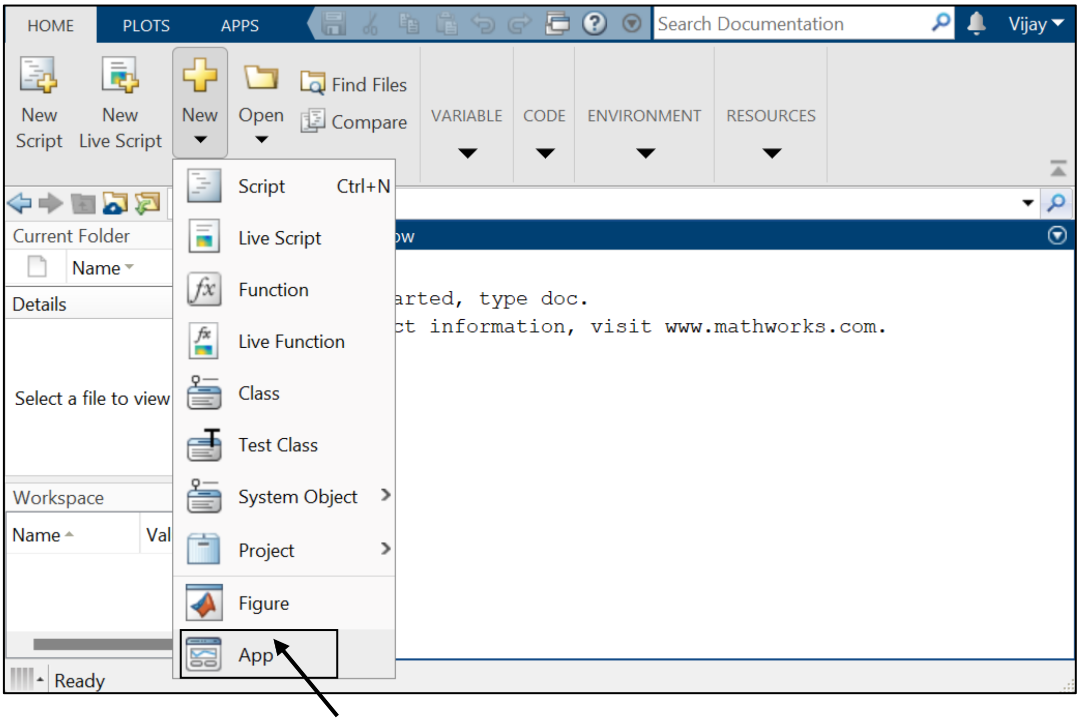


Fig 6.4: Illustrated Setup Guide for Creating a MATLAB app-3

Step-2: Access The Component Library

- Once the App is open, in the App Designer tool, you'll see a panel on the left side of the window called the "Component Library.

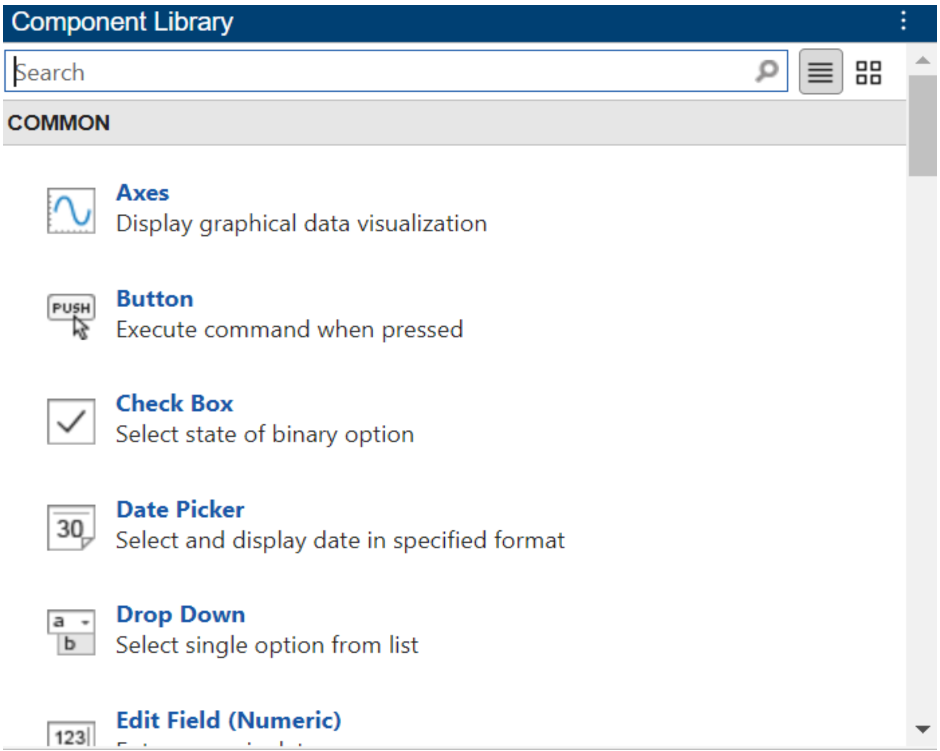


Fig 6.5: Illustrated Setup Guide for Creating a MATLAB app-4

- The Component Library contains a variety of UI components that you can add to your app to create an interactive user interface.
- Components in the library are categorized into sections such as "Common Controls," "Containers," "Numeric & Other Input," "Plots," and more.

Understand the Components:

- **Common Controls:** These include commonly used components like buttons, sliders, checkboxes, radio buttons, and labels. They allow users to interact with your app.
- **Containers:** Containers are components used to organize other components within your app. Examples include panels, tabs, grids, and tabs.
- **Numeric & Other Input:** These components are used for numerical input, such as text fields, sliders, and dropdown lists.
- **Plots:** If your app involves data visualization, you can use plot components like axes, graphs, and plots to display data.

Add Components to Your App:

To add a component to your app, click on the desired component in the Component Library panel and drag it onto the app canvas. Once placed, you can resize the component and customize its properties on the right side of the window.

Example: Adding a Button:

- To add a button to your app, locate the "Button" component in the "Common Controls" section of the Component Library.

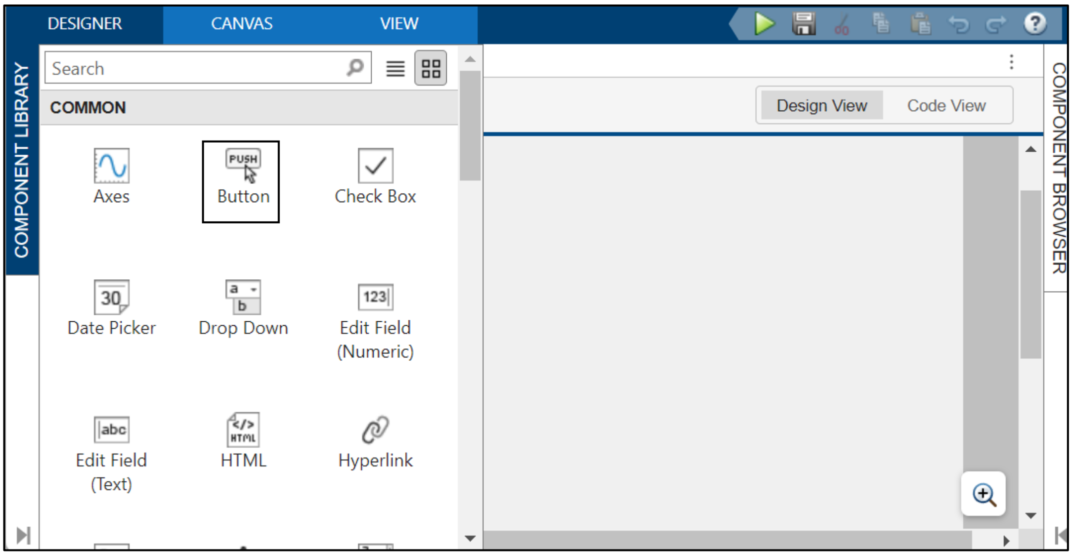


Fig 6.6: Adding Components to App Canvas-1

- Click on the button component and drag it onto the app canvas.

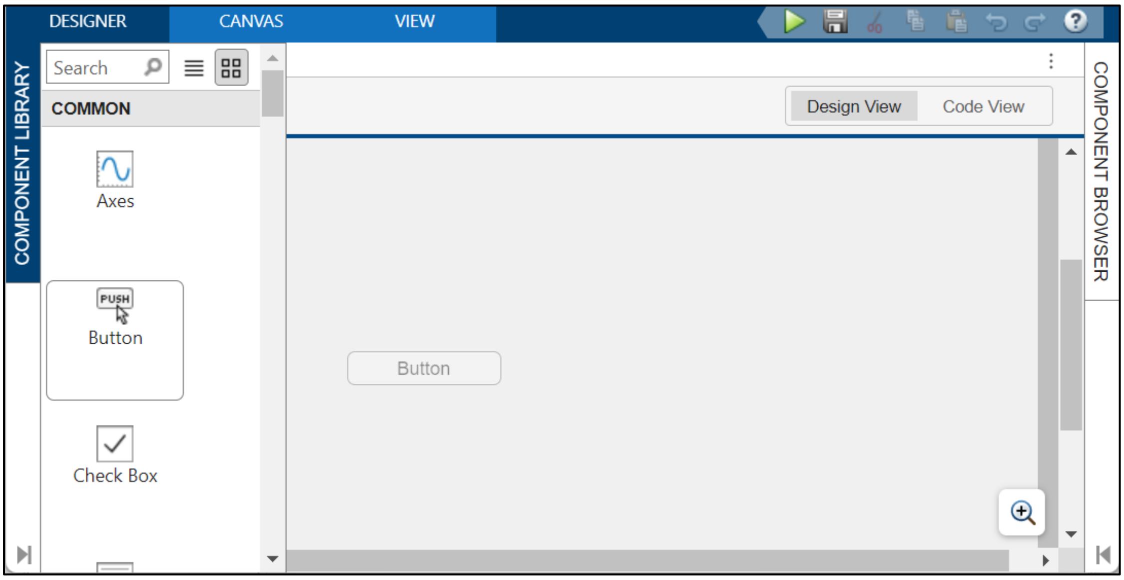


Fig 6.7: Adding Component to App Canvas-2

- Release the mouse button to place the button on the canvas.

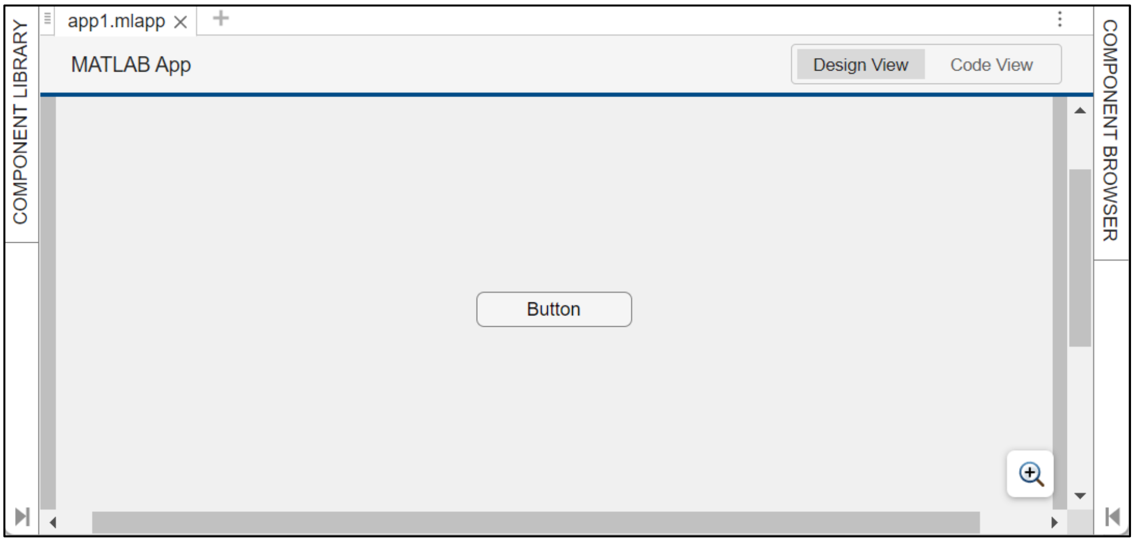


Fig 6.8: Adding Component to App Canvas-3

- You can then customize the button's text, size, colour, and other properties using the Property Inspector.

Step-3: Implementing Backend

Identify the GUI component you wish to add a callback function to, such as a button, slider, or text field.

Access and Navigate to the Callbacks Section: In the "Design View" tab of the App Designer, click on the desired GUI component. This action reveals the component's properties in the "Properties" pane.

Scroll down within the "Properties" pane to locate the "Callbacks" section. Here, you'll find various events associated with the selected GUI component, like ButtonPushedFcn for buttons or ValueChangedFcn for sliders.

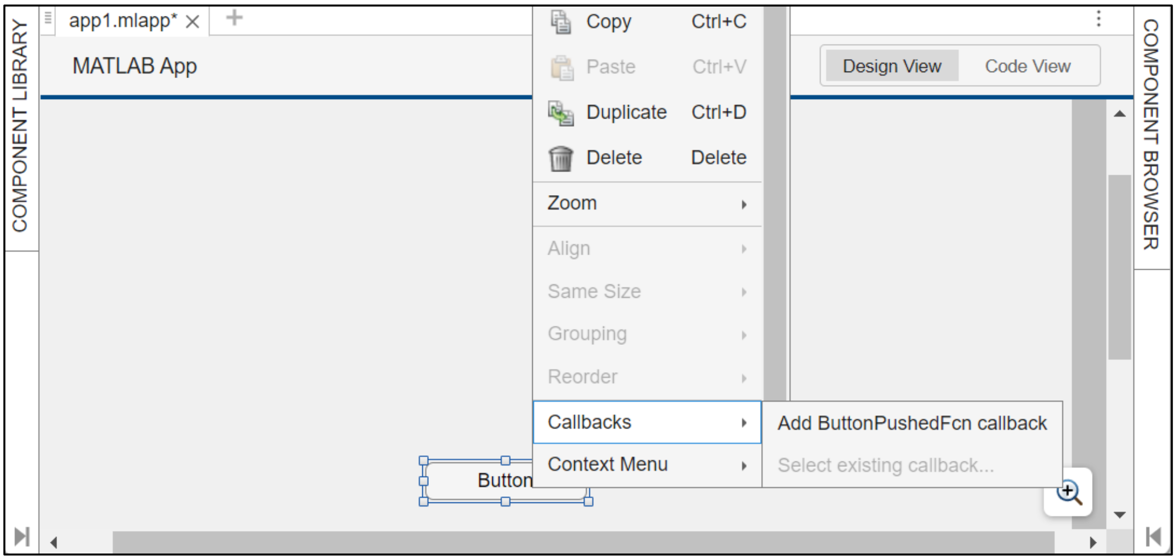


Fig 6.9: Selecting Callback Functions

Write Callback Code:

- Write the MATLAB code within the newly created or selected callback function to define the actions you want to perform when the associated event occurs. For example, for a button click event, you might write code to perform a calculation or update a plot.

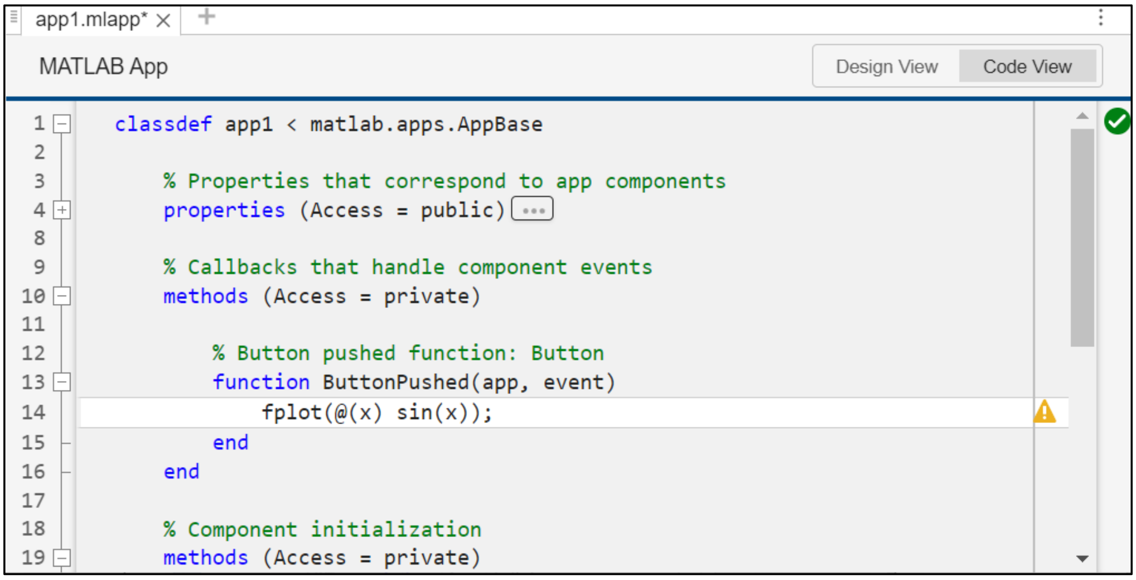


Fig 6.10: Backend Code for Demo App

Test the Callback: Run your app to test the newly added callback function. Interact with the GUI component associated with the callback to ensure that it behaves as expected.

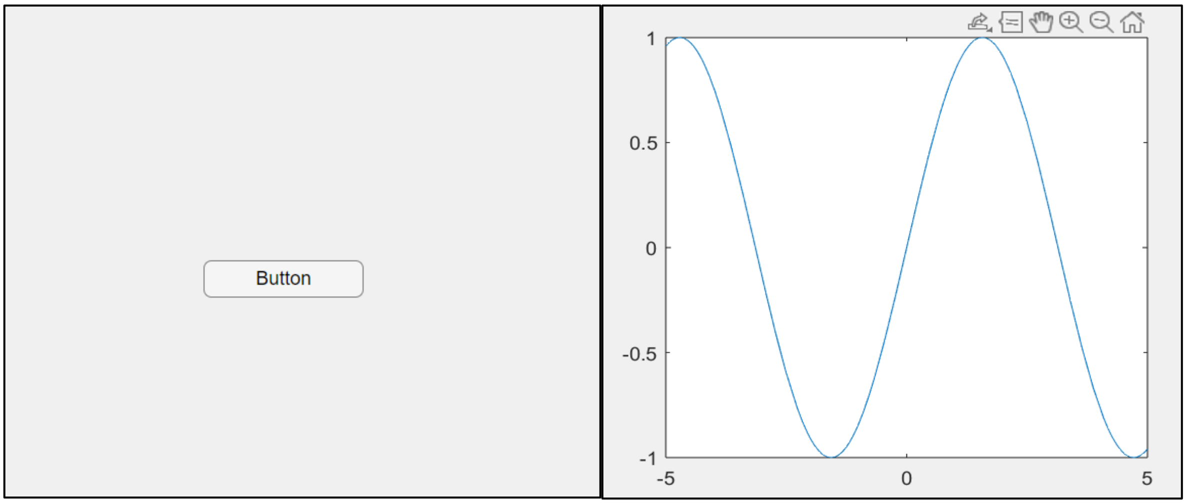


Fig 6.11: Executing a MATLAB App

Designing the Estimation Programme-Frontend

- **Initial Parameters:** This container serves as the initial point of input, housing parameters fundamental to the estimation process's initiation.

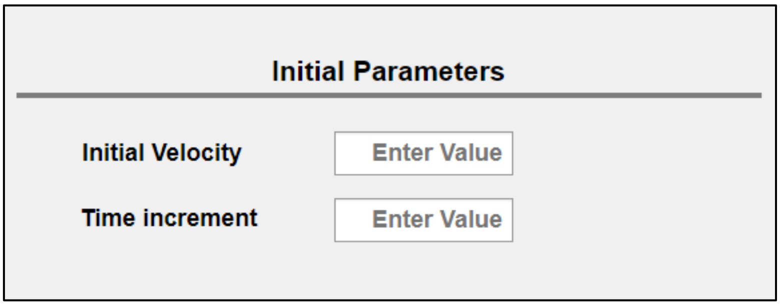


Fig 6.12: Frontend Input Paramaters-1

- **Payload and Fixtures Specification:** In this container, details pertaining to the payload and fixtures specifications are captured, crucial for precise estimation calculations.

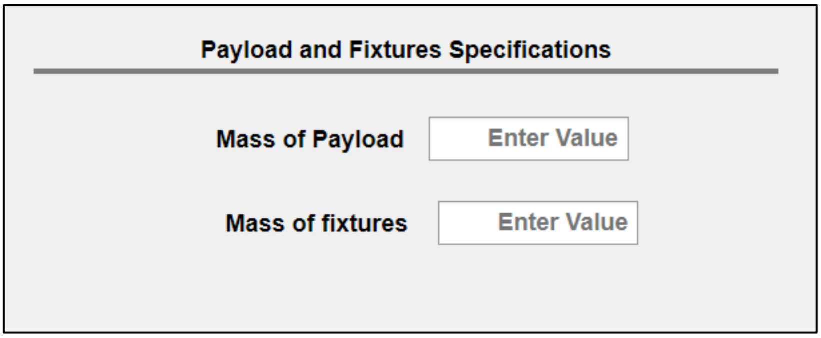


Fig 6.13: Frontend Input Paramaters-2

- **Drag Properties:** Within this container, inputs concerning drag properties are recorded, providing essential data for accurately modelling the drag forces acting upon the system.

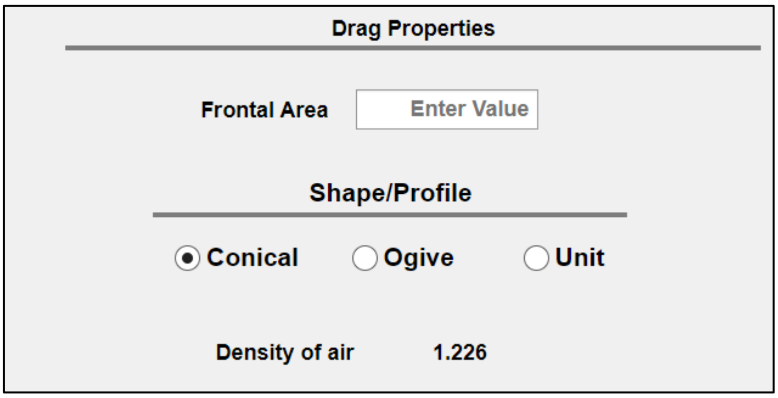


Fig 6.14: Frontend Input Paramaters-3

- **Friction Properties:** This container is dedicated to capturing friction properties, vital for factoring in frictional forces and their impact on the system's dynamics.

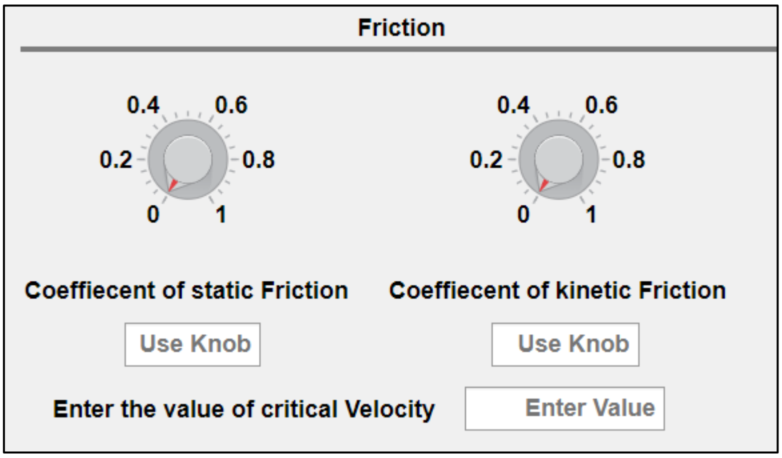
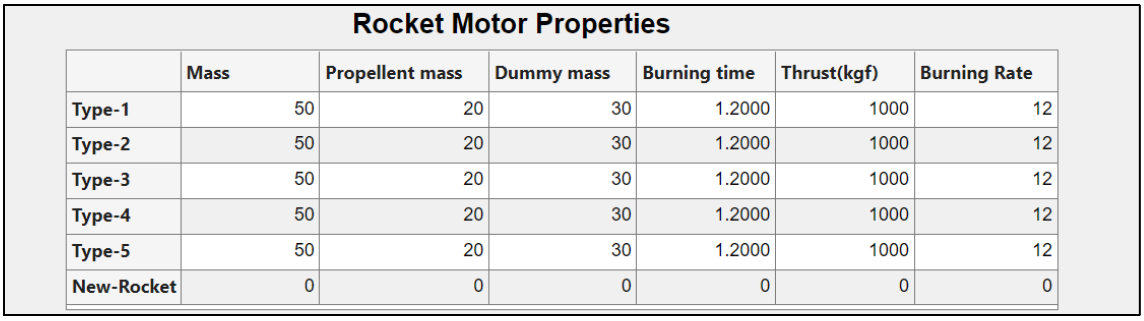


Fig 6.15: Frontend Input Paramaters-4

Additionally, the interface includes two UI tables, each serving distinct purposes:

- **Rocket Properties UI Table:** This table comprehensively displays rocket properties, offering a detailed overview of key parameters necessary for estimation.



Rocket Motor Properties

	Mass	Propellent mass	Dummy mass	Burning time	Thrust(kgf)	Burning Rate
Type-1	50	20	30	1.2000	1000	12
Type-2	50	20	30	1.2000	1000	12
Type-3	50	20	30	1.2000	1000	12
Type-4	50	20	30	1.2000	1000	12
Type-5	50	20	30	1.2000	1000	12
New-Rocket	0	0	0	0	0	0

Fig 6.16: Frontend Input Paramaters-5

- **Phase Firing UI Table:** Here, phase firing data is meticulously organized, allowing users to input and manage data crucial for the estimation process. A dedicated "Update Data" button facilitates seamless data updates post-entry, ensuring data accuracy and integrity.

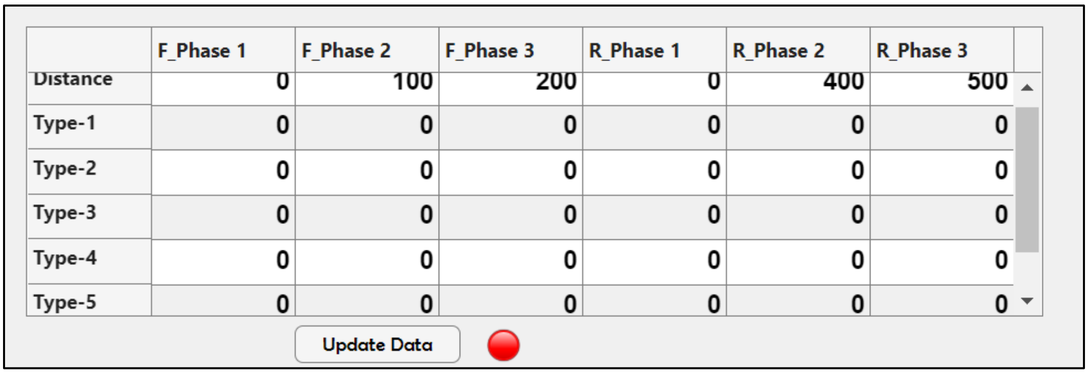


Fig 6.17: Frontend Input Paramaters-6

To streamline the estimation process and enhance user experience, two action buttons are provided:

- **Compute Result Button:** This button initiates the computation process, utilizing the input parameters to generate accurate estimation results.
- **Plot Curves Button:** Following computation, this button enables users to visualize estimation outcomes through plotted curves, facilitating a comprehensive understanding of the system's dynamics.

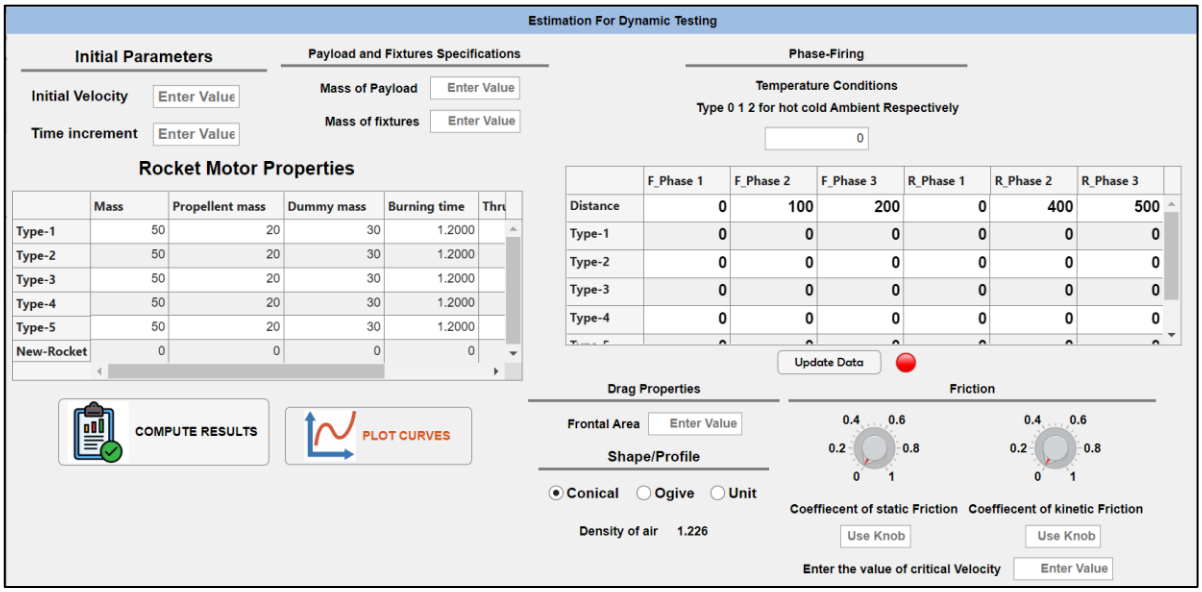


Fig 6.18: MATLAB Estimation Programme for Dynamic Testing

6.3 Navigating the Interface

Accessing input parameters is pivotal for accurate estimation of rocket dynamics. Parameters such as payload mass, clamp mass, drag properties, friction properties, and the number of rockets in different phases are essential. They provide crucial data for modeling and analyzing the rocket's behavior during ascent. Proper consideration of these parameters ensures precise estimation, reflecting real-world conditions and enabling informed decision-making in rocket design and operation.

Step-1: Input all required values, including initial velocity, time increment, masses, and drag properties. Additionally, utilize a knob to adjust the coefficient of friction, enabling fine-tuning of this parameter for precise modeling.

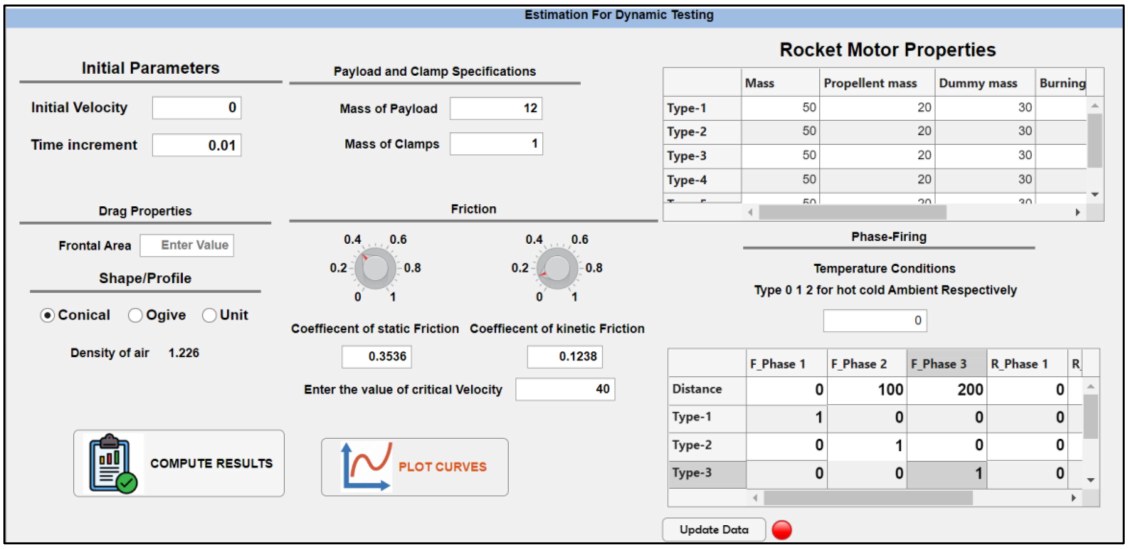


Fig 6.19: Accessing Input Parameters in Estimation App

Step-2: Press the "Update Data" button to refresh the number of rockets in different phases. The lamp will turn green once this action is completed.

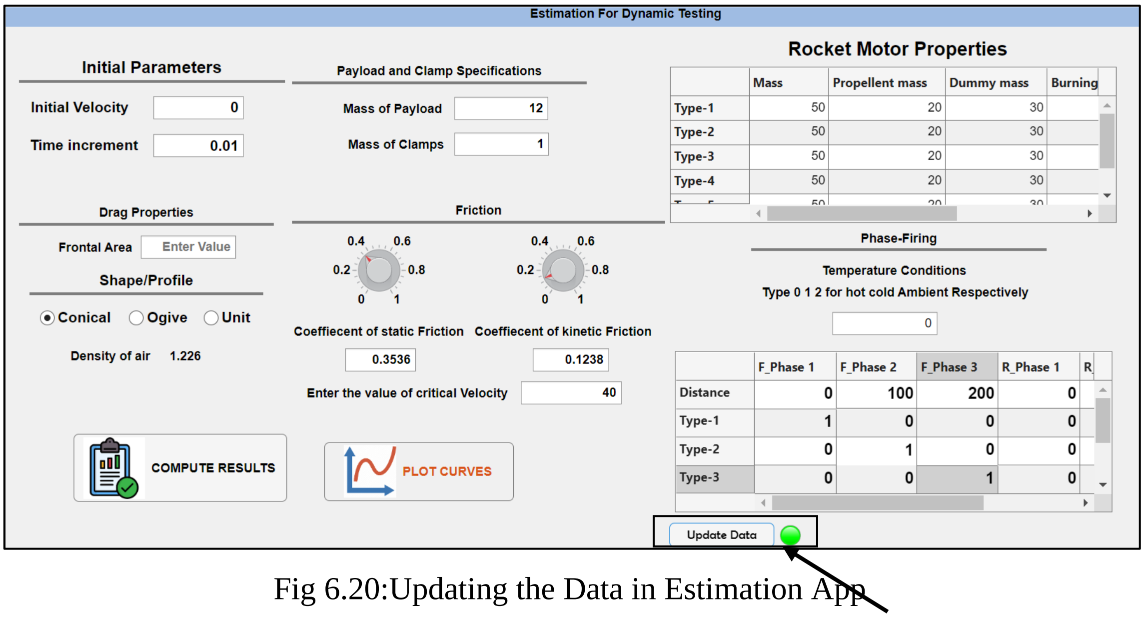


Fig 6.20:Updating the Data in Estimation App

Step 3: Press the "Compute Result" button to obtain output parameters. Use the "Plot Curves" button to visualize various curves.

Salient Features of MATLAB GUI Implementation

Error Handling: The MATLAB GUI incorporates robust error handling mechanisms to ensure smooth operation and user-friendly experience. In case a required input field is left empty or contains invalid data, the GUI promptly detects the error and triggers an alert or dialog box. This proactive approach helps users identify and rectify input errors promptly, minimizing the risk of inaccurate results and enhancing overall usability.

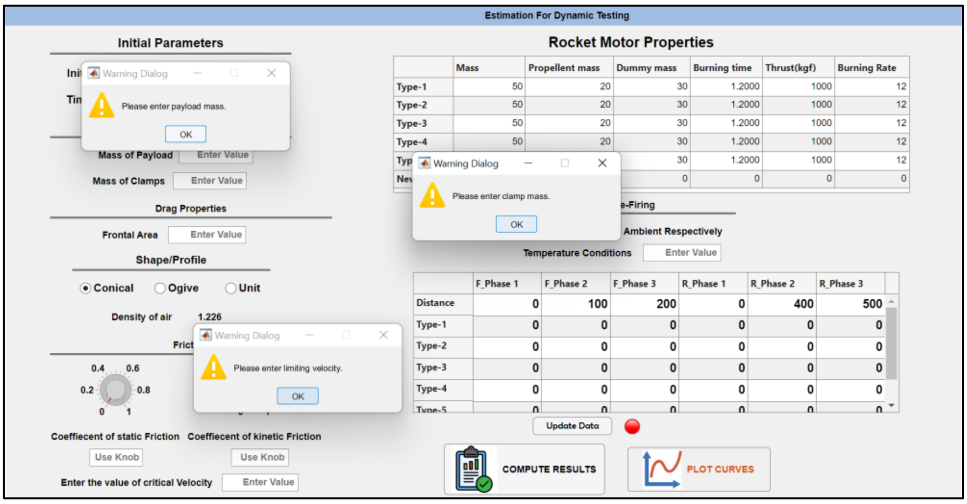


Fig 6.21: Error Handling in Estimation Programme

Multiwindow Application: Enhancing user experience and readability, the MATLAB GUI features a multiwindow application design. Upon initiating the simulation or analysis process, the output results are intelligently displayed in a separate window. This dedicated output window provides users with a clear and uncluttered view of the generated data, facilitating easier interpretation and analysis. By segregating input and output interfaces into distinct windows, the GUI streamlines the user interaction process, improving efficiency and workflow management.[19]

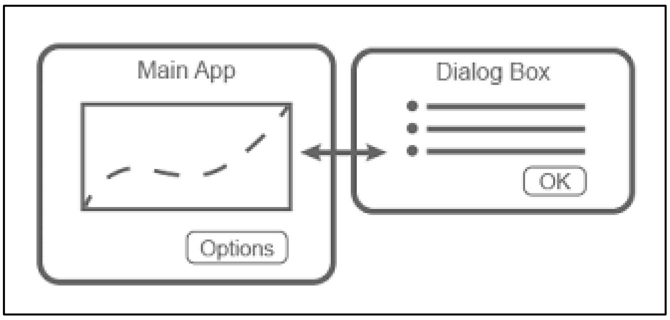


Fig 6.22:Multiwindow Functionality in App Designer

Future Scope of MATLAB GUI Development

The MATLAB GUI for dynamic testing and evaluation has laid a strong foundation, but there are several promising avenues for future development to further enhance its functionality and user experience:

- **Enhanced Rocket Input Options:** Future updates could include the ability to input and configure multiple rocket types within the GUI. Users would be able to specify parameters for each rocket, and the system would dynamically adjust the calculations and simulations based on these inputs. This feature would provide greater flexibility and accuracy in modeling various propulsion scenarios.
- **Data Export and Analysis:** Adding functionality to save test data and export it to Excel spreadsheets is a significant enhancement. This feature would allow users to download and analyse their results in external software, facilitating more detailed data analysis and reporting. The GUI could also include options to format the exported data according to user preferences, enhancing its utility for different applications.
- **Real-time Data Visualization and Reporting:** Incorporating advanced data visualization tools could allow users to view and analyze simulation results in real-time. Features such as interactive graphs, charts, and customizable dashboards could be added to improve the visualization of test results and make the data interpretation process more intuitive.

The successful development and implementation of this GUI underscore its transformative potential in revolutionizing testing practices across various engineering domains. By providing a user-friendly and efficient interface, the GUI streamlines the testing process, allowing for more accurate, comprehensive, and timely evaluations. This, in turn, supports engineers in making informed decisions and enhancing the overall quality of their work. The achievements and innovations presented by this project are highly commendable.

CHAPTER-7

OUTPUT PRESENTATION

7.1 Single-Phase Firing

Single-phase firing in rocketry refers to a propulsion strategy where all rocket engines are ignited at a specific point in time or distance and maintain their thrust characteristics (whether constant or variable) throughout the entire duration of the burn. In this method, the rocket experiences a continuous application of thrust, but the nature of this thrust can vary depending on the rocket design. Both rockets with constant thrust and rockets with variable thrust profiles can be used in a single-phase firing scenario.

Accessing Input Parameters for Single-Phase Firing

Properly defining and accessing the input parameters is crucial for accurately simulating the rocket's behavior. The key input parameters include the mass of the payload, drag properties, friction properties, and the thrust characteristics of the rocket engines.

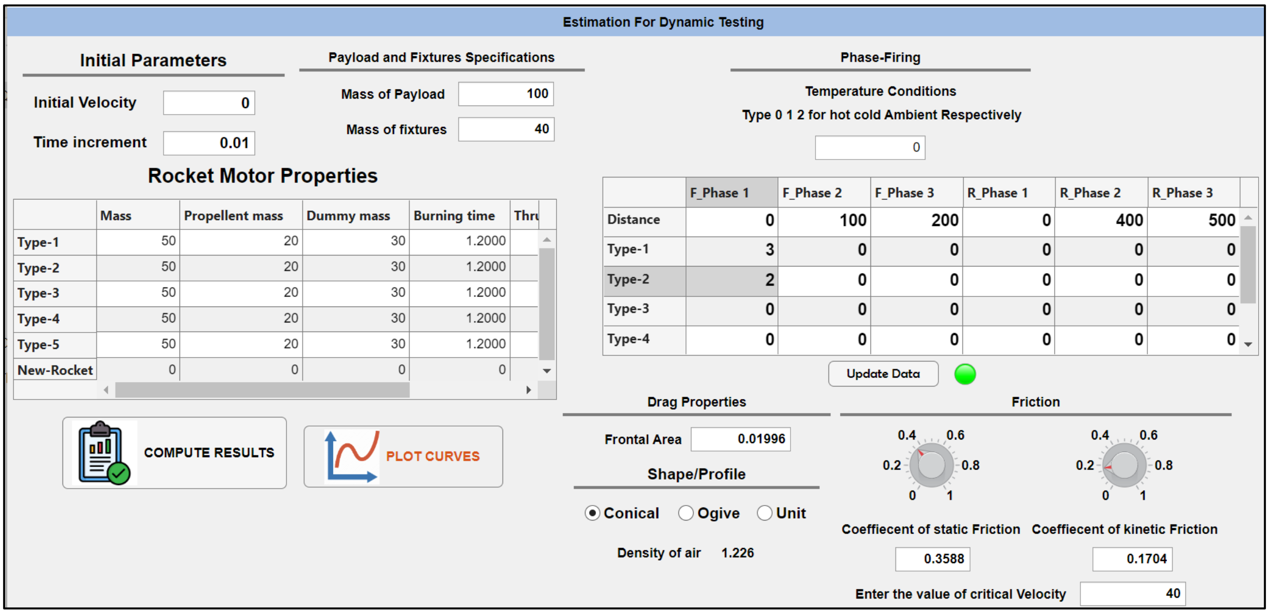


Fig 7.1: Input Parameters for Single Phase Firing

Initial Parameters: This container serves as the initial point of input, housing parameters fundamental to the estimation process's initiation

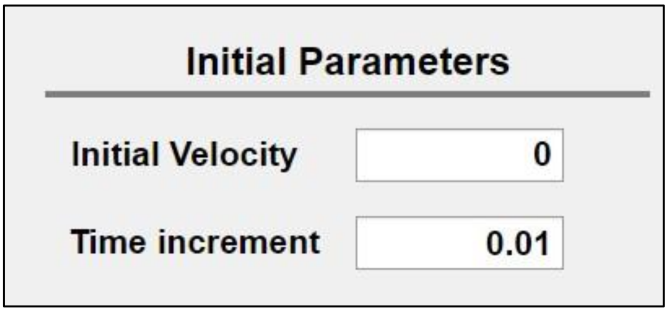


Fig 7.2: Single-Phase Input Details-1

Payload and Fixtures Specification: In this container, details pertaining to the payload and Fixtures specifications are captured, crucial for precise estimation calculations.

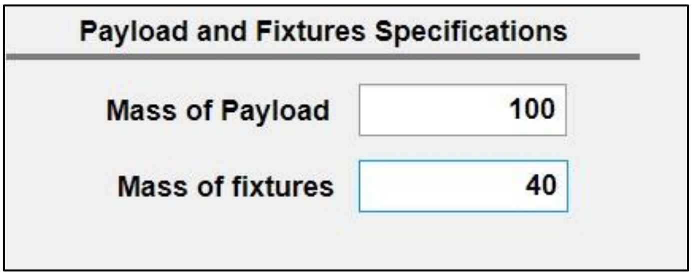


Fig 7.3: Single-Phase Input Details-2

Drag Properties: Within this container, inputs concerning drag properties are recorded, providing essential data for accurately modelling the drag forces acting upon the system.

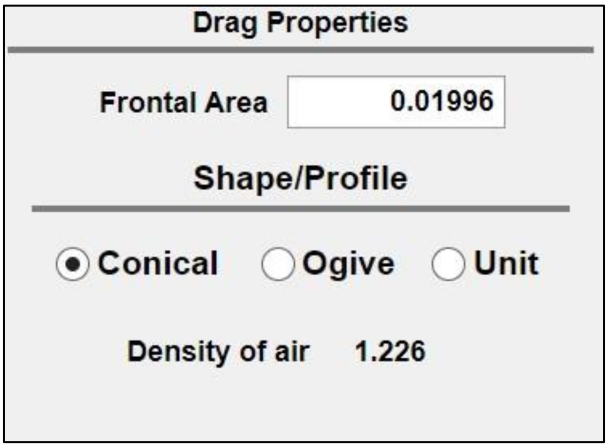


Fig 7.4: Single-Phase Input Details-3

Friction Properties: This section is dedicated to capturing the essential friction properties, which are crucial for accurately modeling frictional forces and their impact on the system's dynamics. The interface includes two knobs: one for adjusting static friction and the other for adjusting kinetic friction. Static friction represents the resistance to the start of motion, while kinetic friction represents the resistance during motion. Additionally, there is a field provided for specifying the limiting velocity, which defines the maximum velocity at which frictional effects are considered. This comprehensive input ensures that all relevant aspects of friction are incorporated into the simulation, leading to more precise and reliable results.

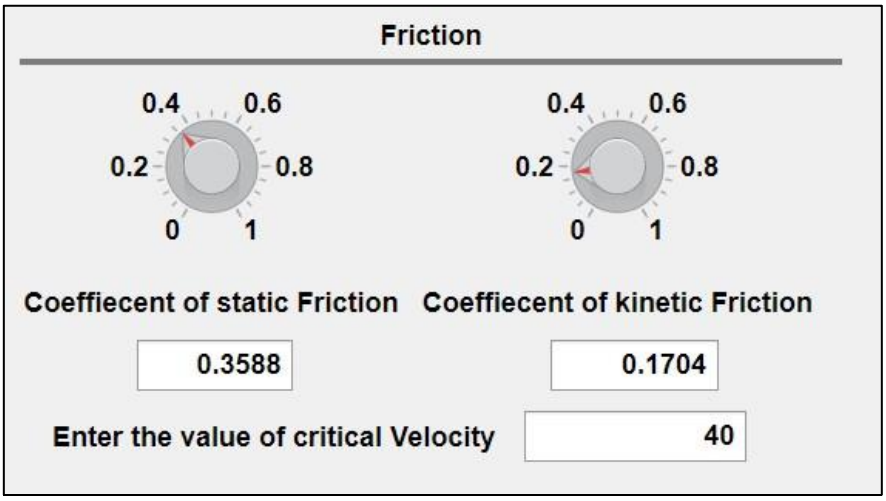
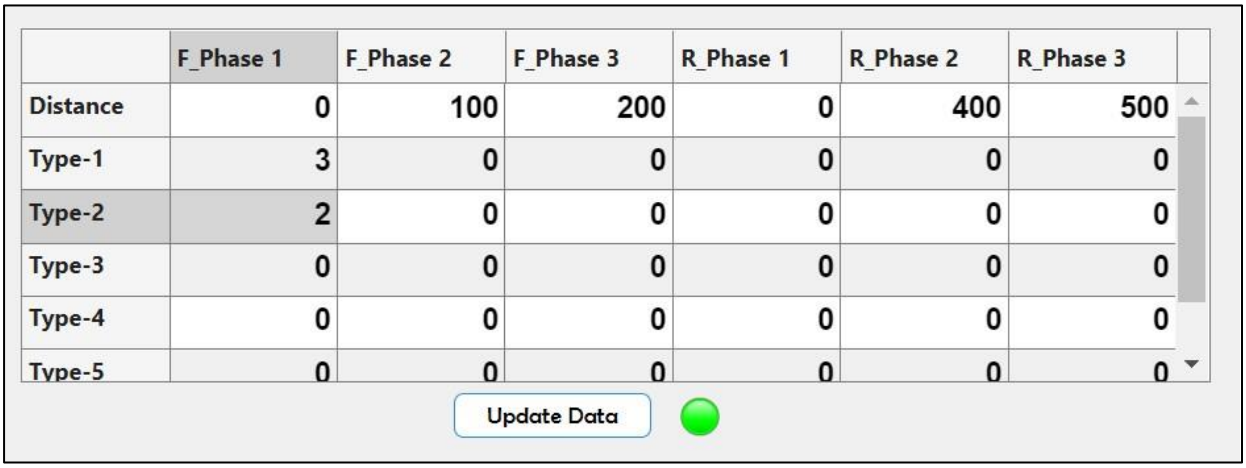


Fig 7.5: Single-Phase Input Details-4

Phase Firing UI Table: Here, phase firing data is meticulously organized, allowing users to input and manage data crucial for the estimation process. A dedicated "Update Data" button facilitates seamless data updates post-entry, ensuring data accuracy and integrity.



	F_Phase 1	F_Phase 2	F_Phase 3	R_Phase 1	R_Phase 2	R_Phase 3
Distance	0	100	200	0	400	500
Type-1	3	0	0	0	0	0
Type-2	2	0	0	0	0	0
Type-3	0	0	0	0	0	0
Type-4	0	0	0	0	0	0
Type-5	0	0	0	0	0	0

Fig 7.6: Single-Phase Input Details-5

Output Values for Single-Phase Firing

The output values from the single-phase firing simulation provide essential insights into the rocket’s performance during the entire propulsion phase. Key metrics include the maximum velocity achieved, along with the corresponding distance at which this velocity is reached, which helps in understanding the rocket’s peak speed and the efficiency of its flight path. The simulation also provides the maximum and minimum acceleration values, crucial for assessing the thrust dynamics and the forces acting on the rocket. Additionally, the maximum loading value, indicating the highest force experienced by the rocket, is vital for evaluating the structural integrity and safety of the rocket during its flight. These output values are fundamental for optimizing the propulsion strategy and ensuring the mission's success.

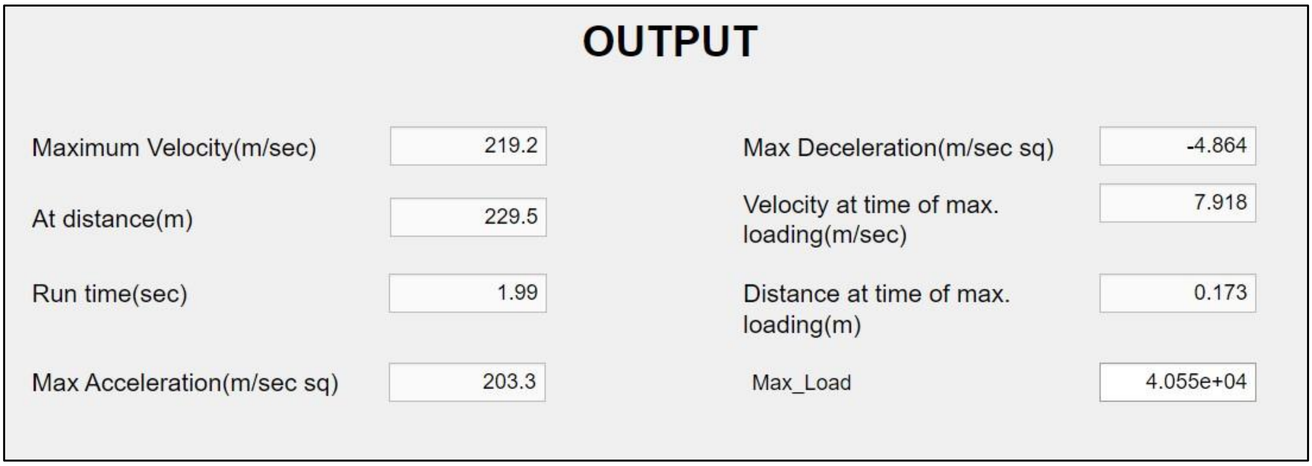


Fig 7.7: Results for Single-Phase Firing

7.2 Visual data presentation for Single Phase Firing

The graphical representation of results provides a comprehensive visualization of the rocket's performance under various conditions. By plotting key metrics such as velocity, acceleration, and distance over time, the simulation results are made clear and accessible. The graphs are crucial for understanding the dynamic behavior of the rocket, allowing for a detailed analysis of its motion.

Plot-1:Velocity Vs Time

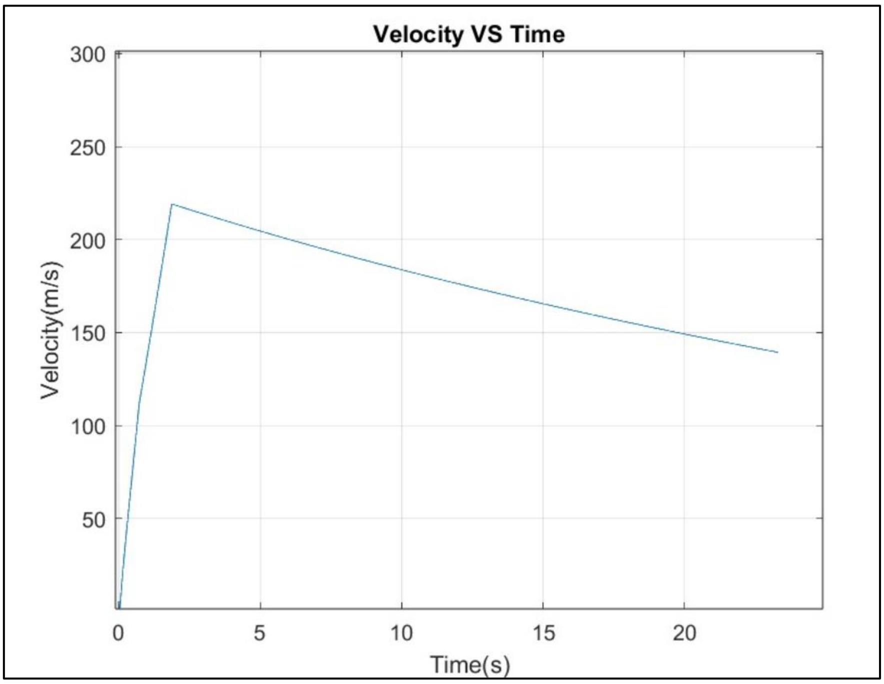


Fig 7.8: Plot of Velocity vs Time for single phase firing

Plot-2:Distance Vs Time

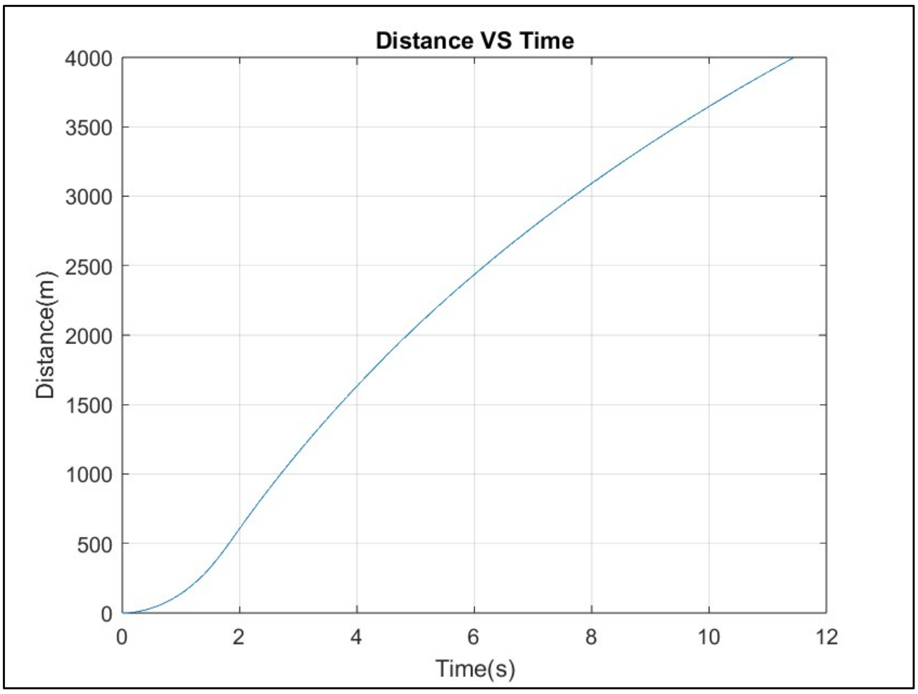


Fig 7.9: Plot of Distance vs Time for single phase firing

Plot-3:Velocity Vs Distance

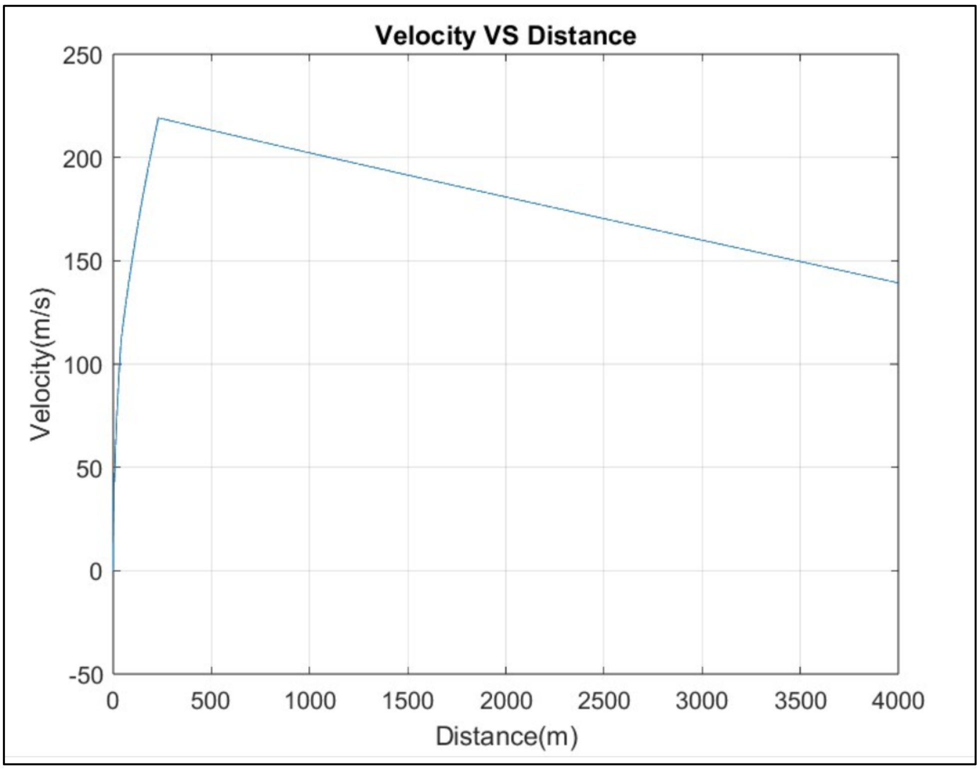


Fig 7.10: Plot of Velocity vs Distance for single phase firing

Plot-5:Thrust Vs Time

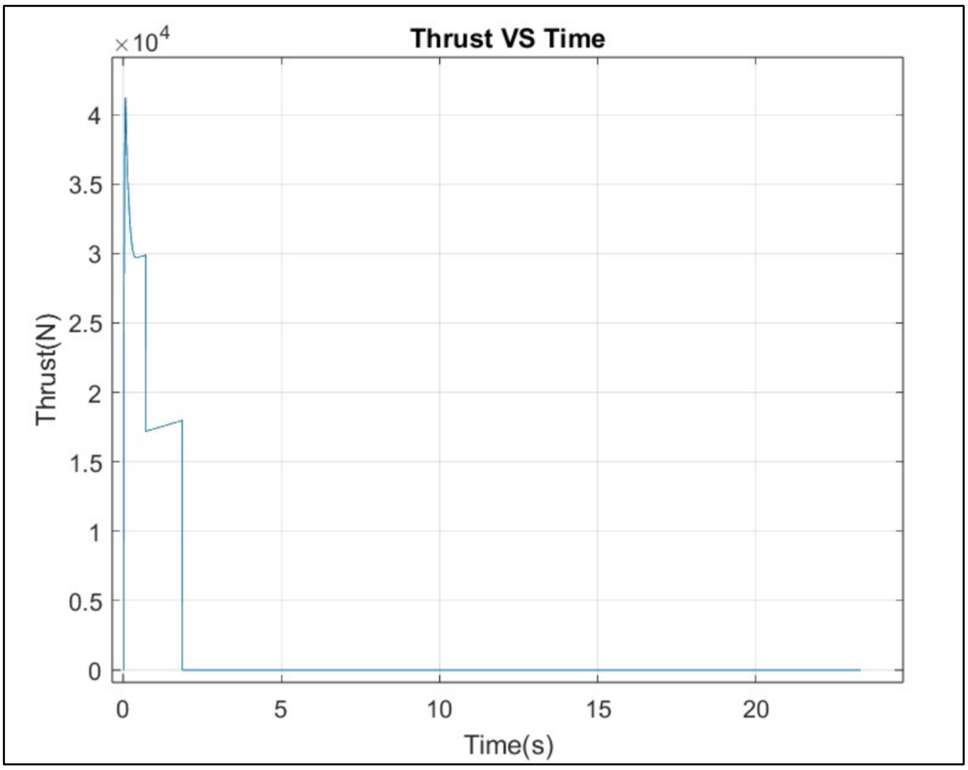


Fig 7.11: Plot of Thrust vs Time for single phase firing

Plot-6:Mass Vs Time

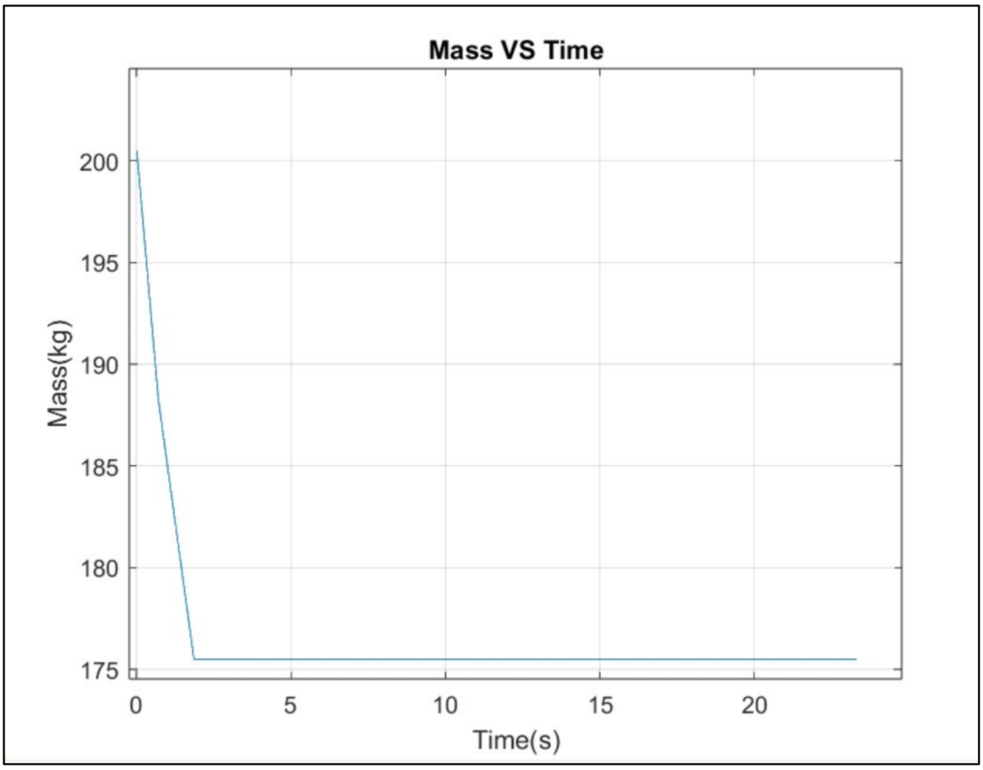


Fig 7.12: Plot of Mass vs Time for single phase firing

7.2 Multi-Phase Firing

Multi-phase firing in rocketry involves executing multiple stages of rocket propulsion, each characterized by distinct thrust profiles and operational conditions. In this approach, the rocket's engines are ignited and extinguished in specific sequences, allowing for varied thrust levels, burn durations, and ignition timings across different phases

Accessing Input Parameters for Multi-Phase Firing

In multi-phase firing simulations, it is crucial to accurately define and access several key input parameters to effectively model the rocket's performance through each phase. These parameters include the mass of the rocket, which encompasses the payload and fuel, as this influences the rocket's acceleration and overall dynamics across different phases. Drag properties, such as the drag coefficient, air density, and cross-sectional area, must be specified to account for aerodynamic resistance at various altitudes. Friction properties, including the friction coefficient, are also important, as they affect the rocket's motion, particularly during launch and ascent.

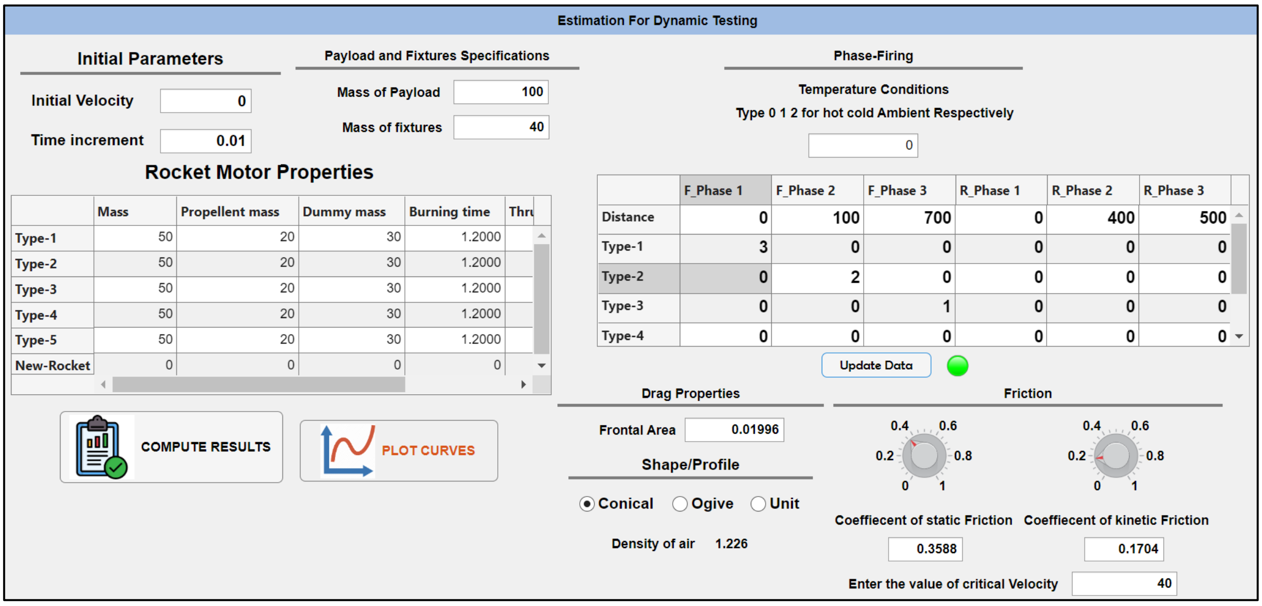


Fig 7.13: Input Parameters for Multi-Phase Firing

Initial Parameters: This container serves as the initial point of input, housing parameters fundamental to the estimation process's initiation

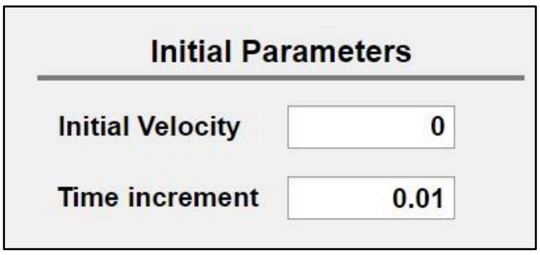


Fig 7.14: Multi-phase input details-1

Payload and Fixtures Specification: In this container, details pertaining to the payload and Fixtures specifications are captured, crucial for precise estimation calculations.

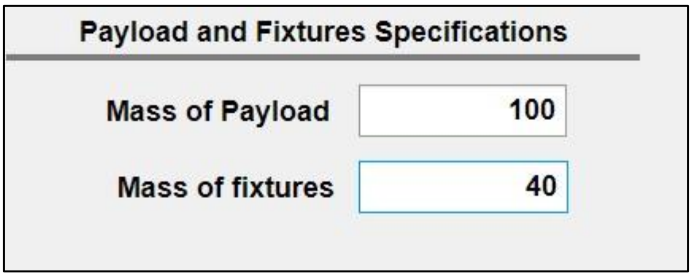


Fig 7.15: Multi-phase input details-2

Drag Properties: Within this container, inputs concerning drag properties are recorded, providing essential data for accurately modelling the drag forces acting upon the system.

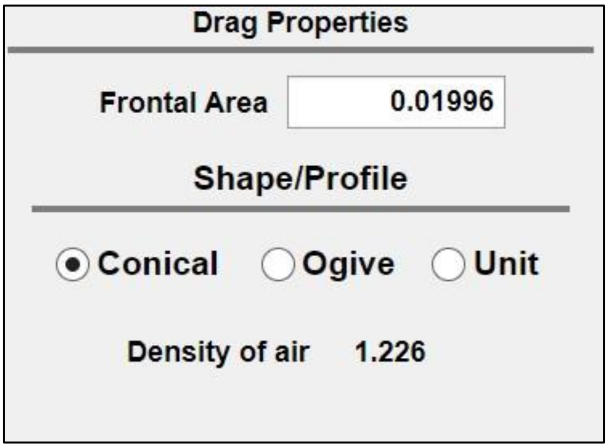


Fig 7.16: Multi-phase input details-3

Friction Properties: This section is dedicated to capturing the essential friction properties, which are crucial for accurately modeling frictional forces and their impact on the system's dynamics. The interface includes two knobs: one for adjusting static friction and the other for adjusting kinetic friction. Static friction represents the resistance to the start of motion, while kinetic friction represents the resistance during motion. Additionally, there is a field provided for specifying the limiting velocity, which defines the maximum velocity at which frictional effects are considered. This comprehensive input ensures that all relevant aspects of friction are incorporated into the simulation, leading to more precise and reliable results.

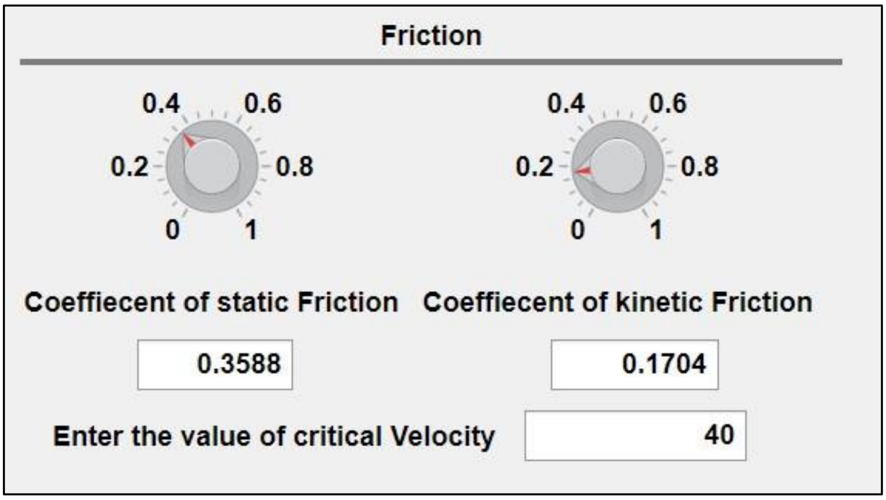
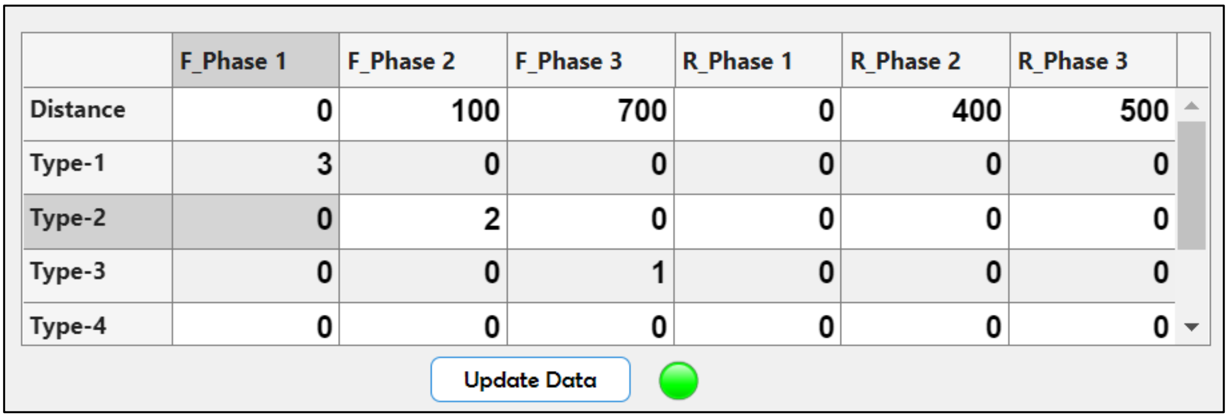


Fig 7.17: Multi-phase input details-4

Phase Firing UI Table: Here, phase firing data is meticulously organized, allowing users to input and manage data crucial for the estimation process. A dedicated "Update Data" button facilitates seamless data updates post-entry, ensuring data accuracy and integrity.



	F_Phase 1	F_Phase 2	F_Phase 3	R_Phase 1	R_Phase 2	R_Phase 3
Distance	0	100	700	0	400	500
Type-1	3	0	0	0	0	0
Type-2	0	2	0	0	0	0
Type-3	0	0	1	0	0	0
Type-4	0	0	0	0	0	0

Fig 7.18: Detailed View of Input Parameters-5

Output Values for Multi-Phase Firing

The output values from the multi-phase firing simulation provide critical insights into the rocket’s performance across different phases. Key metrics include the maximum velocity reached, along with the specific distance at which this velocity is attained, offering a measure of the rocket's peak speed and its trajectory efficiency. Additionally, the simulation yields the maximum and minimum acceleration values, which are essential for understanding the rocket’s thrust dynamics and the stress on its structure. These output values collectively help in evaluating the rocket’s performance, optimizing the propulsion strategy, and ensuring the mission’s success.

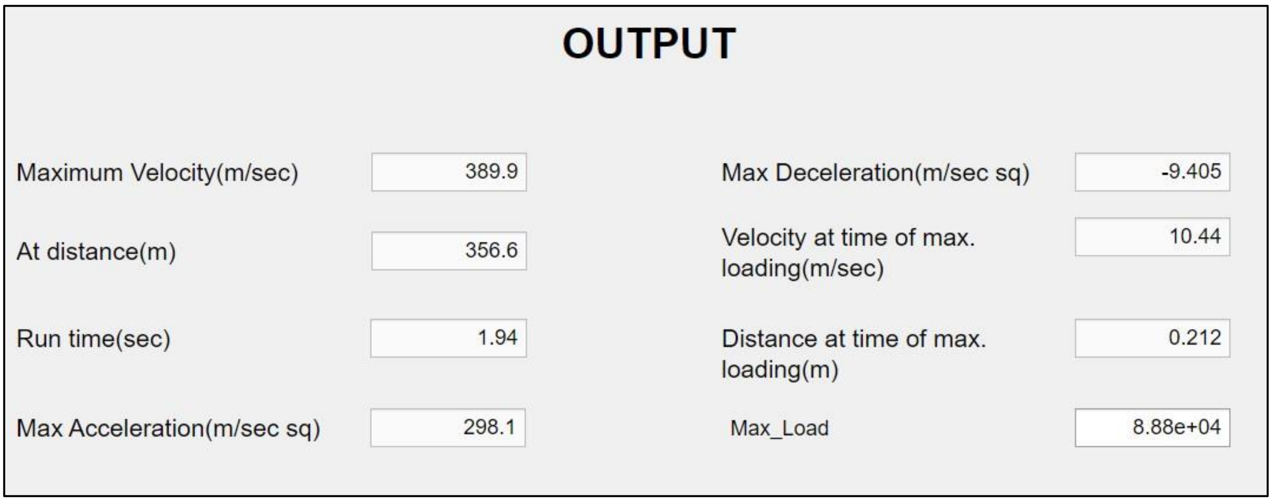


Fig 7.19: Results for Multi-Phase Firing

Visual data presentation for Multi- Phase Firing

The graphical representation of results provides a comprehensive visualization of the rocket's performance under various conditions. By plotting key metrics such as velocity, acceleration, and distance over time, the simulation results are made clear and accessible. The graphs are crucial for understanding the dynamic behavior of the rocket, allowing for a detailed analysis of its motion.

Plot-1:Velocity Vs Time

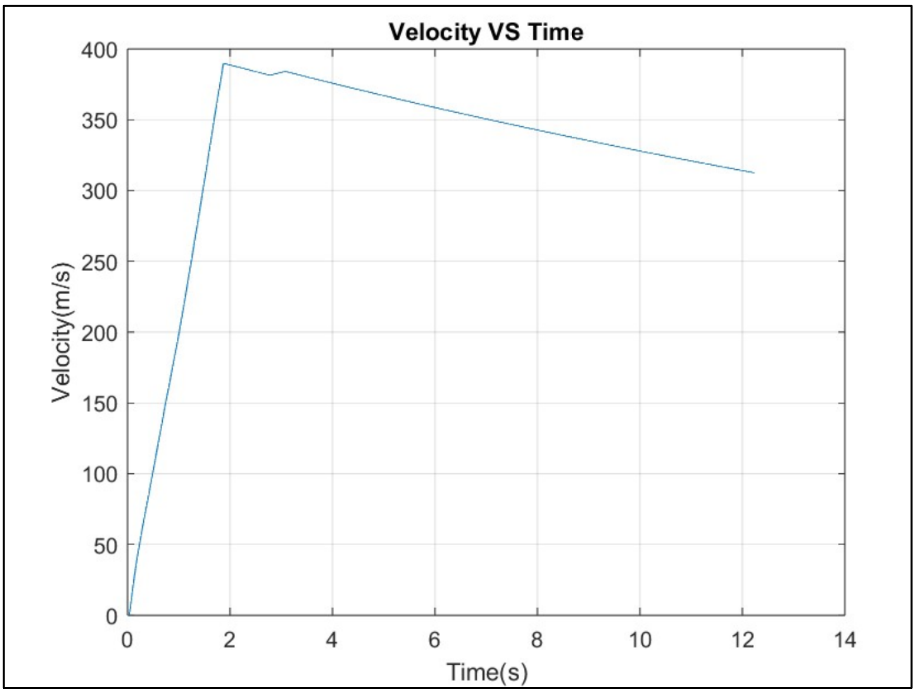


Fig 7.20: Plot of Velocity vs Time for Multi-phase firing

Plot-2:Distance Vs Time

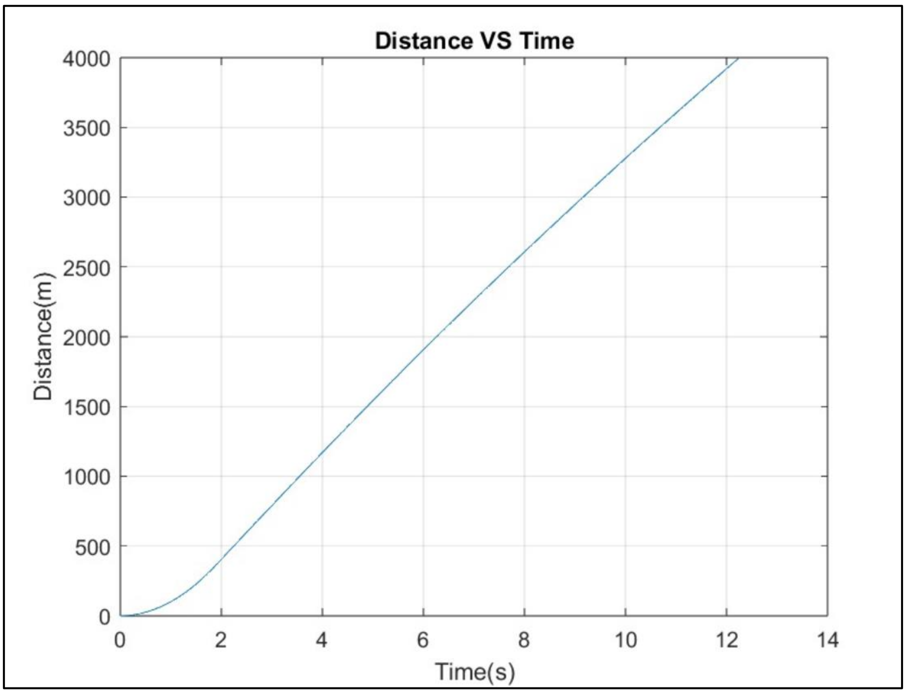


Fig 7.21: Plot of Distance vs Time for Multi-phase firing

Plot-3:Velocity Vs Distance

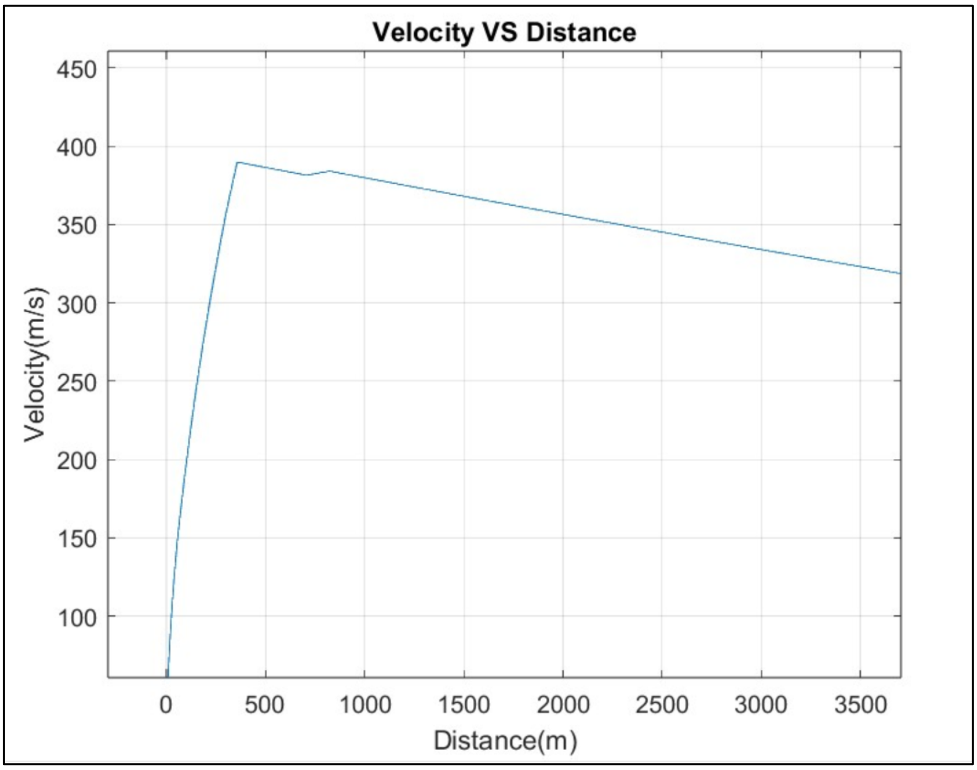


Fig 7.22: Plot of Velocity vs Distance for Multi-phase firing

Plot-5:Thrust Vs Time

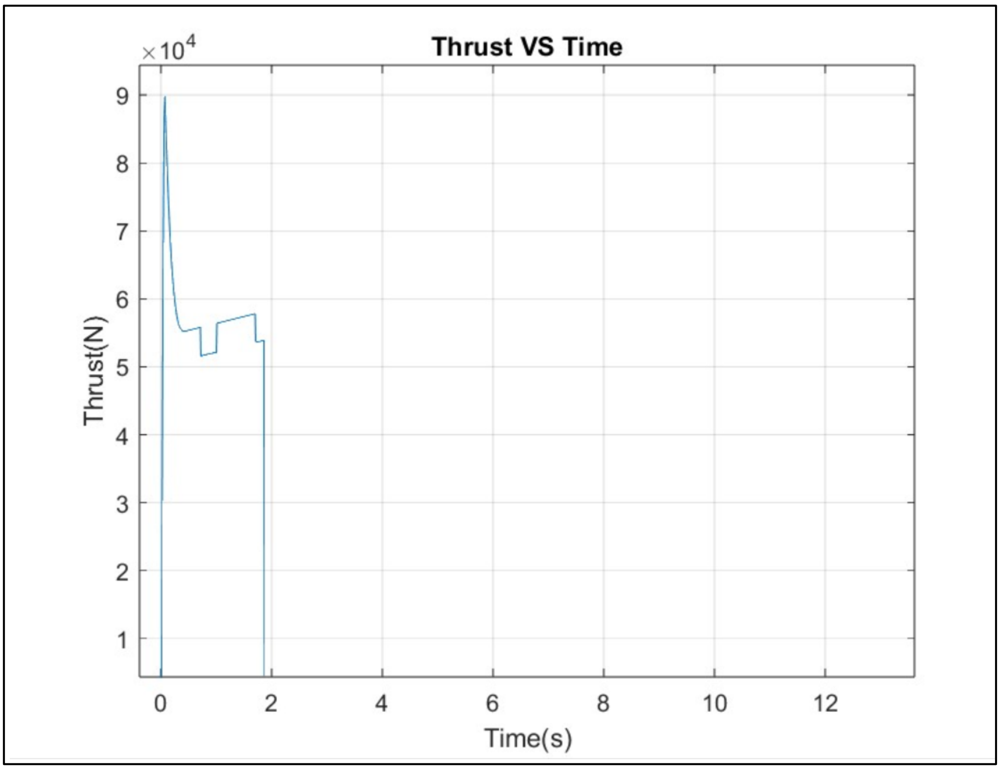


Fig 7.23: Plot of Thrust vs Time for Multi-phase firing

Plot-6:Mass Vs Time

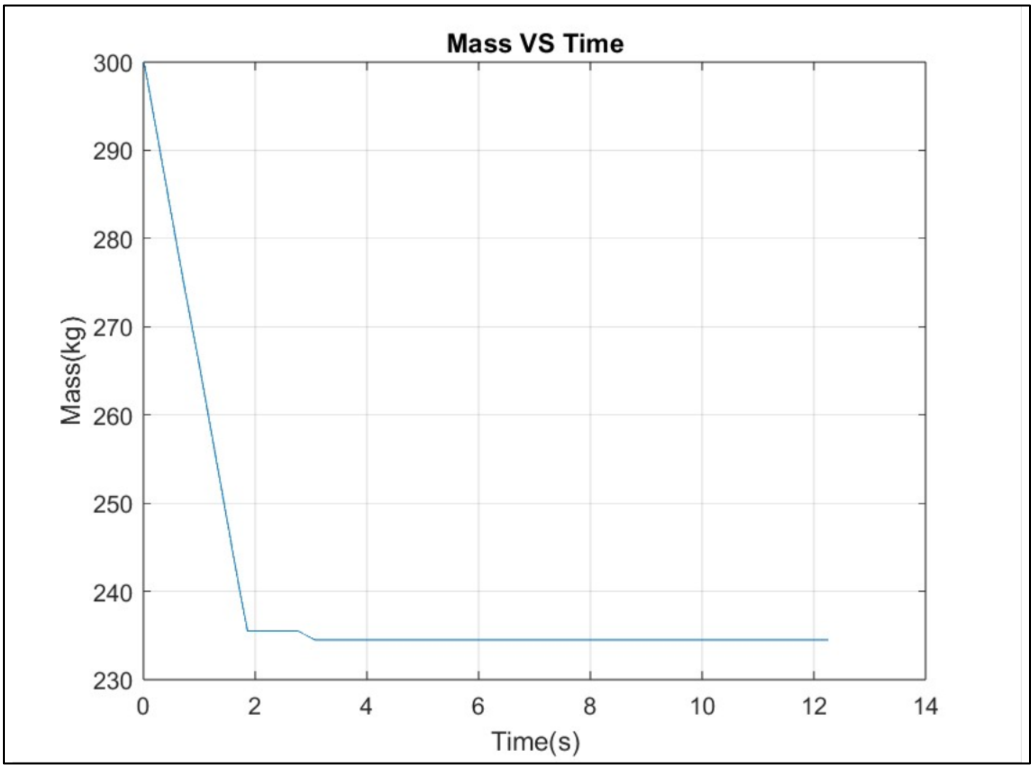
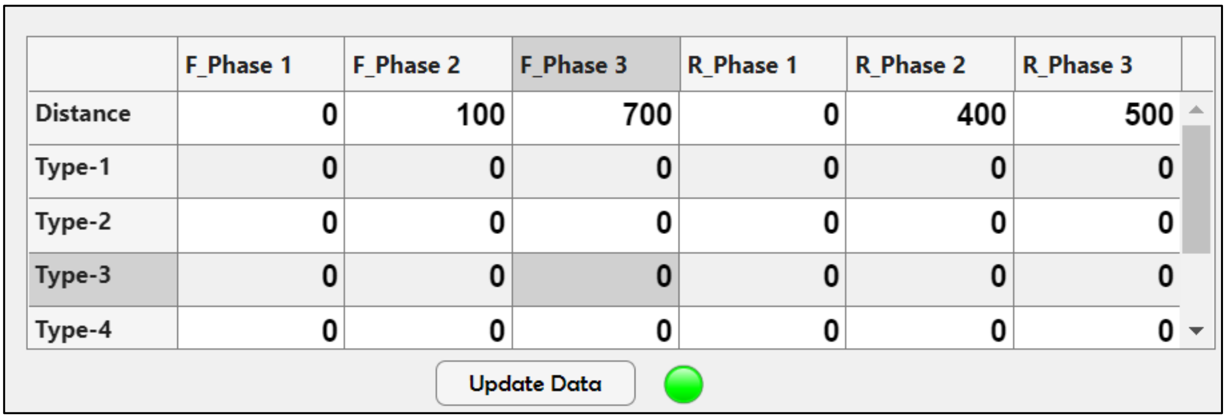


Fig 7.24: Plot of Mass vs Time for Multi-phase firing

Boundary Conditions and Special Cases

This section addresses the boundary conditions and special scenarios tested within the simulation to ensure its robustness and accuracy.

1. **No Rockets Fired:** When no rockets are fired, the simulation estimates the results with zero velocity and no error. This scenario serves as a baseline, confirming that the model correctly predicts a lack of motion when no propulsion force is applied. The output consistently shows zero velocity, indicating that the rocket remains stationary, aligning with theoretical expectations.



	F_Phase 1	F_Phase 2	F_Phase 3	R_Phase 1	R_Phase 2	R_Phase 3
Distance	0	100	700	0	400	500
Type-1	0	0	0	0	0	0
Type-2	0	0	0	0	0	0
Type-3	0	0	0	0	0	0
Type-4	0	0	0	0	0	0

Update Data

Fig 7.25: zero Rocket firing Condition

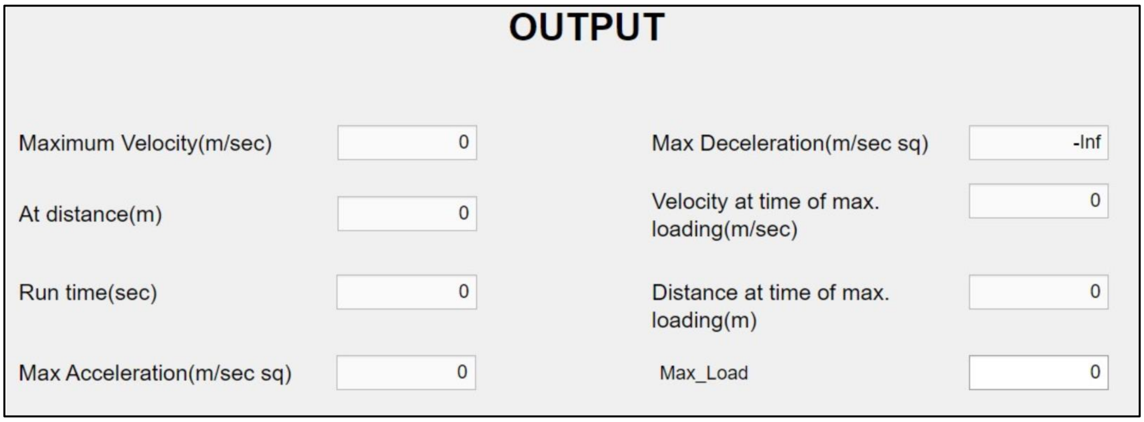


Fig 7.26: Output for zero Rocket firing Condition

2. **Opposing Rockets Fired at the Same Distance and Time:** When two rockets are fired simultaneously at the same distance but in opposite directions, the simulation accurately reflects this setup. Despite the opposing thrust forces, the estimation correctly calculates the resulting velocity and acceleration. This test demonstrates the model's ability to handle complex interactions between multiple forces, ensuring that the results remain accurate and reliable even when considering opposing propulsion forces. The outputs validate the simulation's precision in managing such special cases, showcasing its effectiveness in real-world scenarios where multiple forces may act in opposition.

	F_Phase 1	F_Phase 2	F_Phase 3	R_Phase 1	R_Phase 2	R_Phase 3
Distance	0	100	700	0	400	500
Type-1	1	0	0	1	0	0
Type-2	0	0	0	0	0	0
Type-3	0	0	0	0	0	0
Type-4	0	0	0	0	0	0

Update Data

Fig 7.27: Opposing Rockets Fired Simultaneously

OUTPUT

Maximum Velocity(m/sec)	0.04426	Max Deceleration(m/sec sq)	-Inf
At distance(m)	4.724	Velocity at time of max. loading(m/sec)	0
Run time(sec)	0	Distance at time of max. loading(m)	0
Max Acceleration(m/sec sq)	246.4	Max_Load	0

Fig 7.28: Output when Opposing Rockets Fired Simultaneously

CHAPTER-8

REFERENCES

[1] Defence Research and Development Organisation. "Defence Research and Development Organisation." https://www.drdo.gov.in/drdo/.

[2] Defence Research and Development Organisation. "Terminal Ballistics Research Laboratory." https://www.drdo.gov.in/drdo/labs-and-establishments/terminal-ballistics-research-laboratory-tbrl.

[3] "Terminal Ballistics Research Laboratory." *Defense Research and Development organization*, https://www.drdo.gov.in/drdo/labs-establishment/infrastructure-facilities/terminal-ballistics-research-laboratory-tbrl.

[4] Walia, Shilpa, Vikrant Satya, Sandeep Malik, and Subhash Chander. "Rocket Sled Based High Speed Rail Track Test Facilities." *Defence Science Journal* 72, no. 2 (May 2022): 182-194. https://doi.org/10.14429/dsj.72.17014.

[5] Indian Space Research Organisation. "Drogue Parachute Test." https://www.isro.gov.in/Drogue_Parachute_Test.html.

[6] Zhang, X., Li, Y., and Wang, Z. "Research on Deceleration Technology of Rocket Sled Double Track Sled." *Journal of Physics Conference Series* 2528, no. 1 (July 2023): 012059. https://doi.org/10.1088/1742-6596/2528/1/012059.

[7]Wikipedia. (n.d.). Wedge. Retrieved from https://en.wikipedia.org/wiki/Wedge#:~:text=A%20wedge%20will%20bind%20when,easily%20than%20a%20wide%20one.

[8] LibreTexts. (n.d.). Wedges. In Moore et al., Mechanics Map. Retrieved from https://eng.libretexts.org/Bookshelves/Mechanical_Engineering/Mechanics_Map_(Moore_et_al.)/06%3A_Friction_and_Friction_Applications/6.03%3A_Wedges

[9] EduRev. (n.d.). Wedge Friction - Friction - Engineering Mechanics. Retrieved from https://edurev.in/t/109871/Wedge-Friction-Friction--Engineering-Mechanics

[10] Blau, P. J. (2008). Friction Science and Technology: From Concepts to Applications, Second Edition. Dekker Mechanical Engineering.

[11] Beer, F. P., & Johnston, E. R. (2009). Vector Mechanics for Engineers: Dynamics.

[12] Beer, F. P., Johnston, E. R., DeWolf, J. T., & Mazurek, D. F. (2010). The Statics and Mechanics of Materials, 2nd Edition.

[13] Timoshenko, S., Young, D. H., & Rao, J. V. (2015). Engineering Mechanics.

[14] Lent, C. S. (Year of publication). Learning to Program with MATLAB: Building GUI Tools.

[15] MathWorks. (n.d.). Choose Components for Your App Designer App. Retrieved from https://in.mathworks.com/help/matlab/creating_guis/choose-components-for-your-app-designer-app.html

[16] GeeksforGeeks. (n.d.). Label component in MATLAB GUI. Retrieved from https://www.geeksforgeeks.org/label-component-in-matlab-gui/

[17] MathWorks. (n.d.). MATLAB GUI - Graphical User Interface. Retrieved from https://in.mathworks.com/discovery/matlab-gui.html

[18] Stack Overflow. "How to Do GUI Programming in MATLAB?" https://stackoverflow.com/questions/13539671/how-to-do-gui-programming-in-matlab.

[19] MathWorks. (n.d.). Creating Multiwindow Apps in App Designer. Retrieved from https://in.mathworks.com/help/matlab/creating_guis/creating-multiwindow-apps-in-app-designer.html

MathWorks. (n.d.). MATLAB Documentation. Retrieved from https://in.mathworks.com/help/matlab/