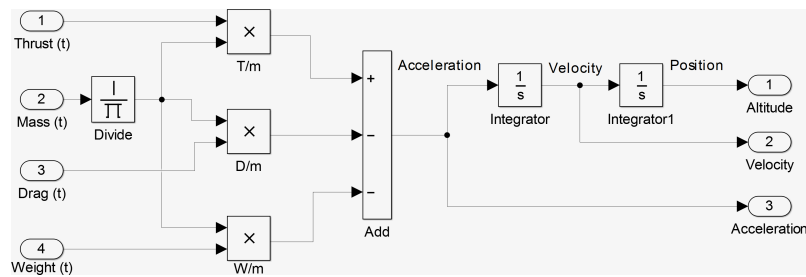




SPACE CONCORDIA - ROCKETRY DIVISION

AURELIUS CR-2-4G - STRUCTURAL TEAM



---

# Engineering Simulation for Rocket Flight Analysis

---

*Author(s):* Shawn Bulger

*Coordinator:* Dr. Ashok Kaushal  
*Supervisor:* Dr. Mehdi Hojjati  
*EIR:* Dominic Ng

March 29, 2016

### **Abstract**

A team from the Space Concordia association at Concordia University is entering a submission into the International Rocket Engineering Competition run by the Experimental Sounding Rocket Association (ESRA). To complement the team's mechanical design process, an Engineering Simulation is required to validate the flight performance criteria set by the competition. Matlab Simulink was used as an environment to model the flight dynamics and output the results. Thorough unit and integration testing was performed, in addition to comparison with experimental data to validate the model. Significant areas beyond the scope of the project are explored, such as a thorough stability analysis, preliminary considerations of a 6DOF simulator based on quaternion rotations, stochastic parameter variation to determine simulation uncertainties, and a generally in-depth review of literature and theory relating to high-powered rocket flight. The primary flight performance criteria of the CR\_2-4G rocket was confirmed to comply with all competition requirements and suggests the rocket is on course to maximize the team scores for performance. Future considerations are recommended for improvement and extension of the simulator.

## Contents

List of Abbreviations . . . . .	1
<b>Engineering Simulation for Rocket Flight Analysis</b>	<b>2</b>
Overview . . . . .	2
Definition of the Problem . . . . .	2
Requirements . . . . .	2
Validation . . . . .	2
Problem Solving Approach . . . . .	3
Kinematics . . . . .	3
Dynamics . . . . .	3
Decoupling the Model . . . . .	3
Assumptions . . . . .	3
<b>Input Parameters</b>	<b>3</b>
Static Parameters . . . . .	4
Dynamic Parameters . . . . .	4
Force . . . . .	4
Impulse . . . . .	4
Thrust . . . . .	4
Mass Flow Rate . . . . .	4
Weight . . . . .	6
Center of Gravity . . . . .	7
Moments of Inertia . . . . .	8
Center of Pressure . . . . .	9
<b>The Barrowman Method</b>	<b>11</b>
Stability Derivative . . . . .	11
Nose Cone . . . . .	11
Rocket Body . . . . .	11
Fins . . . . .	11
Nose Cone COP . . . . .	12
LV-Haack Nose Cone COP . . . . .	12
Von Karman Nose Cone COP . . . . .	12
Rocket Body Lift Correction . . . . .	13
Transonic Considerations . . . . .	13

<b>Atmospheric Model</b>	<b>13</b>
Introduction . . . . .	13
Methodology . . . . .	13
Aerodynamic Geometry . . . . .	15
Overview . . . . .	15
Surface Roughness . . . . .	15
Fineness Ratio . . . . .	15
Fins . . . . .	15
Aerodynamic Chord Length of Fins . . . . .	15
Areas . . . . .	15
Nose Profile . . . . .	16
Von Karman (Haack) . . . . .	16
Aerodynamic Center . . . . .	16
<b>Drag Model</b>	<b>16</b>
Mach Number . . . . .	17
Mach Regions . . . . .	17
Incompressible Flow . . . . .	18
Compressible Flow Correction . . . . .	18
Turbulent Effects . . . . .	18
Stagnation Pressure . . . . .	19
Reynolds Number . . . . .	19
Critical Reynolds Number . . . . .	19
Actual Reynolds Number . . . . .	20
Drag Force and Coefficients . . . . .	20
Viscous Drag Effects . . . . .	20
Pressure (Form/Profile) Drag . . . . .	21
Wave Drag . . . . .	24
Boat-Tail Drag . . . . .	24
Additional Drag at Angle of Attack . . . . .	25
Rocket Body Drag at Angle of Attack . . . . .	25
Alternatively . . . . .	26
Matlab Implementation . . . . .	26
Matlab Validation . . . . .	28
<b>Point-Mass Flight Model</b>	<b>29</b>
Weathercocking . . . . .	30
Altitude accounting for flight angle . . . . .	30
Drift accounting for flight angle . . . . .	30

<b>Rigid-Body Rotation (Pitch, Yaw) Stability Analysis</b>	<b>30</b>
Overview . . . . .	30
Longitudinal Static Stability Margin . . . . .	31
Requirement . . . . .	31
Assumptions . . . . .	31
Definition of Terms . . . . .	31
Rocket Normal Force . . . . .	31
Corrective Moment Coefficient . . . . .	32
Damping Moment Coefficient . . . . .	32
Derivation of the Harmonic Motion Equation . . . . .	33
Linearization Approximation . . . . .	34
General Homogeneous Response . . . . .	34
Complete Response to Step Input . . . . .	36
Complete Response to Impulse Input . . . . .	36
Delta-Dirac Function . . . . .	36
Convolution Theorem . . . . .	36
Steady State Response to Sinusoidal Forcing . . . . .	36
Rocket Damping Ratio . . . . .	36
Rocket Natural Frequency . . . . .	37
Time Constants of the Response . . . . .	37
Complete response to step input . . . . .	37
Complete response to impulse input . . . . .	37
AOA as a function of velocity . . . . .	37
Corrections . . . . .	37
Wind Disturbance . . . . .	38
Impulse Disturbance . . . . .	38
Constant Disturbance . . . . .	38
More Reading . . . . .	38
Solver Algorithm . . . . .	38
Algorithms . . . . .	38
Runge-Kutta-Fehlberg . . . . .	38
<b>Validation</b>	<b>38</b>
Unit Testing . . . . .	38
Integration Testing . . . . .	39
System Testing . . . . .	40

<b>Comparison with Experimental Data</b>	<b>40</b>
ISA Model . . . . .	40
Flight Validation . . . . .	40
University of Louisville . . . . .	40
Flight Details . . . . .	41
Motor Details . . . . .	42
Plots . . . . .	43
Comparison with Arcturus Rocket . . . . .	45
Summary of Comparison with Experimental Data . . . . .	45
<b>Simulation Execution</b>	<b>47</b>
Simulation Configuration . . . . .	47
Historical Weather Data for Green River, Utah . . . . .	47
General Conditions . . . . .	47
Best Case . . . . .	47
Worst Case . . . . .	47
Simulation Execution . . . . .	48
Matlab . . . . .	49
Observations . . . . .	50
Simulation Summary . . . . .	55
<b>Future Enhancements</b>	<b>56</b>
Hardware-in-the-loop . . . . .	56
Porting to Python / OpenModelica . . . . .	56
Plotting with Plot.ly . . . . .	56
Robust Wind Model . . . . .	56
ThrustCurve.org API Integration . . . . .	56
Rocket Orientation . . . . .	56
Rotations . . . . .	58
Quaternions . . . . .	58
Parameters needed for quaternion analysis . . . . .	59
Rocket Moments . . . . .	59
Pitch Moment . . . . .	59
Pitch Moment Coefficient . . . . .	59
Fin-set Pitch Moment Coefficient . . . . .	60
Body Tube and Fin Set Interference Pitch Moment Coefficient . . . . .	60
Barrowman Calculation . . . . .	60
Final Solver Pitch Moment Coefficient . . . . .	60
Pitch Damping Moment . . . . .	60
Stochastic Simulations . . . . .	60

<b>Conventions</b>	<b>60</b>
Data Model . . . . .	60
Static Parameters . . . . .	61
Dynamic Parameters . . . . .	61
Matlab Conventions . . . . .	61
Matlab/Simulink Libraries . . . . .	61
Overview . . . . .	61
Creating a Library . . . . .	61
Add to path . . . . .	61
Add to Library Browser . . . . .	61
Algebraic Loops . . . . .	61
Importing Data . . . . .	62
From File . . . . .	62
Versioning for Matlab Files . . . . .	62
Background . . . . .	62
Versioning for Libraries . . . . .	63
Unit Testing . . . . .	63
Exporting Images . . . . .	63
GhostScript . . . . .	63
GhostScript and GIMP . . . . .	63
Exporting Figures . . . . .	63
Exporting Simulink Models . . . . .	63
File Organization . . . . .	64
data . . . . .	64
documentation . . . . .	64
Naming Conventions . . . . .	64
Variables . . . . .	64
Functions . . . . .	65
Models . . . . .	65
Libraries . . . . .	65
<b>Documentation Conventions</b>	<b>65</b>
Markdown . . . . .	65
Python . . . . .	65
Pandoc . . . . .	65
Haskell . . . . .	65
LaTeX . . . . .	65
Citations . . . . .	66
Equations . . . . .	66
Figures . . . . .	66
Tables . . . . .	66
<b>References</b>	<b>67</b>

## List of Tables

2	List of Abbreviations . . . . .	1
4	Sample Motor Data . . . . .	5
6	Mach Regions . . . . .	17
8	Prandtl-Glauert Compressible Flow Correction Factors . . . . .	18
12	Summary of Comparison with Experimental Data . . . . .	45
14	Historical Weather Conditions, Green River, Utah . . . . .	47
16	General Simulation Conditions . . . . .	47
18	Best Case Simulation Conditions . . . . .	47
20	Worst Case Simulation Conditions . . . . .	47

## List of Figures

1	Sample Thrust Curve . . . . .	6
2	Dynamic Weight Calculation Test Output . . . . .	8
3	COP, COG, and I <sub>1</sub> as a function of time . . . . .	10
4	Rocket Drag Sources - Subsonic Flight . . . . .	17
5	Matlab Implementation of Parasitic Drag Coefficient . . . . .	23
6	Rocket Drag Forces - Axial vs. Normal Caption . . . . .	25
7	Rocket Drag Model . . . . .	26
8	Rocket Drag Coefficient Model . . . . .	27
9	Drag Force as a Function of Mach Number . . . . .	28
10	Vertical Flight Model - Simplified . . . . .	29
11	Angular Flight Model - Simplified . . . . .	35
12	All Integration Tests . . . . .	39
13	Comparison of ISA Model with SLC Weather Balloon Data . . . . .	40
14	Comparison of ISA Model with TUS Weather Balloon Data . . . . .	41
15	Altitude Plot of Simulation Data vs Louisville Rocket Launches . . . . .	43
16	Velocity Plot of Simulation Data vs Louisville Rocket Launches . . . . .	44
17	Altitude Plot of Simulation Data vs Arcturus Rocket Launch . . . . .	45
18	Velocity Plot of Simulation Data vs Arcturus Rocket Launch . . . . .	46
19	Full Model in Simulink, angle-of- attack less than 15 degrees . . . . .	49
20	Altitude as a Function of Time . . . . .	50
21	Mach Number as a Function of Time . . . . .	51
22	Stability (Calibers) as a Function of Time . . . . .	52
23	Angle of Attack Stabilization . . . . .	53
24	Natural Frequency . . . . .	54
25	Simulation Summary . . . . .	55
26	Forces and Moments Experienced by rocket in flight . . . . .	57



## List of Abbreviations

Abbreviation	Description	Function of	Units
AOA, $\alpha$	Angle of Attack		radians
COP	Center of pressure		N/A
COG	Center of gravity	time	N/A
Re	Reynolds Number	$\rho, \mu, \vec{v}, L$	dimensionless
$Re_{crit}$	Critical Reynolds Number	$\rho, \mu, \vec{v}, L$	dimensionless
$I_{zz}$	Pitch/Yaw Moment of Inertia	time	$m^4$
D	Drag Force (combined)		N
W	Weight of the Rocket		N
R	Specific Gas Constant		$Jkg^{-1}K^{-1}$
T	Thrust of the Rocket		N
$t_f$	Fin thickness	distance	m
$L_{cf}$	Aerodynamic Chord Length of Fins	distance	m
c	Speed of sound	$\sqrt{\gamma RT}$	
$R_a$	Surface Finish	distance	microns
M	Mach Number	$\vec{v}, c$	dimensionless
$D_{pa}, C_{pa}$	Parasitic Drag Force, Coefficient		
$D_{fb}, C_{fb}$	Body Drag Force, Coefficient		
$D_{fp}, C_{fp}$	Fin Pressure Drag Force, Coefficient		
$D_{pr}, C_{pr}$	Pressure Drag Force, Coefficient		
$D_{in}, C_{in}$	Interference Drag Force, Coefficient		
$D_{ba}, C_{ba}$	Base Drag Force, Coefficient		
$D_{sk}, C_{sk}$	Skin Friction Drag Force, Coefficient		
$D_{aoa}, C_{aoa}$	Additional Angle of Attack Drag Force, Coefficient		
$C_{MC}$	Corrective Moment Coefficient		
$C_{FN}$	Normal Force Coefficient		
$C_{PDM}$	Propulsive Damping Moment Coefficient		
$C_{ADM}$	Aerodynamic Damping Moment Coefficient		
$A_{wb}$	Area of Wetted Body		$m^2$
$A_{wf}$	Area of Wetted Fins		$m^2$
$A_{fr}$	Frontal Reference Area		$m^2$
$A_{fp}$	Fin Planform Area		$m^2$
$A_{fe}$	Exposed Fin Planform Area		$m^2$
OD, $\phi_{bt}$	Outer Diameter		m
L	Total Length of Rocket		m
h_n	Height of the nose cone		m
$S_{fc}$	Thrust Specific Fuel Consumption		$\frac{g}{s} \cdot \frac{1}{N} = \frac{s}{m}$
$\dot{m}_{fc}$	Mass Flow Rate due to Fuel Consumption		$\frac{g}{s} \cdot \frac{1}{N} = \frac{s}{m}$
$T_{avg}$	Average Thrust		N
$t_{burn}$	Burn Time		s
$m_{m_t}$	Total Motor Mass		g
$W_{m_t}$	Total Motor Weight		N
$F_N$	Aerodynamic Normal Force		N
$F_A$	Aerodynamic Axial Force		N
$F_L$	Aerodynamic Lift Force		N
$S_{lm}$	Longitudinal Stability Margin		Calibers
$f_B$	Fineness Ratio		dimensionless
$\mu$	Dynamic Viscosity		$Ns/m^2$
$\nu$	Kinematic Viscosity	$\mu, \rho$	$m^2/s$
$\lambda$	Angular Acceleration		$rad/s^2$
$\omega$	Angular Velocity		$rad/s$
$\theta$	Angular Position		radians

Table 2: List of Abbreviations

# Engineering Simulation for Rocket Flight Analysis

## Overview

The goal of this project is to create an Engineering Simulation for Rocket Flight Analysis in Matlab. This is not exactly a flight simulator, which generally aims to train pilots and visually simulate aircraft flight. An engineering simulation tests the dynamics and behavior of a system, often employing a combination of analytic and empirical methods, in order to validate an engineering model before its deployment.

Such a simulation is driven by the requirements of the engineering project. This project is constrained by the flight requirements of an international rocketry competition, and as such, it is developed to validate them.

A modular development pattern is performed where possible, in order to support expansion for other simulation purposes. Unit and integration testing of simulator logic is undertaken where reasonable, and further validation is provided by testing the overall model against available 3rd party flight data.

Beyond the primary goal of validating the rocket flight performance through simulation, this project is an educational tool to enhance the knowledge and learning of the team members and of the *Space Concordia* society as a whole. Furthermore, it will serve as a starting point for future controls and simulations applications to come, many of which are discussed in the **Future Enhancements** section.

## Definition of the Problem

An engineering simulation is needed to predict the flight performance of a high-powered rocket. A high-powered rocket is defined as having between 160 Ns and 40,960 Ns total impulse [1]. A team from the *Space Concordia* association at *Concordia University* is entering a submission into the *International Rocket Engineering Competition* (IREC) run by the *Experimental Sounding Rocket Association* (ESRA). The competition provides performance targets which must be met in order to be eligible to win. These targets are held as design requirements for performance, and are listed in the **Requirements** section below.

## Requirements

The Performance Model must provide the maximum altitude and velocity of the rocket in subsonic flight under a known thrust curve and known dimensional parameters.

- 2a Static stability above 2 calibers
- 2b Dynamic stability above 0
- 2c Min velocity at launch rail 30.5 m/s
- 2d Vehicle max speed mach 0.9
- 2e Vehicle reaches 10,000 ft altitude (+1000 feet / - 0 feet)
- 2f Vehicle doesn't experience resonant pitching/yawing moment

[SCRD 2016 Specifications and Requirements]

## Validation

- Sub-Models will be unit tested with 3rd party data to ensure functional validity where possible.
- Once all Sub-Models are validated individually, the overall model will be validated using 3rd-party data.
  - Primary 3rd-party simulation data source will be OpenRocket simulations
  - Where possible, actual 3rd-party rocket flight data will be used to validate the model

## Problem Solving Approach

### Kinematics

*Kinematics* is the study of the motion without consideration of the forces in play. This analysis can be simplified by considering the entire mass of a body at a convenient point, for instance the *Center of Gravity*. We call this the point-mass system. Of particular interest are the position, velocity, and acceleration of the point [2].

### Dynamics

*Dynamics* is the study of motion which considers the forces in play. It is useful to consider bodies as *rigid bodies* in order to simplify the forces that act on them (e.g. ruling out the stiffness of the body) [2]. Forces acting at a distance  $d$  from the center of gravity create a rotation about the center of gravity.

### Decoupling the Model

If it can be said that the translation of the body at its center of gravity (kinematics) does not impact the rotation of the body about its center of gravity by a force applied at some other point (the *Center of Pressure*), the analysis can be greatly simplified. Such a *mutually independent system of equations* could be solved rather easily, as the translational model could be solved independently of the rotational model, and vice-versa [2]. Coupled systems of equations, on the other hand, would have to be solved simultaneously.

As it happens, the translational model of the rocket is not fully decoupled from the rotational model. As the rocket rotates while travelling through the air at an angle of attack, it presents a larger reference area and thus increases the drag force, causing a negative acceleration in the translational point-mass model. Likewise, an increase in the translational velocity of the rocket creates a larger lift force when the rocket is rotating at an angle of attack. This lift force, which is applied at the rocket's center of pressure, creates a rotation about the center of gravity.

However, it is possible to proceed with a decoupled analysis with the assistance of careful assumptions and approximations within acceptable accuracy. If accepted, this would imply a *weak coupling* between the translational model and the rotational model [2].

### Assumptions

- subsonic flight
- axis-symmetric rigid body rocket
- single cylindrical body
- Von Karman nose shape
- three or four trapezoidal fins
- passively controlled (no active thrust or stability control)
- constant fuel expenditure rate
- vertical/linear flight within 5 degrees [3]
- the *Ideal Gas Law* applies throughout the flight
- humidity in the air is ignored
- steady-state irrotational flow around the body [4]
- fully aligned thrust [3]
- smooth transition between nose cone and body tube (no shoulder)
- rocket does not have a boattail
- rocket has a single rectangular launch lug

## Input Parameters

Before beginning the modeling process, it is necessary to understand the inputs of the system.

Many parameters can be considered unchanging during flight, and are from this point referred to as *Static Parameters*. Other inputs change as a function of time or velocity, and must be carefully handled within the system - these parameters are herein referred to as *Dynamic Parameters*.

## Static Parameters

Many rocket design parameters are considered to remain constant during flight.

These parameters are written in the parametric spreadsheet shared with the design team. The CAD software populates values related to the structural design, and others are entered manually. The performance model reads them at the beginning of execution to simulate the latest design iteration.

Find the included file named 'Parametric\_Model.xlsx'

## Dynamic Parameters

Parameters listed as *dynamic* in the table above are provided as initial values which are then recalculated by the model throughout the simulated flight.

### Force

*Force* is a change in momentum with time, and is related by Newton's Second Law

$$F = \frac{m\Delta\vec{v}}{\Delta t} = m\vec{a} \quad (1)$$

### Impulse

*Impulse* is the product of force and integration of a differential (infinitesimal period) of time between the time periods in which it was applied.

$$J = \int_{t_1}^{t_2} F dt \quad (2)$$

This is also known as the *Total Impulse*, or the *total change in momentum*, and can be calculated as the average thrust over a given time period.

### Thrust

Thrust is the mechanical force that drives the flight of the rocket. It is a vector quantity of magnitude and direction. *Thrust* is a reaction force in the opposite direction of accelerating fluid (exhaust gas) caused by the combustion of fuel, and is assumed to be aligned with the longitudinal axis of the rocket.

### Mass Flow Rate

*Mass Flow Rate* is found by the product of fluid density, velocity, and cross-sectional area.

$$\dot{m} = \rho\vec{v}A \quad (3)$$

[5]

## Thrust Equation

Thrust is a mechanical force created by a propulsion system which moves a vessel through a medium. In high-powered rocketry, it is typically created by the expulsion of hot gas generated during the ignition of a solid fuel motor. The acceleration of gases from the motor causes a thrust force in the opposite direction (according to Newton's 2<sup>nd</sup> law), which propels the rocket from the launch pad. The thrust force is the product of the mass flow rate of the hot gases and the change in velocity between the exit of the nozzle and the free stream.

$$T = \dot{m}\Delta\vec{v} \quad (4)$$

[5]

Thrust curves are provided by the manufacturer, based on static tests. The highest performing result from the batch of test motors is chosen for the curve. Therefore, the thrust curve represents the highest expected performance of the motor within the specified operating conditions

Table 4 shows an example of the motor data provided by ThrustCurve.org:

Parameter	Value
Manufacturer	Cesaroni Technology
Entered	May 20, 2009
Last Updated	Jun 26, 2014
Mfr. Designation	6819M1540-P
Common Name	M1540
Motor Type	reload
Delays	P
Diameter	75.0mm
Length	75.7cm
Total Weight	5906g
Prop. Weight	3624g
Cert. Org.	Canadian Association of Rocketry
Cert. Designation	6819-M1540-IM-P
Cert. Date	
Average Thrust	1537.0N
Maximum Thrust	2328.8N
Total impulse	6819.4Ns
Burn Time	4.4s
Case Info	Pro75-5G
Propellant Info	Imax
Availability	regular

Table 4: Sample Motor Data

Source: <http://www.thrustcurve.org/motorsearch.jsp?id=673>

## Thrust Specific Fuel Consumption

*Thrust Specific Fuel Consumption* is how much fuel is burned for a given time.

$$S_{fg} = \frac{m}{t_{burn}} \cdot \frac{1}{T_{avg}} \quad (5)$$

$$\left[ \frac{g}{s} \cdot \frac{1}{N} = \frac{s}{m} \right] \quad (6)$$

Since at the time of writing the  $S_{fc}$  was not provided by the manufacturer, the following calculations are used for a first approximation.

Assumptions

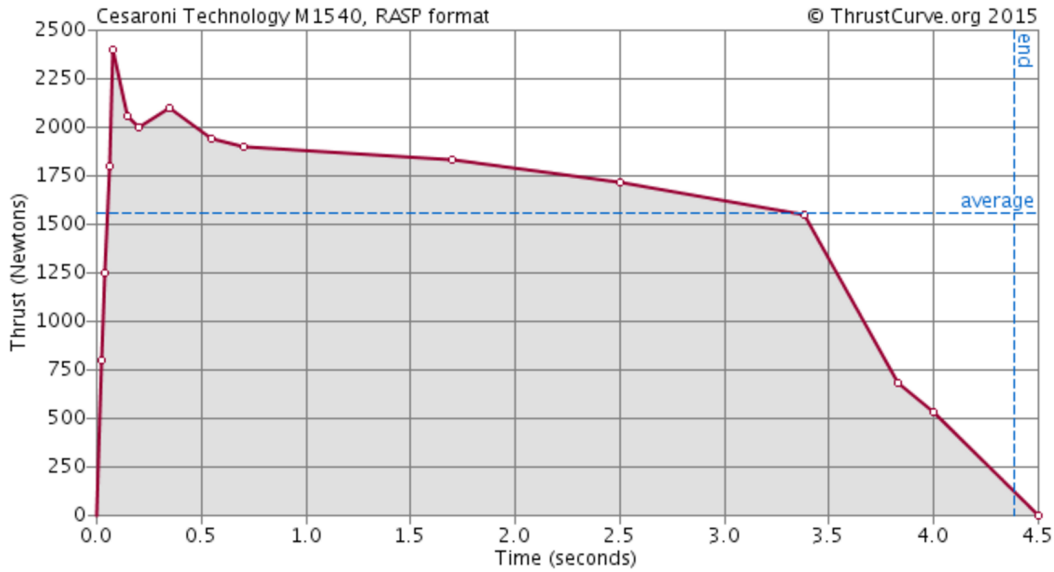


Figure 1: Sample Thrust Curve

- all propellant is spent during the motor burn time
- final  $S_{fg}$  determined is constant during burn
- the motor info provided is accurate

It should be noted that a variance in thrust of  $\pm 20\%$  is possible. This and other variance factors are taken into account in the *Statistical Analysis* section.

From the table above, the dry propellant weight is given as 3624 grams. The Average Thrust is given as 1537.0 Newtons, and the total burn time is given as 4.4 seconds.

Thus, the *Thrust Specific Fuel Consumption* can be determined as follows:

$$S_{fg} = \frac{3.624 \text{ kg}}{4.4 \text{ s}} \cdot \frac{1}{1537.0 \text{ N}} \approx 0.00053587 \frac{\text{kg}}{\text{N} \cdot \text{s}} = 5.3587 \times 10^{-4} \frac{\text{kg}}{\text{N} \cdot \text{s}} \quad (7)$$

This rate is considered constant.

## Weight

As fuel is expended in generating thrust, the weight of the rocket is reduced. One assumption that can be made is that the change of mass with time is “proportional to the impulse of the motor up to that point” [3].

$$\Delta M_i = -\frac{M_f \int_0^i T dt}{\int_0^\infty T dt} \quad (8)$$

Where:

- $M_f$  is the total mass of fuel
- $T$  is the thrust
- $\Delta M_i$  is the change in mass of fuel between time  $i$  and  $t$

[3]

We can also use the *Thrust Specific Fuel Consumption* to determine the corresponding reduction in weight during burn, if we assume that the mass flow rate of fuel during burn is constant.

First remove the Average Thrust term to isolate the mass flow rate:

$$\dot{m} = S_{fG} \cdot T_{avg} = 5.3587 \times 10^{-4} \frac{kg}{N \cdot s} \cdot 1537.0 N = 0.8236 kg/s \quad (9)$$

This equation can be expressed in terms of Weight through Newton's 2<sup>nd</sup> law:  $F = m\vec{a}$

$$\dot{W}_m = \dot{m} \cdot \vec{g} = 0.8236 kg/s \cdot 9.81 m/s^2 \approx 8.0799 N/s \quad (10)$$

To develop a relation for the change in weight as a function of  $S_{fc}$

$$W_f(t) = (m_{fi} kg - \Delta m(t)) \cdot \vec{g} \quad (11)$$

$$W_f(t) = (3.624 kg - \Delta m(t)) \cdot 9.81 m/s^2 \quad (12)$$

$$W_f(t) = W_{fi} - \Delta W_f(t) \quad (13)$$

$$\Delta W_f(t) = \int \dot{W} dt = \dot{W} \cdot t \quad (14)$$

$$\Delta W_f(t) = \frac{\Delta m(t) \cdot \vec{g}}{t} \cdot t = \Delta m(t) \cdot g \quad (15)$$

$$\Delta m_f(t) = S_{fg} \cdot t \quad (16)$$

Finally, the motor weight as a function of time is

$$W_m(t) = W_{m_i} - \Delta W_f(t) = W_{m_i} - \dot{m}_{fc} \cdot t \quad (17)$$

As a further simplification, if we assume that the mass flow rate is constant, we can find it simply by dividing the mass propellant by the motor burn time. This results in only a slight error as shown in Figure 2 below.

Figure 2 shows the output of the dynamic weight calculation in Matlab. The Thrust and Weight curves produce output similar to OpenRocket as expected.

## Center of Gravity

The *Center of Gravity* is a location where we can consider the entire rocket mass to be concentrated in a point-mass system.

$$y_{cg} = \frac{m_1 y_1 + m_2 y_2 + \dots + m_n y_n}{\sum_{j=1}^n m_j} \quad (18)$$

$$COG(t) = \frac{m_1 y_1 + (m_2 - \Delta m) y_2}{m_1 + m_2 - \Delta m(t)} \quad (19)$$

Where  $COG(t)$  is the Center of Gravity as a function of time,  $m_1$  is the static mass (combination of nose cone, body tube, and fins),  $m_2$  is the initial mass of the motor, and  $\Delta m(t)$  is the change of mass as a function of time due to fuel expenditure.

We consider the motor as a point mass centered at the geometric center of the motor casing. This simplifies the calculation of the center of gravity of the rocket as fuel is expended, as only the mass of the motor is changing, and not the location of its particular center of mass.

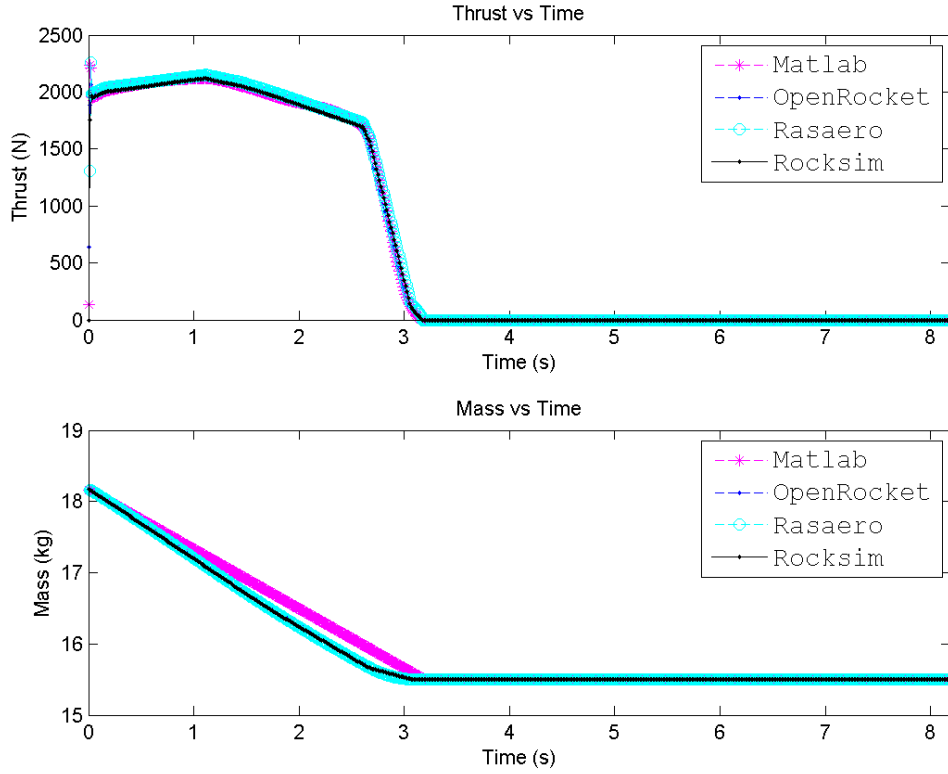


Figure 2: Dynamic Weight Calculation Test Output

### Moments of Inertia

The instantaneous moment of inertia is determined by relating the moment of inertias of the static structure and the dynamic structure through the parallel axis theorem evaluated at the total center of gravity (COG).

The sum of moment of inertias evaluated through the *parallel axis theorem* nets the total rocket moment of inertia.

$$I_n = I_{cm(n)} + M_P d^2 \quad (20)$$

$$I_T(t) = \sum I_n \quad (21)$$

Where:

- $I_T(t)$  is the total moment of inertia of the rocket as a function of time
- $I_n$  is the component vector (either static or dynamic moment of inertia)

[3]

### Longitudinal Moment of Inertia

To the *Moment of Inertia* related to the pitch/yaw of the rocket is the *Longitudinal Moment of Inertia*.

$$I = \frac{mL^2}{12} \quad (22)$$



[TODO source dynamics textbook]

In keeping with the assumption of the motor as a point mass in the volumetric center of the motor casing, the dynamic *Longitudinal Moment of Inertia* is calculated as follows.

$$I_s + m_s r_{0 \rightarrow 1}^2 \quad (23)$$

Where  $r_{0 \rightarrow 1}$  is the distance between the static center of gravity (the COG of the nose cone, body tube, and fins) and the instantaneous center of gravity of the rocket.  $I_s$  is provided by CATIA.

$$I_m = \frac{m_m L_m}{12} + m_m r_{0 \rightarrow 2}^2 \quad (24)$$

Where  $L_{motor}$  is the length of the motor casing, and  $r_{0 \rightarrow 2}$  is the distance between the motor center of gravity and the rocket center of gravity.

Then, the rocket *Longitudinal Moment of Inertia* is the sum, shown as follows

$$I_r = I_m + I_s \quad (25)$$

### Center of Pressure

The *Center of Pressure* (COP) is the location (point) where the aerodynamic forces can be said to be acting, simplifying the complex distribution of forces across the rocket and its features.

The *Center of Pressure* changes with the normal force distribution on the rocket, which is driven by *Angle of Attack* [6].

$$COP = COP(\alpha)$$

A wind tunnel is the best way to approximate this point, but an analytic method is available, discussed in detail in the next section.

Figure 3 shows how our determination of the *COP*, *COG*, and  $I_L$  compares with other simulation software.

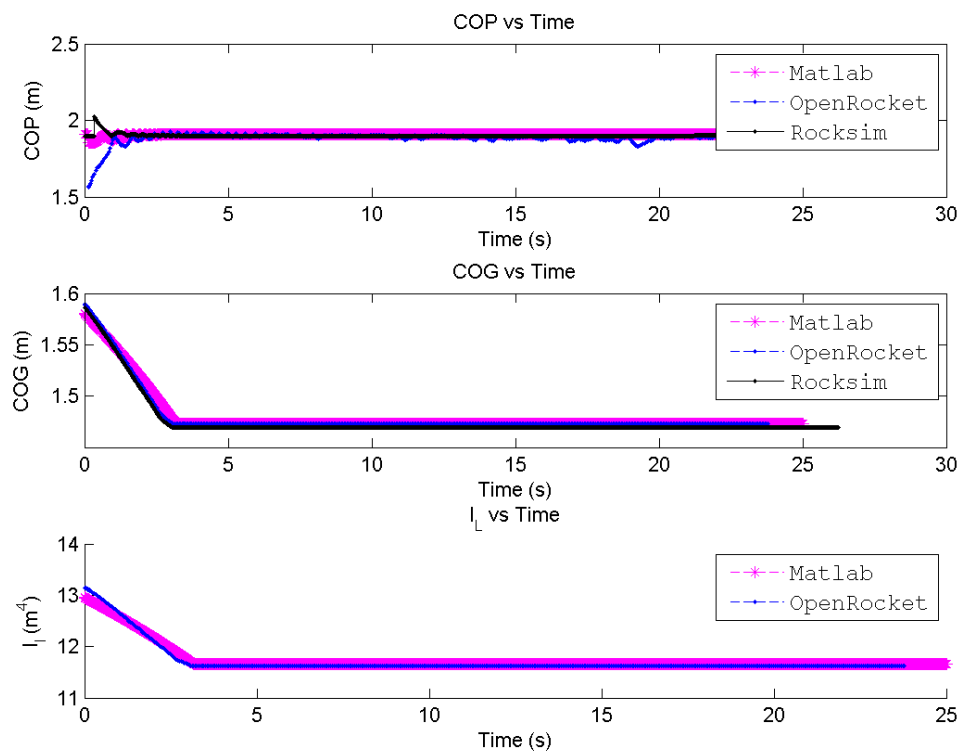


Figure 3: COP, COG, and  $I_L$  as a function of time

## The Barrowman Method

In 1966, James Barrowman published a report called “The Theoretical Prediction of the Center of Pressure” [7]. Despite some modifications and additions since, it remains a fundamental method, ever present in modern high-powered rocketry.

*Barrowman’s Method* is principally used to determine the *Center of Pressure*.

$$\bar{X} = \frac{(C_{N\alpha})_n \bar{x}_n + (C_{N\alpha})_{cb} \bar{x}_{cb} + (C_{N\alpha})_{fb} \bar{x}_{fb}}{C_{N\alpha}} \quad (26)$$

Where:

- $C_{N\alpha}$  is the *Stability Derivative*
- subscript  $_n$  refers to the nose cone
- subscript  $_{cb}$  refers to the cylindrical body
- subscript  $_{fb}$  refers to the fin set in the presence of the body
- $\bar{x}$  refers to the component centroid

[6]

## Stability Derivative

The *Stability Derivative*  $C_{N\alpha}$  is a dimensionless parameter, used to calculate the force normal to the longitudinal axis, and is dependent on the shape of the component. It is the slope of the *Normal Force Coefficient* plotted against the angle-of-attack. For low angles of attack, it is nearly constant.

TODO show figure

[2]

The total *Stability Derivative* is the sum of all  $i$  rocket component stability derivatives

$$C_{N\alpha} = \sum C_{N\alpha(i)} \quad (27)$$

[3]

## Nose Cone

$$C_{N\alpha(n)} = 2 \quad (28)$$

## Rocket Body

The *Barrowman Method* considers the body lift at small angles of attack to be negligible.

$$C_{N\alpha(bt)} = 0 \quad (29)$$

## Fins

The following solution for the fin set stability derivative applies only for identically shaped fins, in sets of 3, 4, or 6.

$$C_{N\alpha(f)} = C_{in} \frac{4n \left(\frac{s}{d}\right)^2}{1 + \sqrt{1 + \left(\frac{2l}{a+b}\right)^2}} \quad (30)$$

Where:

- $a$  is the fin tip chord length
- $b$  is the fin root chord length
- $s$  is the fin height
- $l$  is the distance between the root center and the tip center
- $C_{in}$  is a coefficient for the interference effects of the air flow near the fin-body interface

[6]

$$C_{in} = 1 + \frac{OD/2}{OD/2 + s} \quad (31)$$

[6]

## Nose Cone COP

### LV-Haack Nose Cone COP

$$\bar{X}_n = 0.437h_n \quad (32)$$

[8]

### Von Karman Nose Cone COP

$$\bar{X}_n = 0.500h_n \quad (33)$$

[8]

## Fin Set COP

The location of the *Center of Pressure* for the fin set is as follows.

$$\bar{X}_{fb} = X_f + \frac{m(b+2a)}{3(b+a)} + \frac{1}{6} \left[ b + a - \frac{ba}{b+a} \right] \quad (34)$$

[3]

Where:

- $X_f$  is the distance from the tip of the nose cone to the point where the leading edge of the fin meets the body tube [3]
- $a$  is the fin tip chord length
- $b$  is the fin root chord length
- $s$  is the fin height
- $m$  is the fin sweep length

Also, see this [Center of Pressure Calculator online](#).

## Cylindrical Body COP

The centroid of a cylindrical body will be half its length

$$\bar{x}_{ct} = \frac{1}{2}l_{cb} \quad (35)$$

Wind tunnel tests performed in 1918 and 1919 demonstrated that the normal force generated by a cylindrical body at an angle of attack of less than 10 degrees is negligible

[6].

## Rocket Body Lift Correction

*Barrowman's Method* neglects the lift generated by the rocket body. Galejs [9] suggests the following adjustment to provide a compensated *Coefficient of Normal Force due to Body Lift*

$$C_{N(L)} = K \frac{A_p}{A_{ref}} \alpha^2 \quad (36)$$

Where:

- $K = 1$
- $A_p$  is the rocket planform area excluding the fins
- $A_{ref}$  is the reference area of the rocket

[3]

Equation 36 is divided by  $\alpha$  to be added to  $C_{N\alpha}$  calculated and used in Equation 26.

TODO All COP components must be modified by the lift coefficient

$$C_{N\alpha^2} = K \frac{A_p}{A_{ref}} \alpha \quad (37)$$

This correction is applied at the centroid of the planform area.

[3]

## Transonic Considerations

*Barrowman's Equations* are based on assumptions that are only valid in subsonic flight. In the transonic and supersonic regions, what new effects are introduced that would affect the location of the *Center of Pressure*?

## Atmospheric Model

### Introduction

The *International Standard Atmosphere* model is assumed to describe the pressure, temperature, density and viscosity conditions of the surrounding air during launch.

The International Standard Atmosphere (ISA) model was established to give scientists and engineers an easy way to find the conditions at a certain altitude in the air given the conditions at ground level. The ISA model is extremely useful in aerospace applications and, while it does not take into account the small changes which usually occur in the atmosphere or any wind, it is considered to be reasonably accurate.

### Methodology

The model is created to output atmospheric density and the absolute and kinematic viscosities of the working fluid. To calculate these values, the model requires the launch temperature, launch pressure, the current altitude of the vehicle (a dynamic value over the course of the simulation) and the altitude of the launch site. For instance, the launch conditions for the Arcturus (IREC 2015) were 40 degrees Celsius, 99 *KPa* and 4300 feet. Of course, these inputs have the incorrect units. Temperature must be in Kelvin, pressure must be in Pascals and the altitude of the rocket must be in meters [10].

As well, when using the ISA model, initial ISA conditions of the model must also be defined. This includes the initial pressure (101325 Pascals), initial temperature (288.15 Kelvin or 15 degrees

Celsius) and the gas constant. The ISA model uses these initial conditions to calculate the conditions at altitude assuming that the temperature varies linearly up until the Tropopause [10].

In terms of altitude, the model calculates the ISA deviations from an imaginary sea-level point. These deviations are the first values that are calculated in the simulation and are necessary to obtain correct values for density and viscosity at any altitude. The general equations that were used to find the deviations from standard ISA conditions are as follows:

$$T_{dev} = T_{in} - T_0 + 6.5 \times \frac{alt_{in}}{1000} \quad (38)$$

Where  $T_{dev}$  is the deviation temperature in Kelvin,  $T_{in}$  is the input temperature at the launch site in Kelvin,  $T_0$  is the ISA standard temperature at sea level in Kelvin and  $alt_{in}$  is the launch site altitude in meters. For pressure:

$$P_{dev} = \left( \frac{P_{in}}{\left( 1 - 0.0065 \times \frac{alt_{in}}{T_0 + T_{dev}} \right)^{5.2561}} \right) - P_0 \quad (39)$$

Where  $P_{dev}$  is the deviation pressure in Pascals,  $P_{in}$  is the input pressure in Pascals and  $P_0$  is the ISA standard pressure in Pascals [10]. After having found the initial conditions, the conditions at the altitude at which the rocket is travelling need to be found. The first step to finding the essential properties of the working fluid is to first find the temperature and pressure at the rocket's current altitude. These calculations are executed assuming that the position in the simulation always starts at zero and that the launch site altitude is only taken into account in this script [10].

The temperature modeling equation is as follows

$$T_{act} = T_0 + T_{dev} - 6.5 \times \frac{alt_{in} + alt_{act}}{1000} \quad (40)$$

Where  $T_{act}$  is the actual temperature Kelvin at the rocket's current altitude and  $alt_{act}$  is the current altitude of the rocket in meters. The pressure modelling equation is as follows:

$$P_{act} = \left( (P_0 + P_{dev}) \cdot \left( 1 - 0.0065 \times \frac{alt_{in} + alt_{act}}{T_0 + T_{dev}} \right)^{5.2561} \right) - P_0 \quad (41)$$

Where  $P_{act}$  is the pressure in Pascals at the rocket's current altitude [10]. After having executed these calculations, the next thing that needs to be calculated are the essential properties of the working fluid. This means the pressure and dynamic and kinematic viscosities of the working fluid. To calculate the density, the following equation is used:

$$\rho = \frac{P_{act}}{RT_{act}} \quad (42)$$

Where  $\rho$  is the atmospheric density in  $Kg/m^3$ , and  $R$  is the specific gas constant for air at regular temperatures, which is equal to  $287.058 J/Kg - K$ .

After calculating the density, the absolute viscosity is calculated using Sutherland's law. The expression is as follows:

$$\mu = \mu_{S0} \left( \frac{T_{act}}{T_{S0}} \right)^{3/2} \frac{T_{S0} + S}{T_{act} + S} \quad (43)$$

Where  $\mu$  is the absolute viscosity in  $Kg/m - s$ ,  $\mu_{S0}$  is the reference viscosity, equal to  $1.716 \times 10^{-5} Kg/m - s$ ,  $T_{S0}$  is the reference temperature, equal to  $271.11 K$  and  $S$  is the Sutherland constant, equal to  $110.56 K$ . The constants are only applicable to air at regular temperatures

and pressures. They will lose accuracy at extremely low or high temperatures and at very high or low pressures [11]. After having calculated the absolute viscosity, the kinematic viscosity is found using the following relationship:

$$\eta = \frac{\mu}{\rho} \quad (44)$$

## Aerodynamic Geometry

### Overview

Related to aerodynamic geometry of the rocket, the specific parameters of interest are the following:

- Outer Diameter of Rocket ( $OD$ )
- Total Length of Rocket ( $L$ )
- Height of Nose Cone ( $h_n$ )
- Thickness of Fins
- Number of Fins
- Width of Fins
- Surface Area of Nose

### Surface Roughness

*Surface Roughness* is the deviation in the normal direction from a surface of its features. It contributes to *Skin Friction Drag*

### Fineness Ratio

The *Fineness Ratio* is the ratio of the length to the outer diameter

$$f_B = \frac{L}{OD} \quad (45)$$

### Fins

#### Aerodynamic Chord Length of Fins

Since there is no airfoil on the fin design, the *Aerodynamic Chord Length of the Fins* ( $L_{cf}$ ) is equal to the height of the fins.

### Areas

Reference areas are required to calculate the drag force.

#### Wetted Body Area

The *Wetted Body Area* is the combined area of all surfaces in contact with moving air.

#### Frontal Reference Area

The *Frontal Reference Area* is the projected area of the rocket perpendicular to the direction of air flow. For perfectly vertical flight and quiescent air conditions, this is the precise projection of the tip face of the rocket.

[TODO show figure]

## Frontal Reference Area at Angle of Attack

When the rocket pitches into the free stream at an angle of attack, a greater portion of the rocket comprises the frontal reference area. We do not account for this, as it would be extremely complex to evaluate. We consider the frontal reference area does not change with angle-of-attack.

## Planform Area

## Nose Profile

### Von Karman (Haack)

A *Von Karman* nose profile has been selected by the design team, other profiles will not be supported in the initial version of the model. The *Von Karman* nose profile is a *Haack Series* geometry, designed to minimize theoretical pressure drag [niskanen2013]. This profile excels in subsonic flow conditions, and performs well in transonic flow conditions [nassaNoseCone] - as such is it well suited for the current mission.

The equation for the *Haack Series* is

$$r(x) = \frac{R}{\sqrt{\pi}} \sqrt{\theta - \frac{1}{2} \sin(2\theta) + \kappa \sin^3 \theta} \quad (46)$$

Where

$$\theta = \cos^{-1} \left( 1 - \frac{2x}{L} \right) \quad (47)$$

## Nose Profile Design

## Nose Profile Design

## Aerodynamic Center

The *Aerodynamic Center* is the point where the *Pitching Moment* does not change with angle-of-attack

Sources:

- [http://ocw.mit.edu/courses/aeronautics-and-astronautics/16-100-aerodynamics-fall-2005/lecture-notes/16100lectre10\\_cg.pdf](http://ocw.mit.edu/courses/aeronautics-and-astronautics/16-100-aerodynamics-fall-2005/lecture-notes/16100lectre10_cg.pdf)
- [http://www.digplanet.com/wiki/Aerodynamic\\_center](http://www.digplanet.com/wiki/Aerodynamic_center)

## Drag Model

Rockets in flight experience multiple sources of drag. The total drag effect is the sum of all specific drag effects.

Figure 4 depicts the types of drag forces to be expected in subsonic flight at *zero-angle of attack*. [4]

The two main contributing factors to *Drag Force* are *Skin Friction* and pressure distribution effects. Pressure distribution effect are broken down into body pressure and parasitic drag effects, among others [4]. These and other drag forces are detailed in this section.

The drag model must take the parametric design parameters and applicable dynamics parameters (see *Data Model*) to output the Drag Force and combined drag coefficient.



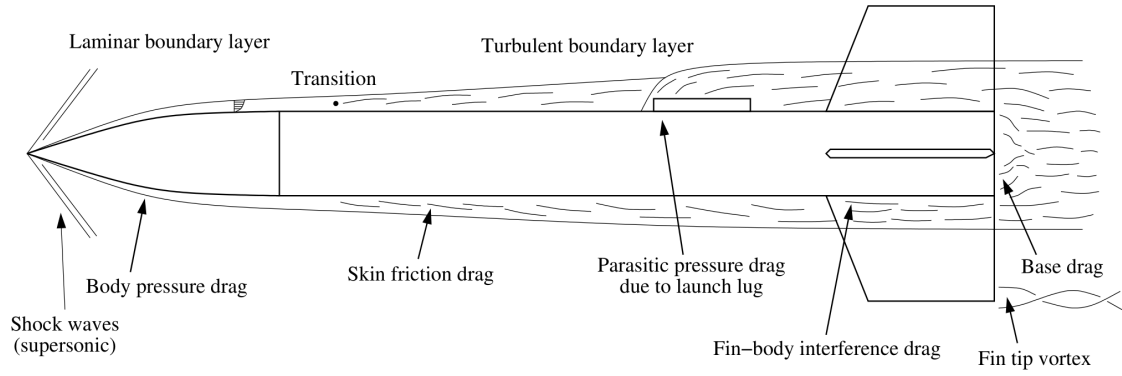


Figure 4: Rocket Drag Sources - Subsonic Flight

## Mach Number

*Mach Number* ( $M$ ) is the ratio of the airspeed to the speed of sound for air at a given temperature. The speed of sound ( $c$ ) is calculated as follows:

$$c = \sqrt{\gamma RT} \quad (48)$$

The *Ideal Gas Law* states that

$$P = \rho RT \quad (49)$$

We assume that the *Ideal Gas Law* applies, and use it to solve for  $RT$  using pressure and density.

$$RT = \frac{P}{\rho}$$

Thus we can calculate the speed of sound as follows:

$$c = \sqrt{\gamma \frac{P}{\rho}} \quad (50)$$

Where  $p$  is the local pressure,  $\rho$  is the local density, and  $\gamma$  is the *adiabatic index*, known as the *isentropic expansion factor* - it is the ratio of the specific heats of a gas at constant pressure and constant volume.

[12]

The *Mach Number* is then the ratio of the air velocity to the sound speed of the local air:

$$M = \frac{\vec{v}}{c} \quad (51)$$

## Mach Regions

*Velocity regions* are defined, in which aerodynamic effects are known to vary considerably. The following velocity regions are established for further discussion.

Mach Region ( $M$ )	Classification
$0.3 < 0.8$	Subsonic
$0.8 < M < 1$	Transonic
$1 < M < \sim 5$	Supersonic
$M > \sim 5$	Hypersonic

Table 6: Mach Regions

## Incompressible Flow

For Mach  $< 0.3$ ,

In the incompressible flow regime the forces can be divided into pressure force and viscous force

*Pressure Force* is due to fluid stagnation on areas of the rocket, as well as due to the low pressure region created beyond the rocket as it passes quickly through the air.

*Viscous Force* is due to boundary layer effects and interactions of moving air with surfaces. These forces are highly dependent on Reynolds number. [3]

## Compressible Flow Correction

Special considerations apply when compressibility effects are in play. These effects occur above Mach 0.3 [3], which will be easily exceeded by the transonic upper limit of Mach 0.9 mandated by the competition.

At low speeds (incompressible flow), the aerodynamic coefficients are functions of the angle of attack ( $\alpha$ ) and Reynolds number ( $Re$ ).

$$C_i(M < 0.3) = C_i(\alpha, Re) \quad (52)$$

At higher speeds (compressible,  $Ma \geq 0.4$ ) they are also a function of Mach number.

$$C_i(M \geq 0.3) = C_i(\alpha, Re, M) \quad (53)$$

Particular correction factors are recommended for ranges of Mach number

Mach Number	Correction Factor
$M < 0.3$	N/A
$0.3 < M < 0.8$	$C'_i = \frac{C_i}{\sqrt{1 - M^2}}$
$0.8 < M < 1.1$	$C'_i = \frac{C_i}{\sqrt{1 - (0.8)^2}}$
$M > 1.1$	$C'_i = \frac{C_i}{\sqrt{M^2 - 1}}$

Table 8: Prandtl-Glauert Compressible Flow Correction Factors

Where  $C_i$  is the incompressible drag coefficient and  $C'_i$  is the compressibility corrected drag coefficient [3].

## Turbulent Effects

A turbulent boundary layer induces a notably larger skin friction drag than a laminar boundary layer

[4]

## Stagnation Pressure

*Stagnation Pressure* is the pressure on the normal surfaces to airflow.

For a cylindrical rocket, it can be approximated as follows [4]

$$\frac{q_{stag}}{q} = \begin{cases} 1 + \frac{M^2}{4} + \frac{M^4}{40} & M < 1 \\ 1.84 - \frac{0.76}{M^2} + \frac{0.166}{M^4} + \frac{0.035}{M^6} & M > 1 \end{cases} \quad (54)$$

Where  $q_{stag}$  is and  $q$  is

Then, the *Pressure Drag Coefficient* can be expressed as a function of *Mach Number*

$$C_{pr} = 0.85 \frac{q_{stag}}{q} \quad (55)$$

## Reynolds Number

The *Reynolds Number* is a dimensionless number which describes the ratio of the kinematic effects of a fluid to viscous effects.

$$Re = \frac{\rho \vec{v} d}{\mu} \quad (56)$$

[12]

### Critical Reynolds Number

The *Critical Reynolds Number* ( $Re_{crit}$ ) is the value of *Reynolds Number* where the flow changes from laminar to turbulent. This is greatly dependent on the surface roughness [munson2013].

[4] gives the *Critical Reynolds Number* as

$$R_{crit} = \frac{\vec{v} x}{\nu} \quad (57)$$

Where:

- $\vec{v}$  is the free stream air velocity
- $x$  is the distance along the body from the nose cone tip where turbulent flow begins
- $\nu$  is the kinematic viscosity of air

For  $Re_{crit} = 5 \times 10^5$

- $\nu = 1.5 \times 10^{-5} m^2/s$
- $v_0 = 100 m/s$
- $x = 7 cm$  from the nose tip, where turbulent flow begins

[4]

[Trinh, Khanh Tuoc See Fluids Text book](#)

Surface roughness has a considerable influence on *Critical Reynolds Number*. It can be determined as follows.

$$R_{crit} = 51 \left( \frac{R_s}{L} \right)^{-1.039} \quad (58)$$

[4]

## Actual Reynolds Number

The *Actual Reynolds Number* can be expressed in the following form:

$$Re = \frac{\vec{v}L}{\nu} \quad (59)$$

Where:

- $\vec{v}$  is the free stream velocity
- $L$  is the length of the rocket
- $\nu$  is the kinematic viscosity of the air in free stream

## Drag Force and Coefficients

The total drag force is a function of air velocity (relative to the rocket body) drag coefficient, reference area, and air density.

$$D_f = D_f(\vec{v}, C_d, A_{ref}, \rho) \quad (60)$$

The drag coefficient  $C_d$  is the sum of all component drag coefficients

$$C_d = \sum C_i = C_{pa} + C_{fo} + C_{pr} + C_{in} + C_{ba} + C_{sk} + C_{fp} + C_{wa} + C_{bt} + C_{aoa} \quad (61)$$

From Fluid Mechanics [source?]

$$D_f = \frac{1}{2} C_d A_{ref} \rho \vec{v}^2 \quad (62)$$

## Viscous Drag Effects

### Skin Friction Drag

Skin Friction Drag is due to viscous effects during flight, and is significantly influenced by surface roughness.

$$D_{sk} = \frac{1}{2} \rho \vec{v}^2 A_{wet} C_{sk} \quad (63)$$

[12]

Where

$$C_{sk}, (A_{wet}, M, \frac{\epsilon}{l}) \quad (64)$$

$\frac{\epsilon}{l}$  is the relative roughness of the surface

With the critical and actual Reynolds Numbers determined, the *Uncorrected Skin Friction Drag Coefficient* can now be conditionally determined

$$C_{sk_{uncorrected}} = \begin{cases} 0.0148 & Re < 10^4 \\ \frac{1}{(1.5 \ln Re - 5.6)^2} & 10^4 < Re < Re_{crit} \\ 0.032 \left( \frac{Re}{L} \right)^{0.2} & Re > Re_{crit} \end{cases} \quad (65)$$

[4]

Two other sources describe the cases for Skin Friction Drag Coefficient differently.

$$C_{sk_{uncorrected}} = \begin{cases} \frac{1.328}{\sqrt{Re}} & Re \leq Re_{crit} \\ \frac{0.074}{Re^{1/5}} & 10^4 < Re < Re_{crit} \end{cases} \quad (66)$$

[3] and [2] agree on the above.

The *Skin Drag Coefficient Corrected for Compressibility* is:

Conversely, Niskanen evaluates the corrected skin drag coefficient as follows

$$C_{sk_{corrected}} = C_{sk_{uncorrected}} \times \begin{cases} (1 - 0.1M^2) & \text{Subsonic} \\ [(1 + 0.15M^2)^{0.58}]^{-1} & \text{Supersonic} \\ (1 + 0.18M^2)^{-1} & \text{Roughness Limited} \end{cases} \quad (67)$$

Finally, the *Normalized and Corrected Skin Friction Drag Coefficient* is:

$$C_{sk} = \frac{C_{sk,c} \left[ \left(1 + \frac{1}{2f_B}\right) \cdot A_{wb} + \left(1 + \frac{2t_f}{L_{cf}}\right) A_{wf} \right]}{A_{ref}} \quad (68)$$

Where  $f_b$  is the *Fineness Ratio*, the ratio of the length of the rocket divided by the outer diameter.  $L_{cf}$  is the aerodynamic chord length of the fins, and  $t_f$  is the thickness of the fins

[4]

$$Re_{crit} = 51 \left( \frac{R_a}{L} \right)^{-1.039} \quad (69)$$

### Pressure (Form/Profile) Drag

This is the drag caused by the pressure exerted on the surface of an object as it moves through a free stream [12].

$$C_{pr}, D_{pr}(A_{ref}, M) \quad (70)$$

### Body Drag

*Body Drag* is the drag on the rocket forebody (pressure drag?)

$$C_{fb} = \left[ 1 + \frac{60}{(l_{TR}/d_b)^3} + 0.0025 \frac{l_b}{d_b} \right] \left[ 2.7 \frac{l_n}{d_b} + 4 \frac{l_b}{d_b} 2 \left( 1 - \frac{d_d}{d_b} \right) \frac{l_c}{d_b} \right] \cdot C_{f(fb)} \quad (71)$$

Where  $l_{TR}$  is the total length of the rocket body,  $l_c$  is the length of the boat tail,  $d_b$  is the maximum body diameter and  $d_d$  is the diameter of the rocket base.  $C_{f(fb)}$  is the coefficient of viscous friction on the rocket forebody (defined later in (45))

### Fin Pressure Drag

The *Fin Pressure Drag* depends on the fin profile. The current rocket will use a square (rectangular) profile, and can be determined as follows.

$$C_{fp}, D_{fp}(A_{ref}, M) \quad (72)$$

### Leading Edge pressure drag

$$C_{D,LE} = C_{D,stag} = 0.85 \frac{q_{stag}}{q} \quad (73)$$

The *Body Base Drag Coefficient* is

$$C_{base} = \begin{cases} 0.12 + 0.13M^2 & M < 1 \\ \frac{0.25}{M} & M > 1 \end{cases} \quad (74)$$

For perpendicular orientation of the fin edges to air flow, the stagnation pressure defined in Equation 54 is used.

$$\frac{q_{stag}}{q} = \begin{cases} 1 + \frac{M^2}{4} + \frac{M^4}{40} & M < 1 \\ 1.84 - \frac{0.76}{M^2} + \frac{0.166}{M^4} + \frac{0.035}{M^6} & M > 1 \end{cases}$$

[4]

### Von Karman Nose Pressure Drag

Most nose cone shapes can be approximated to produce zero pressure drag at subsonic velocities, however complications arise for transonic and supersonic velocities. The cause of this drag is slight flow separation, and as such cannot be corrected due to compressibility effects. A semi-empirical method can be employed, and is explored by [4].

### Base Drag

Base drag is caused by a low pressure region generated behind the base of the rocket as it moves quickly through the atmosphere [4]. Specifically, it is due to boundary separation between the flow past the rocket and the surrounding air [3]. The flowing air attempts to make a sharp turn around the sudden geometry change at the base end of the rocket, however, viscous effects resist this change in direction. As a result, pressure cannot be equalized in the space directly behind the rocket and a low-pressure (vacuum) region forms [13]. This low-pressure region has an effect analogous to *pulling* the rocket against its direction of flight.

$$C_{ba}, D_{ba}((A_{ref}, M)) \quad (75)$$

$$C_{ba} = \begin{cases} 0.12 + 0.13M^2 & M < 1 \\ \frac{0.25}{M} & M > 1 \end{cases} \quad (76)$$

[4]

In reality, this low pressure region is disturbed by the thrust envelope from the motor. Thus, we would expect base drag to be different during the motor burn time than during the free flight after all fuel was exhausted. Considering the thrust envelope is at this moment beyond the scope of the project. Instead, an accepted approximation is to subtract the area of the motor from the area of the base when calculating drag force [4].

$$D_{ba} = \frac{1}{2} C_{ba} \rho (A_{tube,base} - A_{motor,base}) \vec{v}^2 \quad (77)$$

We can normalize the base drag coefficient to take this into account.

$$C_{ba,normalized} = C_{ba} * A_{tube,base} / A_{motor,base} \quad (78)$$

## Shoulder Pressure Drag

The drag coefficient of the shoulder interfacing the body tube is assumed to be equal to that of the body tube itself, and also assumes a smooth interface. This is likely to be sufficient for subsonic velocities [4], and for the scope of this project it is neglected entirely.

## Parasitic Drag

Parasitic drag is the drag due to body features not explicitly designed and/or imperfections not easily approximated. Examples include launch guides, ventilation holes, surface roughness, and any damage during flight.

PARASITIC DRAG IS CURRENTLY NEGLECTED IN THE MODEL

$$C_{pa}, D_{pa}(A_{ref}, M) \quad (79)$$

Where  $C_{stag}$  is the *Stagnation Drag Coefficient* [see equation from fin pressure drag section]

We consider the most significant source of *Parasitic Drag* to be the launch lug. If there is no significant airflow through the launch lug, we can approximate it as a cylinder next to the rocket body. *Niskanen* states that a launch lug with a length at least two times its width has a drag coefficient of 0.74, with its reference area being the frontal area. Stagnation pressure proportionally influences the drag coefficient [4].

The following equation relates the launch lug diameter  $\phi_{lug}$  to the launch lug tube length  $l_{lug}$ .

$$C_{pa} = \left( 1.3 - 0.3 \frac{l_{lug}}{\phi_{lug}}, 1 \right)_{max} \cdot C_{stagnation} \quad (80)$$

Where  $L$  is the rocket length,  $h_n$  is the height of the nose cone,  $OD$  is the outer diameter of the rocket, and  $C_{stagnation}$  is the stagnation coefficient [4].

The reference area of the launch lug is given as follows

$$\pi \cdot (r_{ext,lug}^2 - r_{int,lug}^2) \cdot \left[ 1 - \left( \frac{l_{lug}}{\phi_{lug}} \right) \right]_{+ve} \quad (81)$$

The *Parasitic Drag Coefficient* can be normalized to the reference area of the launch lug.

$$C_{pa_{norm}} = C_{pa} \cdot \left( \pi \cdot (r_{ext}^2 - r_{int}^2) \cdot \left[ 1 - \left( \frac{L - h}{OD} \right) \right]_{+ve} \right) \quad (82)$$

[4]

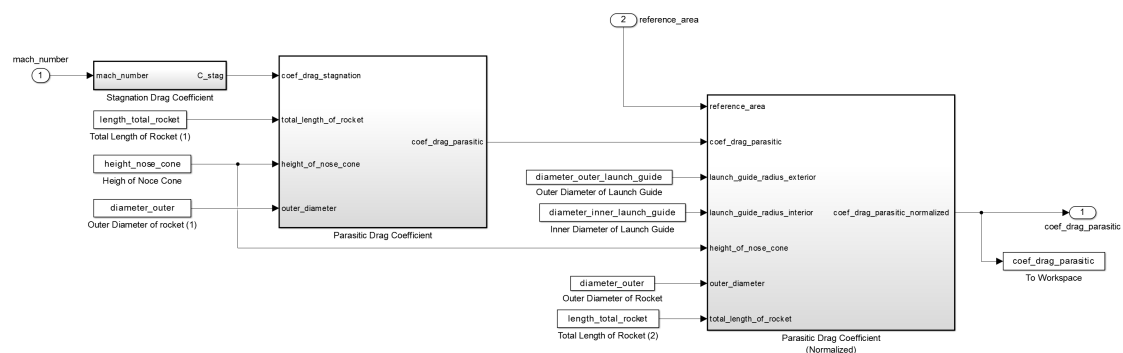


Figure 5: Matlab Implementation of Parasitic Drag Coefficient

### Interference Drag

*Interference Drag* is caused due to effects of air flow at the interfaces of the fins and the body.

$$C_{in}, D_{in}(A_{ref}, M) \quad (83)$$

$$C_{in} = 2C_{sk, fins} \left( 1 + 2\frac{T_f}{l_m} \right) \frac{4n(A_{f_p} - A_{f_e})}{\pi d_f^2} \quad (84)$$

Where:

- $C_{sk, fins}$  is the coefficient of skin friction (due to viscous effects) on the fins
- $n$  is the number of fins
- $A_{f_p}$  is the fin planform area

$$A_{f_p} = A_{f_e} + \frac{1}{2}d_f l_r \quad (85)$$

- $A_{f_e}$  is the exposed planform area of the fin

$$A_{f_e} = \frac{1}{2}(l_r + l_t)l_s \quad (86)$$

[3]

Interference Drag effects are small in comparison to other drag effects [4], and are thus ignored at this stage of the project.

### Wave Drag

*Wave drag* is drag associated with shock waves (independent of viscous effects).

At transonic speed, shock waves form at the nose tip and at the leading edge of the fins ... Momentum is transferred from the rocket to the surrounding air via these shockwaves

### Boat-Tail Drag

A *boat-tail* is a reduction in diameter of the body tube towards the base of the rocket. Our rocket does not have a boat-tail, thus *Boat-Tail Drag* considerations are ignored.



## Additional Drag at Angle of Attack

When the rocket flies at a non-zero angle of attack, additional drag considerations must be made. The reference area the rocket becomes larger as the rocket is pitched into the free stream, exposing more of the rocket body to pressure and stagnation effects.

In Figure 6, velocity  $\vec{v}$  is the apparent velocity of the center of pressure relative to the surrounding air.

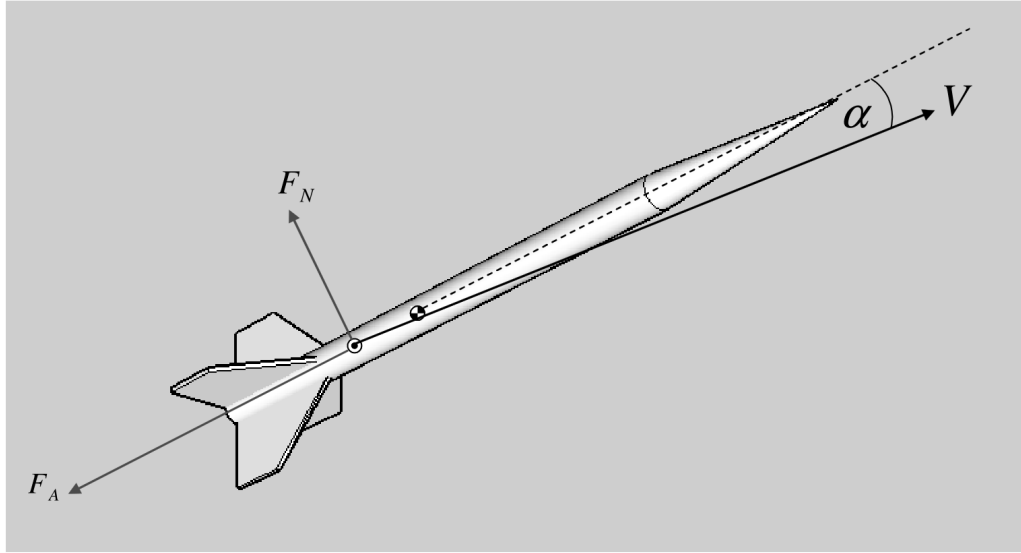


Figure 6: Rocket Drag Forces - Axial vs. Normal Caption

[3]

In the following analysis, additional rocket drag coefficients are determined to be added to the *zero angle of attack* drag coefficient. This analysis is derived with the aid of additional coefficients determined experimentally in wind tunnel tests on rocket models [3] [2].

$$C_{aoa} = C_{Db(\alpha)} + C_{Df(\alpha)} \quad (87)$$

### Rocket Body Drag at Angle of Attack

$$C_{Db(\alpha)} = 2\delta\alpha^2 + \frac{3.6\eta(1.36L - 0.55h_n)}{\pi \cdot OD}\alpha^3 \quad (88)$$

Where:

- $\alpha$  is the angle of attack
- $L$  is the total rocket length
- $OD$  is the outer diameter of the rocket
- $h_n$  is the height of the nose cone
- $\delta$  and  $\nu$  are experimentally determined coefficients
- $OD$  is the outer diameter of the rocket

[3]

### Rocket Fin Drag at Angle of Attack

$$C_{Df(\alpha)} = \alpha^2 \left[ 1.2 \frac{A_{fp}^4}{\pi OD_f^2} + 3.12(k_{fb} + k_{bf} - 1) \left( \frac{A_{fe}^4}{\pi OD_f^2} \right) \right] \quad (89)$$

Where:

- $k_{fb}$  is the fin-body coefficient

$$k_{fb} = 0.8065R_s^2 + 1.1553R_s \quad (90)$$

- $k_{bf}$  is the body-fin coefficient

$$k_{bf} = 0.1935R_s^2 + 0.8174R_s + 1 \quad (91)$$

- $R_s$  is the fin section ratio

$$R_s = \frac{l_{TS}}{d_f} \quad (92)$$

- $l_{TS}$  is the total span of the fins
- $OD_f$  is the diameter of the body tube at the base of the fin mount

[3]

### Alternatively

[2] shares a function determined for *Total drag coefficient due to angle-of-attack*

$$C_d(\alpha) = 16.83\alpha^2 + 8.9\alpha^3 \quad (93)$$

## Matlab Implementation

Figure 7 shows the *Simulink* implementation of the calculation of the drag model

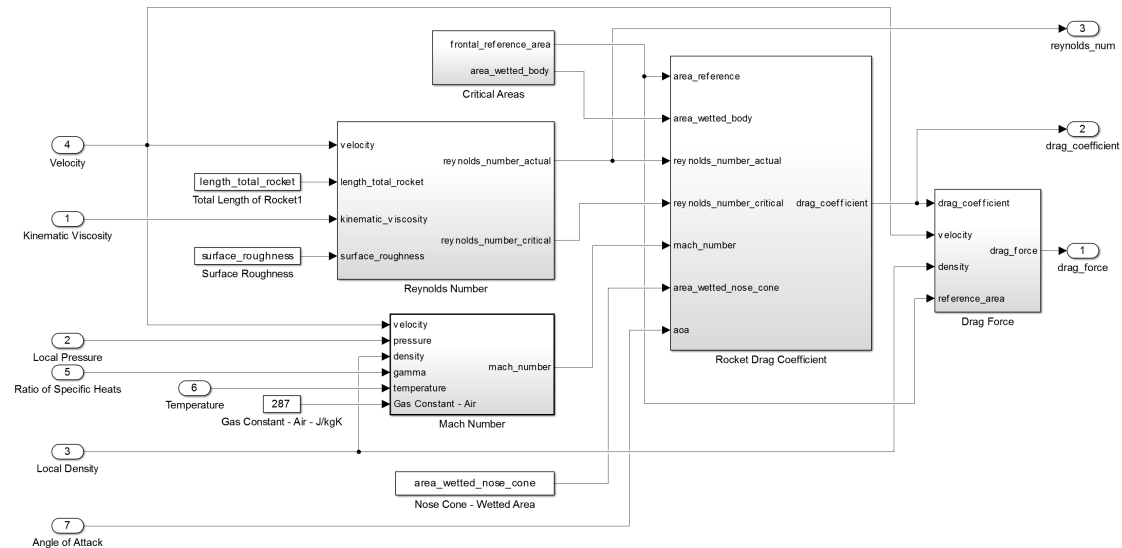


Figure 7: Rocket Drag Model

Figure 8 shows the *Simulink* implementation of the calculation of drag coefficient

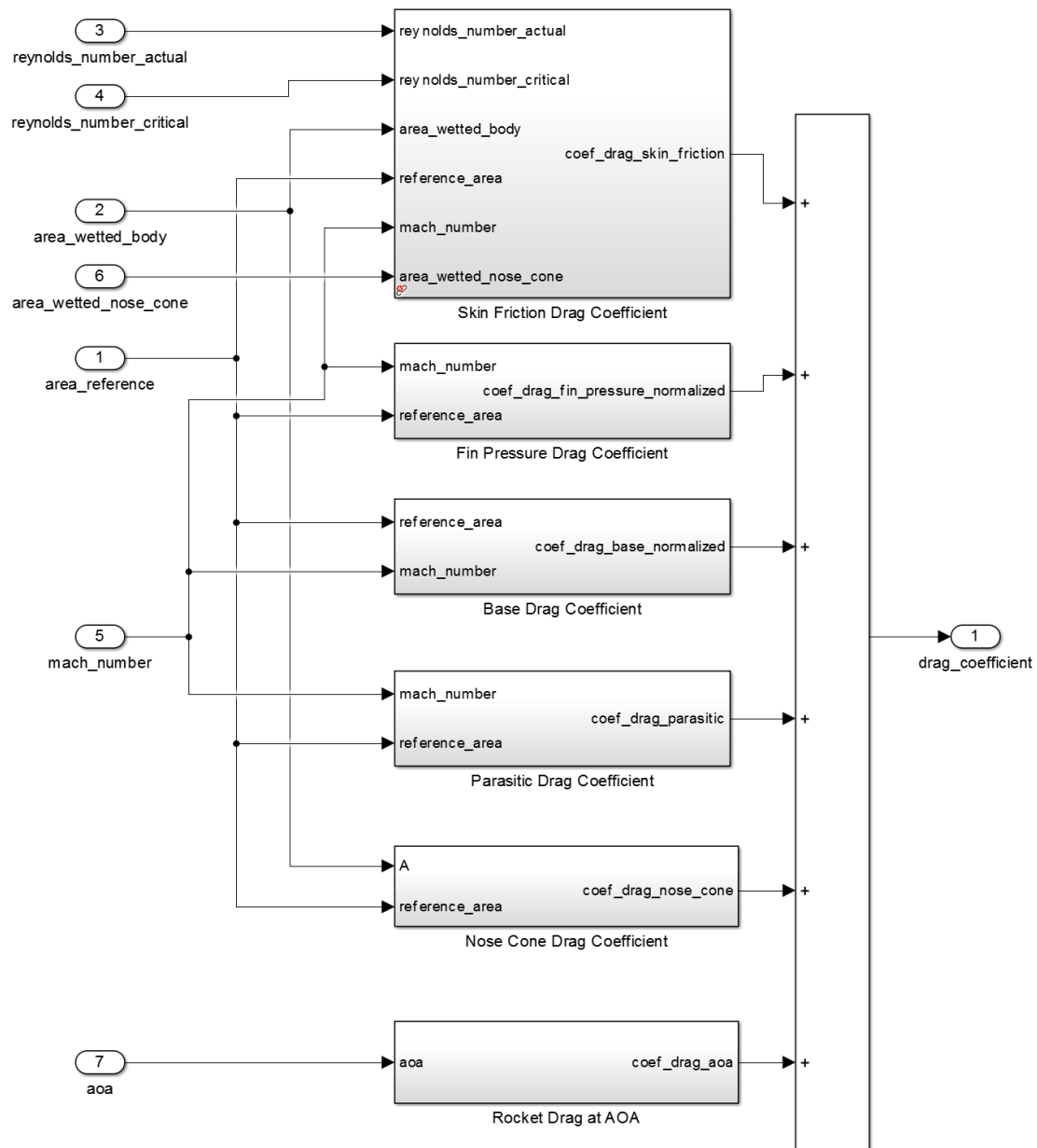


Figure 8: Rocket Drag Coefficient Model

## Matlab Validation

The following plots show the Drag Model compared against OpenRocket, RASAero, and Rocksim. The differences between the commercial simulations are likely due to differing drag analysis methods which are not available due to their closed source nature. However, it can be seen that Matlab and OpenRocket are very close, which validates the Matlab model since it was closely following the methods performed in OpenRocket

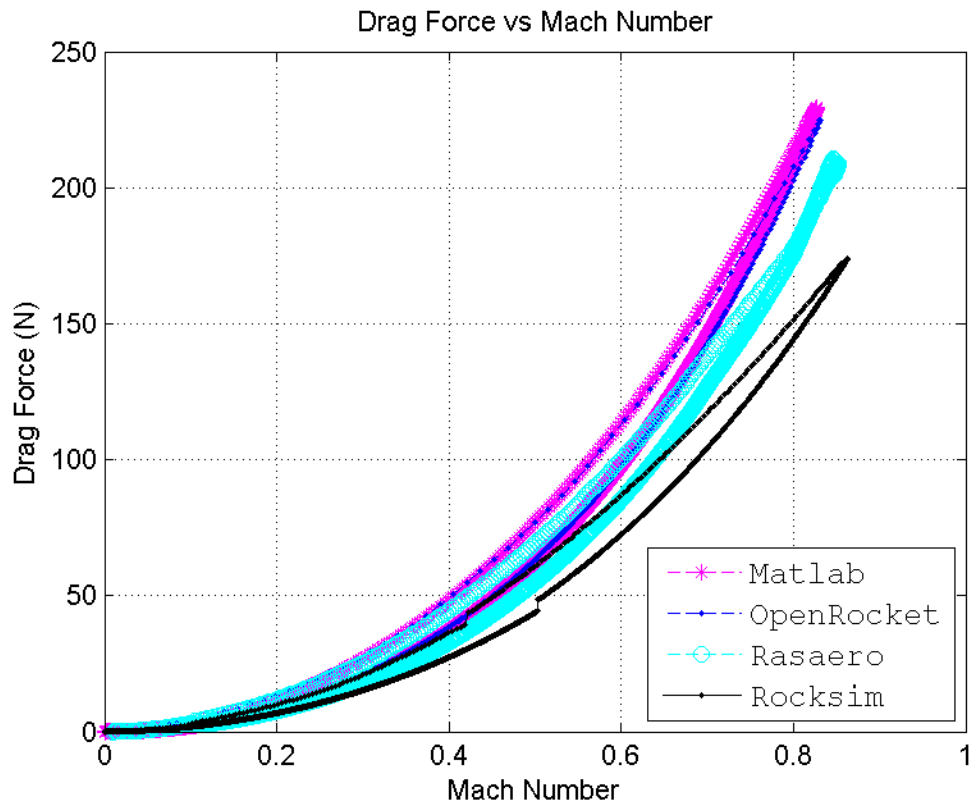


Figure 9: Drag Force as a Function of Mach Number

## Point-Mass Flight Model

The analysis of the point-mass flight model can be simplified to a sum of forces.

Simplifying the rocket flight as ideally one-dimensional, with the positive z-direction being upwards from the launch pad, the impulse is equal to the thrust of the rocket minus the weight of the rocket and the drag forces of the rocket interacting with the surrounding air.

$$m(t)\ddot{z}(t) = T(t) - D(\dot{z}) - W(t) \quad (94)$$

Mass is a function of time, which is explained in the *Dynamic Parameters* section. Drag is a function of velocity, which is explained in *Drag Model* section. Acceleration can be expressed as the first derivative of velocity and also the second derivative of position, each with respect to time.

$$\vec{a} = \dot{v} = \ddot{z} \quad (95)$$

Each force component can be rearranged and expressed as follows:

$$\vec{a}_T = \frac{T(t)}{m(t)}, \vec{a}_W = \frac{W(t)}{m(t)}, \vec{a}_D = \frac{D(v)}{m(t)} \quad (96)$$

The net upward acceleration is:  $\vec{a}_T - \vec{a}_W - \vec{a}_D$

The sum of forces can be rearranged and acceleration can be solved for:

$$\vec{a} = \ddot{z} = \frac{1}{m(t)}(T(t) - D(\dot{z}) - W(t)) \quad (97)$$

Acceleration can be integrated to find position and velocity.

$$\vec{v} = \int \vec{a} dz \quad (98)$$

$$z = \iint \vec{a} dz \quad (99)$$

Integration of equation (97) in the model is represented by the  $\frac{1}{s}$  block. The model is pictured in Figure 10.

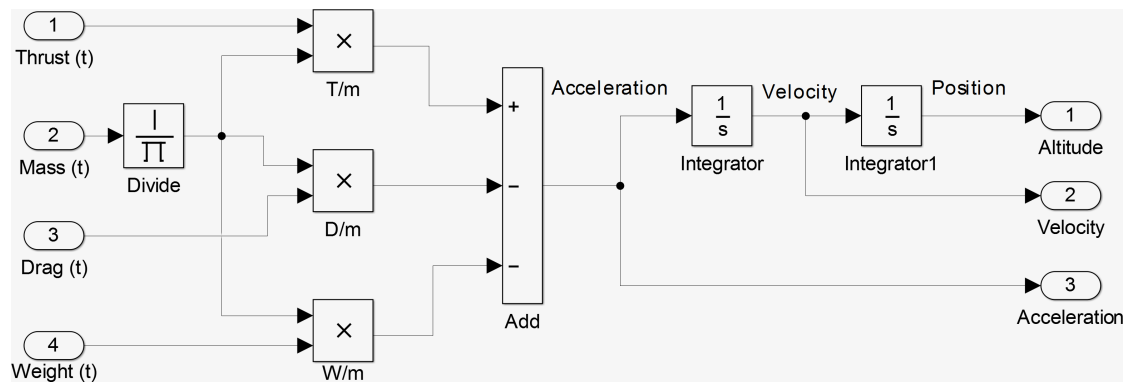


Figure 10: Vertical Flight Model - Simplified

## Weathercocking

*Weathercocking* is a phenomenon when the rocket tends to alter its trajectory and fly into the wind. If the rocket is stable, and has a sufficiently high damping ratio, the rocket eventually reaches a near-zero angle-of-attack parallel to the velocity vector of the wind, only in the opposite direction.

If we apply a wind velocity in the drag calculation, we can then modify Equation 94 to account for weathercocking and provide the actual altitude reached, as well as the amount of drift experienced.

## Altitude accounting for flight angle

$$m(t)\ddot{z}(t) = T(t) \cos \theta - D(\dot{z}) \cos \theta - W(t) \quad (100)$$

Where:

- $z$  is the upward direction (normal from the ground)
- $\theta$  is the angle between the current rocket trajectory and the z-axis

## Drift accounting for flight angle

$$m(t)\ddot{z}(t) = T(t) \sin \theta - D(\dot{z}) \sin \theta \quad (101)$$

# Rigid-Body Rotation (Pitch, Yaw) Stability Analysis

## Overview

Due to disturbances such as wind, and imperfections and imbalances in the construction, the rocket will tend to fly at an *Angle of Attack* into the free stream, wherein the velocity vector (taken from the *Center of Gravity*) is not parallel with the longitudinal axis. This will cause non-linear changes to the magnitude of the aerodynamic forces, which, as a further simplification, can be said to be acting on the *Center of Pressure*. In order for the aerodynamic forces to straighten the rocket in its forward motion, and to stabilize the oscillatory rotation about the COG, the COP must be located behind the COG.

The moment arm about the COG is the distance of the COP from the tip of the nose cone, minus the distance of the COG from the tip of the nose cone. Then, the sum of forces at the COP is the *Restoring Force* ( $F_R$ ) minus the *Damping Force* ( $F_D$ ), and the sum of the Moments about the COG is expressed as follows.

The *Moment* of a rigid body about its COG can be expressed as the product of the *Moment of Inertia* of the rigid body and the *Angular acceleration* of the body.

$$M = I\lambda \quad (102)$$

- $\lambda$  is the *angular acceleration* of the rigid body, which is the second time derivative of the angular displacement

$$\lambda = \ddot{\alpha}$$

$$\omega = \dot{\alpha}$$

- $\omega$  is the *angular velocity*, which is the first time derivative of the angular displacement
- $\alpha$  is the *angle of attack*

## Longitudinal Static Stability Margin

The *Longitudinal Static Stability Margin* ( $S_{lm}$ ) is the distance between the *Center of Gravity* and the *Center of Pressure* divided by the outer diameter of the body tube when the rocket is positioned at an angle-of-attack ( $\alpha$ ) of zero [2].

$$S_{lm} = \frac{COP - COG}{OD}$$

When traveling under a non-zero angle of attack, the Stability Margin is adjusted using the body lift correction factor Equation ??.

The result is dimensionless, however the ratio determined is measured in the number of *calibers*.

2a - The static stability margin falls above 2 (but less than 3) calibers at launch

## Requirement

- 2a - The static stability margin falls above 2 (but less than 3) calibers at launch
- 2b - The dynamic stability is greater than 0 even in winds up to 8.33 m/s
- 2f - The vehicle does not experience resonant pitching/yawing motion in flight

## Assumptions

- small angle of attack (less than  $10^\circ$ )
- incompressible flow
- neglect viscous forces
- neglect compressibility effects [3]
- neglect lift force on the body tube [3]
- neglect the effect of roll due to having 3 fins vs 4

## Definition of Terms

### Rocket Normal Force

The *Rocket Normal Force* is the resultant force applied at the *Center of Pressure* perpendicular to the longitudinal axis of the rocket, when the rocket flies at an angle-of-attack.

$$F_N = \frac{1}{2} \rho \vec{v}^2 A_c C_N \quad (103)$$

[3]

Where  $A_c$  is the cross-sectional area of the body tube, and  $C_N$  is the *Normal Force Coefficient*, and is a function of angle-of-attack ( $\alpha$ ). The small angle approximation is applied, wherein small angles can be approximated as a linear function of the angle.

$$C_N = C_{N\alpha} \cdot \alpha \quad (104)$$

[3]

## Corrective Moment Coefficient

The *Corrective Moment Coefficient* describes the reaction of the rocket against a disturbance about its longitudinal axis.

$$C_{MC} = \frac{1}{2} \rho \vec{v}^2 A_{ref} C_{N\alpha} (COP - COG) \quad (105)$$

Where:

- $\rho$  is the local density of air
- $\vec{v}$  is the velocity of the rocket
- $A_{ref}$  is the reference area of the rocket flying into the free stream
- $C_{N\alpha}$  is the *Stability Derivative Normal Force Coefficient*
- $(COP - COG)$  is the distance between the *Center of Pressure* and *Center of Gravity*

Note: a rocket with a high *Corrective Moment Coefficient* is going to weathercock faster at lower velocities.

## Corrective Moment Coefficient

## Dimensional Analysis

$$\frac{kg}{m^3} \left[ \frac{m}{s} \right]^2 m^2 m = \frac{kg \cdot m}{s^2} \cdot m \quad (106)$$

## Damping Moment Coefficient

As the rocket responds to a disturbance, the *Corrective Moment* reactions forces act in an oscillating manner - weathercocking into the wind, then turning back towards the vertical direction. In order to reach dynamic stability, this oscillation must decay and settle to a reasonable response. The *Damping Moment Coefficient* represents how fast the response settles towards zero.

There are two *Damping Moment Coefficients* to consider, the *Aerodynamic Damping Moment Coefficient* and the *Propulsive Damping Moment Coefficient*.

Then the *Damping Moment Coefficient* is the sum of the two moment components coefficients.

$$C_{DM} = C_{ADM} + C_{PDM} \quad (107)$$

## Aerodynamic Damping Moment Coefficient

Each rocket component contributes to the *Aerodynamic Damping Moment Coefficient*

$$C_{ADM} = \frac{1}{2} \rho \vec{v} A_{ref} \sum \left( C_{N\alpha, x} \cdot [COP_x - COG]^2 \right) \quad (108)$$

NOTE: Why isn't  $\vec{v}$  SQUARED? It might have something to with the fact that the ADM is a function of angular displacement, and DM is a function of angular velocity??

Where:

- $\rho$  is the local density of air
- $\vec{v}$  is the velocity of the rocket
- $A_{ref}$  is the reference area of the rocket flying into the free stream
- $C_{NF, x} C_{N\alpha}$  is the *Normal Force Coefficient Stability Derivative*
- $COP_x$  is the distance of *Center of Pressure* of the rocket component to the nose cone tip
- $COG$  is the distance between the rocket *Center of Gravity* to the nose cone tip



## Dimensional Analysis

$$\frac{kg}{m^3} \frac{m}{s} m^2 m^2 = \frac{kg \cdot m}{s} \cdot m \quad (109)$$

### Propulsive Damping Moment Coefficient

Also known as *Jet Damping*, as propulsion creates forward momentum, it resists rotation of the rocket.

$$C_{PDM} = \dot{m} (d_{tip, nozzle} - COG)^2 \quad (110)$$

### Jet Damping - Dimensional Analysis

$$\dot{m} (d_{tip, nozzle} - COG)^2 : \left[ \frac{kg}{s} \cdot m^2 \right]$$

$$M = fd : \left[ \frac{kg \cdot m^2}{s^2} \right]$$

Note: why is the *Jet Damping Moment* missing a 1/t?

[Damping Moment Coefficient - Source](#)

### Pitch Damping Moment

The *Pitch Damping Moment* is a moment opposing the *Rocket Restoring Moment* and dampens the oscillation.

$$0.55 \frac{l^4 r_t}{A_{ref} d} \frac{\omega^2}{v_0^2} \quad (111)$$

According to [4], the *Pitch Damping Moment* is essentially insignificant until near apogee. This is because it is proportional to  $\frac{\omega^2}{v_0^2}$  (as seen in 111), which will be near zero until apogee due to very small angular velocities made smaller by squaring the  $\omega$  term.

The *Pitch Damping Moment* of each rocket component must be calculated individually. For instance, the *Pitch Damping Moment* of a fin is as follows.

$$C_{damp} = 0.6 \frac{N A_{fin} d_{COP}^3}{A_{ref} d} \frac{\omega^2}{v_0^2} \quad (112)$$

## Derivation of the Harmonic Motion Equation

Suppose a high-powered rocket is launched in quiescent air vertically, and flies straight without wobbling. Then, suppose a small and momentary disturbance (e.g. a short gust of wind) is experienced on the side of the rocket causing an angular deflection, . If the rocket is *stable*, a restoring force causes a *corrective moment* which will act in the opposite direction of the deflection. This *corrective moment* can be considered a function of angular displacement [2].

$$M_{corrective} = F(\alpha) \quad (113)$$

As the rocket gains velocity in the direction opposite the disturbance, a *damping moment* is generated as a result of the relative speed of the air, in the direction orthogonal to the longitudinal axis. As this *damping moment* opposes the angular velocity caused by the *corrective moment*, its sign is opposite to the angular velocity. The *damping moment* is also a function of angular velocity [2].

$$M_{damping} = G(\omega) \quad (114)$$

Then, taking a sum of Moments, the rotation of the rocket can be described as follows [2]

$$\begin{aligned} I\lambda &= -F(\alpha) - G(\omega) \\ I \left( \frac{d^2\alpha}{dt^2} \right) &= -F(\alpha) - G \left( \frac{d\alpha}{dt} \right) \\ I \left( \frac{d^2\alpha}{dt^2} \right) + F(\alpha) + G \left( \frac{d\alpha}{dt} \right) &= 0 \end{aligned} \quad (115)$$

This nonlinear, homogenous, differential equation can not be solved exactly [2].

## Linearization Approximation

Linear Approximations of 115 are made considering small values of  $\alpha$  and  $\omega$ , known as the *small-perturbation theory* [2]. This linearization process provides ~~constant~~ coefficients, which we will denote  $C_1$  for the *Corrective Moment Coefficient* and  $C_2$  for the *Damping Moment Coefficient*.

$$\begin{aligned} F(\alpha) &\approx C_1 \cdot \alpha \\ G \left( \frac{d\alpha}{dt} \right) &\approx C_2 \cdot \frac{d\alpha}{dt} \\ I \left( \frac{d^2\alpha}{dt^2} \right) + C_1(\alpha) + C_2 \left( \frac{d\alpha}{dt} \right) &= 0 \end{aligned} \quad (116)$$

## General Homogeneous Response

The characteristic, linearized, homogeneous yaw/pitch response is given as:

$$I_L \frac{d^2\alpha_x}{dt^2} + C_2 \frac{d\alpha_x}{dt} + C_1\alpha_x = 0 \quad (117)$$

[2]

A solution over a known range of acceptable values of the coefficients above is:

$$\alpha_x = Ae^{-Dt} \sin(\omega t + \phi) \quad (118)$$

Where:

- $t$  is the time passed since the “observation of the dynamic response has begun, not the time elapsed since the rocket was launched” [2]
- $\omega$  is the *frequency of oscillation* (not literally the angular velocity of the rocket)

$$\omega = \sqrt{\frac{C_1}{I_L} - \frac{C_2^2}{4I_L^2}} \quad (119)$$

- $\phi$  is the *phase angle* in radians

$$\phi = \arctan \left( \frac{\alpha_{xo}\omega}{D\alpha_{xo} + \Omega_{xo}} \right) \quad (120)$$

- $D$  is the *inverse time constant*

$$D = \frac{C_2}{2I_L} \quad (121)$$

- $A$  is the *initial amplitude*

$$A = \frac{\alpha_{xo}}{\sin\phi} \quad (122)$$

- $\alpha_{xo}$  is the value of  $\alpha_x$  at  $t = 0$

[2]

The initial conditions for this solution are

- $\alpha_{xo} = \text{some non-zero angle-of-attack}$
- $\Omega_{xo} = 0$

This equation is represented in the model as follows

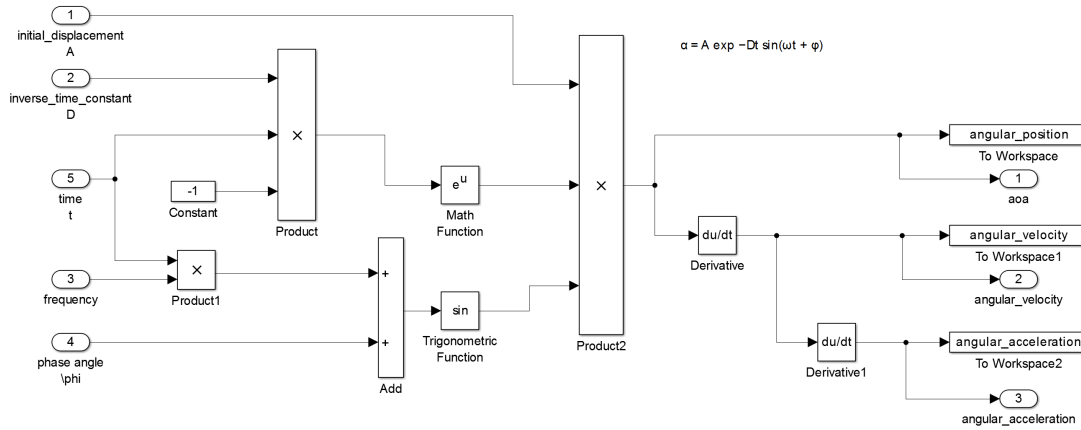


Figure 11: Angular Flight Model - Simplified

$$\begin{aligned} I_L(D^2 - \omega^2) - C_2D + C_1 &= 0 \\ -2I_LD + C_2 &= 0 \end{aligned}$$

[2]

A rocket can be considered restored from a disturbance if the angle of attack decays to 5% of the initial amplitude [2].

[2]

The *Natural Frequency* of the rocket at the current air speed for the homogeneous solution is

$$\omega_n = \sqrt{\frac{C_1}{I_L}} \quad (123)$$

Note: it would appear that this response only reflects the physical system for non-decreasing values of  $C_2$ , which would cause the exponential term to increase with time and cause the amplitude to grow. Although the damping coefficient remains relatively constant, the inverse time-constant is only a function of  $\frac{C_2}{2I_L}$ . As velocity decreases in the rocket coasting phase,  $C_2$  drops proportional to the square of the velocity and thus the inverse-time constant decreases enough with respecting time, that  $Dt$  is decreasing and thus  $e^{-Dt}$  will begin to increase. This is accounted for by the drift velocity at apogee. While the rocket has a zero climbing velocity, it does still travel laterally and the total velocity contributes to the damping moment coefficient, maintaining stability.

## Complete Response to Step Input

## Complete Response to Impulse Input

### Delta-Dirac Function

$$u(t) = \int_{-\infty}^{\infty} \delta(u - \tau) d\tau \quad (124)$$

### Convolution Theorem

## Steady State Response to Sinusoidal Forcing

### Rocket Damping Ratio

The *Rocket Damping Ratio* is calculated as follows.

$$\zeta = \frac{C_2}{2 \cdot \sqrt{C_1 I_L}} \quad (125)$$

Where:

- $C_1$  is the *Corrective Moment Coefficient*
- $C_2$  is the *Damping Moment Coefficient*
- $I_L$  is the *Longitudinal Moment of Inertia*

[2]

### Underdamped Case

$$0 < \frac{C_2^2}{4I_L^2} < \frac{C_1}{I_L} \quad (126)$$

The fastest response is when  $\zeta = \frac{\sqrt{2}}{2}$

### Overdamped Case

$$\frac{C_2^2}{4I_L^2} > \frac{C_1}{I_L} \quad (127)$$

### Critically Damped Case

$$\frac{C_1}{I_L} = \frac{C_2^2}{4I_L^2} \quad (128)$$

[2]

## Rocket Natural Frequency

$$\omega_n = \sqrt{\frac{C_1}{I_L}} \quad (129)$$

Where:

- $C_1$  is the *Corrective Moment Coefficient*
- $I_L$  is the *Longitudinal Moment of Inertia*

[2]

## Time Constants of the Response

### Complete response to step input

### Complete response to impulse input

[2]

## AOA as a function of velocity

In order to plot the real system behavior, it may be possible to solve Equation 115 where  $\alpha_x$  is a function of velocity, and solve for  $\alpha_x$  by twice integrating  $\ddot{\alpha}_x$ .

Since AOA is a function of total velocity through the *Corrective Moment Coefficient* and the *Damping Moment Coefficient*, it may be possible to solve the system by differentiating with respect to velocity, rather than by time.

$$I \left( \frac{d^2 \alpha}{dt^2} \right) + F(\alpha) + G \left( \frac{d\alpha}{dt} \right) = 0$$

$$\frac{\delta^2 \alpha_x}{\delta t^2} = \dot{v} = \ddot{x}$$

$$\frac{\delta \alpha_x}{\delta t} = v = \dot{x}$$

$$\frac{\delta \alpha_x}{\delta t} = v = \dot{x}$$

Eventually we get to:

$$\frac{d^2 \alpha_x}{dv^2} = \dots$$

## Corrections

### Compressibility Correction

*Barrowman's Method* neglects compressibility effects, however these effects cannot be neglected above Mach 0.3.

## Wind Disturbance

We are interested in the damping ratio of the rocket as it stabilizes towards *zero angle of attack* in reaction to angular disturbances.

As we consider the rocket to have ideal dimensional accuracy, the main source of flight disturbance is wind.

## Impulse Disturbance

We can test the ability of the rocket to stabilize due to an initial angular disturbance, by applying an initial angle of attack. This simulates a small gust of wind hitting the rocket just as it takes off.

## Constant Disturbance

We can test the ability of the rocket to stabilize due to a constant disturbance force, as well as applying an initial *angle of attack*. This simulates a constant wind force coming from a single direction. As the density of air goes down with increases altitude, this assumes that the wind speed picks up at higher altitudes to maintain the constant wind force.

Alternatively, we could model a constant wind speed of  $8.33m/s$ , and apply the ISA Model for the density as a function of altitude to determine the changing wind force as the rocket climbs.

## More Reading

There are further resources on rocket flight stability here: - [http://www.apogeerockets.com/Tech/Rocket\\_Stability](http://www.apogeerockets.com/Tech/Rocket_Stability)

## Solver Algorithm

The numerical algorithm chosen for performing the simulation can impact the accuracy of the results. The following discussion introduces available algorithms and their tradeoffs.

### Algorithms

#### Interpolation

#### Matlab Cubic Spline

#### Runge-Kutta-Fehlberg

Integration method order: Integration step size: Convergence Tolerance

## Validation

The engineering flight simulation was tested comprehensively.

## Unit Testing

It was ensured that individual Matlab functions provided the output expected, and that Simulink blocks employing the Matlab functions worked on a unit level - Reference Models were employed for this purpose

## Integration Testing

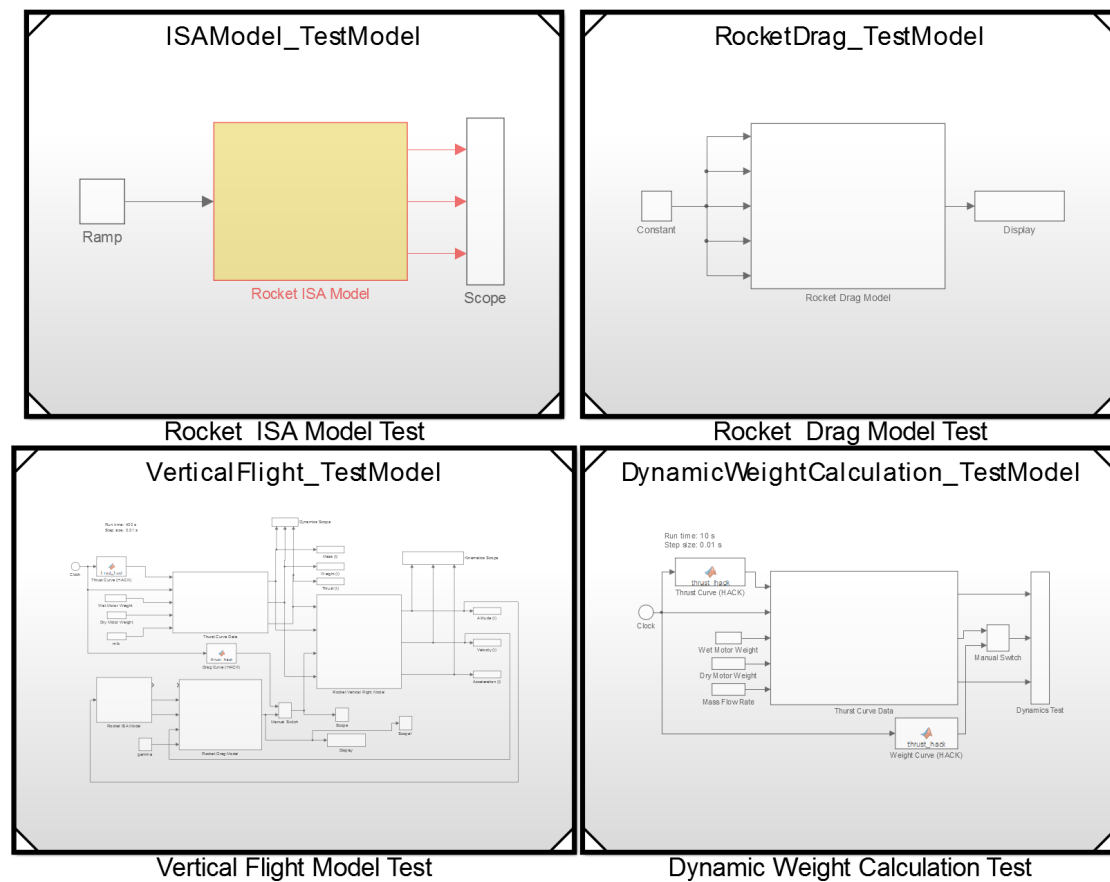


Figure 12: All Integration Tests

## System Testing

The simulator was tested on a system level by simulating the CR\_2-4G rocket flight on all available simulators and comparing all possible results. These results are discussed in detail in the *Simulation Execution* section.

## Comparison with Experimental Data

So far, much of the analysis has relied on theory and comparison with other simulator output. It is important to compare our models with real data to provide the best measure of their accuracy. Without access to a wind tunnel, we determined that we could test our *ISA Model* implementation against weather balloon data, and the final flight performance with actual rocket launches where data is available.

### ISA Model

Our ISA Model was compared to weather balloon data taken as close as possible to the launch site. The sites chosen were Salt Lake City, Utah (SLC) and Tuscon, Arizona (TUS), and data was provided by University of Wyoming Department of Atmospheric Science [14].

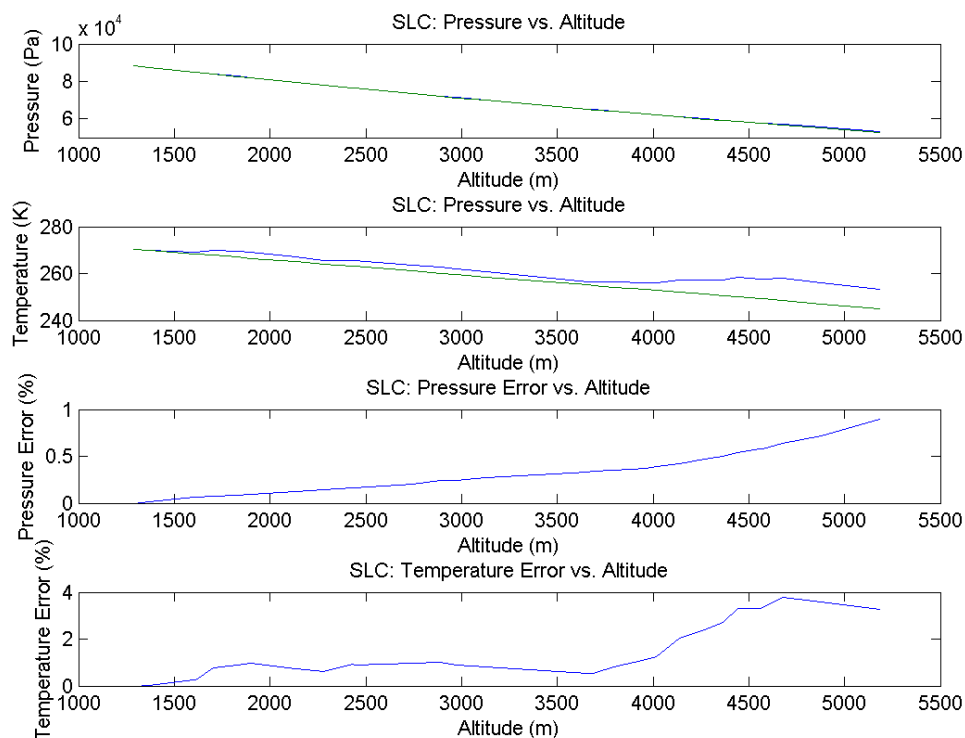


Figure 13: Comparison of ISA Model with SLC Weather Balloon Data

The figures show an extremely close alignment with real conditions, both falling below 1% error

## Flight Validation

### University of Louisville

The University of Louisville was kind enough to share experimental rocket flight data, as well as their rocket dimensions so that our simulation could be compared to real-world experimental



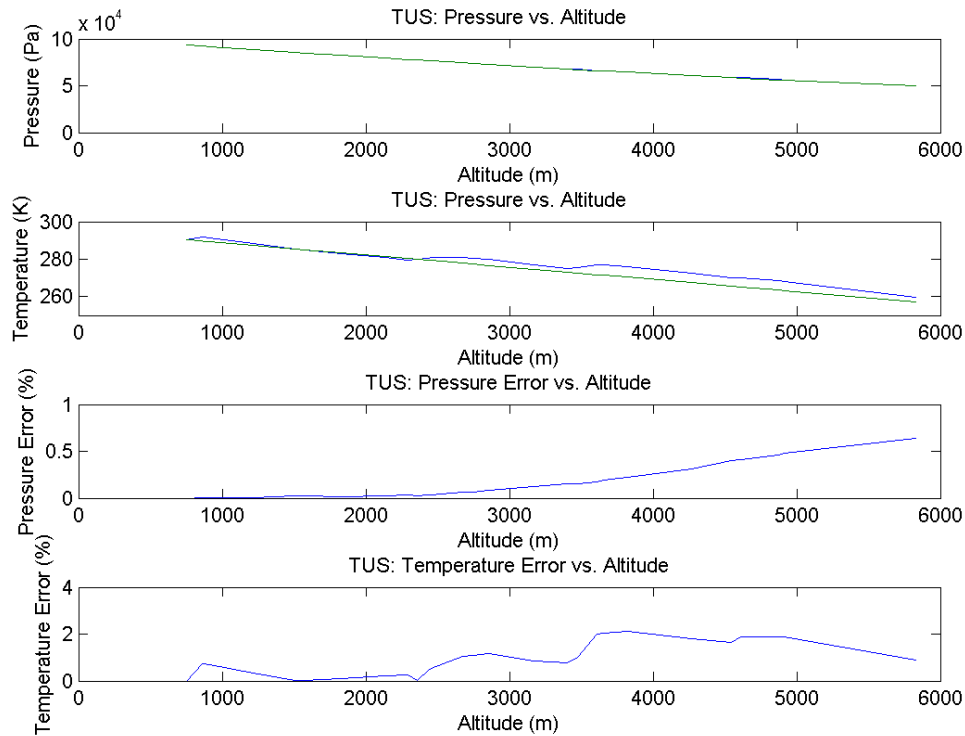


Figure 14: Comparison of ISA Model with TUS Weather Balloon Data

data.

## Flight Details

March 29, 2015

Date	Elevation	Ground Temperature	Wind Speed	Humidity	Launch Guide Length
2015/03/29	405 ft (123.48 m)	289.26 K (61.00 °F)	14 km/h (3.89 m/s)	~39 %	3.05 m
2015/03/29	432 ft (131.71 m)	287.71 K (58.21 °F)	14 km/h (3.89 m/s)	~39 %	3.05 m
2015/03/29	410 ft (125.00 m)	294.70 K (70.79 °F)	14 km/h (3.89 m/s)	~39 %	3.05 m
2015/03/29	417 ft (127.13 m)	289.44 K (61.32 °F)	14 km/h (3.89 m/s)	~39 %	3.05 m
2015/03/29	446 ft (135.97 m)	289.54 K (61.50 °F)	14 km/h (3.89 m/s)	~39 %	3.05 m

## Motor Details

Cesaroni L935

Parameter	Value
Manufacturer:	Cesaroni Technology
Entered:	Oct 6, 2009
Last Updated:	Jun 26, 2014
Mfr. Designation:	3147L935-P
Common Name:	L935
Motor Type:	reload
Diameter:	54.0mm
Length:	64.9cm
Total Weight:	2542g
Prop. Weight:	1567g
Cert. Org.:	Canadian Association of Rocketry
Cert. Date:	Aug 27, 2009
Average Thrust:	933.8N
Maximum Thrust:	1585.6N
Total impulse:	3146.8Ns
Burn Time:	3.4s
Isp:	205s
Case Info:	Pro54-6GXL
Propellant Info:	Imax
Data Sheet:	<a href="#">link</a>
Availability:	regularCesaroni L935

## Plots

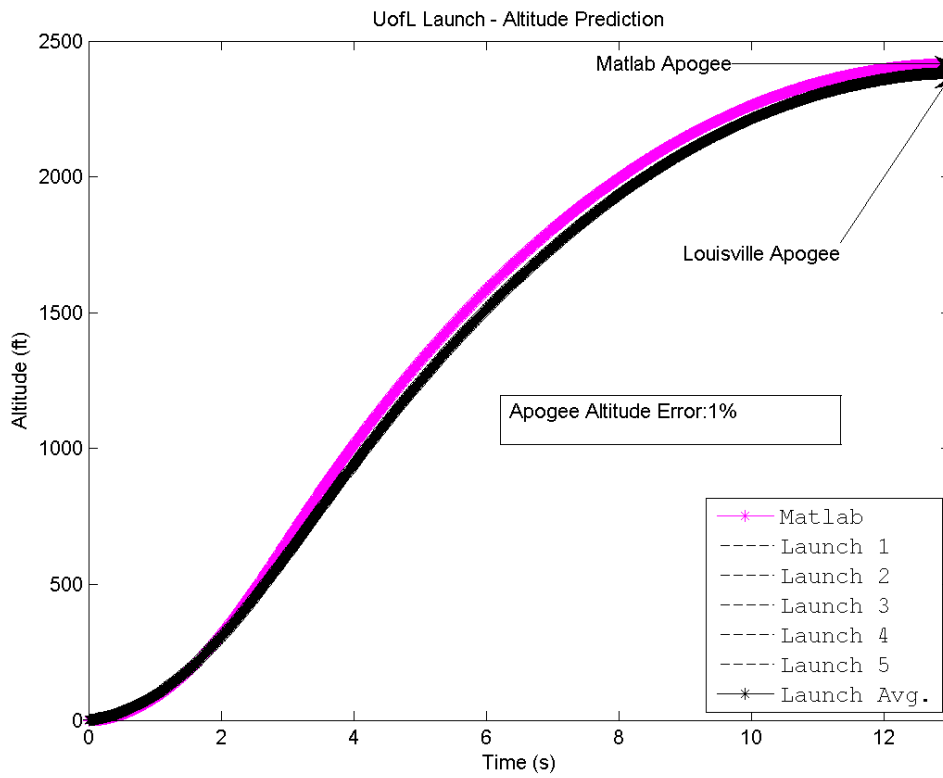


Figure 15: Altitude Plot of Simulation Data vs Louisville Rocket Launches

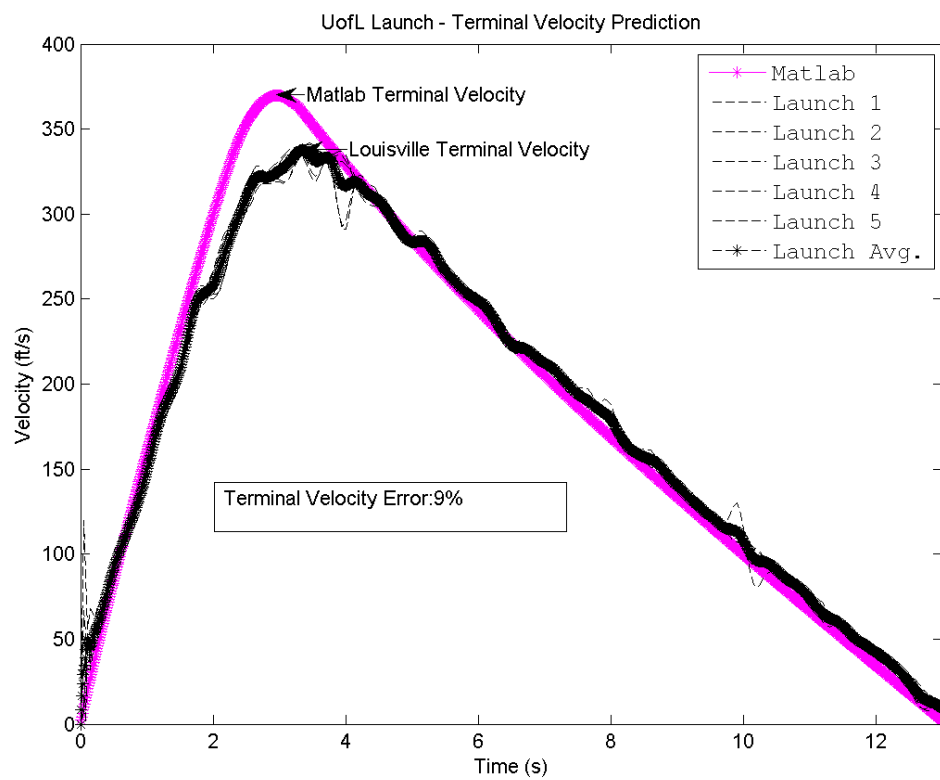


Figure 16: Velocity Plot of Simulation Data vs Louisville Rocket Launches

## Comparison with Arcturus Rocket

The rocket flown at last years competition was also modeled. As shown in Figures 17 and 18, a high degree of accuracy with the avionics data is further confirmed.

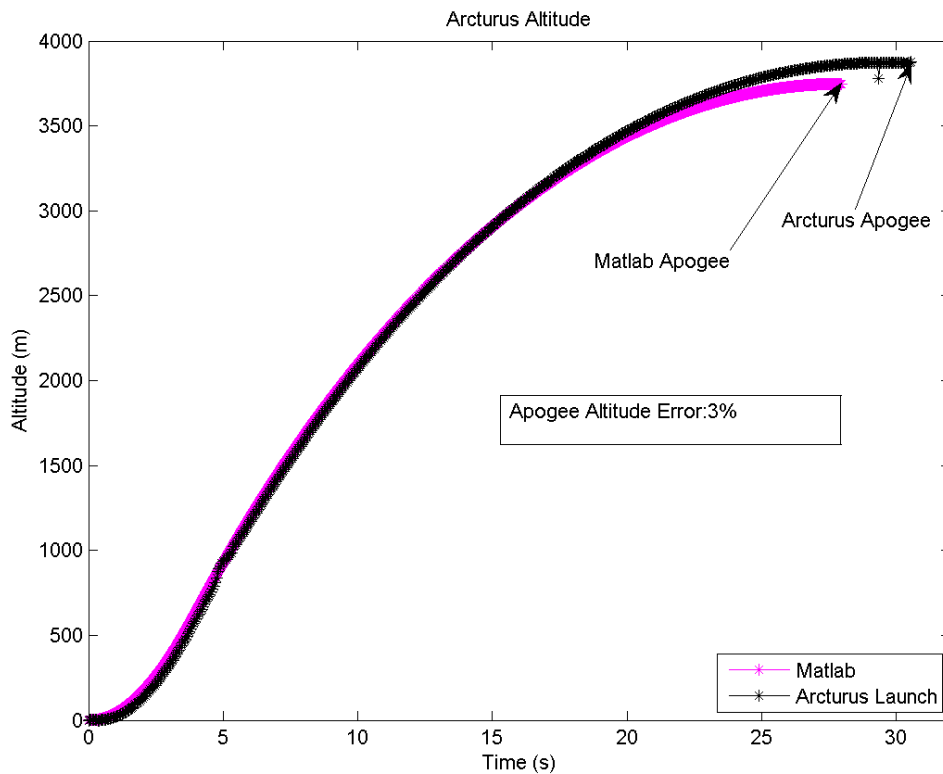


Figure 17: Altitude Plot of Simulation Data vs Arcturus Rocket Launch

## Summary of Comparison with Experimental Data

Data Source	Parameter Tested	% Error
Weather Balloon Data (Salt Lake City, UT)	Pressure with Altitude	< 1%
Weather Balloon Data (Tuscon, AZ)	Pressure with Altitude	< 1%
University Of Louisville (avg. of 5 Launches)	Altitude	1%
University Of Louisville (avg. of 5 Launches)	Velocity	9%
Arcturus Flight Launch	Altitude	3%
Arcturus Flight Launch	Velocity	1%

Table 12: Summary of Comparison with Experimental Data

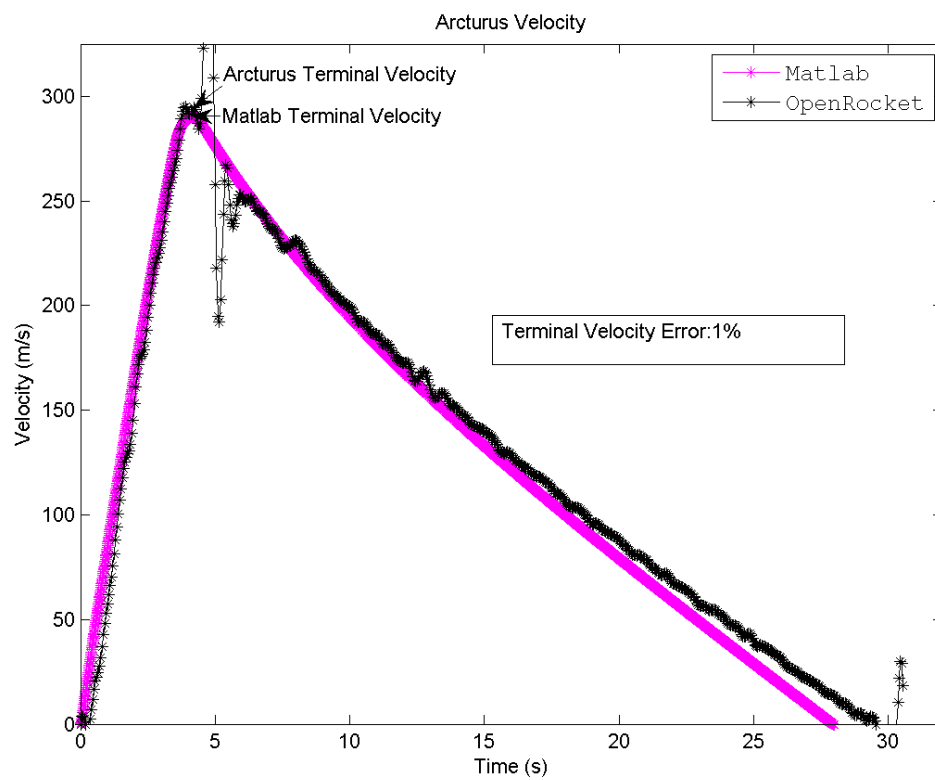


Figure 18: Velocity Plot of Simulation Data vs Arcturus Rocket Launch

## Simulation Execution

### Simulation Configuration

#### Historical Weather Data for Green River, Utah

The following conditions are historical data for Green River, Utah on June 25th, 2015 at 12:00PM (noon) [15].

Date	Elevation	Ground Pressure	Ground Temperature	Wind Speed	Humidity
2015/06/15	4,078 ft (1,243 m)	101300 Pa	298.15 K (25 °C)	6 km/h (1.6 m/s)	33 %
2014/06/15	4,078 ft (1,243 m)	100700 Pa	296.15 K (23 °C)	20 km/h (5.56 m/s)	8 %
2013/06/15	4,078 ft (1,243 m)	101000 Pa	299.15 K (26 °C)	10 km/h (2.78 m/s)	19 %

Table 14: Historical Weather Conditions, Green River, Utah

#### General Conditions

Elevation	Humidity	Launch Guide Length
4300 ft (1,311 m)	33 %	5.4864 (18 ft)

Table 16: General Simulation Conditions

#### Best Case

The best case scenario is with no wind, and a 0° launch angle, and the lowest air pressure.

Wind Speed	Ground Pressure	Ground Temperature	Launch Guide Angle
0 m/s	101000 Pa	298.15 K (25 °C)	0°

Table 18: Best Case Simulation Conditions

#### Worst Case

The worst case scenario, is the maximum wind condition permitted for launch by the competition, and a launch guide angle, and the highest pressure.

Wind Speed	Ground Pressure	Ground Temperature	Launch Guide Angle
8.33 m/s	101325 kPa	303.15 K (30 °C)	10°

Table 20: Worst Case Simulation Conditions

## Simulation Execution

Identical rocket designs were implemented in OpenRocket, RockSim, RASAero, and our Engineering Simulator. The 3rd party simulator results were parsed with Matlab to determine essential performance criteria and to compare with our model.



## Matlab

### Matlab Models

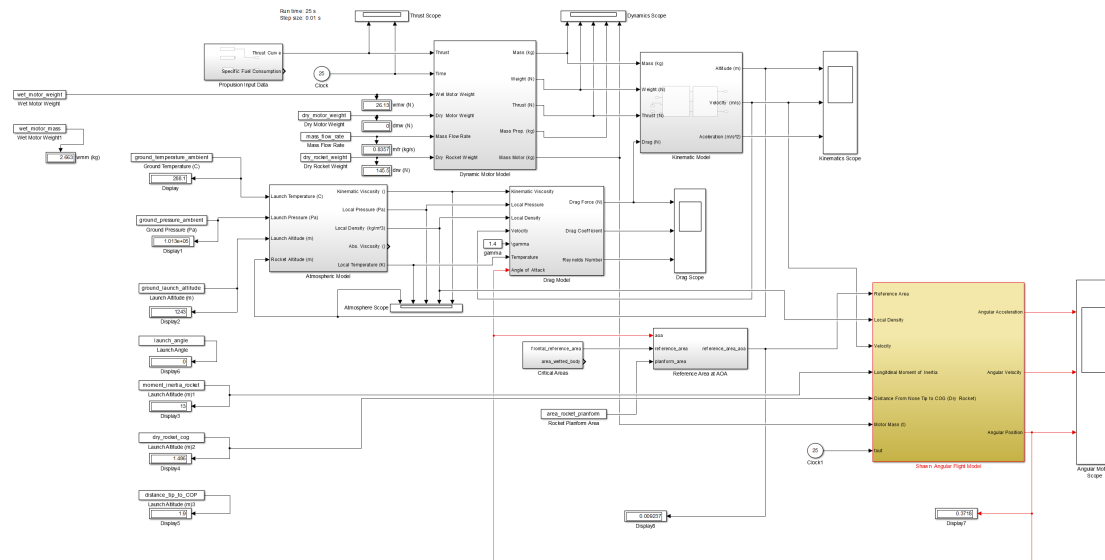


Figure 19: Full Model in Simulink, angle-of- attack less than 15 degrees

## Observations

Figure 20 shows the predicted Altitude of the rocket compared against OpenRocket, RASAero, and Rocksim. It would appear that RASAero and Rocksim predict a higher altitude, perhaps considering their underestimation of the drag forces, shown in Figure 9.

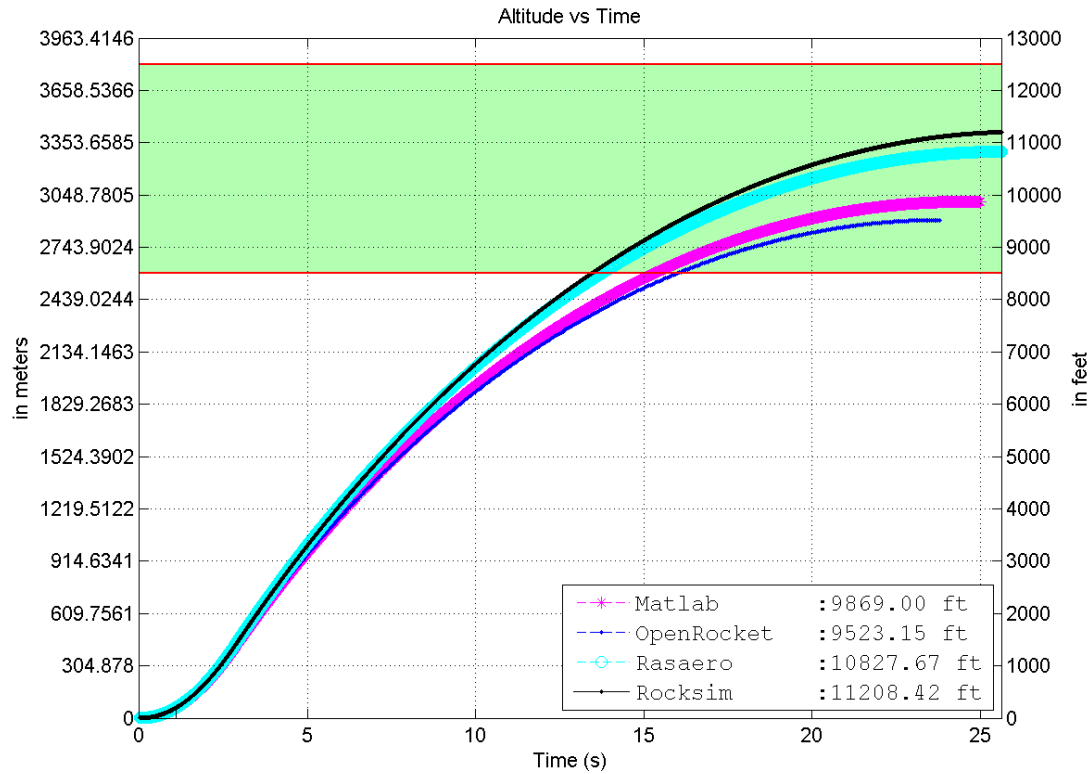


Figure 20: Altitude as a Function of Time

2e Vehicle reaches 10,000 ft altitude (+1000 feet / - 0 feet)

Figure 21 shows that the Mach number predicted by each software is quite close.

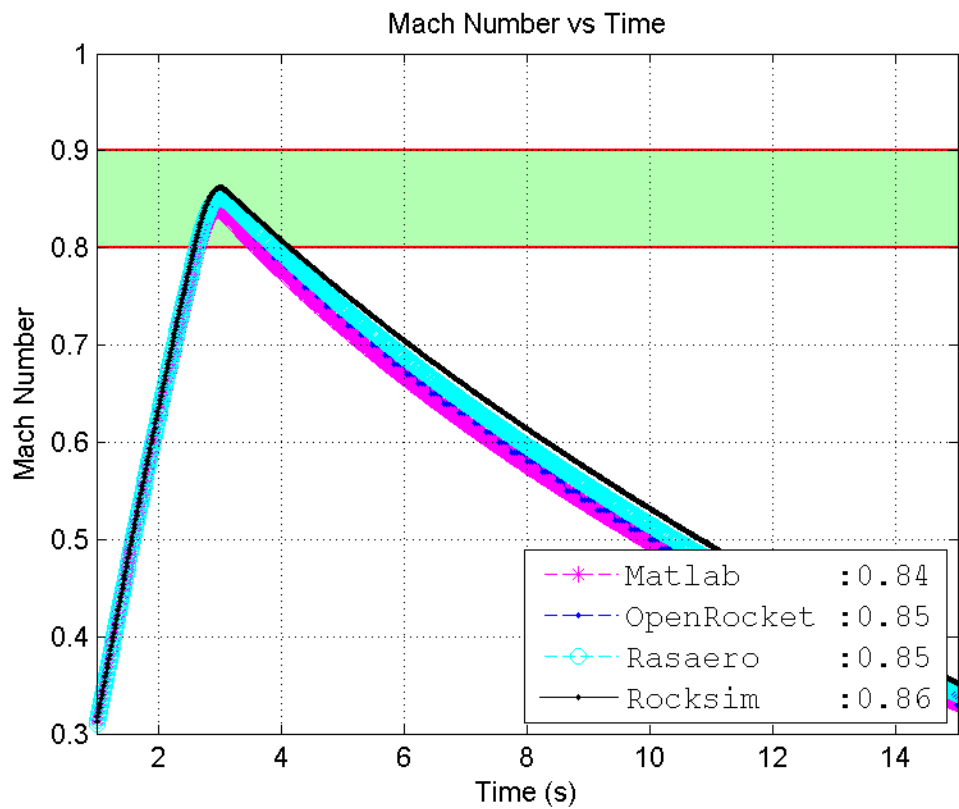


Figure 21: Mach Number as a Function of Time

2d Vehicle max speed mach 0.9

Figure 22 shows that the dynamic stability is quite similarly predicted by all tested simulators. OpenRocket shows a continuous oscillation, which according to my current analysis of their methodology, perhaps does not correctly consider the oscillation damping encountered during flight. In any case, if the OpenRocket model were to be 100% correct, the dynamic stability criteria would still be satisfied by a wide margin.

2a Static stability above 2 calibers

2b Dynamic stability above 0

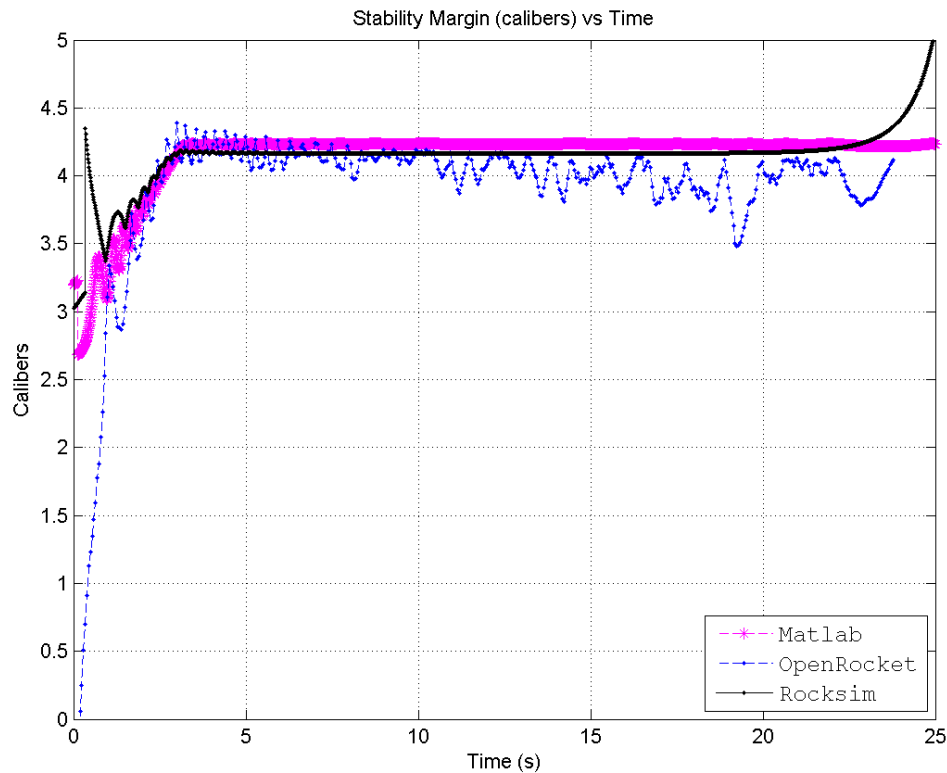


Figure 22: Stability (Calibers) as a Function of Time

As seen in Figure 24, once the rocket leaves the launch pad, the angular frequency of the rigid-body oscillation does not approach the natural frequency of the rocket, confirming requirement:

2f - The vehicle does not experience resonant pitching/yawing motion in flight

In any case, we know that resonant oscillation at the natural frequency does not occur, since the rocket stabilizes. Figure 23 shows the stabilization of the angle-of-attack with time.

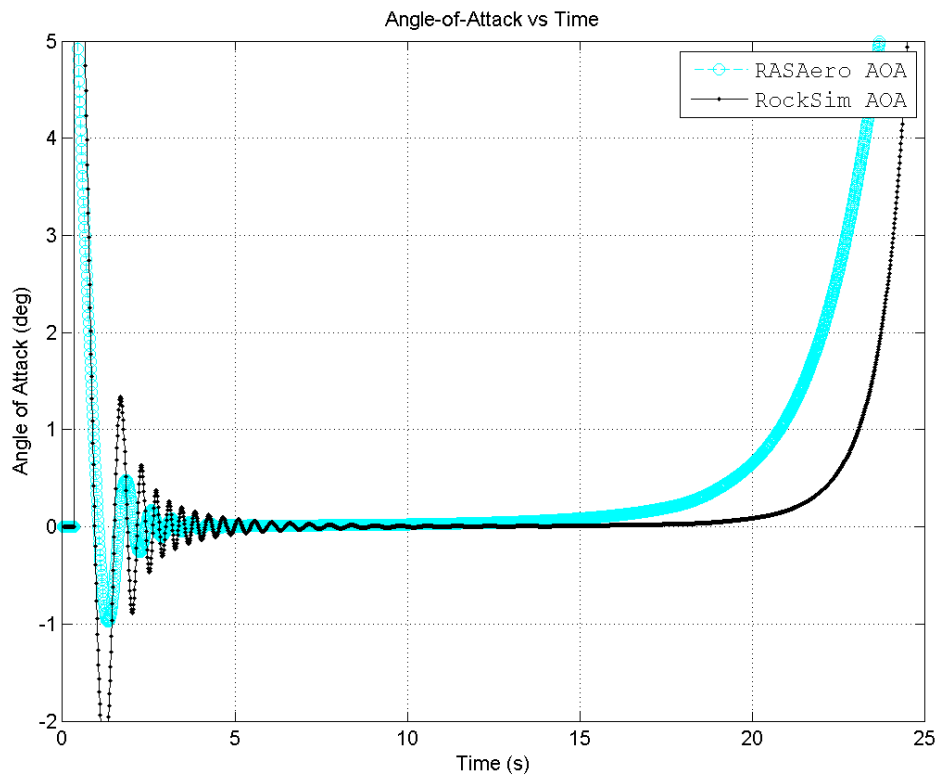


Figure 23: Angle of Attack Stabilization

RockSim and RASAero are chosen for this plot since they have a more mature stability methodology.

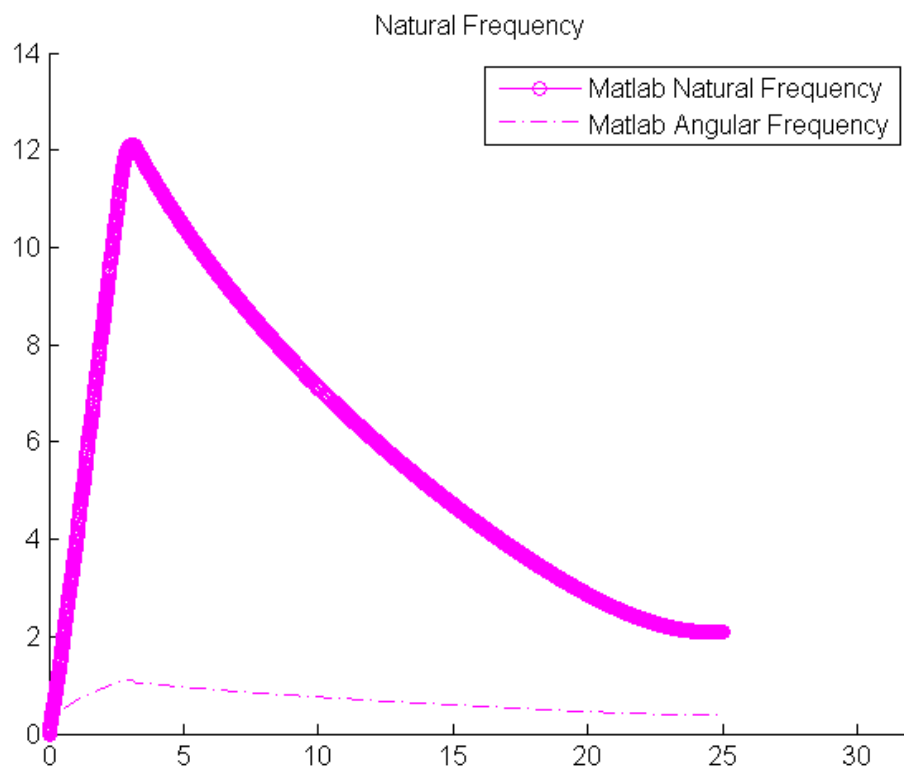


Figure 24: Natural Frequency

## Simulation Summary

Ideal Case Competition Case	Altitude at Apogee	Highest Mach Number	Velocity leaving launch rail	Static Stability Margin	Dynamic Stability Margin	Damping Ratio
RASAero	11,372.39 ft 10,827.67 ft	0.84 0.85	33.5 m/s 33.5 m/s	> 3	N/A	N/A
Rocksim	11,190.78 ft 11,208.42 ft	0.86 0.86	33.5 m/s 33.5 m/s	> 3	> 3.0 < 4.5	~ 0.05
OpenRocket	10377.92 ft 9672.72 ft	0.84 0.83	33.7 m/s 33.8 m/s	> 3	> 0.5 < 4.5	~ 0.3
Matlab	10,364.00 ft 9881.65 ft	0.84 0.84	33.7 m/s 33.6 m/s	> 3	> 2.5 < 4.5	~ 0.05
Requirement	10,000 ft +- 2500 ft ✓	< 0.9 ✓	> 32.8 m/s ✓	> 2 ✓	> 0 ✓	< 0.707 ✓

Figure 25: Simulation Summary

## Future Enhancements

### Hardware-in-the-loop

### Porting to Python / OpenModelica

### Plotting with Plot.ly

### Robust Wind Model

To account for wind turbulence in future models, two commonly used Wind Models are explored.

#### Kaimal Wind Model

$$\frac{S_u(f)}{\sigma_u^2} = \frac{4L_{1u}/U}{(1 + 6fL_{1u}/U)^{5/3}} \quad (130)$$

#### Von Karman Wind Model

$$\frac{S_u(f)}{\sigma_u^2} = \frac{4L_{2u}/U}{(1 + 70.8(fL_{2u}/U)^2)^{5/3}} \quad (131)$$

Where

- $\frac{S_u(f)}{\sigma_u^2}$  is the *Spectral Density Function* of turbulence velocity
- $f$  is the turbulence frequency
- $\sigma_u$  is the standard deviation fo the turbulence velocity
- $L_{1u}$  and  $L_{2u}$  are length parameters
- $U$  is the average wind speed

[4]

## ThrustCurve.org API Integration

ThrustCurve.org has an API we could use to dynamically pull motor data for integration into the simulation

## Rocket Orientation

While good results have been achieved thus far, a 6DOF simulator is preferable to produce the most realistic simulation possible. It would be possible to account for wind turbulence and other disruptions which are not confined to a single axis. If all forces and moments can be clearly defined, it also allows a seamless coupling of the point-mass and rigid body rotation systems described earlier, as well as the pitch/yaw and roll systems.

Building on the material explored in the Angular Stability model, this section introduces topics relating to the orientation of the rocket during flight.



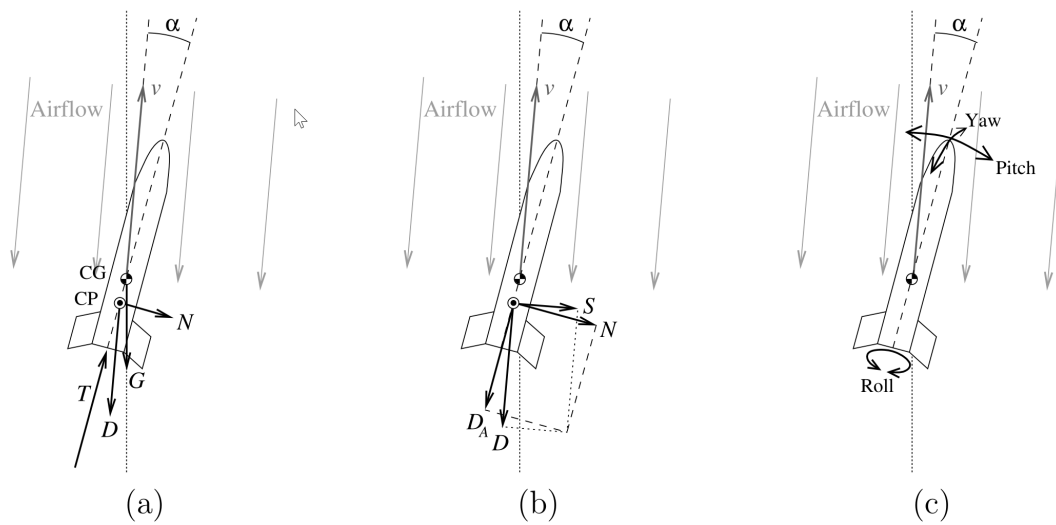


Figure 3.1: (a) Forces acting on a rocket in free flight: gravity  $G$ , motor thrust  $T$ , drag  $D$  and normal force  $N$ . (b) Perpendicular component pairs of the total aerodynamical force: normal force  $N$  and axial drag  $D_A$ ; side force  $S$  and drag  $D$ . (c) The pitch, yaw and roll directions of a model rocket.

Figure 26: Forces and Moments Experienced by rocket in flight

## Rotations

*Spherical Coordinates* and *Euler Angles* are commonly used to describe the orientation of an object, however both systems encounter singularities where the orientation is ambiguous. Special cases are required to handle these singularities, which complicate the analysis and programming.

## Quaternions

*Quaternions* are commonly used to describe spatial rotation, avoiding singularities.

An initial position is taken as reference to describe subsequent changes in orientation. Vectors can be transformed from rocket coordinates to world coordinates, and the reverse.

Leonhard Euler proved that

$$e^{i\phi} = \cos \phi + i \sin \phi \quad (132)$$

Thus,  $e^{i\phi}$  lies on the unit circle in the complex plane, and has a unit length.

Multiplication

$$(a + bi)(c + di) = re^{i\phi} se^{i\theta} = rse^{i(\phi+\theta)}$$

$$i^2 = j^2 = k^2 = ijk = -1$$

Quaternions

$$H = a + bi + cj + dk : a, b, c, d \forall R$$

$i, j, k$  are all square roots of  $-1$

$$ij = k = -ji$$

$$jk = i = -kj$$

$$ki = j = -ik$$

If we consider three-dimensional space to be purely imaginary quaternions:

$$R^3 = xi + yj + zk : x, y, z \forall R$$

Rotations are done using unit quaternions

$$\cos \phi + i \sin \phi$$

$$\cos \phi + j \sin \phi$$

$$\cos \phi + k \sin \phi$$

Which can be rewritten as

$$e^{i\phi}, e^{j\phi}, e^{k\phi}$$

For example, taking any arbitrary unit quaternion (*vector*)  $\mathbf{u}$

$$u = u_1i + u_2j + u_3k = e^{u\phi}$$

We can rotate another arbitrary vector  $\mathbf{v}$  about the axis in the  $\mathbf{u}$  direction

$$e^{u\phi} v e^{-u\phi}$$

<http://math.ucr.edu/~huerta/introquaternions.pdf>

## Parameters needed for quaternion analysis

- rocket mass
- **reference length**
- angle-of-attack
- reference area
- longitudinal moment of inertia
- radial moment of inertia
- thrust force
- drag force
- weight force
- pitching moment
- pitch damping moment
- yawing moment
- yaw damping moment
- ~~roll moment~~
- ~~roll forcing moment~~
- ~~roll damping moment~~
- normal force
- side force
- pitch rate
- yaw rate
- roll rate
- wind velocity

## Rocket Moments

A *Pitch Moment* and *Pitch Damping Moment* are defined, which are different than the *Corrective Moment Coefficient* and the *Damping Moment Coefficient*. Note: a complementary *Yaw Moment* and *Yaw Damping Moment* are implied, with exactly the same considerations for motion along the uncoupled complementary yaw-axis.

### Pitch Moment

The *Pitch Moment* is taken from the tip of the nose cone, it must be moved to the COG to mirror *Corrective Moment Coefficient*.

$$M_{pm}(x) = -\frac{1}{2}\rho v^2 C_m A_{ref} * L_{ref} \quad (133)$$

TODO why minus?

Where:

- $\rho$  is the local atmospheric density
- $v$  is the rocket velocity relative to the surrounding air
- $A_{ref}$  is the reference area (frontal or side?)
- $L_{ref}$  is ???

### Pitch Moment Coefficient

$$C_m = C_m - F_s \cdot COG \cdot L_{ref} \quad (134)$$

Where:

- $F_s$  is the side force

### Fin-set Pitch Moment Coefficient

$$M_{pm} = \frac{C_N \times COP}{L_{ref}} \quad (135)$$

### Body Tube and Fin Set Interference Pitch Moment Coefficient

$$M_{pm} = \frac{C_N \times COP_x}{L_{ref}} \quad (136)$$

### Barrowman Calculation

$$total.getCM + forces.getCM \quad (137)$$

### Final Solver Pitch Moment Coefficient

$$M_{pm} = M_{pm} + (\text{randomness}) \quad (138)$$

Some randomness is added to avoid an “over-perfect” solution

### Pitch Damping Moment

- significant only near apogee

## Stochastic Simulations

The simulator described in this paper is deterministic. It assumes that all input parameters are known and produces the same simulation result each time it is run, as long as no parameters are directly changed. While it was acknowledged that conditions may vary and produce different launch outcomes, only the extreme expected cases were tested - a multitude of possibilities exist in between.

A convincing and complete engineering simulation must account for the uncertainties of certain variables. In high-powered rocket flight, there are many such uncertainties. For example, any error in the shaping of the fins introduces roll during flight, which has many influences on the rocket. Additionally, wind turbulence is extremely difficult to predict, and can have impacts all all stages of the rocket flight which may change its directory. Temperature is a significant factor for the performance of the motor - for instance, its total impulse and burn time are sure to be affected at high temperatures.

The variation of all these parameters together creates a great deal of uncertainty, which must be accounted for in a robust engineering simulation.

Stochastic methods such as the *Monte Carlo* method randomize these parameters and provides a range of uncertainty from which it is possible to consider the probability that a given simulation outcome will occur in reality.

## Conventions

### Data Model

The *Data Model* provides static and dynamics parameters as needed by other models in the simulation.

## Static Parameters

Many parameters are constant throughout the simulation, notably the structural dimensions. All structural dimensions are generated in the CATIA Design and output to a spreadsheet, which the simulation will load and place in the Matlab workspace.

This instance can be accessed by multiple models to clearly and effectively provide parameter access.

## Dynamic Parameters

As discussed in the *Dynamic Parameters* section, many parameters are changing due to flight conditions. An additional *Map Container* instance is created to handle and deliver these changes to the models that need them.

## Matlab Conventions

### Matlab/Simulink Libraries

#### Overview

The goal is to create a robust Simulink model that references Matlab code from files that can be tracked by versioning software (Git). The Matlab source should be editable and effect changes in the Simulink model.

By a combination of both, full versioning control can be achieved in the project.

### Creating a Library

1. Open Matlab
2. Open Simulink
3. Click File -> New -> Library

### Add to path

Permanently add your workspace to the Matlab path. At the command prompt:

```
>> pathtool
```

[Alternatively, try this howto](#)

### Add to Library Browser

[Add to Library Browser](#)

## Algebraic Loops

With systems that involve direct-feedthrough (feedback), it is common to encounter an algebraic loop, wherein the output of a function is also an input of the same function.

e.g.

$$u = f(u)$$

These can commonly be solved with a combination of *Atomic Subsystems*, *Initial Conditions*, or *Unit-Delay*. This matter is discussed thoroughly in the following guide

[Algebraic-Loops](#)

## Importing Data

Tabulated input data is relied upon to drive the simulation (see *Dynamic Parameters*). The following configuration is investigated to support this smoothly

[Recommended Methods for Importing Data](#)

[Load Signal Data for Simulation](#)

[Importing Signal Data in Simulink](#)

[Import Data Structures](#)

### From File

The *From File* block in Simulink allows incremental loading of data

The From File block reads data from a MAT-file and outputs the data as a signal. The data is a sequence of samples. Each sample consists of a time stamp and an associated data value.

[From File](#)

### Mat-File Versions

Data is read incrementally from Mat-File versions 7.3 and above [Mat-file Versions](#)

### nD Lookup Tables

### Specifying Time Data

[Specifying Time Data in Simulink](#)

## Versioning for Matlab Files

### Background

- Older versions allowed providing external ‘.m’ file for the *Matlab Function* block in Simulink
- Newer versions are shifting towards the embedded model, where Matlab code is compiled for execution on test hardware

[Naming conventions in Simulink for Matlab files changed after 2011A](#)

### Interpreted Matlab Function

*Interpreted Matlab Function* blocks are used to reference Matlab files so they can be versioned in Git. *Interpreted Matlab Function* blocks only accept one input and one output, therefore we must pass an array as input and an array as output. The contents of the array will contain our variables of interest. *(De))Mux* and *Bus* blocks may be useful to streamline the model.

1. Write your Matlab function
2. Create a Simulink Model
3. Add the *Interpreted Matlab Function* block
4. Double-click the added block, and enter the name of your function as directed. Select ‘OK’
5. Right-click the block, expand the ‘Mask’, and select ‘Create Mask’
6. Add the following in the ‘Icon Drawing Commands’ box `disp('function_name')`
7. Add a *Mux* block to combine your inputs into a single input array, and a *Demux* port to unpack your output array into outputs

## Versioning for Libraries

- Saving files as libraries and following the existing use cases in the documentation will allow robust versioning and collaboration workflow

## Unit Testing

### Simulink Unit Testing

## Model Referencing

Model Referencing shall be used to test all libraries for expected behavior.

1. Create a Test Model in which you drag the Library
2. Provide all test inputs and output assertion
3. Create another model to contain all the test cases created in 2
4. From the *Simulink Library*, drag a *Model Reference* block
5. Edit the *Model Reference* block, providing the name of the Test Model created in 2
6. Run the model created in 3 to verify the model referencing was successful

More information:

- <http://www.mathworks.com/videos/getting-started-with-model-referencing-68918.html>
- <http://www.mathworks.com/help/simulink/ug/creating-a-model-reference.html>
- <http://www.mathworks.com/help/simulink/slref/model.html>

## Exporting Images

High quality figures brings a great deal of value to a report. Simulink Models and Matlab figures can be exported to scalable vector graphics and PDF formats at high quality.

## GhostScript

[GhostScript](#) is needed to handle the EPS format. It can be downloaded [here](#).

## GhostScript and GIMP

GIMP has problems opening EPS files with the default configuration. Follow the instructions [here](#) to fix GhostScript in GIMP

## Exporting Figures

[export-fig](#) is a Matlab library which provides functions to output figures to various formats

## Exporting Simulink Models

[export Simulink models to publication-quality figures](#)

[publication quality graphics in Matlab](#)

[LaPrint](#)

[Howto](#)

## File Organization

- data
- documentation
- functions
- libraries
- models
- referencing
- scripts
- testing

### data

**data** → **csv**

### documentation

- documentation
  - images
  - template

The *documentation* folder contains all markdown files with project documentation. It also contains

**documentation** → **images**

Contains all images used in the documentation

**documentation** → **template**

Contains LaTeX/Pandoc/Markdown template and styling files

### functions

### libraries

### models

### referencing

### scripts

### testing

## Naming Conventions

### Variables

All variables must be lowercase, separated by underscores

e.g.

`wet_motor_weight`



## Functions

All *Matlab Functions* must be CamelCase

e.g.

DynamicWeightCalculation

## Models

All *Matlab Model* names must be CamelCase, and end in the word 'Model'

e.g.

DynamicWeightCalculationModel

## Libraries

All *Matlab Library* names must be CamelCase, and end in the word 'Library'

e.g.

DynamicWeightCalculationLibrary

# Documentation Conventions

## Markdown

Markdown is a markup language that is meant to be easy to read and easy to write, as well as easy to convert to HTML, LaTeX, PDF, and other output types.

## Python

Python is used to enable additional filters which handle features currently not supported out-of-the-box by Pandoc

[Download Python for Windows here](#)

## Pandoc

Pandoc is a document converter that in our case is useful in converting the Markdown (.md) files into PDF and HTML

[The User Guide is very helpful](#)

## Haskell

Haskell is useful in this environment to do some custom scripting

## LaTeX

LaTeX is a powerful typesetting language useful for academic writing. Its mathematical expressions are particularly useful for this report.

## Citations

[Excellent citation discussion](#)

[Haskell and Bibtex in Pandoc](#)

[IEEE CSL File](#) [Another IEEE CSL File](#)

## Equations

Wrap functions as follows to enable automatic numbering:

```
\begin{equation}
\label{my_equation}
f(x) = \int \cdots e^{xy}
\end{equation}
```

You can refer to the equation by the label you assigned to it

This comment refers to equation `\ref{my_equation}`

[LaTeX equations in Markdown+Pandoc](#)

## Figures

To automatically number figures, use the following syntax to insert an image:

```
[rocket_drag_model_overview]: images/rocket_drag_model_overview.png "Rocket Drag Model Overview"
![Rocket Drag Model Overview \label{rocket_drag_model_overview_label}][rocket_drag_model_overview]
```

Then, in your pandoc command, add the lof variable:

```
pandoc -s ... -V lof=lof
```

You can refer to the figure by the label you assigned to it

This comment refers to Figure `\ref{rocket_drag_model_overview_label}`

## Tables

To automatically number tables and add captions, add the *capt-of* package to your preamble

```
\usepackage{capt-of}
```

Then, in your pandoc command, add the lot variable:

```
pandoc -s ... -V lot=lot
```

## References

- [1] Simon Box Christopher M. Bishop and H. Hunt, “Stochastic six-degree-of-freedom flight simulator for passively controlled high-power rockets,” *J. Aerosp. Eng.*, 10.1061/(ASCE)AS.1943-5525.0000051, 31-45.
- [2] G. J. C. Mandell Gordon K. and W. P. Bengen., *Topics in advanced model rocketry*. Cambridge, Mass: MIT Press, 1973.
- [3] Simon Box Christopher M. Bishop and H. Hunt, “Estimating the dynamic and aerodynamic parameters of passively controlled high power rockets for flight simulation,” February 2009 [Online]. Available: <http://cambridgerocket.sourceforge.net/AerodynamicCoefficients.pdf>
- [4] S. Niskanen, “OpenRocket technical documentation (development of an open source model rocket simulation software),” Master’s thesis.
- [5] NASA, “General thrust equation.” Online [Online]. Available: <http://www.grc.nasa.gov/WWW/k-12/airplane/thrsteq.html>
- [6] J. Barrowman, “Calculating the center of pressure of a model rocket,” 1998.
- [7] J. Barrowman, “The theoretical prediction of the center of pressure,” *NARAM-8*, August 1966.
- [8] G. A. C. Sr., “The descriptive geometry of nose cones.” Online, 1996 [Online]. Available: [https://web.archive.org/web/20110411143013/http://www.if.sc.usp.br/~projetosulfos/artigos/NoseCone\\_EQN2.PDF](https://web.archive.org/web/20110411143013/http://www.if.sc.usp.br/~projetosulfos/artigos/NoseCone_EQN2.PDF)
- [9] R. Galejs, “What barrowman left out,” 1999.
- [10] M. Cavcar, “Viscosity as a function of temperature.” Online, September-2006 [Online]. Available: <http://aerojet.engr.ucdavis.edu/fluenthelp/html/ug/node337.htm>
- [11] F. Inc., “Viscosity as a function of temperature.” Online, September-2009 [Online]. Available: <http://aerojet.engr.ucdavis.edu/fluenthelp/html/ug/node337.htm>
- [12] H. Munson Okiishi, *Munson fundamentals of fluid mechanics*. Cambridge, Mass: John Wiley; Sons Inc, 2013.
- [13] D. G. M. Gregorek, “Aerodynamic drag of model rockets,” *ESTES INDUSTRIES INC.*, 1970.
- [14] U. of Wyoming, “Atmospheric soundings.” Online [Online]. Available: <http://weather.uwyo.edu/upperair/sounding.html>
- [15] F. IO, “Forcastet iO time machine.” Online [Online]. Available: <http://forecast.io/>