

OpenRocket technical documentation

For OpenRocket version 1.0.0

2010-04-06

Sampo Niskanen

Based on the Master's thesis [1]

Development of an Open Source model rocket simulation software

Thesis or technical documentation?

The OpenRocket simulation software was originally developed as the Master's thesis project of Sampo Niskanen, including its written part "*Development of an Open Source model rocket simulation software*" [1]. The thesis is used as the basis of this technical documentation, which is updated to account for later development in the software. This document often still refers to itself as a thesis, as no systematic updating of this fact has yet been performed.

While the original thesis is available online under a Creative Commons no-derivatives license, this document is available under a freer share-alike license.

Version history

2010-04-06 Initial revision. Updates the roll angle effect on three- and four-fin configurations in Section 3.2.2. (OpenRocket software 1.0.0)

“No. Coal mining may be your life, but it’s not mine. I’m never going down there again. I wanna go into space.”

Amateur rocketeer Homer Hickam, Jr. in the movie *October Sky* (1999), based on a true story.

Hickam later became an engineer at NASA, working in spacecraft design and crew training.

Contents

1	Introduction	1
1.1	Objectives of the thesis	3
2	Basics of model rocket flight	5
2.1	Model rocket flight	6
2.2	Rocket motor classification	7
2.3	Clustering and staging	10
2.4	Stability of a rocket	11
3	Aerodynamic properties of model rockets	14
3.1	General aerodynamical properties	15
3.1.1	Aerodynamic force coefficients	16
3.1.2	Velocity regions	18
3.1.3	Flow and geometry parameters	19
3.1.4	Coordinate systems	20
3.2	Normal forces and pitching moments	21

3.2.1	Axially symmetric body components	21
3.2.2	Planar fins	25
3.2.3	Pitch damping moment	35
3.3	Roll dynamics	37
3.3.1	Roll forcing coefficient	38
3.3.2	Roll damping coefficient	38
3.3.3	Equilibrium roll frequency	40
3.4	Drag forces	41
3.4.1	Laminar and turbulent boundary layers	41
3.4.2	Skin friction drag	43
3.4.3	Body pressure drag	46
3.4.4	Fin pressure drag	49
3.4.5	Base drag	50
3.4.6	Parasitic drag	51
3.4.7	Axial drag coefficient	52
4	Flight simulation	54
4.1	Atmospheric properties	54
4.1.1	Atmospheric model	54
4.1.2	Wind modeling	56
4.2	Modeling rocket flight	60

<i>CONTENTS</i>	vi
4.2.1 Coordinates and orientation	61
4.2.2 Quaternions	62
4.2.3 Mass and moment of inertia calculations	64
4.2.4 Flight simulation	65
4.2.5 Recovery simulation	67
4.2.6 Simulation events	68
5 The OpenRocket simulation software	70
5.1 Architectural design	71
5.1.1 Rocket components	71
5.1.2 Aerodynamic calculators and simulators	74
5.1.3 Simulation listeners	75
5.1.4 Warnings	76
5.1.5 File format	76
5.2 User interface design	76
5.3 Future enhancements	78
6 Comparison with experimental data	82
6.1 Comparison with a small model rocket	83
6.2 Comparison with a hybrid rocket	87
6.3 Comparison with a rolling rocket	87
6.4 Comparison with wind tunnel data	89

<i>CONTENTS</i>	vii
7 Conclusion	93
A Nose cone and transition geometries	100
A.1 Conical	100
A.2 Ogival	100
A.3 Elliptical	101
A.4 Parabolic series	102
A.5 Power series	102
A.6 Haack series	102
A.7 Transitions	103
B Transonic wave drag of nose cones	106
B.1 Blunt cylinder	106
B.2 Conical nose cone	107
B.3 Ellipsoidal, power, parabolic and Haack series nose cones . . .	108
B.4 Ogive nose cones	109
B.5 Summary of nose cone drag calculation	109
C Streamer drag coefficient estimation	111

List of symbols and abbreviations

Symbols

A	Area
A_{fin}	Area of one fin
A_{plan}	Planform area
A_{ref}	Reference area
A_{wet}	Wetted area
\mathcal{R}	Aspect ratio of a fin, $2s^2/A_{\text{fin}}$
c	Speed of sound
\bar{c}	Mean aerodynamic chord length of a fin
$c(y)$	Chord length of a fin at spanwise position y
C_A	Axial drag force coefficient
C_D	Drag force coefficient
C_f	Skin friction drag coefficient
C_l	Roll moment coefficient
C_{ld}	Roll damping moment coefficient
C_{lf}	Roll forcing moment coefficient
C_m	Pitch moment coefficient
C_{m_α}	Pitch moment coefficient derivative, $\frac{\partial C_m}{\partial \alpha}$
C_N	Normal force coefficient
C_{N_α}	Normal force coefficient derivative, $\frac{\partial C_N}{\partial \alpha}$
d	Reference length, the rocket diameter
D	Drag force
f_B	Rocket fineness ratio, L/d
L	The rocket length
m	Pitch moment
M	Mach number
N	Normal force; Number of fins
p	Air pressure
$r(x)$	Body or component radius at position x
R	Reynolds number
s	Spanwise length of one fin
T	Air temperature
V	Volume
v_0	Free-stream velocity
x, X	Position along the rocket centerline
y	Spanwise position

α	Angle of attack
β	$\sqrt{ M^2 - 1 }$
γ	Specific heat ratio, for air $\gamma = 1.4$
Γ_c	Fin midchord sweep angle
δ	Fin cant angle
η	Airflow inclination angle over a fin
θ	Roll angle
Λ	Dihedral angle between a fin and the direction of airflow
ν	Kinematic viscosity of air
ξ	Distance from rotation axis
ρ	Density of air
ω	Angular velocity

Abbreviations

CFD	Computational fluid dynamics
CG	Center of gravity
CP	Center of pressure
LE	Leading edge
MAC	Mean aerodynamic chord
RK4	Runge-Kutta 4 integration method
UI	User interface

Chapter 1

Introduction

Model rocketry is a sport that involves designing, constructing and launching self-made rockets. Model rockets vary greatly in size, shape, weight and construction from detailed scale models of professional rockets to lightweight and highly finished competition models. The sport is relatively popular and is often cited as a source of inspiration for children to become engineers and scientists.

The hobby started as amateur rocketry in the 1950's when hobbyists wanted to experiment their skill with building rockets. Designing, building and firing self-made *motors* was, however, extremely dangerous, and the American Rocket Society (now the American Institute of Aeronautics and Astronautics, AIAA) has estimated that about one in seven amateur rocketeers during the time were injured in their hobby. This changed in 1958 when the first commercially-built model rocket motors became available. Having industrially-made, reasonably-priced and safe motors available removed the most dangerous aspect of amateur rocketry. This along with strict guidelines to the design and launching of model rockets formed the foundation for a safe and widespread hobby. [2, pp. 1–3]

Since then model rocketry has spread around the globe and among all age groups. Thousands of rockets ranging from 10 cm high miniatures to large models reaching altitudes in excess of 10 km are launched annually. Model rocket motors with thrusts from a few Newtons up to several kilo-Newtons are readily available. Since its forming in 1957, over 90 000 people have joined the National Association of Rocketry (NAR) in the U.S. alone.

In designing rockets, the *stability* of a rocket is of central priority. A stable rocket corrects its course if some outside force disturbs it slightly. A disturbance of an unstable rocket instead increases until the rocket starts spinning in the air erratically. As shall be discussed in Section 2.4, a rocket is deemed *statically stable* if its center of pressure (CP) is aft of its center of gravity (CG)¹. The center of gravity of a rocket can be easily calculated in advance or determined experimentally. The center of pressure, on the other hand, has been quite hard to determine either analytically or experimentally. In 1966 James and Judith Barrowman developed an analytical method for determining the CP of a slender-bodied rocket at subsonic speeds and presented their results as a research and development project at the 8th National Association of Rocketry Annual Meeting (NARAM-8) [3], and later as a part of James Barrowman’s Master’s thesis [4]. This method has become known as the *Barrowman method* of determining the CP of a rocket within the model rocketry community, and has a major role in determining the aerodynamic characteristics of model rockets.

Another important aerodynamic quantity of interest is the *aerodynamic drag* of a rocket. Drag is caused by the flow of air around the rocket and it can easily reduce the maximum altitude of a rocket by 50–80% of the otherwise theoretical maximum. Estimating the drag of a model rocket is a rather complex task, and the effects of different design choices are not always very evident to a hobbyist.

Knowing the fundamental aerodynamic properties of a rocket allows one to simulate its free flight. This involves numerically integrating the flight forces and determining the velocity, rotation and position of the rocket as a function of time. This is best performed by software designed for the purpose of model rocket design.

RockSim [5] is one such piece of software. It is a commercial, proprietary program that allows one to define the geometry and configuration of a model rocket, estimate its aerodynamic properties and simulate a launch with different rocket motors. It has become the *de facto* standard software for model rocket performance estimation. However, as a proprietary program, it is essentially a “black-box” solution. Someone wishing to study or validate the methods will not be able to do so. Similarly extending or customizing the

¹An alternative term would be *center of mass*, but in the context of model rocketry, we are interested in the effect of gravity on the rocket. Thus, the term center of gravity is widely used in model rocketry texts, and this convention will be followed in this thesis.

functionality or refining the calculations methods to fit ones needs is impossible. The software is also only available on select operating systems. Finally, the cost of the software may be prohibitive especially for younger hobbyists, voluntary organizations, clubs and schools.

Open Source software, on the other hand, has become an increasingly competitive alternative to proprietary software. Open Source allows free access to the source code of the programs and encourages users with the know-how to enhance the software and share their changes [6]. Success stories such as the Linux operating system, the OpenOffice.org office suite, the Firefox web browser and countless others have shown that Open Source software can often achieve and even exceed the quality of expensive proprietary software.

1.1 Objectives of the thesis

The objectives of this thesis work are to:

1. Develop and document relatively easy, yet reasonably accurate methods for the calculation of the fundamental aerodynamic properties of model rockets and their numerical simulation;
2. Test the methods developed and compare the results with other estimates and actual experimental data; and
3. Implement a cross-platform, Open Source model rocket design and simulation software that uses the aforementioned methods, is at the same time easy to use and yet versatile, and which is easily extensible and customizable for user requirements, new types of rocket components and new estimation methods.

The methods presented will largely follow the methods developed by Barrowman [3, 4], since these are already familiar to the rocketry community. Several extensions to the methods will be added to allow for more accurate calculation at larger angles of attack and for fin shapes not accounted for in the original paper. The emphasis will be on subsonic flight, but extensions will be made for reasonable estimation at transonic and low supersonic velocities.

The software developed as part of the thesis is the OpenRocket project [7]. It is an Open Source rocket development and simulation environment written totally in Java. The program structure has been designed to make full use of object oriented programming, allowing one to easily extend its features. The software also includes a framework for creating user-made *listener components* (discussed in Section 5.1.3) that can listen to and interact with the simulation while it is running. This allows a powerful and easy way of interacting with the simulation and allows simulating for example guidance systems.

One possible future enhancement that has also specifically been considered throughout the development is calculating the aerodynamic properties using computational fluid dynamics (CFD). CFD calculates the exact airflow in a discretized mesh around the rocket. This would allow for even more accurate calculation of the aerodynamic forces for odd-shaped rockets, for which the methods explained herein do not fully apply.

It is anticipated that the software will allow more hobbyists the possibility of simulating their rocket designs prior to building them and experimenting with different configuration, thus giving them a deeper understanding of the aerodynamics of rocket flight. It will also provide a more versatile educational tool since the simulation methods are open and everyone will be able to “look under the hood” and see how the software performs the calculations.

In Chapter 2 a brief overview of model rocketry and its different aspects will be given. Then in Chapter 3 methods for calculating the aerodynamic properties of a general model rocket will be presented. In Chapter 4 the aspects of simulating a rocket’s flight are considered. Chapter 5 then explains how the aerodynamic calculations and simulation are implemented in the OpenRocket software and presents some of its features. In Chapter 6 the results of the software simulation are compared with the performance of constructed and launched rockets. Chapter 7 then presents a summary of the achievements and identifies areas of further work.

Chapter 2

Basics of model rocket flight

As rockets and rocket motors come in a huge variety of shapes and sizes, different categories are defined for different levels of rocketry. *Model rocketry* itself is governed by the NAR Model Rocket Safety Code [8] in the U.S. and other similar regulations in other countries. The safety code requires that the model rockets be constructed only of light-weight materials without any metal structural parts, and have a maximum lift-off weight of 1.5 kg. They may only use pre-manufactured motors of classes A–G (see Section 2.2 for the classification).

High power rocketry (HPR) is basically scaled up model rocketry. There are no weight restrictions, and they can use pre-manufactured solid or hybrid rocket motors in the range of H–O. The combined total impulse of all motors may not exceed 81920 Ns.

Experimental or *amateur rocketry* includes any rocketry activities beyond model and high power rocketry. This may include for example using motor combinations that exceed the limits placed by high power rocketry, building self-made motors or utilizing liquid fueled motors. Finally there is *professional rocketry* which is conducted for profit, usually by governments or large corporations.

Even though rockets come in many different sizes, the same principles apply to all of them. In this thesis the emphasis will be on model rocketry, but the results are just as valid for larger rockets as long as the assumptions of for example the speed range remain valid. In this chapter the basics of model

rocketry and differences to high power rocketry are explained.

2.1 Model rocket flight

A typical flight of a model rocket can be characterized by the four phases depicted in Figure 2.1:

1. Launch: The model rocket is launched from a vertical launch guide.
2. Powered flight: The motor accelerates the rocket during the powered flight period.
3. Coasting flight: The rocket coasts freely until approximately at its apogee.
4. Recovery: The recovery device opens and the rocket descends slowly to the ground.

Model rockets are launched from a vertical launch guide that keeps the rocket in an upright position until it has sufficient velocity for the fins to aerodynamically stabilize the flight. The NAR safety code forbids launching a model rocket at an angle greater than 30° from vertical. A typical launch guide for small rockets is a metal rod about 3-5 mm in diameter, and the launch lug is a short piece of plastic tube glued to the body tube. Especially in larger rockets this may be replaced by two extruding bolts, the ends of which slide along a metal rail. Use of a launch lug can be avoided by a tower launcher, which has 3-4 metal bars around the rocket that hold it in an upright position.

After clearing the launch guide, the rocket is in free, powered flight. During this phase the motor accelerates the rocket while it is aerodynamically stabilized to keep its vertical orientation. When the propellant has been used, the rocket is typically at its maximum velocity. It then coasts freely for a short period while the motor produces smoke to help follow the rocket, but provides no additional thrust. Finally, at approximately the point of apogee, a small pyrotechnical ejection charge is fired upwards from the motor which pressurizes the model rocket and opens the recovery device.

High-power rocket motors usually have no ejection charges incorporated in them. Instead, the rocket carries a small flight computer that measures the

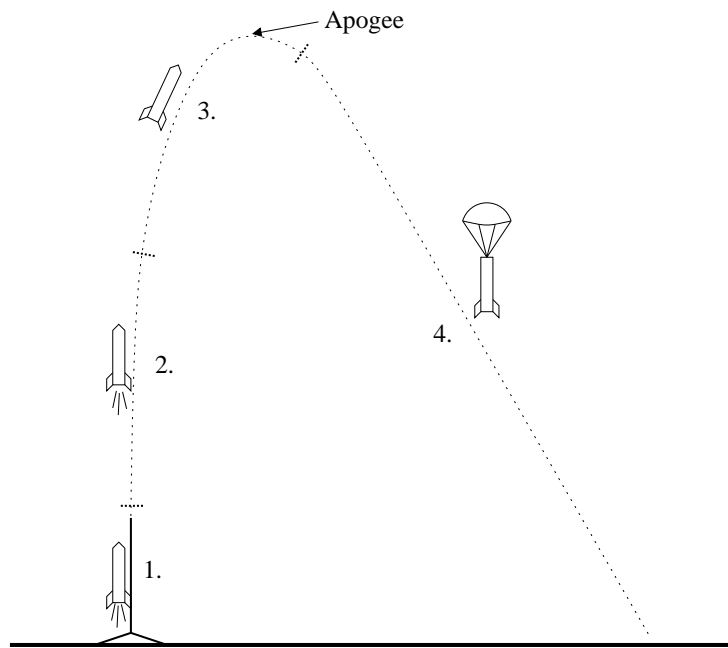


Figure 2.1: The basic phases of a typical model rocket flight: 1. Launch, 2. Powered flight, 3. Coasting and 4. Recovery.

acceleration of the rocket or the outside pressure change to detect the point of apogee and to open the recovery device. Frequently only a small drogue parachute is opened at apogee, and the main parachute is opened at some pre-defined lower altitude around 100–300 meters.

The typical recovery device of a model rocket is either a parachute or a *streamer*. The parachutes are usually a simple planar circle of plastic or fabric with 4–10 shroud lines attached. A streamer is a strip of plastic or fabric connected to the rocket, intended to flutter in the air and thus slow down the descent of the rocket. Especially small rockets often use streamers as their recovery device, since even light wind can cause a light-weight rocket with a parachute to drift a significant distance.

2.2 Rocket motor classification

The motors used in model and high power rocketry are categorized based on their total impulse. A class ‘A’ motor may have a total impulse in the range

Table 2.1: Total impulse ranges for motor classes $\frac{1}{4}$ A–O.

$\frac{1}{4}$ A	0.0–0.625 Ns	E	20.01–40.0 Ns	K	1280.01–2560 Ns
$\frac{1}{2}$ A	0.626–1.25 Ns	F	40.01–80.0 Ns	L	2560.01–5120 Ns
A	1.26–2.50 Ns	G	80.01–160 Ns	M	5120.01–10240 Ns
B	2.51–5.00 Ns	H	160.01–320 Ns	N	10240.01–20480 Ns
C	5.01–10.0 Ns	I	320.01–640 Ns	O	20480.01–40960 Ns
D	10.01–20.0 Ns	J	640.01–1280 Ns		

of 1.26–2.50 Ns. Every consecutive class doubles the allowed total impulse of the motor. Thus, a B-motor can have an impulse in the range 2.51–5.00 Ns and a C-motor in the range 5.01–10.0 Ns. There are also classes $\frac{1}{2}$ A and $\frac{1}{4}$ A which have impulse ranges half and one quarter of those of an A-motor, respectively. Commercial rocket motors are available up to class O with a total impulse of 30 000 Ns [9]. Table 2.1 lists the impulse ranges for model and high-power rocket motors.

Another important parameter of a rocket motor is the thrust given by the motor. This defines the mass that may be lifted by the motor and the acceleration achieved. Small model rocket motors typically have an average thrust of about 3–10 N, while high-power rocket motors can have thrusts in excess of 5 000 N.

The third parameter used to classify a model rocket motor is the length of the delay between the motor burnout and the ignition of the ejection charge. Since the maximum velocity of different rockets using the same type of motor can be vastly different, also the length of the coasting phase varies. Therefore motors with otherwise the same specifications are often manufactured with several different delay lengths. These delay lengths do not apply to high-power rocket motors, since they do not have ejections charges incorporated in them.

Model rocket motors are given a classification code based on these three parameters, for example “D7-3”. The letter specifies the total impulse range of the motor, while the first number specifies the average thrust in Newtons and the second number the delay of the ejection charge in seconds. The delay number can also be replaced by ‘P’, which stands for *plugged*, *i.e.* the motor does not have an ejection charge. Some manufacturers may also use an additional letter at the end of the classification code specifying the propellant

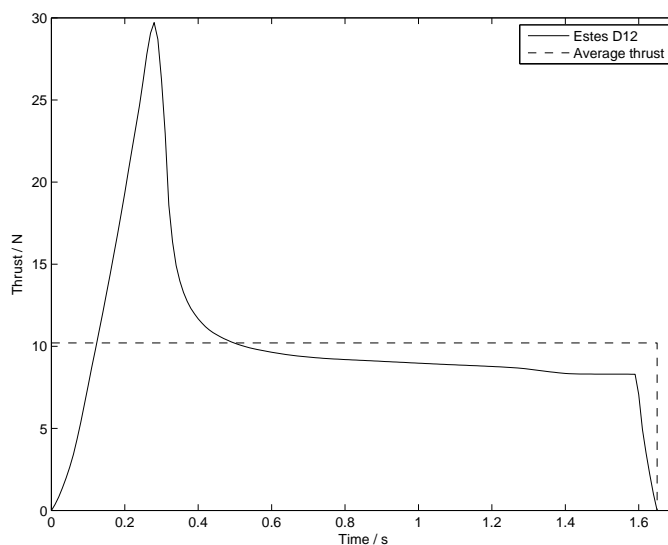


Figure 2.2: A typical thrust curve of an Estes D12-3 rocket motor and its average thrust. [11]

type used in the motor.

Even motors with the same classification code may have slight variations to them. First, the classification only specifies the impulse range of the motor, not the exact impulse. In principle, a D-motor in the lower end of the range might have a total impulse only 1 Ns larger than a C-motor in the upper end of its range. Second, the code only specifies the average thrust of the motor. The thrust rarely is constant, but varies with time. Figure 2.2 shows the typical thrust curve of a small black powder rocket motor. The motors typically have a short thrust peak at ignition that gives the rocket an initial acceleration boost before stabilizing to a thrust level a little below the average thrust. Statically measured thrust curves of most commercial rocket motors are readily available on the Internet [10].

Also the propellant type may affect the characteristics of the motor. Most model rocket motors are made up of a solid, pyrotechnical propellant—typically black powder—that is cast into a suitable shape and ignited on launch. Since the propellant burns on its surface, different thrust curves can be achieved by different mold shapes.

A significantly different motor type, *hybrid motors*, were commercially introduced in 1995. These motors typically include the propellant and oxidizer

in different states, typically a composite plastic as the fuel and a separate tank of liquid nitrous oxide (N_2O) as the oxidizer. The plastic on its own does not burn very well, but provides ample thrust when the nitrous oxide is fed through its core. The nitrous oxide tank is self-pressurized by its natural vapor pressure. However, since temperature greatly affects the vapor pressure of nitrous oxide, the thrust of a hybrid motor is also diminished if the oxidizer is cold. On the other hand, the motor will burn longer in this case, and since nitrous oxide is denser when cold, the motor may even yield a greater total impulse.

The significance of this effect was observed when analyzing the video footage of the launch of the first Finnish hybrid rocket, “Haisunäätä” [12]. The average thrust during the first 0.5 seconds was determined to be only about 70 N, whereas the static tests suggest the thrust should have been over 200 N. Instead, the motor burned for over 10 seconds, while the normal thrust curves indicate a burning time of 5–6 seconds. This shows that the temperature of the hybrid motor oxidizer can have a dramatic effect on the thrust given by the motor, and the static test curve should be assumed to be valid only in similar operating conditions as during the test.

One further non-pyrotechnical rocket type is *water rockets*. These are especially popular first rockets, as they require no special permits and are easy to construct. The water rocket includes a bottle or other chamber that has water and pressurized air inside it. On launch the pressure forces the water out of a nozzle, providing thrust to the rocket. While simulating water rockets is beyond the scope of this thesis, it is the aim that methods for modeling water rockets can be added to the produced software in the future.

2.3 Clustering and staging

Two common methods used to achieve greater altitudes with model rockets are *clustering* and *staging*. A cluster has two or more rocket motors burning concurrently, while staging uses motors that burn consecutively. The motor configuration of a cluster and staged rocket is depicted in Figure 2.3.

When a cluster is launched, the total thrust is the sum of the thrust curves of the separate motors. This allows greater acceleration and a greater liftoff weight. Staging is usually performed by using zero-delay motors, that ignite

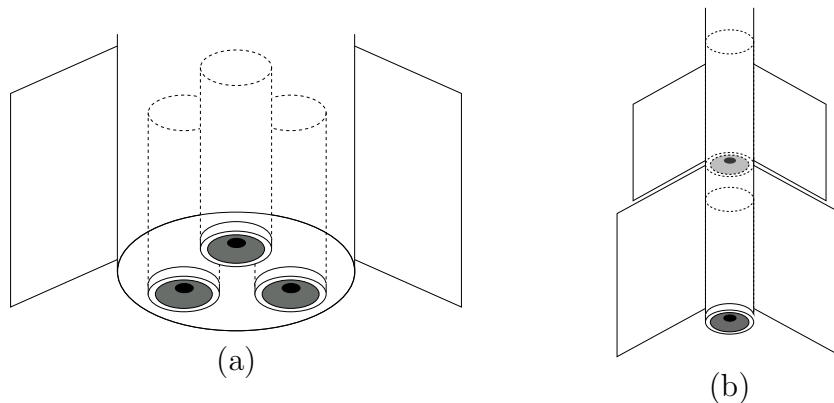


Figure 2.3: The motor configuration for (a) a cluster rocket and (b) a two-staged rocket.

the ejection charge immediately at burnout. The ejection charge fires towards the upper stage motor and ignites the next motor. High power motors with no ejection charges can be clustered by using an onboard accelerometer or timer that ignites the subsequent stages. Staging provides a longer duration of powered flight, thus increasing the altitude.

Clustering provides a greater acceleration at launch, but staging typically provides greater altitude than a cluster with similar motors. This is because a clustered rocket accelerates quickly to a greater speed thus also increasing the aerodynamic drag. A staged rocket has a smaller thrust for a longer period of time, which reduces the overall effect of drag during the flight.

2.4 Stability of a rocket

When designing a new rocket, its stability is of paramount importance. A small gust of wind or some other disturbance may cause the rocket to tilt slightly from its current orientation. When this occurs, the rocket centerline is no longer parallel to the velocity of the rocket. This condition is called flying at an *angle of attack* α , where α is the angle between the rocket centerline and the velocity vector.

When a stable rocket flies at an angle of attack, its fins produce a moment to correct the rocket's flight. The corrective moment is produced by the aerodynamic forces perpendicular to the axis of the rocket. Each component

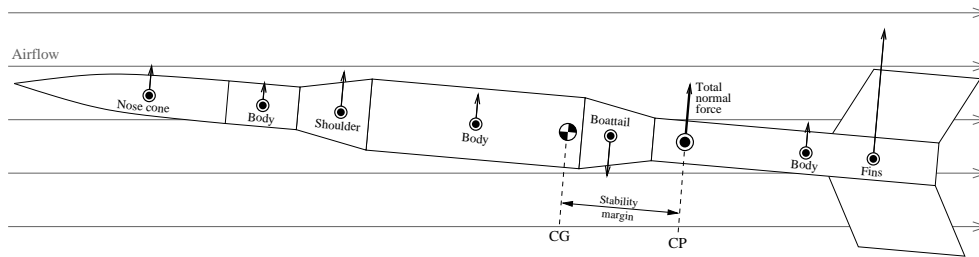


Figure 2.4: Normal forces produced by the rocket components.

of the rocket can be seen as producing a separate normal force component originating from the component's CP, as depicted in Figure 2.4.

The effect of the separate normal forces can be combined into a single force, the magnitude of which is the sum of the separate forces and which effects the same moment as the separate forces. The point on which the total force acts is defined as the center of pressure or the rocket. As can be seen from Figure 2.4, the moment produced attempts to correct the rocket's flight only if the CP is located aft of the CG. If this condition holds, the rocket is said to be *statically stable*. A statically stable rocket always produces a corrective moment when flying at a small angle of attack.

The argument for static stability above may fail in two conditions: First, the normal forces might cancel each other out exactly, in which case a moment would be produced but with zero total force. Second, the normal force at the CP might be in the wrong direction (downward in the figure), yielding an uncorrective moment. However, we shall see that the only component to produce a downward force is a boattail, and the force is equivalent to the corresponding broadening of the body. Therefore the total force acting on the rocket cannot be zero nor in a direction to produce an uncorrective moment when aft of the CG.

The *stability margin* of a rocket is defined as the distance between the CP and CG, measured in *calibers*, where one caliber is the maximum body diameter of the rocket. A rule of thumb among model rocketeers is that the CP should be approximately 1–2 calibers aft of the CG. However, the CP of a rocket typically moves upwards as the angle of attack increases. In some cases, a 1–2 caliber stability margin may totally disappear at an angle of attack of only a few degrees. As side wind is the primary cause of angles of attack, this effect is called *wind caused instability* [13].

Another stability issue concerning rocketeers is the *dynamic stability* of a rocket. A rocket that is statically stable may still be poor at returning the rocket to the original orientation quickly enough. Model rockets may encounter several types of dynamic instability depending on their shape, size and mass [2, pp. 140–141]:

1. *Too little oscillation damping.* In short, light-weight rockets the corrective moment may significantly over-correct the perturbation, requiring a corrective moment in the opposite direction. This may lead to continuous oscillation during the flight.
2. *Too small corrective moment.* This is the case of over-damped oscillation, where the corrective moment is too small compared to the moment of inertia of the rocket. Before the rocket has been able to correct its orientation, the thrust of the motors may have already significantly affected the direction of flight.
3. *Roll-pitch coupling.* If the model has a natural roll frequency (caused *e.g.* by canting the fins) close to the oscillation frequency of the rocket, roll-pitch resonance may occur and cause the model to go unstable.

By definition, dynamic stability issues are such that they occur over time during the flight of the rocket. A full flight simulation that takes into account all corrective moments automatically also simulates the possible dynamic stability problems. Therefore the dynamic stability of rockets will not be further considered in this thesis. For an analytical consideration of the problem, refer to [14].

Chapter 3

Aerodynamic properties of model rockets

A model rocket encounters three basic forces during its flight: thrust from the motors, gravity, and aerodynamical forces. Thrust is generated by the motors by exhausting high-velocity gases in the opposite direction. The thrust of a motor is directly proportional to the velocity of the escaping gas and the mass per time unit that is exhausted. The thrust of commercial model rocket motors as a function of time have been measured in static motor tests and are readily available online [10]. Normally the thrust of a rocket motor is aligned on the center axis of the rocket, so that it produces no angular moment to the rocket.

Every component of the rocket is also affected by gravitational force. When the forces and moments generated are summed up, the gravitational force can be seen as a single force originating from the *center of gravity* (CG). A homogeneous gravitational field does not generate any angular moment on a body relative to the CG. Calculating the gravitational force is therefore a simple matter of determining the total mass and CG of the rocket.

Aerodynamic forces, on the other hand, produce both net forces and angular moments. To determine the effect of the aerodynamic forces on the rocket, the total force and moment must be calculated relative to some reference point. In this chapter, a method for determining these forces and moments will be presented.

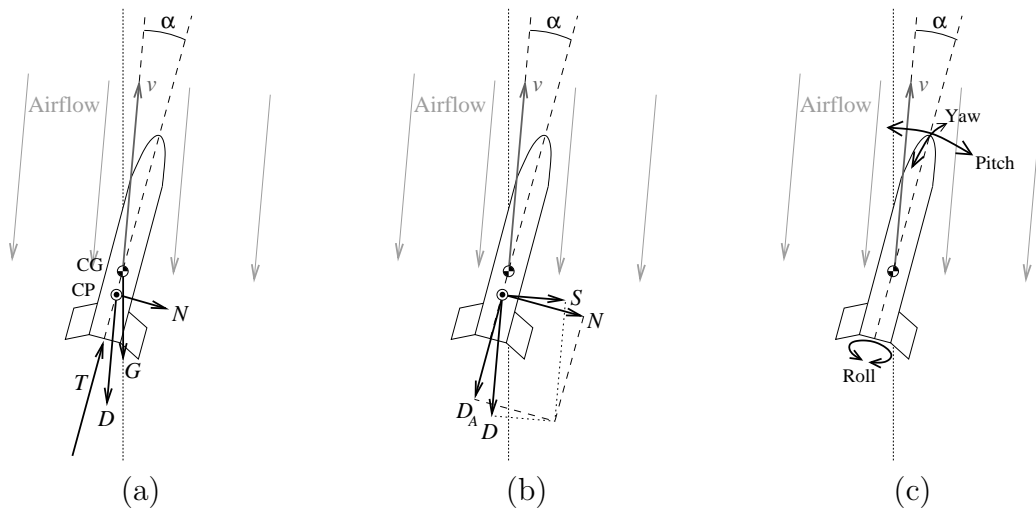


Figure 3.1: (a) Forces acting on a rocket in free flight: gravity G , motor thrust T , drag D and normal force N . (b) Perpendicular component pairs of the total aerodynamical force: normal force N and axial drag D_A ; side force S and drag D . (c) The pitch, yaw and roll directions of a model rocket.

3.1 General aerodynamical properties

The aerodynamic forces acting on a rocket are usually split into components for further examination. The two most important aerodynamic force components of interest in a typical model rocket are the *normal force* and *drag*. The aerodynamical normal force is the force component that generates the corrective moment around the CG and provides stabilization of the rocket. The drag of a rocket is defined as the force component parallel to the velocity of the rocket. This is the aerodynamical force that opposes the movement of the rocket through air.

Figure 3.1(a) shows the thrust, gravity, normal force and drag of a rocket in free flight. It should be noted that if the rocket is flying at an angle of attack $\alpha > 0$, then the normal force and drag are not perpendicular. In order to have independent force components, it is necessary to define component pairs that are always perpendicular to one another. Two such pairs are the normal force and axial drag, or side force and drag, shown in Figure 3.1(b). The two pairs coincide if the angle of attack is zero. The component pair that will be used as a basis for the flight simulations is the normal force and axial drag.

The three moments around the different axis are called the *pitch*, *yaw* and *roll moments*, as depicted in Figure 3.1(c). Since a typical rocket has no “natural” roll angle of flight as an aircraft does, we may choose the pitch angle to be in the same plane as the angle of attack, *i.e.* the plane defined by the velocity vector and the centerline of the rocket. Thus, the normal force generates the pitching moment and no other moments.

3.1.1 Aerodynamic force coefficients

When studying rocket configurations, the absolute force values are often difficult to interpret, since many factors affect them. In order to get a value better suited for comparison, the forces are normalized by the current dynamic pressure $q = \frac{1}{2}\rho v_0^2$ and some characteristic area A_{ref} to get a non-dimensional force coefficient. Similarly, the moments are normalized by the dynamic pressure, characteristic area and characteristic length d . Thus, the normal force coefficient corresponding to the normal force N is defined as

$$C_N = \frac{N}{\frac{1}{2}\rho v_0^2 A_{\text{ref}}} \quad (3.1)$$

and the pitch moment coefficient for a pitch moment m as

$$C_m = \frac{m}{\frac{1}{2}\rho v_0^2 A_{\text{ref}} d}. \quad (3.2)$$

A typical choice of reference area is the base of the rocket’s nose cone and the reference length is its diameter.

The pitch moment is always calculated around some reference point, while the normal force stays constant regardless of the point of origin. If the moment coefficient C_m is known for some reference point, the moment coefficient at another point C'_m can be calculated from

$$C'_m d = C_m d - C_N \Delta x \quad (3.3)$$

where Δx is the distance along the rocket centerline. Therefore it is sufficient to calculate the moment coefficient only at some constant point along the rocket body. In this thesis the reference point is chosen to be the tip of the nose cone.

The *center of pressure* (CP) is defined as the position from which the total normal force alone produces the current pitching moment. Therefore the total

normal force produces no moment around the CP itself, and an equation for the location of the CP can be obtained from (3.3) by selecting setting $C'_m = 0$:

$$X = \frac{C_m}{C_N} d \quad (3.4)$$

Here X is the position of the CP along the rocket centerline from the nose cone tip. This equation is valid when $\alpha > 0$. As α approaches zero, both C_m and C_N approach zero. The CP is then obtained as a continuous extension using l'Hôpital's rule

$$X = \frac{\frac{\partial C_m}{\partial \alpha}}{\frac{\partial C_N}{\partial \alpha}} d \Big|_{\alpha=0} = \frac{C_{m_\alpha}}{C_{N_\alpha}} d \quad (3.5)$$

where the normal force coefficient and pitch moment coefficient derivatives have been defined as

$$C_{N_\alpha} = \frac{\partial C_N}{\partial \alpha} \Big|_{\alpha=0} \quad \text{and} \quad C_{m_\alpha} = \frac{\partial C_m}{\partial \alpha} \Big|_{\alpha=0}. \quad (3.6)$$

At very small angles of attack we may approximate C_N and C_m to be linear with α , so to a first approximation

$$C_N \approx C_{N_\alpha} \alpha \quad \text{and} \quad C_m \approx C_{m_\alpha} \alpha. \quad (3.7)$$

The Barrowman method uses the coefficient derivatives to determine the CP position using equation (3.5). However, there are some significant nonlinearities in the variation of C_N as a function of α . These will be accounted for by holding the approximation of equation (3.7) exact and letting C_{N_α} and C_{m_α} be a function of α . Therefore, for the purposes of this thesis we define

$$C_{N_\alpha} = \frac{C_N}{\alpha} \quad \text{and} \quad C_{m_\alpha} = \frac{C_m}{\alpha} \quad (3.8)$$

for $\alpha > 0$ and by equation (3.6) for $\alpha = 0$. These definitions are compatible, since equation (3.8) simplifies to the partial derivative (3.6) at the limit $\alpha \rightarrow 0$. This definition also allows us to stay true to Barrowman's original method which is familiar to many rocketeers.

Similar to the normal force coefficient, the drag coefficient is defined as

$$C_D = \frac{D}{\frac{1}{2} \rho v_0^2 A_{\text{ref}}}. \quad (3.9)$$

Since the size of the rocket has been factored out, the drag coefficient at zero angle of attack C_{D0} allows a straightforward method of comparing the effect of different rocket shapes on drag. However, this coefficient is not constant and will vary with *e.g.* the speed of the rocket and its angle of attack.

If each of the fins of a rocket are canted at some angle $\delta > 0$ with respect to the rocket centerline, the fins will produce a roll moment on the rocket. Contrary to the normal force and pitching moment, canting the fins will produce a non-zero rolling moment but no corresponding net force. Therefore the only quantity computed is the roll moment coefficient, defined by

$$C_l = \frac{l}{\frac{1}{2}\rho v_0^2 A_{\text{ref}} d} \quad (3.10)$$

where l is the roll moment.

It shall be shown later that rockets with axially-symmetrical fin configurations experience no forces that would produce net yawing moments. However, a single fin may produce all six types of forces and moments. The equations for the forces and moments of a single fin will not be explicitly written out, and they can be computed from the geometry in question.

3.1.2 Velocity regions

Most of the aerodynamic properties of rockets vary with the velocity of the rocket. The important parameter is the *Mach number*, which is the free-stream velocity of the rocket divided by the local speed of sound

$$M = \frac{v_0}{c}. \quad (3.11)$$

The velocity range encountered by rockets is divided into regions with different impacts on the aerodynamical properties, listed in Table 3.1.

In *subsonic flight* all of the airflow around the rocket occurs below the speed of sound. This is the case for approximately $M < 0.8$. At very low Mach numbers air can be effectively treated as an incompressible fluid, but already above $M \approx 0.3$ some compressibility issues may have to be considered.

In *transonic flight* some of the air flowing around the rocket accelerates above the speed of sound, while at other places it remains subsonic. Some local

Table 3.1: Velocity regions of rocket flight

Region	Mach number (M)
Subsonic	0 – 0.8
Transonic	0.8 – 1.2
Supersonic	1.2 – ~ 5
Hypersonic	~ 5 –

shock waves are generated and hard-to-predict interference effects may occur. The drag of a rocket has a sharp increase in the transonic region, making it hard to pass into the supersonic region. Transonic flight occurs at Mach numbers of approximately 0.8–1.2.

In *supersonic flight* all of the airflow is faster than the speed of sound (with the exception of *e.g.* the nose cone tip). A shock wave is generated by the nose cone and fins. In supersonic flight the drag reduces from that of transonic flight, but is generally greater than that of subsonic flight. Above approximately Mach 5 new phenomena begin to emerge that are not encountered at lower supersonic speeds. This region is called *hypersonic flight*.

Methods for predicting the aerodynamic properties of subsonic flight and some extensions to supersonic flight will be presented. Since the analytical prediction of aerodynamic properties in the transonic region is quite difficult, this region will be accounted for by using some suitable interpolation function that corresponds reasonably to actual measurements. Hypersonic flight will not be considered, since practically no model or high power rockets ever achieve such speeds.

3.1.3 Flow and geometry parameters

There exist many different parameters that characterize aspects of flow or a rocket's geometry. One of the most important flow parameters is the *Reynolds number* R . It is a dimensionless quantity that characterizes the ratio of inertial forces and viscous forces of flow. Many aerodynamic properties depend on the Reynolds number, defined as

$$R = \frac{v_0 L}{\nu}. \quad (3.12)$$

Here v_0 is the free-stream velocity of the rocket, L is a characteristic length and ν is the kinematic viscosity of air. It is notable that the Reynolds number is dependent on a characteristic length of the object in question. In most cases, the length used is the length of the rocket. A typical 30 cm sport model flying at 50 m/s has a corresponding Reynolds number of approximately 1 000 000.

Another term that is frequently encountered in aerodynamical equations has been defined its own parameter β , which characterizes the flow speed both in subsonic and supersonic flow:

$$\beta = \sqrt{|M^2 - 1|} = \begin{cases} \sqrt{1 - M^2}, & \text{if } M < 1 \\ \sqrt{M^2 - 1}, & \text{if } M > 1 \end{cases} \quad (3.13)$$

As the flow speed approaches the transonic region β approaches zero. This term appears for example in the *Prandtl factor* P which corrects subsonic force coefficients for compressible flow:

$$P = \frac{1}{\beta} = \frac{1}{\sqrt{1 - M^2}} \quad (3.14)$$

It is also often useful to define parameters characterizing general properties of a rocket. One such parameter is the *caliber*, defined as the maximum body diameter. The caliber is often used to indicate relative distances on the body of a rocket, such as the stability margin. Another common parameter characterizes the “slenderness” of a rocket. It is the *fineness ratio* of a rocket f_B , defined as the length of the rocket body divided by the maximum body diameter. Typical model rockets have a fineness ratio in the range of 10–20, but extreme models may have a fineness ratio as low as 5 or as large as 50.

3.1.4 Coordinate systems

During calculation of the aerodynamic properties a coordinate system fixed to the rocket will be used. The origin of the coordinates is at the nose cone tip with the positive x -axis directed along the rocket centerline. This convention is also followed internally in the produced software. In the following sections the position of the y - and z -axes are arbitrary; the parameter y is used as a general spanwise coordinate when discussing the fins. During simulation, however, the y - and z -axes are fixed in relation to the rocket, and do not necessarily align with the plane of the pitching moments.

3.2 Normal forces and pitching moments

Barrowman's method [4] for determining the total normal force coefficient derivative C_{N_α} , the pitch moment coefficient derivative C_{m_α} and the CP location at subsonic speeds first splits the rocket into simple separate components, then calculates the CP location and C_{N_α} for each component separately and then combines these to get the desired coefficients and CP location. The general assumptions made by the derivation are:

1. The angle of attack is very close to zero.
2. The flow around the body is steady and non-rotational.
3. The rocket is a rigid body.
4. The nose tip is a sharp point.
5. The fins are flat plates.
6. The rocket body is axially symmetric.

The components that will be discussed are nose cones, cylindrical body tube sections, shoulders, boattails and fins, in an arbitrary order. The interference effect between the body and fins will be taken into account by a separate correction term. Extensions to account for body lift and arbitrary fin shapes will also be derived.

3.2.1 Axially symmetric body components

The body of the rocket is assumed to be an axially symmetric body of rotation. The entire body could be considered to be a single component, but in practice it is divided into nose cones, shoulders, boattails and cylindrical body tube sections. The geometry of typical nose cones, shoulders and boattails are described in Appendix A.

The method presented by Barrowman for calculating the normal force and pitch moment coefficients at supersonic speeds is based on a second-order shock expansion method. However, this assumes that the body of the rocket

is very streamlined, and it cannot handle areas with a slope larger than $\sim 30^\circ$. Since the software allows basically any body shape, applying this method would be difficult.

Since the emphasis is on subsonic flow, for the purposes of this thesis the normal force and pitching moments produced by the body are assumed to be equal at subsonic and supersonic speeds. The assumption is that the CP location is primarily affected by the fins. The effect of supersonic flight on the drag of the body will be accounted for in Section 3.4.

C_{N_α} of body components at subsonic speeds

The normal force for an axially symmetric body at position x in subsonic flow is given by

$$N(x) = \rho v_0 \frac{\partial}{\partial x} [A(x)w(x)] \quad (3.15)$$

where $A(x)$ is the cross-sectional area of the body, and the $w(x)$ is the local downwash, given as a function of the angle of attack as

$$w(x) = v_0 \sin \alpha. \quad (3.16)$$

For angles of attack very close to zero $\sin \alpha \approx \alpha$, but contrary to the original derivation, we shall not make this simplification. From the definition of the normal force coefficient (3.1) and equation (3.15) we obtain

$$C_N(x) = \frac{N(x)}{\frac{1}{2}\rho v_0^2 A_{\text{ref}}} = \frac{2 \sin \alpha}{A_{\text{ref}}} \frac{dA(x)}{dx}. \quad (3.17)$$

Assuming that the derivative $\frac{dA(x)}{dx}$ is well-defined, we can integrate over the component length l to obtain

$$C_N = \frac{2 \sin \alpha}{A_{\text{ref}}} \int_0^l \frac{dA(x)}{dx} dx = \frac{2 \sin \alpha}{A_{\text{ref}}} [A(l) - A(0)]. \quad (3.18)$$

We then have

$$C_{N_\alpha} = \frac{C_N}{\alpha} = \frac{2}{A_{\text{ref}}} [A(l) - A(0)] \underbrace{\frac{\sin \alpha}{\alpha}}_{\rightarrow 1 \text{ as } \alpha \rightarrow 0}. \quad (3.19)$$

This is the same equation as derived by Barrowman with the exception of the correction term $\sin \alpha / \alpha$.

Equation (3.19) shows that as long as the cross-sectional area of the component changes smoothly, the normal force coefficient derivative does not depend on the component shape, only the difference of the cross-sectional area at the beginning and end. As a consequence, according to Barrowman's theory, a cylindrical body tube has no effect on the normal force coefficient or CP location. However, the lift due to cylindrical body tube sections has been noted to be significant for long, slender rockets even at angles of attack of only a few degrees [13]. An extension for the effect of body lift will be given shortly.

C_{m_α} of body components at subsonic speeds

A normal force $N(x)$ at position x produces a pitching moment

$$m(x) = xN(x). \quad (3.20)$$

at the nose cone tip. Therefore the pitching moment coefficient is

$$C_m(x) = \frac{m(x)}{\frac{1}{2}\rho v_0^2 A_{\text{ref}} d} = \frac{xN(x)}{\frac{1}{2}\rho v_0^2 A_{\text{ref}} d}. \quad (3.21)$$

Substituting equation (3.17) we obtain

$$C_m(x) = \frac{x C_N(x)}{d} = \frac{2 \sin \alpha x}{A_{\text{ref}} d} \frac{dA(x)}{dx}. \quad (3.22)$$

This can be integrated over the length of the body to obtain

$$C_m = \frac{2 \sin \alpha}{A_{\text{ref}} d} \int_0^l x \left(\frac{dA(x)}{dx} \right) dx = \frac{2 \sin \alpha}{A_{\text{ref}} d} \left[lA(l) - \int_0^l A(x) dx \right]. \quad (3.23)$$

The resulting integral is simply the volume of the body V . Therefore we have

$$C_m = \frac{2 \sin \alpha}{A_{\text{ref}} d} [lA(l) - V] \quad (3.24)$$

and

$$C_{m_\alpha} = \frac{2}{A_{\text{ref}} d} [lA(l) - V] \frac{\sin \alpha}{\alpha}. \quad (3.25)$$

This is, again, the result derived by Barrowman with the additional correction term $\sin \alpha/\alpha$.

Effect of body lift

The analysis thus far has neglected the effect of body lift as negligible at small angles of attack. However, in the flight of long, slender rockets the lift may be quite significant at angles of attack of only a few degrees, which may occur at moderate wind speeds [13].

Robert Galejs suggested adding a correction term to the body component C_{N_α} to account for body lift [13]. The normal force exerted on a cylindrical body at an angle of attack α is [15, p. 3-11]

$$C_N = K \frac{A_{\text{plan}}}{A_{\text{ref}}} \sin^2 \alpha \quad (3.26)$$

where $A_{\text{plan}} = d \cdot l$ is the planform area of the cylinder and K is a constant $K \approx 1.1$. Galejs had simplified the equation with $\sin^2 \alpha \approx \alpha^2$, but this shall not be performed here. At small angles of attack, when the approximation is valid, this yields a linear correction to the value of C_{N_α} .

It is assumed that the lift on non-cylindrical components can be approximated reasonably well with the same equation. The CP location is assumed to be the center of the planform area, that is

$$X_{\text{lift}} = \frac{\int_0^l x 2r(x) dx}{A_{\text{plan}}}. \quad (3.27)$$

This is reminiscent of the CP of a rocket flying at an angle of attack of 90° . For a cylinder the CP location is at the center of the body, which is also the CP location obtained at the limit with equation (3.28). However, for nose cones, shoulders and boattails it yields a slightly different position than equation (3.28).

Center of pressure of body components

The CP location of the body components can be calculated by inserting equations (3.19) and (3.25) into equation (3.5):

$$X_B = \frac{(C_{m_\alpha})_B}{(C_{N_\alpha})_B} d = \frac{lA(l) - V}{A(l) - A(0)} \quad (3.28)$$

It is worth noting that the correction term $\sin \alpha / \alpha$ cancels out in the division, however, it is still present in the value of C_{N_α} and is therefore significant at large angles of attack.

The whole rocket body could be numerically integrated and the properties of the whole body computed. However, it is often more descriptive to split the body into components and calculate the parameters separately. The total CP location can be calculated from the separate CP locations X_i and normal force coefficient derivatives $(C_{N_\alpha})_i$ by the moment sum

$$X = \frac{\sum_{i=1}^n X_i (C_{N_\alpha})_i}{\sum_{i=1}^n (C_{N_\alpha})_i}. \quad (3.29)$$

In this manner the effect of the separate components can be more easily analyzed.

3.2.2 Planar fins

The fins of the rocket are considered separately from the body. Their CP location and normal force coefficient are determined and added to the total moment sum (3.29). The interference between the fins and the body is taken into account by a separate correction term.

In addition to the corrective normal force, the fins can induce a roll rate if each of the fins are canted at an angle δ . The roll moment coefficient will be derived separately in Section 3.3.

Barrowman's original report and thesis derived the equations for trapezoidal fins, where the tip chord is parallel to the body (Figure 3.2(a)). The equations can be extended to *e.g.* elliptical fins [16] (Figure 3.2(b)), but many model rocket fin designs depart from these basic shapes. Therefore an extension is presented that approximates the aerodynamical properties for a free-form fin defined by a list of (x, y) coordinates (Figure 3.2(c)).

Additionally, Barrowman considered only cases with three or four fins. This shall be extended to allow for any reasonable number of fins, even single fins.

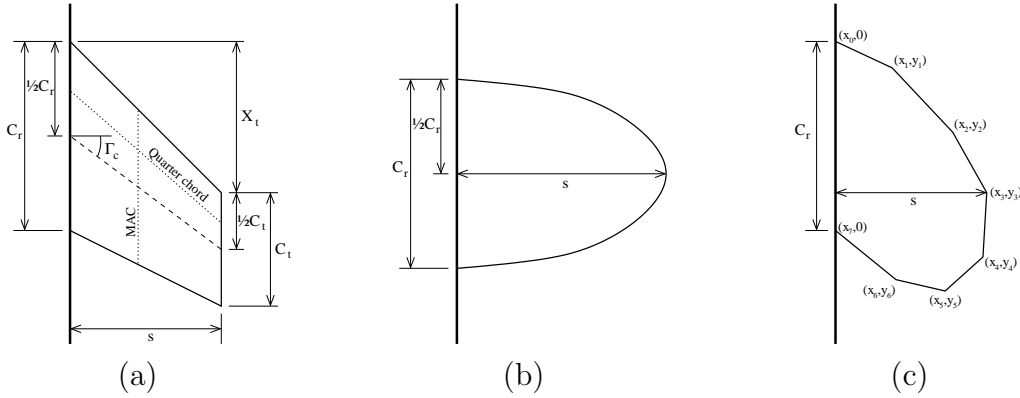


Figure 3.2: Fin geometry of (a) a trapezoidal fin, (b) an elliptical fin and (c) a free-form fin.

Center of pressure of fins at subsonic and supersonic speeds

Barrowman argued that since the CP of a fin is located along its mean aerodynamic chord (MAC) and on the other hand at low subsonic speeds on its quarter chord, then the CP must be located at the intersection of these two (depicted in Figure 3.2(a)). He proceeded to calculate this intersection point analytically from the fin geometry of a trapezoidal fin.

Instead of following the derivation Barrowman used, an alternative method will be presented that allows simpler extension to free-form fins. The two methods yield identical results for trapezoidal fins. The length of the MAC \bar{c} , its spanwise position y_{MAC} , and the effective leading edge location $x_{\text{MAC,LE}}$ are given by [17]

$$\bar{c} = \frac{1}{A_{\text{fin}}} \int_0^s c^2(y) dy \quad (3.30)$$

$$y_{\text{MAC}} = \frac{1}{A_{\text{fin}}} \int_0^s y c(y) dy \quad (3.31)$$

$$x_{\text{MAC,LE}} = \frac{1}{A_{\text{fin}}} \int_0^s x_{\text{LE}}(y) c(y) dy \quad (3.32)$$

where A_{fin} is the one-sided area of a single fin, s is the span of one fin, and $c(y)$ is the length of the fin chord and $x_{\text{LE}}(y)$ the leading edge position at spanwise position y .

When these equations are applied to trapezoidal fins and the lengthwise position of the CP is selected at the quarter chord, $X_f = x_{\text{MAC,LE}} + 0.25 \bar{c}$,

one recovers exactly the results derived by Barrowman:

$$y_{\text{MAC}} = \frac{s}{3} \frac{C_r + 2C_t}{C_r + C_t} \quad (3.33)$$

$$X_f = \frac{X_t}{3} \frac{C_r + 2C_t}{C_r + C_t} + \frac{1}{6} \frac{C_r^2 + C_t^2 + C_r C_t}{C_r + C_t} \quad (3.34)$$

However, equations (3.30)–(3.32) may also be directly applied to elliptical or free-form fins.

Barrowman's method assumes that the lengthwise position of the CP stays at a constant 25% of the MAC at subsonic speeds. However, the position starts moving rearward above approximately Mach 0.5. For $M > 2$ the relative lengthwise position of the CP is given by an empirical formula [18, p. 33]

$$\frac{X_f}{\bar{c}} = \frac{\mathcal{R}\beta - 0.67}{2\mathcal{R}\beta - 1} \quad (3.35)$$

where $\beta = \sqrt{M^2 - 1}$ for $M > 1$ and \mathcal{R} is the aspect ratio of the fin defined using the span s as $\mathcal{R} = 2s^2/A_{\text{fin}}$. Between Mach 0.5 and 2 the lengthwise position of the CP is interpolated. A suitable function that gives a curve similar to that of Figure 2.18 of reference [18, p. 33] was found to be a fifth order polynomial $p(M)$ with the constraints

$$\begin{aligned} p(0.5) &= 0.25 \\ p'(0.5) &= 0 \\ p(2) &= f(2) \\ p'(2) &= f'(2) \\ p''(2) &= 0 \\ p'''(2) &= 0 \end{aligned} \quad (3.36)$$

where $f(M)$ is the function of equation (3.35).

The method presented here can be used to estimate the CP location of an arbitrary thin fin. However, problems arise with the method if the fin shape has a jagged edge as shown in Figure 3.3(a). If $c(y)$ would include only the sum of the two separate chords in the area containing the gap, then the equations would yield the same result as for a fin shown in Figure 3.3(b). This clearly would be incorrect, since the position of the latter fin portion would be neglected. To overcome this problem, $c(y)$ is chosen as the length from the leading edge to the trailing edge of the fin, effectively adding the

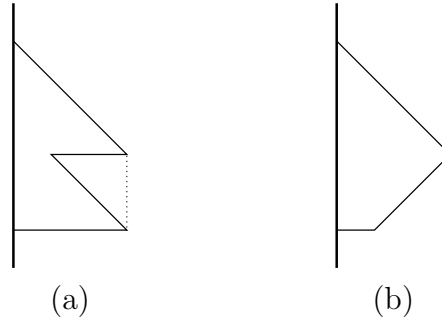


Figure 3.3: (a) A jagged fin edge, and (b) an equivalent fin if $c(y)$ is chosen to include only the actual fin area.

portion marked by the dotted line to the fin. This corrects the CP position slightly rearwards. The fin area used in equations (3.30)–(3.32) must in this case also be calculated including this extra fin area, but the extra area must not be included when calculating the normal force coefficient.

This correction is also approximate, since in reality such a jagged edge would cause some unknown interference factor between the two fin portions. Simulating such jagged edges using these methods should therefore be avoided.

Single fin C_{N_α} at subsonic speeds

Barrowman derived the normal force coefficient derivative value based on Diederich's semi-empirical method [19], which states that for one fin

$$(C_{N_\alpha})_1 = \frac{C_{N_{\alpha 0}} F_D \left(\frac{A_{\text{fin}}}{A_{\text{ref}}} \right) \cos \Gamma_c}{2 + F_D \sqrt{1 + \frac{4}{F_D^2}}}, \quad (3.37)$$

where

$C_{N_{\alpha 0}}$ = normal force coefficient derivative of a 2D airfoil

F_D = Diederich's planform correlation parameter

A_{fin} = area of one fin

Γ_c = midchord sweep angle (depicted in Figure 3.2(a)).

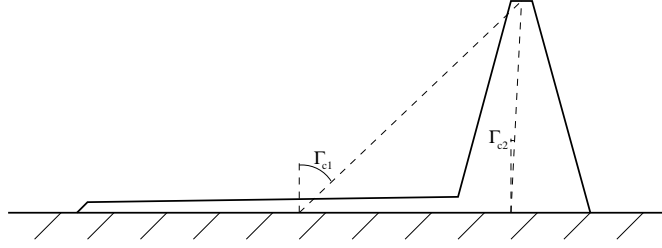


Figure 3.4: A free-form fin shape and two possibilities for the midchord angle Γ_c .

Based on thin airfoil theory of potential flow corrected for compressible flow

$$C_{N_{\alpha 0}} = \frac{2\pi}{\beta} \quad (3.38)$$

where $\beta = \sqrt{1 - M^2}$ for $M < 1$. F_D is a parameter that corrects the normal force coefficient for the sweep of the fin. According to Diederich, F_D is given by

$$F_D = \frac{\mathcal{R}}{\frac{1}{2\pi} C_{N_{\alpha 0}} \cos \Gamma_c}. \quad (3.39)$$

Substituting equations (3.38), (3.39) and $\mathcal{R} = 2s^2/A_{\text{fin}}$ into (3.37) and simplifying one obtains

$$(C_{N_{\alpha}})_1 = \frac{2\pi \frac{s^2}{A_{\text{ref}}}}{1 + \sqrt{1 + \left(\frac{\beta s^2}{A_{\text{fin}} \cos \Gamma_c}\right)^2}}. \quad (3.40)$$

This is the normal force coefficient derivative for one fin, where the angle of attack is between the airflow and fin surface.

The value of equation (3.40) can be calculated directly for trapezoidal and elliptical fins. However, in the case of free-form fins, the question arises of how to define the mid-chord angle Γ_c . If the angle Γ_c is taken as the angle from the middle of the root chord to the tip of the fin, the result may not be representative of the actual shape, as shown by angle Γ_{c1} in Figure 3.4.

Instead the fin planform is divided into a large number of chords, and the angle between the midpoints of each two consecutive chords is calculated. The midchord angle used in equation (3.40) is then the average of all these angles. This produces an angle better representing the actual shape of the fin, as angle Γ_{c2} in Figure 3.4. The angle calculated by this method is also equal to the natural midchord angles for trapezoidal and elliptical fins.

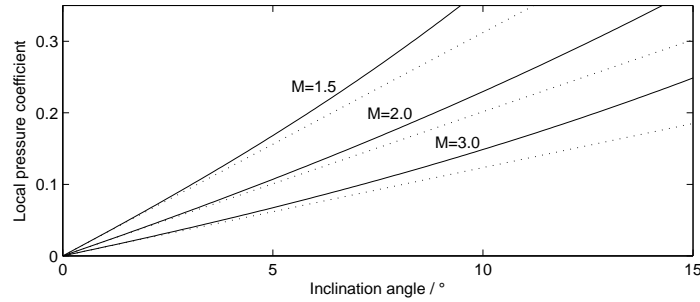


Figure 3.5: The local pressure coefficient as a function of the strip inclination angle at various Mach numbers. The dotted line depicts the linear component of equation (3.41).

Single fin C_{N_α} at supersonic speeds

The method for calculating the normal force coefficient of fins at supersonic speed presented by Barrowman is based on a third-order expansion according to Busemann theory [20]. The method divides the fin into narrow streamwise strips, the normal force of which are calculated separately. In this presentation the method is further simplified by assuming the fins to be flat plates and by ignoring a third-order term that corrects for fin-tip Mach cone effects.

The local pressure coefficient of strip i is calculated by

$$C_{P_i} = K_1 \eta_i + K_2 \eta_i^2 + K_3 \eta_i^3 \quad (3.41)$$

where η_i is the inclination of the flow at the surface and the coefficients are

$$K_1 = \frac{2}{\beta} \quad (3.42)$$

$$K_2 = \frac{(\gamma + 1)M^4 - 4\beta^2}{4\beta^4} \quad (3.43)$$

$$K_3 = \frac{(\gamma + 1)M^8 + (2\gamma^2 - 7\gamma - 5)M^6 + 10(\gamma + 1)M^4 + 8}{6\beta^7} \quad (3.44)$$

It is noteworthy that the coefficients K_1 , K_2 and K_3 can be pre-calculated for various Mach numbers, which makes the pressure coefficient of a single strip very fast to compute. At small angles of inclination the pressure coefficient is nearly linear, as presented in Figure 3.5.

The lift force of strip i is equal to

$$F_i = C_{P_i} \cdot \frac{1}{2} \rho v_0^2 \cdot \underbrace{c_i \Delta y}_{\text{area}}. \quad (3.45)$$

The total lift force of the fin is obtained by summing up the contributions of all fin strips. The normal force coefficient is then calculated in the usual manner as

$$C_N = \frac{\sum_i F_i}{\frac{1}{2} \rho v_0^2 A_{\text{ref}}} \quad (3.46)$$

$$= \frac{1}{A_{\text{ref}}} \sum_i C_{P_i} \cdot c_i \Delta y. \quad (3.47)$$

When computing the corrective normal force coefficient of the fins the effect of roll is not taken into account. In this case, and assuming that the fins are flat plates, the inclination angles η_i of all strips are the same, and the pressure coefficient is constant over the entire fin. Therefore the normal force coefficient is simply

$$(C_N)_1 = \frac{A_{\text{fin}}}{A_{\text{ref}}} C_P. \quad (3.48)$$

Since the pressure coefficient is not linear with the angle of attack, the normal force coefficient slope is defined using equation (3.8) as

$$(C_{N_\alpha})_1 = \frac{(C_N)_1}{\alpha} = \frac{A_{\text{fin}}}{A_{\text{ref}}} (K_1 + K_2 \alpha + K_3 \alpha^2). \quad (3.49)$$

Multiple fin C_{N_α}

In his thesis, Barrowman considered only configurations with three and four fins, one of which was parallel to the lateral airflow. For simulation purposes, it is necessary to lift these restrictions to allow for any direction of lateral airflow and for any number of fins.

The lift force of a fin is perpendicular to the fin and originates from its CP. Therefore a single fin may cause a rolling and yawing moment in addition to a pitching moment. In this case all of the forces and moments must be computed from the geometry. If there are two or more fins placed symmetrically around the body then the yawing moments cancel, and if additionally there

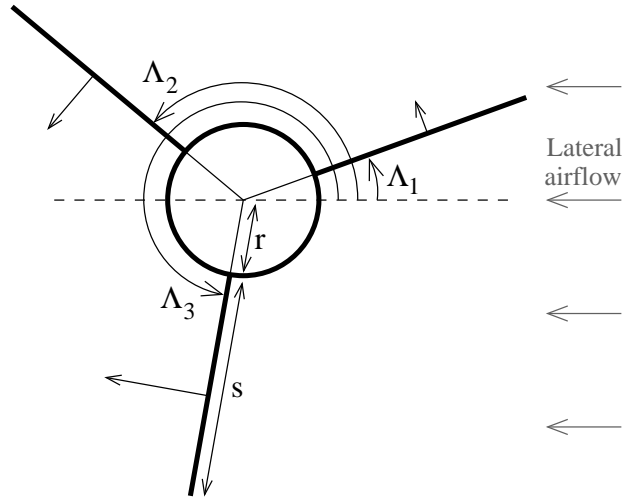


Figure 3.6: The geometry of an uncanted three-fin configuration (viewed from rear).

is no fin cant then the total rolling moment is also zero, and these moments need not be computed.

The geometry of an uncanted fin configuration is depicted in Figure 3.6. The dihedral angle between each of the fins and the airflow direction is denoted Λ_i . The fin i encounters a local angle of attack of

$$\alpha_i = \alpha \sin \Lambda_i \quad (3.50)$$

for which the normal force component (the component parallel to the lateral airflow) is then

$$(C_{N\alpha})_{\Lambda_i} = (C_{N\alpha})_1 \sin^2 \Lambda_i. \quad (3.51)$$

The sum of the coefficients for N fins then yields

$$\sum_{k=1}^N (C_{N\alpha})_{\Lambda_k} = (C_{N\alpha})_1 \sum_{k=1}^N \sin^2 \Lambda_k. \quad (3.52)$$

However, when $N \geq 3$ and the fins are spaced equally around the body of the rocket, the sum simplifies to a constant

$$\sum_{k=1}^N \sin^2(2\pi k/N + \theta) = \frac{N}{2}. \quad (3.53)$$

This equation predicts that the normal force produced by three or more fins is independent of the roll angle θ of the vehicle. Investigation by Pettis [21] showed that the normal force coefficient derivative of a four-finned rocket at Mach 1.48 decreased by approximately 6% at a roll angle of 45° , and the roll angle had negligible effect on an eight-finned rocket. Experimental data of a four-finned sounding rocket at Mach speeds from 0.60 to 1.20 supports the 6% estimate [22].

The only experimental data available to the author of three-fin configurations was of a rocket with a rounded triangular body cross section [23]. This data suggests an effect of approximately 15% on the normal force coefficient derivative depending on the roll angle. However, it is unknown how much of this effect is due to the triangular body shape and how much from the fin positioning.

It is also hard to predict such an effect when examining singular fins. If three identical or very similar singular fins are placed on a rocket body, the effect should be the same as when the fins belong to the same three-fin configuration. Due to these facts the effect of the roll angle on the normal force coefficient derivative is ignored when a fin configuration has three or more fins.¹

However, in configurations with many fins the fin–fin interference may cause the normal force to be less than that estimated directly by equation (3.52). According to reference [24, p. 5-24], the normal force coefficients for six and eight-fin configurations are 1.37 and 1.62 times that of the corresponding four-fin configuration, respectively. The values for five and seven-fin configurations are interpolated between these values.

¹In OpenRocket versions prior to 0.9.6 a sinusoidal reduction of 15% and 6% was applied to three- and four-fin configurations, respectively. However, this sometimes caused a significantly different predicted CP location compared to the pure Barrowman method, and also caused a discrepancy when such a fin configuration was decomposed into singular fins. It was deemed better to follow the tested and tried Barrowman method instead of introducing additional terms to the equation.

Altogether, the normal force coefficient derivative $(C_{N_\alpha})_N$ is calculated by:

$$(C_{N_\alpha})_N = \left(\sum_{k=1}^N \sin^2 \Lambda_k \right) (C_{N_\alpha})_1 \cdot \begin{cases} 1.000 & N_{\text{tot}} = 1, 2, 3, 4 \\ 0.948 & N_{\text{tot}} = 5 \\ 0.913 & N_{\text{tot}} = 6 \\ 0.854 & N_{\text{tot}} = 7 \\ 0.810 & N_{\text{tot}} = 8 \\ 0.750 & N_{\text{tot}} > 8 \end{cases} \quad (3.54)$$

Here N is the number of fins in this fin set, while N_{tot} is the total number of parallel fins that have an interference effect. The sum term simplifies to $N/2$ for $N \geq 3$ according to equation (3.53). The interference effect for $N_{\text{tot}} > 8$ is assumed at 25%, as data for such configurations is not available and such configurations are rare and eccentric in any case.

Fin–body interference

The normal force coefficient must still be corrected for fin–body interference, which increases the overall produced normal force. Here two distinct effects can be identified: the normal force on the fins due to the presence of the body and the normal force on the body due to the presence of fins. Of these the former is significantly larger; the latter is therefore ignored. The effect of the extra fin lift is taken into account using a correction term

$$(C_{N_\alpha})_{T(B)} = K_{T(B)} (C_{N_\alpha})_N \quad (3.55)$$

where $(C_{N_\alpha})_{T(B)}$ is the normal force coefficient derivative of the tail in the presence of the body. The term $K_{T(B)}$ can be approximated by [3]

$$K_{T(B)} = 1 + \frac{r_t}{s + r_t}, \quad (3.56)$$

where s is the fin span from root to tip and r_t is the body radius at the fin position. The value $(C_{N_\alpha})_{T(B)}$ is then used as the final normal force coefficient derivative of the fins.

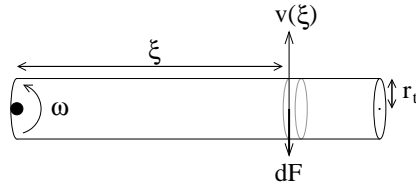


Figure 3.7: Pitch damping moment due to a pitching body component.

3.2.3 Pitch damping moment

So far the effect of the current pitch angular velocity has been ignored as marginal. This is the case during the upward flight of a stable rocket. However, if a rocket is launched nearly vertically in still air, the rocket flips over rather rapidly at apogee. In some cases it was observed that the rocket was left wildly oscillating during descent. The pitch damping moment opposes the fast rotation of the rocket thus damping the oscillation.

Since the pitch damping moment is notable only at apogee, and therefore does not contribute to the overall flight characteristics, only a rough estimate of its magnitude is required. A cylinder in perpendicular flow has a drag coefficient of approximately $C_D = 1.1$, with the reference area being the planform area of the cylinder [15, p. 3-11]. Therefore a short piece of cylinder $d\xi$ at a distance ξ from a rotation axis, as shown in Figure 3.7, produces a force

$$dF = 1.1 \cdot \frac{1}{2} \rho (\omega \xi)^2 \cdot \underbrace{2r_t}_{\text{ref.area}} d\xi \quad (3.57)$$

when the cylinder is rotating at an angular velocity ω . The produced moment is correspondingly $dm = \xi dF$. Integrating this over $0 \dots l$ yields the total pitch moment

$$m = 0.275 \cdot \rho r_t l^4 \omega^2 \quad (3.58)$$

and thus the moment damping coefficient is

$$C_{\text{damp}} = 0.55 \cdot \frac{l^4 r_t}{A_{\text{ref}} d} \cdot \frac{\omega^2}{v_0^2}. \quad (3.59)$$

This value is computed separately for the portions of the rocket body fore and aft of the CG using an average body radius as r_t .

Similarly, a fin with area A_{fin} at a distance ξ from the CG produces a moment

of approximately

$$C_{\text{damp}} = 0.6 \cdot \frac{N A_{\text{fin}} \xi^3}{A_{\text{ref}} d} \cdot \frac{\omega^2}{v_0^2} \quad (3.60)$$

where the effective area of the fins is assumed to be $A_{\text{fin}} \cdot N/2$. For $N > 4$ the value $N = 4$ is used, since the other fins are not exposed to any direct airflow.

The damping moments are applied to the total pitch moment in the opposite direction of the current pitch rate. It is noteworthy that the damping moment coefficients are proportional to ω^2/v_0^2 , confirming that the damping moments are insignificant during most of the rocket flight, where the angles of deflection are small and the velocity of the rocket large. Through roll coupling the yaw rate may also momentarily become significant, and therefore the same correction is also applied to the yaw moment.

3.3 Roll dynamics

When the fins of a rocket are canted at some angle $\delta > 0$, the fins induce a rolling moment on the rocket. On the other hand, when a rocket has a specific roll velocity, the portions of the fin far from the rocket centerline encounter notable tangential velocities which oppose the roll. Therefore a steady-state roll velocity, dependent on the current velocity of the rocket, will result.

The effect of roll on a fin can be examined by dividing the fin into narrow streamwise strips and later integrating over the strips. A strip i at distance ξ_i from the rocket centerline encounters a radial velocity

$$u_i = \omega \xi_i \quad (3.61)$$

where ω is the angular roll velocity, as shown in Figure 3.8. The radial velocity induces an angle of attack

$$\eta_i = \tan^{-1} \left(\frac{u_i}{v_0} \right) = \tan^{-1} \left(\frac{\omega \xi_i}{v_0} \right) \approx \frac{\omega \xi_i}{v_0} \quad (3.62)$$

to the strip. The approximation $\tan^{-1} \eta \approx \eta$ is valid for $u_i \ll v_0$, that is, when the velocity of the rocket is large compared to the radial velocity. The approximation is reasonable up to angles of $\eta \approx 20^\circ$, above which angle most fins stall, which limits the validity of the equation in any case.

When a fin is canted at an angle δ , the total inclination of the strip to the airflow is

$$\alpha_i = \delta - \eta_i. \quad (3.63)$$

Assuming that the force produced by a strip is directly proportional to the local angle of attack, the force on strip i is

$$F_i = k_i \alpha_i = k_i (\delta - \eta_i) \quad (3.64)$$

for some k_i . The total moment produced by the fin is then

$$l = \sum_i \xi_i F_i = \sum_i \xi_i k_i (\delta - \eta_i) = \sum_i \xi_i k_i \delta - \sum_i \xi_i k_i \eta_i. \quad (3.65)$$

This shows that the effect of roll can be split into two components: the first term $\sum_i \xi_i k_i \delta$ is the roll moment induced by a fin canted at the angle δ when

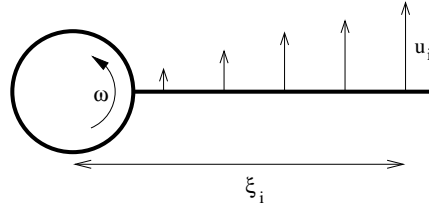


Figure 3.8: Radial velocity at different positions of a fin. Viewed from the rear of the rocket.

flying at zero roll rate ($\omega = 0$), while the second term $\sum_i \xi_i k_i \eta_i$ is the opposing moment generated by an uncanted fin ($\delta = 0$) when flying at a roll rate ω . These two moments are called the roll forcing moment and roll damping moment, respectively. These components will be analyzed separately.

3.3.1 Roll forcing coefficient

As shown previously, the roll forcing coefficient can be computed by examining a rocket with fins canted at an angle δ flying at zero roll rate ($\omega = 0$). In this case, the cant angle δ acts simply as an angle of attack for each of the fins. Therefore, the methods computed in the previous section can be directly applied. Because the lift force of a fin originates from the mean aerodynamic chord, the roll forcing coefficient of N fins is equal to

$$C_{lf} = \frac{N(y_{MAC} + r_t)(C_{N_\alpha})_1 \delta}{d} \quad (3.66)$$

where y_{MAC} and $(C_{N_\alpha})_1$ are computed using the methods described in Section 3.2.2 and r_t is the radius of the body tube at the fin position. This result is applicable for both subsonic and supersonic speeds.

3.3.2 Roll damping coefficient

The roll damping coefficient is computed by examining a rocket with uncanted fins ($\delta = 0$) flying at a roll rate ω . Since different portions of the fin encounter different local angles of attack, the damping moment must be computed from the separate streamwise airfoil strips.

At subsonic speeds the force generated by strip i is equal to

$$F_i = C_{N_{\alpha 0}} \frac{1}{2} \rho v_0^2 \underbrace{c_i \Delta \xi_i}_{\text{area}} \eta_i. \quad (3.67)$$

Here $C_{N_{\alpha 0}}$ is calculated by equation (3.38) and $c_i \Delta \xi_i$ is the area of the strip. The roll damping moment generated by the strip is then

$$(C_{ld})_i = \frac{F_i \xi_i}{\frac{1}{2} \rho v_0^2 A_{\text{ref}} d} = \frac{C_{N_{\alpha 0}}}{A_{\text{ref}} d} \xi_i c_i \Delta \xi_i \eta_i. \quad (3.68)$$

By applying the approximation (3.62) and summing (integrating) the airfoil strips the total roll damping moment for N fins is obtained as:

$$\begin{aligned} C_{ld} &= N \sum_i (C_{ld})_i \\ &= \frac{N C_{N_{\alpha 0}} \omega}{A_{\text{ref}} d v_0} \sum_i c_i \xi_i^2 \Delta \xi_i. \end{aligned} \quad (3.69)$$

The sum term is a constant for a specific fin shape. It can be computed numerically from the strips or analytically for specific shapes. For trapezoidal fins the term can be integrated as

$$\sum_i c_i \xi_i^2 \Delta \xi_i = \frac{C_r + C_t}{2} r_t^2 s + \frac{C_r + 2C_t}{3} r_t s^2 + \frac{C_r + 3C_t}{12} s^3 \quad (3.70)$$

and for elliptical fins

$$\sum_i c_i \xi_i^2 \Delta \xi_i = C_r \left(\frac{\pi}{4} r_t^2 s + \frac{2}{3} r_t s^2 + \frac{\pi}{16} s^3 \right). \quad (3.71)$$

The roll damping moment at supersonic speeds is calculated analogously, starting from the supersonic strip lift force, equation (3.45), where the angle of inclination of each strip is calculated using equation (3.62). The roll moment at supersonic speeds is thus

$$C_{ld} = \frac{N}{A_{\text{ref}} d} \sum_i C_{P_i} c_i \xi_i \Delta \xi_i. \quad (3.72)$$

The dependence on the incidence angle η_i is embedded within the local pressure coefficient C_{P_i} , equation (3.41). Since the dependence is non-linear, the sum term is a function of the Mach number as well as the fin shape.

3.3.3 Equilibrium roll frequency

One quantity of interest when examining rockets with canted fins is the steady-state roll frequency that the fins induce on a rocket flying at a specific velocity. This is obtained by equating the roll forcing moment (3.66) and roll damping moment (3.69) and solving for the roll rate ω . The equilibrium roll frequency at subsonic speeds is therefore

$$f_{\text{eq}} = \frac{\omega_{\text{eq}}}{2\pi} = \frac{A_{\text{ref}} \beta v_0 y_{\text{MAC}} (C_{N_\alpha})_1 \delta}{4\pi^2 \sum_i c_i \xi_i^2 \Delta \xi_i} \quad (3.73)$$

It is worth noting that the arbitrary reference area A_{ref} is cancelled out by the reference area appearing within $(C_{N_\alpha})_1$, as is to be expected.

At supersonic speeds the dependence on the incidence angle is non-linear and therefore the equilibrium roll frequency must be solved numerically. Alternatively, the second and third-order terms of the local pressure coefficient of equation (3.41) may be ignored, in which case an approximation for the equilibrium roll frequency nearly identical to the subsonic case is obtained:

$$f_{\text{eq}} = \frac{\omega_{\text{eq}}}{2\pi} = \frac{A_{\text{ref}} \beta v_0 y_{\text{MAC}} (C_{N_\alpha})_1 \delta}{4\pi \sum_i c_i \xi_i^2 \Delta \xi_i} \quad (3.74)$$

The value of $(C_{N_\alpha})_1$ must, of course, be computed using different methods in the subsonic and supersonic cases.

3.4 Drag forces

Air flowing around a solid body causes drag, which resists the movement of the object relative to the air. Drag forces arise from two basic mechanisms, the air pressure distribution around the rocket and skin friction. The pressure distribution is further divided into body pressure drag (including shock waves generated as supersonic speeds), parasitic pressure drag due to protrusions such as launch lugs and base drag. Additional sources of drag include interference between the fins and body and vortices generated at fin tips when flying at an angle of attack. The different drag sources are depicted in Figure 3.9. Each drag source will be analyzed separately; the interference drag and fin-tip vortices will be ignored as small compared to the other sources.

As described in Section 3.1, two different drag coefficients can be defined: the (total) drag coefficient C_D and the axial drag coefficient C_A . At zero angle of attack these two coincide, $C_{D_0} = C_{A_0}$, but at other angles a distinction between the two must be made. The value of significance in the simulation is the axial drag coefficient C_A based on the choice of force components. However, the drag coefficient C_D describes the deceleration force on the rocket, and is a more commonly known value in the rocketry community, so it is informational to calculate its value as well.

In this section the zero angle-of-attack drag coefficient $C_{D_0} = C_{A_0}$ will be computed first. Then, in Section 3.4.7 this will be extended for angles of attack and C_A and C_D will be computed. Since the drag force of each component is proportional to its particular size, the subscript \bullet will be used for coefficients that are computed using the reference area of the specific component. This reference area is the frontal area of the component unless otherwise noted. Conversion to the global reference area is performed by

$$C_{D_0} = \frac{A_{\text{component}}}{A_{\text{ref}}} \cdot C_{D\bullet}. \quad (3.75)$$

3.4.1 Laminar and turbulent boundary layers

At the front of a streamlined body, air flows smoothly around the body in layers, each of which has a different velocity. The layer closest to the surface

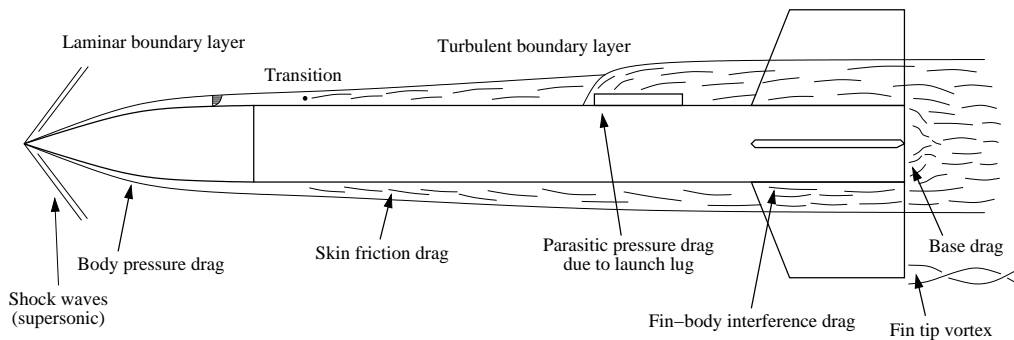


Figure 3.9: Types of model rocket drag at subsonic speeds.

“sticks” to the object having zero velocity. Each layer gradually increases the speed until the free-stream velocity is reached. This type of flow is said to be *laminar* and to have a *laminar boundary layer*. The thickness of the boundary layer increases with the distance the air has flowed along the surface. At some point a transition occurs and the layers of air begin to mix. The boundary layer becomes *turbulent* and thickens rapidly. This transition is depicted in Figure 3.9.

A turbulent boundary layer induces a notably larger skin friction drag than a laminar boundary layer. It is therefore necessary to consider how large a portion of a rocket is in laminar flow and at what point the flow becomes turbulent. The point at which the flow becomes turbulent is the point that has a *local critical Reynolds number*

$$R_{\text{crit}} = \frac{v_0 x}{\nu}, \quad (3.76)$$

where v_0 is the free-stream air velocity, x is the distance along the body from the nose cone tip and $\nu \approx 1.5 \cdot 10^{-5} \text{ m}^2/\text{s}$ is the kinematic viscosity of air. The critical Reynolds number is approximately $R_{\text{crit}} = 5 \cdot 10^5$ [4, p. 43]. Therefore, at a velocity of 100 m/s the transition therefore occurs approximately 7 cm from the nose cone tip.

Surface roughness or even slight protrusions may also trigger the transition to occur prematurely. At a velocity of 60 m/s the critical height for a cylindrical protrusion all around the body is of the order of 0.05 mm [14, p. 348]. The body-to-nosecone joint, a severed paintbrush hair or some other imperfection on the surface may easily exceed this limit and cause premature transition to occur.

Barrowman presents methods for computing the drag of both fully turbulent

boundary layers as well as partially-laminar layers. Both methods were implemented and tested, but the difference in apogee altitude was less than 5% in with all tested designs. Therefore, the boundary layer is assumed to be fully turbulent in all cases.

3.4.2 Skin friction drag

Skin friction is one of the most notable sources of model rocket drag. It is caused by the friction of the viscous flow of air around the rocket. In his thesis Barrowman presented formulae for estimating the skin friction coefficient for both laminar and turbulent boundary layers as well as the transition between the two [4, pp. 43–47]. As discussed above, a fully turbulent boundary layer will be assumed in this thesis.

The skin friction coefficient C_f is defined as the drag coefficient due to friction with the reference area being the total wetted area of the rocket, that is, the body and fin area in contact with the airflow:

$$C_f = \frac{D_{\text{friction}}}{\frac{1}{2}\rho v_0^2 A_{\text{wet}}} \quad (3.77)$$

The coefficient is a function of the rocket's Reynolds number R and the surface roughness. The aim is to first calculate the skin friction coefficient, then apply corrections due to compressibility and geometry effects, and finally to convert the coefficient to the proper reference area.

Skin friction coefficients

The values for C_f are given by different formulae depending on the Reynolds number. For fully turbulent flow the coefficient is given by

$$C_f = \frac{1}{(1.50 \ln R - 5.6)^2}. \quad (3.78)$$

The above formula assumes that the surface is “smooth” and the surface roughness is completely submerged in a thin, laminar sublayer. At sufficient speeds even slight roughness may have an effect on the skin friction. The

Table 3.2: Approximate roughness heights of different surfaces [15, p. 5-3]

Type of surface	Height / μm
Average glass	0.1
Finished and polished surface	0.5
Optimum paint-sprayed surface	5
Planed wooden boards	15
Paint in aircraft mass production	20
Smooth cement surface	50
Dip-galvanized metal surface	150
Incorrectly sprayed aircraft paint	200
Raw wooden boards	500
Average concrete surface	1000

critical Reynolds number corresponding to the roughness is given by

$$R_{\text{crit}} = 51 \left(\frac{R_s}{L} \right)^{-1.039}, \quad (3.79)$$

where R_s is an approximate roughness height of the surface. A few typical roughness heights are presented in Table 3.2. For Reynolds numbers above the critical value, the skin friction coefficient can be considered independent of Reynolds number, and has a value of

$$C_f = 0.032 \left(\frac{R_s}{L} \right)^{0.2}. \quad (3.80)$$

Finally, a correction must be made for very low Reynolds numbers. The experimental formulae are applicable above approximately $R \approx 10^4$. This corresponds to velocities typically below 1 m/s, which therefore have negligible effect on simulations. Below this Reynolds number, the skin friction coefficient is assumed to be equal as for $R = 10^4$.

Altogether, the skin friction coefficient for turbulent flow is calculated by

$$C_f = \begin{cases} 1.48 \cdot 10^{-2}, & \text{if } R < 10^4 \\ \text{Eq. (3.78)}, & \text{if } 10^4 < R < R_{\text{crit}} \\ \text{Eq. (3.80)}, & \text{if } R > R_{\text{crit}} \end{cases}. \quad (3.81)$$

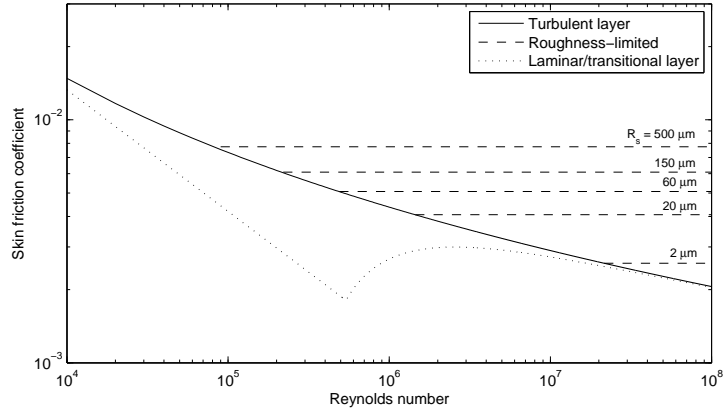


Figure 3.10: Skin friction coefficient of turbulent, laminar and roughness-limited boundary layers.

These formulae are plotted with a few different surface roughnesses in Figure 3.10. Included also is the laminar and transitional skin friction values for comparison.

Compressibility corrections

At subsonic speeds the skin friction coefficient turbulent and roughness-limited boundary layers need to be corrected for compressibility with the factor

$$C_{f_c} = C_f (1 - 0.1 M^2). \quad (3.82)$$

In supersonic flow, the turbulent skin friction coefficient must be corrected with

$$C_{f_c} = \frac{C_f}{(1 + 0.15 M^2)^{0.58}} \quad (3.83)$$

and the roughness-limited value with

$$C_{f_c} = \frac{C_f}{1 + 0.18 M^2}. \quad (3.84)$$

However, the corrected roughness-limited value should not be used if it would yield a value smaller than the corresponding turbulent value.

Skin friction drag coefficient

After correcting the skin friction coefficient for compressibility effects, the coefficient can be converted into the actual drag coefficient. This is performed by scaling it to the correct reference area. The body wetted area is corrected for its cylindrical geometry, and the fins for their finite thickness. The total friction drag coefficient is then

$$(C_D)_{\text{friction}} = C_{f_c} \frac{\left(1 + \frac{1}{2f_B}\right) \cdot A_{\text{wet,body}} + \left(1 + \frac{2t}{\bar{c}}\right) \cdot A_{\text{wet,fins}}}{A_{\text{ref}}} \quad (3.85)$$

where f_B is the fineness ratio of the rocket, and t the thickness and \bar{c} the mean aerodynamic chord length of the fins. The wetted area of the fins $A_{\text{wet,fins}}$ includes both sides of the fins.

3.4.3 Body pressure drag

Pressure drag is caused by the air being forced around the rocket. A special case of pressure drag are shock waves generated at supersonic speeds. In this section methods for estimating the pressure drag of nose cones will be presented and reasonable estimates also for shoulders and boattails.

Nose cone pressure drag

At subsonic speeds the pressure drag of streamlined nose cones is significantly smaller than the skin friction drag. In fact, suitable shapes may even yield negative pressure drag coefficients, producing a slight reduction in drag. Figure 3.11 presents various nose cone shapes and their respective measured pressure drag coefficients. [15, p. 3-12]

It is notable that even a slight rounding at the joint between the nose cone and body reduces the drag coefficient dramatically. Rounding the edges of an otherwise flat head reduces the drag coefficient from 0.8 to 0.2, while a spherical nose cone has a coefficient of only 0.01. The only cases where an appreciable pressure drag is present is when the joint between the nose cone and body is not smooth, which may cause slight flow separation.

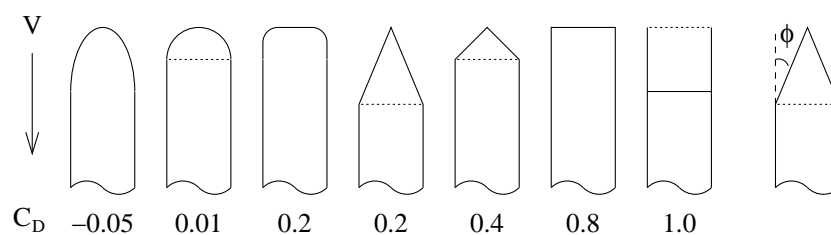


Figure 3.11: Pressure drag of various nose cone shapes [15, p. 3-12].

The nose pressure drag is approximately proportional to the square of the sine of the joint angle ϕ (shown in Figure 3.11) [25, p. 237]:

$$(C_{D\bullet, M=0})_p = 0.8 \cdot \sin^2 \phi. \quad (3.86)$$

This yields a zero pressure drag for all nose cone shapes that have a smooth transition to the body. The equation does not take into account the effect of extremely blunt nose cones (length less than half of the diameter). Since the main drag cause is slight flow separation, the coefficient cannot be corrected for compressibility effects using the Prandtl coefficient, and the value is applicable only at low subsonic velocities.

At supersonic velocities shock waves increase the pressure drag dramatically. In his report Barrowman uses a second-order shock-expansion method that allows determining the pressure distribution along an arbitrary slender rotationally symmetrical body [26]. However, the method has some problematic limitations. The method cannot handle body areas that have a slope larger than approximately 30° , present in several typical nose cone shapes. The local airflow in such areas may decrease below the speed of sound, and the method cannot handle transonic effects. Drag in the transonic region is of special interest for rocketeers wishing to build rockets capable of penetrating the sound barrier.

Instead of a general piecewise computation of the air pressure around the nose cone, a simpler semi-empirical method for estimating the transonic and supersonic pressure drag of nose cones is used. The method, described in detail in Appendix B, combines theoretical and empirical data of different nose cone shapes to allow estimating the pressure drag of all the nose cone shapes described in Appendix A.

The semi-empirical method is used at Mach numbers above 0.8. At high subsonic velocities the pressure drag is interpolated between that predicted

by equation (3.86) and the transonic method. The pressure drag is assumed to be non-decreasing in the subsonic region and to have zero derivative at $M = 0$. A suitable interpolation function that resembles the shape of the Prandtl factor is

$$(C_{D\bullet})_{\text{pressure}} = a \cdot M^b + (C_{D\bullet, M=0})_p \quad (3.87)$$

where a and b are computed to fit the drag coefficient and its derivative at the lower bound of the transonic method.

Shoulder pressure drag

Neither Barrowman nor Hoerner present theoretical or experimental data on the pressure drag of transitions at subsonic velocities. In the case of shoulders, the pressure drag coefficient is assumed to be the same as that of a nose cone, except that the reference area is the difference between the aft and fore ends of the transition. The effect of a non-smooth transition at the beginning of the shoulder is ignored, since this causes an increase in pressure and thus cannot cause flow separation.

While this assumption is reasonable at subsonic velocities, it is somewhat dubious at supersonic velocities. However, no comprehensive data set of shoulder pressure drag at supersonic velocities was found. Therefore the same assumption is made for supersonic velocities and a warning is generated during such simulations (see Section 5.1.4). The refinement of the supersonic shoulder pressure drag estimation is left as a future enhancement.

Boattail pressure drag

The estimate for boattail pressure drag is based on the body base drag estimate, which will be presented in Section 3.4.5. At one extreme, the transition length is zero, in which case the boattail pressure drag will be equal to the total base drag. On the other hand, a gentle slope will allow a gradual pressure change causing approximately zero pressure drag. Hoerner has presented pressure drag data for wedges, which suggests that at a length-to-height ratio below 1 has a constant pressure drag corresponding to the base drag and above a ratio of 3 the pressure drag is negligible. Based on this and the base

drag equation (3.94), an approximation for the pressure drag of a boattail is given as

$$(C_{D\bullet})_{\text{pressure}} = \frac{A_{\text{base}}}{A_{\text{boattail}}} \cdot (C_{D\bullet})_{\text{base}} \cdot \begin{cases} 1 & \text{if } \gamma < 1 \\ \frac{3-\gamma}{2} & \text{if } 1 < \gamma < 3 \\ 0 & \text{if } \gamma > 3 \end{cases} \quad (3.88)$$

where the length-to-height ratio $\gamma = l/(d_1 - d_2)$ is calculated from the length and fore and aft diameters of the boattail. The ratios 1 and 3 correspond to reduction angles of 27° and 9° , respectively, for a conical boattail. The base drag $(C_{D\bullet})_{\text{base}}$ is calculated using equation (3.94).

Again, this approximation is made primarily based on subsonic data. At supersonic velocities expansion fans exist, the counterpart of shock waves in expanding flow. However, the same equation is used for subsonic and supersonic flow and a warning is generated during transonic simulation of boattails.

3.4.4 Fin pressure drag

The fin pressure drag is highly dependent on the fin profile shape. Three typical shapes are considered, a rectangular profile, rounded leading and trailing edges, and an airfoil shape with rounded leading edge and tapering trailing edge. Barrowman estimates the fin pressure drag by dividing the drag further into components of a finite thickness leading edge, thick trailing edge and overall fin thickness [4, p. 48–57]. In this report the fin thickness was already taken into account as a correction to the skin friction drag in Section 3.4.2. The division to leading and trailing edges also allows simple extension to the different profile shapes.

The drag of a rounded leading edge can be considered as a circular cylinder in cross flow with no base drag. Barrowman derived an empirical formula for the leading edge pressure drag as

$$(C_{D\bullet})_{LE\perp} = \begin{cases} (1 - M^2)^{-0.417} - 1 & \text{for } M < 0.9 \\ 1 - 1.785(M - 0.9) & \text{for } 0.9 < M < 1 \\ 1.214 - \frac{0.502}{M^2} + \frac{0.1095}{M^4} & \text{for } M > 1 \end{cases} \quad (3.89)$$

The subscript \perp signifies the the flow is perpendicular to the leading edge.

In the case of a rectangular fin profile the leading edge pressure drag is equal to the stagnation pressure drag as derived in equation B.2 of Appendix B.1:

$$(C_{D\bullet})_{LE\perp} = (C_{D\bullet})_{stag} \quad (3.90)$$

The leading edge pressure drag of a slanted fin is obtained from the cross-flow principle [15, p. 3-11] as

$$(C_{D\bullet})_{LE} = (C_{D\bullet})_{LE\perp} \cdot \cos^2 \Gamma_L \quad (3.91)$$

where Γ_L is the leading edge angle. Note that in the equation both coefficients are relative to the frontal area of the cylinder, so the ratio of their reference areas is also $\cos \Gamma_L$. In the case of a free-form fin the angle Γ_L is the average leading edge angle, as described in Section 3.2.2.

The fin base drag coefficient of a square profile fin is the same as the body base drag coefficient in equation 3.94:

$$(C_{D\bullet})_{TE} = (C_{D\bullet})_{base} \quad (3.92)$$

For fins with rounded edges the value is taken as half of the total base drag, and for fins with tapering trailing edges the base drag is assumed to be zero.

The total fin pressure drag is the sum of the leading and trailing edge drags

$$(C_{D\bullet})_{pressure} = (C_{D\bullet})_{LE} + (C_{D\bullet})_{TE}. \quad (3.93)$$

The reference area is the fin frontal area $N \cdot ts$.

3.4.5 Base drag

Base drag is caused by a low-pressure area created at the base of the rocket or in any place where the body radius diminishes rapidly enough. The magnitude of the base drag can be estimated using the empirical formula [18, p. 23]

$$(C_{D\bullet})_{base} = \begin{cases} 0.12 + 0.13M^2, & \text{if } M < 1 \\ 0.25/M, & \text{if } M > 1 \end{cases} \quad (3.94)$$

The base drag is disrupted when a motor exhausts into the area. A full examination of the process would need much more detailed information about

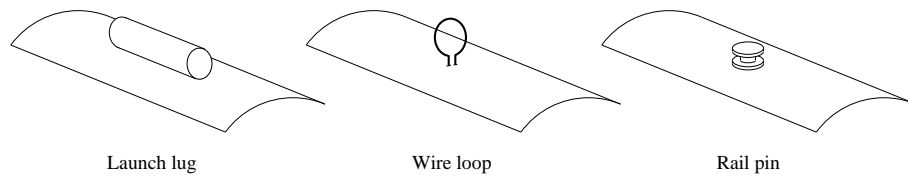


Figure 3.12: Three types of common launch guides.

the motor and would be unnecessarily complicated. A reasonable approximation is achieved by subtracting the area of the thrusting motors from the base reference area [18, p. 23]. Thus, if the base is the same size as the motor itself, no base drag is generated. On the other hand, if the base is large with only a small motor in the center, the base drag is approximately the same as when coasting.

The equation presented above ignores the effect that the rear body slope angle has on the base pressure. A boattail at the end of the rocket both diminishes the reference area of base drag, thus reducing drag, but the slope also directs air better into the low pressure area. This effect has been neglected as small compared to the effect of reduced base area.

3.4.6 Parasitic drag

Parasitic drag refers to drag caused by imperfections and protrusions on the rocket body. The most significant source of parasitic drag in model rockets are the launch guides that protrude from the rocket body. The most common type of launch guide is one or two launch lugs, which are pieces of tube that hold the rocket on the launch rod during takeoff. Alternatives to launch lugs include replacing the tube with metal wire loops or attaching rail pins that hold the rocket on a launch rail. These three guide types are depicted in Figure 3.12. The effect of launch lugs on the total drag of a model rocket is small, typically in the range of 0–10%, due to their comparatively small size. However, studying this effect may be of notable interest for model rocket designers.

A launch lug that is long enough that no appreciable airflow occurs through the lug may be considered a solid cylinder next to the main rocket body. A rectangular protrusion that has a length at least twice its height has a drag coefficient of 0.74, with reference area being its frontal area [15, p. 5-8]. The

drag coefficient varies proportional to the stagnation pressure as in the case of a blunt cylinder in free airflow, presented in Appendix B.1.

A wire held perpendicular to airflow has instead a drag coefficient of 1.1, where the reference area is the planform area of the wire [15, p. 3-11]. A wire loop may be thought of as a launch lug with length and wall thickness equal to the thickness of the wire. However, in this view of a launch lug the reference area must not include the inside of the tube, since air is free to flow within the loop.

These two cases may be unified by changing the used reference area as a function of the length of the tube l . At the limit $l = 0$ the reference area is the simple planform area of the loop, and when the length is greater than the diameter $l > d$ the reference area includes the inside of the tube as well. The slightly larger drag coefficient of the wire may be taken into account as a multiplier to the blunt cylinder drag coefficient.

Therefore the drag coefficient of a launch guide can be approximately calculated by

$$(C_{D\bullet})_{\text{parasitic}} = \max\{1.3 - 0.3 l/d, 1\} \cdot (C_{D\bullet})_{\text{stag}} \quad (3.95)$$

where $(C_{D\bullet})_{\text{stag}}$ is the stagnation pressure coefficient calculated in equation (B.2), and the reference area is

$$A_{\text{parasitic}} = \pi r_{\text{ext}}^2 - \pi r_{\text{int}}^2 \cdot \max\{1 - l/d, 0\}. \quad (3.96)$$

This approximation may also be used to estimate the drag of rail pins. A circular pin protruding from a wall has a drag coefficient of 0.80 [15, p. 5-8]. Therefore the drag of the pin is approximately equal to that of a lug with the same frontal area. The rail pins can be approximated in a natural manner as launch lugs with the same frontal area as the pin and a length equal to their diameter.

3.4.7 Axial drag coefficient

The total drag coefficient may be calculated by simply scaling the coefficients to a common reference area and adding them together:

$$C_{D_0} = \sum_T \frac{A_T}{A_{\text{ref}}} (C_{D\bullet})_T + (C_D)_{\text{friction}} \quad (3.97)$$

where the sum includes the pressure, base and parasitic drags. The friction drag was scaled to the reference area A_{ref} already in equation (3.85).

This yields the total drag coefficient at zero angle of attack. At an angle of attack the several phenomena begin to affect the drag. More frontal area is visible to the airflow, the pressure gradients along the body change and fin-tip vortices emerge. On the other hand, the drag force is no longer axial, so the axial drag force is less than the total drag force.

Based on experimental data an empirical formula was produced for calculating the axial drag coefficient at an angle of attack α from the zero-angle drag coefficient. The scaling function is a two-part polynomial function that starts from 1 at $\alpha = 0^\circ$, increases to 1.3 at $\alpha = 17^\circ$ and then decreases to zero at $\alpha = 90^\circ$; the derivative is also zero at these points. Since the majority of the simulated flight is at very small angles of attack, this approximation provides a sufficiently accurate estimate for the purposes of this thesis.

Chapter 4

Flight simulation

In this chapter the actual flight simulation is analyzed. First in Section 4.1 methods for simulating atmospheric conditions and wind are presented. Then in Section 4.2 the actual simulation procedure is developed.

4.1 Atmospheric properties

In order to calculate the aerodynamic forces acting on the rocket it is necessary to know the prevailing atmospheric conditions. Since the atmosphere is not constant with altitude, a model must be developed to account for the changes. Wind also plays an important role in the flight of a rocket, and therefore it is important to have a realistic wind model in use during the simulation.

4.1.1 Atmospheric model

The atmospheric model is responsible to estimating the atmospheric conditions at varying altitudes. The properties that are of most interest are the density of air ρ (which is a scaling parameter to the aerodynamic coefficients via the dynamic pressure $\frac{1}{2}\rho v^2$) and the speed of sound c (which affects the Mach number of the rocket, which in turn affects its aerodynamic properties).

These may in turn be calculated from the air pressure p and temperature T .

Several models exist that define standard atmospheric conditions as a function of altitude, including the International Standard Atmosphere (ISA) [27] and the U.S. Standard Atmosphere [28]. These two models yield identical temperature and pressure profiles for altitudes up to 32 km.

The models are based on the assumption that air follows the ideal gas law

$$\rho = \frac{Mp}{RT} \quad (4.1)$$

where M is the molecular mass of air and R is the ideal gas constant. From the equilibrium of hydrostatic forces the differential equation for pressure as a function of altitude z can be found as

$$dp = -g_0\rho dz = -g_0\frac{Mp}{RT} dz \quad (4.2)$$

where g_0 is the gravitational acceleration. If the temperature of air were to be assumed to be constant, this would yield an exponential diminishing of air pressure.

The ISA and U.S. Standard Atmospheres further specify a standard temperature and pressure at sea level and a temperature profile for the atmosphere. The temperature profile is given as eight temperatures for different altitudes, which are then linearly interpolated. The temperature profile and base pressures for the ISA model are presented in Table 4.1. These values along with equation (4.2) define the temperature/pressure profile as a function of altitude.

These models are totally static and do not take into account any local flight conditions. Many rocketeers may be interested in flight differences during summer and winter and what kind of effect air pressure has on the flight. These are also parameters that can easily be measured on site when launching rockets. On the other hand, it is generally hard to know a specific temperature profile for a specific day. Therefore the atmospheric model was extended to allow the user to specify the base conditions either at mean sea level or at the altitude of the launch site. These values are simply assigned to the first layer of the atmospheric model. Most model rockets do not exceed altitudes of a few kilometers, and therefore the flight conditions at the launch site will dominate the flight.

Table 4.1: Layers defined in the International Standard Atmosphere [29]

Layer	Altitude [†] m	Temperature °C	Lapse rate °C/km	Pressure Pa
0	0	+15.0	-6.5	101 325
1	11 000	-56.5	+0.0	22 632
2	20 000	-56.5	+1.0	5 474.9
3	32 000	-44.5	+2.8	868.02
4	47 000	-2.5	+0.0	110.91
5	51 000	-2.5	-2.8	66.939
6	71 000	-58.5	-2.0	3.9564
7	84 852	-86.2		0.3734

[†] Altitude is the geopotential height which does not account for the diminution of gravity at high altitudes.

One parameter that also has an effect on air density and the speed of sound is humidity. The standard models do not include any definition of humidity as a function of altitude. Furthermore, the effect of humidity on air density and the speed of sound is marginal. The difference in air density and the speed of sound between completely dry air and saturated air at standard conditions are both less than 1%. Therefore the effect of humidity has been ignored.

4.1.2 Wind modeling

Wind plays a critical role in the flight of model rockets. As has been seen, large angles of attack may cause rockets to lose a significant amount of stability and even go unstable. Over-stable rockets may weathercock and turn into the wind. In a perfectly static atmosphere a rocket would, in principle, fly its entire flight directly upwards at zero angle of attack. Therefore, the effect of wind must be taken into account in a full rocket simulation.

Most model rocketeers, however, do not have access to a full wind profile of the area they are launching in. Different layers of air may have different wind velocities and directions. Modeling such complex patterns is beyond the scope of this project. Therefore, the goal is to produce a realistic wind model that can be specified with only a few parameters understandable to

the user and that covers altitudes of most rocket flights. Extensions to allow for multiple air layers may be added in the future.

In addition to a constant average velocity, wind always has some degree of turbulence in it. The effect of turbulence can be modeled by summing the steady flow of air and a random, zero-mean turbulence velocity. Two central aspects of the turbulence velocity are the amplitude of the variation and the frequencies at which they occur. Therefore a reasonable turbulence model is achieved by a random process that produces a sequence with a similar distribution and frequency spectrum as that of real wind.

Several models of the spectrum of wind turbulence at specific altitudes exist. Two commonly used such spectra are the *Kaimal* and *von Kármán* wind turbulence spectra [30, p. 23]:

$$\text{Kaimal: } \frac{S_u(f)}{\sigma_u^2} = \frac{4L_{1u}/U}{(1 + 6fL_{1u}/U)^{5/3}} \quad (4.3)$$

$$\text{von Kármán: } \frac{S_u(f)}{\sigma_u^2} = \frac{4L_{2u}/U}{(1 + 70.8(fL_{2u}/U)^2)^{5/6}} \quad (4.4)$$

Here $S_u(f)$ is the spectral density function of the turbulence velocity and f the turbulence frequency, σ_u the standard deviation of the turbulence velocity, L_{1u} and L_{2u} length parameters and U the average wind speed.

Both models approach the asymptotic limit $S_u(f)/\sigma_u^2 \sim f^{-5/3}$ quite fast. Above frequencies of 0.5 Hz the difference between equation (4.3) and the same equation without the term 1 in the denominator is less than 4%. Since the time scale of a model rocket's flight is quite short, the effect of extremely low frequencies can be ignored. Therefore turbulence may reasonably well be modelled by utilizing *pink noise* that has a spectrum of $1/f^\alpha$ with $\alpha = 5/3$. True pink noise has the additional useful property of being scale-invariant. This means that a stream of pink noise samples may be generated and assumed to be at any sampling rate while maintaining their spectral properties.

Discrete samples of pink noise with spectrum $1/f^\alpha$ can be generated by applying a suitable digital filter to *white noise*, which is simply uncorrelated pseudorandom numbers. One such filter is the infinite impulse response (IIR) filter presented by Kasdin [31]:

$$x_n = w_n - a_1x_{n-1} - a_2x_{n-2} - a_3x_{n-3} - \dots \quad (4.5)$$

where x_i are the generated samples, w_n is a generated white random number and the coefficients are computed using

$$\begin{aligned} a_0 &= 1 \\ a_k &= \left(k - 1 - \frac{\alpha}{2}\right) \frac{a_{k-1}}{k}. \end{aligned} \quad (4.6)$$

The infinite sum may be truncated with a suitable number of terms. In the context of IIR filters these terms are called *poles*. Experimentation showed that already 1–3 poles provides a reasonably accurate frequency spectrum in the high frequency range.

One problem in using pink noise as a turbulence velocity model is that the power spectrum of pure pink noise goes to infinity at very low frequencies. This means that a long sequence of random values may deviate significantly from zero. However, when using the truncated IIR filter of equation (4.5), the spectrum density becomes constant below a certain limiting frequency, dependent on the number of poles used. By adjusting the number of poles used, the limiting frequency can be adjusted to a value suitable for model rocket flight. Specifically, the number of poles must be selected such that the limiting frequency is suitable at the chosen sampling rate.

It is also desirable that the simulation resolution does not affect the wind conditions. For example, a simulation with a time step of 10 ms should experience the same wind conditions as a simulation with a time step of 5 ms. This is achieved by selecting a constant turbulence generation frequency and interpolating between the generated points when necessary. The fixed frequency was chosen at 20 Hz, which can still simulate fluctuations at a time scale of 0.1 seconds.

The effect of the number of poles is depicted in Figure 4.1, where two pink noise sequences were generated from the same random number source with two-pole and ten-pole IIR filters. A small number of poles generates values strongly centered on zero, while a larger number of poles introduces more low frequency variability. Since the free-flight time of a typical model rocket is of the order of 5–30 seconds, it is desirable that the maximum gust length during the flight is substantially shorter than this. Therefore the pink noise generator used by the wind model was chosen to contain only two poles, which has a limiting frequency of approximately 0.3 Hz when sampled at 20 Hz. This means that gusts of wind longer than 3–5 seconds will be rare in the simulated turbulence, which is a suitable gust length for modeling typical model rocket flight. Figure 4.2 depicts the resulting pink noise spectrum of

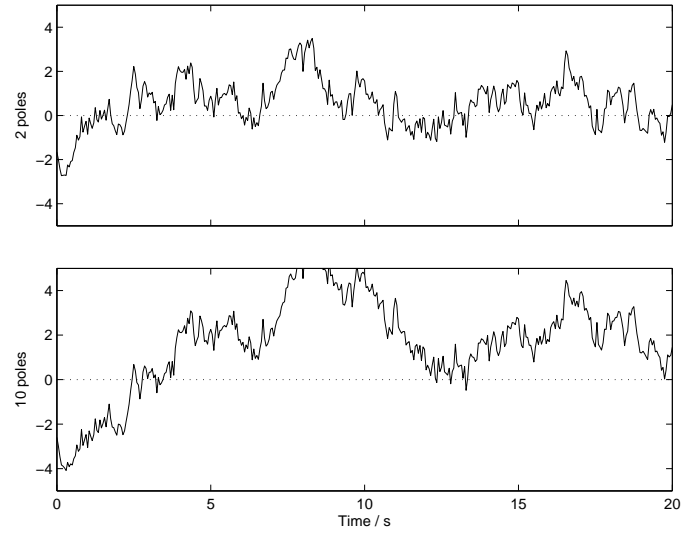


Figure 4.1: The effect of the number of IIR filter poles on two 20 second samples of generated turbulence, normalized so that the two-pole sequence has standard deviation one.

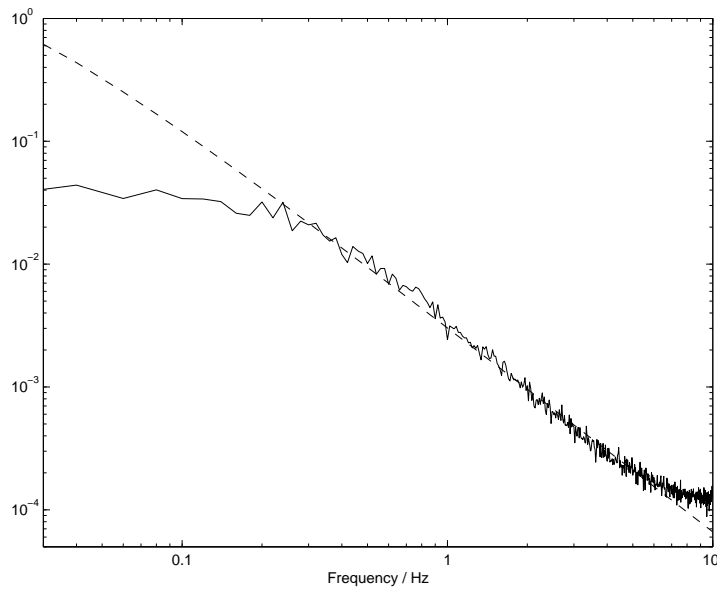


Figure 4.2: The average power spectrum of 100 turbulence simulations using a two-pole IIR filter (solid) and the Kaimal turbulence spectrum (dashed); vertical axis arbitrary.

the two-pole IIR filter and the Kaimal spectrum of equation (4.3) scaled to match each other.

To simplify the model, the average wind speed is assumed to be constant with altitude and in a constant direction. This allows specifying the model parameters using just the average wind speed and its standard deviation. An alternative parameter for specifying the turbulence amplitude is the *turbulence intensity*, which is the percentage that the standard deviation is of the average wind velocity,

$$I_u = \frac{\sigma_u}{U}. \quad (4.7)$$

Wind farm load design standards typically specify turbulence intensities around 10...20% [30, p. 22]. It is assumed that these intensities are at the top of the range of conditions in which model rockets are typically flown.

Overall, the process to generate the wind velocity as a function of time from the average wind velocity U and standard deviation σ_u can be summarized in the following steps:

1. Generate a pink noise sample x_n from a Gaussian white noise sample w_n using equations (4.5) and (4.6) with two memory terms included.
2. Scale the sample to a standard deviation one. This is performed by dividing the value by a previously calculated standard deviation of a long, unscaled pink noise sequence (2.252 for the two-pole IIR filter).
3. The wind velocity at time $n \cdot \Delta t$ ($\Delta t = 0.05$ s) is $U_n = U + \sigma_u x_n$. Velocities in between are interpolated.

4.2 Modeling rocket flight

Modeling of rocket flight is based on Newton's laws. The basic forces acting upon a rocket are gravity, thrust from the motors and aerodynamic forces and moments. These forces and moments are calculated and integrated numerically to yield a simulation over a full flight.

Since most model rockets fly at a maximum a few kilometers high, the curvature of the Earth is not taken into account. Assuming a flat Earth allows

us to use simple Cartesian coordinates to represent the position and altitude of the rocket. As a consequence, the coriolis effect when flying long distances north or south is not simulated either.

4.2.1 Coordinates and orientation

During a rocket's flight many quantities, such as the aerodynamical forces and thrust from the motors, are relative to the rocket itself, while others, such as the position and gravitational force, are more naturally described relative to the launch site. Therefore two sets of coordinates are defined, the *rocket coordinates*, which are the same as used in Chapter 3, and *world coordinates*, which is a fixed coordinate system with the origin at the position of launch.

The position and velocity of a rocket are most naturally maintained as Cartesian world coordinates. Following normal conventions, the xy -plane is selected to be parallel to the ground and the z -axis is chosen to point upwards. In flight dynamics of aircraft the z -axis often points towards the earth, but in the case of rockets it is natural to have the rocket's altitude as the z -coordinate.

Since the wind is assumed to be unidirectional and the Coriolis effect is ignored, it may be assumed that the wind is directed along the x -axis. The angle of the launch rod may then be positioned relative to the direction of the wind without any loss of generality.

Determining the orientation of a rocket is more complicated. A natural choice for defining the orientation would be to use the spherical coordinate zenith and azimuth angles (θ, ϕ) and an additional roll angle parameter. Another choice common in aviation is to use *Euler angles* [32]. However, both of these systems have notable shortcomings. Both systems have singularity points, in which the value of some parameter is ambiguous. With spherical coordinates, this is the direction of the z -axis, in which case the azimuth angle ϕ has no effect on the position. Rotations that occur near these points must often be handled as special cases. Furthermore, rotations in spherical coordinate systems contain complex trigonometric formulae which are prone to programming errors.

The solution to the singularity problem is to introduce an extra parameter

and an additional constraint to the system. For example, the direction of a rocket could be defined by a three-dimensional unit vector (x, y, z) instead of just the zenith and azimuth angles. The additional constraint is that the vector must be of unit length. This kind of representation has no singularity points which would require special consideration.

Furthermore, Euler's rotation theorem states that a rigid body can be rotated from any orientation to any other orientation by a single rotation around a specific axis [33]. Therefore instead of defining quantities that define the orientation of the rocket we can define a three-dimensional rotation that rotates the rocket from a known reference orientation to the current orientation. This has the additional advantage that the same rotation and its inverse can be used to transform any vector between world coordinates and rocket coordinates.

A simple and efficient way of describing the 3D rotation is by using *unit quaternions*. Each unit quaternion corresponds to a unique 3D rotation, and they are remarkably simple to combine and use. The following section will present a brief overview of the properties of quaternions.

The fixed reference orientation of the rocket defines the rocket pointing towards the positive z -axis in world coordinates and an arbitrary but fixed roll angle. The orientation of the rocket is then stored as a unit quaternion that rotates the rocket from this reference orientation to its current orientation. This rotation can also be used to transform vectors from world coordinates to rocket coordinates and its inverse from rocket coordinates to world coordinates. (Note that the rocket's initial orientation on the launch pad may already be different than its reference orientation if the launch rod is not completely vertical.)

4.2.2 Quaternions

Quaternions are an extension of complex numbers into four dimensions. The usefulness of quaternions arises from their use in spatial rotations. Similar to the way multiplication with a complex number of unit length $e^{i\phi}$ corresponds to a rotation of angle ϕ around the origin on the complex plane, multiplication with unit quaternions correspond to specific 3D rotations around an axis. A more thorough review of quaternions and their use in spatial rotations is available in Wikipedia [34].

The typical notation of quaternions resembles the addition of a scalar and a vector:

$$q = w + x\mathbf{i} + y\mathbf{j} + z\mathbf{k} = w + \mathbf{v} \quad (4.8)$$

Addition of quaternions and multiplication with a scalar operate as expected. However, the multiplication of two quaternions is non-commutative (in general $ab \neq ba$) and follows the rules

$$\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1. \quad (4.9)$$

As a corollary, the following equations hold:

$$\begin{aligned} \mathbf{ij} &= \mathbf{k} & \mathbf{ji} &= -\mathbf{k} \\ \mathbf{jk} &= \mathbf{i} & \mathbf{kj} &= -\mathbf{i} \\ \mathbf{ki} &= \mathbf{j} & \mathbf{ik} &= -\mathbf{j} \end{aligned} \quad (4.10)$$

The general multiplication of two quaternions becomes

$$\begin{aligned} (a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k})(w + x\mathbf{i} + y\mathbf{j} + z\mathbf{k}) &= (aw - bx - cy - dz) \\ &+ (ax + bw + cz - dy) \mathbf{i} \\ &+ (ay - bz + cw + dx) \mathbf{j} \\ &+ (az + by - cx + dw) \mathbf{k} \end{aligned} \quad (4.11)$$

while the norm of a quaternion is defined in the normal manner

$$|q| = \sqrt{w^2 + x^2 + y^2 + z^2}. \quad (4.12)$$

The usefulness of quaternions becomes evident when we consider a rotation around a vector \mathbf{u} , $|\mathbf{u}| = 1$ by an angle ϕ . Let

$$q = \cos \frac{\phi}{2} + \mathbf{u} \sin \frac{\phi}{2}. \quad (4.13)$$

Now the previously mentioned rotation of a three-dimensional vector \mathbf{v} defined by \mathbf{i} , \mathbf{j} and \mathbf{k} is equivalent to the quaternion product

$$\mathbf{v} \mapsto q\mathbf{v}q^{-1}. \quad (4.14)$$

Similarly, the inverse rotation is equivalent to the transformation

$$\mathbf{v} \mapsto q^{-1}\mathbf{v}q. \quad (4.15)$$

The problem simplifies even further, since for unit quaternions

$$q^{-1} = (w + x\mathbf{i} + y\mathbf{j} + z\mathbf{k})^{-1} = w - x\mathbf{i} - y\mathbf{j} - z\mathbf{k}. \quad (4.16)$$

Vectors can therefore be considered quaternions with no scalar component and their rotation is equivalent to the left- and right-sided multiplication with unit quaternions, requiring a total of 24 floating-point multiplications. Even if this does not make the rotations more efficient, it simplifies the trigonometry considerably and therefore helps reduce programming errors.

4.2.3 Mass and moment of inertia calculations

Converting the forces and moments into linear and angular acceleration requires knowledge of the rocket's mass and moments of inertia. The mass of a component can be easily calculated from its volume and density. Due to the highly symmetrical nature of rockets, the rocket centerline is commonly a principal axis for the moments of inertia. Furthermore, the moments of inertia around the in the y - and z -axes are very close to one another. Therefore as a simplification only two moments of inertia are calculated, the longitudinal and rotational moment of inertia. These can be easily calculated for each component using standard formulae [35] and combined to yield the moments of the entire rocket.

This is a good way of calculating the mass, CG and inertia of a rocket during the design phase. However, actual rocket components often have a slightly different density or additional sources of mass such as glue attached to them. These cannot be effectively modeled by the simulator, since it would be extremely tedious to define all these properties. Instead, some properties of the components can be overridden to utilize measured values.

Two properties that can very easily be measured are the mass and CG position of a component. Measuring the moments of inertia is a much harder task. Therefore the moments of inertia are still computed automatically, but are scaled by the overridden measurement values.

If the mass of a component is overridden by a measured value, the moments of inertia are scaled linearly according to the mass. This assumes that the extra weight is distributed evenly along the component. If the CG position is overridden, there is no knowledge where the extra weight is at. Therefore as a best guess the moments of inertia are updated by shifting the moment axis according to the parallel axis theorem.

As the components are computed individually and then combined, the overriding can take place either for individual components or larger combinations. It is especially useful is to override the mass and/or CG position of the entire rocket. This allows constructing a rocket from components whose masses are not precisely known and afterwards scaling the moments of inertia to closely match true values.

4.2.4 Flight simulation

The process of simulating rocket flight can be broken down into the following steps:

0. Initialize the rocket in a known position and orientation at time $t = 0$.
1. Compute the local wind velocity and other atmospheric conditions.
2. Compute the current airspeed, angle of attack, lateral wind direction and other flight parameters.
3. Compute the aerodynamic forces and moments affecting the rocket.
4. Compute the effect of motor thrust and gravity.
5. Compute the mass and moments of inertia of the rocket and from these the linear and rotational acceleration of the rocket.
6. Numerically integrate the acceleration to the rocket's position and orientation during a time step Δt and update the current time $t \mapsto t + \Delta t$.

Steps 1–6 are repeated until an end criteria is met, typically until the rocket has landed.

The computation of the atmospheric properties and instantaneous wind velocity were discussed in Section 4.1. The local wind velocity is added to the rocket velocity to get the airspeed velocity of the rocket. By inverse rotation this quantity is obtained in rocket coordinates, from which the angle of attack and other flight parameters can be computed.

After the instantaneous flight parameters are known, the aerodynamic forces can be computed as discussed in Chapter 3. The computed forces are in the rocket coordinates, and can be converted to world coordinates by applying the orientation rotation. The thrust from the motors is similarly calculated from the thrust curves and converted to world coordinates, while the direction of gravity is already in world coordinates. When all of the the forces and moments acting upon the rocket are known, the linear and rotational accelerations can be calculated using the mass and moments of inertia discussed in Section 4.2.3.

The numerical integration is performed using the Runge-Kutta 4 (RK4) integration method. In order to simulate the differential equations

$$\begin{aligned}x''(t) &= a(t) \\ \phi''(t) &= \alpha(t)\end{aligned}\tag{4.17}$$

the equation is first divided into first-order equations using the substitutions $v(t) = x'(t)$ and $\omega(t) = \phi'(t)$:

$$\begin{aligned}v'(t) &= a(t) \\ x'(t) &= v(t) \\ \omega'(t) &= \alpha(t) \\ \phi'(t) &= \omega(t)\end{aligned}\tag{4.18}$$

For brevity, this is presented in the first order representation

$$y' = f(y, t)\tag{4.19}$$

where y is a vector function containing the position and orientation of the rocket.

Next the right-hand side is evaluated at four positions, dependent on the previous evaluations:

$$\begin{aligned}k_1 &= f(y_0, t_0) \\ k_2 &= f(y_0 + k_1 \frac{\Delta t}{2}, t_0 + \frac{\Delta t}{2}) \\ k_3 &= f(y_0 + k_2 \frac{\Delta t}{2}, t_0 + \frac{\Delta t}{2}) \\ k_4 &= f(y_0 + k_3 \Delta t, t_0 + \Delta t)\end{aligned}\tag{4.20}$$

Finally, the result is a weighted sum of these values:

$$y_1 = y_0 + \frac{1}{6} (k_1 + 2k_2 + 2k_3 + k_4) \Delta t\tag{4.21}$$

$$t_1 = t_0 + \Delta t\tag{4.22}$$

Computing the values $k_1 \dots k_4$ involves performing steps 1–5 four times per simulation iteration, but results in significantly better simulation precision. The method is a fourth-order integration method, meaning that the error incurred during one simulation step is of the order $O(\Delta t^5)$ and of the total simulation $O(\Delta t^4)$. This is a considerable improvement over, for example, simple Euler integration, which has a total error of the order $O(\Delta t)$. Halving

the time step in an Euler integration only halves the total error, but reduces the error of a RK4 simulation 16-fold.

The example above used a total rotation vector ϕ to contain the orientation of the rocket. Instead, this is replaced by the rotation quaternion, which can be utilized directly as a transformation between world and rocket coordinates. Instead of updating the total rotation vector,

$$\phi_1 = \phi_0 + \omega \Delta t, \quad (4.23)$$

the orientation quaternion o is updated by the same amount by

$$o_1 = \left(\cos(|\omega| \Delta t) + \hat{\omega} \sin(|\omega| \Delta t) \right) \cdot o_0. \quad (4.24)$$

The first term is simply the unit quaternion corresponding to the 3D rotation $\omega \Delta t$ as in equation (4.13). It is applied to the previous value o_0 by multiplying the quaternion from the left. This update is performed both during the calculation of $k_2 \dots k_4$ and when computing the final step result. Finally, in order to improve numerical stability, the quaternion is normalized to unit length.

Since most of a rocket's flight occurs in a straight line, rather large time steps can be utilized. However, the rocket may encounter occasional oscillation, which may affect its flight notably. Therefore the time step utilized is dynamically reduced in cases where the angular velocity or angular acceleration exceeds a predefined limit. This allows utilizing reasonably large time steps for most of the flight, while maintaining the accuracy during oscillation.

4.2.5 Recovery simulation

All model rockets must have some recovery system for safe landing. This is typically done either using a parachute or a streamer. When a parachute is deployed the rocket typically splits in half, and it is no longer practical to compute the orientation of the rocket. Therefore at this point the simulation changes to a simpler, three degree of freedom simulation, where only the position of the rocket is computed.

The entire drag coefficient of the rocket is assumed to come from the deployed recovery devices. For parachutes the drag coefficient is by default 0.8 [15,

p. 13-23] with the reference area being the area of the parachute. The user can also define their own drag coefficient.

The drag coefficient of streamers depend on the material, width and length of the streamer. The drag coefficient and optimization of streamers has been an item of much interest within the rocketry community, with competitions being held on streamer descent time durations [46]. In order to estimate the drag coefficient of streamers, a series of experiments were performed using the $40 \times 40 \times 120$ cm wind tunnel of Pollux [36]. The experiments were performed using various materials, widths and lengths of streamers and at different wind speeds. From these results an empirical formula was devised that estimates the drag coefficient of streamers. The experimental results and the derivation of the empirical formula are presented in Appendix C. Validation performed with an independent set of measurements indicates that the drag coefficient is estimated with an accuracy of about 20%, which translates to a descent velocity accuracy within 15% of the true value.

4.2.6 Simulation events

Numerous different events may cause actions to be taken during a rocket's flight. For example in high-power rockets the burnout or ignition charge of the first stage's motor may trigger the ignition of a second stage motor. Similarly a flight computer may deploy a small drogue parachute when apogee is detected and the main parachute is deployed later at a predefined lower altitude. To accommodate different configurations a simulation event system is used, where events may cause other events to be triggered.

Table 4.2 lists the available simulation events and which of them can be used to trigger motor ignition or recovery device deployment. Each trigger event may additionally include a delay time. For example, one motor may be configured to ignite at launch and a second motor to ignite using a timer at 5 seconds after launch. Alternatively, a short delay of 0.5–1 seconds may be used to simulate the delay of an ejection charge igniting the upper stage motors.

The flight events are also stored along with the simulated flight data for later analysis. They are also available to the simulation listeners, described in Section 5.1.3, to act upon specific conditions.

Table 4.2: Simulation events and the actions they may trigger (motor ignition or recovery device deployment).

Event description	Triggers
Rocket launch at $t = 0$	Ignition, recovery
Motor ignition	None
Motor burnout	Ignition
Motor ejection charge	Ignition, recovery
Launch rod cleared	None
Apogee detected	Recovery
Change in altitude	Recovery
Touchdown after flight	None
Deployment of a recovery device	None
End of simulation	None

Chapter 5

The OpenRocket simulation software

The flight simulation described in the previous chapters was implemented in the *OpenRocket* simulation software [7]. The software was written entirely in Java for maximum portability between different operating systems. A significant amount of effort was put into making the graphical user interface (UI) robust, intuitive and easy to use. As of the first public release the source code contained over 300 classes and over 47 000 lines of code (including comments and blank lines).

The software was released under the copyleft GNU General Public License (GPL) [37], allowing everybody access to the source code and permitting use and modification for any purposes. The only major restriction placed is that if a modified version is distributed, the source code for the modifications must also be available under the GNU GPL. This ensures that the program stays free and Open Source; a company cannot simply take the code, enhance it and start selling it as their own without contributing anything back to the community.

In this section the basic architectural designs are discussed and the main features of the UI are explained.

5.1 Architectural design

The software has been split into various components within their own Java packages. This enables use of the components without needing the other components. For example, all of the UI code is within the `gui` package, and the simulation system can be used independently of it.

The rocket structure is composed of components, each with their own class defined within the package `rocketcomponent`. This provides the base for defining a rocket. The components are described in more detail in Section 5.1.1. The simulation code is separated from the aerodynamic calculation code in the `simulation` and `aerodynamics` packages, respectively; these are discussed in Section 5.1.2.

The package naming convention recommended in the Java Language Specification is followed [38]. Therefore all package names discussed herein are relative to the package `net.sf.openrocket`. For example, the rocket components are defined in the package `net.sf.openrocket.rocketcomponent`.

5.1.1 Rocket components

The structure of the rocket is split up into *components*, for example nose cones, body tubes and components within the rocket body. Each component type is defined as a subclass of the abstract `RocketComponent` class. Each component can contain zero or more components attached to it, which creates a tree structure containing the entire rocket design. The base component of every design is a `Rocket` object, which holds one or more `Stage` objects. These represent the stages of the rocket and hold all of the physical components.

Inheritance has been highly used in the class hierarchy of the components in order to combine common features of various components. There are also some abstract classes, such as `BodyComponent`, whose sole purpose currently is to allow future extensions. The complete component class hierarchy is presented as a UML diagram in Figure 5.1, and a more detailed description of each class is presented in Table 5.1.

Additionally three interfaces are defined for the components, `MotorMount`,

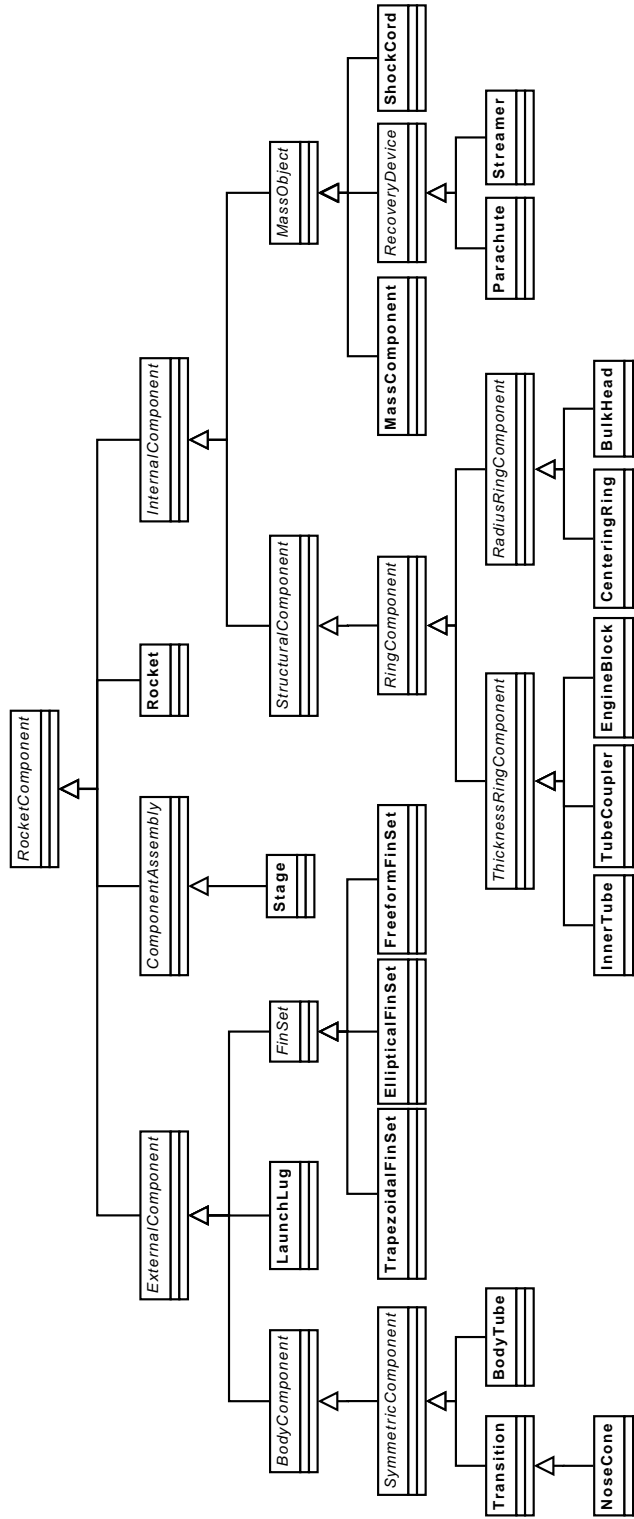


Figure 5.1: A UML diagram of the rocket component classes. Abstract classes are shown in italics.

Table 5.1: Descriptions of the rocket component classes and their functionality. Abstract classes are shown in italics.

Component class	Description
<i>RocketComponent</i> :	The base class of all components, including common features such as child component handling.
Rocket:	The base component of a rocket design, provides change event notifications.
<i>ComponentAssembly</i> :	A base component for an assembly of external components. This could in the future be extended to allow multiple rocket bodies next to each other.
Stage:	A separable stage of the rocket.
<i>ExternalComponent</i> :	An external component that has an effect on the aerodynamics of the rocket.
<i>BodyComponent</i> :	A portion of the main rocket body, defined in cylindrical coordinates by $r = f(x, \theta)$.
<i>SymmetricComponent</i> :	An axisymmetrical body component.
Transition:	A symmetrical transition (shoulder or boattail).
NoseCone:	A transition with the initial radius zero.
BodyTube:	A cylindrical body tube. Can be used as a motor mount.
<i>FinSet</i> :	A set of one or more fins.
TrapezoidalFinSet:	A set of trapezoidal fins.
EllipticalFinSet:	A set of elliptical fins.
FreeformFinSet:	A set of free-form fins.
LaunchLug:	A launch lug or rail pin.
<i>InternalComponent</i> :	An internal component that may affect the mass of the rocket but not its aerodynamics.
<i>StructuralComponent</i> :	A structural internal component, with specific shape and density.
<i>RingComponent</i> :	A hollow cylindrical component.
<i>ThicknessRingComponent</i> :	A component defined by an outer radius and shell thickness.
InnerTube:	An inner tube. Can be used as a motor mount and can be clustered.
TubeCoupler:	A coupler tube.
EngineBlock:	An engine block.
<i>RadiusRingComponent</i> :	A component defined by an inner and outer radius.
CenteringRing:	A ring for centering components.
Bulkhead:	A solid bulkhead (inner radius zero).
<i>MassObject</i> :	An internal component shaped approximately like a solid cylinder and with a specific mass.
MassComponent:	A generic component with specific mass, for example payload.
<i>RecoveryDevice</i> :	A recovery device.
Parachute:	A parachute.
Streamer:	A streamer.
ShockCord:	A shock cord with a specified material and length.

Clusterable and **RadialParent**. Components implementing the **MotorMount** interface, currently **BodyTube** and **InnerTube**, can function as motor mounts and have motors loaded in them. The **Clusterable** interface signifies that the component can be clustered in various configurations. Currently only the **InnerTube** component can be clustered. Components and motors that are attached to a clustered inner tube are automatically replicated to all tubes within the cluster. The **RadialParent** interface allows inner components to automatically identify their correct inner and outer radii based on their parent and sibling components. For example, a coupler tube can automatically detect its radius based on the inner radius of the parent body tube.

Since the software functionality is divided into different packages, all component similarities cannot be directly be exploited through inheritance. For example, the method of drawing a nose cone shape belongs to the `gui.rocketfigure` package, however, it can share the same code that draws a transition. For these purposes, reflective programming is used extensively. The code for drawing both nose cones and transitions is provided in the class `gui.rocketfigure.SymmetricComponentShapes`, while the simpler body tube is drawn by the class `BodyTubeShapes`. The correct class is derived and instantiated dynamically based on the component class. This allows easily sharing functionality common to different components while still having loose coupling between the rocket structure, presentation, computation and storage methods.

5.1.2 Aerodynamic calculators and simulators

One of the key aspects in the design of the simulation implementation was extensibility. Therefore all aerodynamic calculation code is separated in the package `aerodynamics` and all simulation code is in the package `simulator`.

The basis for aerodynamic calculations is the abstract class `AerodynamicCalculator`, while the simulators are subclasses of the `FlightSimulator` class. This allows adding new implementations of the aerodynamic calculators and simulators independently. For example, a simulator using Euler integration was written in the early stages of development, and later replaced by the Runge-Kutta 4 simulator. Similarly, a different method of calculating the aerodynamic forces, such as CFD, could be implemented and used by the existing simulators.

Similar abstraction has been performed for the atmospheric temperature and pressure model by the abstract class `AtmosphericModel` and different rocket motor types by the `Motor` class, among others.

5.1.3 Simulation listeners

Simulation listeners are pieces of code that can dynamically be configured to listen to and interact with a simulation while it is running. The listeners are called after each simulation step, at each simulation event and both when the flight conditions and aerodynamic forces have been calculated. The listeners may simply gather flight data for use outside the simulation or modify the rocket or simulation during the flight. This allows great potential for extensibility both internally and externally.

Listeners are used internally for various purposes such as retrieving flight progress information when the user is running simulations and cancelling the simulations when necessary. Implementing such functionality otherwise would have required a lot of special case handling directly within the simulation code.

Listeners can also be used to modify the simulation or the rocket during its flight. The successor project of Haisunäätä will include an active roll stabilization system, where a flight computer will measure the roll rate using two magnetometers and use a PID controller to adjust two auxiliary fins to cancel out the roll inevitably produced by imperfections in the main fins. A simulation listener was written that first simulated the PID controller purely in Java, which modified the cant angle of the auxiliary fins during the simulation. Later a similar listener interfaced the external flight computer directly using a serial data link. The listener fed the flight data to the controller which computed and reported the control actions back to the simulator. This system helped identify and fix numerous bugs in the flight computer software, which would have otherwise been nearly impossible to fully test. It is expected that the simulation listeners will be an invaluable tool for more ambitious model rocket enthusiasts.

A listener is produced by implementing the `SimulationListener` interface or by extending the `AbstractSimulationListener` class. The UI includes the option of defining custom simulation listeners to be utilized during flight simulation.

5.1.4 Warnings

The aerodynamic calculations and simulations are based on certain assumptions and simplifications, such as a low angle of attack and a smooth, continuous rocket body. The rocket component architecture makes it possible to create designs that break these assumptions. Instead of limiting the options of the design, the aerodynamic calculator and simulator can produce warnings about such issues. These warnings are presented to the user during the design of the rocket or after simulations. It is then left up to the user to judge whether such violations are significant enough to cast doubt to the validity of the results.

5.1.5 File format

An XML-based file format was devised for storing the rocket designs and simulations. The use of XML allows using many existing tools for reading and writing the files, allows easy extensibility and makes the files human-readable. The user has the option of including all simulated data in the file, storing the data at specific time intervals or storing only the simulation launch conditions. To reduce the file size, the files can additionally be compressed using the standard GZIP compression algorithm [39]. The files are compressed and uncompressed automatically by the software. The file extension `.ORK` was chosen for the design files, an abbreviation of the software name that at the same time had no previous uses as a file extension.

5.2 User interface design

The user interface was designed with the intent of being robust but yet easy to use even for inexperienced users. The main window, depicted in Figure 5.2(a) with the design of the original Haisunäättä rocket, consists of a schematic drawing of the rocket, the tree structure of the rocket components and buttons for adding new components to the structure. Components can be selected or edited by clicking and double-clicking either the tree view or the component in the schematic diagram. The selected components are drawn in bold to give a visual clue to the position of the component.

The schematic drawing can be viewed either from the side or from the rear, can be zoomed in and out and rotated along the centerline. The schematic diagram view also presents basic information about the rocket, such as the design name, length, maximum diameter, mass and possible warnings about the design. It also calculates the CG and CP positions continuously during design and shows them both numerically and on the diagram. Additionally, a simulation is automatically run in the background after each modification and the main results are presented in the lower left corner of the diagram. Many users are interested in the maximum altitude or velocity of the rocket, and this allows an immediate feedback on the effect of the changes they are making to the design. The flight information typically takes less than a second to update.

The upper part of the main window can also be changed to view simulation results, Figure 5.2(b). Many simulations can be added with different launch conditions and motor configurations to investigate their effects. Each simulation has a row which presents the basic information about the simulation. The first column gives an immediate visual clue to the status of the simulation; a gray ball indicates that the simulation has not been run yet, green indicates an up-to-date simulation, red indicates that the design has been changed after the simulation was run and yellow indicates that the simulation information was loaded from a file, but that the file states it to be up-to-date. The simulations can be run one or several at a time. The software automatically utilizes several threads when running multiple simulations on a multi-CPU computer to utilize the full processing capacity.

Figure 5.3 shows two dialogs that are used to modify and analyze the designs. The components are edited using a small dialog window that allows the user to either fill in the exact numerical values specifying the shape of the component or use sliders to modify them. The user can change the units by clicking on them, or set default values from the preferences. Different tabs allow control over the mass and CG override settings, figure color options, motor mount and cluster options and more. The Component analysis dialog shown in the figure can be used to analyze the effect of individual components on the total stability, drag and roll characteristics of the rocket.

Similarly, the launch conditions and simulator options can be edited in the corresponding dialog. The simulator options also allow the user to define custom simulation listeners to use during the simulations. The simulation edit dialog is also used for later data analysis. The simulated data can be plotted in a variety of ways as shown in Figure 5.4. The user can use

predefined plot settings or define their own. Up to 15 different variables out of the 47 quantities computed can be plotted at a time. The variable on the horizontal axis can be freely selected, and the other variables can be plotted on one of two vertical axis, on either side of the plot. The user can either specify whether a variable should plot on the left or right axis or let the software decide the most suitable axis. Typical plots include the altitude, vertical velocity and acceleration of the rocket with time or the drag coefficient as a function of the Mach number.

5.3 Future enhancements

Numerous features have been planned and taken into account during the design of the software. Below are listed a few of the planned features and how they have been taken into account:

Alternative aerodynamic calculators. For example CFD could be used to calculate the aerodynamic properties, allowing even better simulation accuracy. The calculators have been abstracted by the `AerodynamicCalculator` class so they can easily be interchanged.

Alternative simulators. These could take into account for example the curvature of the Earth and include the Coriolis effect. New simulators can be created by extending the abstract `FlightSimulator` class.

Export and import of flight data. The simulated data could be exported for further analysis as comma separated values (CSV). Similarly, experimental data could be imported either from files or directly from altimeters. Support for imported data already exists in the core functionalities.

Importing database files. The motor database is easily extendable to read external thrust curves. Also data of commercially available rocket components could be imported and available in the component edit dialog.

Further UI enhancements. These could include for example a 3D view of the rocket, an animation of the rocket flight, a “wizard” dialog for easily creating new designs, *etc.*

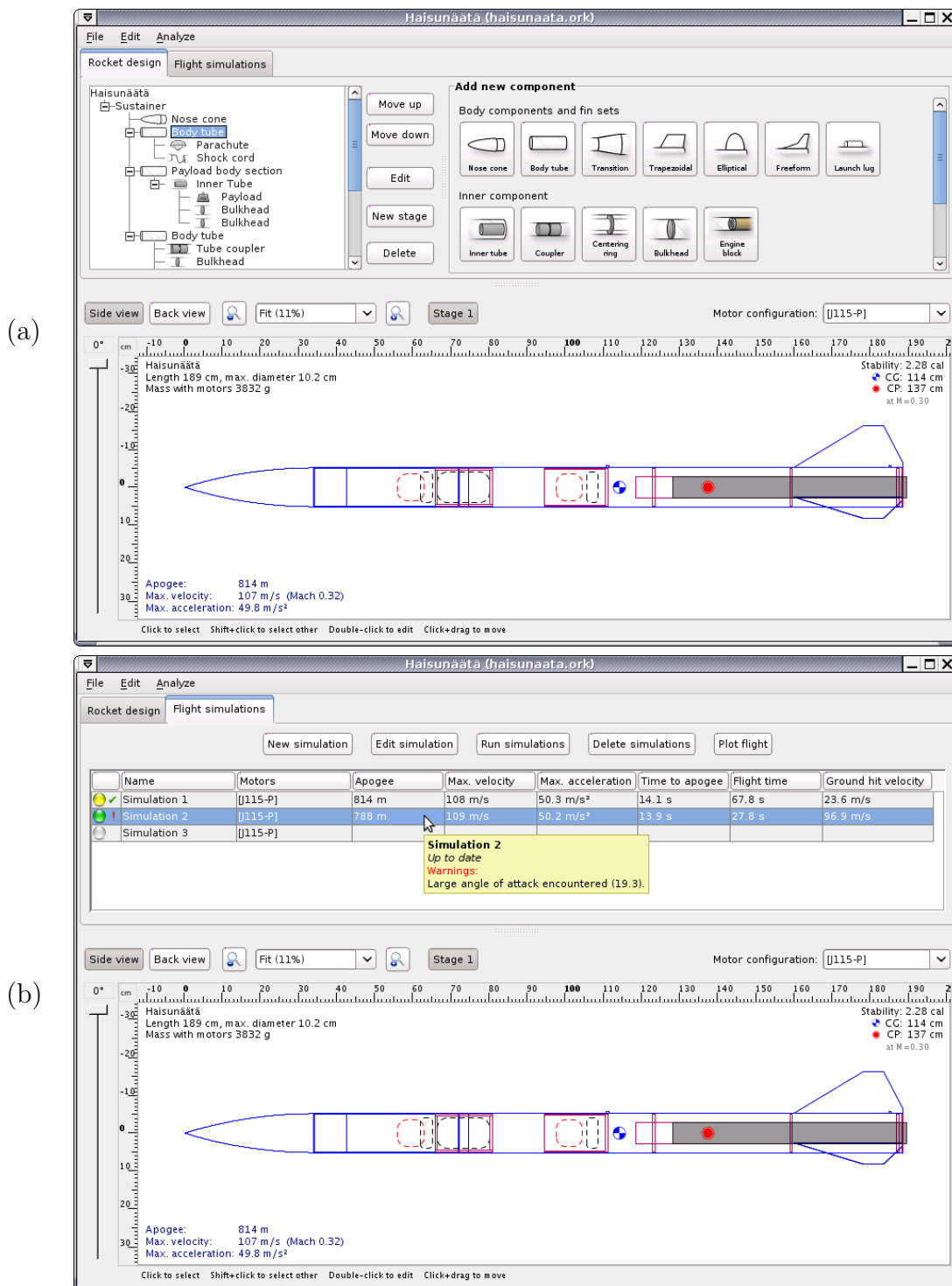


Figure 5.2: The main design window of OpenRocket; (a) the design view and (b) the simulation view.

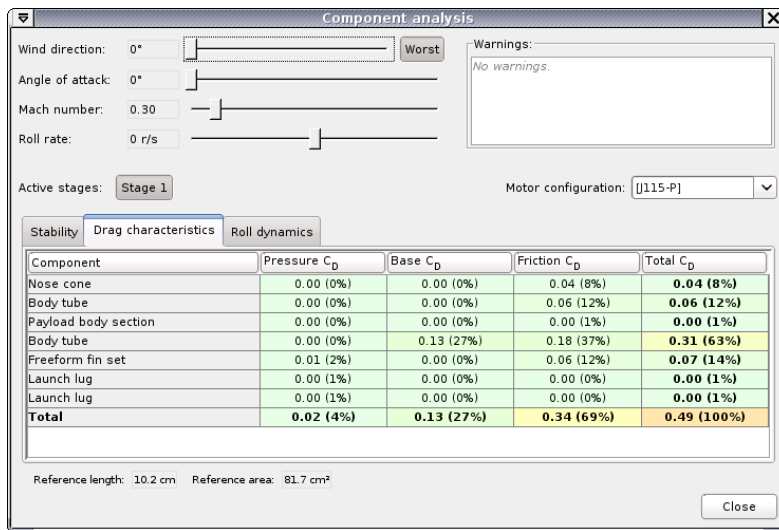
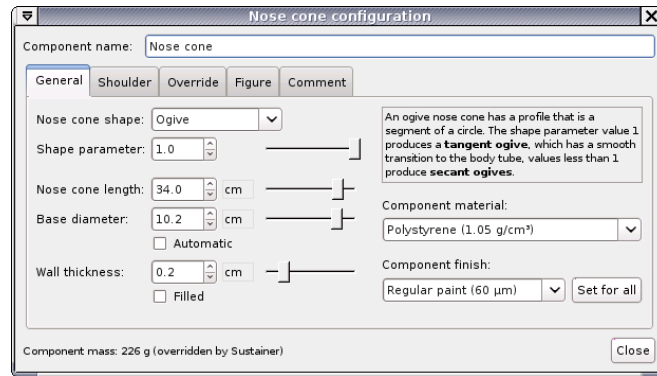


Figure 5.3: Dialog windows for editing the properties of a nose cone and for analyzing the influence of individual components on the stability, drag and roll characteristics of the rocket.

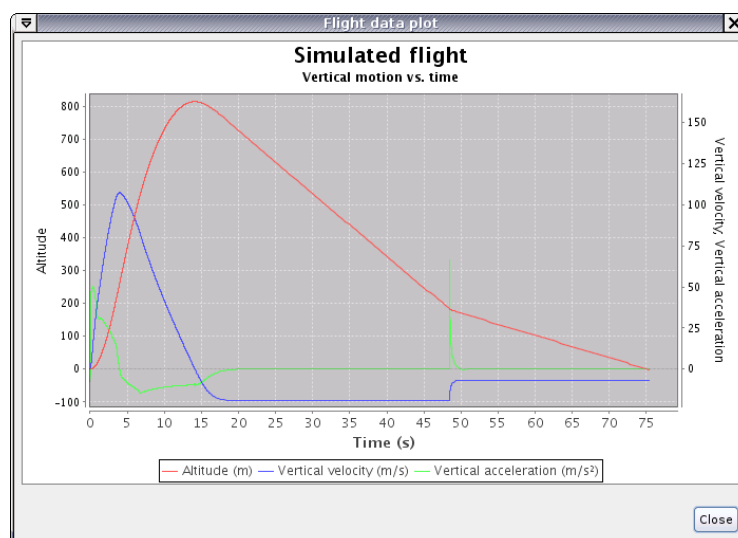
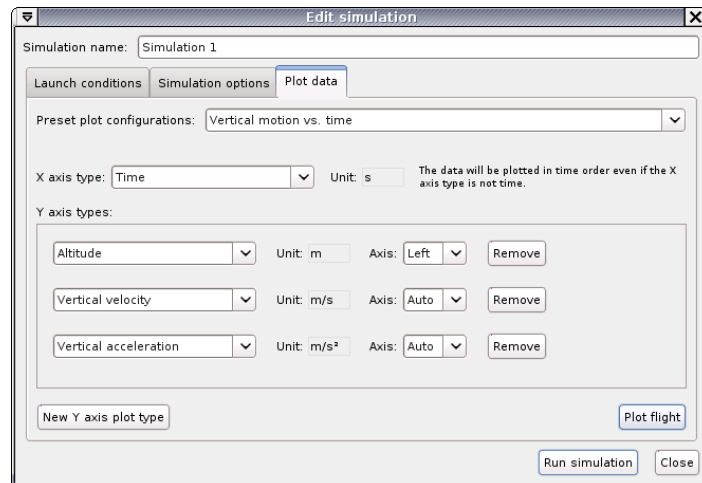


Figure 5.4: Dialog window for changing the simulation plot options and a plot of the simulated flight of the Haisunäättä hybrid rocket.

Chapter 6

Comparison with experimental data

In order to validate the results produced by the software, several test flights were made and compared to the results simulated by the software. In addition to the software produced, the same simulations were performed in the current *de facto* standard model rocket simulator RockSim [5]. The software used was the free demonstration version of RockSim version 8.0.1f9. This is the latest demo version of the software available at the time of writing. The RockSim site states that the demo version is totally equivalent to the normal version except that it can only be used a limited time and it does not simulate the rocket's descent after apogee.

Comparisons were performed using both a typical model rocket design, presented in Section 6.1, and a large hybrid rocket, Section 6.2. A small model with canted fins was also constructed and flown to test the roll simulation, presented in Section 6.3. Finally in Section 6.4 some of the the aerodynamic properties calculated by the software are compared to actual measurements performed in a wind tunnel.

6.1 Comparison with a small model rocket

For purposes of gathering experimental flight data, a small model rocket representing the size and characteristics of a typical model rocket was constructed and flown in various configurations. The rocket model was 56 cm long with a body diameter of 29 mm. The nose cone was a 10 cm long tangent ogive, and the fins simple trapezoidal fins. The entire rocket was painted using an airbrush but not finished otherwise and the fin profiles were left rectangular, so as to represent a typical non-competition model rocket. The velocity of the rocket remained below 0.2 Mach during the entire flight.

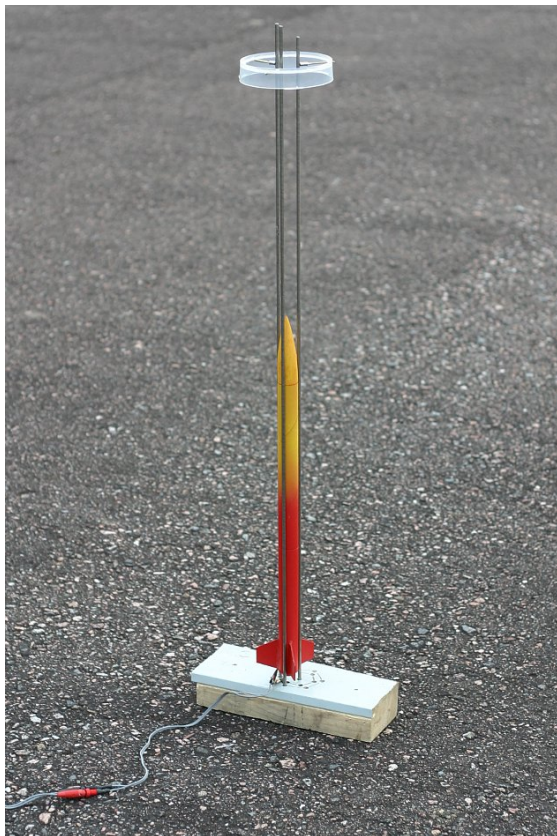
In the payload section of the rocket was included an Alt15K/WD Rev2 altimeter from PerfectFlite [40]. The altimeter measures the altitude of the rocket based on atmospheric pressure changes ten times per second. The manufacturer states the accuracy of the altimeter to be $\pm(0.25\% + 0.6 \text{ m})$. The altimeter logs the flight data, which can later be retrieved to a computer for further analysis.

Four holes, each 1 mm in diameter were drilled evenly around the payload body to allow the ambient air pressure to reach the pressure sensor, as per the manufacturer's instructions. The rocket was launched from a 1 m high tower launcher, which removed the need for any launch lugs. Figure 6.1 presents a picture of the test rocket and the tower launcher.

A design of the same rocket was created in both OpenRocket and RockSim. During construction of the rocket each component was individually weighed and the weight of the corresponding component was overridden in the software for maximum accuracy. Finally, the mass and CG position of the entire rocket was overridden with measured values.

One aspect of the rocket that could not be measured was the average surface roughness. In the OpenRocket design the "regular paint" finish was selected, which corresponds to an average surface roughness of 60 μm . From the available options of "polished", "gloss", "matt" and "unfinished" in RockSim, the "matt" option was estimated to best describe the rocket; the corresponding average surface roughness is unknown.

The rocket was flown using motors manufactured by WECO Feuerwerk (previously Sachsen Feuerwerk) [41], which correspond largely to the motors produced by Estes [42]. The only source available for the thrust curves of Sachsen



(a)



(b)

Figure 6.1: The test rocket awaiting launch on the tower launcher (a) and a close-up of its ventilation holes (b).

Table 6.1: Apogee altitude of simulated and experimental flights with B4-4 and C6-3 motors.

	B4-4		C6-3	
Experimental	64.0 m		151.5 m	
OpenRocket	74.4 m	+16%	161.4 m	+7%
RockSim	79.1 m	+24%	180.1 m	+19%

Feuerwerk motors was a German rocketry store [43], the original source of the measurements are unknown. The thrust curve for the C6-3 motor is quite similar to the corresponding Estes motor, and has a total impulse of 7.5 Ns. However, the thrust curve for the B4-4 motor yields a total impulse of 5.3 Ns, which would make it a C-class motor, while the corresponding Estes motor has an impulse of only 4.3 Ns. Both OpenRocket and RockSim simulated the flight of the rocket using the SF B4-4 motor over 60% higher than the apogee of the experimental results. It is likely that the thrust curve of the SF B4-4 is wrong, and therefore the Estes B4-4 motor was used in the simulations in its stead.

Figure 6.2 shows the experimental and simulated results for the flight using a B4-4 motor (simulations using an Estes motor) and figure 6.3 using a C6-3 motor. The RockSim simulations are truncated at apogee due to limitations of the demonstration version of the software. A summary of the apogee altitudes is presented in Table 6.1.

Both simulations produce a bit too optimistic results. OpenRocket yielded altitudes 16% and 7% too high for the B4-4 and C6-3 motors, respectively, while RockSim had errors of 24% and 19%. The C6-3 flight is considered to be more accurate due to the ambiguity of the B4-4 thrust curve. Another feature that can be seen from the graphs is that the estimated descent speed of the rocket is quite close to the actual descent speed. The error in the descent speeds are 7% and 13% respectively.

The rocket was also launched with a launch lug 24 mm long and 5 mm in diameter attached first to its mid-body and then next to its fins to test the effect of a launch lug on the aerodynamic drag. The apogee altitudes of the tests were 147.2 m and 149.0 m, which correspond to an altitude reduction of 2–3%. The OpenRocket simulation with such a launch lug yielded results approximately 1.3% less than without the launch lug.

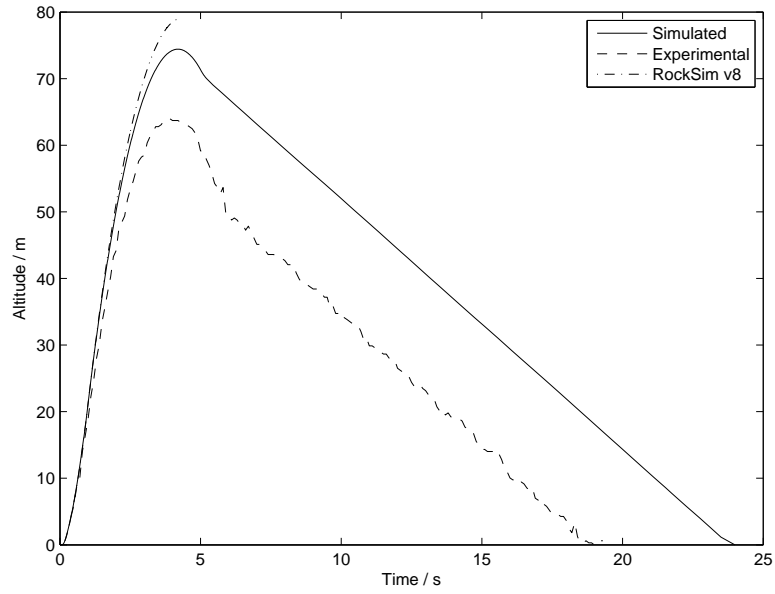


Figure 6.2: Experimental and simulated flight using a B4-4 motor.

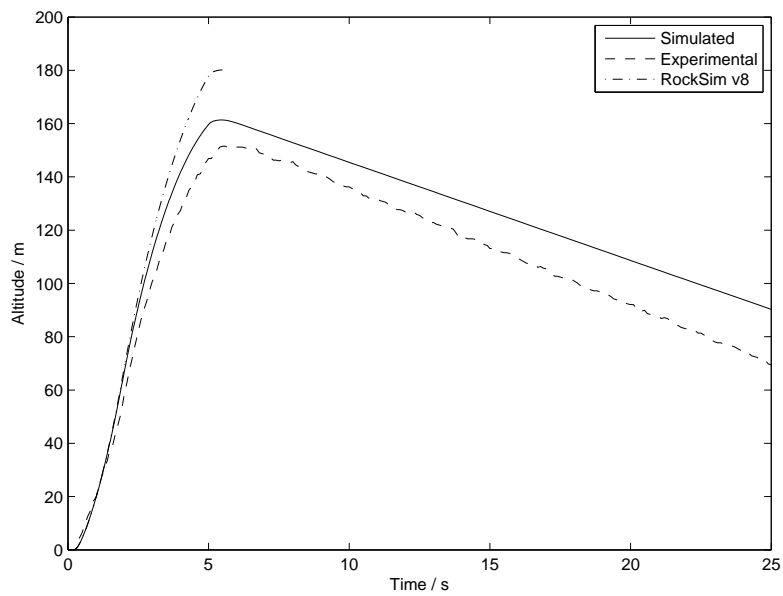


Figure 6.3: Experimental and simulated flight using a C6-3 motor.

6.2 Comparison with a hybrid rocket

The second comparison is with the Haisunäättä hybrid rocket [12], which was launched in September 2008. The rocket is a HyperLOC 835 model, with a length of 198 cm and a body diameter of 10.2 cm. The nose cone is a tangent ogive with a length of 34 cm, and the kit includes three approximately trapezoidal fins.

The flight computer on board was a miniAlt/WD altimeter by Perfect-Flite [40], with a stated accuracy of $\pm 0.5\%$. The flight computer calculates the altitude 20 times per second based on the atmospheric pressure and stores the data into memory for later analysis.

The rocket was modeled as accurately as possible with both OpenRocket and RockSim, but the mass and CG of each component was computed by the software. Finally, the mass of the entire rocket excluding the motor was overridden by the measured mass of the rocket. The surface roughness was estimated as the same as for the small rocket, 60 μm in OpenRocket and “matt” for RockSim.

Figure 6.4 presents the true flight profile and that of the simulations. Both OpenRocket and RockSim estimate a too low apogee altitude, with an error of 16% and 12%, respectively. As in the case of the small rocket model, RockSim produces an estimate 5–10% higher than OpenRocket. It remains unclear which software is more accurate in its estimates.

One error factor also affecting this comparison is the use of a hybrid rocket motor. As noted in Section 2.2, the vapor pressure of the nitrous oxide is highly dependent on temperature, which affects the thrust of the motor. This may cause some variation in the thrust between true flight and motor tests.

6.3 Comparison with a rolling rocket

In order to test the rolling moment computation, a second configuration of the small model rocket, described in Section 6.1, was built with canted fins. The design was identical to the previous one, but each fin was canted by an angle of 5° . In addition, the payload section contained a magnetometer

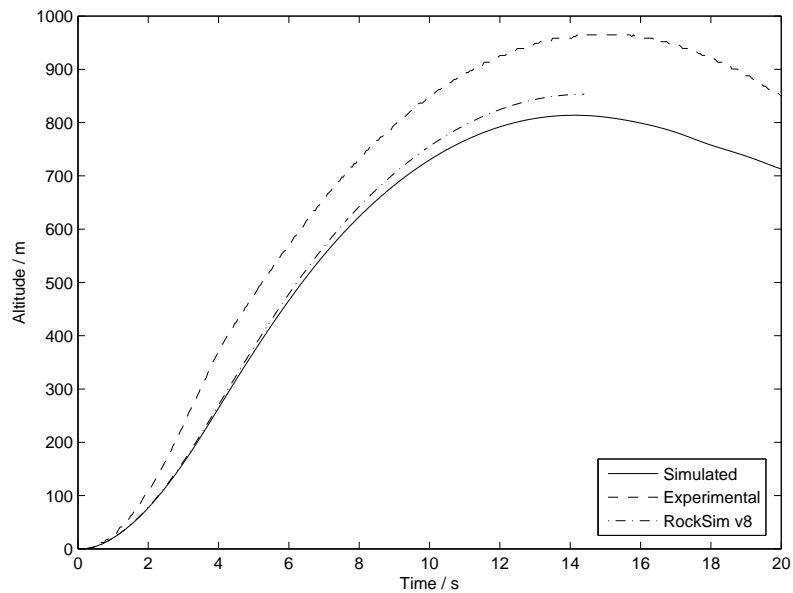


Figure 6.4: Experimental and simulated flight of a hybrid rocket.

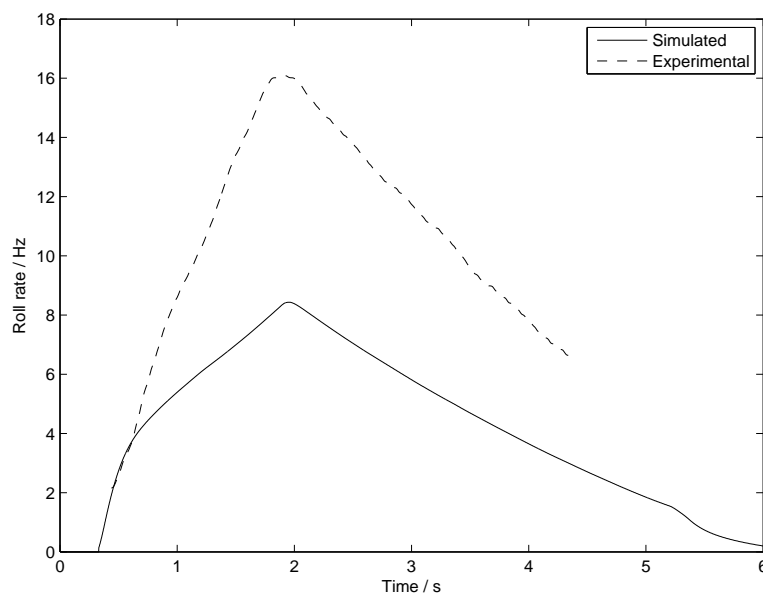


Figure 6.5: Experimental and simulated roll rate results using a C6-3 motor.

logger, built by Antti J. Niskanen, that measured the roll rate of the rocket. The logger used two Honeywell HMC1051 magnetometer sensors to measure the Earth's magnetic field and store the values at a rate of 100 Hz for later analysis. The rocket was launched from the tower launcher using a Sachsen Feuerwerk C6-3 motor. Further test flights were not possible since the lower rocket part was destroyed by a catastrophic motor failure on the second launch.

After the flight, a spectrogram of the magnetometer data was generated by dividing the data into largely overlapping segments of 0.4 seconds each, windowed by a Hamming window, and computing the Fourier transform of these segments. For each segment the frequency with the largest power density was chosen as the roll frequency at the midpoint of the segment in time. The resulting roll frequency as a function of time is plotted in Figure 6.5 with the corresponding simulated roll frequency.

The simulated roll rate differs significantly from the experimental roll rate. During the flight the rocket peaked at a roll rate of 16 revolutions per second, while the simulation has only about half of this. The reason for the discrepancy is unknown and would need more data to analyze. However, after the test flight it was noticed that the cardboard fins of the test rocket were slightly curved, which may have a significant effect on the roll rate. A more precise test rocket with more rigid and straight fins would be needed for a more definitive comparison. Still, even at a cant angle of 7° the simulation produces a roll rate of only 12 r/s.

Even so, it is believed that including roll in the simulation allows users to realistically analyze the effect of roll stabilization for example in windy conditions.

6.4 Comparison with wind tunnel data

Finally, the simulated results were compared with experimental wind tunnel data. The model that was analyzed by J. Ferris in the transonic region [22] and by C. Babb and D. Fuller in the supersonic region [44] is representative of the Arcas Robin meteorological rocket that has been used in high-altitude research activities. The model is 104.1 cm long with a body diameter of 5.72 cm. It includes a 27 cm long tangent ogive nose cone and a 4.6 cm long

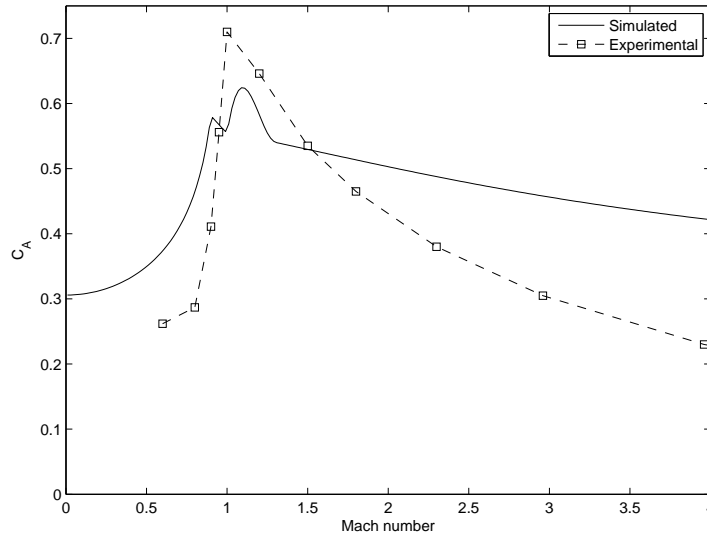


Figure 6.6: Experimental and simulated axial drag coefficient as a function of Mach number.

conical boattail at the rear end, which reduces the diameter to 3.7 cm. The rocket includes four trapezoidal fins, the profiles of which are double-wedges. For details of the configuration, refer to [22].

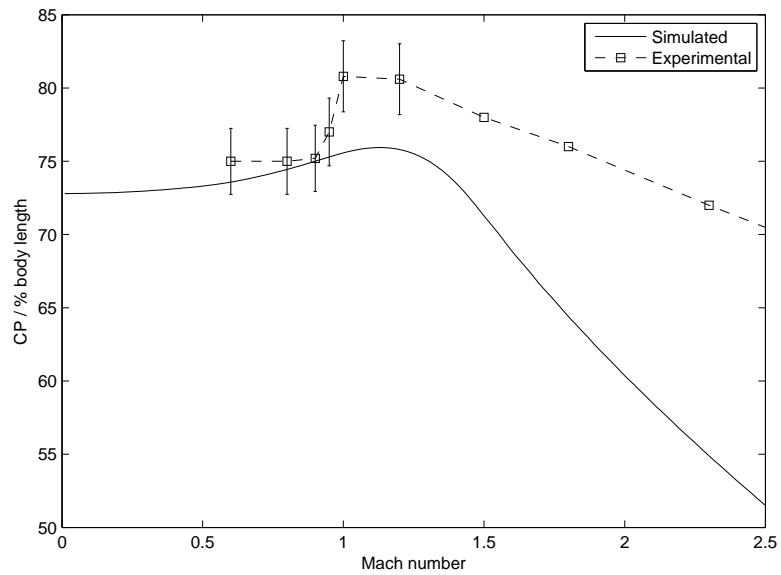
The design was replicated in OpenRocket as closely as possible, given the current limitations of the software. The most notable difference is that an airfoil profile was selected for the fins instead of the double-wedge that is not supported by OpenRocket. The aerodynamical properties were computed at the same Mach and Reynolds numbers as the experimental data.

The most important variables affecting the altitude reached by a rocket are the drag coefficient and CP location. The experimental and simulated axial drag coefficient at zero angle-of-attack is presented in Figure 6.6. The general shape of the simulated drag coefficient follows the experimental results. However, a few aspects of the rocket break the assumptions made in the computation methods. First, the boattail at the end of the rocket reduces the drag by guiding the air into the void left behind it, while the simulation software only takes into account the reduction of base area. Second, the airfoil shape of the fins affects the drag characteristic especially in the transonic region, where it produces the slight reduction peak. Finally, at higher supersonic speeds the simulation produces less reliable results as expected,

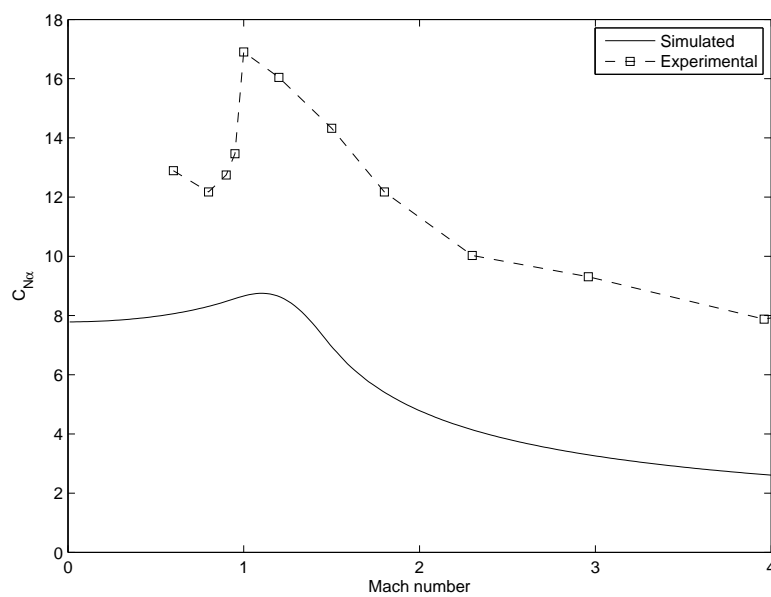
producing a too high drag coefficient. Overall, however, the drag coefficient matches the experimental results with reasonable accuracy, and the results of actual test flights shown in Sections 6.1 and 6.2 give credence to the drag coefficient estimation.

The CP location as a function of Mach number and the normal force coefficient derivative C_{N_α} are presented in Figure 6.7. The 3% error margins in the transonic region were added due to difficulty in estimating the normal force and pitch moment coefficient derivatives from the printed graphs; in the supersonic region the CP location was provided directly. At subsonic speeds the CP location matches the experimental results to within a few percent. At higher supersonic speeds the estimate is too pessimistic, and due to the interpolation this is visible also in the transonic region. However, the CP location is quite reasonable up to about Mach 1.5.

The simulated normal force coefficient derivative is notably lower than the experimental values. The reason for this is unknown, since in his thesis Barrowman obtained results accurate to about 6%. The effect of the lower normal force coefficient on a flight simulation is that the rocket corrects its orientation slightly slower than in reality. The effect on the flight altitude is considered to be small for typical stable rockets.



(a)



(b)

Figure 6.7: Experimental and simulated center of pressure location (a) and normal force coefficient derivative (b) as a function of Mach number.

Chapter 7

Conclusion

Model rocketry is an intriguing sport which combines various fields ranging from aerodynamic design to model construction to pyrotechnics. At its best, it works as an inspiration for youngsters to study engineering and sciences.

This thesis work provides one of the computer-age tools for everybody interested in model rocket design. Providing everybody free access to a full-fledged rocket simulator allows many more hobbyists to experiment with different kinds of rocket designs and become more involved in the sport. The most enthusiastic rocketeers may dive even deeper and get to examine not only the simulation results, but also how those simulations are actually performed.

The software produced contains an easy-to-use interface, which allows new users to start experimenting with the minimum effort. The back-end is designed to be easily extensible, in anticipation of future enhancements. This thesis also includes a step-by-step process for computing the aerodynamical characteristics of a rocket and for simulating its flight. These are the current default implementations used by the software.

Comparison to experimental data shows that the most important aerodynamical parameters for flight simulation—the center of pressure location and drag coefficient—are simulated with an accuracy of approximately 10% at subsonic velocities. In this velocity regime the accuracy of the simulated altitude is on par with the commercial simulation software RockSim. While comparison with supersonic rockets was not possible, it is expected that the

simulation is reasonably accurate to at least Mach 1.5.

The six degree of freedom simulator also allows simulating rocket roll in order to study the effect of roll stabilization, a feature not available in other hobby-level rocket simulators. While the comparison with experimental data of a rolling rocket was inconclusive as to its accuracy, it is still expected to give valuable insight into the effects of roll during flight.

The external listener classes that can be attached to the simulator allow huge potential for custom extensions. For example testing the active roll reduction controller that will be included in the successor project of Haisunäätä would have been exceedingly difficult without such support. By interfacing the actual controller with a simulated flight environment it was possible to discover various bugs in the controller software that would otherwise have gone undetected.

Finally, it must be emphasized that the release of the OpenRocket software is not the end of this project. In line with the Open Source philosophy, it is just the beginning of its development cycle, where anybody with the know-how can contribute to making OpenRocket an even better simulation environment.

Acknowledgments

I would like to express my deepest gratitude to M.Sc. Timo Saileranta for his invaluable advice and consultation on the aerodynamic simulation of rockets. Without his input the creation of this thesis would have been exceedingly laborious. I would also like to thank Prof. Rolf Stenberg for supervising the writing of this thesis, and Juhani Talvela for proofreading and commenting early versions of this text.

I am also deeply grateful for my parents Jouni and Riitta, my entire family, friends and teachers, who have always encouraged me onwards in my life and studies. Above all I would like to thank my brother, Antti J. Niskanen, for being an inspiration throughout my life and also for building the magnetometer logger used in the experimental flights; and my fiancée Merli Lahtinen, for her patience and loving understanding for my passion towards rocketry.

Bibliography

- [1] Niskanen, S., *Development of an Open Source model rocket simulation software*, M.Sc. thesis, Helsinki University of Technology, 2009. Available at <http://openrocket.sourceforge.net/documentation.html>.
- [2] Stine, H., Stine, B., *Handbook of Model Rocketry*, 7th edition, Wiley, 2004.
- [3] Barrowman, J., Barrowman, J., The theoretical prediction of the center of pressure, *National Association of Rocketry Annual Meet 8*, 1966. Available at http://www.apogeerockets.com/Education/downloads/barrowman_report.pdf, retrieved 14.5.2009.
- [4] Barrowman, J., *The practical calculation of the aerodynamic characteristics of slender finned vehicles*, M.Sc. thesis, The Catholic University of America, 1967.
- [5] van Milligan, T., RockSim Model Rocket Design and Simulation Software, <http://www.apogeerockets.com/RockSim.asp>, retrieved 14.5.2009.
- [6] Coar, K., The Open Source Definition (Annotated), <http://www.opensource.org/docs/definition.php>, retrieved 14.5.2009.
- [7] Niskanen, S., The OpenRocket web-site, <http://openrocket.sourceforge.net/>, retrieved 25.5.2009.
- [8] Anon., Model Rocket Safety Code, <http://www.nar.org/NARmrsc.html>, retrieved 14.5.2009.
- [9] Anon., Combined CAR/NAR/TRA Certified Rocket Motors List, <http://www.nar.org/SandT/pdf/CombinedList.pdf>, retrieved 14.5.2009.

- [10] Coker, J., ThrustCurve Hobby Rocket Motor Data, <http://www.thrustcurve.org/>, retrieved 14.5.2009.
- [11] Kane, J., Estes D12, <http://www.nar.org/SandT/pdf/Estes/D12.pdf>, retrieved 14.5.2009.
- [12] Puhakka, A., Haisunäätä—suomalainen hybridiraketiprojekti (in Finnish), <http://haisunaata.avaruuteen.fi/>, retrieved 14.5.2009.
- [13] Galejs, R., Wind instability—What Barrowman left out, <http://projetosulfos.if.sc.usp.br/artigos/sentinel39-galejs.pdf>, retrieved 14.5.2009.
- [14] Mandell, G., Caporaso, G., Bengen, W., *Topics in Advanced Model Rocketry*, MIT Press, 1973.
- [15] Hoerner, S., *Fluid-dynamic drag*, published by the author, 1965.
- [16] Barrowman, J., Elliptical Fin C.P. Equations, *Model Rocketry* (Nov 1970). Available at <http://www.argoshpr.ch/articles/pdf/EllipticalCP.jpg>, retrieved 14.5.2009.
- [17] Mason, W., Applied Computational Aerodynamics, http://www.aoe.vt.edu/~mason/Mason_f/CAtxtTop.html, pp. A-27–A-28, retrieved 14.5.2009.
- [18] Fleeman, E., *Tactical missile design*, 2nd edition, p. 33, AIAA, 2006.
- [19] Diederich, F., *A plan-form parameter for correlating certain aerodynamic characteristics of swept wings*, NACA-TN-2335, 1951.
- [20] Barrowman, J., *FIN A computer program for calculating the aerodynamic characteristics of fins at supersonic speeds*, NASA-TM X-55523, 1966.
- [21] Pettis, W., *Aerodynamic Characteristics of Multiple Fins of Rectangular Planform on a Body of Revolution at Mach Numbers of 1.48 to 2.22*, RD-TM-67-5, US Army Missile Command, 1967.
- [22] Ferris, J., *Static stability investigation of a single-stage sounding rocket at Mach numbers from 0.60 to 1.20*, NASA-TN-D-4013, 1967.
- [23] Monta, W., *Aerodynamic characteristics at mach numbers from 1.60 to 2.16 of a blunt-nose missile model having a triangular cross section and fixed triform fins*, NASA-TM-X-2340, 1971.

- [24] Anon., *Design of aerodynamically stabilized free rockets*, MIL-HDBK-762, US Army Missile Command, 1990.
- [25] Anon., *Handbook of supersonic aerodynamics, Section 8, Bodies of revolution*, NAVWEPS REPORT 1488, 1961.
- [26] Syverston, C., Dennis, D., *A second-order shock-expansion method applicable to bodies of revolution near zero lift*, NACA-TR-1328, 1957.
- [27] Anon., *Standard Atmosphere*, ISO 2533:1975, International Organization for Standardization, 1975.
- [28] Anon., *U.S. Standard Atmosphere 1976*, NASA-TM-X-74335; NOAA-S/T-76-1562, 1976.
- [29] Anon., International Standard Atmosphere, http://en.wikipedia.org/wiki/International_Standard_Atmosphere, retrieved 14.5.2009.
- [30] Burton, T., Sharpe, D., Jenkins, N., Bossanyi, E., *Wind Energy Handbook*, Wiley, 2001.
- [31] Kasdin, J., Discrete Simulation of Colored Noise and Stochastic Processes and $1/f^\alpha$ Power Law Noise Generation, *Proceedings of the IEEE*, **83**, No. 5 (1995), p. 822.
- [32] Anon., Euler angles, http://en.wikipedia.org/wiki/Euler_angles, retrieved 14.5.2009.
- [33] Anon., Euler's rotation theorem, http://en.wikipedia.org/wiki/Euler's_rotation_theorem, retrieved 14.5.2009.
- [34] Anon., Quaternions and spatial rotation, http://en.wikipedia.org/wiki/Quaternions_and_spatial_rotation, retrieved 14.5.2009.
- [35] Anon., List of moments of inertia, http://en.wikipedia.org/wiki/List_of_moments_of_inertia, retrieved 14.5.2009.
- [36] Niskanen, S., Polluxin tuulitunneli (in Finnish), <http://pollux.tky.fi/tuulitunneli.html>, retrieved 14.5.2009.
- [37] Anon., GNU General Public License, Version 3, <http://www.gnu.org/copyleft/gpl.html>, retrieved 14.5.2009.
- [38] Anon., Java Language Specification, Chapter 7, Packages, http://java.sun.com/docs/books/jls/third_edition/html/packages.html#7.7, retrieved 14.5.2009.

- [39] Deutsch, P., *GZIP file format specification version 4.3*, RFC 1952, <http://www.ietf.org/rfc/rfc1952.txt>, 1996. Retrieved on 14.5.2009.
- [40] Anon., Affordable instrumentation for (sm)all rockets, <http://www.perfectflite.com/>, retrieved 14.5.2009.
- [41] Anon., WECO Feuerwerk, <http://www.weco-feuerwerk.de/>, retrieved 14.5.2009.
- [42] Anon., Estesrockets.com, <http://www.estesrockets.com/>, retrieved 14.5.2009.
- [43] Anon., Schubdiagramme SF, <http://www.raketenmodellbautechnik.de/produkte/Motoren/SF-Motoren.pdf>, 14.5.2009.
- [44] Babb, C., Fuller, D., *Static stability investigation of a sounding-rocket vehicle at Mach numbers from 1.50 to 4.63*, NASA-TN-D-4014, 1967.
- [45] Stoney, W., *Collection of Zero-Lift Drag Data on Bodies of Revolution from Free-Flight Investigations*, NASA-TR-R-100, 1961.
- [46] Kidwell, C., Streamer Duration Optimization: Material and Length-to-Width Ratio, *National Association of Rocketry Annual Meet 43*, 2001. Available at <http://www.narhams.org/library/rnd/StreamerDuration.pdf>, retrieved 14.5.2009.

Appendix A

Nose cone and transition geometries

Model rocket nose cones are available in a wide variety of shapes and sizes. In this appendix the most common shapes and their defining parameters are presented.

A.1 Conical

The most simple nose cone shape is a right circular cone. They are easy to make from a round piece of cardboard. A conical nose cone is defined simply by its length and base diameter. An additional parameter is the opening angle ϕ , shown in Figure A.1(a). The defining equation of a conical nose cone is

$$r(x) = \frac{x}{L} \cdot R. \quad (\text{A.1})$$

A.2 Ogival

Ogive nose cones have a profile which is an arc of a circle, as shown in Figure A.1(b). The most common ogive shape is the *tangent ogive* nose cone, which is formed when radius of curvature of the circle ρ_t is selected

such that the joint between the nose cone and body tube is smooth,

$$\rho_t = \frac{R^2 + L^2}{2R}. \quad (\text{A.2})$$

If the radius of curvature ρ is greater than this, then the resulting nose cone has an angle at the joint between the nose cone and body tube, and is called a *secant ogive*. The secant ogives can also be viewed as a larger tangent ogive with its base cropped. At the limit $\rho \rightarrow \infty$ the secant ogive becomes a conical nose cone.

The parameter value κ used for ogive nose cones is the ratio of the radius of curvature of a corresponding tangent ogive ρ_t to the radius of curvature of the nose cone ρ :

$$\kappa = \frac{\rho_t}{\rho} \quad (\text{A.3})$$

κ takes values from zero to one, where $\kappa = 1$ produces a tangent ogive and $\kappa = 0$ produces a conical nose cone (infinite radius of curvature).

With a given length L , radius R and parameter κ the radius of curvature is computed by

$$\rho^2 = \frac{(L^2 + R^2) \cdot \left(((2 - \kappa)L)^2 + (\kappa R)^2 \right)}{4(\kappa R)^2}. \quad (\text{A.4})$$

Using this the radius at position x can be computed as

$$r(x) = \sqrt{\rho^2 - (L/\kappa - x)^2} - \sqrt{\rho^2 - (L/\kappa)^2} \quad (\text{A.5})$$

A.3 Elliptical

Elliptical nose cones have the shape of an ellipsoid with one major radius is L and the other two R . The profile has a shape of a half-ellipse with major axis L and R , Figure A.1(c). It is a simple geometric shape common in model rocketry. The special case $R = L$ corresponds to a half-sphere.

The equation for an elliptical nose cone is obtained by stretching the equation of a unit circle:

$$r(x) = R \cdot \sqrt{1 - \left(1 - \frac{x}{L}\right)^2} \quad (\text{A.6})$$

A.4 Parabolic series

A parabolic nose cone is the shape generated by rotating a section of a parabola around a line perpendicular to its symmetry axis, Figure A.1(d). This is distinct from a paraboloid, which is rotated around this symmetry axis (see Appendix A.5).

Similar to secant ogives, the base of a “full” parabolic nose cone can be cropped to produce nose cones which are not tangent with the body tube. The parameter κ describes the portion of the larger nose cone to include, with values ranging from zero to one. The most common values are $\kappa = 0$ for a conical nose cone, $\kappa = 0.5$ for a 1/2 parabola, $\kappa = 0.75$ for a 3/4 parabola and $\kappa = 1$ for a full parabola. The equation of the shape is

$$r(x) = R \cdot \frac{x}{L} \left(\frac{2 - \kappa \frac{x}{L}}{2 - \kappa} \right). \quad (\text{A.7})$$

A.5 Power series

The power series nose cones are generated by rotating the segment

$$r(x) = R \left(\frac{x}{L} \right)^\kappa \quad (\text{A.8})$$

around the x-axis, Figure A.1(e). The parameter value κ can range from zero to one. Special cases are $\kappa = 1$ for a conical nose cone, $\kappa = 0.75$ for a 3/4 power nose cone and $\kappa = 0.5$ for a 1/2 power nose cone or an ellipsoid. The limit $\kappa \rightarrow 0$ forms a blunt cylinder.

A.6 Haack series

In contrast to the other shapes which are formed from rotating geometric shapes or simple formulae around an axis, the Haack series nose cones are mathematically derived to minimize the theoretical pressure drag. Even though they are defined as a series, two specific shapes are primarily used, the *LV-Haack* shape and the *LD-Haack* or *Von Kármán* shape. The letters

LV and LD refer to length-volume and length-diameter, and they minimize the theoretical pressure drag of the nose cone for a specific length and volume or length and diameter, respectively. Since the parameters defining the dimensions of the nose cone are its length and radius, the Von Kármán nose cone (Figure A.1(f)) should, in principle, be the optimal nose cone shape.

The equation for the series is

$$r(x) = \frac{R}{\sqrt{\pi}} \sqrt{\theta - \frac{1}{2} \sin(2\theta) + \kappa \sin^3 \theta} \quad (\text{A.9})$$

where

$$\theta = \cos^{-1} \left(1 - \frac{2x}{L} \right). \quad (\text{A.10})$$

The parameter value $\kappa = 0$ produces the Von Kármán or LD-Haack shape and $\kappa = 1/3$ produces the LV-Haack shape. In principle, values of κ up to $2/3$ produce monotonic nose cone shapes. However, since there is no experimental data available for the estimation of nose cone pressure drag for $\kappa > 1/3$ (see Appendix B.3), the selection of the parameter value is limited in the software to the range $0 \dots 1/3$.

A.7 Transitions

The vast majority of all model rocket transitions are conical. However, all of the nose cone shapes may be adapted as transition shapes as well. The transitions are parametrized with the fore and aft radii R_1 and R_2 , length L and the optional shape parameter κ .

Two choices exist when adapting the nose cones as transition shapes. One is to take a nose cone with base radius R_2 and crop the tip of the nose at the radius position R_1 . The length of the nose cone must be selected suitably that the length of the transition is L . Another choice is to have the profile of the transition resemble two nose cones with base radius $R_2 - R_1$ and length L . These two adaptations are called *clipped* and *non-clipped* transitions, respectively. A clipped and non-clipped elliptical transition is depicted in Figure A.2.

For some transition shapes the clipped and non-clipped adaptations are the same. For example, the two possible ogive transitions have equal radii of

curvature and are therefore the same. Specifically, the conical and ogival transitions are equal whether clipped or not, and the parabolic series are extremely close to each other.

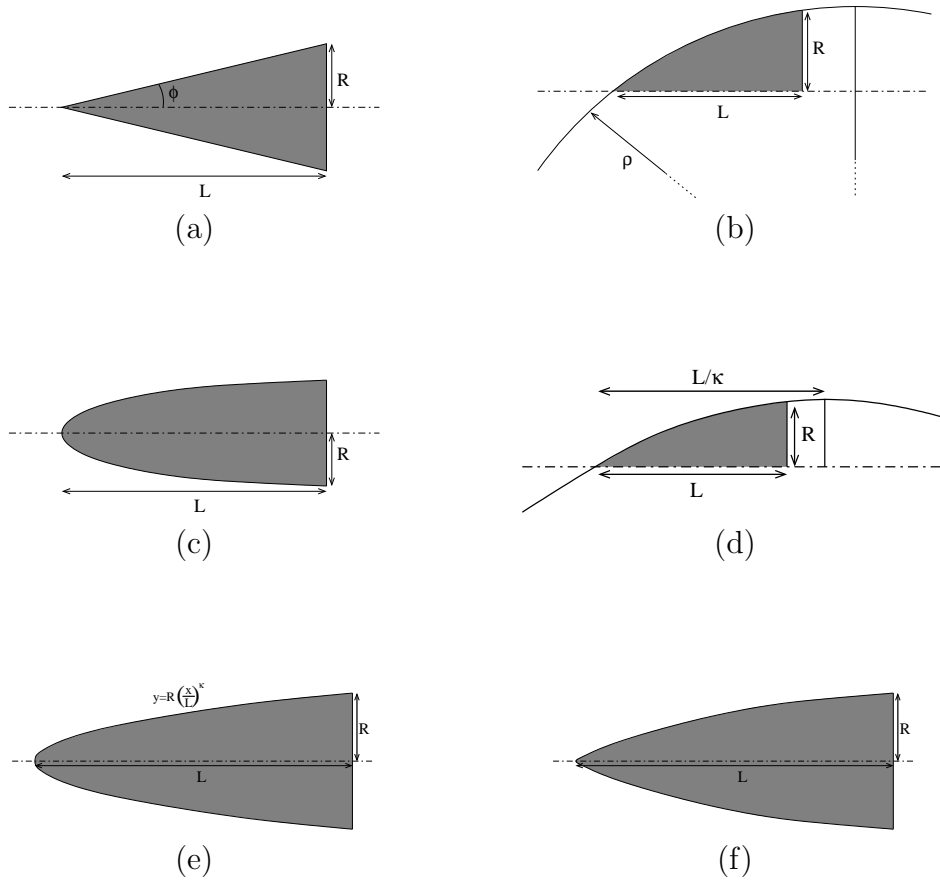


Figure A.1: Various nose cone geometries: (a) conical, (b) secant ogive, (c) elliptical, (d) parabolic, (e) 1/2 power (ellipsoid) and (f) Haack series (Von Kárman).

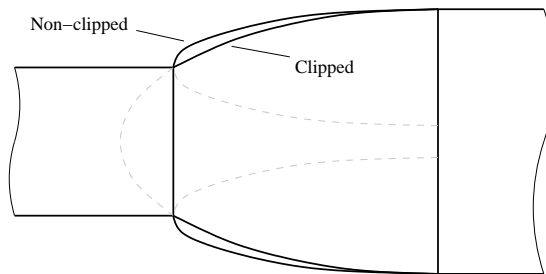


Figure A.2: A clipped and non-clipped elliptical transition.

Appendix B

Transonic wave drag of nose cones

The wave drag of different types of nose cones vary largely in the transonic velocity region. Each cone shape has its distinct properties. In this appendix methods for calculating and interpolating the drag of various nose cone shapes at transonic and supersonic speeds are presented. A summary of the methods is presented in Appendix B.5.

B.1 Blunt cylinder

A blunt cylinder is the limiting case for every nose cone shape at the limit $f_N \rightarrow 0$. Therefore it is useful to have a formula for the front pressure drag of a circular cylinder in longitudinal flow. As the object is not streamline, its drag coefficient does not vary according to the Prandtl factor (3.14). Instead, the coefficient is approximately proportional to the *stagnation pressure*, or the pressure at areas perpendicular to the airflow. The stagnation pressure can be approximated by the function [15, pp. 15-2, 16-3]

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

The pressure drag coefficient of a blunt circular cylinder as a function of the Mach number can then be written as

$$(C_{D\bullet})_{\text{pressure}} = (C_{D\bullet})_{\text{stag}} = 0.85 \cdot \frac{q_{\text{stag}}}{q}. \quad (\text{B.2})$$

B.2 Conical nose cone

A conical nose cone is simple to construct and closely resembles many slender nose cones. The conical shape is also the limit of several parametrized nose cone shapes, in particular the secant ogive with parameter value 0.0 (infinite circle radius), the power series nose cone with parameter value 1.0 and the parabolic series with parameter value 0.0.

Much experimental data is available on the wave drag of conical nose cones. Hoerner presents formulae for the value of $C_{D\bullet}$ at supersonic speeds, the derivative $dC_{D\bullet}/dM$ at $M = 1$, and a figure of $C_{D\bullet}$ at $M = 1$ [15, pp. 16-18...16-20]. Based on these and the low subsonic drag coefficient (3.86), a good interpolation of the transonic region is possible.

The equations presented by Hoerner are given as a function of the half-apex angle ε , that is, the angle between the conical body and the body centerline. The half-apex angle is related to the nose cone fineness ratio by

$$\tan \varepsilon = \frac{d/2}{l} = \frac{1}{2f_N}. \quad (\text{B.3})$$

The pressure drag coefficient at supersonic speeds ($M \gtrsim 1.3$) is given by

$$\begin{aligned} (C_{D\bullet})_{\text{pressure}} &= 2.1 \sin^2 \varepsilon + 0.5 \frac{\sin \varepsilon}{\sqrt{M^2 - 1}} \\ &= \frac{2.1}{1 + 4f_N^2} + \frac{0.5}{\sqrt{(1 + 4f_N^2)(M^2 - 1)}}. \end{aligned} \quad (\text{B.4})$$

It is worth noting that as the Mach number increases, the drag coefficient tends to the constant value $2.1 \sin^2 \varepsilon$. At $M = 1$ the slope of the pressure drag coefficient is equal to

$$\left. \frac{\partial (C_{D\bullet})_{\text{pressure}}}{\partial M} \right|_{M=1} = \frac{4}{\gamma + 1} \cdot (1 - 0.5 C_{D\bullet, M=1}) \quad (\text{B.5})$$

where $\gamma = 1.4$ is the specific heat ratio of air and the drag coefficient at $M = 1$ is approximately

$$C_{D\bullet, M=1} = 1.0 \sin \varepsilon. \quad (\text{B.6})$$

The pressure drag coefficient between Mach 0 and Mach 1 is interpolated using equation (3.86). Between Mach 1 and Mach 1.3 the coefficient is calculated using polynomial interpolation with the boundary conditions from equations (B.4), (B.5) and (B.6).

B.3 Ellipsoidal, power, parabolic and Haack series nose cones

A comprehensive data set of the pressure drag coefficient for all nose cone shapes at all fineness ratios at all Mach numbers is not available. However, Stoney has collected a compendium of nose cone drag data including data on the effect of the fineness ratio f_N on the drag coefficient and an extensive study of drag coefficients of different nose cone shapes at fineness ratio 3 [45]. The same report suggests that the effects of fineness ratio and Mach number may be separated.

The curves of the pressure drag coefficient as a function of the nose fineness ratio f_N can be closely fitted with a function of the form

$$(C_{D\bullet})_{\text{pressure}} = \frac{a}{(f_N + 1)^b}. \quad (\text{B.7})$$

The parameters a and b can be calculated from two data points corresponding to fineness ratios 0 (blunt cylinder, Appendix B.1) and ratio 3. Stoney includes experimental data of the pressure drag coefficient as a function of Mach number at fineness ratio 3 for power series $x^{1/4}$, $x^{1/2}$, $x^{3/4}$ shapes, 1/2, 3/4 and full parabolic shapes, ellipsoidal, L-V Haack and Von Kármán nose cones. These curves are written into the software as data curve points. For parametrized nose cone shapes the necessary curve is interpolated if necessary. Typical nose cones of model rockets have fineness ratios in the region of 2–5, so the extrapolation from data of fineness ratio 3 is within reasonable bounds.

B.4 Ogive nose cones

One notable shape missing from the data in Stoney's report are secant and tangent ogives. These are common shapes for model rocket nose cones. However, no similar experimental data of the pressure drag as a function of Mach number was found for ogive nose cones.

At supersonic velocities the drag of a tangent ogive is approximately the same as the drag of a conical nose cone with the same length and diameter, while secant ogives have a somewhat smaller drag [25, p. 239]. The minimum drag is achieved when the secant ogive radius is approximately twice that of a corresponding tangent ogive, corresponding to the parameter value 0.5. The minimum drag is consistently 18% less than that of a conical nose at Mach numbers in the range of 1.6–2.5 and for fineness ratios of 2.0–3.5. Since no better transonic data is available, it is assumed that ogives follow the conical drag profile through the transonic and supersonic region. The drag of the corresponding conical nose is diminished in a parabolic fashion with the ogive parameter, with a minimum of -18% at a parameter value of 0.5.

B.5 Summary of nose cone drag calculation

The low subsonic pressure drag of nose cones is calculated using equation (3.86):

$$(C_{D\bullet, M=0})_p = 0.8 \cdot \sin^2 \phi.$$

The high subsonic region is interpolated using a function of the form presented in equation (3.87):

$$(C_{D\bullet})_{\text{pressure}} = a \cdot M^b + (C_{D\bullet, M=0})_p$$

where a and b are selected according to the lower boundary of the transonic pressure drag and its derivative.

The transonic and supersonic pressure drag is calculated depending on the nose cone shape as follows:

Conical: At supersonic velocities ($M > 1.3$) the pressure drag is calculated using equation (B.4). Between Mach 1 and 1.3 the drag is interpolated using

a polynomial with boundary conditions given by equations (B.4), (B.5) and (B.6).

Ogival: The pressure drag at transonic and supersonic velocities is equal to the pressure drag of a conical nose cone with the same diameter and length corrected with a shape factor:

$$(C_{D\bullet})_{\text{pressure}} = (0.72 \cdot (\kappa - 0.5)^2 + 0.82) \cdot (C_{D\bullet})_{\text{cone}}. \quad (\text{B.8})$$

The shape factor is one at $\kappa = 0, 1$ and 0.82 at $\kappa = 0.5$.

Other shapes: The pressure drag calculation is based on experimental data curves:

1. Determine the pressure drag C_3 of a similar nose cone with fineness ratio $f_N = 3$ from experimental data. If data for a particular shape parameter is not available, interpolate the data between parameter values.
2. Calculate the pressure drag of a blunt cylinder C_0 using equation (B.2).
3. Interpolate the pressure drag of the nose cone using equation (B.7). After parameter substitution the equation takes the form

$$(C_{D\bullet})_{\text{pressure}} = \frac{C_0}{(f_N + 1)^{\log_4 C_0/C_3}} = C_0 \cdot \left(\frac{C_3}{C_0}\right)^{\log_4(f_N+1)} \quad (\text{B.9})$$

The last form is computationally more efficient since the exponent $\log_4(f_N + 1)$ is constant during a simulation.

Appendix C

Streamer drag coefficient estimation

A streamer is a typically rectangular strip of plastic or other material that is used as a recovery device especially in small model rockets. The deceleration is based on the material flapping in the passing air, thus causing air resistance. Streamer optimization has been a subject of much interest in the rocketry community [46], and contests on streamer landing duration are organized regularly. In order to estimate the drag force of a streamer a series of experiments were performed and an empirical formula for the drag coefficient was developed.

One aspect that is not taken into account in the present investigation is the fluctuation between the streamer and rocket. At one extreme a rocket with a very small streamer drops head first to the ground with almost no deceleration at all. At the other extreme there is a very large streamer creating significant drag, and the rocket falls below it tail-first. Between these two extremes is a point where the orientation is labile, and the rocket as a whole twirls around during descent. This kind of interaction between the rocket and streamer cannot be investigated in a wind tunnel and would require an extensive set of flight tests to measure. Therefore it is not taken into account, instead, the rocket is considered effectively a point mass at the end of the streamer, the second extreme mentioned above.

Experimental methods

A series of experiments to measure the drag coefficients of streamers was performed using the $40 \times 40 \times 120$ cm wind tunnel of Pollux [36]. The experiments were performed using various materials, widths and lengths of streamers and at different wind speeds. The effect of the streamer size and shape was tested separately from the effect of the streamer material.

A tube with a rounded 90° angle at one end was installed in the center of the wind tunnel test section. A line was drawn through the tube so that one end of the line was attached to a the streamer and the other end to a weight which was placed on a digital scale. When the wind tunnel was active the force produced by the streamer was read from the scale. A metal wire was taped to the edge of the streamer to keep it rigid and the line attached to the midpoint of the wire.

A few different positions within the test section and free line lengths were tried. All positions seemed to produce approximately equal results, but the variability was significantly lower when the streamer fit totally into the test section and had a only 10 cm length of free line between the tube and streamer. This configuration was used for the rest of the experiments.

Each streamer was measured at three different velocities, 6 m/s, 9 m/s and 12 m/s. The results indicated that the force produced is approximately proportional to the square of the airspeed, signifying that the definition of a drag coefficient is valid also for streamers.

The natural reference area for a streamer is the area of the strip. However, since in the simulation we are interested in the total drag produced by a streamer, it is better to first produce an equation for the drag coefficient normalized to unit area, $C_D \cdot A_{\text{ref}}$. These coefficient values were calculated separately for the different velocities and then averaged to obtain the final normalized drag coefficient of the streamer.

Effect of streamer shape

Figure C.1 presents the normalized drag coefficient as a function of the streamer width and length for a fixed material of 80 g/m^2 polyethylene plas-

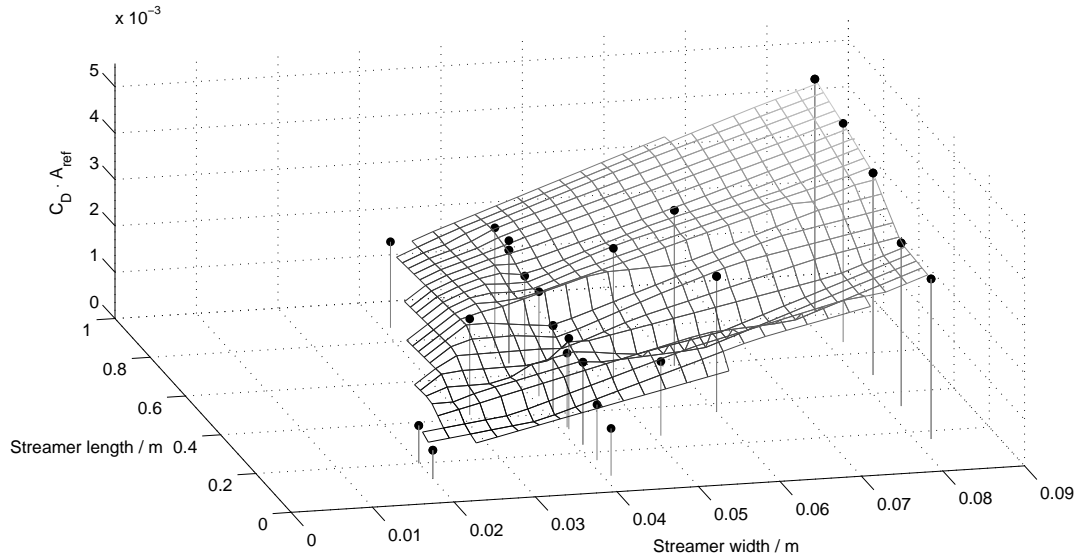


Figure C.1: The normalized drag coefficient of a streamer as a function of the width and length of the streamer. The points are the measured values and the mesh is cubically interpolated between the points.

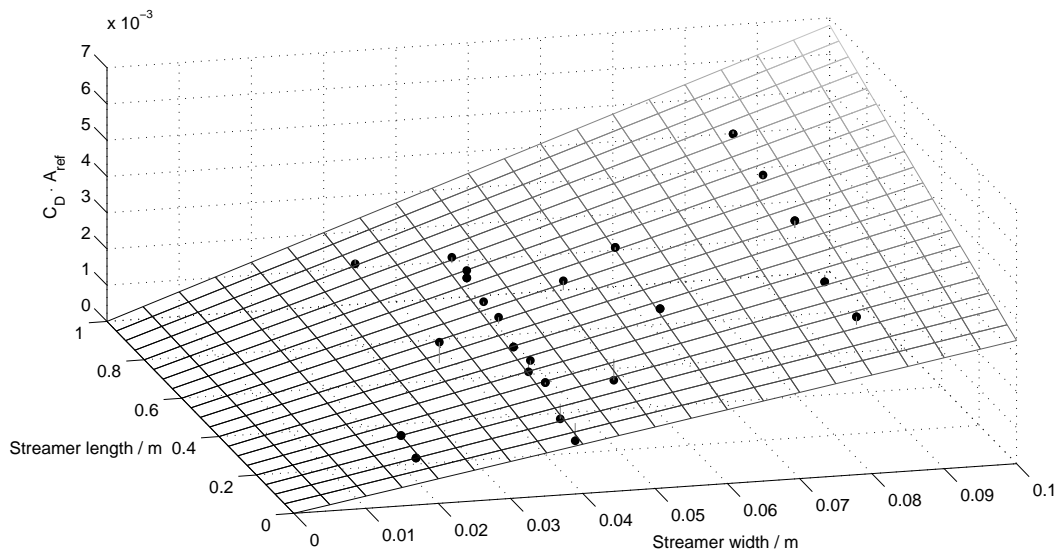


Figure C.2: Estimated and measured normalized drag coefficients of a streamer as a function of the width and length of the streamer. The lines from the points lead to their respective estimate values.

tic. It was noticed that for a specific streamer length, the normalized drag coefficient was approximately linear with the width,

$$C_D \cdot A_{\text{ref}} = k \cdot w, \quad (\text{C.1})$$

where w is the width and k is dependent on the streamer length. The slope k was found to be approximately linear with the length of the streamer, with a linear regression of

$$k = 0.034 \cdot (l + 1 \text{ m}). \quad (\text{C.2})$$

Substituting equation (C.2) into (C.1) yields

$$C_D \cdot A_{\text{ref}} = 0.034 \cdot (l + 1 \text{ m}) \cdot w \quad (\text{C.3})$$

or using $A_{\text{ref}} = wl$

$$C_D = 0.034 \cdot \frac{l + 1 \text{ m}}{l}. \quad (\text{C.4})$$

The estimate as a function of the width and length is presented in Figure C.2 along with the measured data points. The lines originating from the points lead to their respective estimate values. The average relative error produced by the estimate was 9.7%.

Effect of streamer material

The effect of the streamer material was studied by creating 4×40 cm and 8×48 cm streamers from various household materials commonly used in streamers. The tested materials were polyethylene plastic of various thicknesses, cellophane and crêpe paper. The properties of the materials are listed in Table C.1.

Figure C.3 presents the normalized drag coefficient as a function of the material thickness and surface density. It is evident that the thickness is not a good parameter to characterize the drag of a streamer. On the other hand, the drag coefficient as a function of surface density is nearly linear, even including the crêpe paper. While it is not as definitive, both lines seem to intersect with the x -axis at approximately -25 g/m^2 . Therefore the coefficient of the 80 g/m^2 polyethylene estimated by equation (C.4) is corrected for a material surface density ρ_m with

$$C_{D_m} = \left(\frac{\rho_m + 25 \text{ g/m}^2}{105 \text{ g/m}^2} \right) \cdot C_D. \quad (\text{C.5})$$

Combining these two equations, one obtains the final empirical equation

$$C_{D_m} = 0.034 \cdot \left(\frac{\rho_m + 25 \text{ g/m}^2}{105 \text{ g/m}^2} \right) \cdot \left(\frac{l + 1 \text{ m}}{l} \right). \quad (\text{C.6})$$

This equation is also reasonable since it produces positive and finite normalized drag coefficients for all values of w , l and ρ_m . However, this equation does not obey the rule-of-thumb of rocketeers that the optimum width-to-length ratio for a streamer would be 1:10. According to equation (C.3), the maximum drag for a fixed surface area is obtained at the limit $l \rightarrow 0$, $w \rightarrow \infty$. In practice the rocket dimensions limit the practical dimensions of a streamer, from which the 1:10 rule-of-thumb may arise.

Equation validation

To test the validity of the equation, several additional streamers were measured for their drag coefficients. These were of various materials and of dimensions that were not used in the fitting of the empirical formulae. These can therefore be used as an independent test set for validating equation (C.6).

Table C.2 presents the tested streamers and their measured and estimated normalized drag coefficients. The results show relative errors in the range of 12–27%. While rather high, they are considered a good result for estimating such a random and dynamic process as a streamer. Furthermore, due to the proportionality to the square of the velocity, a 25% error in the normalized force coefficient translates to a 10–15% error in the rocket's descent velocity. This still allows the rocket designer to get a good estimate on how fast a rocket will descend with a particular streamer.

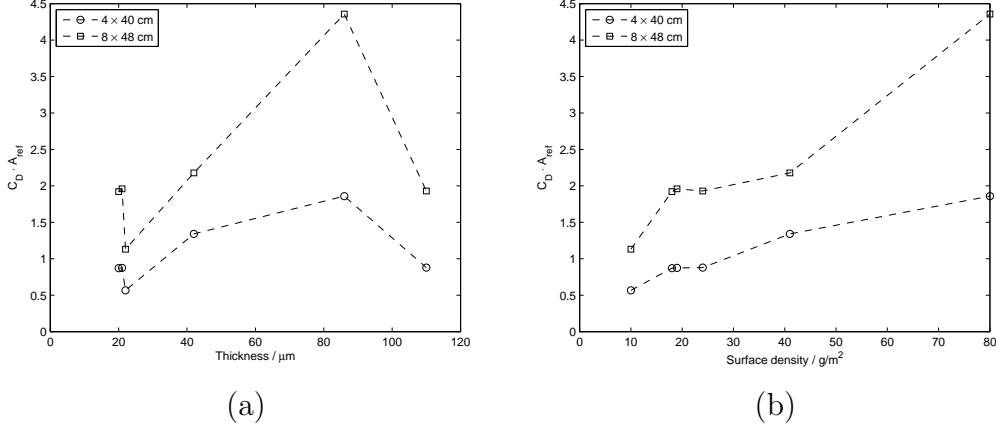


Figure C.3: The normalized drag coefficient of a streamer as a function of (a) the material thickness and (b) the material surface density.

Table C.1: Properties of the streamer materials experimented with.

Material	Thickness / μm	Density / g/m^2
Polyethylene	21	19
Polyethylene	22	10
Polyethylene	42	41
Polyethylene	86	80
Cellophane	20	18
Crêpe paper	110 [†]	24

[†] Dependent on the amount of pressure applied.

Table C.2: Streamers used in validation and their results.

Material	Width m	Length m	Density g/m^2	Measured $10^{-3}(C_D \cdot A_{ref})$	Estimate	Error
Polyethylene	0.07	0.21	21	0.99	1.26	27%
Polyethylene	0.07	0.49	41	1.81	2.23	23%
Polyethylene	0.08	0.24	10	0.89	1.12	26%
Cellophane	0.06	0.70	20	1.78	1.49	17%
Crêpe paper	0.06	0.50	24	1.27	1.43	12%