

A Stochastic Six-Degree-of-Freedom Flight Simulator for Passively Controlled High Power Rockets

Simon Box*, Christopher M. Bishop†, Hugh Hunt‡

This is a pre-publication draft of the paper in the Journal of Aerospace Engineering V. 24(1) pp. 31-45 (2011) doi:10.1061/ASCEAS.1943-5525.0000051

Abstract

This paper presents a method for simulating the flight of a passively controlled rocket in six degrees of freedom, and the descent under parachute in three degrees of freedom. Also presented is a method for modeling the uncertainty in both the rocket dynamics and the atmospheric conditions using stochastic parameters and the Monte-Carlo method. Included within this we present a method for *quantifying* the uncertainty in the atmospheric conditions using historical atmospheric data.

The core simulation algorithm is a numerical integration of the rocket's equations of motion using the Runge-Kutta-Fehlberg method. The position of the rocket's centre of mass is described using three dimensional Cartesian coordinates and the rocket's orientation is described using quaternions.

Input parameters to the simulator are made stochastic by adding Gaussian noise. In the case of atmospheric parameters the variance of the noise is a function of altitude and noise at adjacent altitudes is correlated. The core simulation algorithm, with stochastic parameters, is run within a Monte Carlo wrapper to evaluate the overall uncertainty in the rocket's flight path.

The results of a demonstration of the simulator, where it was used to predict the flight of real rocket, show the rocket landing within the 1σ area predicted by the simulation. Also lateral acceleration during weather cocking, which was measured in the test, shows a strong correlation with simulated values.

1 Introduction

1.1 Background

High power rocketry (HPR) is both a popular hobby for amateur enthusiasts and an academic activity with a number of universities using HPR as a teaching and research tool.

*Microsoft Research Cambridge, now at University of Southampton [s.box@soton.ac.uk]

†Microsoft Research Cambridge

‡University of Cambridge

A high power rocket is defined as a rocket with total impulse of between 160Ns and 40,960Ns NFPA [2002]. Solid fuel HPR motors consisting of an ammonium perchlorate (NH_4ClO_4) and powdered aluminum (Al) mix are sold commercially. Some rockets are alternatively powered by hybrid motors using a liquid oxidizer e.g. Nitrous Oxide (N_2O) and a solid fuel e.g. hydroxyl-terminated polybutadiene (HTPB).

Typically the rockets are passively controlled and fly to a maximum altitude of between 1 and 5 km, but rockets have been flown much higher ($\sim 13\text{ km}$)¹. At apogee it is usual for the rockets to deploy a parachute for safe recovery to earth. The rockets generally carry some avionics and sensors payload to record flight data. They may also carry additional sensors, for example, to record atmospheric measurements.

Rockets are flown at scheduled meetings organized by rocketry clubs and regulated by national rocketry organizations such as the UK Rocketry Association² (UKRA) and, in the US, the National Association of Rocketry³ (NAR). The websites of these organizations are good resources for more information on HPR.

1.2 Motivation

There are two principal motivations for a stochastic six-degree-of-freedom flight simulator for passively controlled rockets. The first is as a tool for predicting the landing location and the second is as a tool for design.

As the rockets are passively controlled the flier cannot control the landing location of the rocket after it has been launched. Therefore accurate predictions of the rocket's landing location are important for safe flying. The stochastic element of the simulator is particularly important for this application as it makes it possible to quantify the uncertainty in the landing position and the probability that the rocket will land in a given area.

As a design tool the a six-degree-of-freedom flight simulator allows the rocket designer to fly prototype rocket designs virtually to assess performance and optimize aspects of design such as the margin of stability (defined in §1.5.1).

1.3 Contents

This paper presents a methodology for simulating HPR flights using a six-degree-of-freedom simulator for the rocket ascent and a three-degree-of freedom simulator for the parachute descent. It is assumed that the rocket is an axisymmetric rigid body and is passively controlled. It is also assumed that the dynamic and aerodynamic properties of the rocket and parachute are known and can be supplied as inputs to the simulator. These properties can be found through experimentation (e.g. using models in a wind tunnel), or they can be estimated from the rocket/parachute geometry. A method for the latter is presented in [Box et al., 2009].

In Section 2 we begin by describing the software architecture used for the core simulation algorithms of the rocket simulator and the parachute simulator. We then proceed to show

¹http://www.canadianrocketry.org/records_hpr.php

²<http://www.ukra.org.uk/>

³<http://www.nar.org/>

how this architecture can be extended to simulate different flight scenarios, such as multi-stage flights.

In Section 3 we describe in detail the dynamic models that are used by the simulator to solve the rocket and parachute equations of motion. This includes descriptions of all the dynamic and aerodynamic data that must be supplied to the simulator and step by step solutions to the equations of motion.

Up to the end of Section 3 the simulation method described is entirely deterministic. In Section 4 we show how to extend the simulator to perform stochastic simulations using the Monte Carlo method. Some of the most important stochastic parameters in the simulation are those describing atmospheric conditions, in particular the wind speed. The wind speed and direction have a significant effect on the flight path of an HPR rocket. So we present a method for quantifying the uncertainty in the forecast wind speed data and modeling this uncertainty in the stochastic simulations.

The simulation method presented in this paper has not been fully verified experimentally. However, in Section 5 we present a demonstration of the simulator where simulation output is compared with data recorded during an HPR rocket flight. The results of this early test indicate that the method could be useful.

In the remainder of this introduction we present a brief explanation of how this research fits within the context of previous work, and a note on passive control aerodynamics.

1.4 Context

Previous work, e.g. [Duncan and Ensey, 1964, Nassiri et al., 2004], has shown how to formulate the six-degree-of-freedom equations of motion for a passively controlled rocket and demonstrated the use of the Monte Carlo method to probe the sensitivity of the flight path to variation in the rocket's dynamic parameters. The main contributions of this paper are enumerated below.

1. An updated formulation of the rocket's equations of motion accounting for modern computational techniques. In particular we use quaternions to describe the rocket's rotational orientation; the computational advantages of quaternions for rigid body dynamics simulation are described in [Baraff, 1997].
2. A software architecture for combining rocket and parachute models to simulate rocket flights with parachute recovery, including multi-stage flights, flights with parachute failure and stochastic flight simulations.
3. A quantitative method for estimating the uncertainty in the atmospheric conditions, where the variance in both wind speed and wind direction is a function of altitude and correlated at adjacent altitudes.

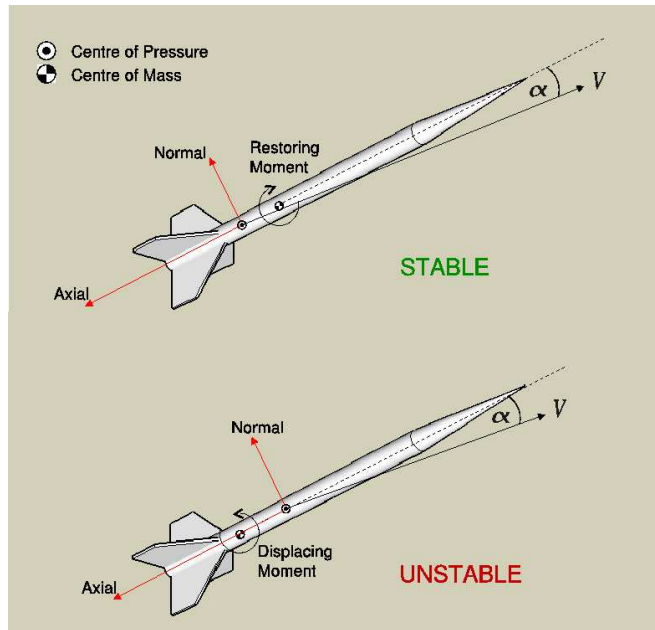


Figure 1: Diagrams of directionally stable (*top*) and directionally unstable (*bottom*) rockets. The atmosphere relative velocity vector is marked V and the aerodynamic force is shown in two orthogonal components *axial* and *normal*. The angle α is the *angle of attack*.

1.5 Passive Control Aerodynamics

1.5.1 Directional Stability

A *passively controlled* or unguided rocket derives its stability from fins like a dart or an arrow. There is no active control or steering to correct or adjust the rocket's trajectory after launch. The addition of fins moves the centre of pressure towards the rear of the rocket. The centre of pressure is the point on the rocket through which all aerodynamic forces can be assumed to act. In order for the rocket flight to be stable the centre of pressure must be aft of the centre of mass (Figure 1). The distance between the centres of pressure and mass is the *margin of stability*.

The angle between the direction of airflow over the rocket and the rocket's roll axis is known as the *angle of attack* α . In the case of a stable rocket the aerodynamic forces will act to reduce the angle of attack to zero, but in the case of an unstable rocket the opposite is true.

1.5.2 Weather Cocking

During the launching stage of a passively controlled rocket flight the trajectory of the rocket is constrained by a launch tower. This allows the rocket to build up some speed and hence aerodynamic stability before the constraint is removed. As the rocket clears the launch tower, and if there is a cross-wind, the rocket will be traveling with an angle of attack. The effects of the directional stability will cause the rocket to rotate into the cross-wind and reduce the angle of attack. The angular momentum of the rocket will then cause an over-rotation leading to a

characteristic damped oscillation in the rocket’s angular position.

This phenomenon is known as weather cocking and, as will be shown in Section 5, it is important when analyzing the efficacy of the simulator.

2 Simulator Architecture

In this section we present a graphical overview of the software architecture using block diagrams. We first describe the core simulation algorithms that are used to integrate numerically the equations of motion for the rocket and the parachute. Then show how these can be used to construct different flight scenarios.

2.1 Core simulation algorithms

The core rocket simulator works by numerically integrating the rocket’s equations of motion over time using the 4th/5th order Runge-Kutta-Fehlberg algorithm. This algorithm is well described in numerical computing literature, a good example can be found in [Press et al., 2007].

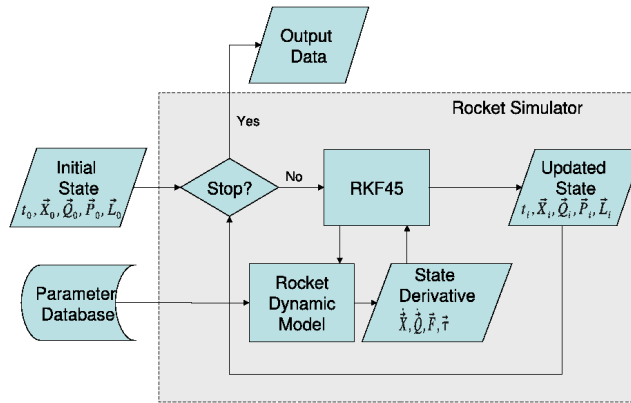


Figure 2: Code block diagram for the core rocket simulation routine.

A block diagram summarizing the rocket simulator is shown in Figure 2. The state of the rocket at any time during the simulation is described by four vectors: \vec{X} is a vector describing the position of the rocket’s centre of mass in global Cartesian coordinates, which are aligned with a tangent plane to the Earth’s surface at the launch site. \vec{Q} is a quaternion describing the rocket’s orientation, \vec{P} and \vec{L} are vectors describing the rocket’s linear and angular momentum respectively.

The initial time t_0 and state $(\vec{X}_0, \vec{Q}_0, \vec{P}_0, \vec{L}_0)$ are passed to the Runge-Kutta-Fehlberg block, labeled RKF45 in Figure 2. The RKF45 block passes values of time and state to the rocket dynamic model, which solves the equations of motion to get the derivatives of the four state vectors with respect to time. To do this the model requires additional data. These are the parameters describing the dynamic and aerodynamic properties of the rocket, the thrust of the rocket and the atmospheric conditions. The model gets these data from the *parameters*

database. The state derivatives are: the linear velocity $\dot{\vec{X}}$, the quaternion derivative $\dot{\vec{Q}}$, the Force \vec{F} and the Torque $\vec{\tau}$. These are passed back to RKF45. A detailed description of the rocket dynamic model and parameters data are given in Section 3.1.

When the RKF45 algorithm completes it's time-step it returns a new time t_i and new state $(\vec{X}_i, \vec{Q}_i, \vec{P}_i, \vec{L}_i)$. The new state is in turn passed back to RKF45 as the starting state for the next step. This loop continues until the stop condition is satisfied. Various stop conditions can be used. For example to make the simulation terminate when the rocket reaches apogee, the vertical momentum of the rocket can be used as a stop trigger.

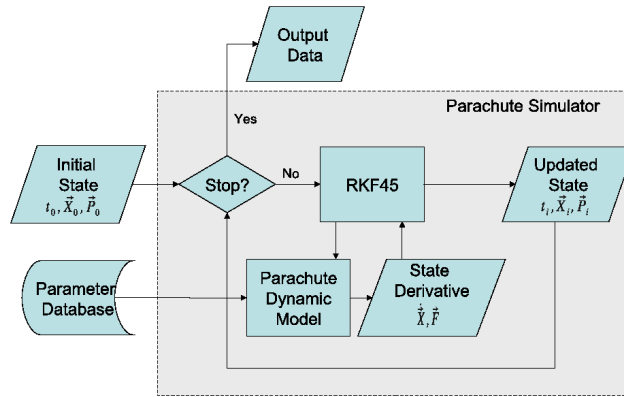


Figure 3: Code block diagram for the core parachute simulation routine.

The parachute simulator (Figure 3) works in a very similar way to the rocket simulator and the parachute dynamic model and parameters are described in Section 3.2. One important difference is that rotations are not modeled in the parachute simulation so there are only linear state vectors (\vec{X}, \vec{P}) and state derivative vectors $(\dot{\vec{X}}, \vec{F})$.

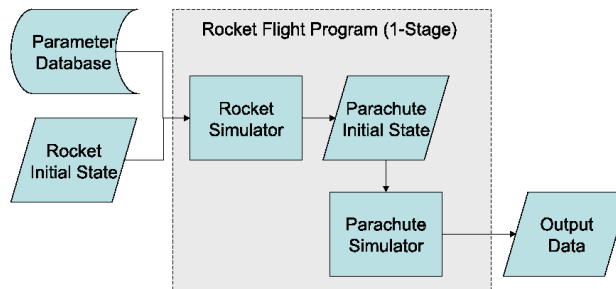


Figure 4: Code block diagram for a simple rocket flight program.

2.2 Full flight simulation

The rocket and parachute simulators depicted in Figures 2 and 3 can be combined together to simulate a complete flight. The block diagram for a simple flight is shown in Figure 4. The

rocket simulator is run first and terminates when the rocket reaches apogee. Then the final state of the rocket simulation is passed to the parachute simulation as the initial state. The rocket's descent under parachute is then simulated until it reaches the ground.

An example flight path from this type of simulation is shown in Figure 5. The launch pad coordinates are $[0, 0, 0]$ and the wind direction is predominantly from the south-west. It can be seen that during the ascent the rocket heads upwind a short distance as it climbs to 3,500 m. At apogee a drogue parachute is deployed and the rocket is blown back downwind as it descends to an altitude of 300 m. Here a second (main) parachute is deployed and the increase in drag can be seen in the path of the rocket. Finally the rocket lands approximately 850 m north and 650 m east of the launch pad.

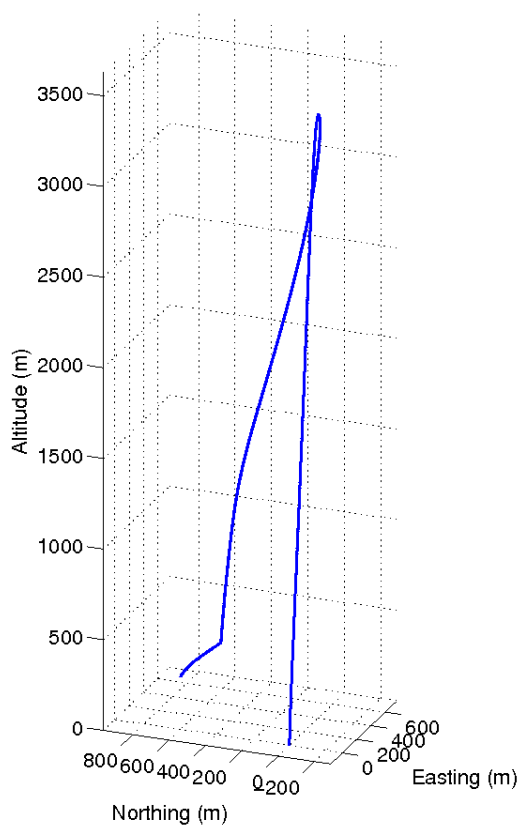


Figure 5: Plot of a simulated rocket flight path generated using the program shown in figure 4.

2.3 More complex flight scenarios

In addition to the simple scenario shown in in Figure 4 more complex flight scenarios can be constructed using the rocket and parachute simulators as building blocks. Two examples are the simulation of a parachute failure and the simulation of a two stage rocket. These are discussed below.

A parachute deployment failure during an HPR rocket flight can pose a safety hazard,

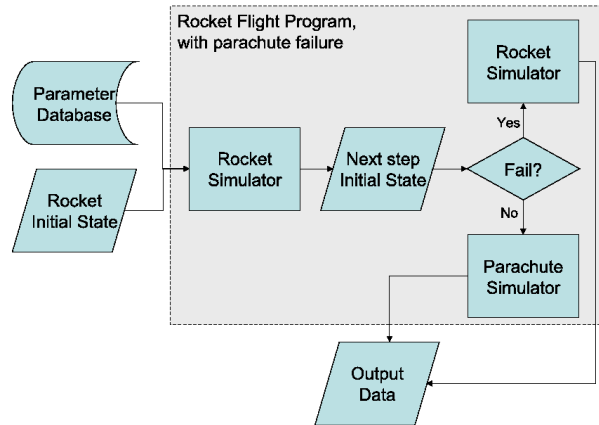


Figure 6: Code block diagram for a rocket flight program with parachute failure.

therefore it is useful to be able to simulate this scenario. Figure 6 shows the block diagram for a full flight simulation which includes the possibility of a parachute deployment failure. In this case the rocket simulation terminates when the rocket reaches apogee, then there is a decision over whether the parachute deployment will fail. This choice can be made through user input or it can be random with an assigned probability. If the decision is for a successful parachute deployment then the final state of the rocket at apogee is passed to the parachute simulator as the initial state. This is effectively the same scenario as shown in Figure 4. However if the parachute deployment fails then the final state is passed to the rocket simulator again and a ballistic descent is simulated.

Figure 7 shows example flight paths for the two scenarios. Note that this approach can only simulate a complete parachute deployment failure i.e. the parachute stays completely within the rocket. In order to model a partial deployment a new dynamic model would be required.

The block diagram for a two stage rocket flight is shown in Figure 8. In this case the database contains dynamic, aerodynamic and thrust data for the rocket in three configurations, firstly the complete rocket as it is on the launch pad and then the upper stage and the booster stage after separation. The first rocket simulation covers the flight of the complete rocket from launch up to the time of separation. At stage separation new initial states for the upper and booster stages are calculated from the state prior to separation. The flights of the upper and booster stages are then simulated to their respective apogees. Following this the two parachute descents are simulated as usual. Figure 9 shows an example of the simulated flight paths for a two stage rocket flight.

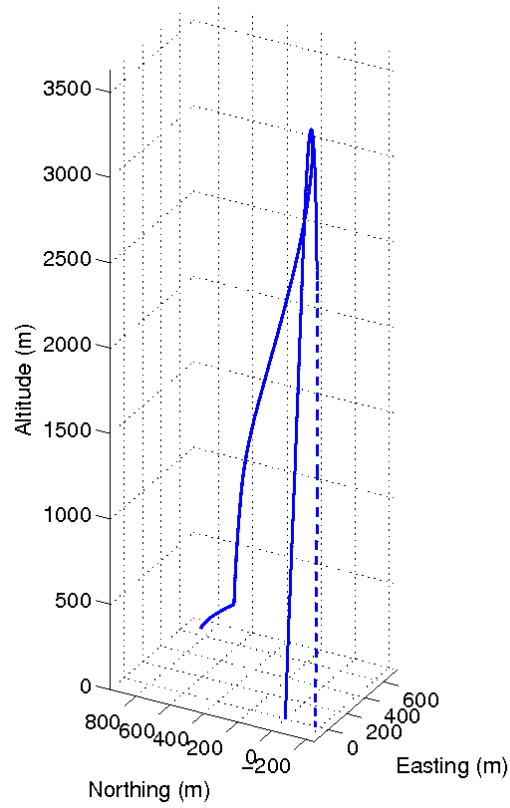


Figure 7: Plot of two simulated rocket flight paths, one for a successful flight (*solid*) and one with parachute failure (*dotted*).

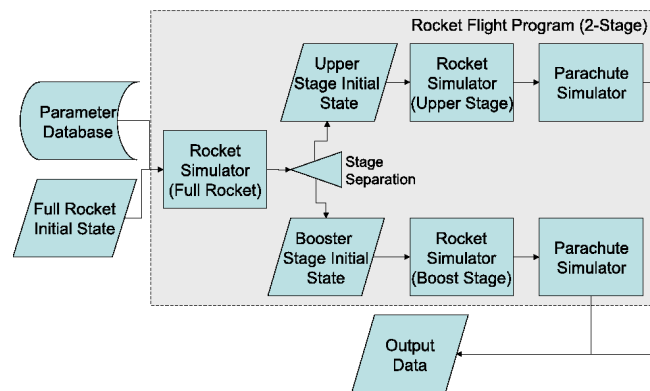


Figure 8: Code block diagram for a 2-stage rocket flight program.

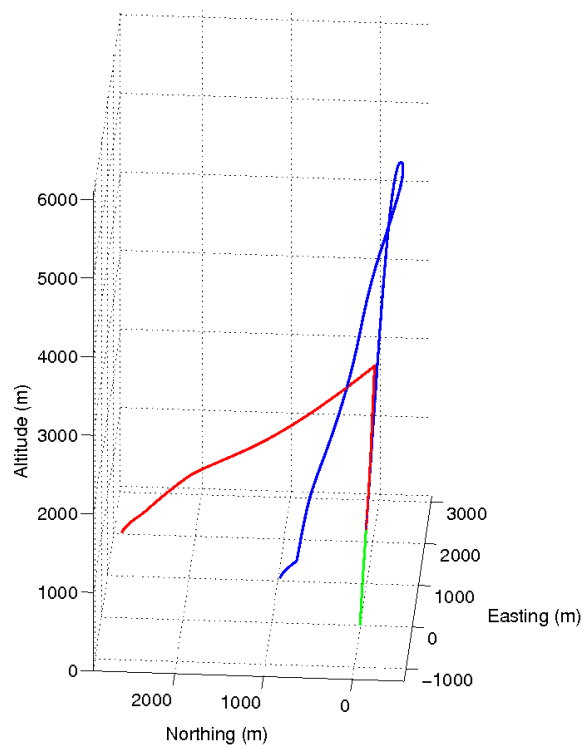


Figure 9: Plot of the simulated flight path for a 2-stage rocket. After separation the paths of both the upper and booster stages are shown.

3 The Dynamic Models

3.1 Rocket Dynamic Model

This section describes in detail the rocket dynamic model introduced in Section 2.1. The task of the model is to solve the rocket equations of motion. That is to take the rocket's current time t and state $(\vec{X}, \vec{Q}, \vec{P}, \vec{L})$ and calculate the state derivatives $(\dot{\vec{X}}, \dot{\vec{Q}}, \dot{\vec{P}}, \dot{\vec{L}})$.

We begin by defining the inputs to the dynamic model and the constant values used in the calculation.

3.1.1 Constants and Inputs

For reference, Table 1 shows all the constants that are used in the rocket dynamic model.

Symbol	Value	Description
γ	1.4	Ratio of specific heats for air
R	287	Gas constant for air (difference in specific heats)
Θ_0	291.15 K	Reference temperature
μ_0	$1.827 \times 10^{-7} \text{ Pa} \cdot \text{s}$	Reference dynamic viscosity
C	120 K	Sutherland's constant
M_E	$5.974 \times 10^{24} \text{ kg}$	Mass of the Earth
r_E	6378100 m	Radius of the Earth
G	$6.673 \times 10^{-11} \frac{\text{m}^3}{\text{kg} \cdot \text{s}^2}$	Universal gravitational constant
$\vec{Y}_{A,0}$	[1, 0, 0]	Reference yaw axis
$\vec{P}_{A,0}$	[0, 1, 0]	Reference pitch axis
$\vec{R}_{A,0}$	[0, 0, 1]	Reference roll axis

Table 1: Constants used used in the dynamic models.

There are two types of input to the rocket dynamic model, as can be seen in Figure 2: the current state and the parameters. The current state vectors are summarized in Table 2.

Symbol	Elements	Description
t		Time
\vec{P}	$[P_x, P_y, P_z]$	Linear momentum vector
\vec{L}	$[L_x, L_y, L_z]$	Angular momentum vector
\vec{Q}	$[s, v_x, v_y, v_z]$	Quaternion
\vec{X}	$[x, y, z]$	Position Vector

Table 2: Rocket dynamic model state vectors.

Table 3 summarizes the parameters that are stored in the database. Most of these data are a function of some quantity like time t or altitude z therefore they are stored in tables and the values can be interpolated as required by the model. The aerodynamic coefficients C_A ,

C_N and C_R can be functions of up to three quantities (Reynolds number (Re), angle of attack α and Mach number (Ma)). However some methods for estimating these coefficients assume that they are independent of one or more of these quantities. For example a common method of estimating C_N described in [Box et al., 2009] assumes that C_N is independent of Reynolds number Re.

Symbol	Function of	Description
T	t	Thrust
M	t	Mass
X_{cm}	t	Distance of the centre of mass from the nose tip
I_{xx}	t	Moments of inertia about the rocket's yaw axis
I_{yy}	t	Moments of inertia about the rocket's pitch axis
I_{zz}	t	Moments of inertia about the rocket's roll axis
C_{da}	t	Thrust damping coefficient
C_A	Re, α , Ma	Coefficient of axial aerodynamic force
C_N	Re, α , Ma	Coefficient of normal force
C_R	Re, α_f , Ma	Coefficient of aerodynamic roll torque
X_{cp}	α	Distance of the centre of pressure from the nose tip
\vec{W}	x, y, z	Wind speed vector
ρ	x, y, z	Atmospheric Density
Θ_A	x, y, z	Atmospheric Temperature
X_{RB}	—	Rocket body length
A_{RB}	—	Rocket body cross-sectional area (maximum)
X_f	—	distance of the plane of the fin's centres from the nose tip
r_f	—	roll moment arm
η	—	fin cant angle

Table 3: Parameters that are stored in the database for the rocket model.

3.1.2 Calculating the Derivatives

To calculate the state derivatives the rocket dynamic model uses the current time and state to access the database and get the values of thrust, mass, aerodynamic coefficients, wind speed etc, that apply to that time and state. This section presents the equations for calculating the state derivatives with these data. The method assumes that the rocket is axisymmetric.

Position and Orientation The position of the rocket's centre of mass in global Cartesian coordinates is given by \vec{X} . The quaternion vector \vec{Q} describes the rocket's rotational orientation. Specifically \vec{Q} describes a rotational transformation between a reference orientation and the current orientation. The transformation can be described as a rotation of θ radians about a rotation axis \vec{a} passing through the centre of mass of the rocket. The elements of the quaternion vector are $\vec{Q} = [s, \vec{v}] = [s, v_x, v_y, v_z]$ where

$$\begin{aligned} s &= \cos\left(\frac{\theta}{2}\right) \\ v_x &= \sin\left(\frac{\theta}{2}\right) a_x \\ v_y &= \sin\left(\frac{\theta}{2}\right) a_y \\ v_z &= \sin\left(\frac{\theta}{2}\right) a_z \end{aligned} \tag{1}$$

\vec{Q} can be converted to a rotation matrix \mathbf{R} using the following transformation

$$\mathbf{R} = \begin{bmatrix} 1 - 2v_y^2 - 2v_z^2 & 2v_xv_y - 2sv_z & 2v_xv_z + 2sv_y \\ 2v_xv_y + 2sv_z & 1 - 2v_x^2 - 2v_z^2 & 2v_yv_z - 2sv_x \\ 2v_xv_z - 2sv_y & 2v_yv_z + 2sv_x & 1 - 2v_x^2 - 2v_y^2 \end{bmatrix} \tag{2}$$

The unit vectors describing the yaw, pitch and roll axes of the rocket in it's current orientation can be calculated using \mathbf{R} and the reference orientations (table 1).

$$\begin{aligned} \vec{Y}_A &= \mathbf{R}\vec{Y}_{A,0}^T \\ \vec{P}_A &= \mathbf{R}\vec{P}_{A,0}^T \\ \vec{R}_A &= \mathbf{R}\vec{R}_{A,0}^T \end{aligned} \tag{3}$$

Linear and Angular Velocity The Earth-relative linear velocity vector of the rocket's centre of mass is given by

$$\dot{\vec{X}} = \vec{P}/M. \tag{4}$$

The angular velocity vector ($\vec{\omega}$) for the rocket is calculated using

$$\vec{\omega} = \mathbf{R}\mathcal{L}_0^{-1}\mathbf{R}^T\vec{L}^T \tag{5}$$

where \mathcal{I}_0 is the reference inertia tensor, defined using values for the rocket's moments of inertia from the database

$$\mathcal{I}_0 = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \quad (6)$$

The angular velocity vector $\vec{\omega}$ and the quaternion \vec{Q} are used to calculate the quaternion derivative $\dot{\vec{Q}}$.

$$\dot{s} = \frac{1}{2}(\vec{\omega} \cdot \vec{v}) \quad (7)$$

$$\dot{\vec{v}} = \frac{1}{2}(s\vec{\omega} + (\vec{\omega} \times \vec{v})) \quad (8)$$

$$\dot{\vec{Q}} = \begin{bmatrix} \dot{s} \\ \dot{\vec{v}} \end{bmatrix} \quad (9)$$

Angle of attack, Reynolds No. and Mach No. In order to calculate the forces and torques on the rocket we need to recover the aerodynamic coefficients from the database. For this we must know the angle of attack, the Reynolds number and the Mach number of the rocket in it's current state.

The angle of attack α is defined as the angle between the unit vector describing the rocket's roll axis and the rocket's apparent velocity vector (\vec{V}), so it is given by

$$\alpha = \cos^{-1}(\hat{V} \cdot \vec{R}_A) \quad (10)$$

where the $\hat{\cdot}$ symbol in \hat{V} indicates that the vector \vec{V} has been normalized.

The apparent velocity vector \vec{V} is the velocity of the rocket's centre of pressure relative to the atmosphere. Unfortunately the location of the centre of pressure X_{cp} must be recovered from the database and may itself be a function of α . If this is the case then the atmosphere relative velocity of the centre of mass (\vec{V}_{cm}) can be used as an approximation of \vec{V} . This is given by

$$\vec{V}_{cm} = \dot{\vec{X}} + \vec{W} \quad (11)$$

This gives an approximate α (α_{cm}) which neglects the effects of the rocket's own rotation on α . Provided the rocket's angular velocity is small compared with it's forward velocity this approximation is good enough to select X_{cp} from the database.

Then the apparent velocity \vec{V} is given by

$$\vec{V} = \vec{V}_{cm} + \vec{V}_\omega \quad (12)$$

where \vec{V}_ω is the linear velocity vector of the centre of pressure due to the angular velocity of the rocket and is given by

$$\vec{V}_\omega = \bar{X} \sin\left(\cos^{-1}\left(\vec{R}_A \cdot \hat{\omega}\right)\right) \left(\vec{R}_A \times \vec{\omega}\right) \quad (13)$$

Here the direction of the vector is given by the cross product of the unit vectors for the roll axis and the axis of rotation; and the magnitude is the angular velocity multiplied by the perpendicular distance between the centre of pressure and the axis of rotation, given by

$\bar{X} \sin \left(\cos^{-1} \left(\vec{R}_A \cdot \hat{\omega} \right) \right)$ where $\bar{X} = |X_{cp} - X_{cm}|$ and $\hat{\omega}$ is the normalized angular velocity vector.

If there *is* a large difference between α and α_{cm} then the estimated value of α can be used to extract a new value of X_{cp} from the database and then an updated α can be calculated, thus α can be found iteratively.

The Reynolds number of the rocket is given by

$$\text{Re} = \frac{\rho V X_{RB}}{\mu} \quad (14)$$

where $V = |\vec{V}|$, X_{RB} is the length of the rocket body and μ is the kinematic viscosity given by

$$\mu = \mu_0 \frac{\Theta_0 + C}{\Theta_A + C} \left(\frac{\Theta_A}{\Theta_0} \right)^{\frac{3}{2}} \quad (15)$$

The Mach number of the rocket is given by

$$\text{Ma} = \frac{V}{\sqrt{\gamma R \Theta_A}} \quad (16)$$

where $\sqrt{\gamma R \Theta_A}$ is the local speed of sound.

Fin angle of attack In some rocket designs the fins are canted with a small angle η to induce roll. If the fins are canted then even if the angle of attack of the rocket is zero the *fins* will have an angle of attack α_f equal to the cant angle. This will cause the rocket to increase roll velocity until the fin angle of attack is zero.

The centres of pressure for each of the rockets fins lie on a circle which can be described by the distance of it's centre from the nose tip X_f and it's radius r_f . The plane of the circle is perpendicular to the rocket's axis. Because the rocket may have an angle of attack and angular velocity, the α_f of each of the fins may be different. In order to maintain axisymmetry and because we are unconcerned with the number of fins, we can estimate a mean fin angle of attack α_f by finding the angle of attack at N evenly spaced points on the circle and taking an average. For the results presented in this paper $N = 4$;

If \vec{P}_i^b is a point on the circle in rocket reference coordinates then the coordinates of the point in space are

$$\vec{P}_i = \mathbf{R} \vec{P}_i^b + \vec{X}. \quad (17)$$

The linear velocity of this point due the rocket's angular velocity is given by

$$\vec{V}_{p,\omega,i} = |\vec{\omega}| |\vec{S}_i| \sin \left(\cos^{-1} \left(\hat{S}_i \cdot \hat{\omega} \right) \right) \left(\hat{S}_i \times \hat{\omega} \right) \quad (18)$$

where $\vec{S}_i = \vec{X} - \vec{P}_i$. The total velocity vector for the point P_i is then given by

$$\vec{V}_{p,i} = \vec{V}_{p,\omega,i} + \dot{\vec{X}} + \vec{W}. \quad (19)$$

If \vec{l}_i is a unit vector along the shortest path from the rocket's axis to \vec{P}_i then we can define the quaternion that describes the cant angle of an imaginary fin at point \vec{P}_i .

$$\vec{Q}_c = \left[\cos \frac{\eta}{2}, \sin \frac{\eta}{2} (\vec{l}_i \times \vec{R}_A) \right], \quad (20)$$

which using (2) gives \mathbf{R}_c . Then the fin angle of attack at point \vec{P}_i is given by

$$\alpha_{f,i} = \frac{\pi}{2} - \cos^{-1} \left(\hat{V}_{p,i} \cdot \mathbf{R}_c \vec{l}_i \right). \quad (21)$$

Unlike rocket angle of attack α fin angle of attack α_f , as defined in this model, can be either positive or negative depending on which side of the fin it is.

The mean fin angle of attack and mean fin velocities can then be calculated over the N points used.

$$\alpha_f = \frac{1}{N} \sum_{i=1}^N \alpha_{f,i} \quad (22)$$

$$V_f = \frac{1}{N} \sum_{i=1}^N |\vec{V}_{p,i}| \quad (23)$$

α_f is used to recover the correct value of the coefficient of roll C_R from the parameters database and V_f is used in the calculation of roll torque in (35).

Force and Torque The force vector on the rocket can be expressed as the sum of four component vectors.

$$\vec{F} = \vec{F}_T + \vec{F}_g + \vec{F}_A + \vec{F}_N \quad (24)$$

\vec{F}_T is the thrust vector, which acts in the opposite direction of the roll axis, so is given by

$$\vec{F}_T = -T \vec{R}_A \quad (25)$$

\vec{F}_g is the gravity vector, which is assumed to be

$$\vec{F}_g = [0, 0, -Mg]^T \quad (26)$$

where g is calculated using

$$g = \frac{M_E}{(r_E + z)^2} \quad (27)$$

\vec{F}_A and \vec{F}_N are the aerodynamic force on the rocket broken down into axial and normal components respectively. The magnitude of the axial aerodynamic force is

$$F_A = \frac{1}{2} \rho V^2 A_{RB} C_A \quad (28)$$

where $V = |\vec{V}|$. The axial force vector acts in the opposite direction to the vector of the rocket's roll axis, so the force vector is given by

$$\vec{F}_A = -F_A \vec{R}_A \quad (29)$$

The magnitude of the normal aerodynamic force is

$$F_N = \frac{1}{2} \rho V^2 A_{RB} C_N \quad (30)$$

The normal aerodynamic force acts in a direction that is orthogonal to the roll axis \vec{R}_A and in the plane formed by the roll axis and the apparent velocity vector \vec{V} .

$$\vec{F}_N = F_N \left(\vec{R}_A \times \left(\vec{R}_A \times \hat{V} \right) \right) \quad (31)$$

The torque vector on the rocket can be expressed as the sum of three component vectors

$$\vec{\tau} = \vec{\tau}_N + \vec{\tau}_{da} + \vec{\tau}_R \quad (32)$$

$\vec{\tau}_N$ is the torque on the rocket due to the normal force is calculated as

$$\vec{\tau}_N = F_N \bar{X} \left(\vec{R}_A \times \hat{V} \right) \quad (33)$$

where $\bar{X} = |X_{cp} - X_{cm}|$ is the moment arm.

When a rocket rotates about a transverse axis during the thrusting phase of flight, hot gas within the motor tube will be accelerated laterally. This produces a damping moment called *thrust damping*. The torque on the rocket due to thrust damping $\vec{\tau}_{da}$ is modeled by

$$\vec{\tau}_{da} = -C_{da} \mathbf{R} \mathbf{m} \mathbf{R}^{-1} \vec{\omega} \quad (34)$$

where \mathbf{m} is a diagonal matrix with elements [1 1 0] on the main diagonal.

$\vec{\tau}_R$ is the roll torque on the rocket due to fin cant, the magnitude of this torque can be expressed as

$$\tau_R = \frac{1}{2} \rho V_f^2 A_{RB} C_R r_f \quad (35)$$

where r_f is the roll moment arm and C_R can be positive or negative describing clockwise or anticlockwise roll respectively. The roll torque is about the rocket's roll axis, therefore the torque vector is defined by

$$\vec{\tau}_R = \tau_R \vec{R}_A \quad (36)$$

In some cases roll torque can be generated by angled thrust from the rocket's nozzle either intentionally or otherwise. A method for modeling this is not presented explicitly here. To account for this an additional term would have to be added to (32).

3.2 Parachute Dynamic Model

3.2.1 Constants and Inputs

The parachute dynamic model works in a very similar way to the rocket model except that rotations are ignored. For examples of more complex parachute models see [Dobrokhodov et al., 2003, Kim and Peskin, 2006]. Tables 4 and 5 define the state vectors and parameters for the model. The parachute model does not use any constants that are not already listed in Table 1.

Table 5 shows the parachute coefficient of drag C_D and parachute area A_P as functions altitude z . This is so that the parachute model can simulate the descent of rockets that deploy two or more parachutes at different altitudes during the descent. The values of C_D and A_P are not interpolated between altitudes as with wind speed or density but rather different values apply to different altitude ranges.

Symbol	Elements	Description
t		Time
\vec{P}	$[P_x, P_y, P_z]$	Linear momentum vector
\vec{X}	$[x, y, z]$	Position Vector

Table 4: Parachute dynamic model state vectors.

Symbol	Function of	Description
\vec{W}	x, y, z	Wind speed vector
ρ	x, y, z	Atmospheric Density
Θ_A	x, y, z	Atmospheric Temperature
C_D	z	Parachute Coefficient of Drag
A_P	z	Parachute Area
M	—	Mass of Rocket and Parachute

Table 5: Parameters that are stored in the database for the parachute model.

3.2.2 Calculating the Derivatives

In this section we show the equations for calculating the state derivatives in the parachute model.

Velocity The earth relative velocity of the parachute $\dot{\vec{X}}$ is given by

$$\dot{\vec{X}} = \vec{P}/M \quad (37)$$

The atmosphere relative velocity (apparent velocity) of the parachute is given by

$$\vec{V} = \dot{\vec{X}} + \vec{W} \quad (38)$$

where \vec{W} is the wind speed vector.

Force The force vector on the parachute can be expressed as the sum of two components

$$\vec{F} = \vec{F}_D + \vec{F}_g \quad (39)$$

\vec{F}_D is the drag force vector, the magnitude of the drag force on the parachute is given by

$$F_D = \frac{1}{2}\rho V^2 C_D A_P \quad (40)$$

The direction of the drag force vector is opposite to the apparent velocity vector.

$$\vec{F}_D = -F_D \vec{V} \quad (41)$$

\vec{F}_g is the gravitational force vector and is assumed to be

$$\vec{F}_g = [0, 0, -Mg]; \quad (42)$$

where g is found using (27).

4 Stochastic simulations using the Monte-Carlo method

In Sections 2 and 3 we have described a simulator for rocket flights which is entirely deterministic. In this section we describe how to perform stochastic simulations and how to quantify the uncertainty in the rocket's landing position by using a Monte-Carlo wrapper around the deterministic simulator.

4.1 Architecture

Many of the inputs to the simulations such as the aerodynamic properties of the rocket and the atmospheric conditions will have uncertain values. If quantitative measures of the uncertainties in these inputs can be known or estimated then the Monte Carlo method can be used to obtain an estimation of the uncertainty in the rocket's flight path and landing position.

The block diagram for the Monte Carlo wrapper is shown in Figure 10. The parameters and initial state are first passed to the Monte-Carlo process block, here uncertain values from

these inputs have random noise added to them generating the random input data. A detailed discussion on this step is presented in Section 4.2. The random input data are then passed to the rocket flight block and a full flight is simulated.

This process is repeated for a pre-defined number of iterations and thus generates a cluster of flight paths and a scatter of landing positions. Figure 11 shows two plots, the left hand plot shows the scatter of 50 simulated landing points using the Monte-Carlo wrapper. Also plotted are Gaussian ellipses marking one and two standard deviations landing probability. The right hand plot shows the same Gaussian ellipses plotted on three dimensional axes together with the mean rocket flight path.

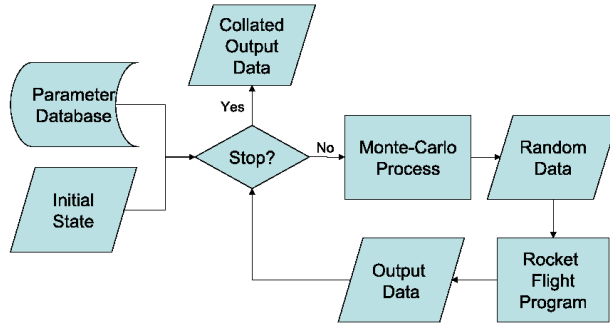


Figure 10: Code block diagram for the Monte-Carlo wrapper.

4.2 Monte-Carlo process

The Monte-Carlo process introduced in the previous section involves taking input parameters with values that are uncertain and making them stochastic by adding random noise. The method employed for doing this in our model is different for the rocket data and the atmospheric data. We describe the two methods below.

4.2.1 Uncertainty in the Rocket Data

Examples of rocket parameters which may be made stochastic are: the aerodynamic coefficients of the rocket and the parachutes, the location of the rocket's centre of pressure, the rocket's mass and centre of mass and the launch tower angles. This list is not exhaustive and in fact all of the parameters can be made stochastic if required.

The method for making a parameter stochastic is to add a random noise term which is sampled from a zero-mean Gaussian distribution. In practice this is done by multiplying the parameter by a noise coefficient ξ which is drawn from a Gaussian distribution with mean 1 and variance σ^2 . Equation (43) shows an example using the coefficient of axial force C_A

$$C_A(+) = C_A \xi, \quad \xi \sim \mathcal{N}(1, \sigma^2) \quad (43)$$

where $C_A(+)$ is the updated value of C_A .

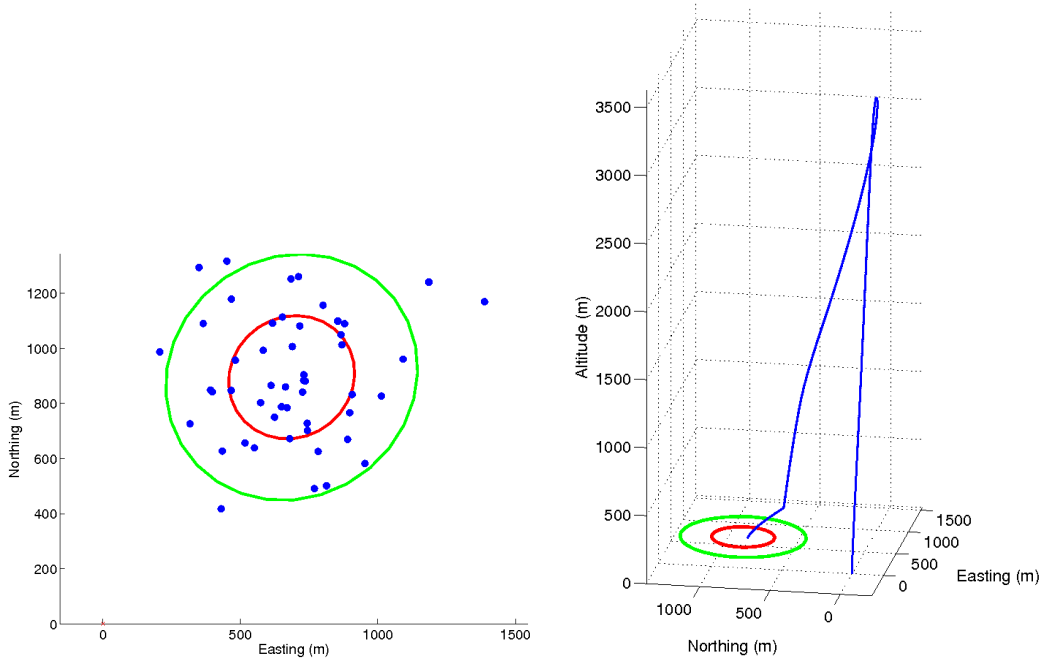


Figure 11: *Left*: Scatter plot of the simulated landing locations after 50 iterations of the Monte-Carlo simulator, including Gaussian ellipses marking 1σ and 2σ probability *Right*: The same Gaussian ellipses plotted together with the mean flight path.

If the stochastic parameter has multiple values in the database, as C_A does, then ξ is sampled once for each simulated flight and used for all values.

The variance of the distribution σ^2 must be carefully chosen to reflect the uncertainty of the corresponding parameter. This variance can be determined experimentally, or estimated from the uncertainty in the measurement or estimation technique that was used to determine the parameter in the first place.

4.2.2 Uncertainty in the Atmospheric Data

In modeling the atmospheric data the approach used in Section 4.2.1 would be too simplistic. Simply sampling from a 1-D Gaussian for wind speed and direction at each altitude increment would ignore the obvious correlation between values at adjacent altitudes. Similarly, sampling once for a wind speed error and then adding that error at every altitude would ignore the fact that perfect knowledge of the wind at one altitude would still leave uncertainty in its value at other altitudes. We must, therefore, sample an entire *profile* of wind speeds from the distribution of such profiles. The sampling of functions is more complex than the sampling of values. If we assume Gaussian uncertainty, then we must sample from a *Gaussian process* [Bishop, 2006]. Here we adopt a relatively simple approach to Gaussian process sampling based on a linear expansion in a set of fixed basis functions.

In order to model the uncertainty in the atmospheric forecast data, we use an approach

which is based on a maximum likelihood estimation using historical forecasts and corresponding historical measurements of the atmospheric conditions.

There are known uncertainties corresponding to both the forecast and measurement data. However these uncertainty data are not readily available to the researcher. Therefore the methodology described here is based on the assumption that the difference between the forecast and measured data is a useful indicator of the uncertainty in the forecast data.

Linearize the difference profile Figure 12a shows a sample plot of the difference between a forecast and measured easterly component of wind speed. A useful method for making a linear approximation of the data in Figure 12a is to use basis functions. Figure 12b shows a number of Gaussian basis functions of the form in (44) super-imposed on the difference profile.

$$\phi_j(z) = \exp \left\{ -\frac{z - \mu_j^2}{2\sigma^2} \right\} \quad (44)$$

where μ is the mean of the basis basis function and σ^2 is the variance.

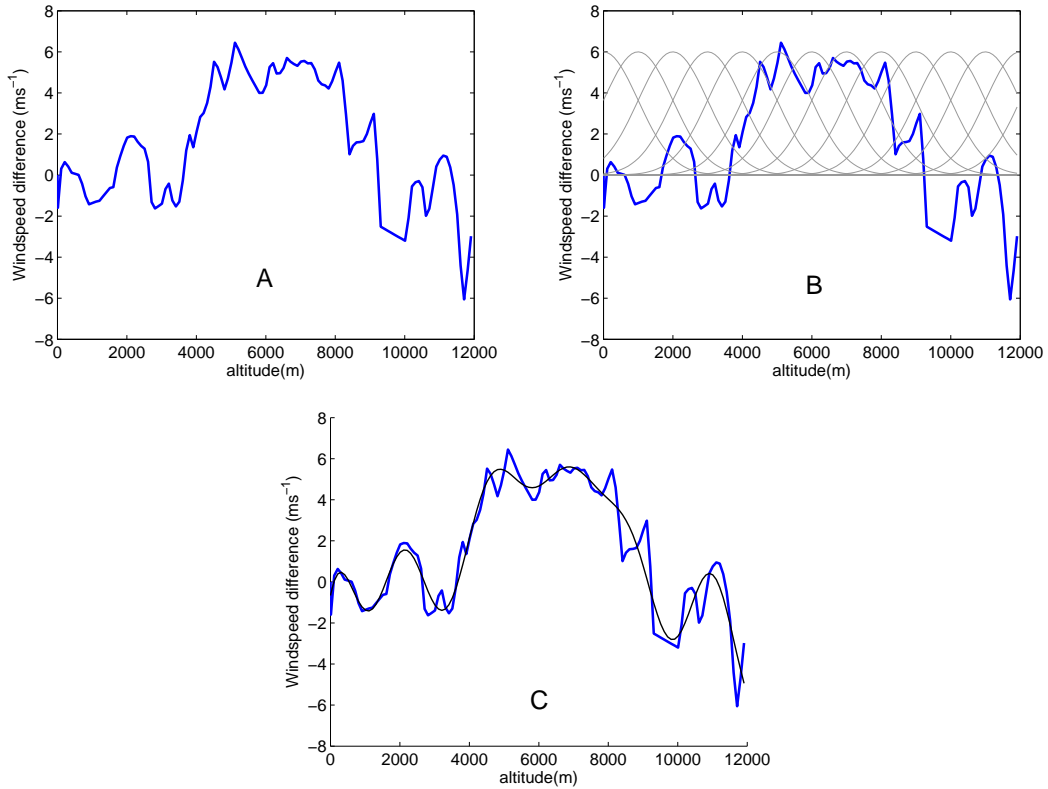


Figure 12: A: Difference between forecast and measured easterly component of wind speed, B: raw Gaussian basis functions superimposed on A, C: sum of weighted basis functions giving a least-squares fit to the difference profile.

By multiplying each of the J basis functions by a scalar w_j and then summing over all J the resulting function can approximate the difference profile as shown in Figure 12c. Thus for

a vector of discrete altitudes \mathbf{z} the difference profile \mathbf{d} can be approximately described using Equation (45) which is a linear function of the vector of scalar weights \mathbf{w} .

$$\mathbf{d} \approx \sum_{j \in J} w_j \phi_j(\mathbf{z}) = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{z}) \quad (45)$$

where \mathbf{w} takes the values that minimize the square error between the real wind speed data \mathbf{d} and the right hand side of (45).

Maximum Likelihood Formulation In a large data set of N difference profiles the probability of each weights vector \mathbf{w}_n is assumed to be Gaussian with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$ (as in (46))

$$p(\mathbf{w}_n) = \mathcal{N}(\mathbf{w}_n | \boldsymbol{\mu}, \boldsymbol{\Sigma}), \quad \forall n \in N \quad (46)$$

where $\boldsymbol{\mu}$ is a $J \times 1$ vector and $\boldsymbol{\Sigma}$ is a $J \times J$ matrix

Using the discrete formulation for an $M \times 1$ vector of altitudes \mathbf{z} , the corresponding vector of difference profile points \mathbf{d}_n is modeled as zero-mean Gaussian white noise added to the function described by the sum of our weighted basis functions i.e.

$$p(\mathbf{d}_n | \mathbf{w}_n) = \mathcal{N}(\mathbf{d}_n | \boldsymbol{\Phi} \mathbf{w}_n, \beta^{-1} \mathbf{I}), \quad \forall n \in N \quad (47)$$

where $\boldsymbol{\Phi}$ is a $M \times J$ matrix with elements $\Phi_{m,j} = \phi_j(z_m)$ (from (44)) and β^{-1} is the variance of the white noise and \mathbf{I} is a $J \times J$ identity matrix.

From Bayes' theorem for Gaussian variables [Bishop, 2006] the marginal distribution of \mathbf{d}_n is given by

$$p(\mathbf{d}_n) = \mathcal{N}(\mathbf{d}_n | \boldsymbol{\Phi} \boldsymbol{\mu}, \beta^{-1} \mathbf{I} + \boldsymbol{\Phi} \boldsymbol{\Sigma} \boldsymbol{\Phi}^T), \quad \forall n \in N \quad (48)$$

and the conditional distribution of \mathbf{w}_n given \mathbf{d}_n is

$$p(\mathbf{w}_n | \mathbf{d}_n) = \mathcal{N}(\mathbf{w}_n | \mathbf{S} (\beta \boldsymbol{\Phi}^T \mathbf{d}_n + \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}), \mathbf{S}), \quad \forall n \in N \quad (49)$$

where

$$\mathbf{S} = (\boldsymbol{\Sigma}^{-1} + \beta \boldsymbol{\Phi}^T \boldsymbol{\Phi})^{-1} \quad (50)$$

The likelihood function \mathcal{L} , which represents the probability of the data given the parameters and viewed as a function of those parameters, is given by

$$\mathcal{L} = \prod_{n \in N} \{p(\mathbf{d}_n | \mathbf{w}_n) p(\mathbf{w}_n)\} \quad (51)$$

By maximizing the likelihood function we can determine the values of the parameters for which the probability of the observed data is maximized. Equivalently we can minimize the log of the likelihood function, this is more convenient both analytically and numerically. It can be shown from (51), (46) and (47) that the Log of the likelihood function is

$$\begin{aligned} \ln \mathcal{L} = & -\frac{\beta}{2} \sum_{n \in N} \{ \mathbf{d}_n^T \mathbf{d}_n - 2 \mathbf{d}_n^T \boldsymbol{\Phi} \mathbf{w}_n + \text{Tr}(\boldsymbol{\Phi}^T \boldsymbol{\Phi} \mathbf{w}_n \mathbf{w}_n^T) \} + \frac{NJ}{2} \ln \beta - \frac{N}{2} \ln |\boldsymbol{\Sigma}| \\ & - \frac{1}{2} \sum_{n \in N} \text{Tr} \{ \boldsymbol{\Sigma}^{-1} (\boldsymbol{\mu} \boldsymbol{\mu}^T - 2 \boldsymbol{\mu} \mathbf{w}_n^T + \mathbf{w}_n \mathbf{w}_n^T) \} \end{aligned} \quad (52)$$

The likelihood is maximized by minimizing (52) with respect to $\boldsymbol{\mu}$, $\boldsymbol{\Sigma}$ and β . The maximum likelihood $\boldsymbol{\mu}$ can be found analytically by maximizing the product over N of Equation (48). This gives

$$\hat{\boldsymbol{\mu}} = \boldsymbol{\Phi}^\dagger \left(\frac{1}{N} \sum_{n \in N} \mathbf{d}_n \right) \quad (53)$$

where \dagger denotes the Moore-Penrose inverse. Because it is unlikely that this approach will uncover any systematic error between measurement and forecast data you would expect that the elements of $\hat{\boldsymbol{\mu}}$ would all be close to zero.

Likelihood Maximization with respect to $\boldsymbol{\Sigma}$ and β^{-1} can be done numerically using the Expectation - Maximization (EM) algorithm. The expected log-likelihood ($\mathbb{E}[\ln \mathcal{L}]$) is given by (52) where the terms \mathbf{w}_n and $\mathbf{w}_n \mathbf{w}_n^T$ take their expected values.

From Equation (49) the expected values are given by

$$\mathbb{E}[\mathbf{w}_n] = \mathbf{S}(\beta \boldsymbol{\Phi}^T \mathbf{d}_n + \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}) \quad (54)$$

$$\mathbb{E}[\mathbf{w}_n \mathbf{w}_n^T] = \mathbf{S} + \mathbb{E}[\mathbf{w}_n] \mathbb{E}[\mathbf{w}_n]^T \quad (55)$$

Minimizing the expected log likelihood with respect to β and $\boldsymbol{\Sigma}$ respectively give the following expressions for $\hat{\beta}$ and $\hat{\boldsymbol{\Sigma}}$.

$$\frac{1}{\hat{\beta}} = \frac{1}{NJ} \sum_{n \in N} \{ \mathbf{d}_n^T \mathbf{d}_n - 2 \mathbf{d}_n^T \boldsymbol{\Phi} \mathbb{E}[\mathbf{w}_n] + \text{Tr}(\boldsymbol{\Phi}^T \boldsymbol{\Phi} \mathbb{E}[\mathbf{w}_n \mathbf{w}_n^T]) \} \quad (56)$$

$$\hat{\boldsymbol{\Sigma}} = \frac{1}{N} \sum_{n \in N} \{ \boldsymbol{\mu} \boldsymbol{\mu}^T - 2 \boldsymbol{\mu} \mathbb{E}[\mathbf{w}_n^T] + \mathbb{E}[\mathbf{w}_n \mathbf{w}_n^T] \} \quad (57)$$

Equations (54) to (57) can be solved iteratively for $\hat{\beta}$ and $\hat{\boldsymbol{\Sigma}}$.

Sampling Wind Difference Profiles Using the learned values of $\hat{\boldsymbol{\mu}}$ and $\hat{\boldsymbol{\Sigma}}$ wind speed difference profiles can be generated randomly by sampling from the distribution in (46). Figure 13 shows examples of sampled wind-speed difference profiles. These were generated using the same atmospheric data as described later in Section 5.2. For more realistic generated difference profiles random noise with variance $\hat{\beta}^{-1}$ can optionally be added to the profiles.

As mentioned at the beginning of this section this method is designed to capture the variation of uncertainty with altitude and the correlation of speeds at adjacent altitudes. The effects of this can be seen in the profiles in Figure 13, where there is a trend of greater uncertainty around 10,000 m. This corresponds to the altitude range which generally has the highest wind speeds (the jet stream). Also the smooth variation of the profiles suggests a correlation in values at adjacent altitudes.

The method described in this section relates to a single component of the measured and forecast wind speed (easterly). In the application of this method the same procedure must be carried out for the northerly component of wind speed, and (if required) the vertical component

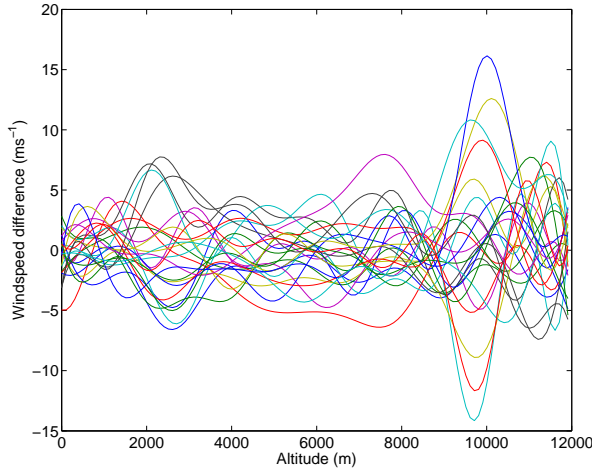


Figure 13: Samples of wind speed difference profiles drawn from (46).

of wind speed. Furthermore that entire procedure must be carried out separately for forecast data with different lead times. For example data for 4 hour forecasts and 8 hour forecasts should not be used in the same data set but should be considered separately, giving different values of $\hat{\boldsymbol{\mu}}$, $\hat{\boldsymbol{\Sigma}}$ and $\hat{\beta}^{-1}$.

At each step of the Monte Carlo simulation sampled difference profiles are added to the forecast wind profile to generate stochastic wind data.

5 Demonstration of the simulator

A demonstration of the approach described in this paper was carried out by flying a high power rocket with some instrumentation on board. Here we present a comparison between measured data obtained during the flight and data generated from a simulated flight.

5.1 Flight demonstration

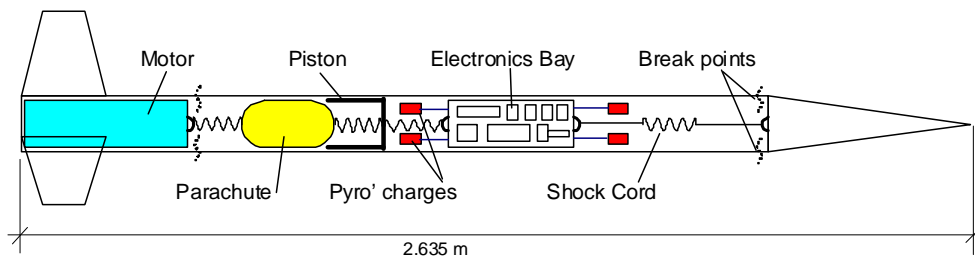


Figure 14: Schematic illustration of the rocket used in the flight demonstration.

A schematic of the rocket used in this demonstration is shown in Figure 14. This is a 2.6 m long 76 mm diameter rocket with a glass reinforced plastic fuselage. The thrust is provided by a Cesaroni L730 solid fuel motor (Figure 15). The rocket uses a dual deploy recovery system. At apogee the forward pyrotechnic charges are fired which causes the nose cone to detach. A Kevlar shock cord connects the nose cone to the rest of the rocket after separation. This causes a significant increase in the drag of the rocket although no parachute is deployed at this stage (this is nevertheless modeled as a parachute descent). When the rocket reaches an altitude of 300 m during the descent then the rear pyrotechnic charges are fired causing the tail section to detach and a piston pushes out the parachute.

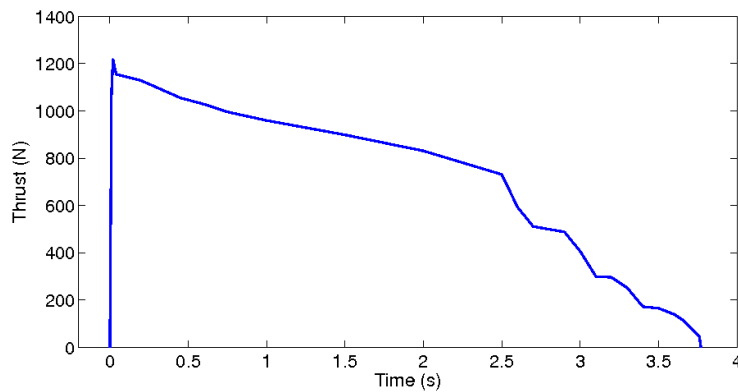


Figure 15: Thrust curve of the Cesaroni L730 solid fuel rocket motor.

The electronics bay contains a number of commercially available rocket avionics devices which consist of various sensors including 3-axis accelerometers, 2-axis Hall effect sensors, GPS receiver and pressure transducers. Microprocessors control the firing of the pyrotechnic devices and log the sensor data. A list of the devices is given in Table 6. The system for detecting altitude and detonating the charges is dual redundant for safety.

Device	Name	Manufacturer
Flight Computer (including pressure transducer and accelerometer)	RDAS	AED electronics
2-axis accelerometer	RDAS 2-axis	AED electronics
2-axis Hall effect sensor	Magneto sensor	Aerocon
GPS receiver	RDAS GPS	AED electronics
Redundant flight computer (including pressure transducer and accelerometer)	miniAlt	PerfectFlite

Table 6: Avionics devices used on board the rocket.

5.2 Flight Simulation

The flight path of the rocket described in Section 5.1 was simulated using the method laid out in this paper. The aerodynamic coefficients and the dynamic properties of the rocket that are required as inputs for the simulation (see Section 3.1) were estimated from the geometry of the rocket following the method laid out in [Box et al., 2009]. Here we have assumed that C_N is independent of Re .

The aerodynamic coefficients for the parachutes were estimated from the descent rate data recorded during previous flights.

To estimate the uncertainty in the atmospheric forecast we used data kindly supplied by the British Atmospheric Data Centre⁴. These consisted of five years of measurement and forecast data covering a period from 2001 to 2006. The measurement data came from the Mesosphere-Stratosphere-Troposphere (MST) radar at the University of Wales, Aberystwyth, and also from sounding balloons that are launched from the same site. Radar and sounding measurements are recorded four times a day at Aberystwyth.

The forecast data came from the Met Office’s Numerical Weather Prediction (NWP) model. Difference profiles between the 1 hour forecast data for the Aberystwyth site and the corresponding measurements were used as described in Section 4.2.2 to generate stochastic difference profiles. These were added to the NWP 1 hour forecast data for the rocket launch site to generate stochastic wind profiles for the simulation.

Other parameters which were made stochastic for this demonstration were the rocket aerodynamic coefficients (C_A and C_N), the centre of pressure X_{cp} and the parachute coefficient of drag C_D . These were made stochastic by multiplying by a random noise coefficient ξ as described in Section 4.2.1. Although the rocket is not designed to roll during flight imperfections in the build may induce some roll. This is modeled by making fin cant angle η a stochastic parameter, because η is zero mean it is modeled as $\eta \sim \mathcal{N}(0, \sigma^2)$. The variances of the noise coefficients were estimated in an ad-hoc manner. The method for estimating the values of the rocket parameters (from [Box et al., 2009]) was used to examine the sensitivity of the parameters to small changes in rocket or parachute geometry. The results were used to inform an expert guess at the variances, the values of σ used are shown in Table 7

Stochastic parameter	(σ)
C_A	0.2
C_N	0.1
X_{cp}	0.05
C_D	0.1
η	0.0087

Table 7: Values of the variance in the noise added to the stochastic parameters.

⁴<http://badc.nerc.ac.uk/home/index.html>

5.3 Demonstration Results

Unfortunately the GPS sensor failed to log any position data during the flight so the only accurate position data for the rocket are at the launch-pad and the landing site. Table 8 shows some selected statistics from the flight together with their simulated values for a run where all the stochastic parameters are set to their mean value. The measured velocities altitudes and times are as reported by the R-DAS avionics system.

	Simulated	Measured
Launch tower clearance velocity	40ms^{-1}	37ms^{-1}
Maximum velocity	372.5ms^{-1}	335ms^{-1}
Apogee altitude	3539m	3594m
Time to apogee	24.5s	24.5s
Total flight time	170s	182s
Landing position $[E, N]$	$[-135\text{m}, 936\text{m}]$	$[-71\text{m}, 1042\text{m}]$
Difference in landing positions	125m	

Table 8: Comparison between simulated and measured flight statistics.

Figure 16 shows a plot of the simulated flight path. The cross marks the launch-pad location and the mean flight path is plotted. The end of the flight path marks the mean simulated landing position. The two ellipses show, respectively, one and two standard deviations in the probability of the landing position as calculated from 500 Monte-Carlo simulation flights. The diamond symbol marks the landing location of the actual rocket at the end of the demonstration flight.

In Figure 17 the lateral acceleration of the rocket is plotted. Specifically this is the lateral acceleration at the point in the rocket where the accelerometers are located and both measured and simulated acceleration is shown. The simulated data were generated with all stochastic parameters at their mean value. The figure focuses on the short period of the rocket flight just after the rocket has cleared the launch tower, when weather cocking (§1.5.2) occurs. The damped oscillation can be seen in both the simulated and measured data and it can be seen that there is broad agreement in the amplitude, wavelength, phase and rate of decay of the oscillation. The figure shows a significant increase in the noise on the accelerometer signal both before and after the weather cocking event. The authors are not certain why this is the case although one possibility is that there is a mode of resonance affecting the board where the accelerometers were mounted that was not excited during weather cocking.

Figure 18 shows plots of linear and angular velocities of the rocket from a sample stochastic simulation. The linear velocities are in the rocket’s pitch, yaw and roll axes and the angular velocities are about these axes. These data show the six degrees of freedom in the simulation. It can be seen that the angular roll velocity is strongly correlated with the rocket’s forward velocity as expected. The damped oscillations during the weather cocking event can be seen in the pitch and yaw angular velocities and some further oscillations occur as the rocket pitches over at apogee.

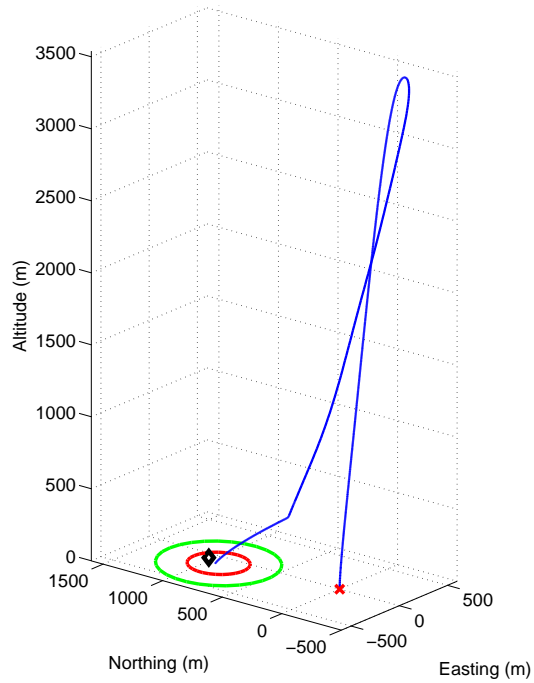


Figure 16: Mean simulated flight path with 1σ and 2σ landing probabilities. The measured landing position is marked by a diamond.

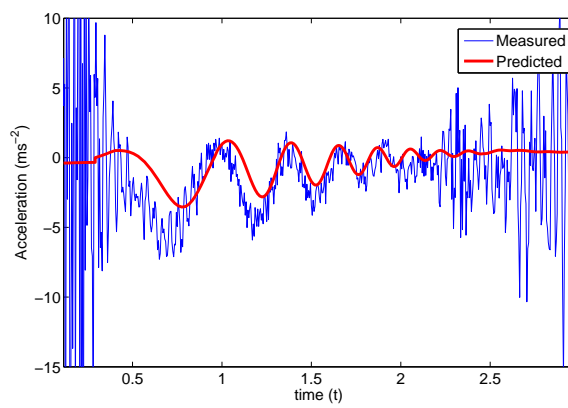


Figure 17: Comparison between the simulated and measured lateral acceleration of the rocket during the weather cocking event.

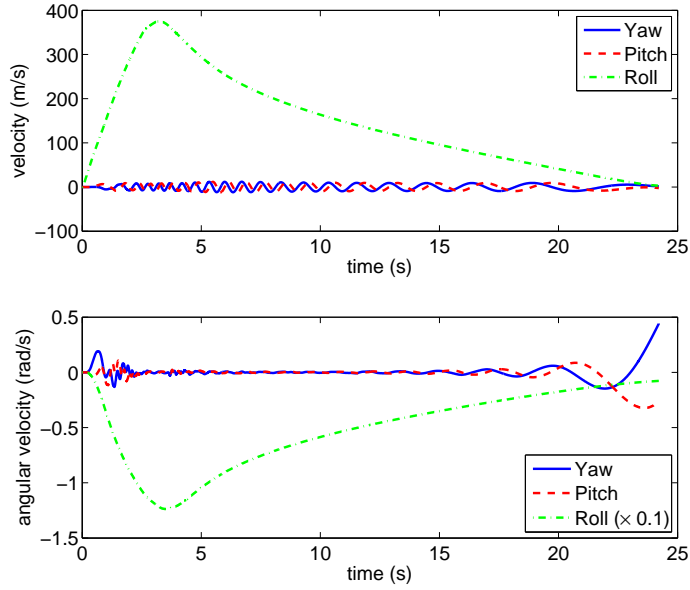


Figure 18: Outputs from a sample stochastic simulation showing the Earth relative linear and angular velocities of the rocket in reference axes (yaw, pitch and roll). The angular velocity about the roll axis is divided by 10 to improve scaling.

5.4 The Effect of Varying Uncertainty

To generate the stochastic data in Figure 16 best estimates of the variance in the stochastic parameters were used. However it is also interesting to examine the effect that changing the variance has on the results of the simulations. A full sensitivity analysis covering all stochastic parameters is beyond the scope of this paper but below we present some results from additional experiments where the standard deviations of some of the stochastic parameters were increased systematically.

5.4.1 Varying the Uncertainty in C_A

The Monte Carlo simulations that generated the results shown in Figure 16 were repeated twice. Once with the standard deviation of C_A increased to double its default value ($\sigma = 0.4$) and once with it doubled again ($\sigma = 0.8$). The standard deviations of all other stochastic parameters retained their default values (given in Table 7).

Figure 19 shows Gaussian ellipses marking the two standard deviations area of confidence in the landing position of the rocket. These were generated using the data from the original experiment and the two additional experiments described above.

It is interesting to note the directionality of the increase of the landing area. There is very little increase in the North West – South East direction, but significant increase in the North East – South West direction.

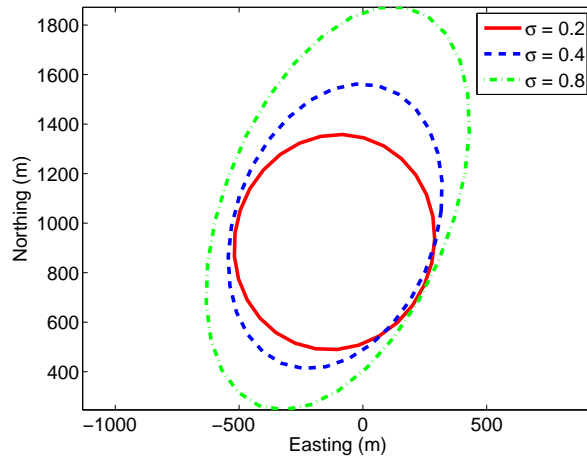


Figure 19: Gaussian ellipses showing the 2σ areas of confidence in landing position. C_A is stochastic with varying σ . All other stochastic parameters have their default values of σ (see Table 7)

Figure 20 shows a scatter plot of simulated apogee points from the above experiments. To avoid clutter only 100 apogees from each of the three experiments are shown. This plot shows the spread of the apogee points increasing with the increase in σ . With $\sigma = 0.8$ the variance is high enough that the drag force on the rocket can drop to zero, or even become negative. This is why some of the apogee points are very high. In this case the variance in C_A is unrealistically large.

5.4.2 Varying the Uncertainty in C_N

A similar test to the one described above was carried out for the rocket's coefficient of normal force C_N . Here the standard deviation of C_A was returned to its default value ($\sigma = 0.2$) and the standard deviation of C_N was doubled, once to $\sigma = 0.2$ and again to $\sigma = 0.4$. The results for landing position are shown in Figure 21.

As standard deviation is increased to $\sigma = 0.2$ the landing area increases slightly in the North West – South East direction, but actually reduces in the North East – South West direction. As the standard deviation is increased again to $\sigma = 0.4$ the area continues to grow in the North West – South East direction with no further reduction in the other direction.

Figure 22 shows a scatter plot of simulated apogee points from the C_N experiments. Again only 100 apogees from each of the three experiments are shown. This plot shows the spread of the apogee points increasing with the increase of the uncertainty in C_N but in general the growth in the spread of the apogee scatter is less than that observed when varying C_A .

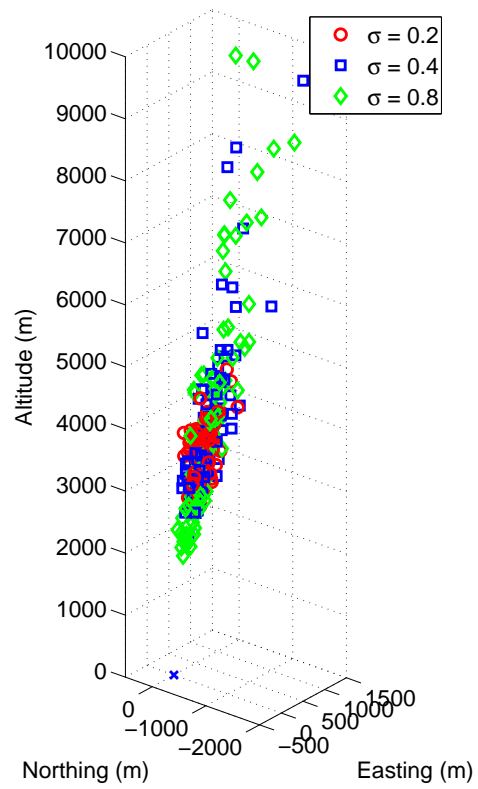


Figure 20: Scatter plot showing the spread of apogee points for 300 simulated flights. 100 for each value of σ relating to C_A .

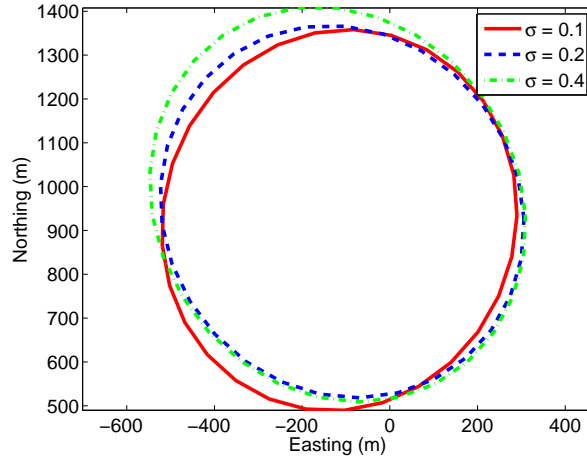


Figure 21: Gaussian ellipses showing the 2σ areas of confidence in landing position. C_N is stochastic with varying σ . All other stochastic parameters have their default values of σ (see Table 7)

5.4.3 Varying the Uncertainty in C_D

Variation of the uncertainty in the drag coefficients of the parachutes (C_D) was investigated following the same procedure as the previous two tests. The standard deviation of C_D for both the drogue and main parachutes was increased to $\sigma = 0.2$ and then $\sigma = 0.4$. The plots of landing position are shown in Figure 23. As with the previous examples, the increase in the landing area is highly directional. In this case the directionality is approximately aligned with the prevailing wind direction, which is predominantly blowing from the South West in these experiments.

6 Conclusions

The accuracy of the simulation method presented in this paper will depend upon the validity of the assumptions used in the rocket model (e.g. axisymmetric, rigid body), the error tolerance in the numerical integration and – to a significant extent – the accuracy of the user supplied parameters describing the rocket dynamics, aerodynamics and the atmospheric conditions.

There is a domain of uncertainty in the rocket’s trajectory that arises from the inaccuracies in these input parameters. The approach that we have used of stochastic simulation using the Monte-Carlo method allows us to explore this domain using knowledge of the inaccuracies in the estimation of these parameters.

We have presented some results from a demonstration where the simulator was used to predict the flight path of a real rocket. The results of this demonstration produced encouraging evidence that the deterministic simulation method can be effective. Of course a single comparison can tell us nothing about the efficacy of our method to estimate quantitatively the

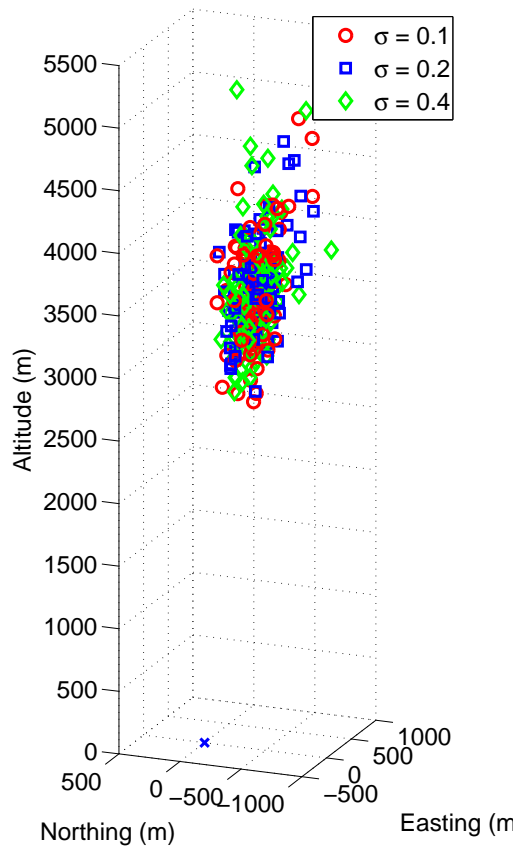


Figure 22: Scatter plot showing the spread of apogee points for 300 simulated flights. 100 for each value of σ relating to C_N .

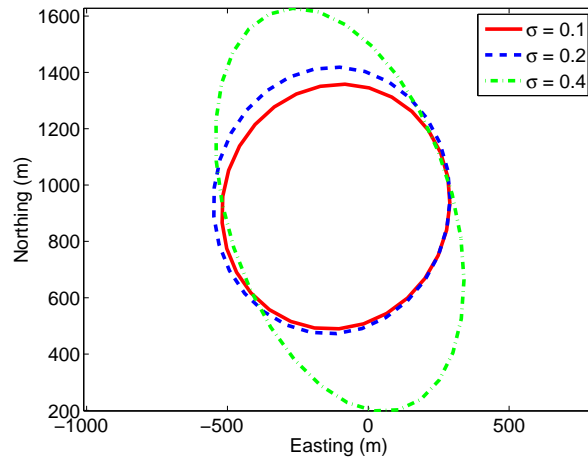


Figure 23: Gaussian ellipses showing the 2σ areas of confidence in landing position. C_D is stochastic with varying σ . All other stochastic parameters have their default values of σ (see Table 7)

uncertainty in the flight path. Validating this method would require data from many more test flights and this is a goal for future work.

Acknowledgements

Many thanks to Phil Charlesworth and Brett Saunders for numerous valuable discussions, and to the British Atmospheric Data Centre for granting access to their of data.

References

- D. Baraff. Rigid body simulation. In *Siggraph 97 24th International conference on computer graphics and interactive techniques – Course notes*, 1997.
- C. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.
- S. Box, C. Bishop, and H. Hunt. Estimating the dynamic and aerodynamic properties of small rockets for flight simulations. *Technical Report, available online at <http://cambridgerocket.sourceforge.net/>*, 2009.
- V. N. Dobrokhodov, O. A. Yakimenko, and C. J. Junge. Six-degree-of-freedom model of a controlled circular parachute. *Journal of Aircraft*, 40(3), 2003.
- L. D. Duncan and R. J. Ensey. Six degree of freedom digital simulation model for unguided fin-stabilized rockets. *US Army electronics research & development activity*, 1964.
- Y. Kim and C. S. Peskin. 2d parachute simulation by the immersed boundary method. *SIAM. J. Sci. Comput.*, 28(2294), 2006.
- N. Nassiri, J. Roushanian, and S. Haghigat. Stochastic flight simulation applied to a sounding rocket. In *55th International Astronautical Congress 2004 – Vancouver, Canada*, 2004.
- NFPA. National fire protection association – report of the committee on pyrotechnics. *NFPA 1122/1127*, 2002.
- W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. *Numerical Recipes*. Cambridge University Press, 2007.