How Best to Achieve Vertical Take Off/ Vertical Landing using Model Rockets

December 8, 2021

Contents

Notation

A_{ref} Reference area (m^2) used for aerodynamic calculations.

d_{ref} Reference diameter (m) used for aerodynamic calculations.

ρ Air density (kgm^{-3}).

$\boldsymbol{v}_{\text{inf}}$ Freestream velocity (ms^{-1}).

COT to COM Distance from application of thrust force to centre of mass (m).

$\boldsymbol{\omega}$ Angular velocity ($rads^{-1}$).

$\boldsymbol{I}$ Moment of inertia tensor (m).

$\boldsymbol{v}$ Velocity (ms^{-1}).

$\boldsymbol{r}$ Displacement (m).

m Mass (kg).

$\boldsymbol{x}_t$ State vector $\boldsymbol{x}$ at time t.

$\boldsymbol{y}_t$ Measurement vector $\boldsymbol{y}$ at time t.

$\boldsymbol{x}_{n|m}$ The estimate of $\boldsymbol{x}$ at time n given observations up to and including at time $m \leq n$.

$\hat{\boldsymbol{x}}$ Estimate of $\boldsymbol{x}$.

$\boldsymbol{q}^{nb}$ Unit quaternion representing the rotation to the n coordinate system from the b coordinate system.

$\boldsymbol{R}^{nb}$ Rotation matrix representing the rotation to the n coordinate system from the b coordinate system.

$\boldsymbol{x}^n$ Vector $\boldsymbol{x}$ expressed in the n coordinate system.

$[\boldsymbol{x}]^n$ A derivative $\boldsymbol{x}$ that has been taken in the n reference frame.

$[\boldsymbol{x}]^{nn}$ A derivative $\boldsymbol{x}$ taken in the n reference frame of a derivative that has been taken in the n reference frame.

$[\boldsymbol{\omega}]^{bn}$ The angular velocity of the b frame with respect to the n frame.

$SO(3)$ Special orthogonal group in three dimensions.

$\in$ Element of.

$\mathcal{N}(\mu, \Sigma)$ Gaussian distribution with mean μ and covariance Σ.

$\sim$ Distributed according to.

Operators

argmin Minimizing argument.

$\boldsymbol{a} \cdot \boldsymbol{b}$ Dot product of the vectors $\boldsymbol{a}$ and $\boldsymbol{b}$.

$\boldsymbol{a} \times \boldsymbol{b}$ Cross product of the vectors $\boldsymbol{a}$ and $\boldsymbol{b}$.

$[\boldsymbol{a}]_\times$ Cross product matrix of the vector $\boldsymbol{a}$ as defined in (1).

$\|\boldsymbol{a}\|$ Two-norm of the vector $\boldsymbol{a}$.

$\boldsymbol{A}^\mathsf{T}$ Transpose of the matrix $\boldsymbol{A}$.

$\boldsymbol{A}^{-1}$ Inverse of the matrix $\boldsymbol{A}$.

$\otimes$ Quaternion multiplication

Abstract

The goal of this paper is to provide a critical review of different guidance, navigation and control (GNC) schemes that can achieve vertical take off/ vertical landing (VTVL), which is where a rocket takes off and lands propulsively, using a model rocket. Reusable launch vehicles (RLVs), many of which use VTVL, could lower costs, decrease waste and allow for more scientific research extraterrestrially. Existing research typically only presents an attitude controller, which is insufficient for VTVL, whilst our main contribution is presenting GNC schemes for VTVL and comparing them to determine the best. The schemes are simulated with varying parameters to test robustness and to evaluate their performance, with linear control and non-linear navigation and guidance performing best.

1 Introduction

In this paper we compare different guidance, navigation and control (GNC) schemes to achieve vertical take off/ vertical landing (VTVL), which is where a rocket takes off and lands propulsively. VTVL reusable launch vehicles (RLVs) have become very popular in the last decade with the successful landings and subsequent reuse of Space X's Falcon-9 and Blue Origin's New Shepard. These rockets can be reused easily as having VTVL capabilities allows for precise landing, without this the rocket would likely land in the sea which would cause significant damage and so would need to repaired before reuse. Therefore implementing VTVL in other rockets, such as Rocket Lab's Electron, would significantly reduce costs and waste, allowing for cheaper launches and so significant innovation in areas such as small satellites or planetary soft landing. Planetary soft landing is where rockets land on extraterrestrial planets without significant damage to the vehicle or its payload. This requires a solution to the power descent guidance problem for pinpoint landing, 'defined as finding the fuel optimal trajectory that takes a lander with a given initial state (position and velocity) to a prescribed final state in a uniform gravity field, with magnitude constraints on the available net thrust, and various state constraints' [1]. This allows for landing near scientifically interesting targets or supplies, like fuel, in dangerous terrain or whilst avoiding other important objects.

With the newfound popularity of VTVL RLVs, model rocket enthusiasts wish to achieve VTVL with model rockets, in order to better emulate them. So, achieving VTVL with model rockets would provide a testing bed from which VTVL can be achieved in rockets, such as the Electron, and it will provide enjoyment to model rocket enthusiasts.

There are many existing papers outlining control on small scale rockets, however many only focus on attitude control [19, 18, 29], which is insufficient for VTVL and for most uses of rockets, such as the insertion of satellites into orbit. This is due to the fact that translational motion is not controlled; when landing the velocity of the rocket has to be minimal otherwise it will be damaged and may damage people and the surrounding areas. Also, when delivering a satellite, it is desired that they are placed into a certain orbit or position by the rocket and therefore the translational motion has to be controlled. So this paper presents full GNC schemes that can control both translational and rotational motion in order to achieve thrust vector control (TVC). These different schemes can be broadly grouped depending on whether they are linear or non-linear. A mix of linear and non-linear techniques are presented for guidance and control, including proportional-integral-derivative (PID), state space and model predictive control (MPC).

Non linear techniques are also used for state estimation, an extended kalman filter (EKF), for navigation. I hypothesise that non-linear techniques will perform better than linear techniques based on the criteria: the new dynamics of the system; the robustness of the controller and safety.

Existing research typically uses linear techniques; [18] shows a PID controller being used to control the roll axis using a reaction wheel effectively up to the point of saturation. The control on the pitch and yaw axis using TVC however has a rather large settling time in excess of 4s and possibly a steady state error, however insufficient data is presented, e.g. data with larger impulses and for a greater time is needed.

[19] shows how a linear quadratic Gaussian (LQG) controller can perform worse than a linear quadratic regulator (LQR) controller for attitude control and proposes a novel controller to solve this problem and improve performance.

[29] proposes a non-linear controller, a fuzzy PID controller, with no overshoot and lower settling times than a PID and LQR controller.

In order to develop the GNC schemes the system dynamics will be derived based on existing work [23], which outlines the aerodynamics of model rockets, and using actuators based on the existing work by [4] in section 3, Dynamics,. In section 4, Models,, all the different models of the system dynamics that are used in this paper will be presented. Subsequently in section 5, Guidance,, different algorithms that generate trajectories for VTVL will be presented and evaluated. Following this, in section 6, Navigation,, a series of state estimators will be derived and their performance compared. In section 7, Control,, different controllers will be designed and tuned, where they will be implemented and simulated so that they can be judged and compared.

MATLAB/SIMULINK will be used to simulate the model rocket and the different GNC schemes due to ease of use and existing capabilities, such as in built wind models. Mathematica and Maple were used for their symbolic maths capabilities to obtain symbolic LQR and for trajectory optimisation.

2 Background

The coordinate systems and reference frames that will be used in this paper are adapted from [20] and are laid out below:

The body frame and coordinate system $\boldsymbol{b}$ are the non inertial frame and coordinate system of the moving model rocket. Its origin is located in the centre of mass of the model rocket and z axis point through the centre line upwards, x to the right and y forwards.

The navigation frame and coordinate system $\boldsymbol{n}$ are a non inertial frame and coordinate system of the launch site, which we use to navigate.

The inertial frame and coordinate system $\boldsymbol{i}$ are a inertial frame and coordinate system whose origin is located at the centre of the earth and its axes are aligned with respect to the stars.

The earth frame and coordinate system e are a non inertial frame and coordinate system whose origin is located at the centre of the earth and its axes are aligned with respect to the earth.

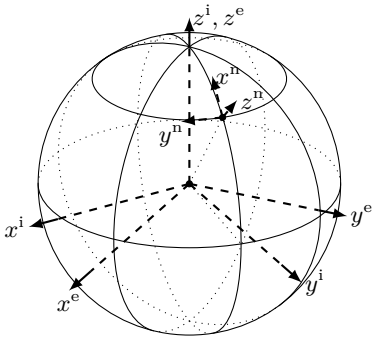


Figure 1: The n, i and e coordinate systems shown on the Earth. [20]

g is taken to be positive for the purposes of this paper.

The quaternion convention used is the Hamilton convention defined in [27].

Axis angle is a parameterisation of $SO(3)$ which is defined as $[\boldsymbol{u}^{\mathsf{T}}\ \phi]^{\mathsf{T}}$, where ϕ is the angle of rotation in radians and $\boldsymbol{u}$ is a unit vector representing the axis of rotation.

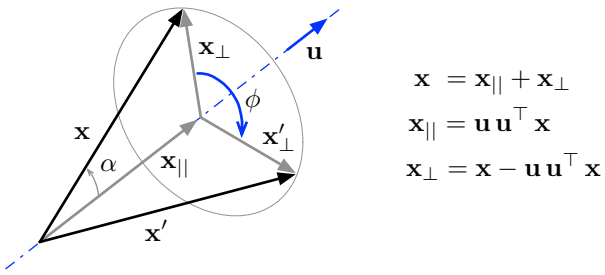


Figure 2: The vector $\mathbf{x}$ rotated about $\mathbf{u}$ by ϕ gives vector $\mathbf{x}'$.[27]

The skew symmetric matrix of a vector is defined as

$$[\boldsymbol{\omega}]_\times \equiv \begin{bmatrix} 0 & -\boldsymbol{\omega}_z & \boldsymbol{\omega}_y \\ \boldsymbol{\omega}_z & 0 & -\boldsymbol{\omega}_x \\ -\boldsymbol{\omega}_y & \boldsymbol{\omega}_x & 0 \end{bmatrix} \tag{1}$$

2.1 State Space

A state space representation is when a system is described using first order differential or difference equations. To express a continuous state space representation in matrix vector form the equations

must be linear:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}(t)\boldsymbol{x}(t) + \boldsymbol{B}(t)\boldsymbol{u}(t) \tag{2}$$
$$\boldsymbol{y}(t) = \boldsymbol{C}(t)\boldsymbol{x}(t) + \boldsymbol{D}(t)\boldsymbol{u}(t) \tag{3}$$

Most controllers require linear time invariant (LTI) systems which are described as such in a state space representation if described using differential equations:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t) \tag{4}$$
$$\boldsymbol{y}(t) = \boldsymbol{C}\boldsymbol{x}(t) + \boldsymbol{D}\boldsymbol{u}(t) \tag{5}$$

If the state space system is implemented on a digital system, if the actuators are discrete or difference equations are used the state space system may be described in a discrete representation. The LTI discrete state space representation is expressed as such:

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{u}_k \tag{6}$$
$$\boldsymbol{y}_k = \boldsymbol{C}\boldsymbol{x}_k + \boldsymbol{D}\boldsymbol{u}_k \tag{7}$$

To convert from a time varying state space representation to a time invariant representation, the representation at an operating point can be taken to hold for the entire range of values. Typically, the equilibrium points are used as the system is likely to stabilize around that point [17]. This means the state will be close to the equilibrium point most of the time so the representation stays reasonably accurate. This has disadvantages, e.g. if the state deviates from the operating points for extended periods or for a short time if the plant is strongly non-linear the LTI representation will be inaccurate.

Many systems are non-linear, so to apply linear techniques such as LQR non-linear models need to be linearised about operating points $\bar{\boldsymbol{x}}, \bar{\boldsymbol{u}}, \bar{\boldsymbol{y}}$. We often wish to use linear techniques as they are simpler, linear control theory is more developed than non-linear control theory and control designers will have more knowledge about linear techniques. Additionally, non-linear control techniques are typically only useful for certain non-linear systems whilst linear techniques can work for a wide range of systems. A non-linear system expressed as:

$$\dot{\boldsymbol{x}}(t) = f(\boldsymbol{x}(t), \boldsymbol{u}(t), \bullet) \tag{8}$$
$$\boldsymbol{y}(t) = g(\boldsymbol{x}(t), \bullet) \tag{9}$$

Can be linearised as such [5]:

$$\tilde{\boldsymbol{x}} = \boldsymbol{x} - \bar{\boldsymbol{x}} \tag{10}$$
$$\tilde{\boldsymbol{u}} = \boldsymbol{u} - \bar{\boldsymbol{u}} \tag{11}$$
$$\tilde{\boldsymbol{y}} = \boldsymbol{y} - \bar{\boldsymbol{y}} \tag{12}$$

The taylor series expansion of $\dot{x}$ and y are as follows:

$$\dot{\boldsymbol{x}} = f(\bar{\boldsymbol{x}}, \bar{\boldsymbol{u}}, \bar{\bullet}) + \frac{\partial f(\bar{\boldsymbol{x}}, \bar{\boldsymbol{u}}, \bar{\bullet})}{\partial \boldsymbol{x}}\tilde{\boldsymbol{x}} + \frac{\partial f(\bar{\boldsymbol{x}}, \bar{\boldsymbol{u}}, \bar{\bullet})}{\partial \boldsymbol{u}}\tilde{\boldsymbol{u}} \ldots \tag{13}$$
$$\boldsymbol{y} = g(\bar{\boldsymbol{y}}, \bar{\bullet}) + \frac{\partial g(\bar{\boldsymbol{y}}, \bar{\boldsymbol{u}}, \bar{\bullet})}{\partial \boldsymbol{y}}\tilde{\boldsymbol{y}} + \frac{\partial g(\bar{\boldsymbol{y}}, \bar{\bullet})}{\partial \boldsymbol{u}}\tilde{\boldsymbol{u}} \ldots \tag{14}$$

To obtain a linear state system use only the first order terms.

$$\dot{\tilde{\boldsymbol{x}}} = \frac{\partial f(\bar{\boldsymbol{x}}, \bar{\boldsymbol{u}}, \bar{\bullet})}{\partial \boldsymbol{x}} \tilde{\boldsymbol{x}} + \frac{\partial f(\bar{\boldsymbol{x}}, \bar{\boldsymbol{u}}, \bar{\bullet})}{\partial \boldsymbol{u}} \tilde{\boldsymbol{u}} \tag{15}$$

$$\tilde{\boldsymbol{y}} = \frac{\partial g(\bar{\boldsymbol{y}}, \bar{\bullet})}{\partial \boldsymbol{y}} \tilde{\boldsymbol{y}} + \frac{\partial g(\bar{\boldsymbol{y}}, \bar{\bullet})}{\partial \boldsymbol{u}} \tilde{\boldsymbol{u}} \tag{16}$$

In order to obtain better performance, gain scheduling can be used where multiple operating points are chosen to generate linear controllers. Depending on the current state, controllers can be appropriately selected or interpolated between.

Time delays present in a system may be approximated using a padé approximation which may add additional states and so a state observer or estimator is also required.

A system is observable if the current state can be estimated using the output for all possible states and control vectors. A state space system of the form (4), (5) is observable if the rank of $\boldsymbol{\mathcal{O}}$ is the same as the number of state variables [15].

$$\boldsymbol{\mathcal{O}} = \begin{bmatrix} \boldsymbol{C} \\ \boldsymbol{CA} \\ \boldsymbol{CA}^2 \\ \vdots \\ \boldsymbol{CA}^{n-1} \end{bmatrix} \tag{17}$$

A system is controllable if any state can be achieved from any initial state in a finite time. A state space system of the form (4), (5) is controllable if the rank of $\boldsymbol{\zeta}$ is the same as the number of state variables [15].

$$\boldsymbol{\zeta} = \begin{bmatrix} \boldsymbol{B} \\ \boldsymbol{AB} \\ \boldsymbol{A}^2\boldsymbol{B} \\ \vdots \\ \boldsymbol{A}^{n-1}\boldsymbol{B} \end{bmatrix} \tag{18}$$

A system can be controlled using LQR or pole placement if the LTI state space representation is controllable. Pole placement and LQR are similar as they both set the eigenvalues of the system, which determines the system's dynamics and so stability. However, pole placement is given eigenvalues whilst LQR determines the optimal eigenvalues and gain matrix, $\boldsymbol{K}$, minimising the quadratic cost function, J, where $\boldsymbol{u} = -\boldsymbol{K}\boldsymbol{x}$ and $\dot{\boldsymbol{x}} = (\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K})\boldsymbol{x}$ [15].

The infinite-horizon, continuous-time LQR defines the cost, J, as

$$J = \int_0^\infty (\boldsymbol{x}^\mathsf{T}\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{u}^\mathsf{T}\boldsymbol{R}\boldsymbol{u} + 2\boldsymbol{x}^\mathsf{T}\boldsymbol{N}\boldsymbol{u})\,\mathrm{d}t. \tag{19}$$

The infinite-horizon, discrete-time LQR defines the cost, J, as

$$J = \sum_{k=0}^{\infty} (\boldsymbol{x}_k^\mathsf{T}\boldsymbol{Q}\boldsymbol{x}_k + \boldsymbol{u}_k^\mathsf{T}\boldsymbol{R}\boldsymbol{u}_k + 2\boldsymbol{x}_k^\mathsf{T}\boldsymbol{N}_k\boldsymbol{u}). \tag{20}$$

$\boldsymbol{K}$ can be found using the algebraic Riccati equation.

Additionally, more advanced controllers can be used that are designed for non-linear, time varying systems or that can account for constraints, such as iLQR or MPC, can be designed using a state space model. However, these are more computationally expensive than pole placement and LQR and may be less robust.

3 Dynamics

The dynamics of the rocket have to be derived in order to simulate the system (the rocket being controlled) so that the performance of each GNC scheme can be compared. Additionally, the system dynamics are required to develop/ tune the GNC schemes, which is where the simpler models will be used to deal with the limitation of different types of GNC schemes.

A review of the dynamics used to model similar systems, such as quadcopters and other model rockets, revealed that the prominent differences between the different methods were: the reference frames and coordinate systems used; the rotation formalism used, commonly a choice between euler angles or quaternions; and the value of the aerodynamic coefficients. Additionally, it shows that a common way to model a model rocket is to assume it is a rigid body, when the body does not deform under the action of applied forces, and to use 6DOF equations of motion.

The reference frames and coordinate systems used make no difference to the accuracy of the simulation and as such those which reduce the computational effort of the simulation and simplify the analysis of the model are being used.

It is shown in [28] that all three dimensional parameterisations of $SO(3)$ have singularities or discontinuities however, modified rodrigues parameters (MRPs) have a singularity at 360° and so switching to the shortest rotation the singularity can be avoided. Therefore the choice of rotation formalism used can affect the accuracy of the simulation if a singularity is encountered, which is present in the popular euler angle parameterisation. Additionally, euler angles 'are less accurate than quaternions when used to integrate incremental changes in attitude over time' [11]. Also, quaternions are more computationally efficient, which will be important when implemented on a microcontroller, and so will be used instead of euler angles even though they are arguably less intuitive.

There are many different methods to calculate aerodynamic coefficients, falling under three categories: computational fluid dynamics (CFD); theoretical and empirical formulae; and wind tunnel testing. The preliminary results obtained from SolidWorks Fluid Simulation were inaccurate as they did not fit with empirical data and so CFD was deemed to be unsuitable for the calculation of aerodynamic coefficients. Two common pieces of software used to calculate the aerodynamic coefficients using formulae are USAF Digital DATCOM and OpenRocket, [23]. A brief comparison of the coefficients produced by both software led me to believe that OpenRocket was more accurate as DATCOM used a linear relationship for the pitching moment coefficient, which would seem to be an oversimplification especially when compared to empirical data. Also, OpenRocket's simulation was compared to empirical data from model rockets and was found to be reasonably accurate and its assumptions are mostly true for our case. Wind tunnel testing was not available

and so was not considered. So OpenRocket will be used to calculate the aerodynamic coefficients which should be indicative of their real world performance allowing for comparison between different controllers.

The following dynamics are derived in the navigation frame and the earth is assumed to be flat.

3.1 Actuator

There has been previous success [4] in controlling attitude in model rockets using gimbaled thrust, where the thrust direction is controlled in order to impart moments and forces, and a reaction wheel, that will exert a moment about the roll axis. Roll control is necessary because at high roll rates a corkscrew like flight will develop which could lead to instability, shown both in testing and in [18]. Also, high roll rates can lead to bad performance due to aerodynamics couplings [12]. Additionally, roll control is desired to point the rocket in a certain direction, e.g. to deliver a payload.

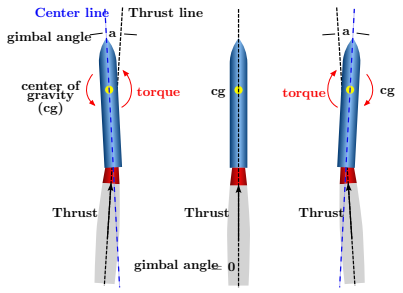


Figure 3: The gimbal rotates the thrust force in order to impart a torque and a horizontal force [22].

Let $\boldsymbol{u}(t)$ be the controller's output at time t, where $\boldsymbol{u}(t) = \begin{bmatrix} u_x(t) & u_y(t) & u_z(t) \end{bmatrix}^\mathsf{T}$. The gimbal actuator lags by 0.07s and saturates at a 5° angle, whilst the reaction wheel has a 0.01s lag time.

$$\boldsymbol{f}(t) = \text{COT to COM}^{-1} \begin{bmatrix} -u_y\big(\text{round}(t-0.07, 0.02)\big) \\ u_x\big(\text{round}(t-0.07, 0.02)\big) \end{bmatrix}, \tag{21}$$

where $\text{round}(x, y)$ rounds x to the nearest multiple of y

$$\textbf{Actuator Force}^b(t) = \begin{cases} \boldsymbol{f}(t) & \|\boldsymbol{f}(t)\| \leq \text{Thrust}(t)\sin(5°) \\ \text{Thrust}(t)\sin(5°) * \frac{\boldsymbol{f}(t)}{\|\boldsymbol{f}(t)\|} & \text{otherwise.} \end{cases} \tag{22}$$

$$\textbf{Thrust}^b = \begin{bmatrix} \textbf{Actuator Force}^b(t) \\ \sqrt{\|\text{Thrust}(t)\|^2 - \left\|\textbf{Actuator Force}^b(t)\right\|^2} \end{bmatrix} \tag{23}$$

$$\textbf{Actuator Moment}^b(t) = \begin{bmatrix} u_x\big(\text{round}(t-0.07, 0.02)\big) \\ u_y\big(\text{round}(t-0.07, 0.02)\big) \\ u_z(t-0.01) \end{bmatrix} \tag{24}$$

3.2 Translation

The aerodynamic forces acting on the body are as follows, with **Rot** and the coefficients being defined in subsection A.1, Coefficients.

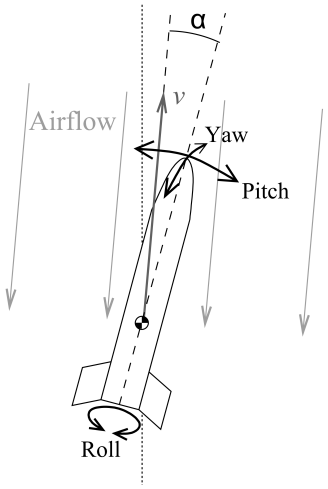
$$\mathbf{AeroForces}^b = \frac{1}{2}\rho\|\boldsymbol{v}_\infty\|^2 A_{ref} * \mathbf{Rot}\begin{bmatrix}-C_N\\-C_S\\-C_A\end{bmatrix} \tag{25}$$

The force is therefore the sum of the aerodynamic forces, the thrust force rotated by the TVC gimbal and the weight force in the body coordinate system

$$\boldsymbol{F}^b = \mathbf{AeroForces}^b + \mathbf{Thrust}^b + \boldsymbol{R}^{bn}\begin{bmatrix}0\\0\\-mg\end{bmatrix} \tag{26}$$

$$[\boldsymbol{a}^b]^{nn} = \frac{\boldsymbol{F}_b}{m} \tag{27}$$

$$[\boldsymbol{a}^n]^{nn} = \boldsymbol{R}^{nb}[\boldsymbol{a}^b]^{nn} \tag{28}$$



The pitch, yaw and roll directions in the b frame. [23]

3.3 Attitude

The aerodynamic moments acting on the body are as follows, with the coefficients again defined in subsection A.1, Coefficients.

$$\mathbf{AeroMoments}^b = \frac{1}{2}\rho\|\boldsymbol{v}_\infty\|^2 A_{ref} * d_{ref} * \mathbf{Rot}\begin{bmatrix}-C_y\\C_m\\C_l\end{bmatrix} \tag{29}$$

The net moment is the sum of the aerodynamic moments and the actuator moments from the TVC gimbal and the reaction wheel

$$\boldsymbol{\tau}^b = \mathbf{AeroMoments}^b + \mathbf{Actuator\ Moment}^b(t) \tag{30}$$

$$[\dot{\boldsymbol{\omega}}^b]^{bn} = (\boldsymbol{I}^b)^{-1}(\boldsymbol{\tau}^b - [\boldsymbol{\omega}^b]^{bn} \times \boldsymbol{I}^b[\boldsymbol{\omega}^b]^{bn}) \tag{31}$$

$$\dot{\boldsymbol{q}}^{nb} = \frac{1}{2}\boldsymbol{q}^{nb}\begin{bmatrix}0\\[\boldsymbol{\omega}^b]^{bn}\end{bmatrix} \tag{32}$$

The dynamics derived in the section have been deemed to be sufficiently accurate and so indicative of real world performance, because the dynamics are similar to those found in existing literature where they have been shown to be accurate to the real world. This will allow for a fair comparison between different GNC schemes and will show if VTVL has been achieved.

4 Models

In this section, different models of varying complexities will be proposed as applications, such as LQR, require modifications of the model derived in the previous section, section 3, Dynamics.

4.1 First Model

This is the most accurate model of the system which will be used in the simulation to evaluate the performance of a proposed GNC scheme.

$$[\dot{\boldsymbol{\omega}}^b]^{bn} = (\boldsymbol{I}^b)^{-1}(\mathbf{AeroMoments} + \mathbf{Actuator\ Moments}(t) - [\boldsymbol{\omega}^b]^{bn} \times \boldsymbol{I}^b[\boldsymbol{\omega}^b]^{bn}) \tag{33}$$

$$\dot{\boldsymbol{q}}^{bn} = \frac{1}{2}\boldsymbol{q}^{bn}\begin{bmatrix}0 & [\boldsymbol{\omega}^b]^{bn}\end{bmatrix}^\mathsf{T} \tag{34}$$

$$[\dot{\boldsymbol{v}}^n]^{nn} = \boldsymbol{R}^{nb}m^{-1}\left(\mathbf{AeroForces}^b + \mathbf{Thrust}^b\right) + \begin{bmatrix}0 & 0 & -g\end{bmatrix}^\mathsf{T} \tag{35}$$

$$[\dot{\boldsymbol{r}}^n]^n = [\boldsymbol{v}^n]^n \tag{36}$$

4.2 Second Model

This model is for used for translational control using linear techniques such as LQR.

As little translational control can be achieved in the z axis and it would be difficult to incorporate this into state space form, so the z axis is not included. Therefore, the weight force is omitted. Additionally the quaternion error term, $\boldsymbol{q}_v$ is used instead of $\boldsymbol{q}$. The choice of $\boldsymbol{q}_v$ is explained in section 7, Control,.

$$[\dot{\boldsymbol{\omega}}^b]^{bn} = (\boldsymbol{I}^b)^{-1}(\mathbf{Actuator\ Moments}(t) - [\boldsymbol{\omega}^b]^{bn} \times \boldsymbol{I}^b[\boldsymbol{\omega}^b]^{bn}) \tag{37}$$

$$\dot{\boldsymbol{q}}_v^{bn} = \left(\frac{1}{2}\boldsymbol{q}^{bn}\begin{bmatrix}0 & [\boldsymbol{\omega}^b]^{bn}\end{bmatrix}^\mathsf{T}\right)_v \tag{38}$$

$$\frac{d}{dt}\int \boldsymbol{q}_v^{bn} = \boldsymbol{q}_v^{bn} \tag{39}$$

$$[\dot{\boldsymbol{v}}_{x,y}^n]^{nn} = \left(\boldsymbol{R}^{nb}m^{-1}\mathbf{Thrust}^b\right)_{x,y} \tag{40}$$

$$[\dot{\boldsymbol{r}}_{x,y}^n]^n = \boldsymbol{v}_{x,y}^n \tag{41}$$

4.3 Third Model

This model is for used for attitude control using linear techniques such as LQR and is based on the previous model, Second Model, 4.2, so eqs. (37) to (39) are used.

4.4 Fourth Model

This model, in contrast to the Second Model, 4.2, includes the translation variables in the z axis and $\boldsymbol{q}_w$ so that it can be used to build a state estimator or observer. Let the gyroscope bias be $\boldsymbol{\omega}_b$ and the accelerometer bias $\boldsymbol{a}_b$.

Using equations (34), (36), (37)

$$\dot{\boldsymbol{\omega}}^b{}_b = 0 \tag{42}$$

$$\dot{\boldsymbol{a}}^b_b = 0 \tag{43}$$

$$[\dot{\boldsymbol{v}}^n]^{nn} = \boldsymbol{R}^{nb} m^{-1} \mathbf{Thrust}^b + \begin{bmatrix} 0 & 0 & -g \end{bmatrix}^\mathsf{T} \tag{44}$$

$$\tag{45}$$

4.5 Fifth Model

Different actuator models are needed, e.g. for controllers which take into account the time delay in the actuator and those that don't.

This model is the most accurate model of the system and will be used in the simulation to evaluate the performance of a proposed GNC scheme.

Equations (21) to (24) describe this model.

4.6 Sixth Model

This actuator model ignores the saturation and can be continuous or discrete with any sample time by appropriately modifying (21) and (24).

This model is appropriate for controllers that cannot be tuned/ developed with models including saturation terms such as a LQR or PID controller.

Using equations (21), (23), (24)

$$\mathbf{Actuator\ Force}^b(t) = \boldsymbol{f}(t) \tag{46}$$

4.7 Seventh Model

This actuator model additionally assumes that there is no time delay in the actuators compared to the Sixth Model, 4.6.

Using equations (46) and (24).

$$\boldsymbol{f}(t) = \text{COT to COM}^{-1} \begin{bmatrix} -u_y(t) \\ u_x(t) \end{bmatrix} \tag{47}$$

$$\textbf{Actuator Moment}^b(t) = \boldsymbol{u}(t) \tag{48}$$

5 Guidance

Guidance refers to the generation of a trajectory in order to achieve a goal such as VTVL, where a trajectory is useful in determining how to land safely. This is necessary as without a trajectory the rocket would not know how to land safely or achieve any other goal and therefore would likely fail at achieving this goal. There are existing guidance algorithms for VTVL in rockets, such as GFOLD [3, 2, 5] or solutions to the moon soft landing problem, which generate an optimal trajectory to land a rocket whilst typically minimising fuel consumption. However, these algorithms are unsuitable as a solid motor is going to be used so the fuel use is not a variable which can be optimised.

5.1 Optimal Motor Ignition Time

A novel algorithm is proposed, which can be used to determine an optimal motor ignition time that minimizes the impact velocity with the ground, ensuring the least damage to the rocket on landing, when a simple trajectory, which minimises horizontal velocity, is used.

During the powered portion of the flight, weight and thrust are taken to be the only forces acting on the rocket and thus the dynamics of the rocket are: Let x^n be the height of the rocket and $x^n = 0$ be when the rocket is touching the ground

$$[\ddot{x}^n]^{nn}(t) = \text{Thrust}(t-\gamma)m^{-1} - g \tag{49}$$

$$t \in \{a : 0 \leq a \leq b\} \text{ where } s(b) = 0 \tag{50}$$

The optimisation problem to be solved is thus $\underset{\gamma}{\text{argmin}} \, |[\dot{x}^n]^n(b)|$ subject to $b \geq$ burn time (taken to be 3.45), where the speed at which the rocket hits the ground is minimised with respect to γ when the motor is expended. When the rocket hits the ground before burn time the landing is considered a fail as the rocket's dynamics would be hard to model and could result in dangerous behaviour. There are many different ways to solve this.

One method is to first sample $|[\dot{x}^n]^n(b)|$ subject to $b \geq$ burn time for a range of initial conditions, $[\dot{x}^n]^n(\gamma)$ and $x^n(\gamma)$. Then using nearest neighbour interpolation and extrapolation, as other methods such as linear, cubic, makima resulted in noisy data and curve fitting with polynomials and neural networks were not a close enough fit to the data, the optimisation problem can be solved for a range of initial conditions, $[\dot{x}^n]^n(0)$ and $x^n(0)$.

Alternatively, the optimisation problem can be solved directly for a range of $\dot{x}(0)$ and $x(0)$ using an ODE solver to calculate $|\dot{x}(b)|$ during optimisation. This provides better accuracy, however this is significantly slower than evaluating the interpolation function which can be generated very quickly in MATLAB.

When $\gamma <$ *constant*, the motor should be ignited, so the region $\gamma <$ *constant* is approximated in

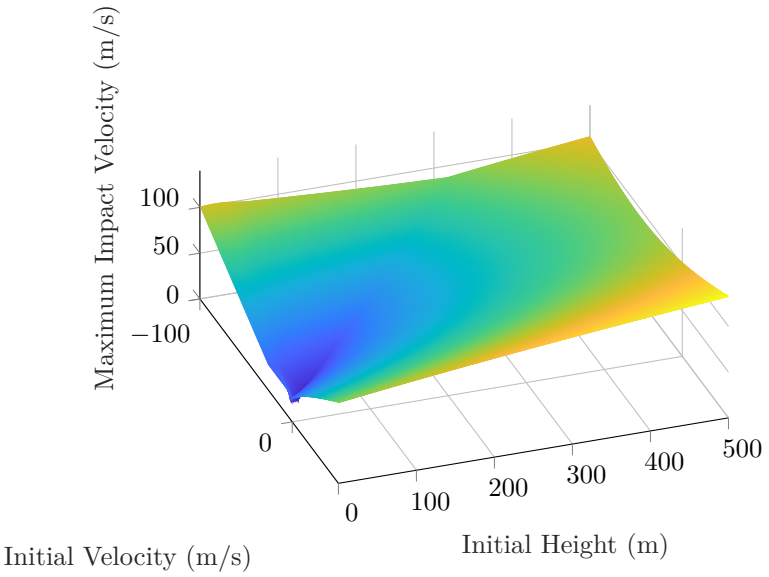


Figure 5: Surface showing speed at which rocket will hit the ground for a range of initial conditions, $[\dot{x}^n]^n(\gamma)$ and $x^n(\gamma)$

order to facilitate the online calculation of gamma at a frequent rate to allow for deviations from the model of the dynamics, e.g. due to aerodynamic forces or due to an ideal 1D case being used. The use of *constant* accounts for a delay in the motor ignition.

As shown by figure 5 and 6 the speed at which the rocket will hit the ground is not 0 m/s except in specific cases and so more advanced algorithms, that generate a more complex trajectory where the rocket does not simply try to minimise horizontal velocity, could improve on this algorithm.

5.2 Landing Trajectory Optimisation

A novel algorithm for landing trajectory optimisation using solid motors is proposed:

$$\boldsymbol{\theta} = [\theta_1 \ ... \ \theta_n] \tag{51}$$

$$\boldsymbol{\phi} = [\phi_1 \ ... \ \phi_n] \tag{52}$$

$$\theta_1, \phi_1 \text{ describe the current orientation of the rocket} \tag{53}$$

$$\boldsymbol{u}_n = \begin{bmatrix} \sin\theta_n \cos\phi_n \\ \sin\theta_n \sin\phi_n \\ \cos\theta_n \end{bmatrix} \tag{54}$$

$$\theta_n = 0, \ \phi_n = 0 \text{ to land upright.} \tag{55}$$

$$\arccos \frac{\boldsymbol{u}_{n+1} \cdot \boldsymbol{u}_n}{\Delta t} \leq \boldsymbol{\omega}_{max} \tag{56}$$

$$[\ddot{\boldsymbol{x}}^n]^{nn}(t) = \boldsymbol{u}(t) \frac{\text{Thrust}(t)}{m(t)} + \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} \tag{57}$$

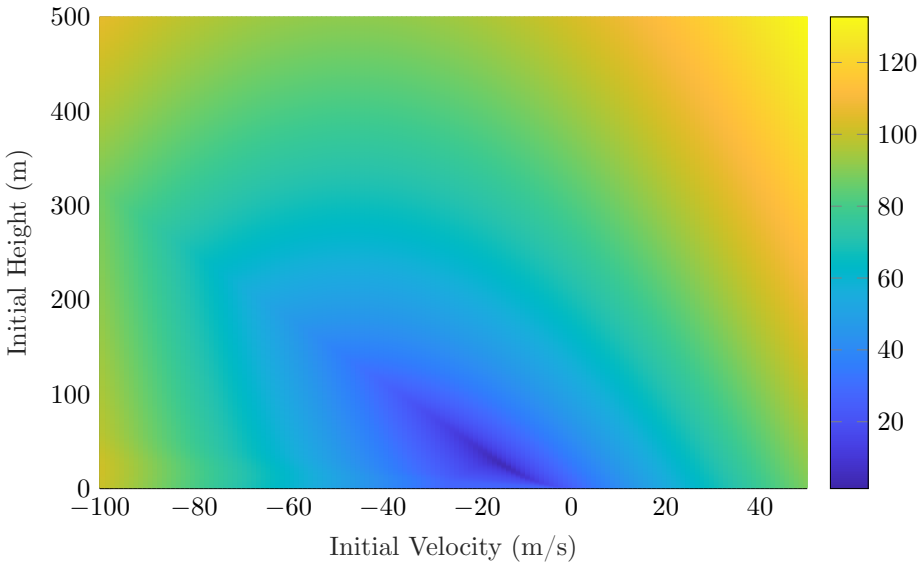


Figure 6: Heatmap showing speed at which rocket will hit the ground for a range of initial conditions, $[\dot{x}^n]^n(\gamma)$ and $x^n(\gamma)$

$$\boldsymbol{\theta}, \boldsymbol{\phi} = \underset{\boldsymbol{\theta},\boldsymbol{\phi}}{\operatorname{argmin}} \|[\dot{x}^n]^n(t)\|, \text{ where } x^n(t) = 0 \tag{58}$$

$\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{u}$ are are discrete vectors which are linearly interpolated between to obtain $\boldsymbol{\theta}(t), \boldsymbol{\phi}(t), \boldsymbol{u}(t)$. A constraint on the maximum angular velocity of the rocket is enforced to ensure the rocket is able to rotate to the desired orientation in time. This algorithm is too slow for real time use, however through convexification and the offline computation of trajectories real time use could be possible. An optimal time to ignite the motors using this algorithm could be approximated by the using the result from the previous algorithm.

If an attitude controller is being used, a guidance algorithm can be used to generate an attitude trajectory in order to control $[\boldsymbol{v}^n_{x,y}]^n$ or $\boldsymbol{r}^n_{x,y}$. A PID controller can be used to generate an acceleration command from an error term, which can be used to calculate the desired attitude and angular velocity. The desired angular velocity may be taken to be 0 as it could provide better performance that taking it to be $2 \log \boldsymbol{q}$.

The algorithms presented in this section should be able to generate trajectories which are sufficient for the purpose of VTVL however, future work will be needed to verify this.

6 Navigation

Control and guidance need to know certain information accurately, typically the state or the error, in order to perform well. In order to obtain this information state estimation or observation must be employed and so state estimation or observation is critical in being able to achieve VTVL.

State observation involves estimating states assuming no noise in the measurements whilst state estimation accounts for noise in the measurements [25, 7]. State estimation or observation requires that the system be observable as discussed in subsection 2.1, State Space. We will be using state estimation or observation to estimate the additional states from a padé approximation and the states in the Fourth Model, 4.4, using Fifth Model, 4.5 from sensor measurements. Fourth Model, 4.4, will be used as guidance and control collectively require knowledge of position, velocity, orientation and angular velocity.

The state estimate is referred to as $\hat{\boldsymbol{x}}$ and the true state is referred to as $\boldsymbol{x}$. $\hat{\boldsymbol{x}}_{n|m}$ represents the estimate of $\boldsymbol{x}$ at time n given observations up to and including m where $m \leq n$.

6.1 Luenberger observer

The Luenberger observer, a state observer, sets $\boldsymbol{L}$ such that the observer error $\boldsymbol{e}$, $\boldsymbol{x} - \hat{\boldsymbol{x}}$, approaches 0 as $t \to \infty$

$$\dot{\hat{\boldsymbol{x}}} = \boldsymbol{A}\hat{\boldsymbol{x}} + \boldsymbol{B}\boldsymbol{u} + \boldsymbol{L}(\boldsymbol{y} - \hat{\boldsymbol{y}}) \tag{59}$$
$$\hat{\boldsymbol{y}} = \boldsymbol{C}\hat{\boldsymbol{x}} + \boldsymbol{D}\boldsymbol{u} \tag{60}$$
$$\dot{\boldsymbol{e}} = (\boldsymbol{A} - \boldsymbol{L}\boldsymbol{C})\boldsymbol{e} \tag{61}$$

The gain $\boldsymbol{L}$ can be obtained either through pole placement or LQR as $\dot{\boldsymbol{e}} = (\boldsymbol{A}^\mathsf{T} - \boldsymbol{C}^\mathsf{T}\boldsymbol{L}^\mathsf{T})\boldsymbol{e}$ [15].

6.2 Kalman filter

The Kalman filter, a state estimator, is typically separated into two stages: predict and update. In the predict stage, the current state and covariance is estimated using the previous state and covariance and the system dynamics. In the update stage, the current state and covariance is updated using the current observation. The Kalman filter is an optimal state estimator for a system described as:

$$\boldsymbol{x}_{k+1} = \boldsymbol{F}\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{u}_k + \boldsymbol{w}_k \tag{62}$$
$$\boldsymbol{w} \sim \mathcal{N}\{0, \boldsymbol{W}\} \tag{63}$$
$$\boldsymbol{y}_k = \boldsymbol{H}\boldsymbol{x}_k + \boldsymbol{v}_k \tag{64}$$
$$\boldsymbol{v} \sim \mathcal{N}\{0, \boldsymbol{V}\} \tag{65}$$

Predict:

$$\hat{\boldsymbol{x}}_{k|k-1} = \boldsymbol{F}\hat{\boldsymbol{x}}_{k-1|k-1} + \boldsymbol{B}\boldsymbol{u}_{k-1} \tag{66}$$
$$\boldsymbol{P}_{k|k-1} = \boldsymbol{F}\boldsymbol{P}_{k-1|k-1}\boldsymbol{F}_k^\mathsf{T} + \boldsymbol{W} \tag{67}$$

Update:

$$\boldsymbol{K}_k = \boldsymbol{P}_{k|k-1}\boldsymbol{H}^\mathsf{T}(\boldsymbol{H}\boldsymbol{P}_{k|k-1}\boldsymbol{H}^\mathsf{T} + \boldsymbol{V})^{-1} \tag{68}$$
$$\hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}(\boldsymbol{y}_k - \boldsymbol{H}\hat{\boldsymbol{x}}_{k|k-1}) \tag{69}$$
$$\boldsymbol{P}_{k|k} = (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H})\boldsymbol{P}_{k|k-1} \tag{70}$$

(71)

[15]

Equation (70) is known to have poor numerical stability as its outcome is not guaranteed to be symmetric nor positive definite.

The symmetric form of the covariance update is:

$$\boldsymbol{P}_{k|k} = \boldsymbol{P}_{k|k-1} - \boldsymbol{K}_k(\boldsymbol{H}\boldsymbol{P}\boldsymbol{H}^\mathsf{T} + \boldsymbol{V})\boldsymbol{K}^\mathsf{T} \tag{72}$$

The symmetric and positive Joseph form is:

$$\boldsymbol{P}_{k|k} = (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H})\boldsymbol{P}_{k|k-1}(\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H})^\mathsf{T} + \boldsymbol{K}_k\boldsymbol{V}\boldsymbol{K}_k^\mathsf{T} \tag{73}$$

[27]

6.2.1 Asymptotic form

The limit of $\boldsymbol{K}_k$ can be found as $k \to \infty$ if the conditions in [30] hold. And so the limit of $\boldsymbol{K}_k$ as $k \to \infty$ can be used in a discrete version of the Luenberger observer instead of $\boldsymbol{L}$.

Both a Kalman filter and Luenberger observer are unsuitable to estimate the states for a system as described as by Fourth Model, 4.4, using Fifth Model, 4.5, as the system is non-linear, and therefore cannot be estimated sufficiently by linear techniques. Non-linear state estimators should be used for this system as they are designed for non-linear systems with noisy measurements.

A Luenberger observer is best suited in order to observe the additional states added by a padé approximation as the original states can be estimated using a state estimator so noise can be ignored. A state estimator that directly estimates the additional state added by a padé approximation could be used, however this would require the state estimator to be changed depending on the controller being used.

6.3 Extended Kalman Filter

The EKF is an extension to the Kalman filter for non-linear systems of the form, where the state transition and observation functions are differentiable:

$$\boldsymbol{x}_k = f(\boldsymbol{x}_{k-1}, \boldsymbol{u}_{k-1}, \bullet) + \boldsymbol{w}_{k-1} \tag{74}$$
$$\boldsymbol{y}_k = h(\boldsymbol{x}_k, \bullet) + \boldsymbol{v}_k \tag{75}$$
$$\boldsymbol{w} \sim \mathcal{N}\{0, \boldsymbol{W}\} \tag{76}$$
$$\boldsymbol{v} \sim \mathcal{N}\{0, \boldsymbol{V}\} \tag{77}$$

It does this by linearising the system at each time step:

$$\boldsymbol{F}_k = \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\right|_{\hat{\boldsymbol{x}}_{k-1|k-1}, \boldsymbol{u}_{k-1}, \bullet} \tag{78}$$

$$\boldsymbol{H}_k = \left.\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}}\right|_{\hat{\boldsymbol{x}}_{k-1|k-1},\bullet} \tag{79}$$

Predict:

$$\hat{\boldsymbol{x}}_{k|k-1} = \boldsymbol{f}(\hat{\boldsymbol{x}}_{k-1|k-1}, \boldsymbol{u}_{k-1}, \bullet) \tag{80}$$

Also, equation (67) is used.

Update:

Equation (68) is used.

$$\hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}(\boldsymbol{y}_k - \boldsymbol{h}(\hat{\boldsymbol{x}}_{k|k-1})) \tag{81}$$

And any form of the covariance update.

The EKF is not an optimal state estimator unlike the Kalman filter. Additionally, EKFs are sensitive to errors in the process model or the initial state, where the filter may quickly diverge, and the estimated covariance matrix is underestimated as it assumes non-linear functions are linear when calculating the covariances. However, EKFs are the industry standard due to their ease of use.

Unscented kalman filters (UKFs) are similar in computation complexity to an EKF but better estimate the covariance matrix as it doesn't linearise the model. The unscented quaternion estimator (USQUE) is a variant of the UKF for quaternion state estimation. However, as shown in this [31], bias estimation can be poor even though settle time for orientation is fast.

To convert Fourth Model, 4.4, to a discrete state space representation, euler integration will be used. However, $\boldsymbol{q}_{k|k-1} = \boldsymbol{q}_{k-1|k-1} \otimes e^{\boldsymbol{\omega}_{k-1|k-1}\Delta t/2}$ will be used to update the unit quaternion as this equation conserves the unit length constraint of unit quaternions.

A popular choice to measure states such as acceleration, attitude and angular rate are inertial measurement units (IMUs) which can contain gyroscopes, accelerometers and possibly magnetometers. Additionally, a barometer and/or gps can be used to measure position and velocity

We will use a gyroscope, accelerometers and barometer. Let the gyroscope bias be $\boldsymbol{\omega}_b$ and noise $\boldsymbol{\omega}_n$. Let the accelerometer bias be $\boldsymbol{a}_b$ and noise $\boldsymbol{a}_n$. Let the barometer noise be ρ_n, the bias in the barometer is ignored as adding sea level pressure to the Kalman state in order to account for calibration error and biases in the barometer, performed poorly with sea level pressure changing drastically even though it was meant to stay relatively constant. This may have been due to an implementation error or could be fixed in future work.

The input to the functions are not listed for brevity. The measurement function, $\boldsymbol{h}$, in this case is:

$$[\boldsymbol{\omega}^b]_m^{bn} = [\boldsymbol{\omega}^b]^{bi} + \boldsymbol{\omega}_b^b + \boldsymbol{\omega}_n^b \tag{82}$$

$$= [\boldsymbol{\omega}^b]^{bn} + \boldsymbol{R}^{bn}[\boldsymbol{\omega}^n]^{ie} + [\boldsymbol{\omega}^b]_b^{bn} + \boldsymbol{\omega}_n^b \tag{83}$$

$$\text{where } [\boldsymbol{\omega}^n]^{ie} \approx \begin{bmatrix} 0 & 0 & 7.29 \times 10^{-5} \end{bmatrix} \text{ and } [\boldsymbol{\omega}^n]^{ni} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^\mathsf{T} \tag{84}$$

$$[\boldsymbol{a}^b]_m^{ii} = [\boldsymbol{a}^b]^{ii} + \boldsymbol{a}_b^b + \boldsymbol{a}_n^b - \boldsymbol{R}^{bn} \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix}^\mathsf{T} \tag{85}$$

$$\rho_m = \rho_0 \left(1 + \frac{L_b * r_z^n}{T_b}\right)^{\frac{-g_0 * M}{R^* * L_b}} + \rho_n \tag{86}$$

$$\boldsymbol{h}(\boldsymbol{x}_k, \bullet) = \begin{bmatrix} [\boldsymbol{\omega}^b]_m^{bn} \\ [\boldsymbol{a}^b]_m^{ii} \\ \rho_m \end{bmatrix} \tag{87}$$

[20]

6.4 Right Invariant Extended Kalman Filter

The EKF variant, right invariant extended kalamn filter (RIEKF), uses an EKF to estimate deviations from a estimated nominal state and these deviations are used to update the estimated nominal state. This method has many advantages over an EKF [27, 21, 9]:

- The orientation deviation will have 3 parameters, the same as its degrees of freedom. This does not lead to any problems, such as gimbal lock, as the deviation will always be close to the origin. This means that there is not a risk of a singularity in the covariance matrices due to over-parametrization and thus enforcing constraints. An EKF is unsuitable for attitude estimation as a continuous non singular parameterisation of $SO(3)$ has to be greater than three dimensional, as discussed in section 3, Dynamics, and this can lead to a singularity in the covariance matrices.
- The deviations should be small, so the computation of the Jacobians will be easy and fast, as second order products are negligible.
- 'The error dynamics are slow because all the large-signal dynamics have been integrated in the nominal-state. This means that we can apply KF corrections at a lower rate than the predictions' [27].

The nominal state $\boldsymbol{x}$ can be predicted with a system obtained from Fourth Model, 4.4,, and Fifth Model, 4.5,:

$$\hat{\boldsymbol{x}}_{k|k-1} = \boldsymbol{f}(\hat{\boldsymbol{x}}_{k-1|k-1}, \boldsymbol{u}_{k-1}, \bullet) \tag{88}$$

The error state $\boldsymbol{\delta x}$ can be modelled using estimated acceleration values $[\boldsymbol{a}^b]_m^{ii}$ by:

$$\begin{bmatrix} \boldsymbol{\delta p}^n \\ \boldsymbol{\delta}[\boldsymbol{v}^n]^n \\ \boldsymbol{\delta\theta}^n \\ \boldsymbol{\delta a}_b^b \\ \boldsymbol{\delta\omega}_b^b \\ [\boldsymbol{\omega}^b]^{bn} \end{bmatrix} \leftarrow \begin{bmatrix} \boldsymbol{\delta p}^n + \boldsymbol{\delta}[\boldsymbol{v}^n]^n \Delta t \\ \boldsymbol{\delta}[\boldsymbol{v}^n]^n + (-[\boldsymbol{R}^{nb}([\boldsymbol{a}^b]_m^{ii} - \boldsymbol{a}_b^b)]_\times \boldsymbol{\delta\theta}^n - \boldsymbol{R}^{nb}\boldsymbol{\delta a}_b^b)\Delta t + \boldsymbol{v}_i^n \\ \boldsymbol{\delta\theta}^n - \boldsymbol{R}^{nb}\delta\boldsymbol{\omega}_b^b \Delta t + \boldsymbol{\theta}_i^n \\ \boldsymbol{\delta a}_b^b + \boldsymbol{a}_{b_i}^b \\ \boldsymbol{\delta\omega}_b^b + \boldsymbol{\omega}_{b_i}^b \\ [\boldsymbol{\omega}^b]^{bn} + \boldsymbol{\omega}_i \end{bmatrix} \tag{89}$$

where $x \leftarrow f(x, \bullet)$ stands for a time update $x_{k+1} = f(x_k, \bullet)$ [27].

The measurement function is taken as defined in eqs. (82) to (87) and the measurement jacobian $\boldsymbol{H}$ is evaluated at the best true-state estimate $\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_{k|k-1} \oplus \hat{\boldsymbol{\delta x}}_{k|k-1}$.

$$\boldsymbol{H} = \left.\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{\delta x}}\right|_{\hat{\boldsymbol{x}}_k} \tag{90}$$

The Jacobians $\boldsymbol{F}$ and $\boldsymbol{H}$ can be partially seen in [27] or calculated using software, such as MATLAB. The truncated taylor series approximation of $\frac{sinx}{x}$ and $\arctan\frac{\|\boldsymbol{q}_v\|}{\boldsymbol{q}_w}$ will be used to avoid division by 0 in the Jacobians due to the presence of the quaternion exponential and logarithm.

Use predict and update steps from EKF where state is $\boldsymbol{\delta x}$

The nominal state update is defined as:

$$\hat{\boldsymbol{x}}_{k|k} = \begin{bmatrix} \boldsymbol{p}^n \\ [\boldsymbol{v}^n]^n \\ \boldsymbol{q}^{nb} \\ \boldsymbol{a}_b^b \\ \boldsymbol{\omega}_b^b \\ [\boldsymbol{\omega}^b]^{bn} \end{bmatrix} \leftarrow \begin{bmatrix} \boldsymbol{p}^n + \boldsymbol{\delta p}^n \\ [\boldsymbol{v}^n]^n + \boldsymbol{\delta}[\boldsymbol{v}^n]^n \\ e^{\boldsymbol{\delta\theta}^n/2} \otimes \boldsymbol{q}^{nb} \\ \boldsymbol{a}_b^b + \boldsymbol{\delta a}_b^b \\ \boldsymbol{\omega}_b^b + \boldsymbol{\delta\omega}_b^b \\ [\boldsymbol{\omega}^b]^{bn} \end{bmatrix} \tag{91}$$

Let $\boldsymbol{g}$ be the error reset function [27]:

$$\boldsymbol{\delta x} \leftarrow \boldsymbol{g}(\boldsymbol{\delta x}) = \begin{bmatrix} \boldsymbol{\delta p}^n - \hat{\boldsymbol{\delta p}}^n \\ \boldsymbol{\delta v}^n - \boldsymbol{\delta}\hat{[\boldsymbol{v}^n]}^n \\ \hat{\boldsymbol{\delta q}}^{bn} \otimes \boldsymbol{\delta q}^{nb} \\ \boldsymbol{\delta a}_b^b + \hat{\boldsymbol{\delta a}}_b^b \\ \boldsymbol{\delta\omega}_b^b + \hat{\boldsymbol{\delta\omega}}_b^b \\ [\boldsymbol{\omega}^b]^{bn} \end{bmatrix} \tag{92}$$

$$\boldsymbol{P}_{k|k} = \boldsymbol{G}_k \boldsymbol{P}_{k|k-1} \boldsymbol{G}_k^{\mathsf{T}} \tag{93}$$

$$\boldsymbol{G}_k = \left.\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{\delta x}}\right|_{\hat{\boldsymbol{\delta x}}_{k|k}} \tag{94}$$

In conclusion, the RIEKF should be used to estimate the state using sensor measurements due to its many advantages of an EKF. The advantages it has outweighs its increased complexity. Additionally, a Luenberger observer should be used to estimate the additional states when using a padé approximation for the time delays in the actuator.

7 Control

As outlined in the introduction, a controller is necessary in order to maintain the rocket on a specified trajectory which is necessary to achieve VTVL for the same reasons why guidance is important outlined in section 5, Guidance.

The system that will be analysed is a multi-input, multi-output (MIMO) system, this means that there are multiple inputs to the system (the forces and moments imparted on it) and there are multiple outputs (such as the attitude and position). MIMO systems are typically controlled using state space methods however, classical methods can be used if the system is represented using a transfer function matrix.

The output of the controller will be a vector of moments because this means attitude controllers can account for changing thrust easily. Additionally, attitude control is very important for horizontal velocity controllers as attitude control is the predominant way to control velocity.

7.1 Error Terms

Closed loop control requires error terms or the variable being controlled and setpoint to be fed in. Calculating the error term for quaternions is not as simple for other variables so the error terms that can be used will be presented here.

When using feedback control a error term is commonly used. Even though state space methods typically forgo this instead opting to add a multiple of the reference to the input directly, we will use an error term as the state size and input size are not the same.

The quaternion error is defined as the rotation from the desired orientation to the current orientation.

$$\boldsymbol{q}^{en} = \boldsymbol{q}^{bd} = \boldsymbol{q}^{bn} \otimes \boldsymbol{q}^{nd} \tag{95}$$

If the roll angle is not being controlled, e.g. because roll angular velocity is being controlled instead or because there is no control on the roll axis, the error term should not describe a rotation where control on the roll axis would be needed. This error term, $\boldsymbol{q}^{re}$, can be obtained as follows.

$$\tag{96}$$

$$\boldsymbol{q}^{en} = \boldsymbol{q}^{er} \otimes \boldsymbol{q}^{rn} \tag{97}$$

$$\begin{bmatrix} w \\ i \\ j \\ k \end{bmatrix} = \begin{bmatrix} \cos\theta/2 \\ x\sin\theta/2 \\ y\sin\theta/2 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} \cos\alpha/2 \\ 0 \\ 0 \\ \sin\alpha/2 \end{bmatrix} \tag{98}$$

$$= \begin{bmatrix} \cos\theta/2\cos\alpha/2 \\ x\sin\theta/2\cos\alpha/2 + y\sin\theta/2\sin\alpha/2 \\ x\sin\theta/2\sin\alpha/2 + y\sin\theta/2\cos\alpha/2 \\ \cos\theta/2\sin\alpha/2 \end{bmatrix} \tag{99}$$

$$\frac{k}{w} = \tan\frac{\alpha}{2} \tag{100}$$

$$\frac{\alpha}{2} = \arctan\frac{k}{w} \tag{101}$$

$$\boldsymbol{q}^{re} = \boldsymbol{q}^{rn} \otimes \boldsymbol{q}^{ne} \tag{102}$$

To obtain the shortest rotation, the rotation that is less than or equal to 180°, the negative of

the quaternion should be used if the rotation is greater than 180°.

$$\boldsymbol{q} = \begin{cases} \boldsymbol{q} & \text{if } \boldsymbol{q}_w \geq 0 \\ -\boldsymbol{q} & \text{otherwise.} \end{cases} \tag{103}$$

The error term used typically is 3d perhaps because this decreases size of state space representation and the scalar term provides no additional information. There are multiple different options proposed such as $\boldsymbol{q}_v$, $R\{q\}\boldsymbol{q}_v$, $2 * \boldsymbol{q}_w * \boldsymbol{q}_v$ or $2\log \boldsymbol{q}$ in [5, 10, 8].

The dynamics of $\boldsymbol{q}_v$ are easy to obtain from the dynamics of $\boldsymbol{q}$ which are usually known.

$R^{bn}\boldsymbol{q}_v^{bn}$ has tracking performance that is as good as using the euler angle parameterisation of $\boldsymbol{q}^{nb} \otimes \boldsymbol{q}^{dn}$ which describes the desired orientation in the body reference frame. A further advantage is represented by the contained computational burden.

$2 * \boldsymbol{q}_w^{bn} * \boldsymbol{q}_v^{bn}$ performs worse than $R^{bn}\boldsymbol{q}_v^{bn}$ for large orientation errors but similarly for small orientation errors. Its computational burden is lower than $R^{bn}\boldsymbol{q}_v^{bn}$. The quaternion here will be limited to rotations in the interval $(-\pi, \pi)$ otherwise the norm will decrease as rotation increases in the interval $(\pi, 2\pi)$. Additionally describing the dynamics of the state including this term will be more complicated than $\boldsymbol{q}_v^{bn}$.

$2\log \boldsymbol{q}^{bn}$ is an axis angle parameterisation of $\boldsymbol{q}^{bn}$. The magnitude of the error term is therefore equal to the error angle so a larger control output is given for larger errors, which is ideal.

The error term $\boldsymbol{q}_v^{bn}$ will be used as its dynamics are simple and the error term is simple.

7.2 Proportional-Integral-Derivative Controller

PID control is a form of closed loop control that aims to minimise the error $e(t)$ in the long run, typically the difference between the desired value $r(t)$ and the measured value $y(t)$, by manipulating the control variable $u(t)$. A PID controller uses the derivative, integral of the error and the error itself to generate a control variable $u(t)$ through the equation:

$$u(t) = K_p e(t) + K_i \int_0^t e(t)dt + K_d \frac{\mathrm{d}}{\mathrm{d}t} e(t) \tag{104}$$

where K_p, K_i, K_d are values that are obtained from tuning and may be constant or a function, e.g. of time. The K_p, K_i, K_d terms have various effects on the controller's performance as summarised in table 1. Some effects not included in the table are that if the controller saturates the K_i term can worsen performance as the integral increases leading to overshoot. Techniques such as anti-windup can be employed to deal with this problem. The K_d term can worsen performance if there is significant measurement noise in the error term, especially high frequency noise. However, a filtered derivative can be used where a low pass filter is used to ignore high frequency noise. The K_d term can also worsen stability if a transport delay exists [24]. The K_i or K_d terms may sometimes be set to 0 to improve performance or due to difficult in tuning, especially with K_d.

Cascade control, when the output on the first PID controller sets the setpoint for the second controller, allows for PID controllers to control MIMO systems and can improve the response

to disturbances. Typically the first PID controller controls a slow parameter whilst the second controller deals with a more rapidly changing parameter [6].

PID control has limitations if the system is highly non-linear, open loop unstable, has lots of delay or is non minimum phase, more advanced controllers may be necessary [13].

TABLE 1 **Effects of independent P, I, and D tuning on closed-loop response. For example, while K_I and K_D are fixed, increasing K_P alone can decrease rise time, increase overshoot, slightly increase settling time, decrease the steady-state error, and decrease stability margins.**

	Rise Time	Overshoot	Settling Time	Steady-State Error	Stability
Increasing K_P	Decrease	Increase	Small Increase	Decrease	Degrade
Increasing K_I	Small Decrease	Increase	Increase	Large Decrease	Degrade
Increasing K_D	Small Decrease	Decrease	Decrease	Minor Change	Improve

[24]

7.3 Linear Quadratic Regulator

To obtain a LQR controller a LTI state space representation has to be used. These are the operating points used to linearise the model, as these are the expected values for the control period, as well as the mean of Thrust and $\boldsymbol{I}$

$$[\boldsymbol{\omega}^b]^{bn} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^\mathsf{T} \tag{105}$$

$$\boldsymbol{u}(t) = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^\mathsf{T} \tag{106}$$

$$\boldsymbol{q} = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}^\mathsf{T} \tag{107}$$

These are the ranges that will be used for gain scheduling, where the first and last terms are the first term and last terms in the range and the middle term is the difference between each term in the range:

$$\text{Thrust} = 0.5 : 7 : 28.5 \tag{108}$$

$$\boldsymbol{q}_i = \boldsymbol{q}_j = -0.7 : 0.1 : 0.7 \tag{109}$$

$\boldsymbol{q}_z$ is not included as this should stay close to 0 for our purposes and this would limit the ranges of $\boldsymbol{q}_i$ and $\boldsymbol{q}_j$.

Each controller will be an infinite horizon LQR controller with four variants that are continuous or discrete and gain scheduled or not gain scheduled.

An attitude controller can be designed using Third Model, 4.3, and Seventh Model, 4.7.

An attitude controller accounting for the actuator time delay can be designed using Third Model, 4.3, and Sixth Model, 4.6.

A horizontal velocity controller can be designed using Second Model, 4.2, and Seventh Model, 4.7.

A horizontal velocity controller accounting for the actuator time delay can be designed using Second Model, 4.2, and Sixth Model, 4.6.

7.4 Model Predictive Control

MPC numerically solves for an optima of the cost function over a time horizon, typically short, subject to hard constraints and then returns the control input at the current time step k. If the optimisation problem is convex the global optimum will be found otherwise a optima will be found with no guarantee that it is the global optimum. It is possible to achieve stability without requiring the globally optimum solution [16]. MPC is computationally more expensive than LQR and may give suboptimal solutions as the time horizon is typically shorter than the whole horizon and the global minimum may not be found for non-convex problems. However, MPC can handle non-linear systems and hard constraints and allows for custom cost functions unlike controllers such as LQR.

MPC has many advantages over other controllers, including but not limited to: ease of use and tuning; quick to develop and modify; explicitly handles constraints and a model [26].

The theory about tuning MPCs requires large prediction horizons or a terminal constraint. Adding a terminal state constraint to MPC leads to the MPC being equivalent to an infinite horizon LQR controller. Also, performance can be usually improved with a terminal cost function [16].

MPC control is only discrete however, as it is typically implemented digitally this is desired.

MPCs controllers that are equivalent to LQRs controllers will not be designed as they are unlikely to provide any benefit as only a constant constraint on the controller's output will be present and it is unlikely to be enforced and MPC will more computationally taxing than LQR.

A linear MPC using Third Model, 4.3, and both Sixth Model, 4.6, and Seventh Model, 4.7, will be designed. The linearisation will take place using the same operating points as in subsection 7.3, Linear Quadratic Regulator.

A non-linear MPC using Third Model, 4.3, and Fifth Model, 4.5, will also be designed.

7.5 Results

100 simulations were run each with random conditions for each controller for the time the motor was burning, 3.45s, and the mean, variance and maximum of the controlled variable was recorded. State estimators were not used in order to not affect the results due to the performance of the state estimator.

7.5.1 Attitude Control

The following results depict the mean, variance and maximum of the axis angle parameterisation, explained in section 2, Background, of the rocket's orientation as the desired orientation was upright, the identity orientation.

	No Control	Continuous LQR	Continuous gain scheduled LQR	Discrete LQR	Discrete gain scheduled LQR	Discrete time delay approximating LQR	Discrete time delay approximating gain scheduled LQR
mean	0.1689	0.1922	0.1923	0.1919	0.1920	0.1898	0.1899
	0.9254	0.9197	0.9196	0.9195	0.9195	0.9201	0.9201
	0.1072	0.0820	0.0820	0.0886	0.0886	0.0820	0.0821
	14.0318	0.3595	0.3595	0.4057	0.4057	0.3843	0.3843
variance	0.0020	0.0066	0.0066	0.0066	0.0066	0.0060	0.0061
	0.0525	0.0522	0.0522	0.0522	0.0522	0.0525	0.0525
	0.0492	0.0518	0.0518	0.0511	0.0511	0.0521	0.0521
	367.5029	0.0646	0.0646	0.0830	0.0830	0.0769	0.0769
maximum	0.2217	0.4378	0.4380	0.4390	0.4393	0.4241	0.4244
	0.9864	0.9935	0.9935	0.9930	0.9930	0.9924	0.9924
	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	76.1426	0.6687	0.6688	0.7562	0.7562	0.7263	0.7264

Continuous time delay approximating LQR	Continuous time delay approximating gain scheduled LQR	Continuous time delay approximating PID	Discrete time delay approximating PID	Discrete MPC	Discrete time delay approximating MPC
0.1898	0.1898	0.1781	0.1233	0.1837	0.1843
0.9204	0.9204	0.8700	0.6269	0.9277	0.9266
0.0697	0.0697	0.3086	0.6265	0.0762	0.0645
0.3475	0.3475	0.4808	0.9413	0.3963	0.6563
0.0060	0.0060	0.0039	0.0044	0.0045	0.0047
0.0525	0.0525	0.0513	0.0912	0.0523	0.0523
0.0534	0.0534	0.0609	0.1049	0.0523	0.0536
0.0627	0.0627	0.1356	0.4676	0.0942	0.2548
0.4219	0.4221	0.3280	0.2158	0.3611	0.3827
0.9927	0.9927	0.9848	0.9841	0.9937	0.9919
1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
0.6599	0.6599	1.1246	2.6143	0.8829	1.3588

All the controllers have the same maximum value on the z axis due to the initial axis angle being $[0\ 0\ 1\ 0]^\mathsf{T}$. The non-linear MPC was unable to control the system, perhaps due to an implementation error, and due to its high computational burden it is unlikely to be fast enough to run in real time even on professional hardware. This is illustrated by the work in convexifying non-convex problems to speed up solutions and guarantee finding a solution in a known number

of operations, whilst we were trying to solve a non-convex problem using the non-linear MPC. Therefore, results from the non-linear MPC were not collected.

The controllers all perform better in terms of the error angle in all metrics compared to the baseline, the lack of a controller. MPC does not perform better than LQR, which may be due to only a constant constraint on the controller's output being placed which is unlikely to be exceeded. The continuous controllers consistently perform better than discrete controllers, this may be due to the roll actuators not being discrete or the time delays not being a multiple of the sample time. PID controllers perform the worst out of all the types, this may be due to poor tuning or due to the PID controllers using the derivative of the error term instead of angular velocity like the rest of the controllers. Controllers which use a more accurate model of the system taking into account the time delay in the actuator tended to perform better better than controllers that do not however the MPC controllers performed worse when accounting for the time delay than not. However, this may have been due to poor tuning which if true highlights that tuning MPC controllers is difficult. Gain scheduling provides no significant benefit which may be due to likely due to deviations from the operating points being minor, so the difference between the non-linear controllers and linear controllers are minimal.

Overall, non-linear controllers performed no better than linear controllers.

7.5.2 Horizontal Velocity Control

The following results depict the mean, variance and maximum of the error in the x and y velocities ($[\boldsymbol{v}^n_{x,y}]^n$) from the desired velocities, $[0\ 0]^\mathsf{T}$.

	No Control	Continuous LQR	Discrete LQR	Discrete gain scheduled LQR	Discrete time delay approximating gain scheduled LQR	Continuous time delay approximating LQR	Continuous time delay approximating gain scheduled LQR
mean	1.8519	0.0255	0.0268	0.0219	0.0218	0.0271	0.0268
	0.3205	0.0066	0.0069	0.0057	0.0057	0.0070	0.0069
variance	7.0541	$10^{-3}\times$ 0.2022	$10^{-3}\times$ 0.2252	$10^{-3}\times$ 0.1443	$10^{-3}\times$ 0.1466	$10^{-3}\times$ 0.2315	$10^{-3}\times$ 0.2223
	0.2229	$10^{-3}\times$ 0.0359	$10^{-3}\times$ 0.0399	$10^{-3}\times$ 0.0276	$10^{-3}\times$ 0.0287	$10^{-3}\times$ 0.0412	$10^{-3}\times$ 0.0400
maximum	9.8716	0.0440	0.0463	0.0366	0.0414	0.0459	0.0461
	1.8027	0.0247	0.0256	0.0220	0.0227	0.0263	0.0262

The controllers all perform better in terms in all metrics compared to the baseline, the lack of a controller. Gain scheduling provides no significant benefit over time delay approximating controllers and led to instability for controllers not accounting for the actuator's time delay. The non time delay approximating continuous controller performs better than the discrete one

however the discrete time delay approximating controllers perform better than the continuous one. Therefore, in general a continuous controllers has no benefit over a discrete controller and vice versa.

Overall, for horizontal velocity control non-linear control performs worse than linear control, this may be due to poor tuning, errors in the simulation or an incorrect implementation of gain scheduling.

Tests were performed from an upside down position however results were not collected as changing the $\boldsymbol{Q}$ or $\boldsymbol{R}$ matrices led to larger than expected variability and making controllers less aggressive improved performance. This may have been due to incorrect aerodynamic modelling, an extremely unstable rocket that could not be controlled with the limited actuators or or an incorrect implementation of gain scheduling. In preliminary results it was difficult to see a significant difference between controllers that couldn't be easily fixed by adjusting the tuning. As tuning the controllers for this scenario was difficult and would have taken too long, results were therefore omitted.

Additionally, when the upside down position was rolled 90° no controller could stabilise the rocket. The cause for this is unknown however, it may have been due to an error in the controllers which may have been subsequently fixed.

8 Conclusion

In this paper, different GNC schemes that could achieve VTVL were compared. The results collected from the proposed controllers in this paper demonstrate the validity of controlling the attitude or velocity. The RIEKF is shown in [31, 20, 21] to perform sufficiently well and the control should work well when using the estimated state however, future work is needed to confirm this. Whilst results were not collected for the guidance scheme, preliminary tests indicate that VTVL can be achieved using the proposed guidance laws, with the algorithm proposed in subsection 5.2, Landing Trajectory Optimisation, minimising impact velocity. So VTVL should be achieved if a suitable GNC scheme from this paper is implemented.

Time delay approximating LQR controllers perform best in terms of mean, variance and maximum error. Additionally, MPC controllers can perform the same as infinite horizon LQR controllers so MPC would also perform as well as time delay approximating LQR controllers however, it would be computationally more expensive and more complicated to implement so LQR is overall the best. Controllers which accounted for the time delay present in the actuators consistently performed better than those that did not in terms of mean, variance and maximum error. The non-linear controllers proposed on the whole provide little benefit or worsen performance compared to their linear counterpart. This may be due to poor tuning, more advanced non-linear controllers being needed or the deviation from the operating points being minimal and so further testing would be required with greater deviations from the operating points.

Results were not collected for the navigation schemes as extensive results have already been collected as can be seen in [31, 20, 21]. These results has led me to conclude that the RIEKF would perform the best out of all the state estimator and observers discussed in this paper. Whilst a

Luenberger observer is used for the additional states added by using a padé approximation as the dynamics are linear and as there should be little to no noise in the states as they have been approximated using a state estimator. Additionally, a Kalman filter can be implemented as a Luenberger observer if the limit of its gain is taken as $k \to \infty$.

The guidance schemes proposed for landing are all non-convex and non-linear however, the problems could be convexified to improve the speed at which solution is found. Linear schemes could not be used for landing as the dynamics of the system cannot be accurately modelled using linear dynamics for the whole flight. However, linear schemes are sufficient for take-off where perhaps horizontal velocity is being controlled.

Achieving VTVL in model rockets allows for the GNC schemes to be cheaply and easily tested, so they can then be implemented of professional rockets. As discussed before this will greatly benefit society as whole. The GNC schemes presented could also be used to develop small rockets that launch small satellites into orbit as discussed in [18], which will greatly increase research by scientists and engineers across a wide range of fields. The navigation and control can be implemented on board however, the guidance scheme would need to be improved to increase the speed at which a solution is found. Furthermore, VTVL can inspire children and teenagers into science, technology, engineering, and mathematics (STEM).

9 Future Work

There is a lot of future work that could be done to improve upon this paper. For example, results could be obtained for the computational complexity, robustness, settling time, overshoot and stability of the controllers and results could be graphed in order to better compare different controllers and determine their performance. Additionally, H_∞, finite horizon LQR, feedforward and other controllers could also be evaluated. Empirical results could also be obtained for a more accurate comparison between controllers. The optimal time to ignite motors when using trajectory optimisation algorithm could also be solved for. Additionally, optimisation problems could be sped up, so guidance can run in real time. Also, a global navigation satellite system (GNSS) sensor could be added to the EKF and RIEKF as preliminary tests show low accuracy in horizontal position and velocity estimates with current sensors. The discrete time quaternion update could be used in discrete control models to improve the accuracy of the models and therefore controller performance. More accurate aerodynamics could be obtained, e.g. from wind tunnel testing. A more accurate model of the dynamics could improve performance, e.g. by incorporating aerodynamics into models being used by GNC. Other state estimators such as USQUE could be explored to improve performance. Instead of switching to the shorter rotation automatically, the rotation could only be switched if it would decrease the time to reaching the desired orientation. A more accurate model of the actuator could be obtained so the results are more indicative of real world performance. The angular velocity deviation could be added to the RIEKF instead of using the angular velocity. Finally, roll control could be separated from pitch and yaw control to allow it to run at a higher frequency and so improve performance.

A Appendix

A.1 Coefficients

[23] planform area of ellipsoid nose cone is $\frac{dl\pi}{4}$
Planform Area of Conical Nose cone is πrl
Planform Area of Cyclinder is $2rl$
Reference Area is πr^2

$$C_N = \sum_{i \in Components} K \frac{A_{plan}}{A_{ref}} \text{cnmul} \sin^2 \alpha + \frac{2 \sin \alpha}{A_{ref}} [A(l) - A(0)] \tag{110}$$

$$\tag{111}$$

$$= C_N^{Body\ Tube} + C_N^{Nose\ Cone} \tag{112}$$

$$\tag{113}$$

$$mul = 3 \left(\frac{0.275 * \bar{d}}{A_{ref} * d_{ref}} * 0.95^4 \right) \tag{114}$$

$$C_{m\ damping} = min \left(\sum_{i \in Components} \frac{C_N * CP_x}{l}, mul * \frac{\omega_y^2}{\|\boldsymbol{v}_\infty\|^2} \right) * \text{sign}(\omega_y) \tag{115}$$

$$C_m = \left(\sum_{i \in Components} \frac{C_N * CP_x}{d_{ref}} \right) - \frac{C_N * CM_x}{d_{ref}} - C_{m\ damping} \tag{116}$$

[14]

$$C_{base} = \begin{cases} 0.12 + 0.13M^2 & \text{if } M < 1 \\ 0.25/M & \text{if } M > 1 \end{cases} \tag{117}$$

$$C_{pressure} = 0.8 \frac{d_{ref}}{2\sqrt{\frac{d_{ref}}{2}^2 + l^{Nose\ Cone^2}}} \tag{118}$$

$$Cfc1 = (\frac{3}{2} \ln Reynolds - 5.6)^{-2} \tag{119}$$

$$1 - \frac{mach^2}{10} \tag{120}$$

$$\max(Cfc1, 0.032) \tag{121}$$

$$\begin{aligned} C_s &= 0 \\ C_y &= 0 \\ C_l &= 0 \end{aligned} \tag{122}$$

$$\boldsymbol{v}_\infty = [\boldsymbol{v}^n]^n + v_{wind} \tag{123}$$

$$\alpha = \begin{cases} 0 & \text{if } v_\infty = 0 \\ \arccos\left(\frac{v_{\infty_z}}{\|v_\infty\|}\right) & \text{otherwise.} \end{cases} \tag{124}$$

$$\begin{aligned} length &= \sqrt{v_{\infty_x}^2 + v_{\infty_y}^2} \\ sin &= \frac{v_{\infty_y}}{length} \\ cos &= \frac{v_{\infty_x}}{length} \\ \mathbf{Rot} &= \begin{bmatrix} cos & -sin & 0 \\ sin & cos & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{125}$$

Glossary

Attitude The orientation of the rocket with respect to a reference

Abbreviations

CFD Computational Fluid Dynamics 11

EKF Extended Kalman Filter 7, 20–23

GNC Guidance, Navigation And Control 6, 7, 11, 13–15, 30

IMU Inertial Measurement Unit 21

LQG Linear Quadratic Gaussian 7

LQR Linear Quadratic Regulator 7, 9–11, 14, 15, 19, 26–30

LTI Linear Time Invariant 9, 10, 26

MIMO Multi-input, Multi-output 24, 25

MPC Model Predictive Control 6, 11, 27–29

MRP Modified Rodrigues Parameter 11

PID Proportional-integral-derivative 6, 7, 15, 18, 25, 26, 28, 29

RIEKF Right Invariant Extended Kalamn Filter 22, 23, 30

RLV Reusable Launch Vehicle 6

TVC Thrust Vector Control 6, 7, 13

UKF Unscented Kalman Filter 21

USQUE Unscented Quaternion Estimator 21

VTVL Vertical Take Off/ Vertical Landing 6, 7, 13, 16, 18, 23, 30

Coordinate Systems

n Navigation frame

b Body frame

i Inertial frame

e Earth frame

Bibliography

[1] A. Behcet Acikmese and Scott Ploen. "A Powered Descent Guidance Algorithm for Mars Pinpoint Landing". In: *AIAA Guidance, Navigation, and Control Conference and Exhibit*. American Institute of Aeronautics and Astronautics, Aug. 2005. DOI: 10.2514/6.2005-6288.

[2] Behcet Acikmese, John M. Carson, and Lars Blackmore. "Lossless Convexification of Nonconvex Control Bound and Pointing Constraints of the Soft Landing Optimal Control Problem". In: *IEEE Transactions on Control Systems Technology* 21.6 (Nov. 2013), pp. 2104–2113. DOI: 10.1109/tcst.2012.2237346.

[3] Behcet Acikmese and Scott R. Ploen. "Convex Programming Approach to Powered Descent Guidance for Mars Landing". In: *Journal of Guidance, Control, and Dynamics* 30.5 (Sept. 2007), pp. 1353–1366. DOI: 10.2514/1.27553.

[4] Joe Barnard. *BPS.space*. URL: https://bps.space/ (visited on 09/19/2019).

[5] Mogens Blanke and Martin Birkelund Larsen. *Satellite Dynamics and Control in a Quaternion Formulation (2nd edition)*. English. Lecture note for course 31365. Version 2f - September, 2010. Technical University of Denmark, Department of Electrical Engineering, 2010.

[6] William Bolton. *Instrumentation and Control Systems (Second Edition)*. Newnes, 2015.

[7] Oreste S. Bursi. *State Observers and the Kalman filter*. July 2012. URL: http://www.ing.unitn.it/~bursi/Download/PHD_LECTURES_02JULY/Observer%20and%20Kalman%20Filter_28%20May%202011%20_v2_27062012.pdf.

[8] Fabrizio Caccavale et al. "Resolved-acceleration control of robot manipulators: A critical review with experiments". In: *Robotica* 16.5 (Sept. 1998), pp. 565–573. DOI: 10.1017/s0263574798000290.

[9] John L. Crassidis and F. Landis Markley. "Unscented Filtering for Spacecraft Attitude Estimation". In: *Journal of Guidance, Control, and Dynamics* 26.4 (July 2003), pp. 536–542. DOI: 10.2514/2.5102.

[10] Travis DeWolf. *Force control of task-space orientation*. Dec. 2018. URL: https://studywolf.wordpress.com/2018/12/03/force-control-of-task-space-orientation/ (visited on 07/24/2020).

[11] James Diebel. "Representing Attitude: Euler Angles, Unit Quaternions, and Rotation Vectors". In: *Matrix* 58 (Jan. 2006).

[12] Tian Dong, Changjian Zhao, and Zhiguo Song. "Autopilot Design for a Compound Control Small-Scale Solid Rocket in the Initial Stage of Launch". In: *International Journal of Aerospace Engineering* 2019 (Mar. 2019), pp. 1–10. DOI: 10.1155/2019/4749109.

[13] Brian Douglas. *Understanding PID Control, Part 4: A PID Tuning Guide*. MATLAB. July 2018. URL: https://www.youtube.com/watch?v=sFOEsA0Irjs (visited on 09/10/2020).

[14] Eugene L. Fleeman. *Tactical missile design*. 2nd ed. American Institute of Aeronautics and Astronautics, 2006.

[15] Bernard Friedland. *Control system design: an introduction to state-space methods*. Dover Publications, 2005.

[16] Dimitry Gorinevsky. *Lecture 14 - Model Predictive Control*. 2004. URL: http://web.stanford.edu/class/archive/ee/ee392m/ee392m.1056/Lecture14_MPC.pdf.

[17] Jonathan How and Emilio Frazzoli. *16.30 Feedback Control Systems. Fall 2010. Massachusetts Institute of Technology: MIT OpenCourseWare*. URL: https://ocw.mit.edu/courses/aeronautics-and-astronautics/16-30-feedback-control-systems-fall-2010/lecture-notes/MIT16_30F10_lec05.pdf.

[18] Florian Kehl, Ankur M. Mehta, and Kristofer S. J. Pister. "An Attitude Controller for Small Scale Rockets". In: *Springer Tracts in Advanced Robotics*. Springer International Publishing, 2015, pp. 201–214. DOI: 10.1007/978-3-319-07488-7_14.

[19] Aliyu Bhar Kisabo, Aliyu Funmilayo Adebimpe, and Sholiyi Olusegun Samuel. "Pitch Control of a Rocket with a Novel LQG/LTR Control Algorithm". In: *Journal of Aircraft and Spacecraft Technology* 3.1 (Jan. 2019), pp. 24–37. DOI: 10.3844/jastsp.2019.24.37.

[20] Manon Kok, Jeroen D. Hol, and Thomas B. Schön. "Using Inertial Sensors for Position and Orientation Estimation". In: *Foundations and Trends® in Signal Processing* 11.1-2 (2017), pp. 1–153. DOI: 10.1561/2000000094.

[21] Venkatesh Madyastha et al. "Extended Kalman Filter vs. Error State Kalman Filter for Aircraft Attitude Estimation". In: *AIAA Guidance, Navigation, and Control Conference*. American Institute of Aeronautics and Astronautics, Aug. 2011. DOI: 10.2514/6.2011-6615.

[22] NASA. *Gimballed Thrust*. URL: https://www.grc.nasa.gov/WWW/k-12/rocket/gimbaled.html (visited on 09/19/2019).

[23] Sampo Niskanen. "Development of an Open Source model rocket simulation software". PhD thesis. 2009. URL: https://github.com/openrocket/openrocket/releases/download/Development_of_an_Open_Source_model_rocket_simulation-thesis-v20090520/Development_of_an_Open_Source_model_rocket_simulation-thesis-v20090520.pdf.

[24] "PID control system analysis and design". In: *IEEE Control Systems* 26.1 (Feb. 2006), pp. 32–41. DOI: 10.1109/mcs.2006.1580152.

[25] Junjian Qi, Ahmad F. Taha, and Jianhui Wang. "Comparing Kalman Filters and Observers for Power System Dynamic State Estimation With Model Uncertainty and Malicious Cyber Attacks". In: *IEEE Access* 6 (2018), pp. 77155–77168. DOI: 10.1109/access.2018.2876883.

[26] James B. Rawlings, David Q. Mayne, and Moritz M. Diehl. *Model predictive control theory, computation, and design*. Nob Hill Publishing, 2017.

[27] Joan Solà. *Quaternion kinematics for the error-state Kalman filter*. 2017. arXiv: 1711.02508 [cs.RO].

[28] John Stuelpnagel. "On the Parametrization of the Three-Dimensional Rotation Group". In: *SIAM Review* 6.4 (Oct. 1964), pp. 422–430. DOI: 10.1137/1006093.

[29] R. Sumathi and M. Usha. "Pitch and Yaw Attitude Control of a Rocket Engine Using Hybrid Fuzzy- PID Controller". In: *The Open Automation and Control Systems Journal* 6.1 (Apr. 2014), pp. 29–39. DOI: 10.2174/1874444301406010029.

[30] Jean Walrand and Antonis Dimakis. *State Observers and the Kalman filter*. Aug. 2006. URL: https://people.eecs.berkeley.edu/~wlr/226F06/226a.pdf (visited on 09/10/2020).

[31] M. Zamani, J. Trumpf, and R. Mahony. *Nonlinear Attitude Filtering: A Comparison Study*. 2015. arXiv: 1502.03990 [cs.SY].