A Thesis

entitled

Variable Pitch quadrotor Control with Blade Element Theory

by

Paul Hans Guentert

Submitted to the Graduate Faculty as partial fulfillment of the requirements for the

Master of Science Degree in Department of Mechanical and Materials Engineering

Dr. Manish Kumar, Committee Chair

Dr. Rajnikant Sharma, Committee Member

Dr. David Thompson, Committee Member

Dr. Teik C. Lim, Dean
College of Graduate Studies

University of Cincinnati
March 2017

Abstract

Variable Pitch quadrotor Control with Blade Element Theory

by

Paul Hans Guentert

Submitted to the Graduate Faculty as partial fulfillment of the requirements for the Master of Science Degree in Department of Mechanical and Materials Engineering

University of Cincinnati
March 2017

Thesis Supervisor: Manish Kumar

Title: Associate Professor

Acknowledgments

All ideas expressed in this paper that are not explicitly noted by the author to be his should assumed to have come from the work presented in the reference section of the paper. The author has made his best efforts to give credit to the work of all the notable and hardworking authors that supersedes this work.

Hans Guentert

Contents

List of Figures

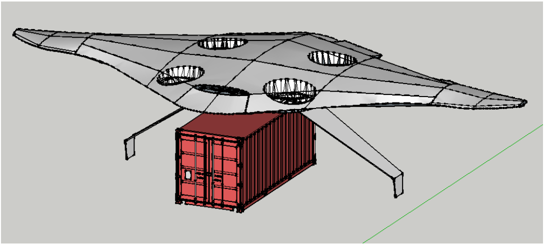
Chapter 1

Introduction

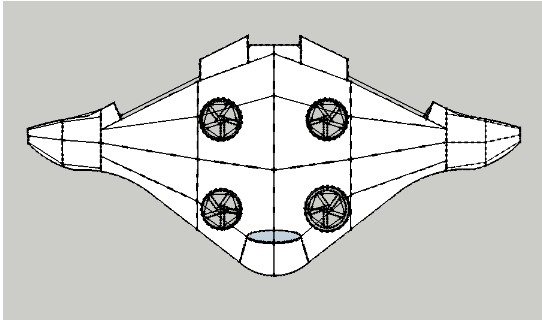
1.1 Motivation - done

During a Senior Capstone Project three years ago, my team and I built an autonomous quadrotor; my interests were peaked technically, theoretically, and academically. Unmanned autonomous vehicles (UAVs) have expanded from simple toys and odd interest to an industry of autonomous functionality and complex problem solving. Quadrotors have enabled short-term low altitude dexterous exploration, monitoring, and task management. Examining quadrotors from a technical perspective in terms of hardware and coding, the necessity for theoretical fundamentals such as reference frames, linear control, and non-linear control methods became apparent. This fostered an interest in system modeling and controls as well as further study in current systems that can be modeled in simulation and tested in experimentation.

While working through a class project in a Systems Engineering course, customer requirements for vertical takeoff/vertical landing vehicles that use heavy fuel started an interesting thought experiment on the implementation of quadrotors in place. However, the change of inertia in the blades and required change in torque in the blades made scalability of quadrotors difficult. Variable pitch blades provide a partial solution by generating thrust through change in pitch, but more questions arise: Can we build a simulation robust enough for scalability? Can we build a heavy lift/heavy fuel variable pitch quadrotor?



(a) Heavy List Transport



(b) Concept Drawing

Figure 1-1: Heavy Lift Concept Drawings

Cursory digging into variable pitch quadrotors unveiled a research domain of versatile functionality that variable pitch quadrotors could achieve instantaneous inverted thrust. As noted in [7], "traditional pod-and-boom autonomous helicopters have demonstrated agile maneuvers outside of the flight regime of small fixed-pitch quadrotors". These features implemented on a quadcopter should increase the range of achievable trajectories over traditional quadcopters. Such uniqueness of this platform hosts a plethora of unknown applications.

Initial experimentation began with a variable pitch quadrotor to see if autonomous way-point navigation could be achieved in an existing traditional quadrotor controller. Initial flight tests proved successful in terms of attitude stabilization and paved a path for proving theoretically that a variable pitch quadrotor could fly with autonomous way-point navigation using existing quadrotor control methods with a few modifications.

1.2 Purpose - done

Initiated by the challenge to construct a variable pitch quadrotor that flies using way-points, the purpose of this research investigation is to pursue the construction of a variable pitch quadrotor controller that uses way-points and trajectories. As the initial investigation proved successful, it deemed worthwhile to take a systematic approach to modeling the variable pitch quadcopter to fundamentally understand and achieve the features of inverted flight. The process began with developing a mathematical model that describes the flight dynamics of the variable pitch quadrotor. The initial experimental implementation focus uses a Pixhawk running PX4 firmware, an open source platform that is versatile and offers a good structure for closed loop control. It proved useful as a working framework for development. Simulation of different controller methods are used as proof of concept and perform analysis of controller performance. The first approach used a traditional quadrotor controller that takes roll, pitch, yaw, and thrust commands and mixes the control inputs to individual motor commands. These motor commands are manipulated to produce similar pitch commands. The control signal that operates on a normalized range of 0-1 was readjusted to the PWM range of the servos. This approach was successfully implemented on hardware.

A simulation model was constructed to model the variable pitch dynamics and demonstrate the required control inputs. This systematic approach uses blade element theory to model the desired pitch angle in simulation. The first attempted simulation approach used Euler angles with PID position, attitude, and attitude-rate controllers. This controller performs well but has points of failure if an inverted flight controller were to be pursued because singularities occur in Euler-based controllers. It seems appropriate to pursue a quaternion-based control method to enable the inverted flight features of variable pitch quadrotors. The final controller design combines innovative methods from quaternion-based controllers, blade element theory based motor models, and allows for differential-flatness-based control schemes. The simulation model follows the back-stepping control process for variable pitch control presented in [12] that describes the motor model in Blade Element Theory to derive the force of the blades of the quadrotor. In this process, the derivative of the forces acting on the rigid body is performed. This operation results in demanding an input of angular jerk. To calculate angular jerk, second order error dynamics are used. This process is demonstrated in [12].

The goal of the simulation model is to replicate the above process in a quaternion representation to achieve inverted flight trajectories. Separation from the irrational concept of angles [5], representation of force vectors, and consequent derivation of orientation directly into quaternions is also used. On top of this process, a Linear Quadratic Regulator is used in tandem with deferentially flat trajectories and tested in simulation. An area of concern is identified in Section 2: quaternion operations do not easily translate to algebra operations of second order error dynamics. Further areas of concern include delineating body frame quaternion, world frame quaternion, body rate frames, transformation from world to body and back, and when to use unit quaternions.

1.3 Contributions - done

The main contributor to this work is the process and construction of a variable pitch quadrotor simulation using a dynamic motor model in conjunction with quaternion based attitude controller. A PID control is implemented to produce the desired angular acceleration as inputs to the motor model. A linear quadratic regulator is implemented as a position controller and proven in simulation successful in following differentially flat trajectories. This work shows LQR implementation to quaternion based attitude controller that uses force vector representations. It lastly shows experimental proof of quad dynamics with existing variable pitch frames. This work is different from [5] because it uses a simplified motor model and is different from [12] because it uses quaternion based attitude control.

Chapter 2

Literature Review

2.1 Vertical Takeoff and Vertical Landing Vehicles - done

Aerial vehicles have been the subject of research endeavors with breakthroughs cascading through both industry and society. [3] Vertical takeoff and vertical landing craft, aerial vehicles characterized by their ability to vertically takeoff from a stationary position without use of a runway, have been studied since the 19th century but development was limited until the invention of the engine (as noted by Thomas Edison). Unlike typical winged aircraft, the take-off and landing sequence requires downward generated thrust from a propeller or redirected exhaust from a turbine engine. [30] Innovators and pioneers led the development into today's companies: Bell, Sikorsky, Boeing-Vertol helicopters; VTOL craft; to name a few. These craft would enable closer proximity to destinations and landing in areas which airplanes could not reach due to the limitations of large areas for runways access.

Vertical take-off and vertical landing vehicles are a subgroup of aerial vehicles that serve a particular mission which gliding or winged aircraft cannot achieve, operating at low altitude to dexterous operation. The operational range of the VTOL craft is typically much shorter than the winged aircraft and also has less lift efficiency than an airplane. The most notable –manned VTOL craft is the helicopter with long blades that generate thrust as they are rotated at different pitch angles. There are ranges of hybrid tilt rotorcraft such as the Osprey, which seek to fill the operational range of a winged aircraft and the dexterity of a helicopter with a transitioning system from vertical take-off to horizontal winged flight. These craft serve particular operation purposes that airplanes cannot achieve. As autonomous unmanned aerial vehicles are the focus of development of the 21st century, the VTOL craft ability of the quadrotors has become a dependable development platform.

2.2 Quadrotors - done

Multirotor crafts have been pursued since the 18th century. The multirotor family contains vehicle platforms such as coaxial blades, tricopters, hexacopters, and many more configurations of multiple rotors or blades. A subset of this multirotor group is the quadrotor. Louis Breguet developed one of the first quadrotor-like platforms in 1907. [30] This large, four-rotor helicopter was ultimately abandoned by the U.S. military due to its technical complexity. The success of quadrotors would ultimately come with the advent of electric motors and integrated chips. The most typical modern definition of quadrotors is explained as a vertical takeoff, vertical landing vehicle that uses four propellers or rotors mounted near the perpendicularity of the plane created by the arms on which they are mounted. Although ever-evolving, these systems typically use electronic power distribution opposed to mechanical drive systems. Each motor is independently controlled by an electronic speed controller, which receives control signals from a central flight controller. Each propeller or rotor combination produces a thrust again perpendicular to the plane of the arms, and a torque. As each motor is controlled independently, combinations of forces and toques allow the quadrotor to dexterous trajectories at low altitudes. With a slower spinning rotor due to increased size of the rotor, helicopters have a better hover efficiency than quadrotors of the same capacity. Quadrotors must have four smaller rotors spinning faster to obtain the same thrust. Helicopters have variable pitch blades and do not experience the greater inertia of the blade that traditional quad-rotors would. [23]

Multirotor research was rarely seen until the 21st century with the resurgence propelled by strides in the micro-electronics industry. The reduced cost of micro-electronics and microchips brought the entry cost of research and development down to allow for less costly ventures in multirotor craft research ad autonomous functionality. Multirotor craft have distinct advantages compared to other rotor dependent VTOL craft. The endurance of a rotor aircraft in the most simplistic terms comes down to rotor length: the slower the revolutions per minute (RPM) of the engine, the greater the endurance of the aircraft. However, as the rotor gets sized up, the inertia is greater. Multirotors allow for large, single-rotor total thrust amongst the separate rotors to be broken up. Having a series of separate rotors gives the craft a similar lift capability of a large rotor but has reduced inertia as well as increased thrust capability. Coaxial setups allow for the same amount of footprint of a non-coaxial vehicle and can produce up to 40 percent more thrust with the second blade beneath.

Quadrotors are particularly popular in research, industry, and personal life due to the lowering cost of hardware and level of control, which as made them viable in the market place. The smaller blades of a quadrotor are safer as the inertia of the blades is less damaging than a helicopter. Due to the reduced cost of micro-controllers and sensors quadrotors are easily purchased off-the-shelf. Research has now been heavily focused on autonomous control and decision-making. However, the scalability of quadrotor platforms are limited mainly due to the concerns of the inertia of rotors for quick responsive flight.

2.2.1 Modeling Quadrotors

The most common method of developing controls for a system starts with modeling the system. This enables representation of the reaction of the system to inputs from the controller. The level of sophistication of the model is dependent on the application of the controller such as research or end product. Often, complex models are highly nonlinear and are simplified for building a controller. Quadrotors are a structure that operate in 3 dimensional space and therefore should be modeled as such. They often include a combination of kinematics and classical mechanics. ***Representations of motion in simulation are ranged from Newtonian, Lagrangian, and Hamiltonian representations.*** The baseline assumption uses rigid body dynamics, with a total of 3 moments, and one force vector controlling the system in 6 degree of freedom environment defines the quadrotor as under-actuated. The coupled rotational and translational motion results in a non-linear system. **??** The quadrotor has to provide its own stabilization force, the only resultant forces acting on it is gravity. This requires the quadcopter to provide its own dampening force.

All aspects of controls have been highly investigated over the past few years from system modeling, feedback control, and state estimation. The University of Pennsylvania Grasp lab has been another large contributor to autonomous multirotors exploration and mapping. [15] provides a great survey of quadcopter control structures and methods. [18] provides a great overview of modeling. ETH Zurich has published a large quantity of work relating to multirotors with demonstrations from cooperative task management to autonomous sensing such as "Collaborative Ariel Robotic Workers" and topics such as search and rescue. BYU provides great overview of quadcopter state estimation modeling with [17] and linearized control method with vision assisted state estimation [1].

There exist many methods of quadcopter control under the branches of linear, nonlinear, robust, and adaptive control. Linear control starts with building a model and linearizing the equations of motion and motor model at a point of equilibrium, which transfer functions and pole placement methods can be used to develop controllers gain constants. The delineation of linear and non-linear methods are represented by modeling linear and non-linear systems. The difference between linear and non-linear systems is described by the superposition principle, if an input A to a system produces response X and input B to a system produces response Y then input (A + B) produces a system response (X + Y). A non-linear system will not produce the same results due to non-linear functions such as sin, cosine, and the exponential as a part of the dynamic model. Non-linear methods, however, model aerodynamic effects, motor and actuator limitations, uncertainty, and dynamics that change with time.

Quadcopter approaches to modeling for controls use non-linear dynamics to simulate reaction of control inputs. Breaking the dynamics into controllable parts is a popular method to visualize as well as implement more control inputs in control methods. Cascading control sequence starts with a position controller that takes in current position and desired position, the output is typically a desired acceleration vector. The attitude controller translates this acceleration vector to desired orientation and compares it to the current orientation of vehicle to find a desired attitude orientation as well as a thrust value. The using the error between the current attitude orientation and desired attitude orientation, the attitude rate controller will output a desired angular acceleration. This angular acceleration can be used to derive torqure from change in momentum, the difference in angular acceleration multiplied by the moment of inertia.

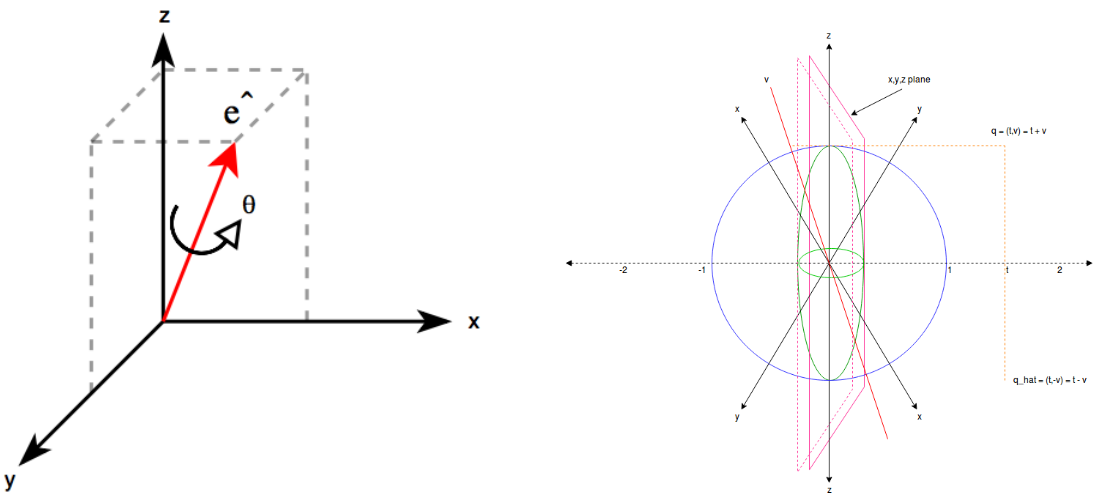
2.3 Quaternions

To derive equations of motion and kinematics requires a system that can describe orientation of a rigid body in three-dimensional space. The most well-known system, Trait-Bryan angles, uses a sequence of rotations about a fixed point of the rigid body to describe change in attitude in respect to a reference frame as either extrinsic or intrinsic. Euler angles in aerospace are notably known as Trait-Bryan angles or roll, pitch, yaw. Euler was able to further realize that two rotations could be described as rotation about a fixed, arbitrary axis. With the invention of matrix algebra, orthogonal rotation matrices could be generated and showed the Euler Angle-Axis rotation vectors where actually the eigenvector of the rotation matrices. These systems are traditionally used in robotics applications to describe the pose of the rigid body itself and/or a manipulator attached to the rigid body. Rotation matrices allow us to construct methods describing the relationship of pose between two objects, for example, the relationship of the manipulator with respect to the robot itself.

These systems are crucial to the development of control laws, which use these descriptive systems to formulate movement over time. Euler rotations face an issue called Gimbal Lock, where a single rotation is broken into three constitutive rotations in which the proceeding rotation occurs about the new frames most recent rotation. This can be visualized when the orientation of an object is pointed straight up or straight down, two rotation axes will be aligned and further rotations will not rotate the shortest path. Changing the rotation order can sometimes fix this problem. Euler angles face other issues associated with gimbals: sometimes due to the rotation order, the rotation path or interpolation does not take the most efficient path. Euler angles use trigonometric functions such as tangent, which carry ambiguity at 90 and 270 degrees of rotation. Quaternions overcome ambiguity, complexity, and computational intensity by encoding a rotation into a single scalar, vector combination. The scalar exists only on the real plane, while the vector can have components that exist on the imaginary number scale.

When quaternions are multiplied, the complex components, i, j, k, will produce new imaginary components based on the order and combination of the complex components. This is the reason for the non-commutative property that exists for quaternions.

$$\begin{aligned} i^2 &= j^2 = k^2 = ijk = -1 && (2.1) \\ ij &= -ji = k && (2.2) \\ jk &= -kj = i && (2.3) \\ ki &= -ik = j && (2.4) \\ & && (2.5) \end{aligned}$$



(a) Euler Rotation Axis (b) Quaternion Concept

Figure 2-1: Axis-Angle Rotation Quaternion concept

One method of visualizing quaternions is to use the rotation axis or Euler rotation theorem which any rotation in the SO(3) can be represented by an arbitrary axis and rotation about that axis, this transformation is shown in equation 2.6. The rotation or change in pose of an object can be described as a unit vector which the object is rotated about quaternions are a vector based system but in the 4 dimensional domain. Quaternions encode a single rotation around a single arbitrary axis opposed to 3 sequence rotations.

$$q = e^{\frac{\theta}{2}(u_x\mathbf{i}+u_y\mathbf{j}+u_z\mathbf{k})} = \cos\frac{\theta}{2} + (u_x\mathbf{i} + u_y\mathbf{j} + u_z\mathbf{k})\sin\frac{\theta}{2} \tag{2.6}$$

The inverse dynamics of Euler angles also carry singularities. Quaternion based kinematics are numerically stable and are free of singularity which occur as 90 and 270 degrees. Quaternion algebraic operations are different from Euler based systems. Instead of the final position being a sum of rotations, the quaternion representation is a product of previous rotations as shown 2.7.

$$P' = Q \otimes P \otimes Q^{-1} \tag{2.7}$$

More formal definitions and derivations can be found in resources [14] as well as

Group theory allows the description of number system properties. Often quaternions are described apart of the SO(3) group which denotes the special orthogonal group. This gives the properties.... Using the description of manifolds, smoothness conditions, and other group conditions, the isomorphism can be shown for unit quaternions from the SU(2) to the SO(3) group. The unit quaternion corresponds of the SU(2) or special unitary group with the properties of 2:1 homomorphism from unit quaternion to the SO(3) group. This states that the unit quaternion covers the sphere of rotation twice over. Since unit quaternion group su(2) double covers the SO(3), unit quaternions are not unique, each $\pm q \in \mathbb{S}^3$ corresponds to the same attitude. [2] This aspect of ambiguity in quaternions must be handled in order to achieve an asymptotically stable yet robust attitude control representation.

2.3.1 Quaternion Control Law

Control law is a mathematical relationship of inputs to outputs. It is standard to have the inputs be a state variable or a function of a state variable. The output is either an input to the system itself or another controller apart of the system. An attitude controller translates desired orientation, current orientation, desired angular velocity, and current angular velocity to desired moments to act on the rigid body. The attitude controller can be represented in many different formats and is sometimes broken into two separate controllers, one for desired attitude orientation and another for desired angular velocity. The effectiveness of a control law is examined by its ability to handle uncertainty and over all robustness to handling environments outside of its designed operating environment.

Quaternion attitude based control has been investigated for its robustness and usefulness in aerospace as well as computer image processing. Aerospace applications have transitioned away from gimbal measurement systems to "strapped down" methods which integrate rotation and acceleration to find orientation and position. Quaternion control methods have shown ability to over come singularities of angle based methods, as well as reduce computational complexity as seen in matrix rotation methods.

Most times, attitude controllers and their proof of stability are built on the assumption of small angle approximations. [20] and [21] has forayed this assumption in seeking to develop robust global attitude tracking control law based on quaternion representation of attitude. This research exacts singularities caused from using memoryless quaternion feedback control laws and proposes solutions in which handles an issue known the unwinding phenomenon. This phenomenon is described in the situation which the quaternion will rotate about the axis of rotation through the larger angle. They use a hybrid control law which incorporates the sign of the quaternion scalar value. This insures that the quaternion error vector is incorporated appropriately to the desired angular velocity term (described in section ...)

A more formally PID quaternion controller approach is presented by [26]. This method seeks to offer the same resolution presented by [20] using filtered command equation which can compute the analytical derivative of the quaternion using the transformation of angular velocity without differentiation. The method enables implementation of back-stepping control in quaternion based feedback control from analytical derivation of the quaternion and relationships to angular velocity.

[5], [10] works with full quaternion attitude control for quadrotors . [32] explores the complexity of second order derivatives of quaternion error for proof of Lyponuav stability using velocity measurements in the derivation of angular velocity.

Robustness of quaternion control law can be tested through an adaptive controller type, which manages a system in changing uncertainty such as changing moments of inertia due to fuel consumption. This control approach is tested by[4] and in a controller with the simple quaternion error tracking which also lack full state velocity terms updates. [9] takes an optimal control approach to quaternion attitude control to find the minimal quaternion rotation using Hamiltonian methods. This method is superseded by methods presented in [19] which can use resulting inertial force and body force vectors to determine the best transition. This process was ultimately adopted in the [5].

Apart of the attraction to variable pitch quadrotors is the instantaneous inverted thrust capability which means that the variable pitch quadrotor is able to exhibit aggressive inverted flight characteristics and aggressive transitions. Switch mechanisms for inverted flight orientation much reverse the resultant trajectory force vector. [5] all have identified switching mechanisms related to the quaternion representation in the scalar value.

2.4 Variable Pitch Quadrotor

Characterized as a quadrotor with varying pitch blades which are controlled by 4 independent swash plates attached to servos. In theory the change in thrust is instantaneous opposed to the traditional quadrotor which must overcome the inertia of the blade. Furthermore the variable pitch system allows for instantaneous reverse thrust for applications such as inverted flight or braking maneuvers.

Variable pitch quadrotors offer many advantages to the multi-rotor craft platform family. Scaling quadrotor platforms can be quite limiting due to the current size of batteries and motors required. Variable pitch platforms allow for a single motor instead 4 independent motors with simplistic mechanical design. This feature combined with the less requirement of instantaneous change in angular velocity can allow for a more moderate heavy fuel based engine with can be refuel more quickly than a strictly electrical system. The flight maneuvers are still be investigated. "However, traditional pod-and-boom autonomous helicopters have demonstrated agile maneuvers outside of the flight regime of small fixed-pitch quadrotors" [7].

Variable Pitch quadrotor systems are an quadrotor configuration studied by a notable few from MIT [6], [7], [5] Nanjing University [26], University of Cape Town [24] and Indian Institute of Technology Kanpur [12]. Their work shows that variable pitch quadrotors rigid body dynamics can be modeled similar to traditional quadrotor but require a different method of modeling the pitch of the blades, typically with blade element theory or experimental correlation.

[5] has been the corner stone of work on small scale variable pitch quadrotors and is cited

by most other variable pitch work post 2012. This work implements a variable pitch quaternion based controller with 4 independent motors as well as 4 independent pitch servos. The control structure models the entire vehicle with actuator modeling and experimental testing. The simulation implements a motor model that was derived and then correlated with experimental test data. The model shows data for a range of RPM speeds and pitch angles. This data allowed derivation of a controller that correlated voltage with thrust. Using a quaternion based controller construction allowed for inverted flight trajectories without the issues familiar with Euler angle singularities.

[26] focuses on energy efficient control of the variable pitch platform, building a controller that determines the best RPM and pitch angle based on environmental and state conditions. Dependent on platform setup, 2 separate actuator controls can affect lift of the variable pitch quadrotor, the RPM of the blades and angle of attack of the blades.

Another work which exemplifies back-stepping control coupled with Blade Element Theory is [12]. This work uses an Euler Angle based attitude controller scheme and requires the derivative of moments to calculate the change of pitch angle. This method allows the implementer to apply it to range of different blade configurations using known blade lift and drag coefficients. This method was verified in simulation using real known platform data of a Reaper 500 sold by Hobby King.

2.4.1 Trajectory

Trajectory control is a limiting factor of quadrotor as most position control methods are limited by the range of stability of the attitude controller which is only stable for small angle approximations. **(UNSURE) Trajectory is limited to the under actuated platform. Over coming this issue new methods for trajectory generation a few researchers attempt to build control structure which can take in more desired states of the quadrotor based on the trajectory generation.

Attempting to achieve coordinate free tracking reduces singularities and removing the need for local coordinate systems is the goal of [16]. The usefulness of nonlinear geometric tracking methods is proved for almost global asymptotic tracking in the case which the rotation is not a hundred and 180 degrees . [22] Minimum snap trajectory generation and control for quadrotors uses this process to extend the ability of aggressive trajectory following. This concept is further extended to trajectory design and control for aggressive formation flight with quadrotors in [28].

With the quaternion representation rotational group double covering the SO(3), rotating a object 360 degrees results in half a quaternion rotation. A full quaternion rotation results in a 720 rotation. While the geometric approach can be applied in a linear and non-linear control structure, a most common control method can be shown to work with aggressive trajectory following with LQR.

[31] Quaternion-Based LQR Spacecraft Control Design Is a Robust Pole Assignment Design.

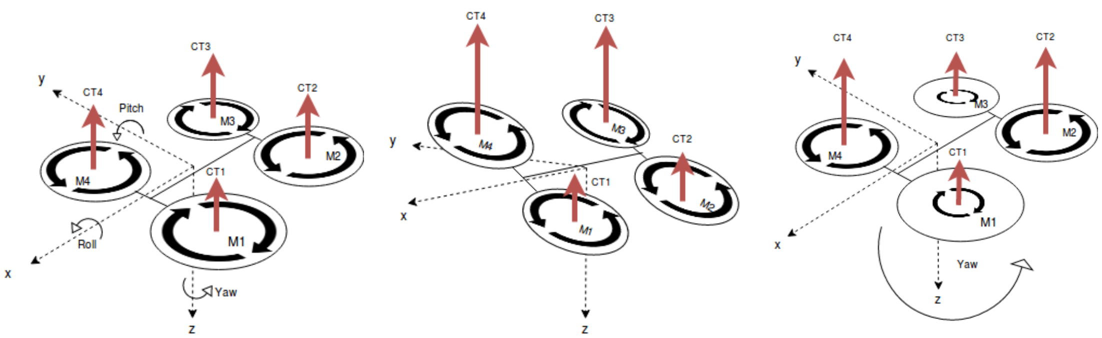
This idea of coupling LQR control with quaternion attitude control is further investigated by [13] for aggressive trajectory tracking with a micro-quadrotor UAV. This work however did not fully include global stability of quaternion control where the sign of the quaternion scalar is included.

[27] Indirect Kalman filter for 3D attitude estimation. This subject matters give nice overview and understanding of quaternion control. This topic describes quaternion rotations.

Chapter 3

Dynamic Model

3.1 Quadrotor Flight Dynamics



(a) Quadrotor Dynamics (b) Roll Moment (c) Yaw Moment

Figure 3-1: Quadrotor Dynamics

Quadrotors use 4 propellers to produce 4 force vectors perpendicular to the blade of the rotor. The force vector can be broken into a thrust vector perpendicular to the x-y plane of the quadrotor and moment vector in-line with the x-y plane. The moment at an rotor is the arm distance from the center of gravity which is assumed to be at the center of quadrotor. At level flight and stable hover, each thrust vector is equal and each moment is each as demonstrated in 3-1a. To induce a roll to port, the thrust vectors CT3 and CT4 are increased while CT1 and CT2 are decreased as shown in 3-1b. It can be noted that the increased moment in rotor 3 is negated by rotor 4, as well as the decreased drag moment 1 with drag moment 2. It can be inferred to create a negative pitch that CT1 and CT4 are decreased while CT2 and CT3 are increased. In 3-1c Adjacent rotors are manipulated to induce yaw, to yaw to port, CT1 and CT3 are decreased while CT2 and CT4 are increased. The resulting forces creates a yawing moment while maintaining enough lift force to hold altitude.

3.2 Quaternions Rotations

Closed loop control for quadrotors includes an array of mathematical subsystems from the position control all the way to the attitude control. Often times these mathematical subsystems contain singularities which can cause failure in the system if not accounted for. Prudent robust design would seek to utilize systems which reduce singularities. The singularities of Euler Based pose tracking have already been mentioned. The following sections, resulting force vectors will be derived to produce resulting a resulting orientation. A vector system is used to describe angular velocity in the body frame. The transition from a force vector to resulting angle based system and then back to vector based system seems unruly. A quaternion based system which uses linear algebraic like processes can be used to reduce the unnecessary conversions.

The quaternion system is defined a combination of scalar and vector, a scalar and three imaginary units. The scalar is represented as:

$$\eta = q_w \tag{3.1}$$

The vector of imaginary units is described in the arrangement of the following vector:

$$\varepsilon = \begin{bmatrix} q_x \\ q_y \\ q_z \end{bmatrix} \tag{3.2}$$

The ordered pair a quaternion representation in 4 dimensional space as follows:

$$Q = \begin{bmatrix} \eta \\ \varepsilon \end{bmatrix} \tag{3.3}$$

Angular rates describe the angular velocity in the body frame, they are represented in the form:

$$\omega_b = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{3.4}$$

Equations **??**, 3.5, 3.6, 3.7, 3.8 use the same representation and derivation shown in [26]. The complex conjugate or adjoint operation that is a matrix transpose in quaternion operations is represented in **??**. The conjugate can be used to extract the scalar and vector parts of the quaternion.

3.5 represents the skew matrix which can be used as to represent the cross products of vectors as matrix multiplication.

$$S(x) = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix} \tag{3.5}$$

The rotation matrix format is useful for transformation operations from inertial frame to body frame and body to inertial. 3.6 is a representation for determining the rotation matrix from the quaternion state.

$$\mathrm{R}(\bar{Q}) = \mathrm{I}_{3\times3} - 2\eta S(\varepsilon) + 2S(\varepsilon)^2 \tag{3.6}$$

The quaternion cross product is separate from the linear algebra cross product operation and holds distributive, associative, but not communicative properties. 3.7 represents quaternion multiplication and essential the sum of two rotations.

$$\bar{Q} \otimes \bar{P} = \begin{bmatrix} \eta_Q \varepsilon_P + \eta_P \varepsilon_Q - \varepsilon_Q \times \varepsilon_P \\ \eta_Q \eta_P - \varepsilon_Q^{\mathrm{T}} \varepsilon_P \end{bmatrix} \tag{3.7}$$

The rotation property of the quaternions is such that the rotation Q and the rotation P when combined is a resulting rotation matrix of the quaternion product of the quaternion Q and P as shown in [**?**]

$$R(\bar{Q})R(\bar{P}) = R(\bar{Q} \otimes \bar{P}) \tag{3.8}$$

The operation to determine the quaternion error or the vector relating the error from the inertial rotation vector to the desired inertial rotation vector is described in the body frame in the following equations:

$$Q_d^i = Q^i \otimes \tilde{Q}^b \tag{3.9}$$

$$\tilde{Q}^b = Q_i^* \otimes Q_d^i \tag{3.10}$$

The rate transformation matrix of the vector omega to the rate quaternion used for solving the ODE in the dynamic equations is represented as a function of the quaternion in **??**.

$$\Phi = \begin{bmatrix} S(\varepsilon) + \eta\mathbf{I} \\ \varepsilon^{\mathrm{T}} \end{bmatrix} \tag{3.11}$$

3.3 Reference Frames - done - needs grammar

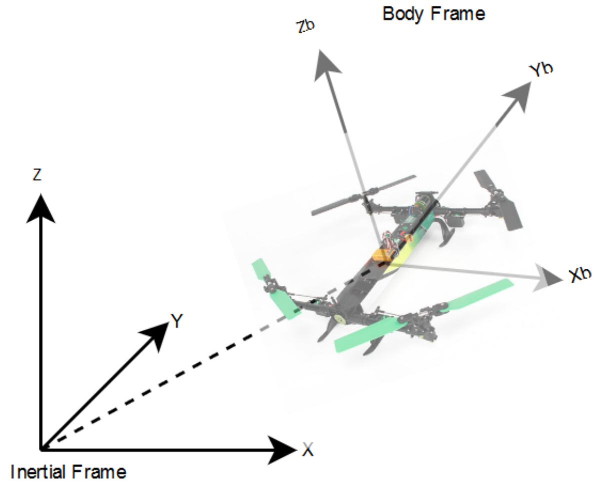


Figure 3-2: Frame System

Reference frames are mentioned in the introduction with respect to orientation. They however play a bigger role when describing position as well. Two common frames are used to describe position and pose are the inertial i, and body frame b. The inertial frame can also be described as the world frame while to body fixed frame describes the displacement, velocity, and acceleration in a local fixed frame to the orientation of the body. The inertial frame is used to describe the relation to the gravity vector which is assumed to remain constant in simulation. The body frame is fixed to the center of gravity of the object. It is an orthogonal coordinate frame and aligned to the principle axises of rotation enables simplification of the equations of motion. The inertial frame stays constant with respect to the world. Accelerometers measure changes in orientation with respect to the body and integrate to pose and position in the body frame. The relationship between the body frame and inertial frame is used to describe changes in pose. Angular rate sensors called rate gyros measure the changes in angular velocity. [29]

Modeling the different states of a quadrotor in these two reference frames greatly simplifies the equations of motion and dynamics. Frame rotation enables a translation between the inertial and body frame axis. Acceleration, velocity, and position of the quadcopter are typically modeled in the world or inertial frames while the angular acceleration, angular velocity, and orientation are in the body frame. The desired force however is defined in the body frame and therefore must be transformed to the inertial frame if used in an inertial frame based trajectory controller.

The quaternion cross product can be used to transform velocity from the body frame to the inertial frame. Using an ODE solver, the following differential equation using the quaternion vector rotation method 3.12.

$$\begin{bmatrix} 0 \\ \dot{v}^i \end{bmatrix} = \bar{Q}^* \otimes \begin{bmatrix} 0 \\ v^b \end{bmatrix} \otimes \bar{Q} \tag{3.12}$$

3.4 Time Derivative between two frames - done - needs grammar

The time derivative between two frames is important in solving rigid body dynamics to described the change in unit vectors as they are rotated over time. This can be described by an example of unit vector $\hat{u}$, either force vector or orientation vector, and omega is a vector describing rotation speed. The change in the pose or change with respect to the frame of vector u is shown in 3.13.

$$\frac{\mathrm{d}}{\mathrm{d}t}\hat{\boldsymbol{u}} = \boldsymbol{\omega} \times \hat{\boldsymbol{u}} \tag{3.13}$$

This approach can be applied a vector function which is described in term of force such as 3.14.

$$f(t) = f_x(t)\hat{\boldsymbol{\imath}} + f_y(t)\hat{\boldsymbol{\jmath}} + f_z(t)\hat{\boldsymbol{k}} \tag{3.14}$$

The product rule returns:

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{f} = \frac{\mathrm{d}f_x}{\mathrm{d}t}\hat{\boldsymbol{\imath}} + \frac{\mathrm{d}\hat{\boldsymbol{\imath}}}{\mathrm{d}t}f_x + \frac{\mathrm{d}f_y}{\mathrm{d}t}\hat{\boldsymbol{\jmath}} + \frac{\mathrm{d}\hat{\boldsymbol{\jmath}}}{\mathrm{d}t}f_y + \frac{\mathrm{d}f_z}{\mathrm{d}t}\hat{\boldsymbol{k}} + \frac{\mathrm{d}\hat{\boldsymbol{k}}}{\mathrm{d}t}f_z \tag{3.15}$$

From the above equation, it can be the derivative of the vector u can be rewritten as the second half of the equation below:

$$= \frac{\mathrm{d}f_x}{\mathrm{d}t}\hat{\boldsymbol{\imath}} + \frac{\mathrm{d}f_y}{\mathrm{d}t}\hat{\boldsymbol{\jmath}} + \frac{\mathrm{d}f_z}{\mathrm{d}t}\hat{\boldsymbol{k}} + [\boldsymbol{\omega}\times(f_x\hat{\boldsymbol{\imath}} + f_y\hat{\boldsymbol{\jmath}} + f_z\hat{\boldsymbol{k}})] \tag{3.16}$$

Simplifying the above equation, the basic kinematic equation can be realized where the derivative of the force with respect to time is understood in the rotation frame.

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{f} = \left(\frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}t}\right)_r + \boldsymbol{\omega} \times \boldsymbol{f}(t) \tag{3.17}$$

This concept,known as the Transport Theorem, can be applied to deriving the orientation. This concept can be applied to the rigid body dynamic which describes the change force with respect to the inertial frame of the quadrotor.

$$\frac{d}{dt}\begin{bmatrix} 0 \\ \bar{F}^i \end{bmatrix} = \frac{d}{dt}(Q^* \otimes \begin{bmatrix} 0 \\ \bar{F}^i \end{bmatrix} \otimes Q) - \begin{bmatrix} 0 \\ \omega_d^b \times \bar{F}^i \end{bmatrix} \tag{3.18}$$

The rotation of the force in the inertial frame can be realized as the normalized force in the body frame, which is shown below to be 1 and therefore the derivative would be 0. The equation simplifies to 3.19.

$$\dot{\bar{F}}^i = \bar{F}^i \times \omega_{dxy}^b \tag{3.19}$$

The rearrangement of 3.19 allows the description of the angular velocity in 3.20.

$$\omega_{dxy}^b = \bar{F}^i \times \dot{\bar{F}}^i \tag{3.20}$$

3.5 Rotational Dynamics

Euler's equations for rigid body dynamics describe the resulting torques acting the body in 3 dimensional space as a function of the angular velocities and angular acceleration using the principal moment of inertia. The moment of inertia is the ratio of angular momentum to the the angular velocity about the its corresponding principle axis. The time derivative of the angular momentum equation is gives us the resultant torque. This is important as quadrotors are modeled as a rigid body in 3 dimensional space with 4 torques and thrust vectors that are applied to the rigid body at some length from the center of mass. The vector form can be expressed in 3.21.

$$\tau = I \cdot \dot{\omega} + \omega \times (\mathbf{I} \cdot \omega) . \tag{3.21}$$

$$\dot{\omega} = I^{-1}(\tau - \omega \times (I\omega)) \tag{3.22}$$

Applying the orthogonal coordinate system to 3.22, the torque about each axis can be redefined to 3.23. The moments of inertia are fixed principle moments of inertia in the body frame and their for are easier to manage.

$$\begin{aligned} \tau_1 &= I_1\dot{\omega}_1 + (I_3 - I_2)\omega_2\omega_3 \\ \tau_2 &= I_2\dot{\omega}_2 + (I_1 - I_3)\omega_3\omega_1 \\ \tau_3 &= I_3\dot{\omega}_3 + (I_2 - I_1)\omega_1\omega_2 \end{aligned} \tag{3.23}$$

3.6 Blade Element Theory

Blade element theory provides a relationship between coefficient of thrust, pitch angle, and inflow ratio while momentum theory offers the relationship between the coefficient of thrust and inflow ratio.[25] Basic Helicopter Aerodynamics. Quadrotors are similar to helicopters such that they both uses propellers to increase air velocity in an imaginary cylinder. Momentum theory based on fluid mechanics theory, law of conversation of momentum, equal and opposite reactions, and in-compressible flow dictate that air accelerated through a column will have an equal and opposite reaction force called thrust. Since modeling propellers is highly nonlinear with much uncertainty in the blade element Theory combined with momentum theory, a simplified approach from [12] is used. Blade Element Theory describes the coefficient of thrust as $Thrust = C_T * K$ where K is described in 3.25. This can be more formally presented in 3.24.

$$\mathrm{C}_{Ti} = \mathrm{T}_i/(\rho\pi\mathrm{R}^2\mathrm{V}_{tip}^2) \tag{3.24}$$

$$K = \rho\pi\mathrm{R}^2\mathrm{V}_{tip}^2 \tag{3.25}$$

$$\mathrm{V}_{tip} = \omega R \tag{3.26}$$

$$\theta_0 = \frac{6C_T}{\sigma C_{l\alpha}} + \frac{3}{2}\sqrt{\frac{C_T}{2}} \tag{3.27}$$

3.27 describes the collective pitch with relationship to the thrust coefficient and blade solidity. The change in thrust from the model is calculated by taking the derivative of the Thrust and Moment model as function of Thrust Coefficient represented as:

$$\begin{bmatrix} T \\ l \\ m \\ n \end{bmatrix} = \begin{bmatrix} K & K & K & K \\ K\,L & -K\,L & -K\,L & K\,L \\ K\,L & K\,L & -K\,L & -K\,L \\ 1.5\,K\,R\,\sqrt{\frac{\mathrm{CT1}}{2}} & -1.5\,K\,R\,\sqrt{\frac{\mathrm{CT2}}{2}} & 1.5\,K\,R\,\sqrt{\frac{\mathrm{CT3}}{2}} & -1.5\,K\,R\,\sqrt{\frac{\mathrm{CT4}}{2}} \end{bmatrix} \begin{bmatrix} \mathrm{CT1} \\ \mathrm{CT2} \\ \mathrm{CT3} \\ \mathrm{CT4} \end{bmatrix} \tag{3.28}$$

3.6.1 Geometric Attitude Tracking

Position controllers typically return a vector describing the difference between the current position and desired position as a function of position error, velocity error, as well as acceleration error. The orientation of quadrotor is denoted as unit quaternion vector Q_i in the inertial frame. The goal is to find a desired rotation from the current attitude to the desired attitude. The desired force vector in the inertial frame described in 3.29, the desired force in the body frame can be calculated using the quaternion rotation of a vector in 3.30.

$$\mathrm{F}_i = m(\ddot{\mathrm{r}}^i_{fb} + \ddot{\mathrm{r}}^i_d + \mathrm{g}^i) \tag{3.29}$$

To insure proper rotation, all the vectors are normalized with the function using unit quaternions. Remember that the inertial unit quaternion describes orientation as rotation. Therefore, a rotation between the F_i and F_b would be described as an intermediate rotation in the body frame between the two vectors. The desired quaternion can be easily solved using 3.9.

$$\begin{bmatrix} 0 \\ \bar{F}^i \end{bmatrix} = (\tilde{Q}^* \otimes \begin{bmatrix} 0 \\ \bar{F}^b \end{bmatrix} \otimes \tilde{Q}_d) \tag{3.30}$$

However, the axis of rotation is arbitrary and not always optimal. A method to overcome this issue is presented by [19] and concludes using a minimal loss function that the minimal quaternion rotation can be a written as a function body frame **b** and reference frame **r** in

$$Q_{min} = \frac{1}{\sqrt{2(1 + \mathrm{b}_3 * \mathrm{r}_3)}} \begin{bmatrix} 1 + \mathrm{b}_3 * \mathrm{r}_3 \\ \mathrm{b}_3 \times \mathrm{r}_3 \end{bmatrix} \tag{3.31}$$

$$Q_{180} = \frac{1}{\sqrt{2(1 + \mathrm{b}_3 * \mathrm{r}_3)}} \begin{bmatrix} 1 + \mathrm{b}_3 * \mathrm{r}_3 \\ 0 \end{bmatrix} \tag{3.32}$$

With the minimal rotation between the Desired force in the inertial frame and current body force in the inertial frame, the desired orientation of the quadrotor can be added to the rotation vector.

$$Q_d = Q_{min} \otimes \begin{bmatrix} \cos(\frac{\psi_d}{2}) & 0 & 0 & \sin(\frac{\psi_d}{2}) \end{bmatrix} \tag{3.33}$$

3.7 Quaternion Control Law

$$Q_e^b = Q_d^i \otimes Q_i^* \tag{3.34}$$

Quaternion error is the product of the sequential rotation from the current pose in the inertial frame to the desired pose in the inertial frame. The resulting error is in the body frame. This error is proportional to the control effort sequence from the vector terms of the quaternion. A simplified controller relating quaternion error and angular velocity can be implemented but as noted in [29] and further investigated in [20], there are issues of the attitude controller seeking the larger angle of rotation or not seeking the angle of rotation that would require the least amount of effort due to the equal representation of the scalar of 1 and -1 when the vector is equivalent to zero. This is called the unwinding phenomenon [21] developed a control law that always considers the scalar element sign to ensure proper rotation.

$$u = -sgn(\tilde{\eta})K_p\tilde{\varepsilon} - K_d w \tag{3.35}$$

Emulating the process in [12], the derivation of $\dot{\omega}_d$ requires solving kinematics equation 3.36 for $\dot{\omega}$ and taking the time derivative. The desired body accelerations are calculated following [8] using the previous quaternion acceleration.

$$\dot{\bar{Q}}_c = \frac{1}{2}\Phi(\bar{Q}_c)\bar{\omega} \tag{3.36}$$

$$\begin{bmatrix} 0 \\ \dot{p}_d \\ \dot{q}_d \\ \dot{r}_d \end{bmatrix} = 2\ddot{Q} \otimes \bar{Q} + 2 \begin{bmatrix} \|\dot{Q}\|^2 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.37}$$

Where form is also transferable to use PID control which is tuned using a linearized model of the state equations, it is represented as:

$$\ddot{Q} = \ddot{Q}_d + K_d * \dot{Q}_{error} + K_p * Q_{error} + K_i * \int Q_{error} dt \tag{3.38}$$

Following the assertion that quaternion error is already in body frame, the derivation of the preceding equation is a transformation from inertial frame to body frame and there for negated. Therefore it can be written as

$$\dot{\omega}_d = -sgn(\tilde{\eta}) * \mathrm{Kp} * (\tilde{\varepsilon}) - \mathrm{Kd} * (\dot{\tilde{\varepsilon}}) - \mathrm{Ki} * \int \tilde{\varepsilon} \tag{3.39}$$

which is similar to equations presented in [26] with the exception of quaternion error derivative instead of $\dot{\omega}$ as $\dot{\omega}_d$ is no longer derived as zero due to it being solved by Force Inertial in 4.14.

3.7.1 Analytical Desired Angular Jerk

The analytical method which was pursued but avoided due to computational complexity is presented here. The following equations are derived from work presented in [26] who has cited equation derivation from [11].

$$\tilde{\bar{Q}} = \bar{Q}_c \otimes (\bar{Q}_0^c)^{-1} \tag{3.40}$$

$$\tilde{\omega} = \omega - \omega_c \tag{3.41}$$

$$\dot{\tilde{\bar{Q}}} = \frac{1}{2} \Phi(\tilde{\bar{Q}}) \tilde{\omega} \tag{3.42}$$

The control law presented in [26] work is as follows. It was sought to be used as the desired angular acceleration control law initially but was later discard for computational complexity in the analytical derivative.

$$\omega_c^o = -K_3 h\tilde{\epsilon} - W^T\tilde{v} + R(\tilde{Q})\bar{\omega} \tag{3.43}$$

The time derivative of equation 3.43, omega acceleration desired can be represented in 3.44.

$$\dot{\omega}_c^o = -K_3 h\dot{\tilde{\epsilon}} - \dot{W}^T\tilde{v} - W^T\dot{\tilde{v}} + \frac{\mathrm{d}R(\tilde{Q})}{\mathrm{d}t}\bar{\omega} + R(\tilde{Q})\dot{\bar{\omega}} \tag{3.44}$$

The derivative of the rotation matrix shown in 3.6 is expanded in 3.45.

$$\frac{\mathrm{d}R(\tilde{Q})}{\mathrm{d}t} = -2\dot{\eta}S(\varepsilon) - 2\eta S(\dot{\varepsilon}) + 2(S(\varepsilon)\times S(\dot{\varepsilon}) + S(\dot{\varepsilon})\times(S(\varepsilon)) \tag{3.45}$$

The rotation of the linear velocity described in the initial control equation 3.44 is expanded to 3.46 and its time derivative shown below it in 3.47.

$$W^T(\bar{Q}_c, \tilde{Q}, F_c) = 2 * \frac{F_c}{m}(\tilde{\eta}I - S(\tilde{\varepsilon}))S(R(\bar{Q}_c)e_3) \tag{3.46}$$

$$\dot{W}^T = 2*\frac{\dot{F}_c}{m}(\tilde{\eta}I - S(\tilde{\varepsilon}))S(R(\bar{Q}_c)e_3) + 2*\frac{F_c}{m}(\dot{\tilde{\eta}}I - S(\dot{\tilde{\varepsilon}}))S(R(\bar{Q}_c)e_3) + 2*\frac{F_c}{m}(\tilde{\eta}I - S(\tilde{\varepsilon}))S(\frac{\mathrm{d}R(\tilde{Q})}{\mathrm{d}t})e_3) \tag{3.47}$$

As shown above, the compilation of 3.47 and 3.46 in 3.44 is algebraically intensive even after attempted simplification processes attempted in MATLAB.

Chapter 4

Theory of Developing Controller

4.1 Analytical Dynamics

The system is described as a dynamical system as the inputs effects are delayed due to the actuation constraints and how inputs are correlated to desired states. In the most simple case, a desired velocity would require an acceleration but the acceleration is limited by the system actuators and therefore the system acceleration is limited, therefore the desired velocity if large is unable to be achieved abruptly.

Taking the time derivative of Euler's second law of angular momentum is equivalent to torque. This can be solved for the change in angular velocity as a function of torque as in equation 4.24.

$$\dot{\omega} = \begin{pmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{pmatrix} = \begin{pmatrix} \frac{q\,r\,(\mathrm{Jy}-\mathrm{Jz})}{\mathrm{Jx}} \\ -\frac{p\,r\,(\mathrm{Jx}-\mathrm{Jz})}{\mathrm{Jy}} \\ \frac{p\,q\,(\mathrm{Jx}-\mathrm{Jy})}{\mathrm{Jz}} \end{pmatrix} + \begin{pmatrix} \frac{l}{\mathrm{Jx}} \\ \frac{m}{\mathrm{Jy}} \\ \frac{n}{\mathrm{Jz}} \end{pmatrix} \tag{4.1}$$

The dynamics used to calculate the change in coefficient of thrust is derived from the derivative of equation 4.24 which has been solved for thrust and torque as demonstrated in 4.3.

$$\dot{T} = \mathrm{Kp}\ (\mathrm{T_d} - T) \tag{4.2}$$

$$
\begin{pmatrix} \dot{l} \\ \dot{m} \\ \dot{n} \end{pmatrix} = \begin{pmatrix} \mathrm{Jx}\, \ddot{p} + (\mathrm{Jx} - \mathrm{Jy})\ (q\, \dot{r} - \dot{q}\, r) \\ \mathrm{Jy}\, \ddot{q} + (\mathrm{Jx} - \mathrm{Jz})\ (p\, \dot{r} - \dot{p}\, r) \\ \mathrm{Jz}\, \ddot{r} - (\mathrm{Jx} - \mathrm{Jy})\ (p\, \dot{q} - \dot{p}\, q) \end{pmatrix} \tag{4.3}
$$

As noted in [12], solving motor model presented would give non-rational solutions and therefore requires the additional loop of solving for the change in desired pitch of the blades. The change in thrust coefficient can be used as a virtual input in the model.

$$
\begin{pmatrix} C\dot{T}1 \\ C\dot{T}2 \\ C\dot{T}3 \\ C\dot{T}4 \end{pmatrix} = Th^{-}1 \begin{pmatrix} \dot{T} \\ \dot{l} \\ \dot{m} \\ \dot{n} \end{pmatrix} \tag{4.4}
$$

4.2 Simulation Model

Following the methods presented by [5], the control method seeks to reduce the necessity for geometric methods such as sine, cosine, tangent, squared as these functions contain singularities that must be managed and therefore cumbersome to the control process. To reduce the complexity of implementation, blade element theory was used as presented in [12] as a form of deriving thrust from the moments. This requires the derivation of angular jerk presented as a control law similar [5] but with added changes influenced by [26] proposed Command filter.

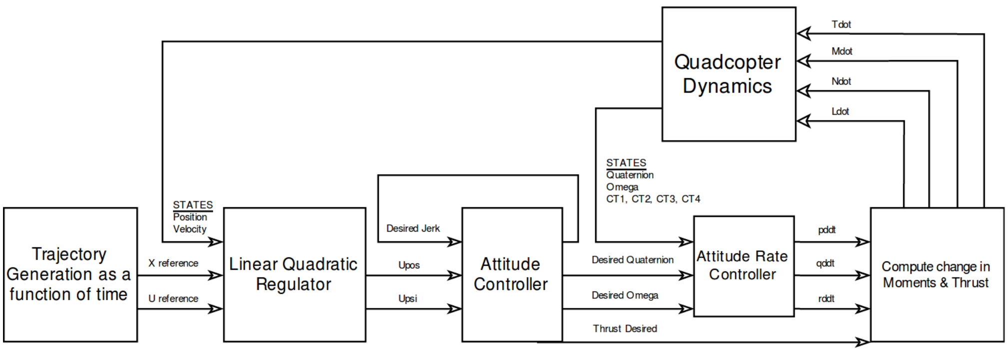


Figure 4-1: Simulation Control Structure

The feedback force is formed from the positional PID loop which outputs feedback acceleration which is inherently a function of desired acceleration, velocity error, and positional error.

$$\ddot{r}_{fb} = \ddot{r}_d + K_d * \dot{r}_{error} + K_p * r_{error} + K_i * \int r_{error} \tag{4.5}$$

From 4.5 a resulting vector is produced which in Euler based methods would covert to Euler angles. Instead, this vector termed as force in the inertial frame commands the trajectory which the quadrotor will follow.

$$\mathrm{F}_i = m(\ddot{\mathrm{r}}^i_{fb} + \ddot{\mathrm{r}}^i_d + \mathrm{g}^i) \tag{4.6}$$

As noted in [5], the Newton-Euler forces can be represented with quaternion operation shown in 4.8.

$$\begin{bmatrix} 0 \\ \ddot{r}^i \end{bmatrix} = \frac{1}{m} Q^* \otimes \begin{bmatrix} 0 \\ F^b \end{bmatrix} \otimes Q - \begin{bmatrix} 0 \\ g^i \end{bmatrix} \tag{4.7}$$

The desired quaternion is described in [19] but observed by Reynolds, we can find the the minimal rotation between to vectors at some arbitrary angle in 4.11 where the vectors are: the normalized inertial force vector 4.8 and the normalized body force vector 4.9. Finding this rotation vector is akin to finding the desired rotation vector or in terms of quaternions, the desired quaternion rotation.

Force bar inertial is the normalized force inertial as shown in 4.8 and the normal of the force in the body frame in 4.9.

$$\bar{\mathrm{F}}^i = \frac{\mathrm{F}^i}{\left\|\mathrm{F}^i\right\|} \tag{4.8}$$

$$\bar{\mathrm{F}}^b = \frac{\mathrm{F}^b}{\left\|(\mathrm{F}^b)\right\|} = \begin{bmatrix} 0 & 0 & \pm 1 \end{bmatrix}^T \tag{4.9}$$

$$\mathrm{F}_b = \begin{bmatrix} 0 & 0 & \mathrm{F}_{total} \end{bmatrix}^T \tag{4.10}$$

$$\tilde{Q}_d = \frac{1}{\sqrt{2(1 + \bar{\text{F}}^{iT}\bar{\text{F}}^{b})}} \begin{bmatrix} 1 + \bar{\text{F}}^{iT}\bar{\text{F}}^{b} \\ \bar{\text{F}}^{i} \times \bar{\text{F}}^{b} \end{bmatrix} \tag{4.11}$$

$$Q_d = \tilde{Q}_d \otimes \begin{bmatrix} \cos(\frac{\psi_d}{2}) & 0 & 0 & \sin(\frac{\psi_d}{2}) \end{bmatrix} \tag{4.12}$$

To find the desired body angular rates, we take the cross product of the normalized force inertial and the derivative of the normalized force inertial as should in 4.14.

$$\frac{d}{dt}\begin{bmatrix} 0 \\ \bar{F}^i \end{bmatrix} = \frac{d}{dt}(Q^* \otimes \begin{bmatrix} 0 \\ \bar{F}^i \end{bmatrix} \otimes Q) - \begin{bmatrix} 0 \\ \omega_d^b \times \bar{F}^i \end{bmatrix} \tag{4.13}$$

$$\omega_{dxy}^b = \bar{\text{F}}^i \times \dot{\bar{\text{F}}}^i \tag{4.14}$$

The derivative the normalized force inertial in derived in 4.15.

$$\dot{\bar{F}}^i = \frac{\dot{F}^i}{||\dot{F}^i||} - \frac{F^i(F^{iT}\dot{F}^i)}{||F^i||^3} \tag{4.15}$$

The derivative of force inertial can be described through the numerical derivative of jerk as shown in 4.16.

$$\dot{F}_i = m * (\dddot{r}_{fb} + \dddot{r}_d) \tag{4.16}$$

This formulation describes pose without desired yaw, desired body yaw is described in 4.17

$$\omega_{dz}^b = \dot{\psi}_d \tag{4.17}$$

As noted in [], quaternion error is described as the quaternion rotation between the desired orientation and the actual orientation. This rotation would in effect describe the rotation in the body frame.

$$\tilde{Q}^b = Q_i^* \otimes Q_d^i \tag{4.18}$$

As the desired angular velocity is derived in equation 4.17, the angular velocity error can be calculated. This is required for calculation of the derivative of quaternion error which uses the angular velocity error in equation 4.21.

$$\tilde{\omega} = \omega_d - \omega \tag{4.19}$$

The time derivative of the quaternion is a skew systematic multiplied by the angular velocity.

$$\dot{Q}^i = \frac{1}{2}\Phi(Q^i)\omega^b \tag{4.20}$$

$$\dot{\tilde{Q}} = \frac{1}{2}\Phi(\tilde{Q})\tilde{\omega}^b \tag{4.21}$$

Equations 4.21 and 4.20 are derived in [26] but are used differently for the purposes of generating a new control law show in equation 4.22.

$$\dot{\omega}_d = -sgn(\tilde{\eta}) * \mathrm{Kp} * (\tilde{\varepsilon}) - \mathrm{Kd} * (\dot{\tilde{\varepsilon}}) - \mathrm{Ki} * \int \tilde{\varepsilon} \tag{4.22}$$

Picking up now from the process presented in [12], the angular jerk is calculated as they are used as inputs in 4.25.

$$\begin{bmatrix} \ddot{p} \\ \ddot{q} \\ \ddot{r} \end{bmatrix} = \begin{bmatrix} \ddot{p}_d \\ \ddot{q}_d \\ \ddot{r}_d \end{bmatrix} + Kd \begin{bmatrix} \dot{p}_d - \dot{p} \\ \dot{q}_d - \dot{q} \\ \dot{r}_d - \dot{r} \end{bmatrix} + Kp \begin{bmatrix} p_d - p \\ q_d - q \\ r_d - r \end{bmatrix} + Ki \int \begin{bmatrix} p_d - p \\ q_d - q \\ r_d - r \end{bmatrix} dt \tag{4.23}$$

Actual angular acceleration is calculated from the current torques acting the rigid body represented in the following equation:

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \frac{q\,r\,(\mathrm{Jy}-\mathrm{Jz})}{\mathrm{Jx}} \\ -\frac{p\,r\,(\mathrm{Jx}-\mathrm{Jz})}{\mathrm{Jy}} \\ \frac{p\,q\,(\mathrm{Jx}-\mathrm{Jy})}{\mathrm{Jz}} \end{bmatrix} + \begin{bmatrix} c\frac{l}{\mathrm{Jx}} \\ \frac{m}{\mathrm{Jy}} \\ \frac{n}{\mathrm{Jz}} \end{bmatrix} \tag{4.24}$$

The dynamics used to calculate the change in coefficient of thrust is derived from the derivative of equation 4.24 which has been solved for thrust and torque as demonstrated in 4.25.

$$\begin{bmatrix} \dot{T} \\ \dot{l} \\ \dot{m} \\ \dot{n} \end{bmatrix} = \begin{bmatrix} \text{Kp } (\text{T}_d - T) \\ \text{Jx } \ddot{p} + (\text{Jx} - \text{Jy}) \ (q\,\dot{r} - \dot{q}\,r) \\ \text{Jy } \ddot{q} + (\text{Jx} - \text{Jz}) \ (p\,\dot{r} - \dot{p}\,r) \\ \text{Jz } \ddot{r} - (\text{Jx} - \text{Jy}) \ (p\,\dot{q} - \dot{p}\,q) \end{bmatrix} \tag{4.25}$$

Using the kinematics described in equation 3.28, we can find the change in coefficient of thrust through the inverse and derivative of it shown in equation 4.26.

$$\begin{bmatrix} c\dot{CT}1 \\ \dot{CT}2 \\ \dot{CT}3 \\ \dot{CT}4 \end{bmatrix} = Th^{-}1 \begin{bmatrix} c\dot{T} \\ \dot{l} \\ \dot{m} \\ \dot{n} \end{bmatrix} \tag{4.26}$$

Using the mathematical models described above and derived control laws, a simulated variable pitch quadrotor can be constructed using ordinary differential equation solvers in programs such as MATLAB.

Chapter 5

Simulation Results

5.1 Control Methods

The attitude controller is actually two separate controllers. The first controller is the attitude rate controller whose output is the angular acceleration is it based on the second order error dynamics equations 5.2 and then later converted to a PID function listed in 5.1.

Second order error dynamic equations were used in the initial design phase of the controller.

$$u(t) = \ddot{x}(t) + 2\zeta w_n \dot{x}(t) + w_n^2 x(t) \tag{5.1}$$

The control method presented in this paper which emulates partials of many other methods follows traditional back stepping methods. Starting with the attitude rate controller which is a constructed PID controller, a linearized form was generated at hover conditions and then used to produce transfer functions which allowed pole placement and tuning of the attitude rate controller. Due to the motor model, the transfer functions are second order.

$$u(t) = K_{\mathrm{p}} e(t) + K_{\mathrm{i}} \int_0^t e(\tau)\, d\tau + K_{\mathrm{d}} \frac{de(t)}{dt} \tag{5.2}$$

After successful tuning of the attitude rate controller, implementation of the attitude orientation controller was completed. Again, using linearized form of equations at hover conditions to generate transfer functions which can be used to generate PID gains for the resulting attitude orientation controller. It should be noted that desired orientation inputs were described in the Euler representation and converted to quaternion representation for use by the controller. A quaternion to Euler angle conversion process was used on the output data as a sanity check for the controller as well. Figure 5-2 shows the results from the tunning process of the attitude pose controller.

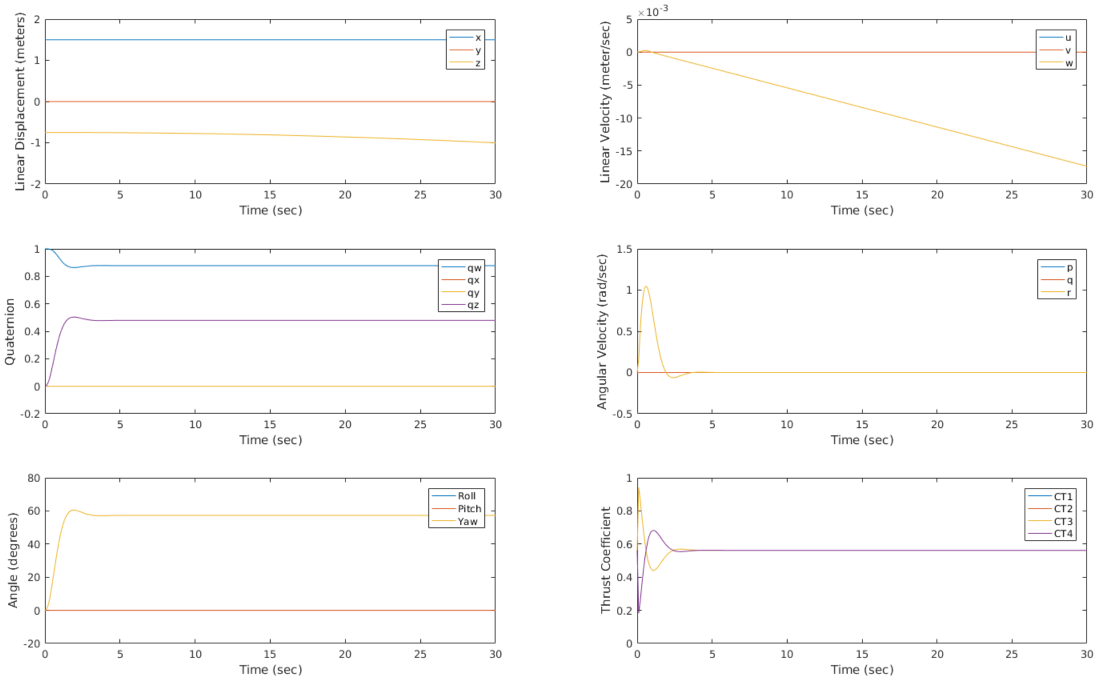


Figure 5-1: Attitude Tuning Results

These results independently verify the control law presented in 4.22 which is explicitly noted as the sole idea of the author inspired by equations derived in [12], [26], and [5]. After verification of the attitude controller, implementation of the position controller derived from [5].

Linear quadratic regulators are used for loop closure of multiple input, multiple output systems. It is useful when it becomes difficult to place Akermann's Formula, K gains at the desired pole locations due to their obscurity. They are useful when systems deviate from 2nd order systems and loose intuition of controls. LQR controls allow engineers to evaluate the effect of many inputs such as position error, velocity error, acceleration error, or combinations of these.

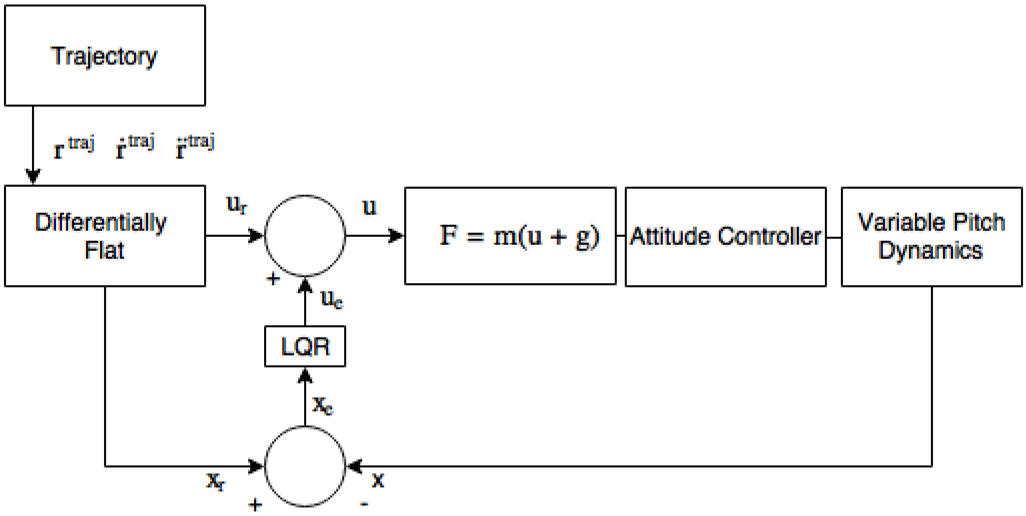


Figure 5-2: Attitude Tuning Results

To determine effective pole placement, a cost function is used where state cost x and control cost u, state cost weight Q, and control cost weight. Using a cost function J, we want to integrate our state vector which is a signal, multiplied with with Q. We also multiply our inputs with matrix R. With the task of minimizing J, Q and R matrices are chosen to minimize the state and input vectors. We the Q and R matrices to prioritize states and combinations of states of positions, velocities, or combinations of positions and velocities. Q and R must be insured to be positive definite will return appropriate values and settle at zero unlike negitive matrices or semi-positive definite matrices which will cause J to negative infinity.

$$J = \int_0^\infty \left(x^T Q x + u^T R u + 2x^T N u \right) dt \tag{5.3}$$

where inputs u vector is represented as:

$$u(t) = -Kx(t) \tag{5.4}$$

$$\dot{\vec{x}} = (A - Bk)\vec{x} \tag{5.5}$$

the feedback control law that minimizes the value of the cost is

where K is given by

$$K = R^{-1}(B^T P + N^T) \tag{5.6}$$

and $\hat{P}$ is found by solving algebraic Riccati equation:

$$A^T P + PA - (PB + N)R^{-1}(B^T P + N^T) + Q = 0 \tag{5.7}$$

Starting with the linear state equation represented as in

$$\dot{x} = Ax + Bu \tag{5.8}$$

Since the LQR only is concerned as a position controller, the states can be reduced to:

$$x = \begin{bmatrix} p_n & p_e & p_d & \dot{p}_n & \dot{p}_e & \dot{p}_d & \psi \end{bmatrix} \tag{5.9}$$

The state equations can be linearized to the following forms:

$$A = \begin{bmatrix} 0_{3\times3} & I_{3\times3} & 0_{3\times1} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times1} \\ 0_{1\times3} & 0_{1\times3} & 0_{1\times1} \end{bmatrix}' \tag{5.10}$$

$$B = \begin{bmatrix} 0_{3\times3} & 0_{3\times3} \\ I_{3\times3} & 0_{3\times3} \\ 0_{1\times3} & 0_{1\times3} \end{bmatrix} \tag{5.11}$$

$$\mathrm{b} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}' \tag{5.12}$$

This equation can be solved for K-gain matrix which is used in the Linear quadratic regulator position controller. This K-gain matrix is used to multiply the current states to generates the inputs.

$$u_r = \begin{bmatrix} u_p^r \\ u_\psi^r \end{bmatrix} \tag{5.13}$$

$$\begin{bmatrix} \ddot{p}_n^r \\ \ddot{p}_e^r \\ \ddot{p}_d^r \end{bmatrix} = u_p^r + \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \tag{5.14}$$

$$u_\psi = \dot{\psi}^r \tag{5.15}$$

$$\tilde{u} = -K\tilde{x} \tag{5.16}$$

$$\mathrm{F}_i = m * ((u_r + \tilde{u})_{1:3} + \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix}) \tag{5.17}$$

Once the attitude controller design was finalized, PID controllers were pursued from the approach of tuning from pole place from transfer functions generated from the linearitazion process of the attitude controller at hover conditions.

5.1.1 Differential Trajectory LQR Results

The controller was tested in simulation following two different differentially flat trajectories, the circle and figure 8. The LQR tuning process did not proceed further than implementing Bryson's Rule as a starting point and minor changes to the Q and R matrices. The two separate trajectories with minimal tuning results were astonishing in simulation. 5-3 is a circle trajectory that varies on all three axis.

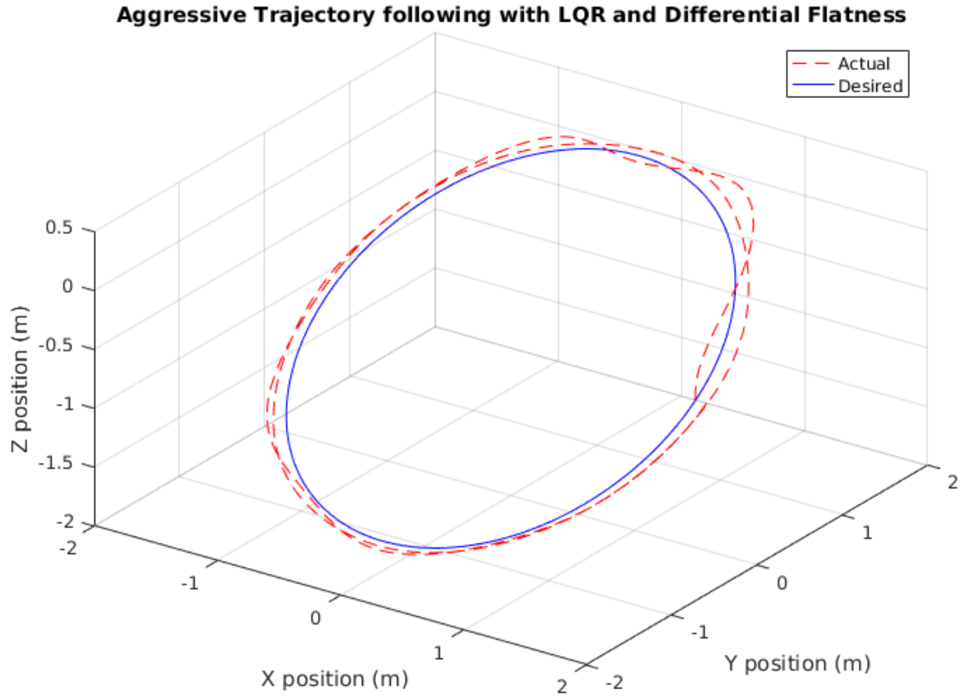


Figure 5-3: Circle Trajectory

The control effort is concerned for any controller, in 5-3 the control effort in monitored in the change of Coefficient of Thrust shown in **??** stay in the managable range of available thrust of the blades in the given environment which is modelled at STP where the air density is assumed 1.225 kg/m3. Better LQR tuning may of resulted in a more stable roll command as can be seen to gradually smooth out over time in the Angular Velocity graph of 5-4.

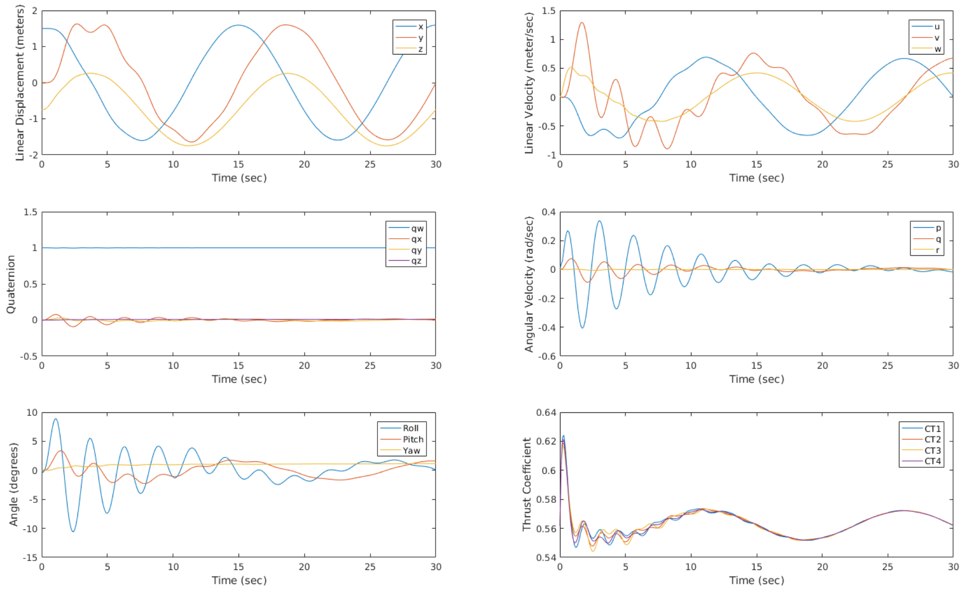


Figure 5-4: Circle Trajectory States

The figure eight trajectory shown in **??** examines the further tracking ability of the controller in simulation. Again with minimal tuning using Bryson's Rule, the resulting trajectory confirms the established qualities of the variable pitch quadcopter in simulation.

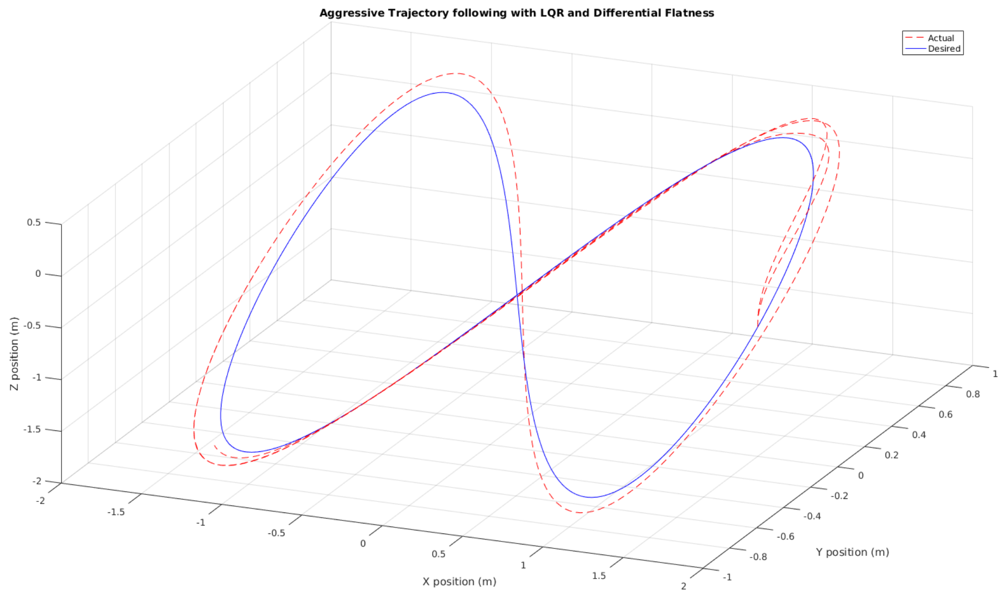


Figure 5-5: Figure 8 Trajectory

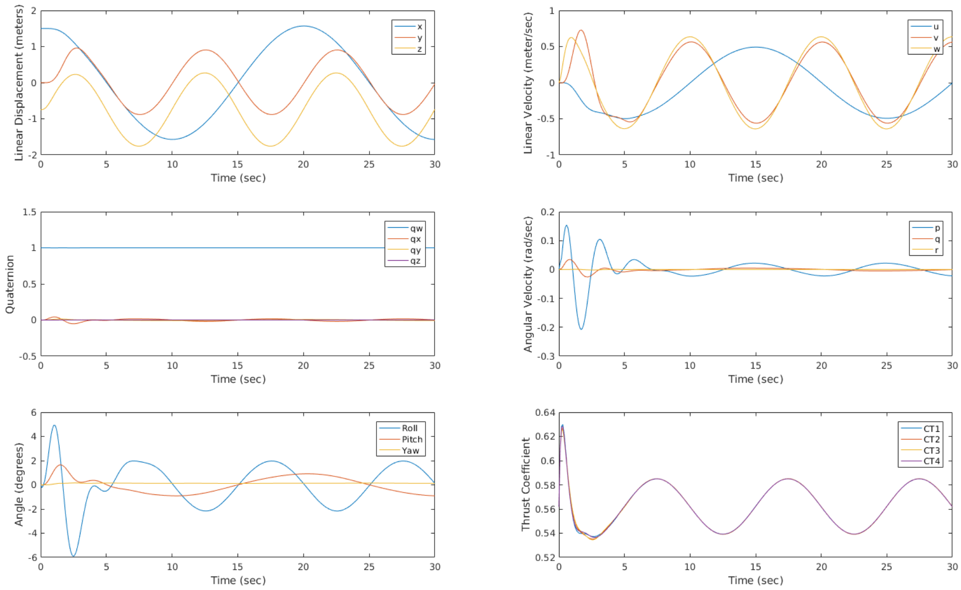


Figure 5-6: Figure 8 States

1.Tuning

2.Results v.LQR with differential Flatness vi.Non-linear Control Implementation

Chapter 6

Experimentation

Call this chapter as: Experimentation, Results and Discussion.

Put all the information about ROS, Pixhawk, mavlink and qgroundcontrol. Explain in detail by means of block diagrams about the communication between ROS, Mavlink, QGroundControl and Mavlink. Write about Rosnodes, rostopics in pixhawk architecture and how commands are given to UAV. It should be self explanatory and detailed so that others can learn from your thesis. Write from flight testing perspective by defining your test cases for flight and then show how the quadrotor performed during flight. Put time history plots of various flight parameters and explain flight performance.

The commercially available Reaper 500 variable pitch quadrotor is used as a test bed for the constructed autopilot. This platform comes with a RC based controller but no autopilot structure. It makes a useful platform as it supports the plug and play ability of a Pixhawk autopilot. This platform can be easily modeled in simulation as all the parts are standard. This platform can be easy to replicate. The firmware structure used is available in open-source community as well as the ground station software. All of the software platforms are interloped using an open source communication protocol called Mavlink.

Figure 6-1: Reaper 500 Platform

The robotic operating system provides an infrastructure useful to developing controllers for small scale robotics. The support packages that can integrate into the infrastructure make it a robust development platform. It carries many integration packages to existing flight controllers available. The Pixhawk flight controller is a roust hardware platform that carries all the sensors required for achieving inertia guided autonomous flight. The Pixhawk platform also supports a range of open source flight controller firmwares. This is useful to developing a controller using already developed controller substructures such as low end controllers for managing servos, motors, and other peripherals. The PX4 firmware not to be confused with Ardupilot supports a broad base of development objectives and was chosen due to its wide support community and development environment. PX4 has previously been integrated into a simulation environment called Gazebo. This simulation environment can be used to test position controller scripts that either send commands via MAVROS to the Pixhawk PX4 firmware or directly communicate with PX4 firmware. MAVROS is a ROS package that supports the communication between the PX4 firmware and ROS environment. This allows for quick paced development of flight controllers as most of all the Pixhawk sensor data can be subscribed to and much of the actuator topics, high level, and low level commands can be published to.

The PX4 firmware already implements a quaternion based attitude controller, way point trajectory controller, and attitude rate controller. The firmware also supports a range of control commands such as takeoff, land, and follow a mission. Using this controller architecture, a linear quadratic regulator position controller was written in Python that follows a deferentially flat trajectory. This outputs of the of the LQR are commanded quaternion pose, desired angular velocity, desired angular acceleration, desired velocity, and desired acceleration. All of these can be feed a command topics to the PX4 firmware attitude and throttle controllers. The implementation of the quaternion based LQR system is crucial to the development of the variable pitch quadrotor and does not use Euler angles or include any trigonometric based functions so to limit the occurrence of singularities.

The PX4 attitude controller is tuned experimentally in a flight test cage using a telemetry based communication system between a ground station and the Pixhawk. The ground control station runs QGroundcontrol which can be used to tune the PID attitude controller on the PX4 firmware. The throttle is hard coded to 80% of the full throttle of the motor. When the vehicle is armed, the throttle will spin up to 80% and does not change until the vehicle is disarmed. During the tuning process all the Integral and Derivative gains are reduced to zero. Proportional gain tuning is performed in small 10% increments until takeoff and hover in ground effect is achieved. The attitude is slowly tuned to a stabilized control in this ground effect hover, slowly tuning the roll and pitch proportional gains. After a mostly stabilized control is achieved, the quadrotor is flown outside of the ground effect hover to a hover about a meter and a half from the ground. At this height the the integral and derivative gains are slowly tuned to garner appropriate response of the stabilization of the attitude controller. The PX4 records the flight logs during each of its flights, this data can be used to analyze the response of the flight controller. This data will show the inputs of the raw IMU (gyro and accolerometer) data, the filtered IMU data, the controller response, and actuator outputs. All of this data can be used to validate the stability of the tuned controller.

Chapter 7

Conclusion and Future work

try to summarize all the thesis in 2-3 pages and once you finish other parts of your thesis, you will be comfortable writing this one automatically.

7.1 Conclusions

The investigation into variable pitch quadrotor controls shows promising control structure for aggressive flight maneuvers that can not be achieved by traditional variable RPM quadrotors. The simulation model proves that second order quaternion error dynamics prove useful to deriving a control structure that uses angular acceleration as an input.

7.2 Future Works

In accordance with the objectives of achieving aggressive and inverted flight maneuvers, singularity free attitude controller must be proven asymptotically stable in the global SO(3) realm. This would be accomplished by proving the Lyponauv Stability analysis of the second order quaternion control law. Much previous work has shown proof of global asymptotic stability with the use of managing the sign of the quaternion scalar. These appropriate to proving stability of the above presented control law.

Further experimentation should be sought in order to show the differential flatness trajectories being processed by the Linear Quadratic Regulator. These test flights would prove the effective integration of the LQR controller as shown in the implementation shown in this paper.

Experimentation could also verify the tuning requirements used in this papers implementation.

The last objective sought by this paper was to verify experimentally the ability of the variable pitch quadrotor to fly way points. This experiment would prove the effectiveness of the contrived control method and the usefulness of the quadrotor simulation model. The model could be further used in an adaptive control structure as well.

Appendix

Publications

Intellectual Property

References

[1] R. Beard. Quadrotor dynamics and control rev 0.1. 2008.

[2] D. Brescianini, M. Hehn, and R. D'Andrea. Nonlinear quadrocopter attitude control. 2013.

[3] J. R. Cares and J. Q. Dickmann Jr. *Operations Research for Unmanned Systems*. John Wiley & Sons, 2016.

[4] B. Costic, D. Dawson, M. De Queiroz, and V. Kapila. Quaternion-based adaptive attitude tracking controller without velocity measurements. *Journal of Guidance, Control, and Dynamics*, 24(6):1214–1222, 2001.

[5] M. Cutler. Design and control of an autonomous variable-pitch quadrotor helicopter. 2012.

[6] M. Cutler and J. How. Actuator constrained trajectory generation and control for variable-pitch quadrotors. In *AIAA Guidance, Navigation, and Control Conference*, page 4777, 2012.

[7] M. Cutler, N.-K. Ure, B. Michini, and J. How. Comparison of fixed and variable pitch actuators for agile quadrotors. In *AIAA Guidance, Navigation, and Control Conference*, page 6406, 2011.

[8] J. Diebel. Representing attitude: Euler angles, unit quaternions, and rotation vectors. *Matrix*, 58(15-16):1–35, 2006.

[9] K. Djamel, M. Abdellah, and A. Benallegue. Attitude optimal backstepping controller based quaternion for a uav. *Mathematical Problems in Engineering*, 2016, 2016.

[10] E. Fresk and G. Nikolakopoulos. Full quaternion based attitude control for a quadrotor. In *Control Conference (ECC), 2013 European*, pages 3864–3869. IEEE, 2013.

[11] C. V. Girish, F. Emilio, H. P. Jonathan, and L. Hugh. Nonlinear flight control techniques for unmanned aerial vehicles. In *Handbook of Unmanned Aerial Vehicles*, pages 577–612. Springer, 2015.

[12] N. Gupta, M. Kothari, et al. Flight dynamics and nonlinear control design for variable-pitch quadrotors. In *American Control Conference (ACC), 2016*, pages 3150–3155. IEEE, 2016.

[13] A. Kehlenbeck. *Quaternion-based control for aggressive trajectory tracking with a micro-quadrotor UAV*. PhD thesis, 2014.

[14] J. B. Kuipers. *Quaternions and rotation sequences: a primer with applications to orbits, aerospace, and virtual reality*. Princeton University Press, Princeton, N.J, 1999. ISBN 9780691058726;0691058725;.

[15] V. Kumar and N. Michael. Opportunities and challenges with autonomous micro aerial vehicles. *The International Journal of Robotics Research*, 31(11):1279–1291, 2012.

[16] T. Lee, M. Leoky, and N. H. McClamroch. Geometric tracking control of a quadrotor uav on se (3). In *Decision and Control (CDC), 2010 49th IEEE Conference on*, pages 5420–5425. IEEE, 2010.

[17] R. C. Leishman, J. C. Macdonald, R. W. Beard, and T. W. McLain. Quadrotors and accelerometers: State estimation with an improved dynamic model. *IEEE Control Systems*, 34(1):28–41, 2014.

[18] R. Mahony, V. Kumar, and P. Corke. Multirotor aerial vehicles. *IEEE Robotics and Automation magazine*, 20(32), 2012.

[19] F. L. Markley. Fast quaternion attitude estimation from two vector measurements. *Journal of Guidance, Control, and Dynamics*, 25(2):411–414, 2002.

[20] C. G. Mayhew, R. G. Sanfelice, and A. R. Teel. Robust global asymptotic attitude stabilization of a rigid body by quaternion-based hybrid feedback. In *Decision and Control,*

2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on, pages 2522–2527. IEEE, 2009.

[21] C. G. Mayhew, R. G. Sanfelice, and A. R. Teel. Quaternion-based hybrid control for robust global attitude tracking. *IEEE Transactions on Automatic Control*, 56(11):2555–2566, 2011.

[22] D. Mellinger and V. Kumar. Minimum snap trajectory generation and control for quadrotors. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 2520–2525. IEEE, 2011.

[23] P. Pounds, R. Mahony, and P. Corke. Modelling and control of a quad-rotor robot. In *Proceedings Australasian Conference on Robotics and Automation 2006*. Australian Robotics and Automation Association Inc., 2006.

[24] A. Pretorius and E. Boje. Design and modelling of a quadrotor helicopter with variable pitch rotors for aggressive manoeuvres. *IFAC Proceedings Volumes*, 47(3):12208–12213, 2014.

[25] J. M. Seddon and S. Newman. *Basic Helicopter Aerodynamics, 3rd Edition*. John Wiley & Sons, 2011. ISBN 9780470665015;0470665017;.

[26] S. Sheng and C. Sun. Control and optimization of a variable-pitch quadrotor with minimum power consumption. *Energies*, 9(4):232, 2016.

[27] N. Trawny and S. I. Roumeliotis. Indirect kalman filter for 3d attitude estimation. *University of Minnesota, Dept. of Comp. Sci. & Eng., Tech. Rep*, 2:2005, 2005.

[28] M. Turpin, N. Michael, and V. Kumar. Trajectory design and control for aggressive formation flight with quadrotors. *Autonomous Robots*, 33(1-2):143–156, 2012.

[29] K. P. Valavanis and G. J. Vachtsevanos. *Handbook of unmanned aerial vehicles*. 2015. ISBN 9048197066;9789048197064;2014944662;9789048197071;9048197074;.

[30] C. Venkatesan. *Fundamentals of helicopter dynamics*. CRC Press, 2014.

[31] Y. Yang. Quaternion-based lqr spacecraft control design is a robust pole assignment design. *Journal of Aerospace Engineering*, 27(1):168–176, 2012.

[32] S. Zhao, W. Dong, and J. A. Farrell. Quaternion-based trajectory tracking control of vtol-uavs using command filtered backstepping. In *American Control Conference (ACC), 2013*, pages 1018–1023. IEEE, 2013.