**Quad-Motor Unmanned Aerial Vehicle (QMUAV) Individual Project**

Contents

[1 Introduction 3](#_Toc54277863)

[1.1 Aims and Objectives 3](#_Toc54277864)

[1.2 Quad-Motor Fundamentals 3](#_Toc54277865)

[2 Avionics and Software 4](#_Toc54277866)

[2.1 Microcontroller Hardware and Software 4](#_Toc54277867)

[2.2 Sensors 5](#_Toc54277868)

[2.2.1 Attitude 5](#_Toc54277869)

[2.2.2 Position 6](#_Toc54277870)

[2.3 Controller Logic 7](#_Toc54277871)

[2.3.1 Overview 7](#_Toc54277872)

[2.3.2 Position Controller 8](#_Toc54277873)

[2.3.3 Attitude Controller 8](#_Toc54277874)

[2.3.4 Motor Mixing Algorithm 8](#_Toc54277875)

[2.3.5 Flight Stage Controller 8](#_Toc54277876)

[2.3.6 Abort Controller 8](#_Toc54277877)

[2.4 PCB Design 8](#_Toc54277878)

[3 Simulation and Modelling 8](#_Toc54277879)

[3.1 Physics Simulation 9](#_Toc54277880)

[3.1.1 Rotational Motion 9](#_Toc54277881)

[3.1.2 Linear Motion 11](#_Toc54277882)

[4 References 12](#_Toc54277883)

# Introduction

## Aims and Objectives

The Main aim of the QMUAV project is to build an autonomously hovering Quadcopter using ‘off the self-consumer hardware’ and a uniquely designed Flight Control System (FCS). The purpose of this being to develop and learn skills in the GNC or Guidance Navigation and Control sector of Aerospace engineering. The project also allows for the demonstration of key engineering skills including the use of technical software (MATLAB – SIMULINK/SOLIDWORKS CAD/Autodesk EAGLE CAD), software engineering in a predominately C++ based language and using Current engineering knowledge to develop a system and make valid design choices.

## Quad-Motor Fundamentals

A quadcopter utilises four fixed propeller motors displayed in a cross formation. Each motor produces a force commonly referred to as thrust, which during level flight opposes the gravity vector. The speed of each motor can be varied to unbalance the forces on the vehicle and enable control. By doing this the quadcopter can be rotated in 3 degrees of freedom. The first is yaw which can be described as a rotation around the z-axis. Roll and Pitch can be described as rotation around the x and y axis respectively. The axes are determined by the body reference frame shown in figure 1.

The quadcopter is an under actuated system since it has 6 degrees of movement freedom (3 in translation and 3 in rotation), however it only has 3 degrees of control freedom (3 in rotation). However, since the motors are fixed to the vehicle frame as the quad-motor rotates in either pitch or roll the thrust vector is no longer only in the z-axis and instead has a horizontal component. Therefore, even though the system is under actuated the vehicle can be translated in x and y by pitching or rolling the vehicle. Translation in z is the simplest knowing that the thrust force must be greater than the force of gravity acting on the vehicle for an increase in altitude and the opposite for a decrease.

Thrust

W=mg

Thrust

Z

x

y

Figure 1: Diagrams showing the body reference frame and the vertical Forces

The yaw or rotation around the z-axis is controlled by the summation of the torques produced by the motors. In order for the system to be at equilibrium when all 4 motors are spinning at the same speed 2 of the motors must be rotating clockwise while the other two rotate counter-clockwise. Therefore, the torques cancel each other out.

Since half of the motors spin opposite to the other half it is now required to determine the layout of the motors. The obvious choice for this is to have the CW motors diagonally opposite each other and the same with the CCW motors as shown in figure 2. This layout allows for the yaw to be controlled independently of roll and pitch as well as allowing roll and pitch to be controlled independently of yaw. This concept will be further explained later on.

x

y

CCW

CCW

CW

CW

Figure 2: Diagrams showing the CW and CCW motor layout

# Avionics and Software

## Microcontroller Hardware and Software

The Arduino range of microcontrollers is selected for use on-board the QMUAV due to its low-cost accessibility and its wide range of public documentation. The Arduino also sports a wide range of easily compatible sensors and modules in the form of breakout boards. Specifically, the Arduino UNO was used as a prototyping board while the onboard avionics package contains a Teensy 3.2. While the UNO utilises the ATmega328P microcontroller the teensy is equipped with a much more powerful MK20DX256 Cortex-M4 micro-processor.

Arduino also provides an Integrated Development Environment (IDE) for writing and compiling software to the microcontroller. The Arduino IDE is based on the C++ programming language but also supports Arduino specific structures.

The Teensy 3.2 is a 32-bit processor which offers high clock rates which makes it ideal for controlling the QMUAV. The large on-board program memory (64 Kbytes) allows for an extensive flight control program and the use of multiple sensor libraries.

## Sensors

### Attitude

Attitude is measured using a Bosch BNO055 9DOF MEMs Inertial Measurement Unit integrated into an ADAFRUIT breakout board. The BNO055 is Comprised of an accelerometer, gyroscope and magnetometer. While the accelerometer measures linear acceleration in xyz, the magnetometer provides a 3D vector which directs to magnetic north. The gyroscope measures angular velocity about xyz therefore can measure yaw, pitch and roll:

The Gyroscope is susceptible to drift over time. Therefore, measuring attitude with the gyroscope alone is impossible as over time the error would increase until the measurement is no longer usable.

Fortunately pitch and roll can also be measured using an accelerometer. This can be done by measuring the acceleration due to gravity in xyz. If the accelerometer is perfectly horizontal then it can be expected that the and . Therefore, pitch and roll can be described as.

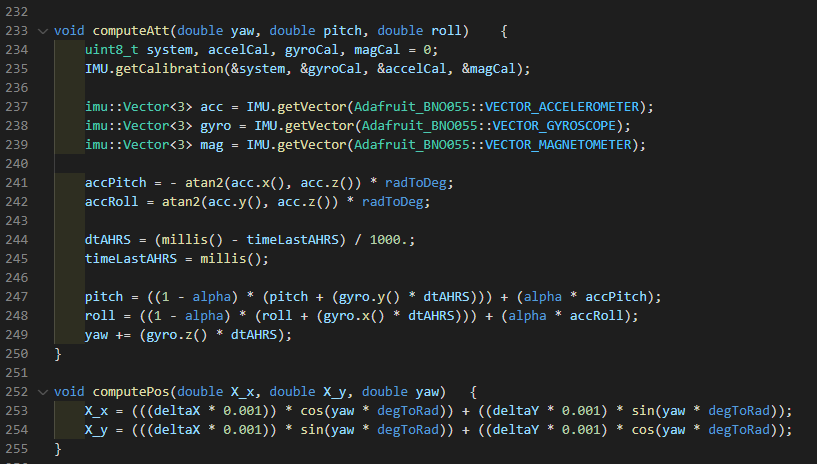
Where pitch and roll are given in radians.

Accelerometers are very noisy sensors, which significantly reduces their accuracy. By fusing the accelerometer and gyroscope data the drift and noise can be eliminated. To do this a complimentary filter is used. A complimentary filter is a simplified combination of a low pass filter (to remove noise from accelerometer) and a high pass filter (to remove drift from gyroscope).

Alpha is a value used to weight the complimentary filter, where increasing alpha reduces responsiveness of the sensor but increases noise. shows the integration of the rotational velocity measured by the gyroscope. dt indicates the time since the last calculation.

Figure 3 shows the attitude computation function which includes the pitch and roll complimentary filters implemented into the flight software. Lines 241 and 242 compute pitch and roll from the acceleration data and convert to degrees. The time between computations is managed by lines 244 and 245. Lines 247 and 248 calculate the final roll and pitch using the complimentary filter with an alpha value of 0.1.

Figure 3: Software computation of pitch, roll and yaw



Yaw cannot be calculated using accelerometer data as change in yaw has no effect on the gravity vector. Therefore, yaw is calculated on line 249 using just the integrated gyroscope data. This means that the yaw value is susceptible to drift. While this is undesirable, due to the flight characteristics the QMUAV is designed for, change in yaw and flight times are short. This means that drift is negligible.

### Position

The noise associated with low cost micro electro-mechanical IMU’s means that dead reckoning is not achievable (without sensor fusion). After integrating twice to find displacement, the noise produces an error that after a matter of seconds is out of control.

One solution is fusing the accelerometer data with a GPS. Although this has not been implemented in the current version of the QMUAV, it is the next logical step to upgrade it and as a result the following paragraph will offer a brief description of how this could be possible.

GPS and GNSS satellites can be used to stream position data (latitude and longitude) to a receiver which could be positioned on the drone. The GPS and accelerometer data can be fused using an extended Karman filter. This system dynamically weights each data stream to close in on the true value and decrease the error. The acceleration data for x and y will need to be integrated twice to calculate the displacement from the initial point. This position data is still in the sensor or body reference frame. In order to fuse it with the GPS data it will have to be converted to the inertial or NED (North, East, Down) reference frame. Yaw data is no longer enough, instead the vehicles heading must be calculated. This can be done by fusing the magnetometer data with the gyroscope.

Now the change of reference frame is possible.

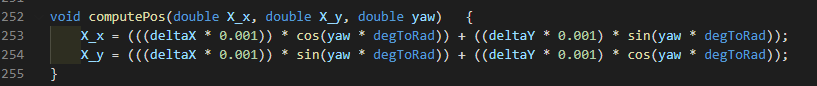
By comparing the incoming latitude and longitude from the GPS to the previous position, displacement can be calculated and summed up from the GPS. Both the accelerometer and GPS data are ready to be fused together.

In order to measure position in x and y the QMUAV will utilise the PMW3901 Optical Flow sensor. The PMW3901 uses a high-speed camera to take pictures of the ground below the vehicle. Using the Bitcraze PMW3901 Library the sensor can identify landmarks and estimate the distance the landmarks have moved between each picture. This allows an accurate measurement of displacement up to a maximum altitude of 1 meter. The displacement from each frame can be summed up to calculate the position from an initial point. This position data has to be converted from the body to the ground reference frame. The ground reference frame is fixed and is characterised by the body reference frame before take-off. As a result, conversion to the ground reference frame is simple so long as the yaw before take-off is also zero.

Where is position from an initial point in the ground reference frame, x is position in the body reference frame and is yaw. By converting to the ground reference frame, the vehicle position will always be valid regardless of any change in yaw.

Figure 4 shows the real time conversion to the ground reference frame. Note the yaw values must be converted back into radians for the trigonometric functions.

Figure 4: Real time conversion from body to ground reference frame



Position in the z axis can be referred to as altitude. This is measured using a VL53L0X LIDAR Time of Flight sensor. Light detection and ranging sensors measure the properties of scattered and reflected light to measure range to a surface [1].

## Controller Logic

### Overview

The flight control system is made up of two main parts; the position controller and the attitude controller. These two components are encased in a closed loop system with the onboard position and attitude sensors.

Position controller

Attitude Controller

Motors

Position/Attitude Sensors

Figure 5: Basic overview of flight controller logic

### Position Controller

### 

The position sensor data converted to the ground reference frame is fed into the position controller as well as flight parameter set points. The controller is made up of three Proportional, integral derivative controllers (PID).

### Attitude Controller

### Motor Mixing Algorithm

### Flight Stage Controller

### Abort Controller

## PCB Design

# Simulation and Modelling

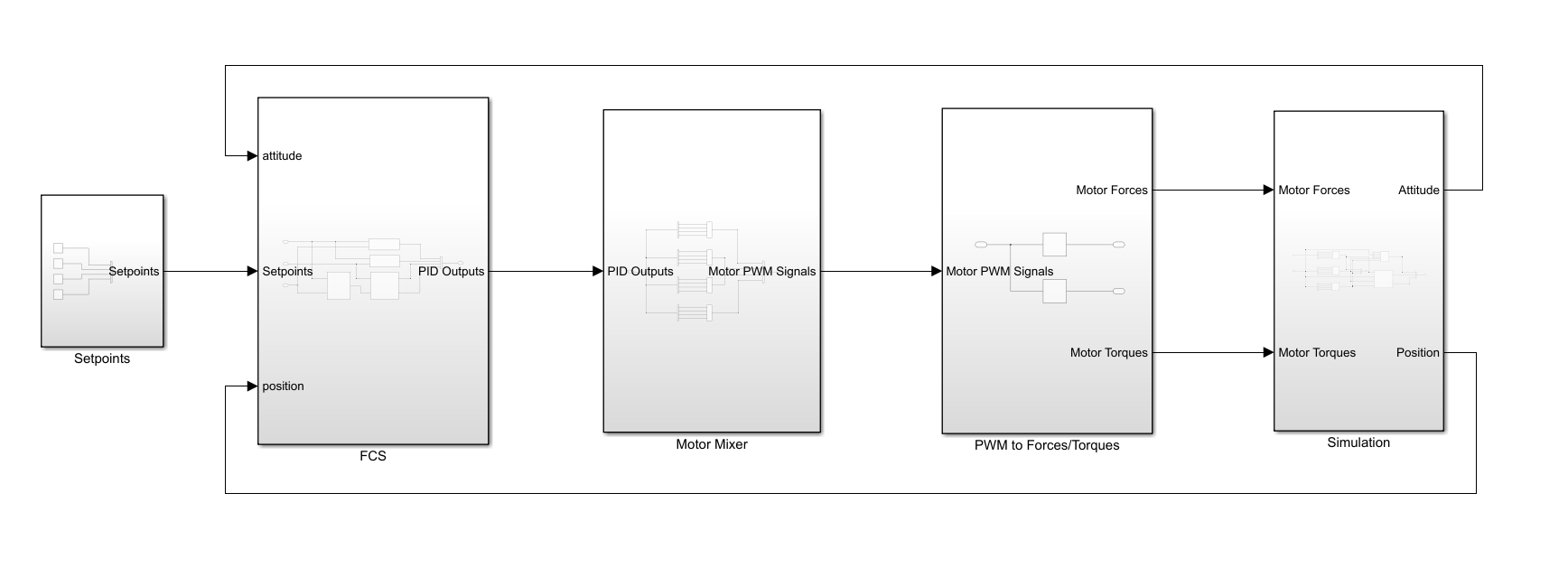
Developing a simulation for a control system is crucial for predicting its behaviour and is also very useful for tuning control systems in a non-destructive environment. The following model was designed using MATLAB SIMULINK. It includes a physics simulation derived from Newtons second law for both linear and rotational motion. It also includes the flight controller logic which is made up of a conversion from a body to ground reference frame, position controller, and an attitude controller. The PID outputs are inputted to the motor mixing algorithm and then converted form a PWM signal to a force in newtons.

Figure 6: Overview of SIMULINK Model

## Physics Simulation

This section is designed to determine of the outputs from the controller will affect the attitude and position of the vehicle. This is done by deriving equations for linear and rotational motion. The force produced by each motor can be predicted using test stand data. This means the rotational acceleration can be predicted using .

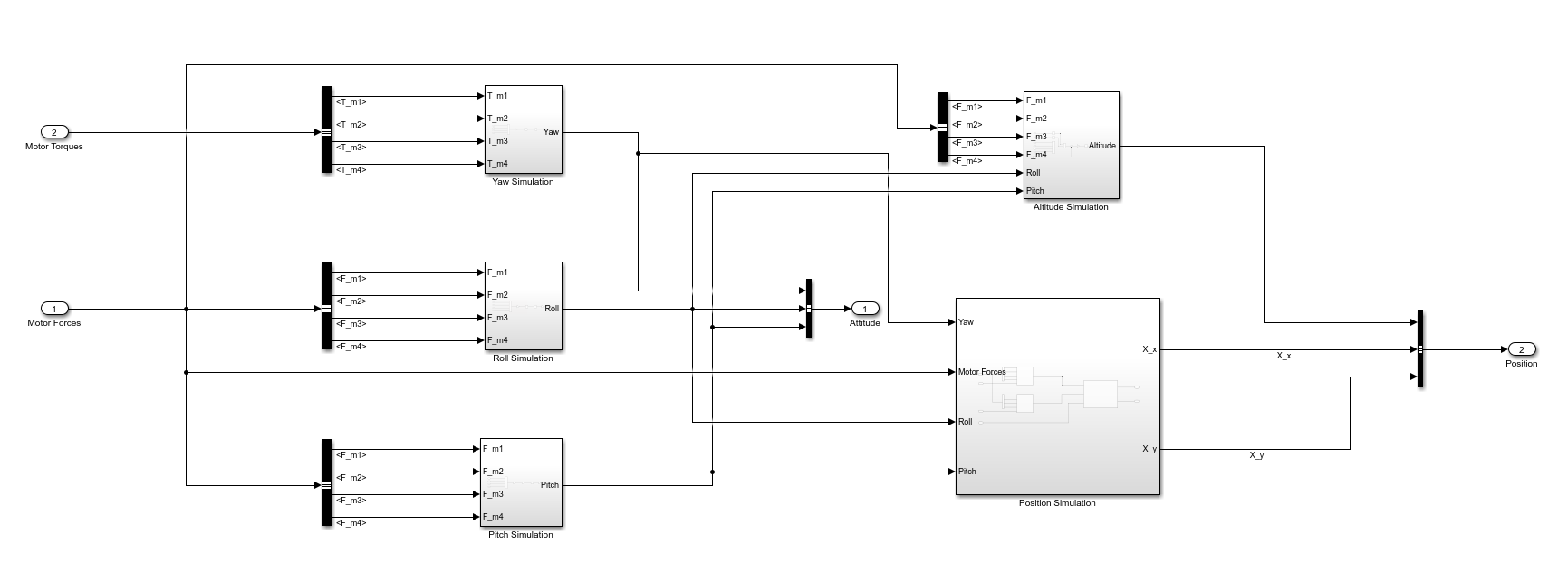


Figure 7: Overview of SIMULINK Physics Model

### Rotational Motion

Converting Newtons 2nd law to rotational forces gives . The sum of the forces produced by the four motors can be summed up depending on the axis on which motion is being predicted. For example, when predicting roll the sum of the motor forces acting on the drone is:

This can be converted into a torque by multiplying by the distance from the vehicle centre of mass to the applied force.

This can be rearranged to solve for :

Integrating over with respect to time will give the vehicles rotational velocity () in a single axis. Integrating with respect to time will give the vehicles rotational displacement from an initial attitude in radians. Therefore, the attitude of the drone can be described by the following equations:

Yaw is governed by the torque produced by each motor:

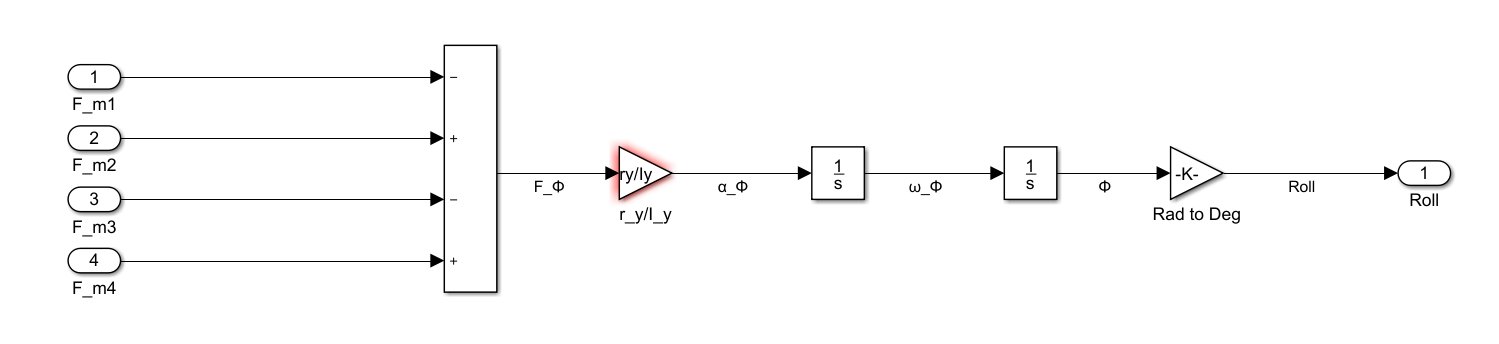
The pitch and roll tiles are shown in figure 7 highlighted by the blue square. Within the roll tile the equation given from above is implemented into SIMULINK as shown in figure 8.

Figure 8: Double Integration to estimate Roll implemented in SIMULINK

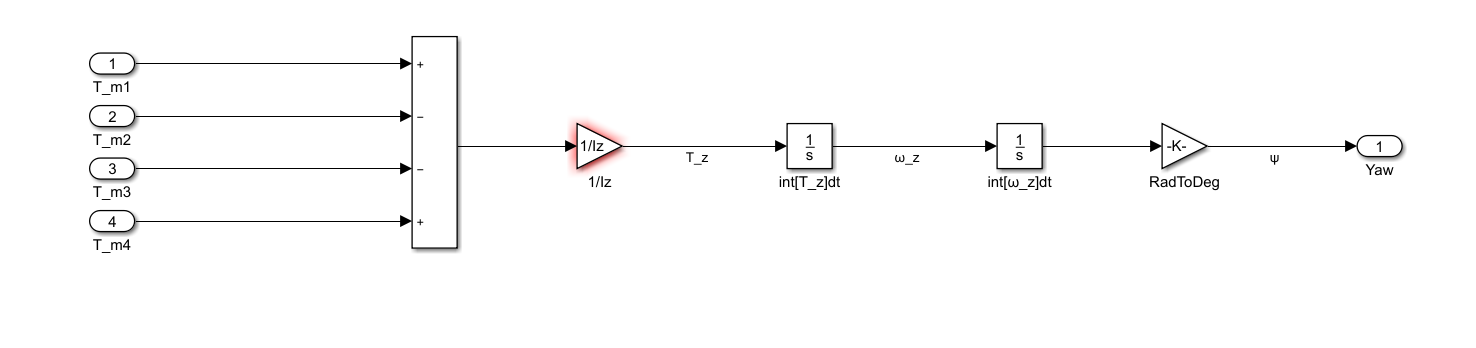
The yaw tile is highlighted in green in figure 7 and the SIMULINK solution is given by figure 9.

Figure 9: Double Integration to estimate yaw implemented in SIMULINK

### Linear Motion

Deriving equations to describe the linear motion of the drone is slightly more complicated then the rotational motion due to the under actuated nature of a quad-motor UAV. Linear motion is also governed by Newtons 2nd law . Here the force in only the x, y or z plane is wanted. This requires the force produced by the motors to be solved in the xyz directions using roll and pitch angles.

The forces required can be solved as follows:

Rearranging Newtons second law for acceleration gives . The double integral method can also be applied for linear motion:

The force in the z direction is given by the sum of the motor forces accounting for cosine losses due to roll and pitch angles minus the gravitational force .

# References

1. Karp, S., & Stotts, L. (2013). *Fundamentals of electro-optic systems design communications, lidar, and imaging.* Cambridge University Press. Chapter 9. Light detection and ranging