Full-State Feedback Controller

With an Eighth Order Observer and Defined Matrices

Design and Derivation by Marlon Hodge, The University of Texas at Dallas, 2012

Programming and Document by Gary Steelman, The University of Texas at Dallas, 2012

Contents

[Introduction 3](#_Toc338543293)

[The Quadcopter Hardware 3](#_Toc338543294)

[The States 4](#_Toc338543295)

[The Controller Equations 5](#_Toc338543296)

[Control Signal Updates 5](#_Toc338543297)

[Equation (1a): 5](#_Toc338543298)

[U Matrix 5](#_Toc338543299)

[K Matrix 6](#_Toc338543300)

[Equation (1b): 6](#_Toc338543301)

[Estimation Updates 6](#_Toc338543302)

[Equation (2a): 6](#_Toc338543303)

[A Matrix 7](#_Toc338543304)

[B Matrix 8](#_Toc338543305)

[L Matrix 8](#_Toc338543306)

[Y Matrix 9](#_Toc338543307)

[Equation (2b): 10](#_Toc338543308)

[Calculating the State Updates 10](#_Toc338543309)

[Velocity Updates 11](#_Toc338543310)

[Position Updates 11](#_Toc338543311)

[Angular Acceleration Updates 11](#_Toc338543312)

[Angular Position Updates 11](#_Toc338543313)

[Time Slice 11](#_Toc338543314)

[Using the Controller 11](#_Toc338543315)

[Works Cited 12](#_Toc338543316)

[Appendix A 13](#_Toc338543317)

# Introduction

The UTD Quadcopter (“the copter”) was created by The University of Texas at Dallas Robotics and Automation Society in 2011. Its development continues into 2012 with a full-state feedback controller (“the controller”) designed and derived by Marlon Hodge.

The controller accounts for twenty values (“states”) on the copter including world coordinate position, velocity, orientation, motor currents, and other values. Only twelve of the states are directly measurable given the copter’s current instrumentation, so the remaining eight states are estimated using an observer (“estimator”).

The controller takes as input a desired future state and calculates the changes necessary in the copter’s current state to reach . This is how the copter is controlled: through specifying a desired location, motor speed, or other value, and the controller executing the action automatically and within safety limits. See “Using the Controller” for more info.

# The Quadcopter Hardware

* An aluminum cross-bar frame.
* One motor on each end of the bars, for a total of four motors.
* One fan blade affixed to each motor, for a total of four blades.
* One accelerometer.
* One gyroscope.

The motors can be controlled independently, allowing the copter to change direction in mid-flight and do other assorted tricks. The motors are controlled through changing their voltages. Increased voltage supplied to a motor makes it spin faster, and a decreased voltage makes it spin slower. The accelerometer and gyroscope provide up to date acceleration and rotation speed readings, which allow the controller to predict how the copter will react and where it will go in the next time step. The copter does not have tachometers to read the angular speed of the blades nor does it have ammeters to read the motor voltages.

# The States

The copter tracks twenty states describing its current position, velocity, orientation, motor currents, etc., shown in the figure below.

The accelerometer (“accel”) allows us to directly measure acceleration in the x, y, and z directions. The gyroscope (“gyro”) allows us to directly measure the angular speeds of the yaw, pitch, and roll. All other states are based on these measurements, either by integrating or estimating. The bottom eight elements in the state are estimates, as denoted by the subscript est, and can be collectively called . These are estimates because the copter does not have instrumentation to measure these values, but these values are still necessary for copter functionality. All of the states, measured, calculated, and estimated, are stored together in a 20x1 column vector called *x* and are known as a “copter state”:

# The Controller Equations

The general equation for a full-state feedback controller is:

Where:

* is a matrix whose values are the first derivative of the machine states with respect to time.
* *A* is a matrix which represents the machine’s dynamics linearized at steady state. Its values are the constants of twenty state variables’ first-order differential equations. These values change if the copter itself changes, i.e., if a new frame is used, or different motors, etc.
* *X* is a matrix whose values are the machine states.
* *B* is a matrix with constants 1 over the motor Inductance. The product B multiplied by U describes how the machine state derivatives are changing.
* *U* is a matrix consisting of the four time varying control signals.

This is the model equation, and is not used directly because we must include an estimator. The full state update for can be calculated using two main equations:

1. Control signal update equation. This equation calculates the new values for control signals to feed into the controller to achieve . For the current copter, there are only four control signals, which correspond directly to the four motor voltages.
2. Estimation update equation. This equation calculates the values for the new state’s given the updated control signal values from equation 1a, the previous eight estimated states, the first twelve states, and first twelve state derivatives.

The equations are implemented as a set of matrices of varying sizes whose values represent different parts of the controller. The specifics are discussed in the following sections.

# Control Signal Updates

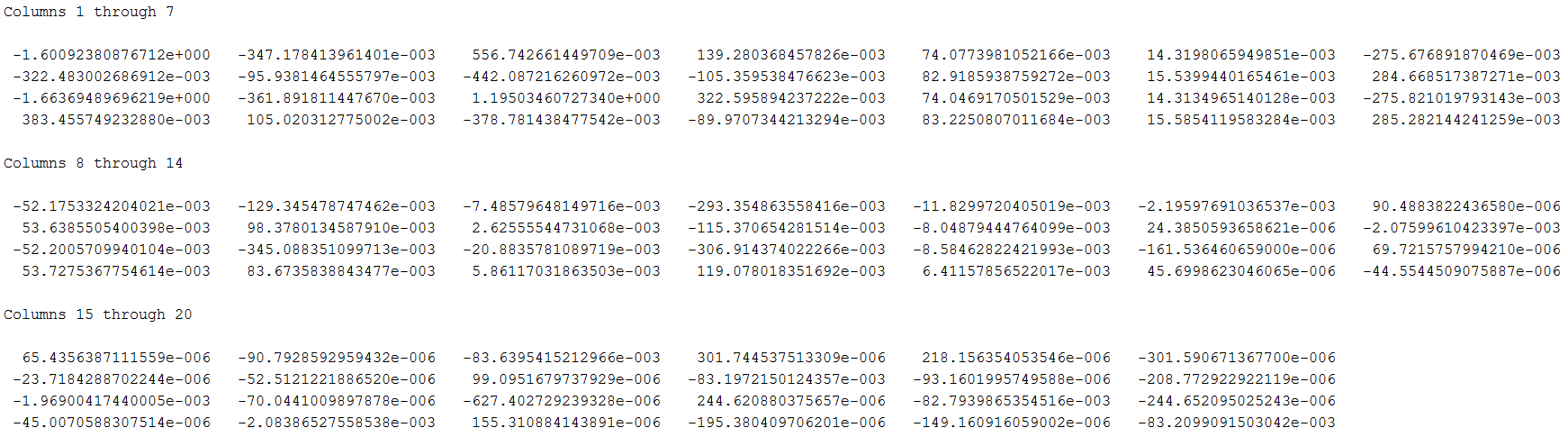
## Equation (1a):

### U Matrix

is the internal control signal update column vector whose values correspond directly to the voltage sent to the four motors:

### K Matrix

is the feedback gain matrix whose constants make the unstable poles of matrix A stable. It is generated from derivations in MATLAB (Hodge, 2012):



Equation (1a) calculates the difference between and to find how much change must happen to achieve the desired state and then multiplies that with the gain matrix *K*. However, we desire the copter to reach a steady state: the state of balance when hovering in mid-air where it doesn’t move. The controller accounts for this by altering equation 1a with an additional constant. This constant effectively cancels the force of gravity with constant motor forces, which suspends the copter indefinitely. The control signals in equation 1a must be further altered by the controller speed gain coefficient *G.* Equation 1b converts the internal control signal from voltages to the actually external control signal sent to the electronic speed controller:

## Equation (1b):

is the column vector of new controller signals that directly controls the motors.

# Estimation Updates

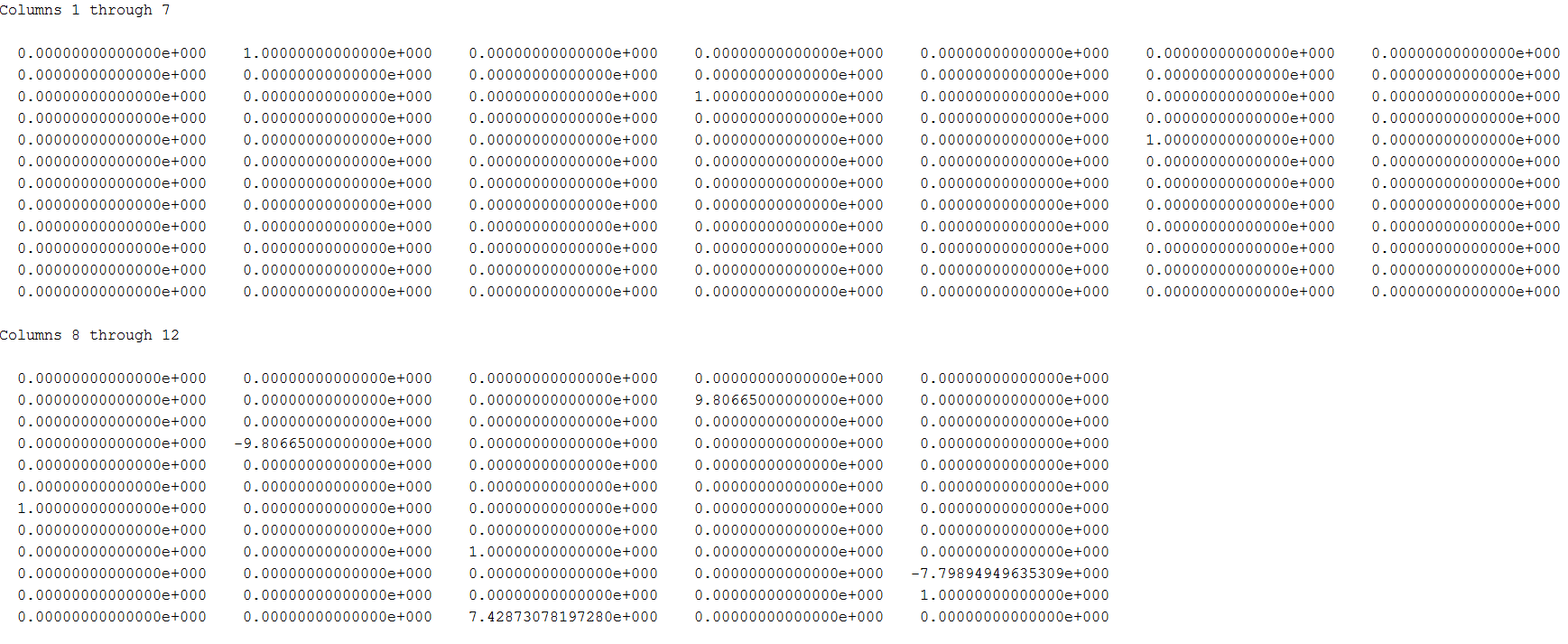
Once the controller calculates it can use the updated internal control signal values in conjunction with the previous eight estimated states, the first twelve states, and first twelve state derivatives to drive the estimates of the missing copter state values to the actual values of the machine. Recall the copter has an accelerometer and gyroscope, but no ammeter or tachometer, and thus cannot directly measure the voltage or angular speed of the blades. The controller estimates these values using Equation (2a) and Equation (2b).

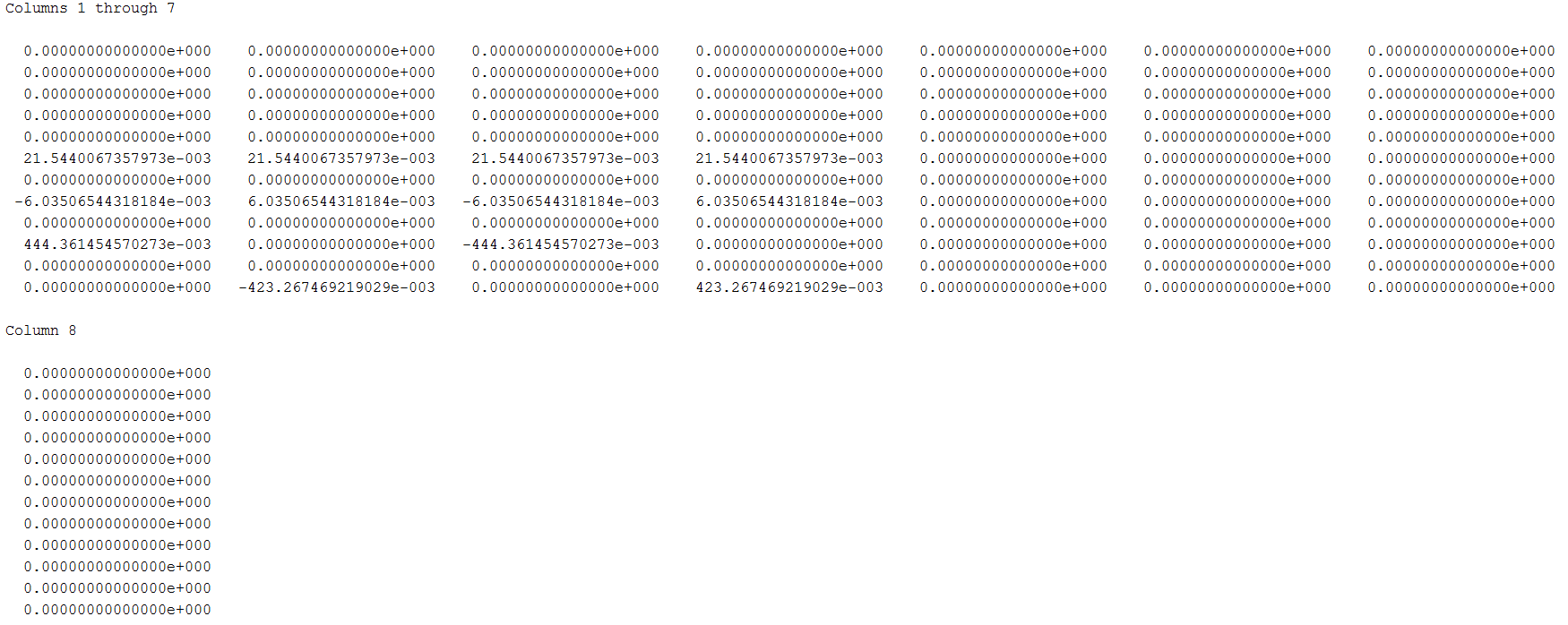
## Equation (2a):

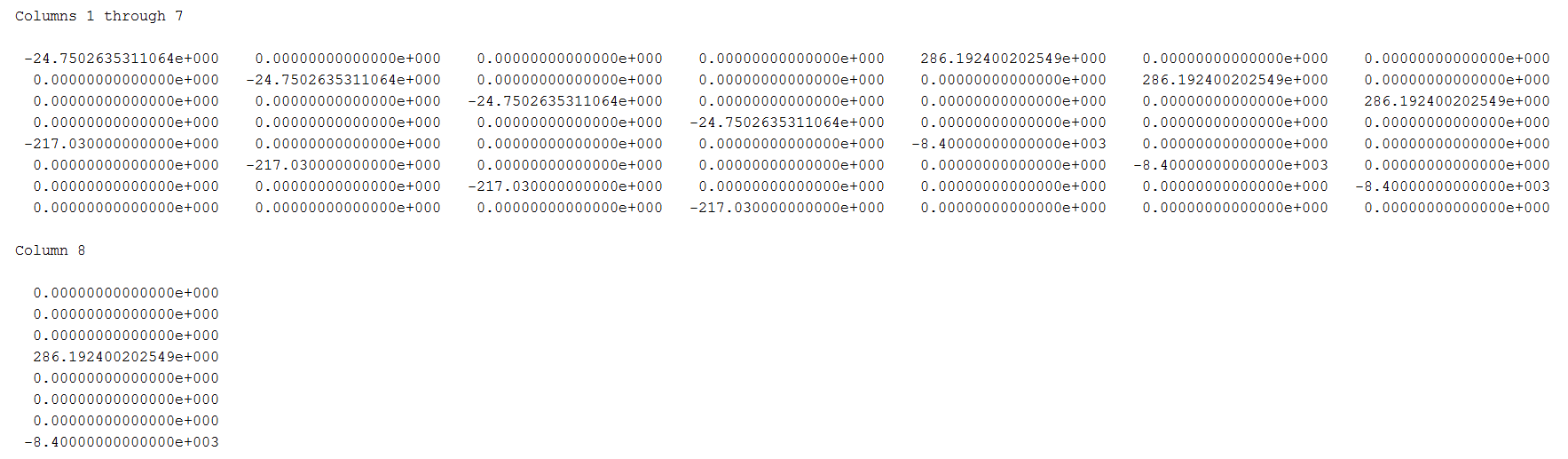
### A Matrix

The matrix represents the plant (“the copter” in this case) and is broken down into four sub-matrices:

Whose components are used in the estimation updates equation. It is broken down this way for efficiency. The *A* matrix is derived from the linearized nonlinear copter dynamics in Appendix A and will change when the copter’s physical form changes:

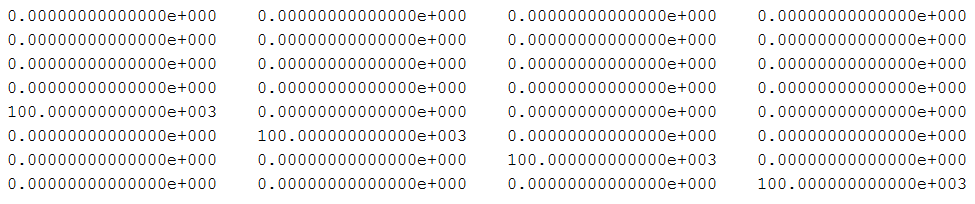






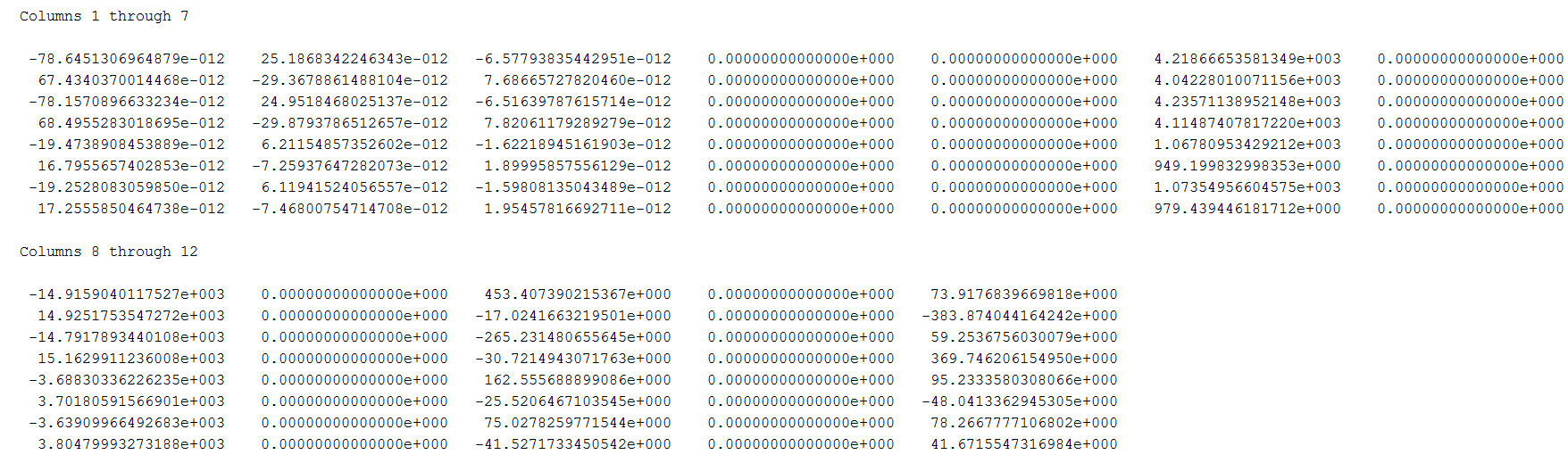
### B Matrix

The matrix represents a scaling on how much the change in *U* from Equation (1b) will affect . In the case of the copter, we break down *B* into two sub-matrices and concern ourselves only with the lower part, . This *B2* matrix contains values representing the inverse inductance which causes a linear change in the four currents as the control voltage *U* changes. These values are come directly from the nonlinear model in Appendix A.



### L Matrix

represents the 8th order observer’s gain matrix. The L matrix is generated from derivations in MATLAB (Hodge, 2012) similar to matrix K:



### Y Matrix

The Y column vector represents the correct sensor readings with gains applied. These correspond to the upper eight values of an *x* vector, .

Equation (2a) gives the state estimation in the form of its derivative. To obtain the actual estimate for we must integrate it.

## Equation (2b):

It is also important to correct the estimated states by adding the actual steady state values as illustrated below.

+

And last but not least, we combine the new along with the real values read from sensors (and their integrals) to obtain the new state .

# Calculating the State Updates

The equations derived and shown for state updates and estimation updates use differentials and integrals. These are entirely correct, mathematically speaking. However, in a real-time computer system calculating continuous integrals and differentials is technically challenging and impossible with low computing power. Thus we must discretize these calculations.

The integrals and differentials are with respect to time *t*. Therefore:

If two states are observed sufficiently close in time, the changes in acceleration, velocity, position, etc. can be approximated linearly.

Many of these equations can be pulled directly from a classical mechanics physics course.

## Velocity Updates

From the accelerometer’s measurements , , and and the previous state’s velocities, we can calculate the next velocities , , and , with:

## Position Updates

From the accelerometer’s measurements , , and and the previous state’s velocities , , and , we can calculate the next positions , , and with:

## Angular Acceleration Updates

From the gyroscope’s measurements , , and for the current and the previous states we can calculate the next accelerations , , and with:

## Angular Position Updates

From the gyroscope’s measurements , , and and the previous state’s accelerations , , and , we can calculate the next angular positions , , and with:

## Time Slice

is the time difference between one state measurement and the next:

The choice of time instants to evaluate the integrals and differentials must be carefully chosen.

This is a TODO and will depend on how the software is implemented.

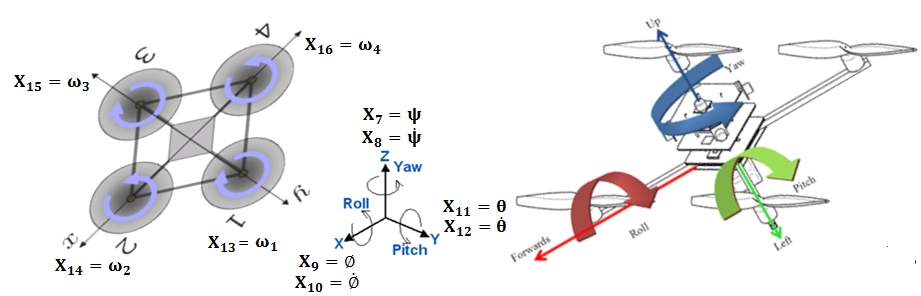
# Using the Controller

To move the copter, feed the controller a desired reference state matrix of the form

# Works Cited

Hodge, M. (2012). UTD Quadcopter MATLAB Scripts.

# Appendix A



Roll Ø, Pitch θ, Yaw ψ motor 1 & 3 clock wise; motor 2 & 4 counter clockwise