FPGA Controlled Quadcopter, Prototype Flight Control Algorithm

Capstone Team 32: Ethan Grinnell

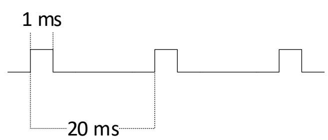
Revision 0.1

Sunday, December 24, 2017

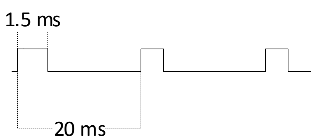
Standard and Continuous Rotation Servo PWM Signals

Duty Cycle: Every 20ms a servo expects a digital pulse to determine its operating position. The pulse ranges from 1ms to 2ms. The pulse width is always 1ms plus some additional pulse width given by the duty cycle. The duty cycle percentages indicate the fraction of 1ms that is added to the initial 1ms pulse. A 1ms pulse with no additional width is considered 0% duty cycle [1ms + 1ms (0%)]. A 2ms pulse is considered 100% duty cycle [1ms + 1ms (100%)].

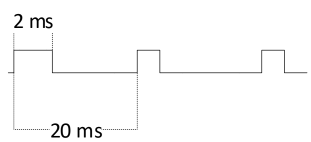
0% duty: Rotate to 0° OR Reverse full speed (Continuous Rotation Servo)



50% duty: Rotate to 90° OR Full stop (Continuous Rotation Servo)

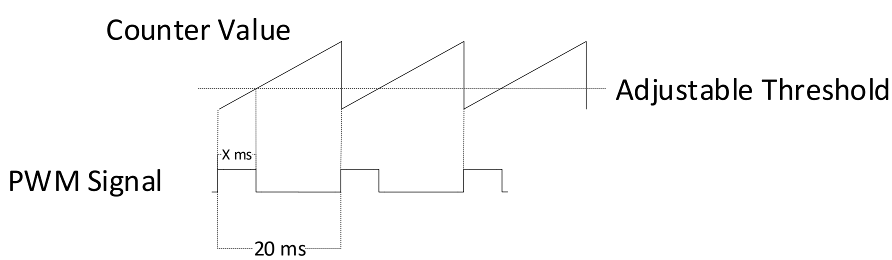


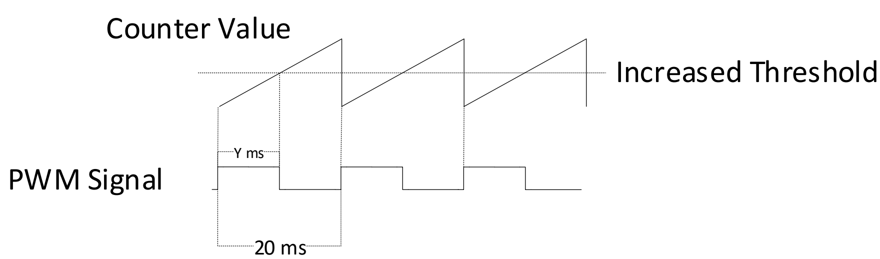
100% duty: Rotate to 180° OR Forward full speed (Continuous Rotation Servo)



Clocked Counter based PWM generation

The PWM signal is generated by way of a clock controlled up counter. Each clock tick increments the counter by 1. Every 20ms the up counter is reset to 0. When the counter is at 0 the PWM signal positive edge is produced. When the up counter reaches an adjustable threshold the PWM signal negative edge is produced. Increasing the threshold increases the pulse width of the PWM signal while decreasing the threshold decreases the pulse width. The clock interval, counter threshold maximum, minimums, and counter maximum values need to be selected so that it takes 1ms to achieve the minimum allowable threshold and 2ms to achieve the maximum threshold. The counter maximum value should be reached in exactly 20ms.





Aircraft Movement Terms

Z-axis: Vertical axis, increasing Z velocity increases the speed that the quadcopter gains altitude

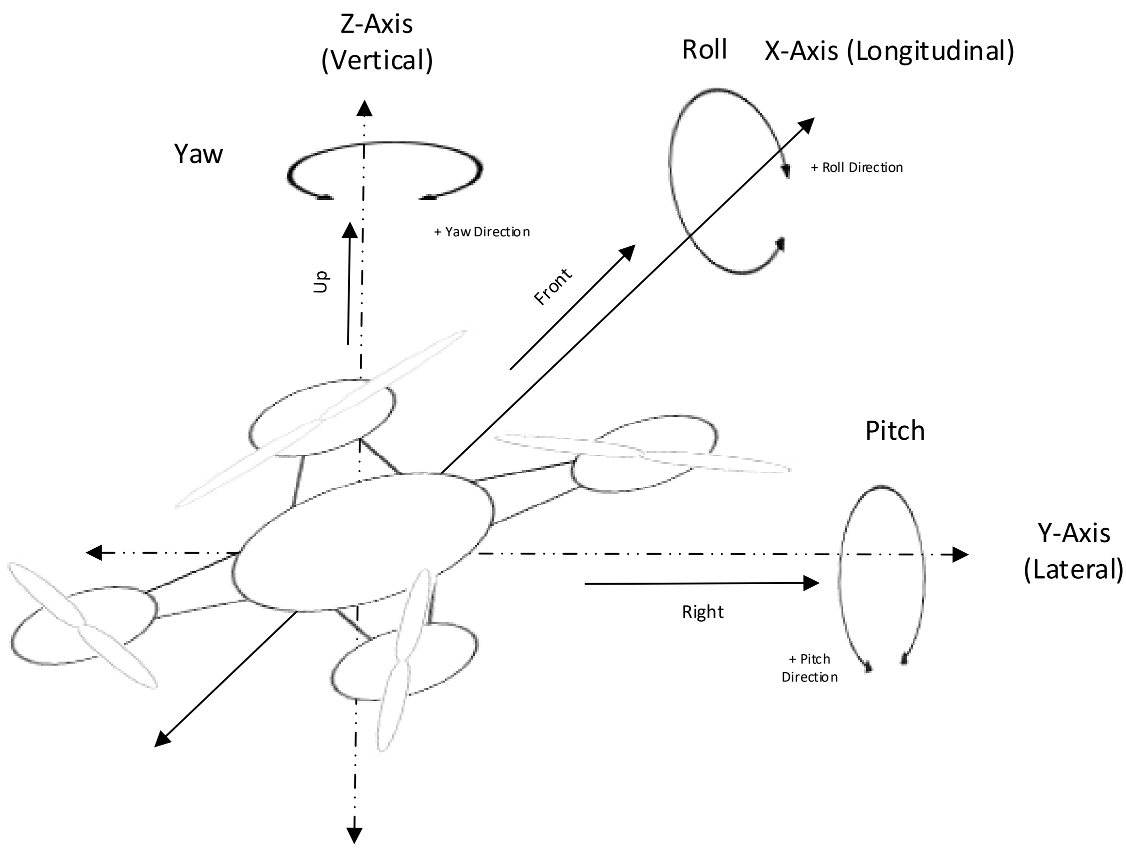
X-Axis: Front/Back horizontal axis, increasing X velocity increases speed in the longitudinal direction

Y-Axis: Left/Right horizontal axis, increasing Y velocity increases speed in the lateral direction

Yaw: Rotation about the Z-axis

Roll: Rotation about the X-Axis

Pitch: Rotation about the Y-Axis



Terms used in this document:

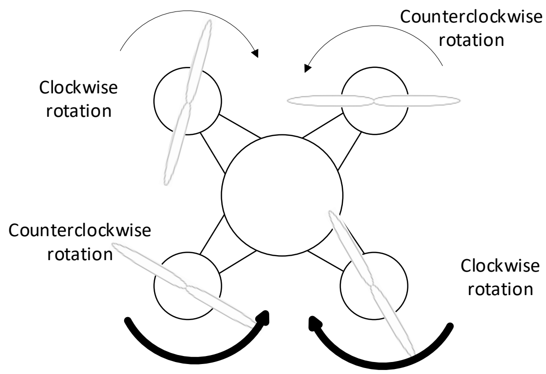
Control Plane: The plane formed by the controller input for the X, Y, and Z-axis. The control plane includes an offset from control stick inputs for automatic adjustment by the flight control algorithm for lateral and longitudinal axis drift. The control plane is used as the reference plane for angles used in the control algorithm.

Sensor Plane: The actual plane that the quadcopter is aligned to. This is calculated based on the gyroscope sensor's output which is used to determine the X and Y angles relative to the ground.

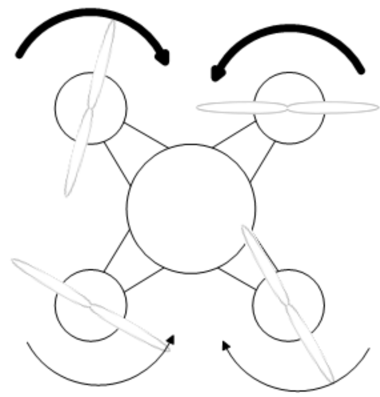
Rotor RPM changes for desired movements:

(Thicker arrows indicate an increase in rotor RPM, thinner arrows indicate a decrease in rotor RPM)

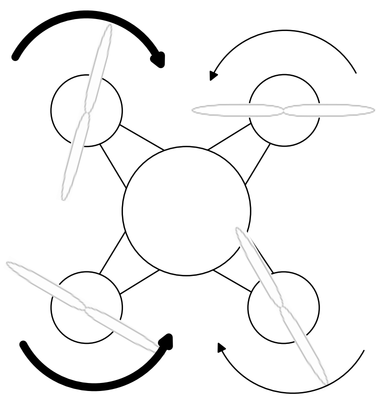
Forward Flight (Pitch Rotate Forward)



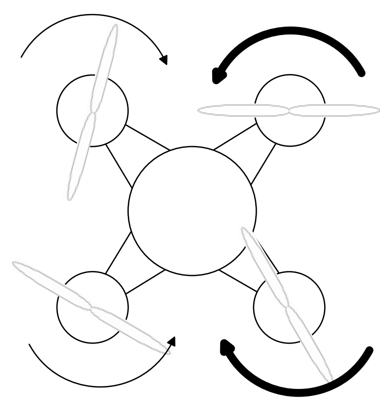
Reverse Flight (Pitch Rotate Rearward)



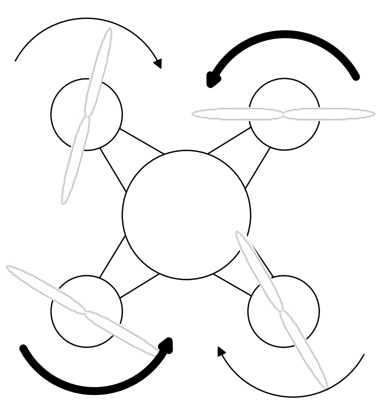
Lateral Shift Right (Roll Right)



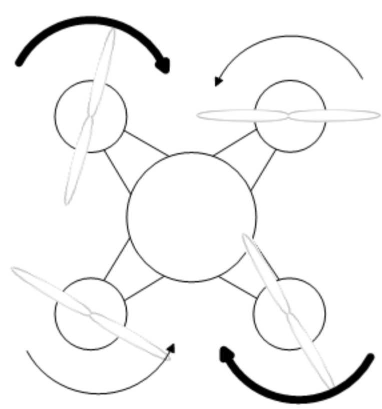
Lateral Shift Left (Roll Left)



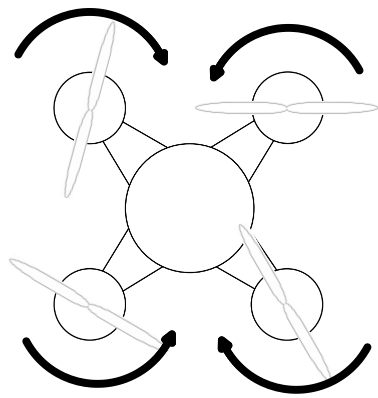
Yaw Rotate Clockwise



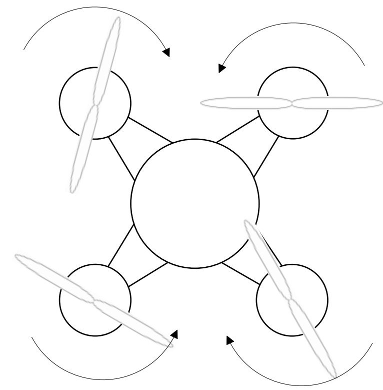
Yaw Rotate Counter Clockwise



Increase Altitude

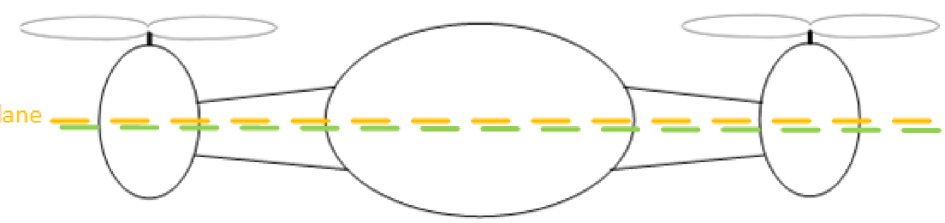


Decrease Altitude



Neutral controls at stable hover with no position/velocity errors

X/Y Sensor plane and X/Y control plane are identical, no change is made to motor RPMs. These planes are shown slightly offset below to indicate that they are separate planes, they would actually overlap entirely at a neutral stable hover. Otherwise the control algorithm would see this as a Z-axis deviation, and compensate.



Neutral hover horizontal motion cancellation (X/Y axis movement)

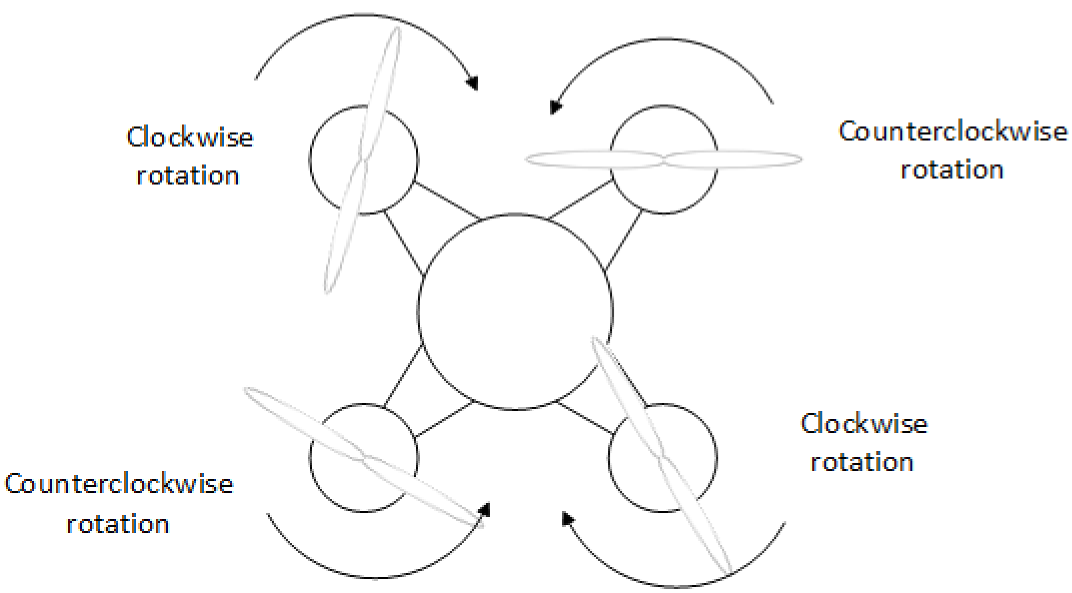
The quadcopter gets its horizontal velocity by tilting in the direction that it will travel. More tilt in that direction increases the speed. While hovering in the neutral position the control algorithm will temporarily add a small offset to the control plane angles to provide a negative input to cancel the unintended motion. Once this motion is gone the added offset is eliminated. The amount of offset will have to be tuned experimentally and will be proportional to the amount of unintended motion. We'll need to determine a mapping between velocity and roll/pitch angles to better predict the required negative angle to cancel out this motion.

Neutral hover vertical motion cancellation (Z axis movement)

The quadcopter attains vertical speed by increasing or decreasing the RPMs for all four motors in unison. An increase in motor RPM increases vertical upward velocity. A decrease in motor RPM decreases upward velocity/force. Once the upward force is no longer able to counteract gravity the quadcopter descends. The control system will continuously adjust all four motor RPMs to achieve a neutral hover. These RPM adjustments will probably be very small. We'll need to figure out a mapping to vertical speed to RPMs. Also, due to gravity, upward speed takes more force to attain than downward speed, thus, algorithmic compensations will need to take this into account as automatic adjustments are made. This control algorithm will also need to compensate for differences in air density (Air temperature, air pressure, and altitude) and altitude (Weaker gravity)

Neutral hover yaw rotation cancellation (Z-axis rotation)

The quad copter rotates it left front and right rear rotors in the clockwise direction, while the right front and left rear motors are rotated counter clockwise. The opposite set of rotations also works, so long as opposite corners rotate in the same direction and adjacent corners rotate in opposite directions.



If untended yaw rotation is detected the rotor set that is rotating in that direction will have its RPMs increased. Since these motors are on opposite corners from each other, increasing their respective RPMs will not result in any pitch or roll angle changes, this may result in changes in Z velocity however. As an example, if the quadcopter has yaw rotation in the clockwise direction the front left and rear right motors will increase their RPMs to compensate. The Z-axis drift compensation will then adjust all 4 motors in unison to cancel out the change in Z-axis velocity.

Neutral hover pitch/roll rotation cancellation (X/Y-axis rotation)

Both pitch and roll rotation deviations are detected the same way. If the control plane deviates from the sensor plane the quadcopter motors are adjusted to compensate. The details of this algorithm are explained in the example of X/Y motor control below.

Example of X/Y motor control for 10° roll right on control stick

Note: Yaw and Pitch inputs are assumed to be unchanged for this example

Terms:

$NeutralPWM$: The current PWM counter threshold value for each motor required to rotate at the set throttle position and for the quadcopter to maintain its neutral position (no change in orientation). This is different for each motor and is calculated elsewhere.

$K_{proportional}$: Experimentally determined proportional constant used to determine the amount of correction required to compensate for differences in control and sensor plane

$AdjustmentPWM$: Amount that $NeutralPWM$ will change to achieve targeted rotor RPM

$OutputPWM$: New threshold for PWM counter

Functions:

$OutputPWM = AdjustmentPWM + NeutralPWM$: New PWM counter threshold

$AdjustmentPWM = K_{proportional}(ControlAngle - SensorAngle)$: Control Algorithm

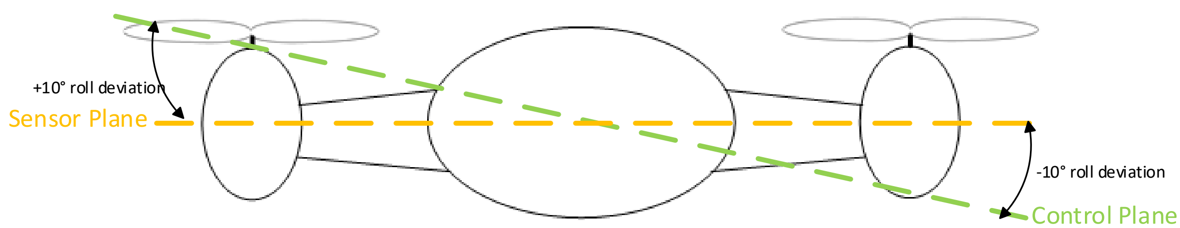
Constants: (Just used as examples, not known correct values)

$K_{proportional} = 5$

$NeutralPWM = 500$

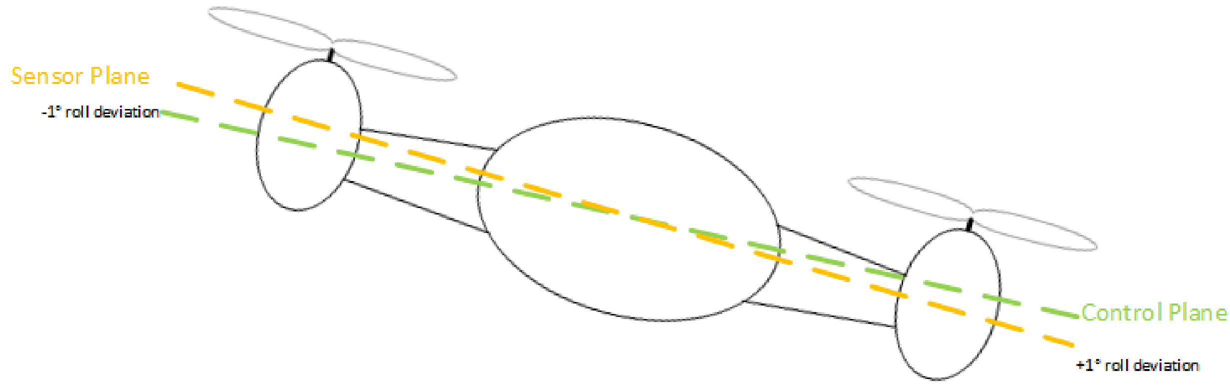
All motor RPMs = 1000

1. Control stick moved right 10°, sensor plan and control plane now are different, the control algorithm starts to make motor RPM adjustments



2. AdjustmentPWM for right motors are -50, and for the left motors are 50.
3. Final OutputPWM for right motors are 450 while the left motors are 550.
4. Right motors have shorter PWM pulse duty and reduce RPM. The left motors have larger PWM pulse duty and increase RPM.

5. This continues until the quadcopter rolls to the right enough so that the sensor plane overshoots the control plane.



(With a more complete control feedback system this overshoot could be reduced or eliminated)

6. A new AdjustmentPWM value is calculated for the right and left motor pairs. For right motors it becomes +5 while the left motors become -5.
7. New OutputPWM is calculated, the right motors change from 450 -> 455, the left motors change from 550 -> 545.
8. The right motors increase their RPMs slightly while the left motors decrease their RPMs, the quadcopter stabilizes at this angle.