Project report on

Aircraft Autopilot System Prototype

EEE 318: Control System 1 Laboratory

Submitted By:		**Submitted To:**
Name	Student ID	
Shakil Anwar	1506178	**Towsif Taher**
Shadman Shahid	1506179	Lecturer,
Saiful Bari Siddiqui	1506185	Depertment of Electrical and
Aswad Dewan Abir	1506189	Electronic Engineering

Department of Electrical and Electronic Engineering
Bangladesh University of Engineering and Technology

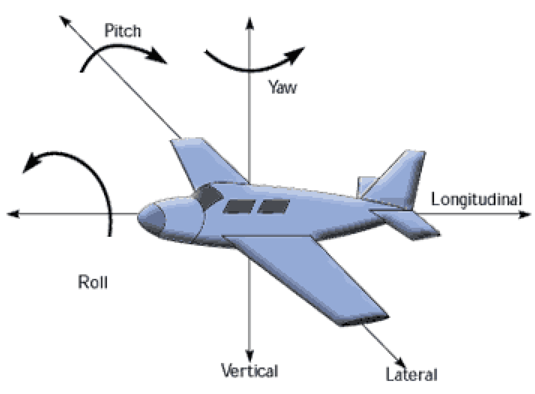
Abstract

An autopilot is a mechanical, electrical, or hydraulic system used to guide an aerial vehicle without assistance from a human being. It also maintains the orientation of the plane by monitoring the relevant flight data from inertial measurement instruments and then using that data to cause corrective actions. In this project an attempt has been made to design, implement and develop an autopilot for a glider plane. 3-axis accelerometer and gyroscopes are used to input the acceleration and tilt data into the controller. This data is then used for further estimation and fuzzy logic is implemented for decision making. The required corrective measures are affected by a set of servo motors which helps the flight path and orientation to be maintained at the desired levels. Our main target was to control the airplane's movement in case of a straight line flight with response of air and obstacles. We also tried to demonstrate a circular flight where we would control the turning angles of the airplane.

Introduction

In the early days of aviation, aircraft required the continuous attention of a pilot in order to fly safely. As aircraft range increased allowing flights of many hours, the constant attention led to serious fatigue. An autopilot is designed to perform some of the tasks of the pilot. Along the flight path the vehicle is under the influence of various accelerating forces in all directions and these factors cause it to deviate from its desired path. So the plane loses its heading as well as orientation. This is where autopilot comes into picture.

There are three levels of control in autopilots for smaller aircrafts. A single-axis autopilot controls an aircraft in the roll axis only. A two-axis autopilot controls an aircraft in the pitch axis as well as roll axis with pitch-oscillation-correcting ability. A three-axis autopilot adds control in the yaw axis and is not required in many small aircraft. The 3 different axes mentioned are shown in the following figure. The flight may also receive inputs from on-board radio navigation systems to provide true automatic flight guidance once the aircraft has taken off until shortly before landing.



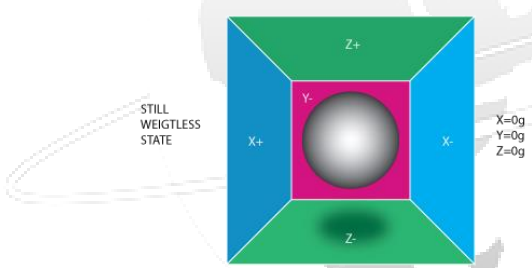
The basic objective of our project is to design and develop an auto pilot control system which can maintain the desired orientation of the glider. The acceleration data in all 3 axes are obtained by the combination of accelerometer and gyroscopes and the angles of roll, pitch and yaw are calculated. These values are taken for estimation using a PID controller and the resulting values helps us in the decision making. The flight is kept in its path and desired orientation with the help of servo motors.

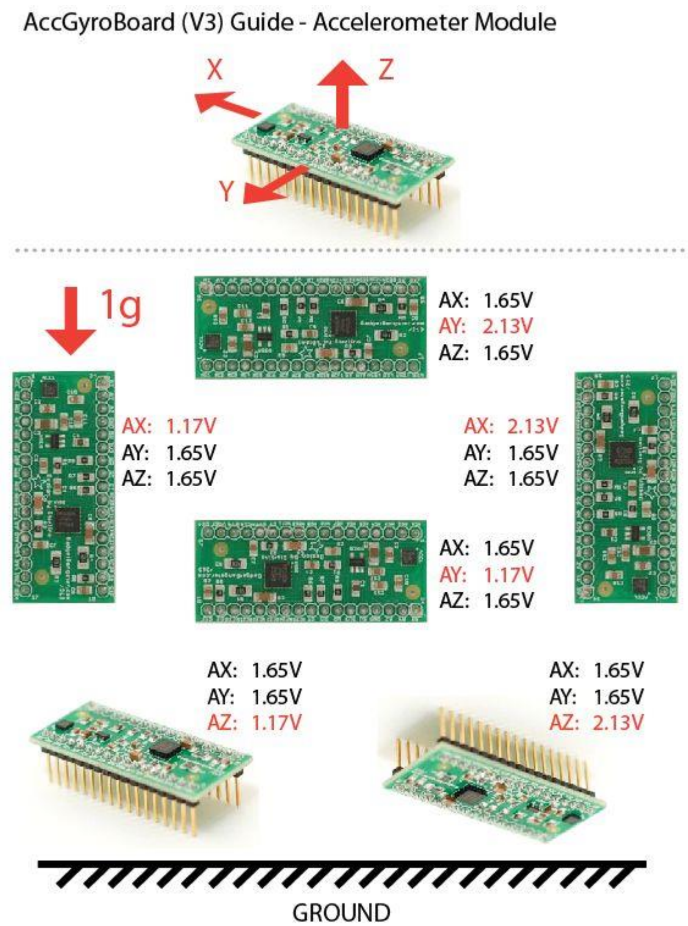
The basic system as shown comprises of an input block, a controller block and an actuator block. The input measures the angular velocity and acceleration of the setup. A 3-axis accelerometer axis used which gives precise acceleration measurements along the 3 axes. These acceleration values are later used to obtain the angular tilt along the 3 directions. Also 3 different gyroscopes are used for the angular measurements in the 3 different axes. A combination of the values from the accelerometer and gyroscopes are used and different weightages are given these 2 sets of values.

The microcontroller used is ATMEGA32. The input data is taken to the controller ADC modules. It is then passed to the Kalman filter for estimation. The estimated values are further taken to the fuzzy controller unit where the magnitude change from the desired orientation is used for the decision making. The servo motor interfacing unit decides the amount of rotation of the servos to help the flight maintain the desired orientation.

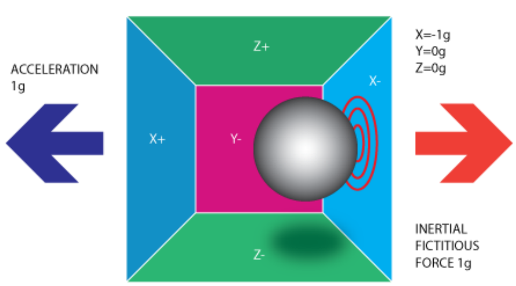
The actuator unit is the set of servo motors used for affecting the change in direction of the glider.

Accelerometer: To understand this unit we'll start with the accelerometer. When thinking about accelerometers it is often useful to image a box in shape of a cube with a ball inside it. You may imagine something else like a cookie or a donut, but I'll imagine a ball:

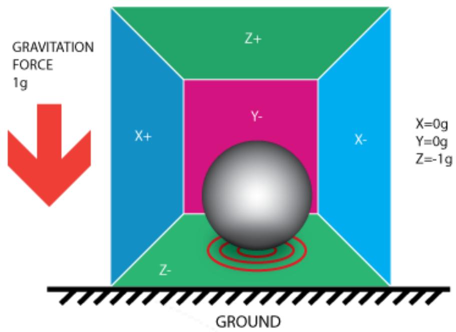




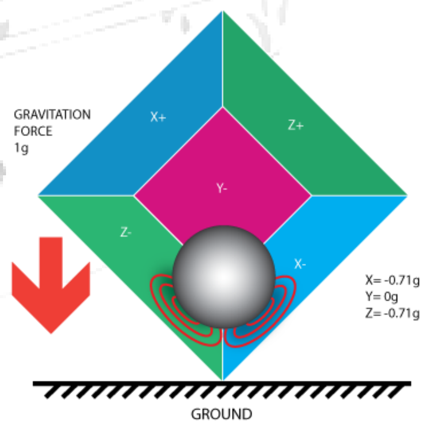
From the picture above you can see that we assign to each axis a pair of walls (we removed the wall Y+ so we can look inside the box). Imagine that each wall is pressure sensitive. If we move suddenly the box to the left (we accelerate it with acceleration 1g = 9.8m/s^2), the ball will hit the wall X-. We then measure the pressure force that the ball applies to the wall and output a value of -1g on the X axis.



Accelerometer will actually detect a force that is directed in the opposite direction from the acceleration vector. This force is often called Inertial Force or Fictitious Force . One thing you should learn from this is that an accelerometer measures acceleration indirectly through a force that is applied to one of its walls.
If we take our model and put it on Earth the ball will fall on the Z- wall and will apply a force of 1g on the bottom wall, as shown in the picture below:

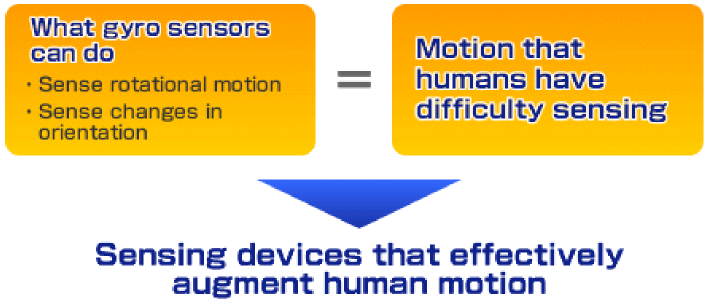


In this case the box isn't moving but we still get a reading of -1g on the Z axis. The pressure that the ball has applied on the wall was caused by a gravitation force. In theory it could be a different type of force. So far we have analyzed the accelerometer output on a single axis and this is all you'll get with a single axis accelerometers. The real value of tri-axial accelerometers comes from the fact that they can detect inertial forces on all three axes. Let's go back to our box model, and let's rotate the box 45 degrees to the right. The ball will touch 2 walls now: Z- and X- as shown in the picture below:

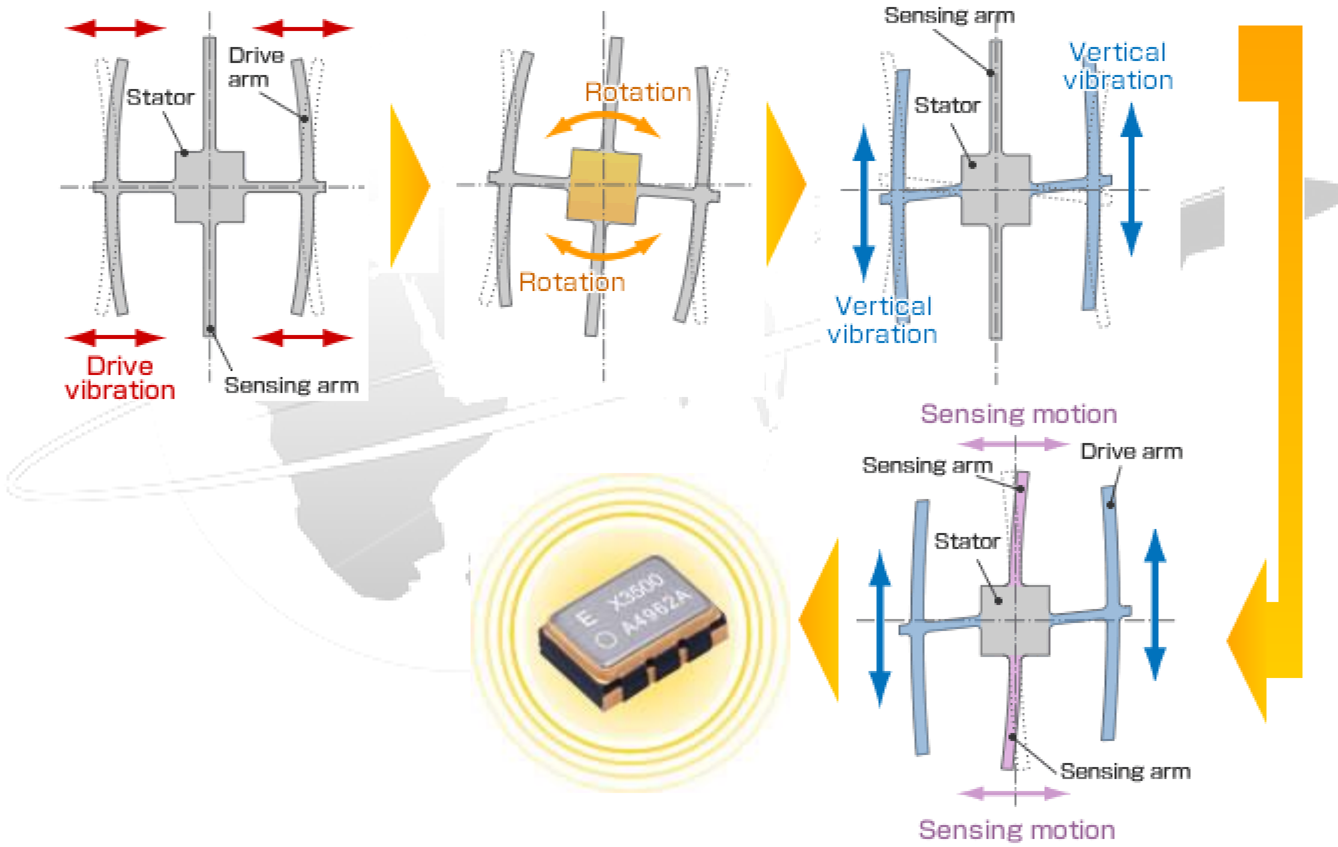


The values of 0.71 are not arbitrary, they are actually an approximation for SQRT(1/2). This will become clearer as we introduce our model.

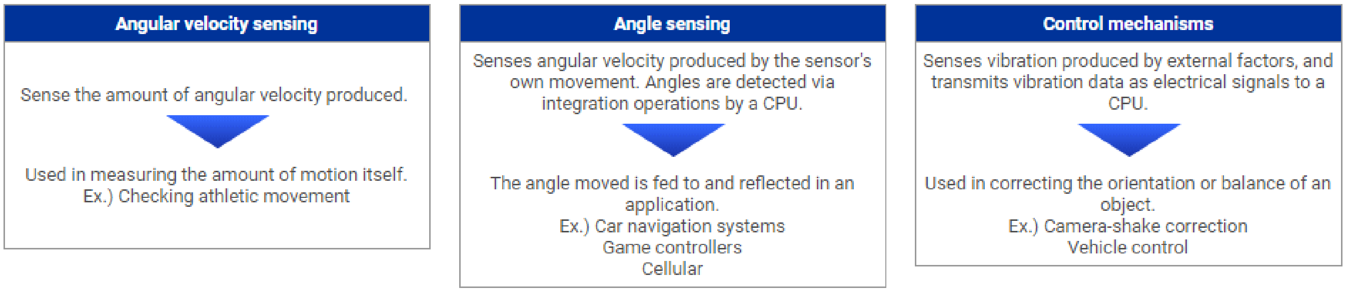
Gyroscope: Gyro sensors, also known as angular rate sensors or angular velocity sensors, are devices that sense angular velocity.



Vibration gyro sensors sense angular velocity from the Coriolis force applied to a vibrating object. Here, we explain how this works, using as an example Epson 's double-T structure crystal element.



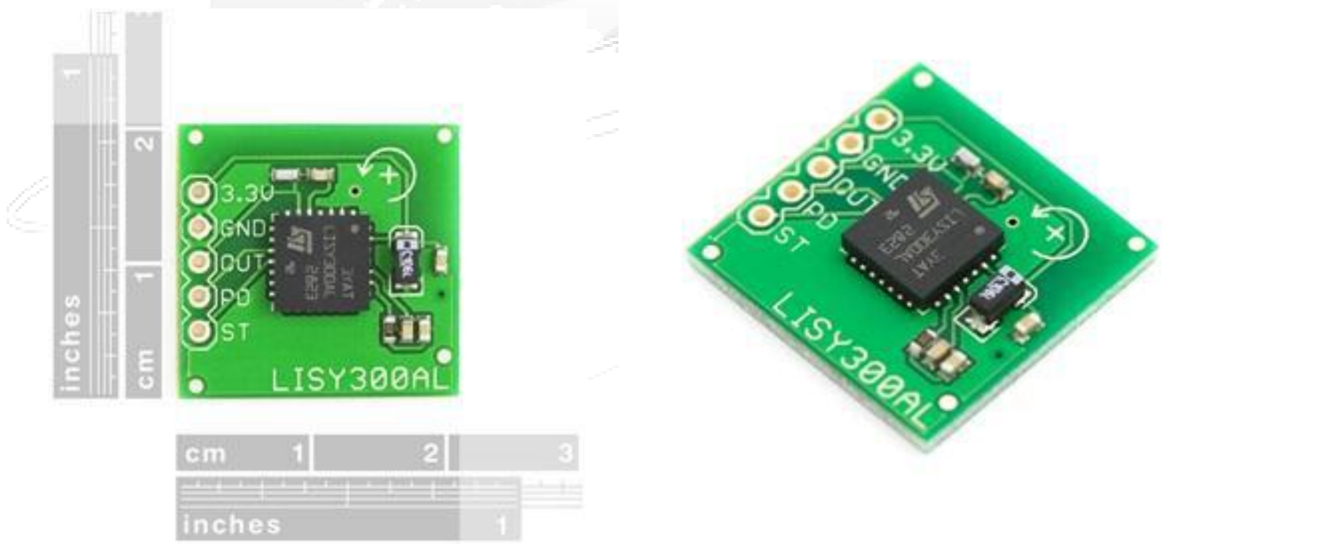
There are three main applications for gyro sensors.



The gyro used is a low power single axis one from ST Microelectronics with a 300 degree per second maximum range. A low-pass filter is integrated into the board along with a power down feature. The gyroscope (LISY300AL) outputs an analog voltage in proportion to the angular rate.

Features

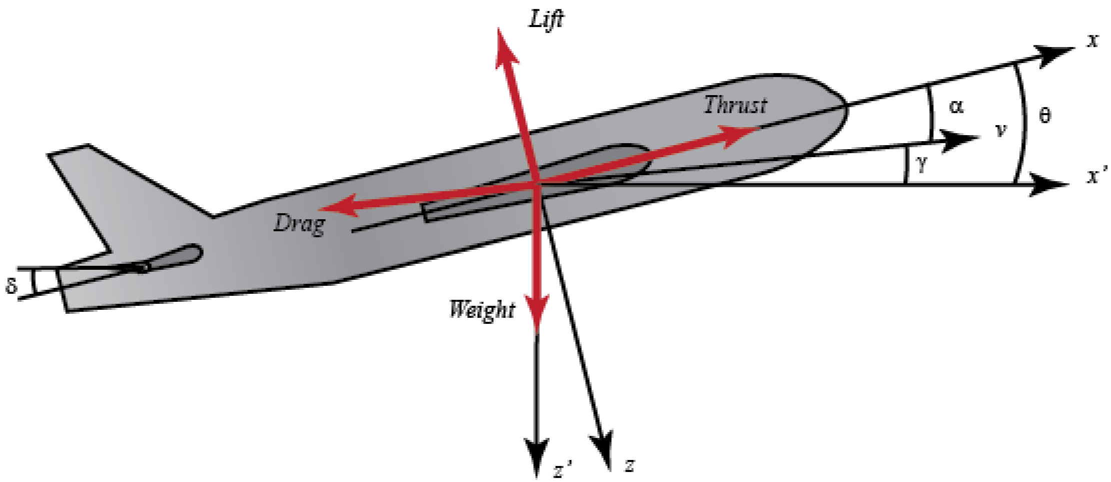
- 2.7V to 3.6V DC supply
- +/-300 degree/second output
- Analog rate out
- High shock survivability
- Embedded power-down feature



One main disadvantage of these gyroscopes is the drifting of measurement values. Even at static conditions after a period of time these gyroscopes give drifted values causing the system to interpret its position and orientation in the incorrect manner. The significant advantage with these gyroscopes is the smoothness in the values it shows. In accelerometers we obtain rough output values with a lot of noisy fluctuations.

Physical Setup and System Equations

The basic coordinate axes and forces acting on an aircraft are shown in the figure given below.



We will assume that the aircraft is in steady-cruise at constant altitude and velocity; thus, the thrust, drag, weight and lift forces balance each other in the x- and y-directions. We will also assume that a change in pitch angle will not change the speed of the aircraft under any circumstance (unrealistic but simplifies the problem a bit).

Before finding the transfer function and state-space models, let's plug in some numerical values to simplify the modeling equations shown above:

$$\dot{\alpha} = -0.313\alpha + 56.7q + 0.232\delta$$

$$\dot{q} = -0.0139\alpha - 0.426q + 0.0203\delta$$

$$\dot{\theta} = 56.7q$$

These values are taken from the data from model aircraft.

To find the transfer function of the above system, we need to take the Laplace transform of the above modeling equations. Recall that when finding a transfer function, zero initial conditions must be assumed. The Laplace transform of the above equations are shown below.

$$sA(s) = -0.313A(s) + 56.7Q(s) + 0.232\Delta(s)$$

$$sQ(s) = -0.0139A(s) - 0.426Q(s) + 0.0203\Delta(s)$$

$$s\Theta(s) = 56.7Q(s)$$

After few steps of algebra, you should obtain the following transfer function.

$$P(s) = \frac{\Theta(s)}{\Delta(s)} = \frac{1.151s + 0.1774}{s^3 + 0.739s^2 + 0.921s}$$

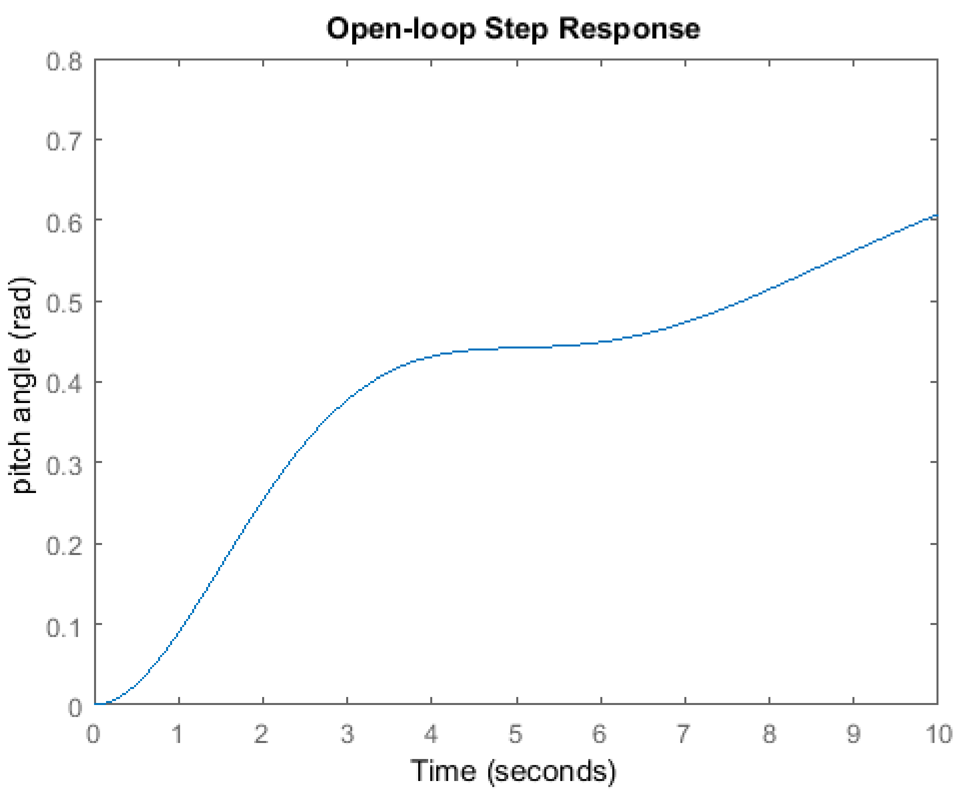
Design Requirements

The next step is to choose some design criteria. In this example we will design a feedback controller so that in response to a step command of pitch angle the actual pitch angle overshoots less than 10%, has a rise time of less than 2 seconds, a settling time of less than 10 seconds, and a steady-state error of less than 2%. For example, if the reference is 0.2 radians (11 degrees), then the pitch angle will not exceed approximately 0.22 rad, will rise from 0.02 rad to 0.18 rad within 2 seconds, will settle to within 2% of its steady-state value within 3 seconds, and will settle between 0.196 and 0.204 radians in steady-state.
In summary, the design requirements are the following.

- Overshoot less than 10%
- Rise time less than 2 seconds
- Settling time less than 3 seconds
- Steady-state error less than 2%

Now, we are ready to represent the system using MATLAB. Running the following code in the command window will generate the open-loop transfer function model described above.

```
s = tf('s');
P_pitch = (1.151*s+0.1774)/(s^3+0.739*s^2+0.921*s)
```



From the above plot, we see that the open-loop response does not satisfy the design criteria at all. In fact, the open-loop response is unstable. Stability of a system can be determined by examining the poles of the transfer function where the poles can be identified using the MATLAB command pole as shown below.

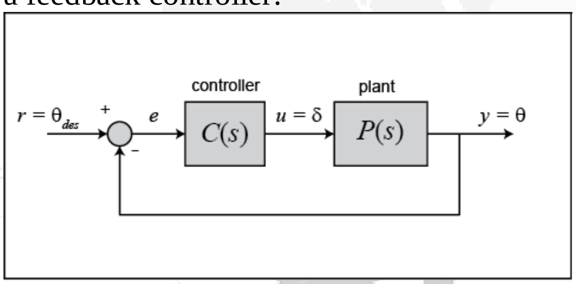
```
pole(P_pitch)
```

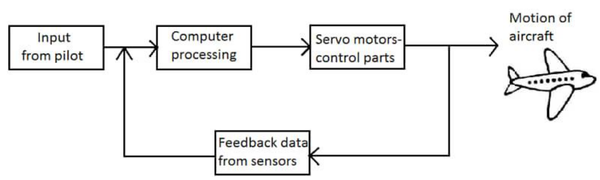
ans =
  0.0000 + 0.0000i
 -0.3695 + 0.8857i
 -0.3695 - 0.8857i

As indicated by this function, one of the poles of the open-loop transfer function is on the imaginary axis while the other two poles are in the left-half of the complex s-plane. A pole on the imaginary axis indicates that the free response of the system will not grow unbounded, but also will not decay to zero. Even though the free response will not grow unbounded, a system with a pole on the imaginary axis can grow unbounded when given an input, even when the input is bounded. This fact is in agreement with what we have already seen. In this particular case, the pole at the origin behaves like an integrator. Therefore, when the system is given a step input its output continues to grow to infinity in the same manner that an integral of a constant would grow to infinity as the upper limit of the integral is made larger.

Designing Controller

In order to stabilize this system and eventually meet our given design requirements, we will add a feedback controller:





A PID controller uses three terms to calculate an output that should correct the error.
Output = (kp*error)+(ki*errorSum)+(kd*dError);
The first term (the proportional term)

- It looks at our current inputs (pitch & roll in our case) and compares it to the desired position aka setpoint (0 degrees for level flight).
- The calculated error (difference) is multiplied by a constant (KP) to contain the value within our output range (0 to 180 deg is servo range, but it is typically smaller once installed in an RC aircraft).
- Basically, the greater the error, the greater the necessary change.

The second term is the integral term.

- It sums all previous error (as captured by the proportional term) over time.
- If the error remains uncorrected for too long, the integral term grows larger thus increasing the magnitude of our output until the correction is made.
- Again, multiply it by you constant (KI) to contain the term within the output range

The third term is the derivative term,

- The derivative term (dError) looks at the rate that your sensor inputs are approaching your setpoint in order to project when the error will be corrected.
- The derivative term can be used to slow down the output if it is correcting too fast and is likely to overshoot the setpoint.
- Multiply it by the constant KD to keep the output within the possible range.

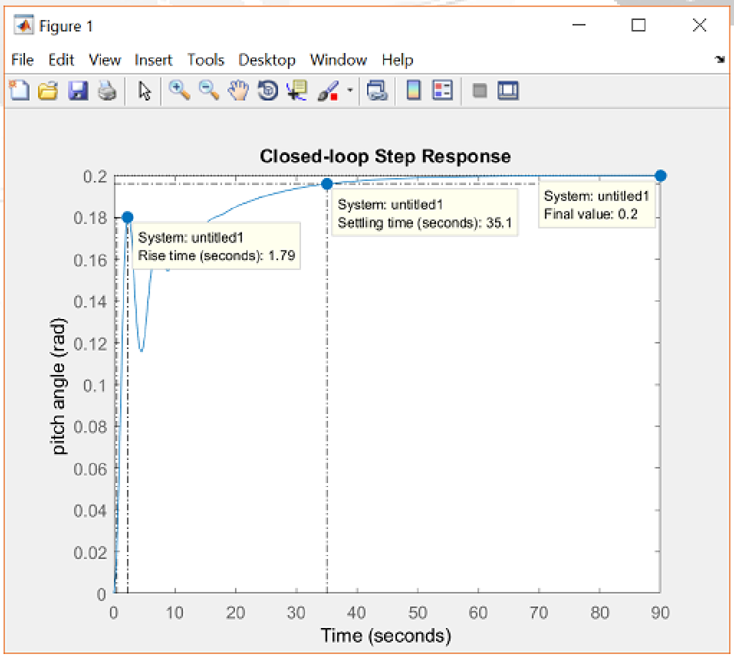
The closed-loop transfer function for the above with the controller C(s) simply set equal to one can be generated using the MATLAB command feedback as shown below.

```
sys_cl = feedback(P_pitch,1)
```

The corresponding step response can be generated by adding the above and following commands to your m-file. Note that the response is scaled to model the fact that the pitch angle reference is a 0.2 radian (11 degree) step. Running your m-file at the command line will produce the plot shown below where the annotations for the rise time, settling time and final value can be added to the plot from the right-click menu under Characteristics.

```
step(0.2*sys_cl);
ylabel('pitch angle (rad)');
title('Closed-loop Step Response');
```

Examining the above closed-loop step response, the addition of feedback has stabilized the system. In fact, the



steady-state error appears to be driven to zero and there is no overshoot in the response, though the rise-time and settle-time requirements are not met. The character of the resulting step response is indicated by the location of the poles and zeros of the system's transfer function, in a similar manner to the way the system's stability properties were. The MATLAB commands pole and zero can be used to reveal the poles and zeros of the closed-loop transfer function as shown below.

```
poles =
  -0.3255 + 1.3816i
  -0.3255 - 1.3816i
```

```
  -0.0881 + 0.0000i

zeros =
   -0.1541
```

The above results demonstrate that the closed-loop transfer function is third order with a zero. Most of the relationships that we are familiar with for predicting the character of a system's step response assume a standard underdamped second-order system with no zeros. Therefore, we cannot rely on these relationships for this system. We can, however, transform the output back to the time domain to generate a time function for the system's response to get some insight into how the poles and zeros of the closed-loop transfer function affect the system's response. Assuming the closed-loop transfer function has the form Y(s) / R(s), the output Y(s) in the Laplace domain is calculated as follows where R(s) is a step of magnitude 0.2.

$$Y(s) = \frac{1.151s + 0.1774}{s^3 + 0.739s^2 + 2.072s + 0.1774} R(s)$$

$$= \frac{0.2(1.151s + 0.1774)}{s^4 + 0.739s^3 + 2.072s^2 + 0.1774s}$$

We can then perform a partial fraction expansion in order to break this expression into simpler terms that we hopefully recognize and are able to covert from the Laplace domain back to the time domain. First we will use the MATLAB command zpk() to factor the numerator and denominator of our output Y(s) into simpler terms.

```
R = 0.2/s;
Y = zpk(sys_cl*R)
```

```
Y =

            0.2302 (s+0.1541)
  --------------------------------------
  s (s+0.08805) (s^2 + 0.6509s + 2.015)

Continuous-time zero/pole/gain model.
```

Based on the above, the denominator of our output Y(s) can be factored into a first-order term for the real pole of the transfer function, a second-order term for the complex conjugate poles of the transfer function, and a pole at the origin for the step input. Therefore, it is desired to expand Y(s) as shown below.

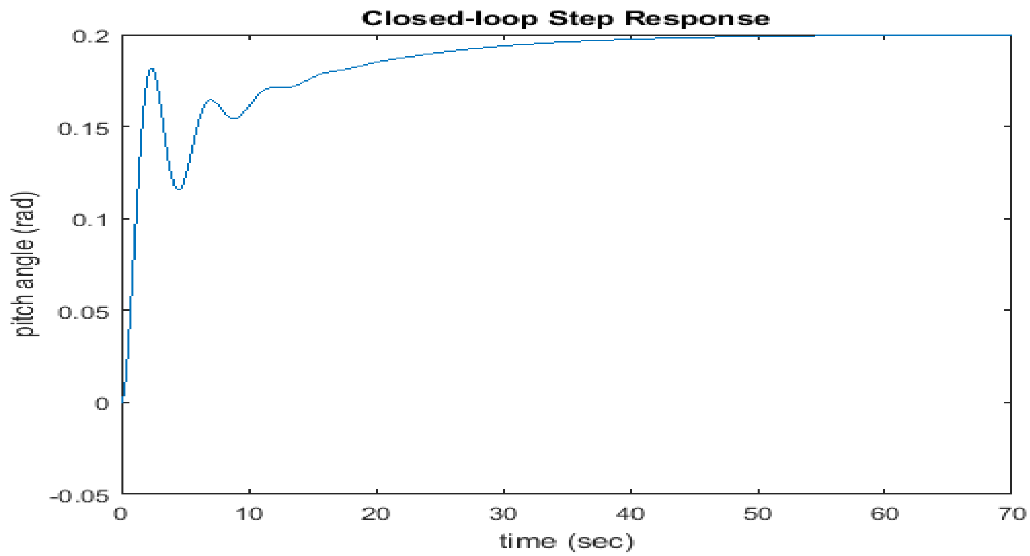
$$Y(s) = \frac{0.2302(s + 0.1541)}{s(s + 0.08805)(s^2 + 0.6509s + 2.105)}$$

$$= \frac{A}{s} + \frac{B}{s + 0.08805} + \frac{Cs + D}{s^2 + 0.6509s + 2.015}$$

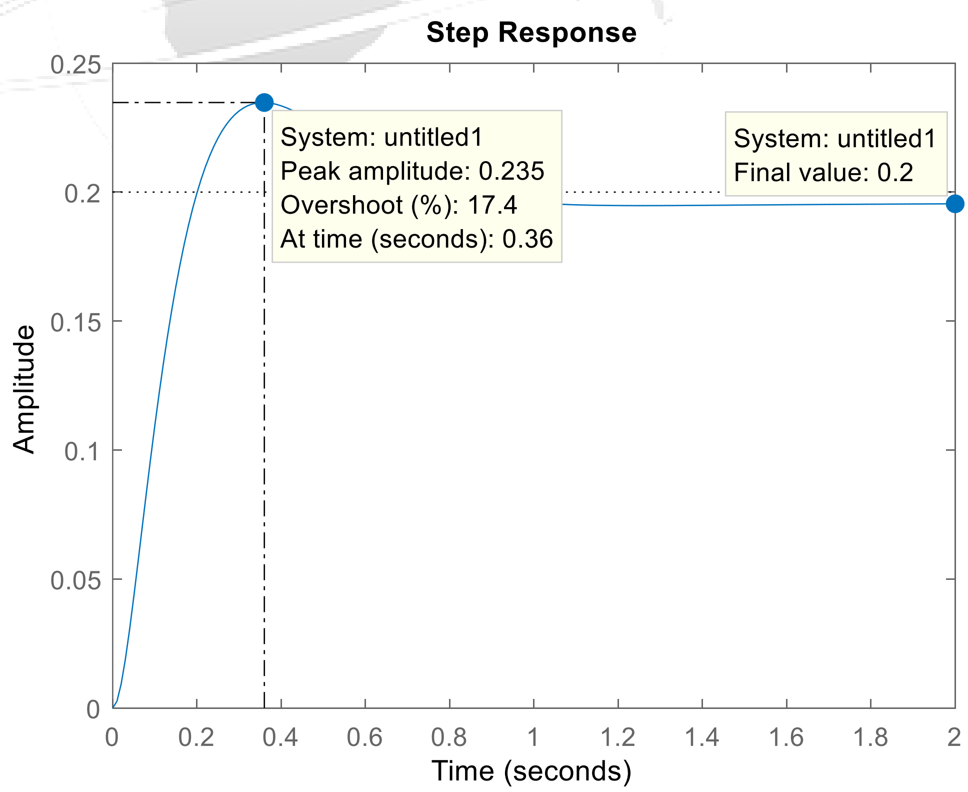
The specific values of the constants A, B, C, and D can be determined by hand calculation or by using the MATLAB command residue to perform the partial fraction expansion as shown below. Here the syntax is [r,p,k] = residue(num,den) where num and den are arrays containing the coefficients of the numerator and denominator, respectively, of the Laplace function being expanded. Notice that the denominator array includes a placeholder zero since there is no constant term in the denominator of Y(s).

```
[r,p,k] = residue(0.2*[1.151 0.1774],[1 0.739 2.072 0.1774 0])
```

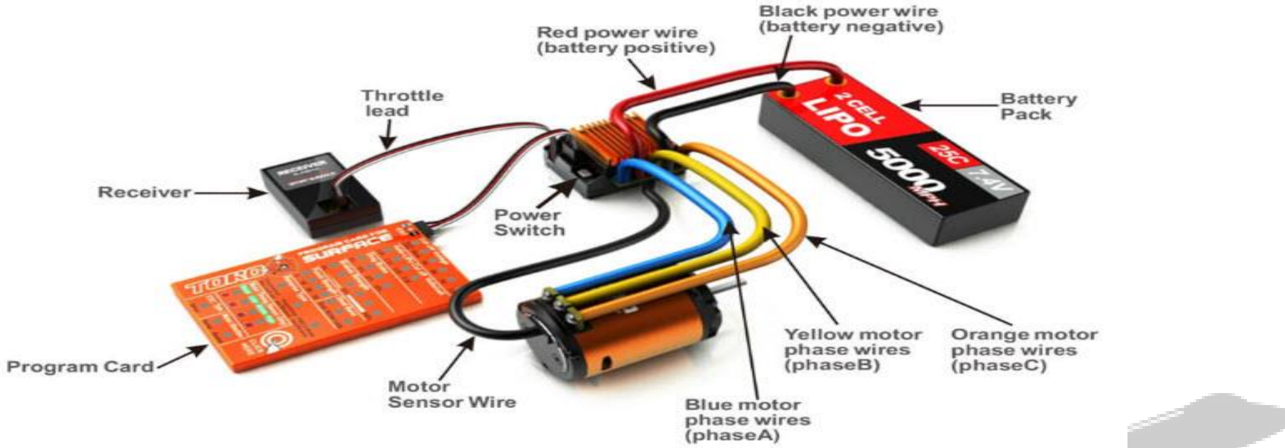
A desired output of the system response is shown below:



A simplified more desiring step response:



Electronic Speed Control: The term ESC stands for an electronic speed control is an electronic circuit used to change the speed of an electric motor, its route and also to perform as a dynamic brake. These are frequently used on radio controlled models which are electrically powered, with the change most frequently used for brushless motors basically providing an electronically produced 3-phase electric power low voltage source of energy for the motor.



As we know that, an ESC controls the speed of the motors spin of an airplane. It helps the similar purpose as the throttle servo of a glow powered airplane. It is an edge between the radio receiver of an airplane and the power plant. An electronic speed control will have 3- sets of wires. One wire will plug into the main battery of an airplane. The second wire will have a typical servo wire that plugs into the receiver's throttle channel. And lastly, a third of wire is used for powering the motor. The main features of an electronic speed control include battery eliminator circuit, low voltage cutoff, brake etc.

Battery Eliminator Circuit: A battery eliminator circuit (BEC) is an electronic circuit designed to deliver electrical power to other circuitry without the need for multiple batteries. Historically the expression was sometimes used to describe devices used to power battery-driven equipment from mains electricity.
Typically, the BEC is part of electronic speed control (ESC) which acts as the motor's throttle. Conceptually, it is completely separate from the part of the ESC which controls the motor speed, although the two parts do interact somewhat, and share some of the same wiring. Following

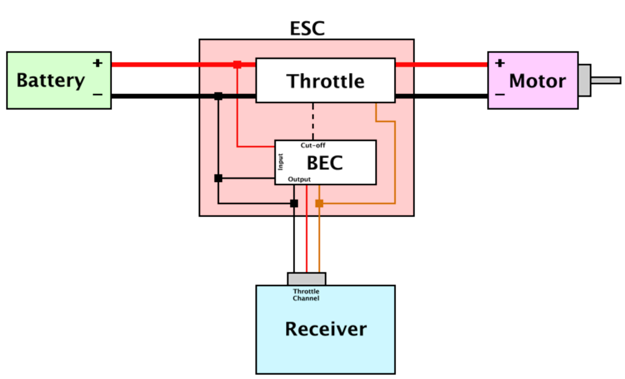


figure 1 shows the components of a typical power system, and how it relates to the internal arrangement of a BEC-equipped ESC.

Summary of the main components:

Components	Description
Brushless DC Motor	Rating- 2450 KV
Battery Eliminator Circuit	Rating- 16V , 25A
Electronic Speed Controller	Rating- 30A

The picture of our model aircraft is as follows:

Result

Flight No.	Date and Time	Duration
Flight 1	5 February, 4.00 pm	12s
Flight 2	5 February, 4.12 pm	15s
Flight 3	5 February, 4.30 pm	20s
Flight 4	5 February, 4.45 pm	18s
Flight 5	5 February, 5.12 pm	17s
Flight 6	6 February, 5.00 pm	10s
Flight 7	6 February, 5.10 pm	12s
Flight 8	10 February, 4.40 pm	18s
Flight 9	10 February, 4.55 pm	22s
Flight 10	12 February, 11.00 am	8s

After the 10th flight, our aircraft fell from 15 ft height in the mud and broke into pieces.

Fig: Broken Aircraft

Drawbacks and Challenges

- The first challenge we have encountered in our project is the landing of the aircraft. During most of the flight we had, the plane ended in crash landing.
- As the radio control (RC) device was being used, we often experienced the problem of going out of the range, whenever this happened, the plane's power supply would get disconnected.
- The size of the propeller and rating of the motor were complex things to choose. In most cases, they were creating raft in the flight of the plane.
- The size of the rechargeable Lipo battery slightly shifts the center of gravity of the whole craft to the front.
- Switching was tough as the auxiliary pin of the receiver was not showing any output.
- We had to use another new body for the aircraft since our previous plane got ruined.

Future Scope

- This autopilot technology can be developed into a quadcopter or drone that can be used to supply reliefs to the calamity-affected areas, in rough weather.
- Object handling capability can also be added so that, risky tasks for human can be controlled by it.
- As a future initiative, we have a plan to participate in the next year's Techfest, at IIT, Mumbai, in their respective segment.