AUTOPILOT CONTROL SYSTEM

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.

INTRODUCTION

Overview:

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 Fig 1.1. 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.

Fig 1.1 - Angles of Rotation

History:

The first aircraft autopilot was developed by Sperry Corporation in 1912. Lawrence Sperry demonstrated it two years later in 1914, and proved the credibility of the invention by flying the aircraft with his hands away from the controls and visible to onlookers. The autopilot connected a gyroscopic heading indicator and attitude indicator to hydraulically operated elevators and rudders. It permitted the aircraft to fly straight and level on a compass course without a pilot's attention, greatly reducing the pilot's workload.

The autopilot control systems have evolved drastically since the turn of the century. Modern autopilots use computer software to control the aircraft. The software reads the aircraft's current position, and controls a flight control system to guide the aircraft. In such a system, besides classic flight controls, many autopilots incorporate thrust control capabilities that can control throttles to optimize the air-speed, and move fuel to different tanks to balance the aircraft in an optimal attitude in the air. Although autopilots handle new or dangerous situations inflexibly, they generally fly an aircraft with a lower fuel-consumption than a human pilot.

Problem Defination

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 Kalman filter 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 in Fig 2.1 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 controller also includes the LCD interfacing unit, the memory card interfacing unit and the computer interfacing unit. The interfacing units are for transferring the log data into the computer and memory card. The data is taken into the computer for calibration of the input devices. The in-flight data is continuously logged into the memory card on board. An LCD panel is used for displaying the angle values.

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

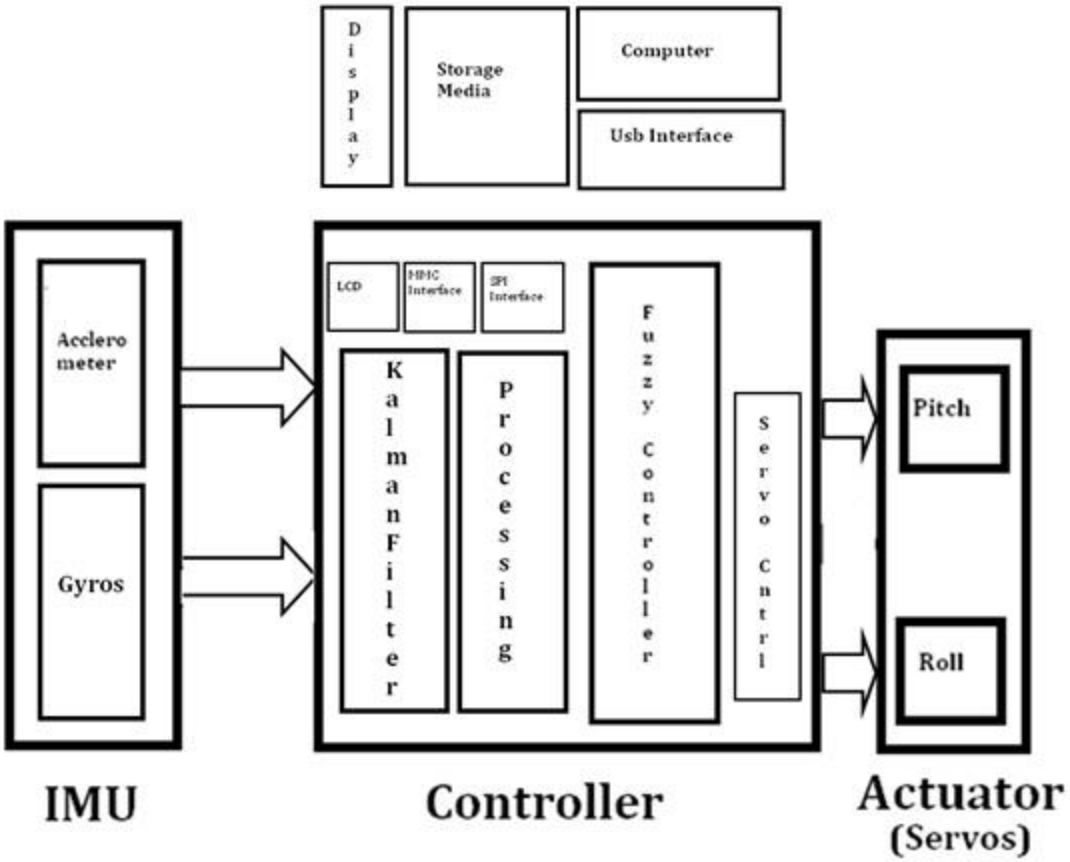

Fig 2.1 - Block Diagram

INERTIAL MEASUREMENT UNIT

An inertial measurement unit, or IMU, is the main component of inertial guidance systems used in air, space, and watercraft, including guided missiles. They measure inertial acceleration, also known as G-forces. An IMU works by sensing motion — including the type, rate, and direction of that motion — using a combination of accelerometers and gyroscopes. The data collected from these sensors allows a computer to track a craft's position, using the method known as dead reckoning.

An IMU works by detecting the current rate of acceleration, as well as changes in rotational attributes, including pitch, roll and yaw. This data is then fed into a computer or a microcontroller, which calculates the current speed and position, given a known initial speed and position.

A major disadvantage of IMUs is that they typically suffer from accumulated error. Because the guidance system is continually adding detected changes to its previously-calculated positions, any errors in measurement, however small, are accumulated from point to point. This leads to 'drift', or an ever-increasing difference between where the system thinks it is located, and the actual location.

The IMU of our autopilot system includes a 3-axis accelerometer along with 3 gyroscopes. A combination of values from these devices is used as input.

An accelerometer measures the acceleration it experiences relative to freefall. Single- and multi-axis models are available to detect magnitude and direction of the acceleration as a vector quantity, and can be used to sense orientation, vibration and shock. This measurement is equivalent to inertial acceleration minus the local gravitational acceleration, where inertial acceleration is understood in the Newtonian sense of acceleration with respect to a fixed reference frame, which the Earth is often considered to approximate.

For the purpose of finding the acceleration of objects knowledge of local gravity is required. This can be obtained either by calibrating the device at rest, or from a known model of gravity at the approximate current position. We are calibrating the accelerometer by taking sample readings at rest and then finding the error offset.

Conceptually, an accelerometer behaves as a damped mass on a spring. When the accelerometer experiences an external force such as gravity, the mass is displaced until the external force is balanced by the spring force. The displacement is translated into acceleration.

The accelerometer we have used in our project is based on micro electro-mechanical systems (MEMS). These chips are the simplest MEMS devices possible, consisting of little more than a cantilever beam with a proof mass (also known as seismic mass). Under the influence of external accelerations the proof mass deflects from its neutral position. This deflection is measured in an analog or digital manner. Most commonly, the capacitance between a set of fixed beams and a set of beams attached to the proof mass is measured. 3 such devices are integrated perpendicularly to form the 3-axis accelerometer.

The accelerometer we have used for this project is from Freescale Semiconductors. It is a MEMS device which gives an analog voltage output proportional to accelerations along its different axis. Fig 3.1 is a snapshot of the accelerometer used along with its evaluation board.

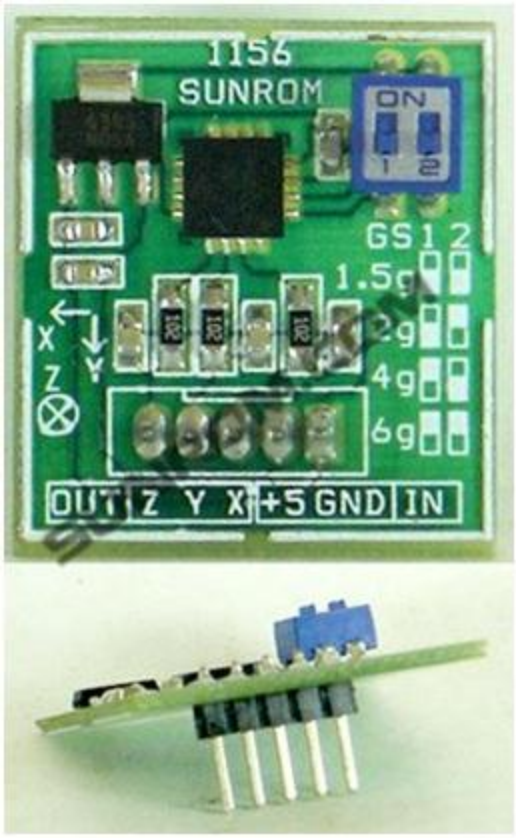

Fig 3.1 - MEMS Accelerometer

Determination of Orientation from acceleration data

When there is no other acceleration other than normal g-force, it is easy to determine the orientation from the acceleration data since the acceleration measured along each axis would be a component of this acceleration. The various required angles can be then calculated as:

$$\text{Pitch} = \tan^{-1}\left(\frac{A_x}{\sqrt{A_y^2+A_z^2}}\right)$$

$$\text{Roll} = \tan^{-1}\left(\frac{A_y}{\sqrt{A_x^2+A_z^2}}\right)$$

$$\text{Theta} = \tan^{-1}\left(\frac{\sqrt{A_x^2+A_y^2}}{A_z}\right)$$

Algorithms for division, determination of square root and calculation of $\tan^{-1}x$ were implemented. The resultant angles were calculated.

The accelerometers generally give high precision measurements. But these are significantly noisy. The angle tilts which obtained from the acceleration values are not very reliable. So in conjunction with these accelerometers, gyroscopes is also used which gives much smoother values.

Gyroscope

A gyroscope is a device for measuring or maintaining orientation, based on the principles of angular momentum. The traditional form of a gyroscope is a spinning wheel or disk whose axle is free to take any orientation. This orientation changes much less in response to a given external torque than it would without the large angular momentum associated with the gyroscope's high rate of spin. Since external torque is minimized by mounting the device in gimbals, its orientation remains nearly fixed, regardless of any motion of the platform on which it is mounted.

We used MEMS gyroscopes available in the form of microchips for angular measurements. Since we require measurements in all 3 axes and with the gyroscopes being single axis, we used 3 such devices. The analog outputs from these were taken into the microcontroller for sampling. The following is an account on the single-axis gyroscope which we used in our project.

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

Fig 3.2 is the breakout board of the gyroscope used.

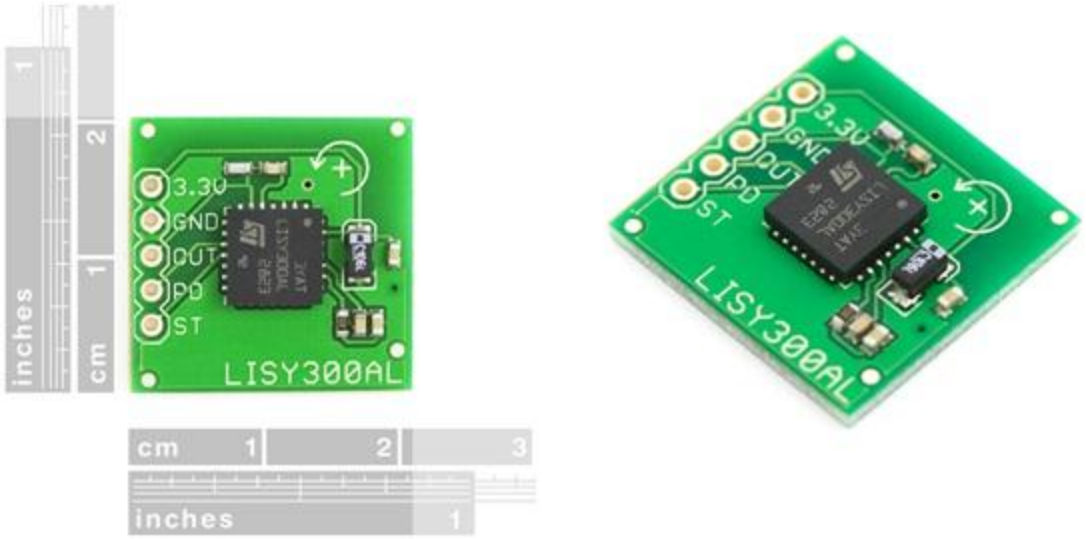

Fig 3.2 - Gyroscope with evaluation board

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.

Calibration of input devices

Accelerometer and gyroscopes were calibrated. The data from the accelerometer for various orientations were recorded into the memory card. About 170 samples were recorded for each orientation. This data was analyzed on the PC using a visual basic program. The mean and variance (as a measure of noise) of each accelerometer reading was determined. Few screenshots of these readings are given in Fig 3.3(a) and Fig 3.3(b).

These readings were then used to determine the offset for each axis.

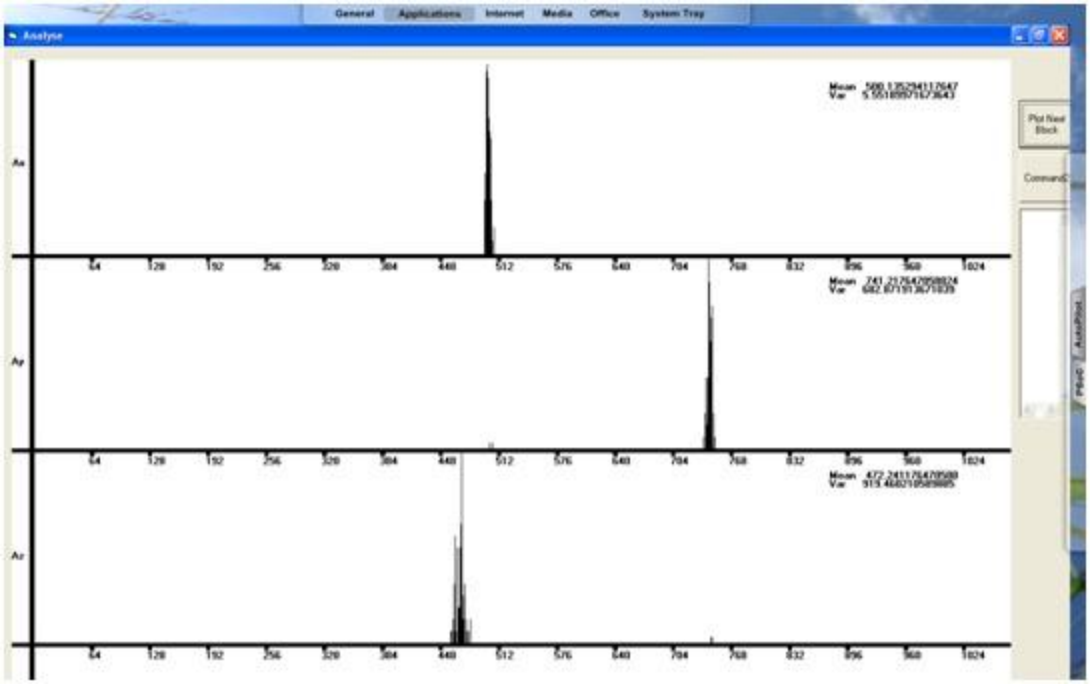

Fig 3.3(a)

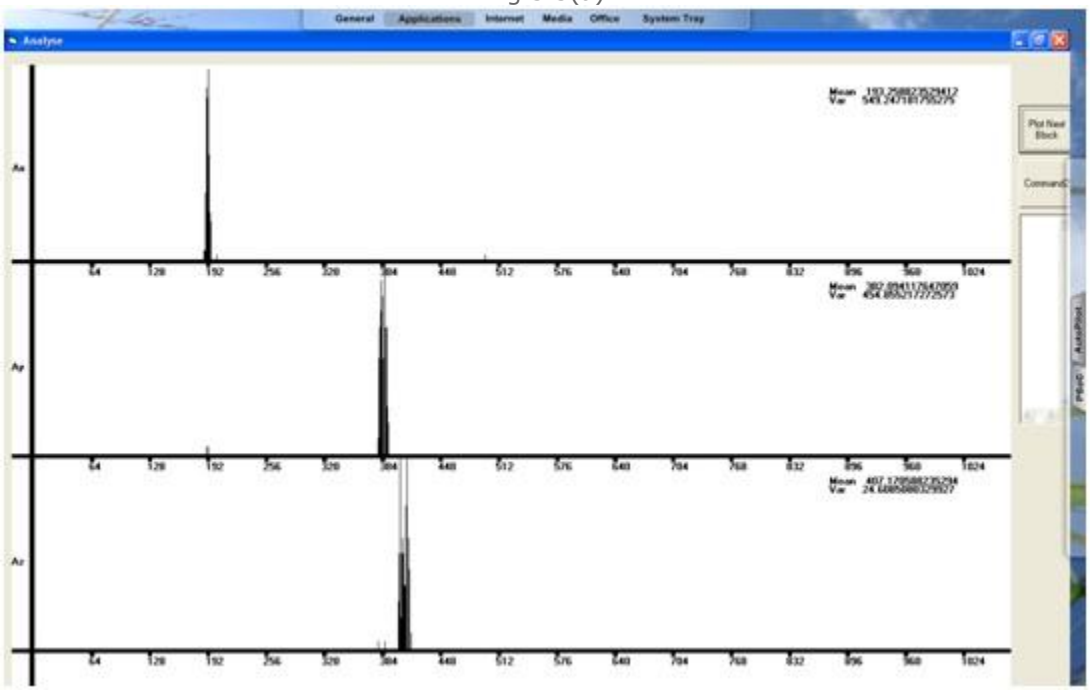

Fig 3.3(b)

The following screenshot shows the data logged in from the gyroscope for calibration purposes:

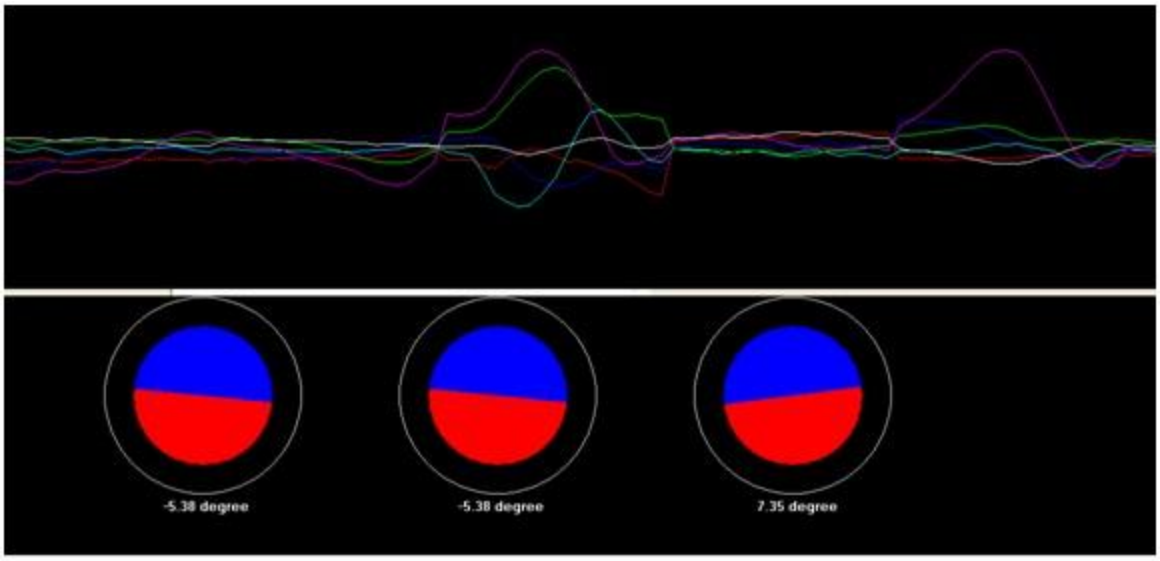

Fig 3.3(c)

CONTROLLER

The data from the inertial measurement unit is taken into the controller for processing. The input being analog requires to be passed through an ADC before being processed. Kalman filter is used for the estimation of the states and then fuzzy controller comes into picture. The controller also interfaces with the LCD panel, the memory card and also the computer for the logging of data.

ATMEGA32 Microcontroller

The microcontroller used is ATMEGA32. This is because we have previous experience on working with ATMEGA series microcontroller. It has a 32KB flash memory for program storage, 2KB RAM, 8 ADC channels, 3 timers with PWM support (used for servo control). These functionalities are sufficient for the blocks we had to create code. Various modules of codes were implemented for the different blocks (Kalman filter, Fuzzy controller, LCD interfacing etc.).

Kalman Filter

The measured data is noisy and the process of navigation also is not precise. Kalman filter is a way to get a best estimate about the process variables (position i.e. location and orientation of our plane)

from these noisy measurements. It is an efficient recursive filter that estimates the state of a linear dynamic system from the series of noisy measurements. A recursive estimator means that only the estimated state from the previous time step and the current measurement are needed to compute the estimate for the current state.

The following basics equations (4 - 8) were implemented in the code:

$$\bar{\mathbf{x}} = \Phi \hat{\mathbf{x}} \quad \textbf{(Equation 1)}$$

Where Ø the state transition matrix and x is the state variable

$$\bar{\mathbf{P}} = \Phi \mathbf{P} \Phi^{\mathbf{T}} + \mathbf{Q} \quad \textbf{(Equation 2)}$$

Where P is the covariance matrix

$$\hat{\mathbf{x}} = \bar{\mathbf{x}} + \mathbf{K}(\mathbf{y} - \mathbf{M}\bar{\mathbf{x}}) \quad \textbf{(Equation 3)}$$

Where K is the Kalman filter gain

$$\mathbf{K} = \bar{\mathbf{P}}\mathbf{M}^{\mathbf{T}}(\mathbf{M}\bar{\mathbf{P}}\mathbf{M}^{\mathbf{T}} + \mathbf{R})^{-1} \quad \textbf{(Equation 4)}$$

Where M is the measurement matrix

$$\mathbf{P} = (\mathbf{I} - \mathbf{KM})\bar{\mathbf{P}}(\mathbf{I} - \mathbf{KM})^{\mathbf{T}} + \mathbf{KRK}^{\mathbf{T}} \quad \textbf{(Equation 5)}$$

Fuzzy Controller

A fuzzy control system is a control system based on fuzzy logic - a mathematical system that analyzes analog input values in terms of logical variables that take on continuous values between 0 and 1, in contrast to classical or digital logic, which operates on discrete values of either 0 and 1 (true and false).

Since in autopilot we require the control system to behave similar to a human response we planned to use fuzzy logic. It has the advantage that the solution to the problem can be cast in terms that human operators can understand, so that their experience can be used in the design of the controller. This makes it easier to mechanize tasks that are already successfully performed by humans.

The following is an overview of the fuzzy control system. They consist of an input stage, a processing stage, and an output stage. The input stage maps sensor or other inputs, such as switches, thumbwheels, and so on, to the appropriate membership functions and truth values. The processing stage invokes each appropriate rule and generates a result for each, then combines the results of the rules. Finally, the output stage converts the combined result back into a specific control output value. These stages are respectively referred to as fuzzification, fuzzy inference with knowledge base and defuzzification. Fig 4.2 shows these stages as block diagrams.

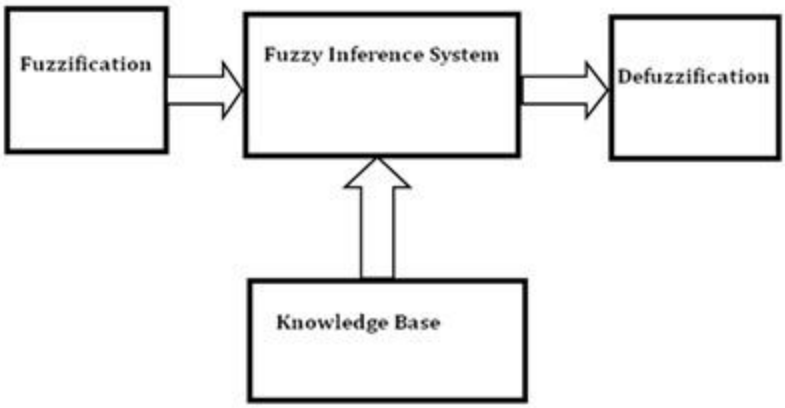

Fig 4.2 - Fuzzy Controller in blocks

We implemented the basic requirements for a fuzzy controller in assembly code for ATMEGA32. Triangular transfer function was used as membership functions. Codes were written and implemented in the microcontroller.

Output Interfacing Units

The system has to be interfaced with the peripherals. Code is written in the microcontroller for these interfacing modules. The modules include LCD interface, memory card interface, and computer interface. Servo control also needs to be done with the microcontroller.

LCD Interface

This is to have an output device to print out variables values and messages during test runs of the code. This interface would also be used to display messages from the autopilot later in the design. It was tested on hardware and is working satisfactorily.

Memory Card Interface

While programming for the accelerometers we came to the conclusion that we need to have to save the data from the accelerometer over a run to later work with it and study both its output (and thereby calibrate) and also working of our controller by saving its parameter and variable values during the running of the code. We needed a storage device to store large amount of data.

Of the two alternatives for a light weight but sufficient memory capacity (USB pen drives and MMC/SD card used in mobile phones) we decided to use MMC cards since they are lighter and easy to interface. This interface was created and tested on hardware. MMC cards use an SPI interface with some commands for transfer of data. We implement FAT32 file system so that we can read our data on any operating system (We are now using windows XP). We can now dump any amount of data (up to 4 GB based on capacity of card) for later analysis.

Computer Interface

Interfacing with the computer enables the data logging of accelerometer and gyroscope which is important for calibration purposes. To calculate the offset errors in the accelerometer and gyroscopes, we take a set of sample values and the error covariance is calculated in the computer. This shows how

much reliable the accelerometer and gyroscope readings could be and thus we can set the weightage for both sets of values. USB interface is the mode of transfer of data between the microcontroller and the computer .The code has been implemented for the USB protocol.

Servo Control

This is implemented to control the servo motors to affect the change in path specified by the processors (fuzzy controller) to maintain the desired orientation. This basically involves enabling the timers and generating appropriate PWM signals for the required angle.

Since the microcontroller doesn't have required number of timers appropriate for servo control we have implemented it in software. We get an accuracy of about .1 degree resolution for each servo.

ACTUATORS

The change in path has to be brought into action using a set of motors and we are using a set of servo motors for that purpose. The processing unit outputs the change in path required to maintain the desired path. This is communicated to the servos through the servo control unit.

Servo Motors

The motors we used for testing were Futaba S3003 model servos.

Fig 5.1 Servo motor

Conclusion

An autopilot control system was successfully designed, implemented and developed. The inertial measurement unit comprising of the accelerometer and gyroscope were calibrated and mounted on the glider. These devices measure the acceleration and tilt of the glider plane and log the data onto the onboard memory card. The deviation from the desired path and orientation, calculated by the central processor, was corrected by the servo motor set which maintained the tilts of the plane at desired levels. In-flight data is also available for analysis since it was logged onto the memory card.

Future scope of work

Processor limitations restricted the Kalman filter implementation to one-dimensional only. Because of the same limitations, in the processing stage of the fuzzy controller, the rules were based on one

dimensional values only (either pitch or roll or theta and not a combination of 2 or more of these angles).

These limitations can be removed by upgrading the controller and the software part. The basic hardware can remain the same. So we have the essential setup to upgrade the system to a fully fledged autopilot in the future.

Source: http://www.botskool.com/tutorials/fourth-year-projects/autopilot-control-system