***I.J. Engineering and Manufacturing*, 2018, 6, 26-41**
Published Online November 2018 in MECS (http://www.mecs-press.net)
DOI: 10.5815/ijem.2018.06.03

Available online at http://www.mecs-press.net/ijem

Angle Measurement on a flat Surface using High Frequency Ultrasonic Pulse

Shashi Suman

Department of Electronics and Communication, JECRC University, Jaipur 303905, India

Received: 29 May 2018; Accepted: 16 October 2018; Published: 08 November 2018

Abstract

Ultrasonic waves are most commonly used to measure the presence of an object and its distance from the source using time of flight concept. These are pulses of sound waves that have frequency range higher than the human hearing range. In this paper, we will discuss the measurement of the tilt angle of a robot with respect to a flat base using ultrasonic waves and time of flight concept [2]. An Arduino platform was used with Atmel328P as the processing microcontroller chipset which will then compute the angle of tilt using the distance calculated from an ultrasonic sound transmitter and a receiver using coordinate geometry and trigonometric functions. A combination of gyroscope and accelerometer is also used to find the true tilt angle of the apparatus and then it is compared with the angle readings obtained from the ultrasonic sensor system. A high response time and low delay is necessary for instantaneous angle measurement. Hence, gyroscope-based angles have been used as a reference to adjust filter parameters to decrease error and noise at every iteration.

Index Terms: Tilt Angle**,** Ultrasonic, Distance, Arduino, Kalman, Exponential.

© 2018 Published by MECS Publisher. Selection and/or peer review under responsibility of the Research Association of Modern Education and Computer Science.

1. Introduction

Embedded systems like the self-balancing robot or the inverted pendulum have a common aim that is to balance itself. To balance itself, the processing unit must know whether it is falling to either side and then accelerate towards the direction of the tilt. The system must have an accurate control of the acceleration to prevent the bot from falling on the other side. This is usually done using an accelerometer that detects acceleration forces. The translational motion and natural static acceleration of earth help the robot to determine the angle of tilt which in turn helps to determine the power required for the motors. We generally measure the

*Corresponding Author. Tel: +91 9304620643
E-mail address: shashisuman17@aol.com

tilt angle in robots by using a gyroscope or an accelerometer. The combination of a gyroscope and accelerometer with a proper proportion of low-pass and high -pass filter makes it possible to measure the tilt angle of robots. For small robots that run on flat base or platforms, we can use basic trigonometric functions to determine the angle.

This would require accurate measurement of the length of all axis from a selected position which will be used to determine any change in the state of the robot. To measure this distance researchers often use optical sensors, ultrasonic sensors, LIDAR sensors, Lasers, etc.

To remain on the budget side so that young researchers can learn and work, we can rely on sensors like an infrared sensor or an ultrasonic sensor. However, there are issues with every sensor present around which is noise. It is impossible to remove noise as it entirely depends upon resolution and quality of sensors as well as surroundings which will directly affect the response time and behavior of the robot. This can also cause delay in response which will lead to improper functioning and failure of any system.

In inverted pendulums, fusion of accelerometer and gyroscope is used to measure tilt angles. It becomes very important to use filters to avoid unwanted signals and spikes. For these robots, low pass filter is used for accelerometer-based angle and high pass filter is used for gyroscope-based angle. Combination of both angles gives a true filtered angle although heavy filtering of a raw signal will remove almost all noise, but its response time will be low. On the other hand, a responsive filer can be used, but will also include noise. With research in the past years, several filtering techniques have evolved like the Kalman filter by Rudolf E Kalman [4] who was the first person to develop this filter based on linear quadratic estimation. Another filter developed and modified by Holt is the exponential filter which is somewhat based on a recursive technique that helps in smoothing a noisy signal. These filters showed significant results than the moving average and average filtering techniques. Young researchers that work in the controls field have used sensors like MPU-6050 [5] through a complementary filter for measurement and digital motion processing itself on the sensor.

In this paper, we will work with the Kalman filter and the exponential filter. Based on the estimation of future values from the covariance matrix, Kalman filter can be used to filter the data of the ultrasonic sensor and give us the true distance from a reference object. Similarly, the exponential filter can help to smooth out the output. We can generate ultrasonic sound pulses at a frequency of 40 kHz and constantly measure the distance from a flat base. When there is a change in the system by revolving the sensor around the perimeter of a circular section, we can measure the difference in the distance with accuracy and then measure the change in the angle. The angles are then matched with the angles measured by MPU-6050 module to determine the accuracy. The measurements were taken in a surrounding with constant humidity. Hence, we have ignored the effect of humidity in the noise. Ultrasound waves when traveling through the air has its own errors. These sonic waves are suitable for detecting the presence of objects even though it is far, but the distance measured is an approximate distance. Farther is the object, more noise and error will be present. We also know that the speed of sound depends on the temperature as well as moisture. Thus, measuring long distance with low-resolution ultrasonic sensor poses a problem in the measurement of angles slightly. Close distance measurement reduces the noise as the amount of time required to travel through air is lessened.

Nomenclature	
R	Perpendicular distance (Fixed)
D	Distance between sensor and support shaft
I2C	Inter Integrated Circuit
S	Speed of sound in air
T	Time between transmission and reception
Temp	Temperature in Centigrade
X	Variable distance between source and target
A_x	Acceleration along X-axis
A_y	Acceleration along Y-axis
G	Angular Velocity

dt	A very small duration of time
SCL	Serial Clock Line
SDA	Serial Data Line
D_n	Exponential Filter Output
D_{n-1}	Exponential Filter Previous Output
β	Exponential Filter Weight
α	Complementary Filter Weight
θ_U	Instantaneous Angle measured by Ultrasonic Sensor
θ_C	Instantaneous Angle measured by Gyroscope and Accelerometer fusion
θ_{C-1}	Previous Angle measured by Gyroscope and Accelerometer fusion

2. Proposed Methodology

The tilt angle of a robot can be measured by various ways. We can use accelerometer which measures acceleration along each axis. However, accelerometer measurement varies for longer durations. Gyroscopes completely nullifies this drawback and shows accurate readings for longer durations however has variations in short durations. Gyroscopes and accelerometer when combined can measure the angle irrespective of surface orientation. Another technique to measure the angle is by using infrared sensors. However, infrared sensors are limited to the surrounding light intensity.

For this paper, ultrasonic waves-based sensor was used to measure the change in the distance which could be used to deduce the angle of rotation. The use of trigonometry on any polygon for angle calculation between its sides, we require the length of its sides. We can say that the distance between consecutive vertexes of the polygon, if known can help to deduce angles between various sides. Hence, to measure the tilt we can imagine the robot moving in a translational motion. For angle calculations, horizontal changes are not required as the sensor has a fixed distance from the body. We only need vertical changes in the coordinates of the robot. By measuring vertical changes in the distance between the sensor and the base, we can get all parameters for angle calculation. The readings calculated in real time must be passed through appropriate filters to remove noise and then we can measure the tilt angle with smooth output. To measure the distance from an object in robotics field, ultrasonic sensors are used most of the time. We can consider the SONAR as an example which is an instrument based on ultrasonic sound waves that are used on ships. These sensors come with various frequencies ranging above the human audible range. Distance measured from an object is based on Time of Flight concept which states that distance is the product of time and speed. The perpendicular distance is termed as R. The distance between the corner joint of the arm holding the sensor and the center point of the transmitter and the receiver is termed as D as shown in Fig. 1. This distance is very crucial and must be accurate. The temperature can be calculated using module MPU-6050 [5] that works on I2C communication protocol which returns the angular velocity, acceleration of each X, Y and Z axis along with the temperature. The temperature parameter is also responsible for the speed of sound in the air.

The sound waves emitted from the transmitter of the ultrasonic sensor travels towards the subject. After striking the subject, waves get reflected. After receiving the reflected waves, the echo pin of the sensor goes high. The distance [2] covered is twice the distance between the source and the object. In this paper, the distance between the sensor and the support shaft was kept fixed at 8.9 centimeters.

Since the waves travel twice the distance, a scalar factor of 0.5 is multiplied. The basic advantage of using an ultrasonic sensor over an optical sensor in this paper is to reduce variations in readings due to surrounding light conditions. As light intensity may change, it will change the output values of the optical sensor.

The speed of sound in the air can be measured using the following expression

$$S = (331.3).\left[\sqrt{1 + \frac{Temp}{273.15}}\right] \tag{1}$$

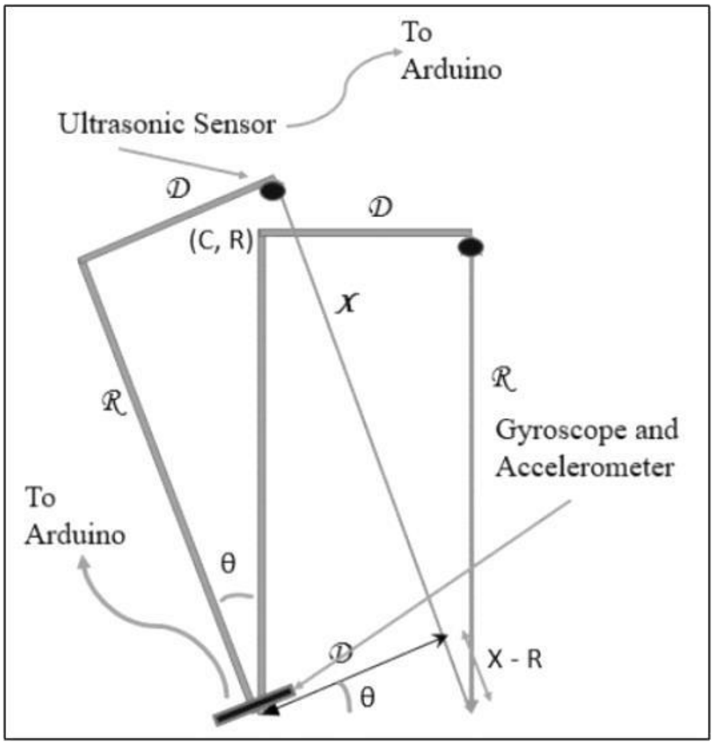

Fig.1. Coordinate System Representation of Setup

The change in the distance is measured when there is a change in the position of the ultrasonic sensor along the curvature.

With reference to the Fig. 1, the tilt angle can be found by the following equation

$$\theta_U = \tan^{-1}\left(\frac{X - R}{D}\right) \tag{2}$$

The ultrasonic waves of the sensor travel double the actual distance. The waves from the sensor include noise and disturbance from air, temperature and humidity as well. We will consider the temperature in speed measurement for this paper. The variable distance [2] between sensor and base can thus be measured by the following equation

$$X = (165.65).\left(\sqrt{1 + \frac{Temp}{273.15}}\right).T \tag{3}$$

The accuracy of angle measured by the ultrasonic sensor depends solely on the resolution of transmitter and receiver, and surrounding disturbances. As the distance between sensors and base surface increases the error in the readings also increases. The minimum and maximum distance that can be measured by the ultrasonic sensor is limited. With a large increase in the tilt angle, the error is also increased because of greater amount of time travel required for the ultrasonic waves to measure the distance.

3. Filters

A filter is a software or hardware-based process that helps to eliminate noise or pass a certain bandwidth of a signal. If we consider a spike, then only its upper peak values would be passed through a high pass filter. If passed through a low pass filter, only its lower peak values would be passed. For noisy signal filtration, smoothing [6] is used where any significant change in the value is flattened out and removed. Smoothing of a signal depends on the smoothing factor. Greater is the smoothing factor, smoother is the wave giving more priority to the historical values. On the other hand, with a small smoothing factor, immediate values are given more priority.

For this paper, an appropriate fusion of low pass and high pass filter is required. The complementary filter is an example of fusion used for angle measurement using gyroscope and accelerometer. For ultrasonic sensor reading, we used exponential filter. Initially, an average filter [7] was applied to the readings from the ultrasonic sensor. However, the average filter [7] has a disadvantage. The time taken to filter out the data is large which is not suitable for our requirements. With the increased number of values, complexity also increases. In terms of image processing, it does not help to preserve the edges. Thus, with the average filter, any sudden change in the readings is neither eliminated nor prioritized but an average value is taken which in turn helps to trace out the raw signal more closely as shown in Fig. 2. The response time was low, and the smoothness of the curve remains low maintaining a very slight deviation from the raw signal.

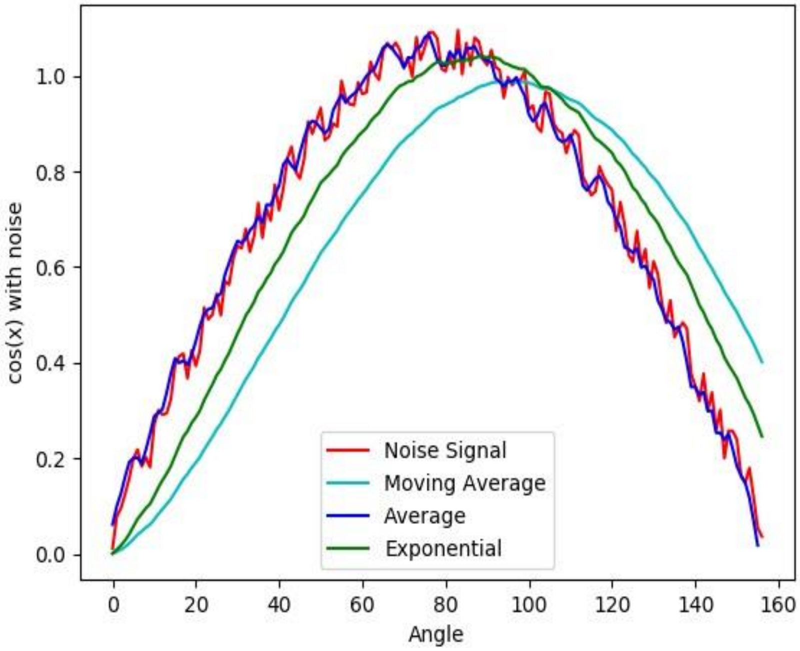

Fig.2. Comparison of Signal, Moving Average, Average, Exponential Filter

The second filter used to remove noise is the moving average filter [7] which takes M values at an instant from the input readings and then calculates the average to produce a single value. Higher is the value of M, higher is the complexity which in turn also increases the filtration of a noisy signal, thus producing a smooth signal.

However, to take these samples, time complexity is involved which increases the lag of this filter thus didn't suffice our requirements of quick response with less delay. From Fig. 2, we can deduce that there is an initial delay between moving average filter and the raw signal. Although, it has a smoother output when compared with the average filter but also takes more time to process.

The third filter applied to remove the noise was the exponential filter [10] which has a slightly higher response time than the average filter but had a lower response time than the moving average filter. Higher is the response time, slower will be the filter output when compared to MPU-6050 [5] based angles. Thus, a delay will always be present. The mathematical form of the exponential filter can be represented by equation 4 below.

$$D_n = \beta . D_{n-1} + \left(1 - \beta\right) . \theta_U \tag{4}$$

With this filter, we were able to get a smooth filtered output with high noise immunity. In the equation 4, past values and instantaneous value are complement of each other. For this filter, the weight of the filter must be set precisely. It must remain between 0 and 1. Setting the weight to 1 will flatten out the readings as initial value of the filter will be zero and no proportion of the instantaneous angel be given any priority. On the other hand, setting the weight to 0 will give priority to only instantaneous measured angles and no proportion from the previous filtered angle will be added. This can give a responsive curve but only when the sensor is accurate enough with high noise immunity and high resolution. Since, this is not practical for analog sensors, we cannot use 0 or 1 as the weight of an exponential filter.

The final filter we used for approximation and smoothing of ultrasonic readings was the Kalman [8, 12] Filter to maintain fast response with very low delay and low difference from the readings of the MPU-6050 [5] sensor. The sensor returns angular velocity and acceleration along each axis after which inverse trigonometric functions can be used to deduce the angle. The deduced angles are then passed through the filter. The prediction of next values by the Kalman [8, 12] filter can be manipulated by changing the filter parameters which in return changes the response and smoothness of the quantity. The experimentation was done using the Simple Kalman Filter [8] library.

There was a consistent noise involved in all the experiments generated from the servo motor and surroundings. Hardware filters can also be used for filtering the servo noise which can further reduce the errors involved in the distance measurement.

4. Experimental Setup

For this paper, we will use an Arduino [3] Uno development board to compute the mathematical part. The microcontroller on the development board runs at the frequency of 16 megahertz. The microcontroller embedded on the board is an Atmel 328P IC [1]. The Arduino [3] is powered by a 7-volt power source. Power was supplied with a jack which is connected to a power regulator on board. The onboard regulator drops the input voltage by approximately 2 volts and the minimum power requirement for the IC operation is 5 volts. Every pin on the Arduino can provide a maximum current of 40mA and a voltage of 5 volts. The analog ports of Arduino span across 5 volts into 1024 steps from 0 to 1023. Each value is controlled by a digital to analog converter to get the required voltage.

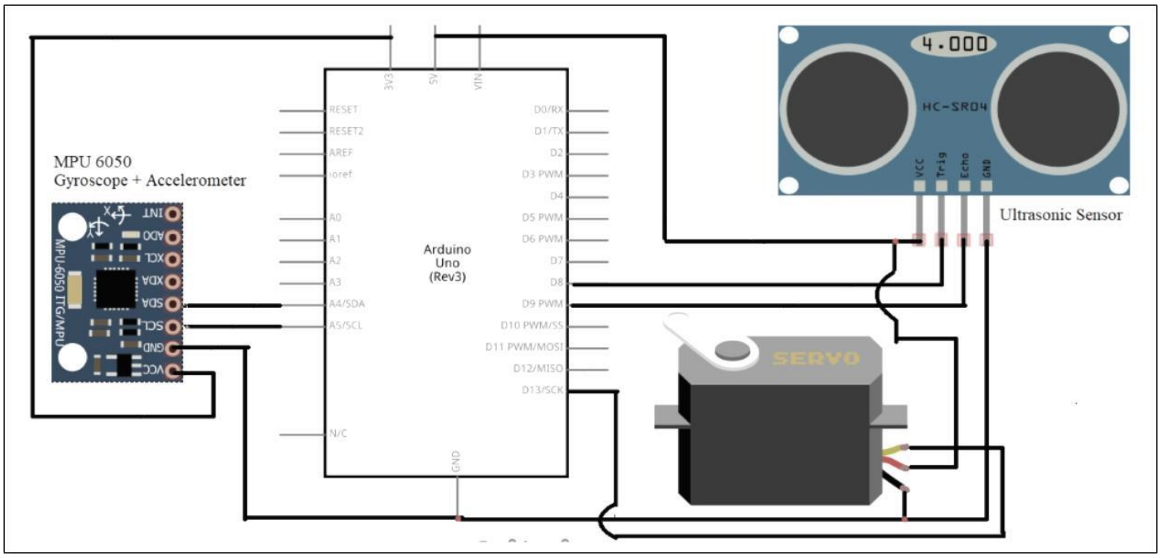

Fig.3. The connection of MPU6050, Ultrasonic Sensor and servo with Arduino Uno

4.1. Ultrasonic Interfacing

The ultrasonic sensor requires minimum voltage of 5 volts which can be provided with the Arduino. The trigger pin was connected to digital pin 8 and the echo pin was connected to digital pin 9. Since we must continuously switch the states of the trigger and echo pin, therefore, the digital port was necessary. This fast switching is managed by the Arduino itself. The switching of signals having frequency of 40 kHz is managed by the oscillator on the ultrasonic sensor. The transducer embedded on the ultrasonic sensor is programmed for 8 pulses. To begin the operation of the sensor, the transmitter pin or the trigger pin is held high for 10μS by setting the digital pin 8 to high. Within this time, the amplitude of the voltage is stabilized for proper operation. The stabilization helps in the tuning of transmitter and receiver to the fixed frequency. The transmitter embedded then sends 8 pulses of ultrasonic burst. After the transmission, the trigger pin is pulled low by setting digital pin 8 to low. Without delay, the receiver pin or the echo pin is pulled high by setting digital pin 9 high. The receiver is separated by 2.8 centimeters from the transmitter. It then waits for the pulses sent by the transmitter. The width of the signal on the echo pin is directly proportional to the distance between the base and the source. When the receiver receives the pulses, time difference is calculated for which the echo pin is kept high. With the velocity of the sound in the air calculated from equation 1 along with the time, the distance can be deduced.

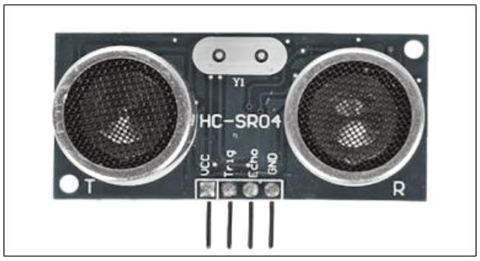

Fig.4. Ultrasonic Sensor with pin outs (Left to Right: Vcc, Trigger, Echo, Ground)

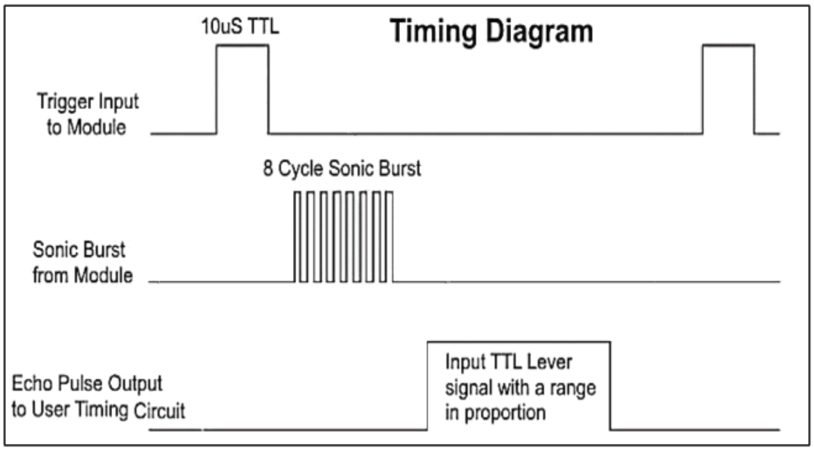
Timing Diagram
10uS TTL
Trigger Input to Module
8 Cycle Sonic Burst
Sonic Burst from Module
Echo Pulse Output to User Timing Circuit
Input TTL Lever signal with a range in proportion

Fig.5. Timing Diagram of Ultrasonic Sensor

4.2. Gyroscope and Accelerometer Interfacing

MPU-6050 [5] module is a combination of 3 axis accelerometer along with 3 axis gyroscopes embedded together with a temperature sensor. It operates through I2C communication through Arduino. The analog pin (A4) is specifically set for SDA line and the analog pin (A5) is specifically set for SCL line on Arduino to connect any I2C device. The module is powered with a 3.3V power supply through Arduino. With 5 volts, the sensor used to freeze many times during operation. The sensor stores its data within 14 registers. Each register returns 8-bit data. To communicate with the device, we require the device address. For the MPU module, 0x68 is register address. This address can be retrieved by accessing WHO_AM_I register with address 0x75. When the bits are received, they are incomplete. To retain information, the first 8 bits are shifted left 8 times and logical OR operation is performed with the next 8 bits. These 16 bits of data contains information about acceleration, temperature or angular velocity.

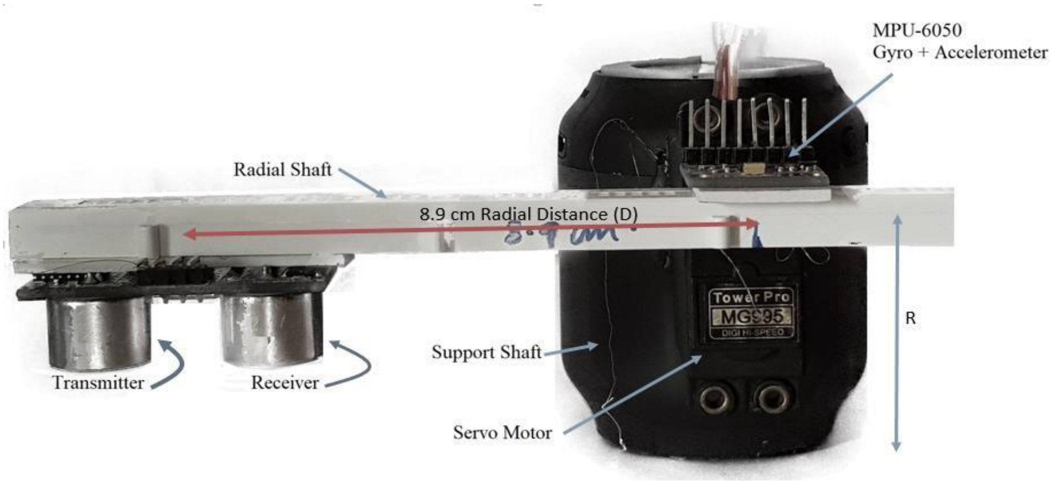

Fig.6. Experimental Apparatus for Angle Measurement

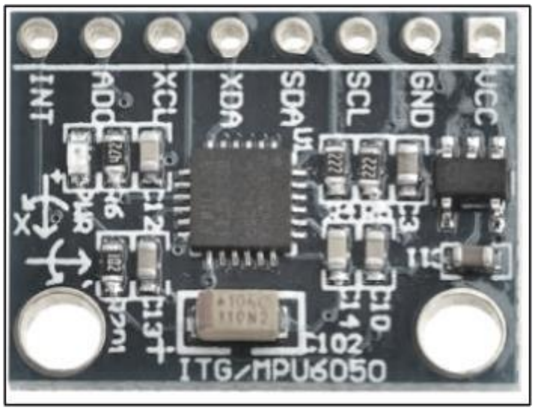

Fig.7. MPU-6050 module with Gyroscope and Accelerometer

Table 1. MPU-6050 Registers Address Mapping

Measurable Quantity with Axis	Upper 8 bits	Lower 8 bits
Acceleration X	0x3B	0x3C
Acceleration Y	0x3D	0x3E
Acceleration Z	0x3F	0x40
Temperature	0x41	0x42
Angular Velocity X	0x43	0x44
Angular Velocity Y	0x45	0x46
Angular Velocity Z	0x47	0x48

The module was placed exactly at the pivot on the lower end of support shaft as shown in Fig. 1 and Fig. 6. This reduces the drift in the instantaneous readings from the gyroscope. The support shaft’s pivot point is attached to the servo horn of a metal geared servo which was programmed to change the inclination as per coded in the microcontroller. To calculate the behavior of this system, five different tests were done with different parameters. The weight of the exponential filter was changed for first four cases for better response and smoothness. This helped us to predict the correct parameters which will return accurate results from the ultrasonic sensor.

The accelerometer gives acceleration along each X, Y and Z axis but also includes the rest force which deviates from the real angle. The short-term readings are accurate, but the long-term readings get noisy. The gyroscope returns angular velocity of each axis. The gyroscope gives accurate long-term integrated readings, but the short-term readings are noisy. Thus, the accelerometer and gyroscope readings were combined by a complementary filter according to equation 5 below. Gyroscope’s angular velocity values are passed through high pass filter and the accelerometer’s acceleration values are passed through low pass filter.

$$\theta_C = \alpha.\left(\theta_{C-1} + G\right).dt + \left(1-\alpha\right).\tan^{-1}\left(\frac{A_Y}{A_X}\right) \tag{5}$$

Where α̨ is a floating value between 0 and 1. For general requirement, the value of α̨ was chosen close to 1. Thus, it will behave more as a high pass filter. This complementary filter was used with MPU-6050 [5] module for all cases except for case 3 where an additional exponential filter was used over the complementary for further smoothing [6, 10] of readings MPU 6050 [5]. Kalman filter was also used in the place of exponential filter for filtering ultrasonic readings. Initially, the perpendicular distance i.e. the length of the support shaft was measured by the ultrasonic sensor. This distance was also confirmed with a digital scale reading instrument. With this, we got the constant value of R. Similarly, D was also measured using a digital scale for accuracy. Randomly changing distances were measured by the ultrasonic sensor which were relatively accurate for high angular velocity. This gives us the value of X. The values obtained are then put in equation 2 which will give us the angle that will be passed through the filters.

In this methodology, the base must be completely flat to measure the angle because the ultrasonic sensor readings will differ with a rough base.

The results obtained from the exponential filter showed deviation when the angular velocity was low. A difference in the angle was observed when the shaft was suspended in midair because of the motor vibrations and weight of the shaft. Thus, an advanced filtering algorithm was required. The Kalman filter [8] used in this paper is based on a covariance matrix. Three parameters are passed down to the filter. The first parameter is the measurement variation indicating average differences in the input. The second parameter is the uncertainty value determining the estimated next value from the present state value. The third parameter passed is the covariance matrix indicates, how fast our readings will move. The parameters can be changed according to ours needs. For this paper, 0.01 was chosen as measurement variation as well as uncertainty. The uncertainty value changes with filtering process. The covariance parameter was set to 0.001. The values gave closest reading and prediction of next readings for the ultrasonic sensor.

Table 2. All Five Cases with Different Parameters

Case	Weight of the Filter (Ultrasonic)	Accelerometer and Gyroscope Filter	Filter for Ultrasonic Sensor
1	0.75	Complementary	Exponential
2	0.8	Complementary	Exponential
3	0.8	Complementary + Exponential	Exponential
4	0	Complementary	No Filter
5	Kalman	Complementary	Kalman Filter

Table 3. Arduino Pro Mini Development Specifications

Parameters	Values or Specification
Microcontroller	Atmel 328
Power (RAW)	5 volts to 12 volts
Power (V_{cc})	5 volts
Analog Pins	6 (A0, A1, A2, A3, A4, A5)
Digital Pins	14 (0 to 13)
PWM Pins	3, 5, 6, 9, 10 and 11
Flash Memory	32 kb
SRAM	2 kb
Clock	16 MHz
I2C Pins	A4(SDA), A5(SCL)

5. Results

To gather the accurate parameters required for error free application of the ultrasonic sensor, five different tests were performed, each test with a different weight for the exponential filter [10], complementary and the Kalman filter [8]. These tests were performed to determine the right coefficients for least deviation in the ultrasonic angle from MPU-6050 angle. This will give us an insight about the response time for sudden change in angle, vibration and noise compensation as well.

5.1. Case 1

The weight of the exponential filter was set to 0.75 in the Arduino code. As shown in the Fig. 8, changes were responsive. There was a slight difference when the tilt angle of the apparatus was less. The initial position of the experimental setup was set to ±0.30 degrees for gyroscope as well as ultrasonic sensor measured angles. When the servo motor suspended the support shaft in midair, a slight difference between MPU-6050 angles and ultrasonic angles was observed because of the vibrations in the shaft caused by the backlash error of the motor. This was due to the weight of the shaft holding the sensor. The differences were least when the angular velocity was high because motors dead band and backlash errors are least whenever there is a high change in speed. When the weight of shaft's center of mass was along the center of servo horn, the minimum error was very close to zero. Only backlash errors were present. Gravitational force was cancelled by its own normal reaction force from the pivot.

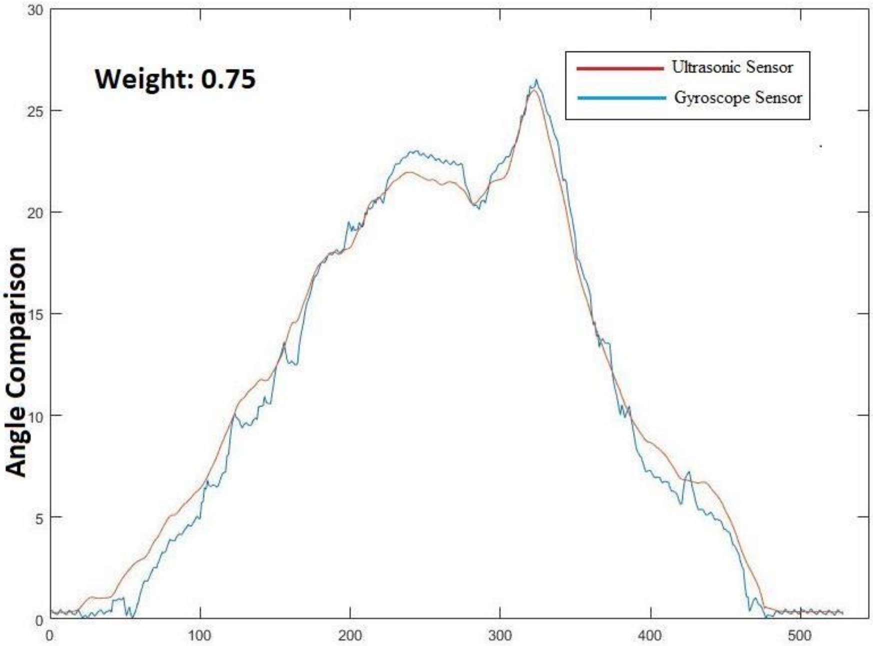

Fig.8. Angle Comparison: Case 1 Weight of Complementary set to 0.75

5.2. Case 2

The weight for the exponential filter was set to 0.8 in the Arduino code and 0.8 for the complementary filter applied on the angles from MPU-6050. The observed response time was high with a major difference in the still angles. The response time was like Case 1. The smoothness of the graph of ultrasonic sensor was smoother than Case 1. The noise in distance measurement were due to the irregularities of the base which was compensated by the filter. When the shaft was suspended in midair by the servo, a slight difference in the angles were observed with a maximum error of 1.1 degrees as shown in Fig. 9 between 300 and 400 milliseconds. The average response time differed from Case 1 by 3.56 %. As shown in the Fig. 9, initially the system involves no motion however, MPU-6050 [5] was picking up the vibrations because of the motor. The filtered ultrasonic angles were successfully able to remove those noises. The noise immunity for Case 2 showed better results than Case 1.

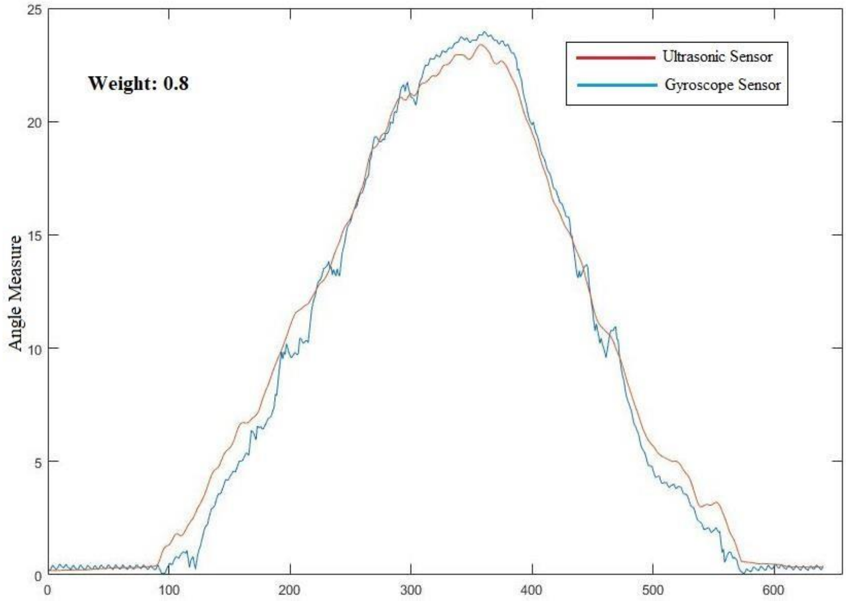

Fig.9. Angle Comparison: Case 2 Weight of Complementary set to 0.8

5.3. Case 3

For Case 3, MPU-6050 readings were passed with dual filtering. Firstly, the raw readings were passed through the exponential filter. The output from the exponential filter were fed as input to the complementary filter. This was done to prevent motor noise from interfering the angle readings. With this change, the backlash error and the dead band error of motor were highly minimized but were not completely removed. The weights for both filters were set to 0.8. The weight of the exponential filter for ultrasonic sensor was also set to 0.8. The response time was quick, however slower when compared to Case 2. The difference in the angles measured were less as compared to Case 1 and Case 2. This minimal difference was visible for high and low angular velocity of the shaft. With the sudden change in angle, which created peaks, the difference in the angle was increased. The difference sustained for a very less time and decreased with time. When the shaft was suspended still, a maximum error of 1.14 degrees was observed at most. Minimum error difference observed was 0 degrees. Stationary angles were mostly suppressed as shown in Fig. 10. For most of the time during operation, ultrasonic angles were able to track the MPU-6050 filtered angles accurately. From Fig. 10 error free angle comparison can be visualized between 400 milliseconds to 500 milliseconds.

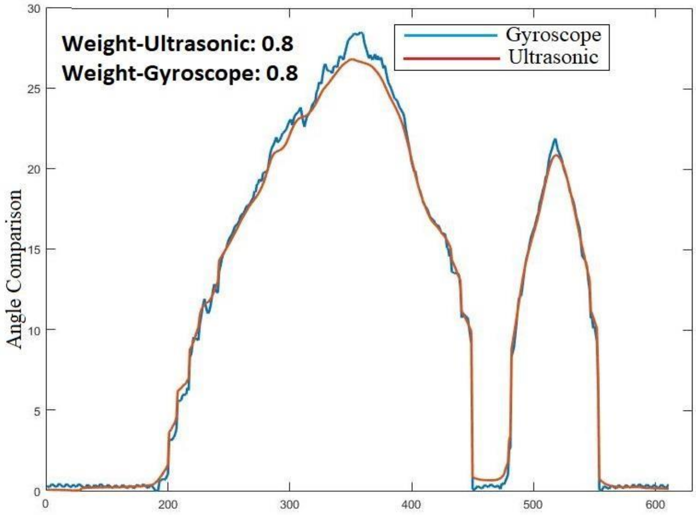

Fig.10. Angle Comparison: Case 3 Weight 0.8 for both sensors

5.4. Case 4

For Case 4, the weight of the exponential filter [10] for the ultrasonic sensor was set to 0 thereby, disabling the filtration of ultrasonic readings completely. No filter was applied to the MPU-6050 module for angle measurement other than the complementary filter. The weight of complementary filter was set to 0.95. The ultrasonic output was not much smooth with occasional deviations from the MPU-6050 angles as shown in Fig. 11. The response time observed was least when compared to all other cases and responded as soon as the gyroscope angles were changed. The maximum deviation for this case was 1.94 degrees and minimum difference of 0.22 degree thus giving a noisy output. This won’t be suitable for robots requiring stable smooth variation of angles as it may lead to erratic behavior.

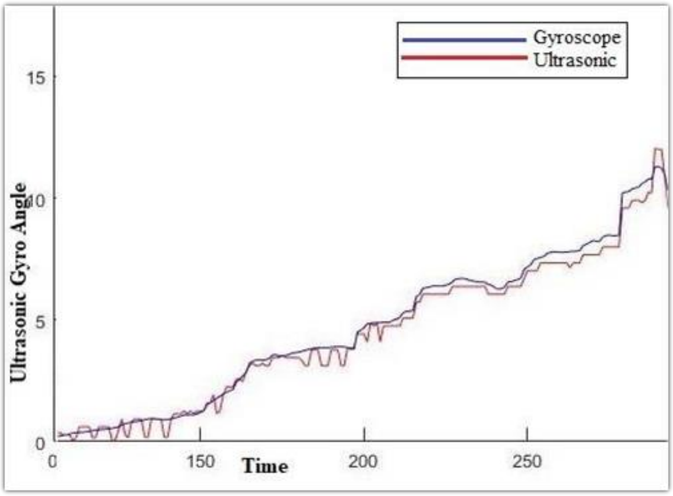

Fig.11. Angle Comparison: Case 4 Weight of Exponential set to 0

The angles from the ultrasonic sensor were found to be fluctuating frequently between short durations. Sudden change with high angular velocity showed less fluctuation and error. However, for medium and low angular velocity, error in the angles were high. In Fig. 11 between 0 to 150 milliseconds, fluctuations are visible for low angular velocity. From 150 milliseconds to 160 milliseconds, the amount of error is less because of high angular velocity.

5.5. Case 5

For Case 5, Kalman filter [4,8] was used for filtering ultrasonic angles. The response time was slightly higher than Case 4 with a maximum difference in angles being 0.85 degrees at peaks. The difference between the MPU-6050 based angles and ultrasonic-based angles were almost consistent with a smooth output. Angle difference remained consistent for high as well as low angular velocity. When the shaft was suspended in midair, a consistent but negligible error was present as shown in Fig. 12 between 0 to 100 milliseconds. It was only because of the motor backlash error which is a physical error of motor.

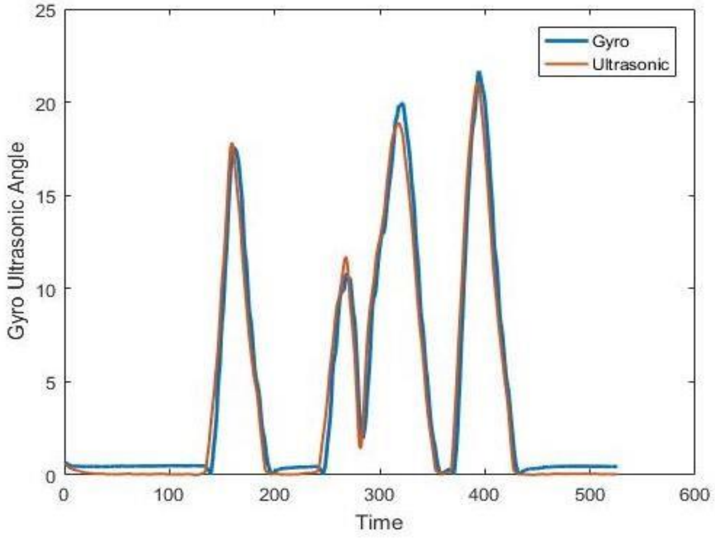

Fig.12. Case 5: Kalman Filter Output

6. Data Processing

For data processing of the angles from Arduino for ultrasonic as well as MPU-6050 [5] sensor, a serial connection was established between the Arduino and MATLAB software. The readings were sent through a serial communication and were stored in an array using Python 2.7. Both ultrasonic sensor readings and MPU-6050 readings were stored in separate arrays. Few readings from the beginning and the end were removed as they were erratic because of sensor stabilization and transducer tuning of transmitter and receiver. Each array value was plotted simultaneously. The serially printed readings were first extracted using Python 2.7 from the Arduino serial monitor [11] and saved in the text format. On MATLAB, data from each array were stored in their respective vector. Both the vectors of MPU-6050 angles and the ultrasonic angles had to be of same length for performing plotting operations on MATLAB.

In MATLAB, the plot function [9] can hold multiple plots for graphical implementation which were used for the comparison of the angles.

Table 4. MATLAB Pseudo Code for Comparing the Angles

MATLAB Pseudo Code for Data Processing

```
UltraSonic = [ ]
Gyro = [ ]
plot (UltraSonic, ‘Linewidth’, 5) hold on
plot (Gyro, ‘Linewidth’, 5)
legend (‘Ultrasonic Sensor’,’Gyroscope’)
ylabel (‘Angle Comparison’)
xlabel (‘Time’)
Max_Error = max (UltraSonic - Gyro)
Min_Error = min (UltraSonic - Gyro)
Average_Err = sum (abs ((UltraSonic – Gyro))/length(UltraSonic)
```

The following table shows maximum, minimum and average error for each experiment. Both Ultrasonic and MPU-6050 based angles were plotted using the plot function on MATLAB. Linewidth of value 5 describes the width of graph plotting. Legend describes the color along with its indication to the parameter it belongs to. Maximum error was gathered by the max function which finds the maximum value from the resultant vector containing the difference between the Ultrasonic and Gyro vectors. Both vectors were equal in length containing 784 values each. The minimum error was also calculated the very same way. Each error calculated were converted to positive values and their sum was divided with the length of vectors i.e. 784 giving us average error for each case.

Table 5. Data Obtained from Each Case

Case	Weight (Ultrasonic Sensor)	Maximum Error (degrees)	Min. Error (degrees)	Average error (degrees)
1	0.75	1.08	0.09	0.38
2	0.8	1.10	0.06	0.33
3	0.8 | 0.8	1.14	0.00	0.24
4	0	1.94	0.22	1.09
5	Kalman Filter	0.85	0.00	0.19

The data obtained from each case clearly shows that Case 4 had shortest response time with high error rate and high average error when compared to other cases. Case 5 showed least error rate with minimum average error. The response time was least after Case 4 and Case 1. In Case 3 with double filtration applied on MPU-6050 based angles, output showed highest noise immunity. Servo based noise were mostly filtered out.

7. Conclusion

In this paper, we have discussed about angle measurement for robots on a flat surface using an ultrasonic sensor. Exponential and Moving Average low pass filters were used with exponential filter showing better results. The angle calculated by mathematical computations from the distance measured by the ultrasonic sensor was accurate with a deviation of ±0.55 ° at most of the times. The response time decreases with weight value being closer to 0 and increases when the value gets closer to 1. However, values of weight closer to 0 also increases the amount of noise in the sensor readings. Thus, a suitable value had to be chosen to keep a balance between the response time and noise interference. Kalman filter was also used which showed better results than exponential filters both in terms of response time and noise immunity. Case 3 showed highest noise immunity. Case 5 showed the least error.

Through the results, we can conclude that the ultrasonic sensor with a combination of appropriate filters is very effective for measuring angle for mobile robots on a flat base. They can be used in the place of gyroscope and accelerometers for measuring angles with accuracy and can successfully be used in the field of robotics.

References

[1] Barrett, S. F., & Pack, D. J. (2007). Atmel AVR Microcontroller Primer: Programming and Interfacing. Synthesis Lectures on Digital Circuits and Systems, 2(1), 1-194. DOI:10.2200/s00100ed1v01y200712dcs015

[2] Shrivastava, A. K., Verma, A., & Singh, S. P. (2009). Distance Measurement of an Object or Obstacle by Ultrasound Sensors using P89C51RD2. *International Journal of Computer Theory and Engineering,* 64-68. doi:10.7763/ijcte.2010.v2.118

[3] Badamasi, Y. A. (2014). The working principle of an Arduino. *2014 11th International Conference on Electronics, Computer, and Computation (ICECCO)*. doi:10.1109/icecco.2014.6997578

[4] Kalman, R. E. (1960). A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering, 82*(1), 35. doi:10.1115/1.3662552

[5] "MPU-6050 Product Specification", Retrieved from: *http://invensense.com/mems/gyro/mpu6050.html*. [accessed August 19, 2013]

[6] "Intro. to Signal Processing: Smoothing." *Cell Differentiation by Gram's Stain*, *terponnect.umd.edu/~toh/spectrum/Smoothing.html*.

[7] Martinsen, Paul. "Three Methods to Filter Noisy Arduino Measurements." *MegunoLink*, MegunoLink, 4 June 2017, *www.megunolink.com/articles/3-methods-filter-noisy-arduino-measurements/?nabe=6011995085340672:0.*

[8] Denyssene. "Denyssene/SimpleKalmanFilter." *GitHub*, github.com/denyssene/SimpleKalmanFilter.

[9] "Plot Function." *Neural Network - Multi Step Ahead Prediction - MATLAB Answers - MATLAB Central*, in.mathworks.com/help/MATLAB/ref/plot.html.

[10] Mercer, C. "Data Smoothing: RC Filtering and Exponential Averaging" Retrieved from: *http://blog.prosig.com/2003/04/28/data-smoothing-rc-filtering-and-exponential-averaging/*

[11] "Arduino and Python" Retrieved from *https://playground.arduino.cc/Interfacing/Python*

[12] J.R. Guerci, R.A. Goetz, J.Di Modica, "A method for improving extended Kalman filter performance for angle for angle-only passive ranging", *IEEE Transactions on Aerospace and Electronic Systems*" Vol 30, pp 1090-1093, 1994.

Authors' Profile

Shashi Suman completed his under graduation in the Department of Electronics and Communication Engineering from JECRC University (India). His research interests include automation, designing of mobile robots, machine learning and its implementation on FPGA designing intelligent robots.

How to cite this paper: Shashi Suman,"Angle Measurement on a flat Surface using High Frequency Ultrasonic Pulse", International Journal of Engineering and Manufacturing(IJEM), Vol.8, No.6, pp.26-41, 2018.DOI: 10.5815/ijem.2018.06.03