# Design Implementation of a Two Wheel Self Balancing Robot with a Two Level Adaptive Control

Ammar Wasif [1], Danish Raza[2], Waqas Rasheed[3], Zubair Farooq[4], Syed Qaseem Ali[5]
[1]Department KESC, Karachi, Pakistan
[2]Solution Engineering Sysnet, Karachi, Pakistan
[3]NED University, Karachi, Pakistan
[4]Bank Islamic
Karachi, Pakistan
[5]King Saud University
Riyadh, Saudi Arabia
{ammar_wasif, danish03}@live.com, waqasrasheed90@ymail.com, zubairfarooq12@hotmail.com, qaseemali@gmail.com

***ABSTRACT:*** *An attempt to tackle the inherent instability of the two wheel platform has been made. Control of this unbalanced system is achieved by implementing an adaptive rendition of the classic PD control. The implementation procedure and the relevant theory behind the autonomous balancing and movement of the robot is also demonstrated. The stability and responses of this two wheel self-balancing system is verified by practical results.*

**Keywords:** Adaptive Control, Inverted Pendulum, PID Controller, Segway, Self-balancing, Two Wheel

**Received:** 1 January 2013, Revised 18 January 2013, Accepted 23 January 2013

## 1. Introduction

The two-wheeled, self-balancing robot is a non-linear multi-variable and naturally unstable system. Controlling such a system is a challenge therefore it attracts attention of many modern control researchers. Control concepts such as system stability and robust control of systems can be verified by experimenting on such systems. Therefore it serves as a facility for testing and verifying various control techniques. Many techniques for the control of a two-wheel self-balancing robot have been proposed. They include the PID Back Stepping Controller, presented in [1] shows that fuzzy logic can improve the robot performance.

The sliding mode control presented in [2] and [3] is a robust control strategy based on a combination of SMC and disturbance observer and obtains high performance despite the presence of disturbance and noise.

Linear Quadratic Regulator (LQR) method presented in [4] is a type of system in which dynamics are described by a set of linear differential equations .

In Fuzzy Back stepping controllers presented in [5] the controller uses fuzzy logic based on discrete values.

The two-wheeled self-balancing robot is modeled as an inverted pendulum, which is widely used by the researchers not only to

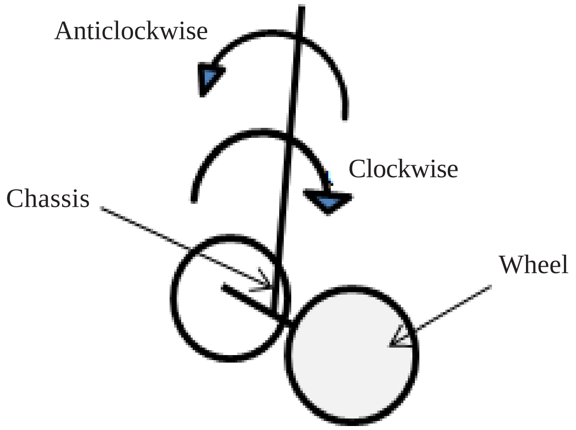

Figure 1. Schematic of Two Wheel Robot

model wheeled robot but also their legged counterparts [6].

## 2. Working of Two Wheel Robot

This kind of robot consists of two wheels supported by a chassis as shown in Figure 1, carrying the drive motors (DC motors in our case). When the body loses its balance and starts to fall, the motors generates a counter torque to avoid the structure from falling and tries to balance the robot.

If the robot falls in clockwise direction the motor rotates clockwise to prevent the robot from falling and vice versa. To balance the robot, the angle from the normal, $\theta$ bal (Figure 1), should be maintained as close to zero as possible for e.g. if angle is 10 degree the controller will try to reduce the angle to zero using its actuators (motors). The angle is reduced by moving the robot in direction of tilt. In this way the robot will try to maintain itself at the reference angle which is defined by the designer to balance itself. There are several ways to achieve the movement of robot. One of them is to add a little offset to the reference angle. For example, instead of balancing at 0 degree, the robot tries to balance at 3 degree, hence keeping itself moving in the direction of the maintained tilt. Another one, which is also discussed in this paper, requires to have an outer speed control loop to maintain robot movement at a reference speed.

## 3. Control and Sensing

To measure the angle sensors like gyroscope and accelerometer are the most commonly used sensors. However, these sensors have their own discrepancies. The accelerometer measures acceleration so it also contains the linear component, along with the angular component present in the acceleration, as shown in Figure 2.

The gyroscope measures angular velocity and by integrating it, angle can be obtained. The disadvantage of gyroscope is that it shows small rate of rotation, even at rest. Furthermore the integration results in a slow creeping tilt error.

To overcome the mentioned issue a filter must be designed because the system performance is very sensitive to the measured angle. There are various options for the filter but the most commonly used filters are:

- Kalman Filter
- Complementary filter

After bit of research, it was concluded that Complementary filter should be implemented due to its various advantages over Kalman Filter. The Kalman filter has a complex design compared to the design of the Complementary filter. Moreover, due to complex calculations involved in the Kalman filter it requires higher computational resources and time. Although Kalman filter has a more accurate result, but to save computational resources and time the Complimentary filter provides a good compromise.

Some of the details of the Complementary filters are presented in the next section.

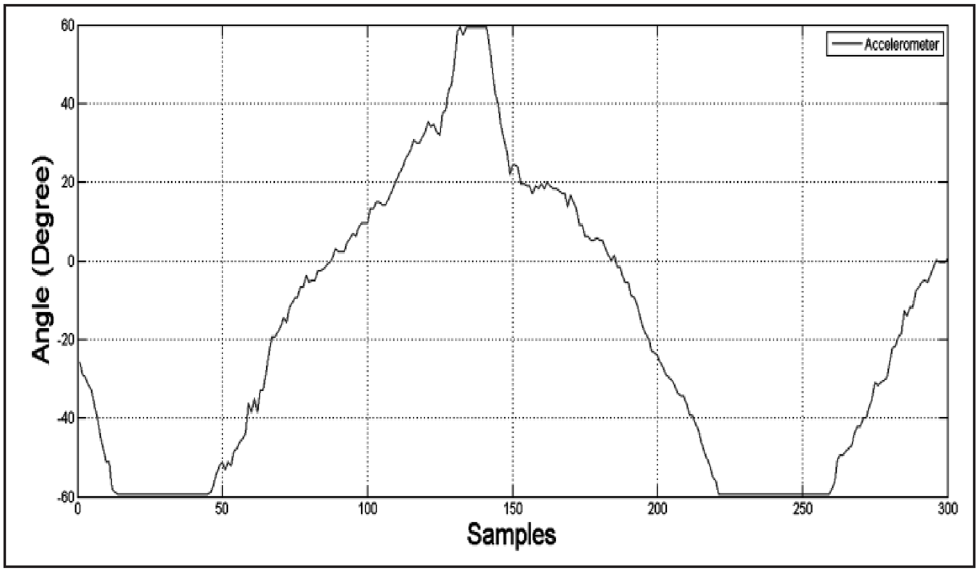

Figure 2. Accelerometer Reading with the Notches represents the linear acceleration

### 3.1 Complementary Filter

To remove the high frequency distortion in the reading of the accelerometer it is passed through a low pass filter. The angular velocity obtained from gyroscope is integrated to obtain the angle and then passed through a high pass filter to remove the low frequency distortions. The result obtained from low pass and high pass filter are then summed to find the estimated angle. Block diagram of the complementary filter is shown in Figure 3.

### 3.2 PID Controller

A simple PID controller was used for the system, as shown in Figure 4. The control equation is defined in Equation 1.

$$u(t) = k_p e(t) + k_i \int_0^t e(\tau)\, d\tau + k_d \frac{d}{dt} e(t) \qquad (1)$$

Where $e(t)$ is the error i.e. it is the difference between reference $R$ and feedback, $u(t)$ is the control output of the controller, $k_p$ is the proportional gain, $k_i$ is the gain of integral term $k_d$ and is the gain of derivative term.

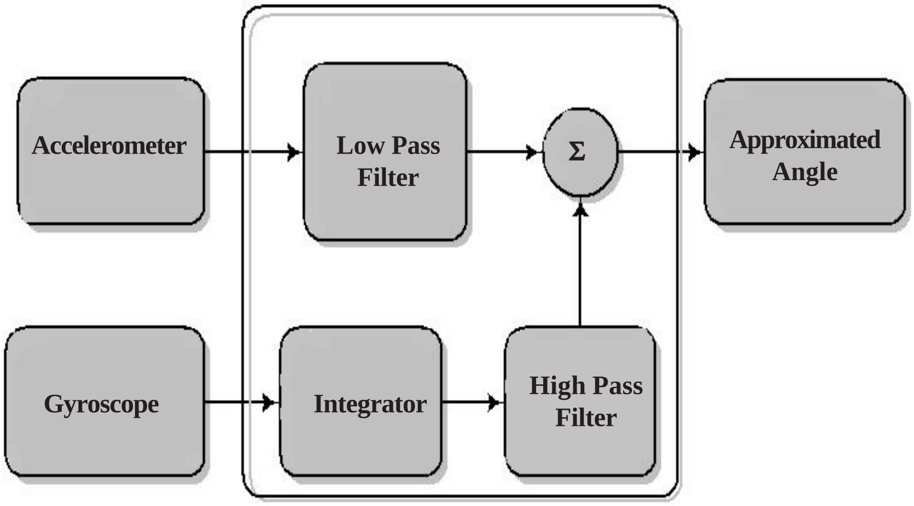

Figure 3. Block Diagram of Complementary Filter

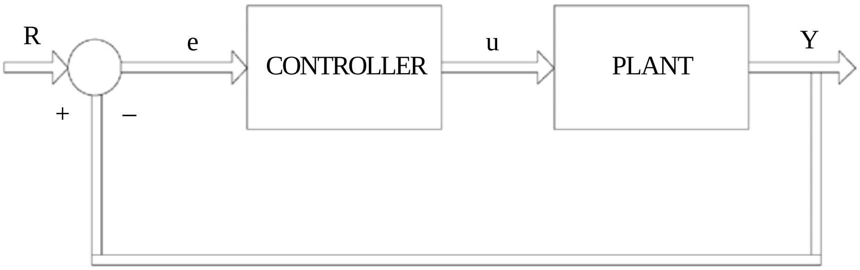

Figure 4. PID Controller

## 4. Developed System

### 4.1 System Diagram

**Controller**

Arduino UNO has been selected as the main controller, it has a flash memory of 32KB, 6PWM channels, 6 AD ports, 2 interrupts which meet our requirements. It is open source device therefore its resources are easily available.

**Motors**

The motors selected were PITTMAN 24V DC with rated speed of 6118 rpm and is stepped down 46:1 using a gear head. The DC motors were selected because of easy availability of the DC supply from the batteries and the ease with which they can be controlled.

**Motor Driver**

A simple H-bridge was designed and developed to drive the motor. It consists of four NMOS and two PMOS transistors for each motor in a configuration shown in Figure 6. PMOS transistors are used because in NMOS the gate voltage must be greater than drain voltage for switching making the design of the high side switch driver complex, whereas in PMOS it is not a problem.

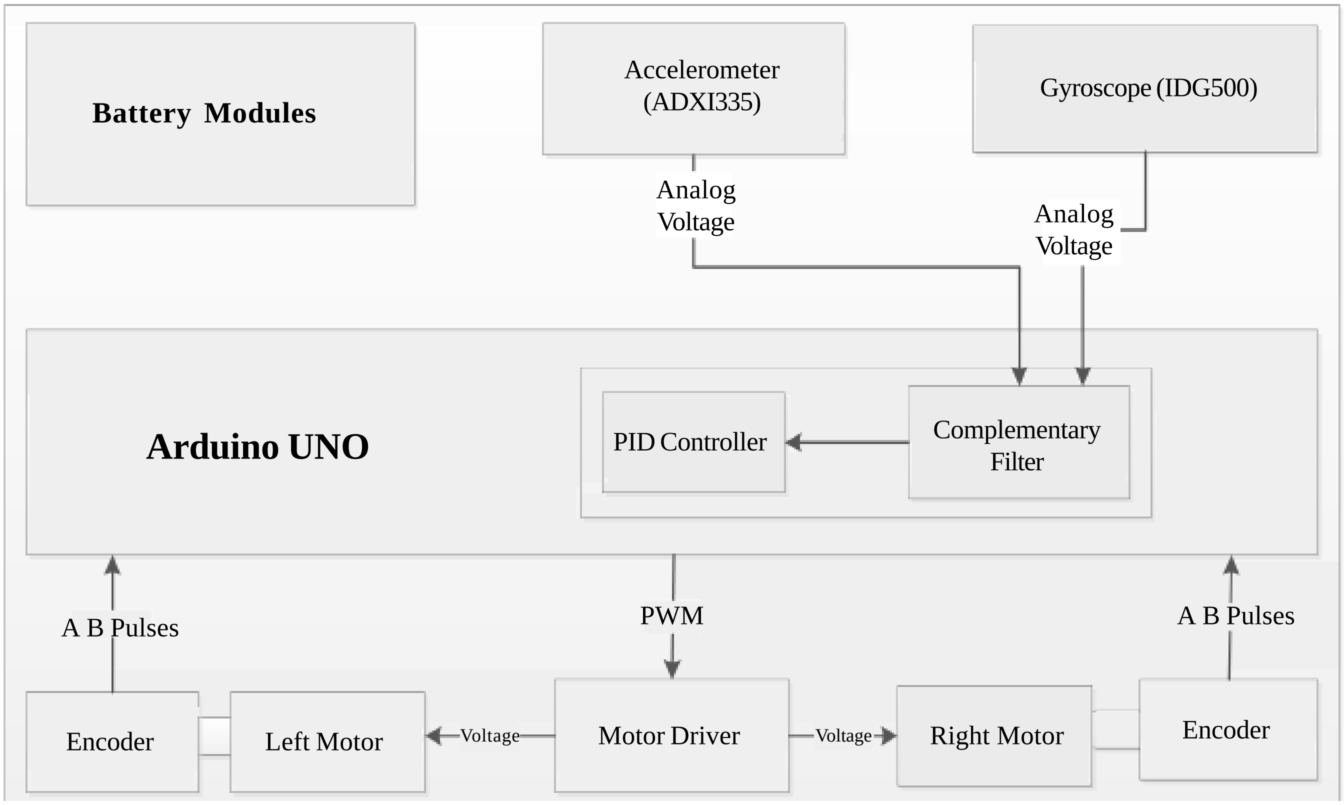

Figure 5. Block Diagram of System

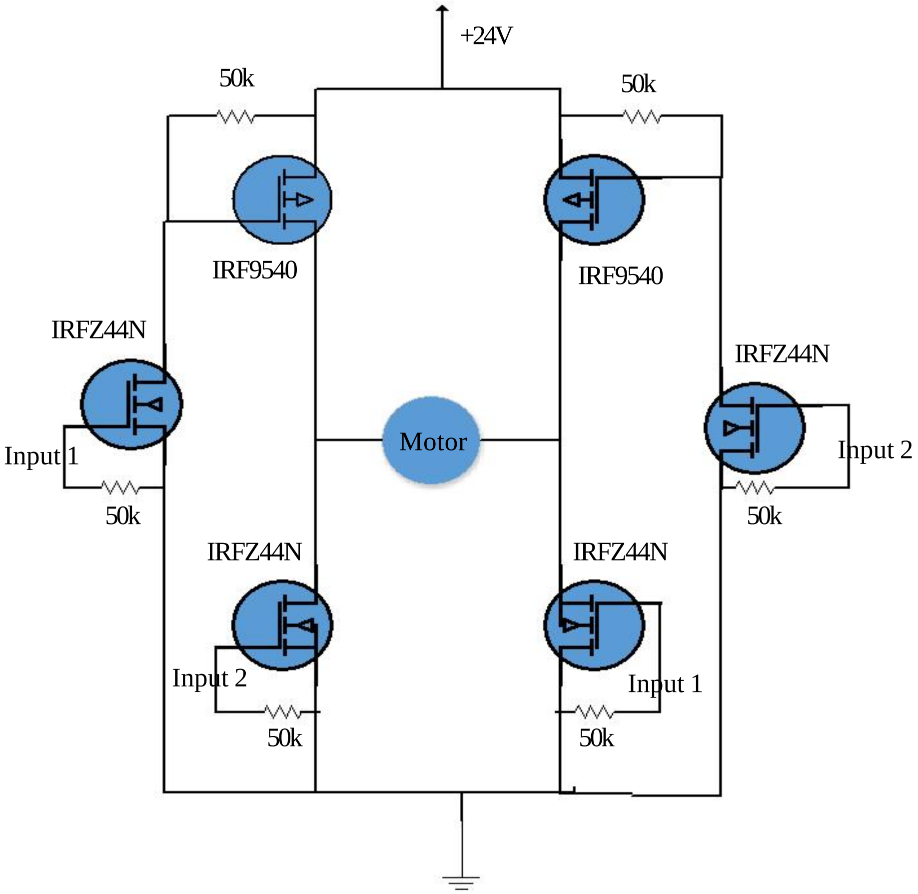

Figure 6. H-Bridge

**Encoder**

The encoder selected was from the manufacturer TamagawaFa and has 1024 counts/ revolution which means it can sense the change  of $0.35^o$ at least

**Accelerometer**

The accelerometer selected was ADXL335 by Analog Devices available as a breakout board. It was selected because of its sensitivity, which fulfills our requirement i.e. 300mV/g at 3V.

**Gyroscope**

The gyroscope selected was IDG500 dual-axis gyro by Invensense, which is also available on a breakout board and provides the facility of switching between two sensitivities XOUT or X4.5OUT, however XOUT mode fulfills our requirements.

**Battery Unit**

Two battery modules were selected, a 12V 2.3Ah module to drive the motors and a 12V 1.2Ah module for the control board and sensors.

### 4.2 Designing of Complementary Filter

The filter time "$\tau$" plays important role in the performance of the complementary filter. Relation between time constant "$\tau$" and filter constant "***a***" is given in the Equation 2.

$$a = \frac{\tau}{dt + \tau} \qquad (2)$$

Where "*dt*" is termed as loop timing.

Therefore the performance of the filter was evaluated for several randomly selected time constants against the value of *a*, the results of which are shown in Figures 8-11, the one with the best performance was selected.

From Figures 8-11, the buffering action of complementary filter can be observed with diffrent values of constant "*a*" in Figure 8 the filtering of distortion in accelerometer reading is minimum but the delay introduced by the filter is high. In Figure 9 the delay in measured angle decreases but remains significant. In Figure 10 the delay is further decreased. In Figure 11 the delay is decreased even further and the reading approaches real time but the distortion is too high and higher frequency components are visible on the signal. It was decided to set "*a* = 0.80" (shown in Figure 10), as this value is a good compromise between distortion and delay.

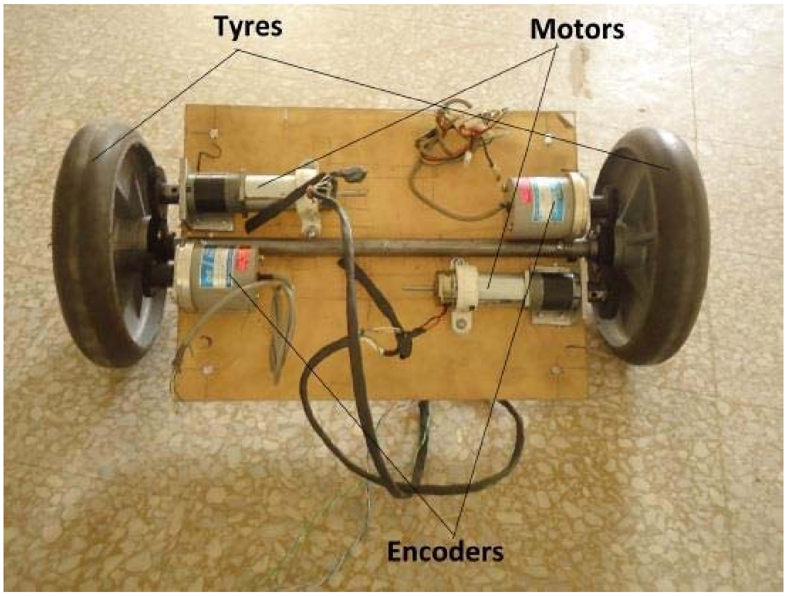

Figure 7. Picture of base of robot

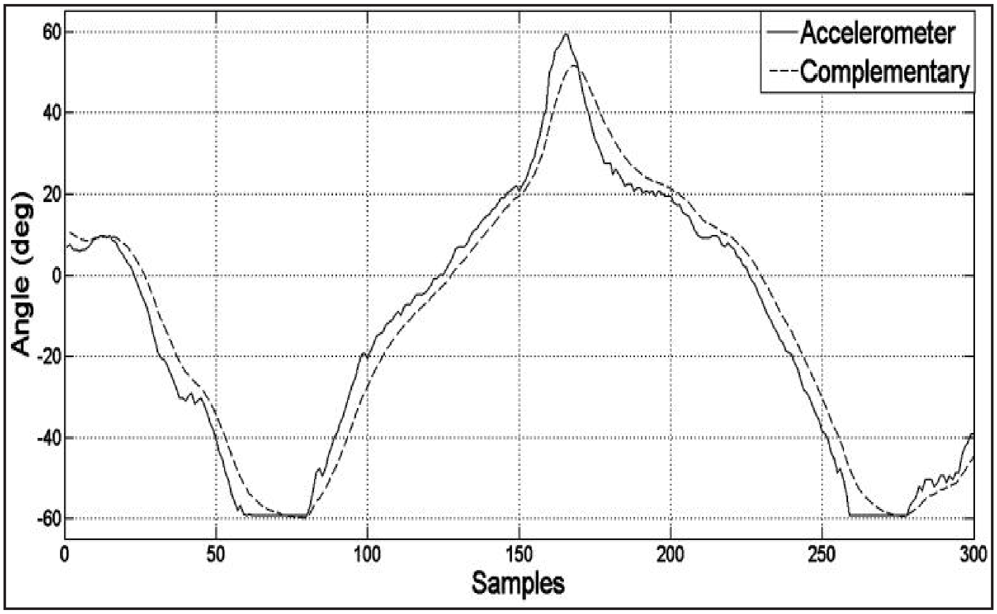

Figure 8. Plot with *a* = 0.90

**4.3 Controller Designing**

Set of equations used for designing control system.

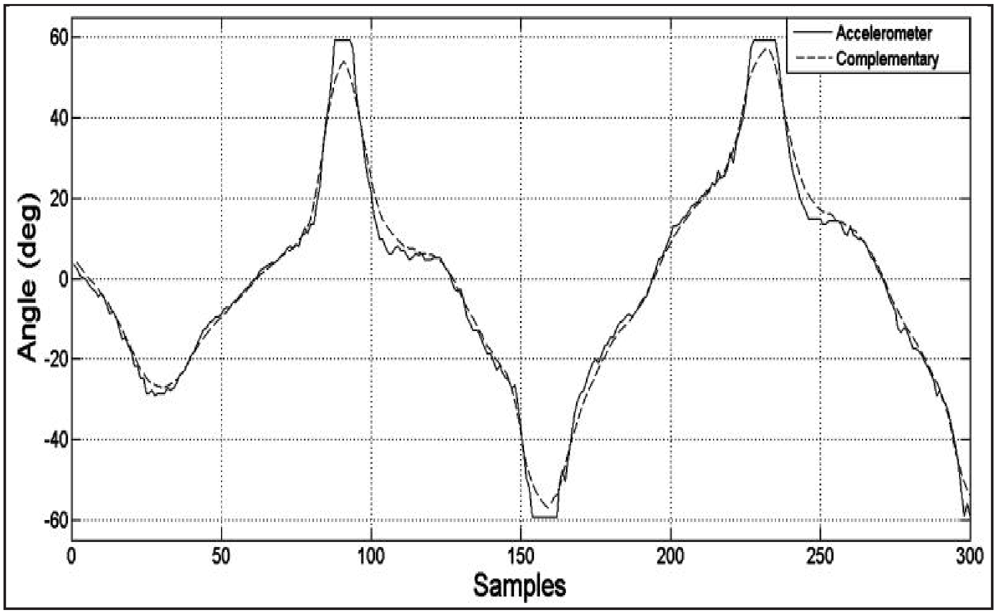

Figure 9. Plot with $a = 0.85$

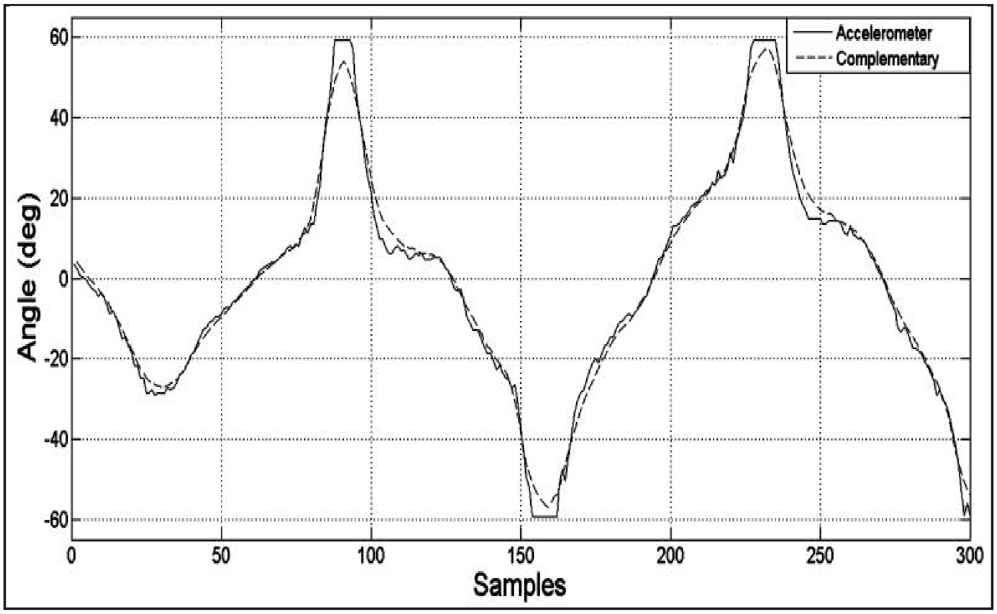

Figure 10. Plot with $a = 0.80$

$$e(k) = r(k) - y(k) \tag{3}$$

$$e_i(k+1) = e_I(k) + \Delta t * e(k) \tag{4}$$

$$e_d(k+1) = (e(k) - e(k-1) / \Delta t \tag{5}$$

$$u(k) = k_p * e(k) + k_i * e(k) + k_d * e_d(k) \tag{6}$$

Where $e(k)$ is the error for Proportional Controller, $e_i(k+1)$ is the error of Integral controller and $e_d(k+1)$ is the error for Derivative Controller.

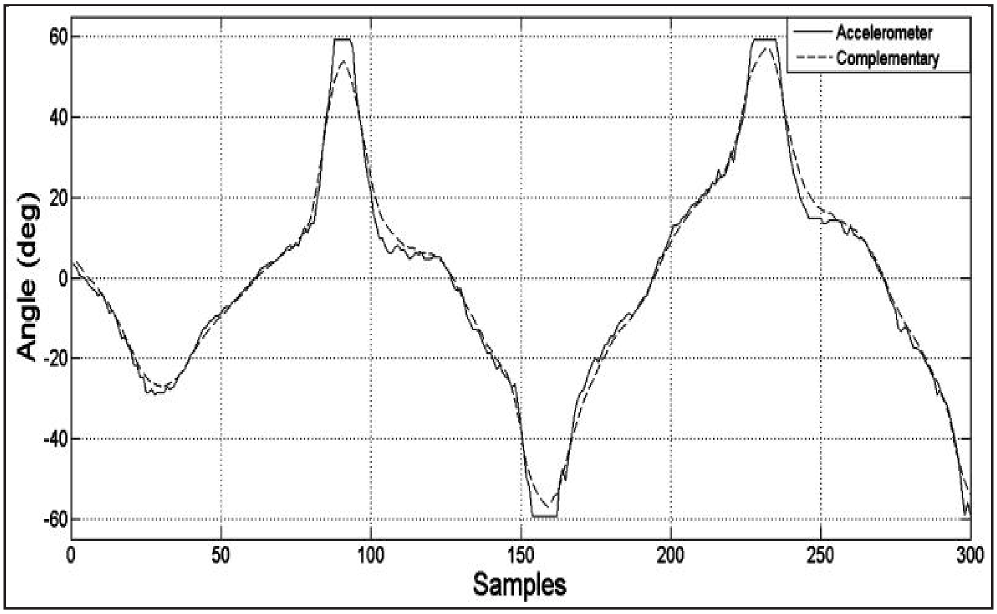

Figure 11. Plot with *a* = 0.75

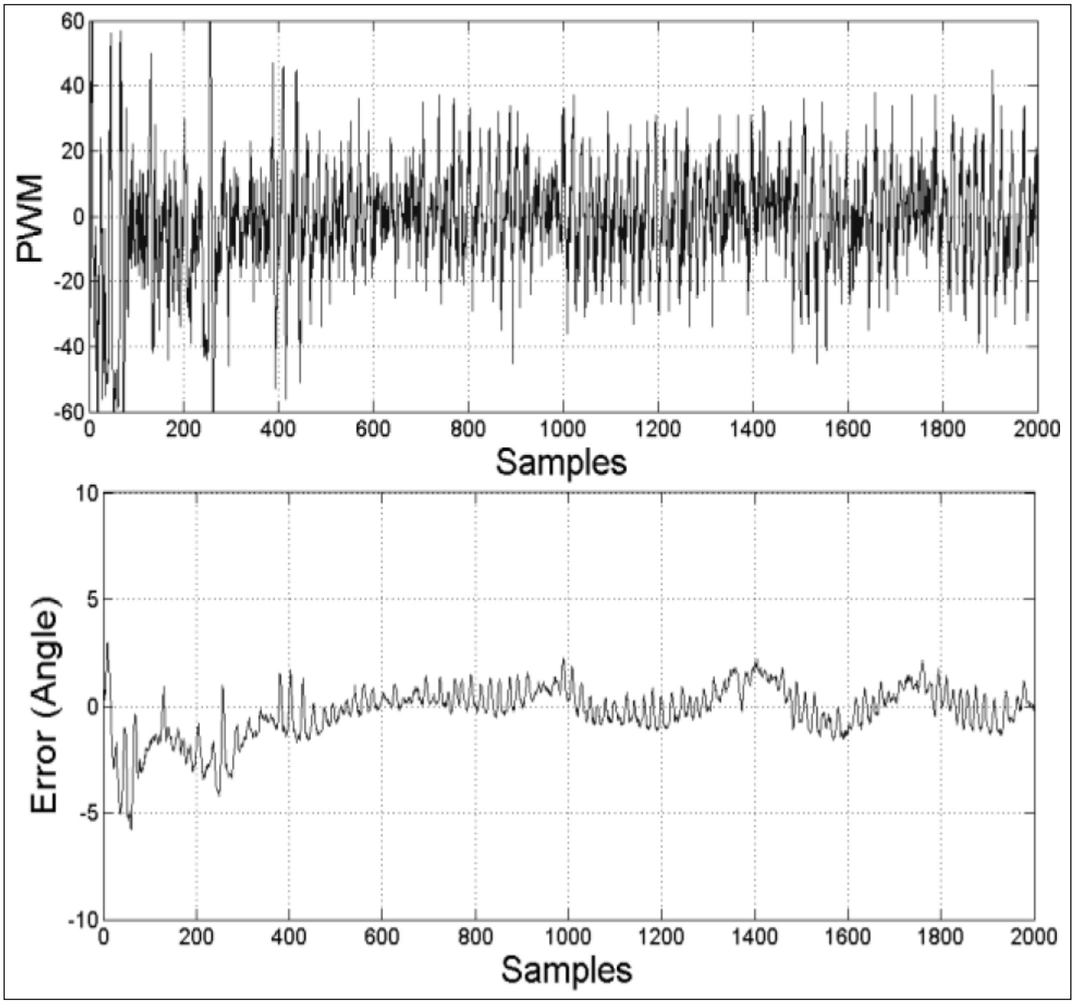

Figure 12. Response of P controller Error and PWM plot

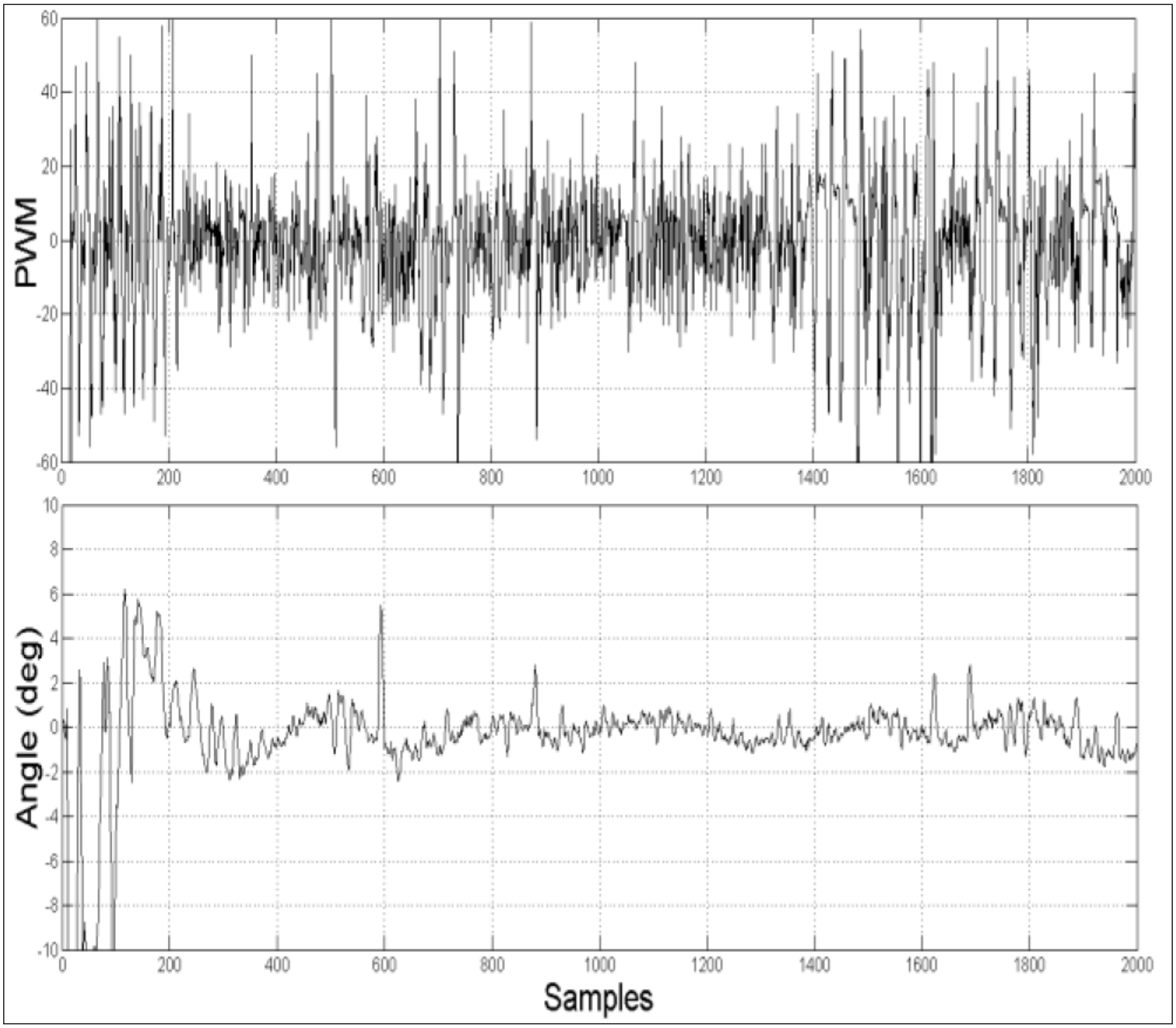

Figure 13. Response of PD Controller ($k_p = 7$, $k_p = 20$) Error and PWM Plot

**Implementing the *P* Controller**

$$u(k) = k_p * e(k) \tag{7}$$

In this Equation 6 of controller a gain $k_p$ (tuning parameter) is set according to the stability of the system which is selected by hit and trial.

As it is evident from the Figure 12 the large error values present in the system which is the major drawback for the stability of the system.

The following observations were made when P Controller was implemented:

- High dependence on the voltage of the battery
- *P* controller balances the robot, but there are more oscillations present in the error values which are shown in Figure 12.

To minimize the error PD controller was implemented which is discussed next.

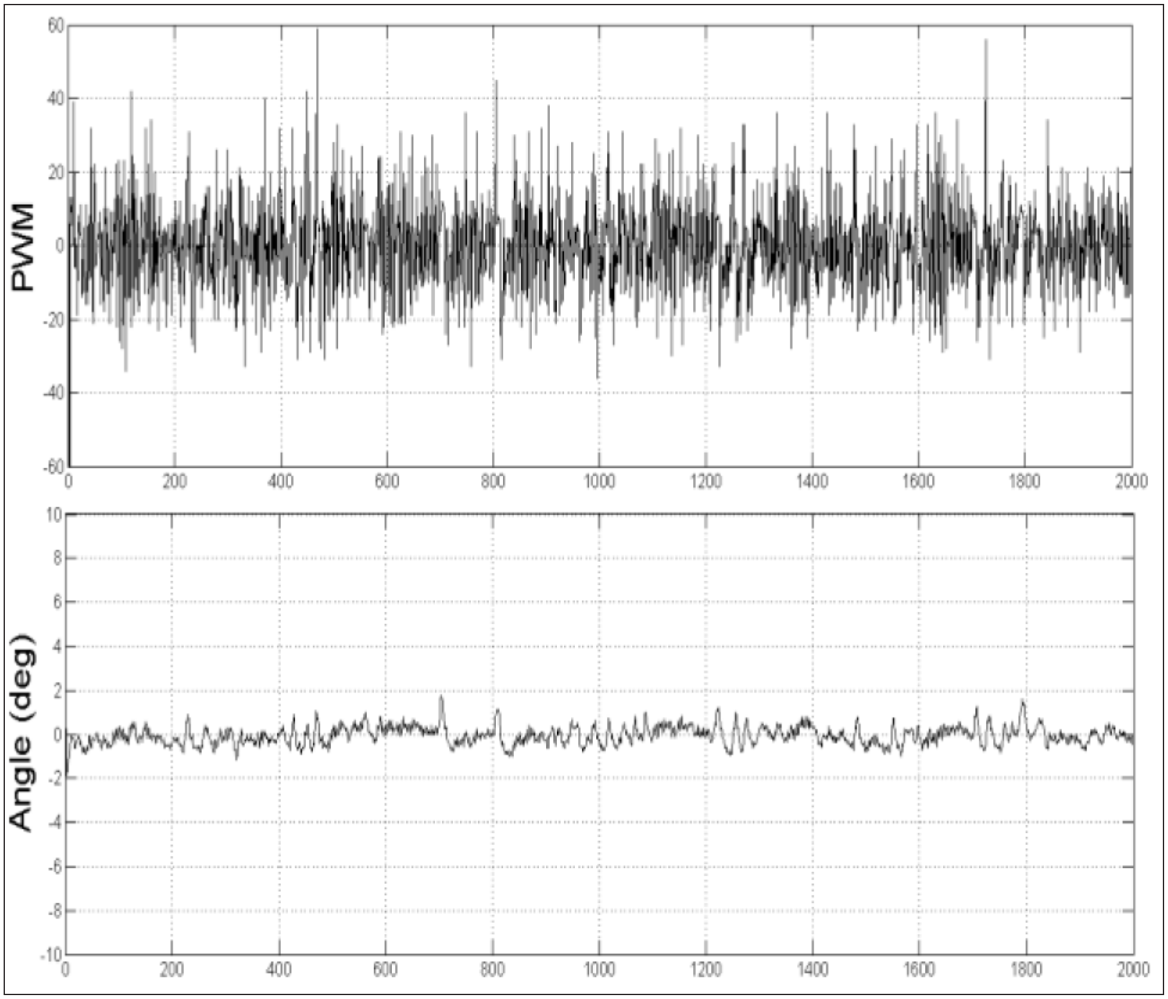

Figure 14. Response of PD controller ($k_p = 4$, $k_d = 40$) Error and PWM plot

**Implementing PD Controller**

$$u(k) = k_p * e(k) + k_d * e_d(k) \qquad (8)$$

Equation 7 is implemented for PD control system. To minimize the oscillations in the error a derivative controller was introduced in the control equation.

In this equation gain $k_p$ and $k_d$ were set by hit and trial to achieve system stability and reduce the errors as shown in Figure 13 and Figure 14.

It can be inferred from the graphs that the controller is trying to minimize the error as well as maintaining a constant value without increasing oscillations.

The following observations were made while assessing the system:

- Dependence on the battery and the constant need of parameter retuning is minimized.
- The robot is balanced with reduced oscillations making it more stable which was not possible by a simple P controller.
- The robot loses its resistance against the disturbing forces because the proportional relationship $k_p$ gain has been reduced to a minimum level.

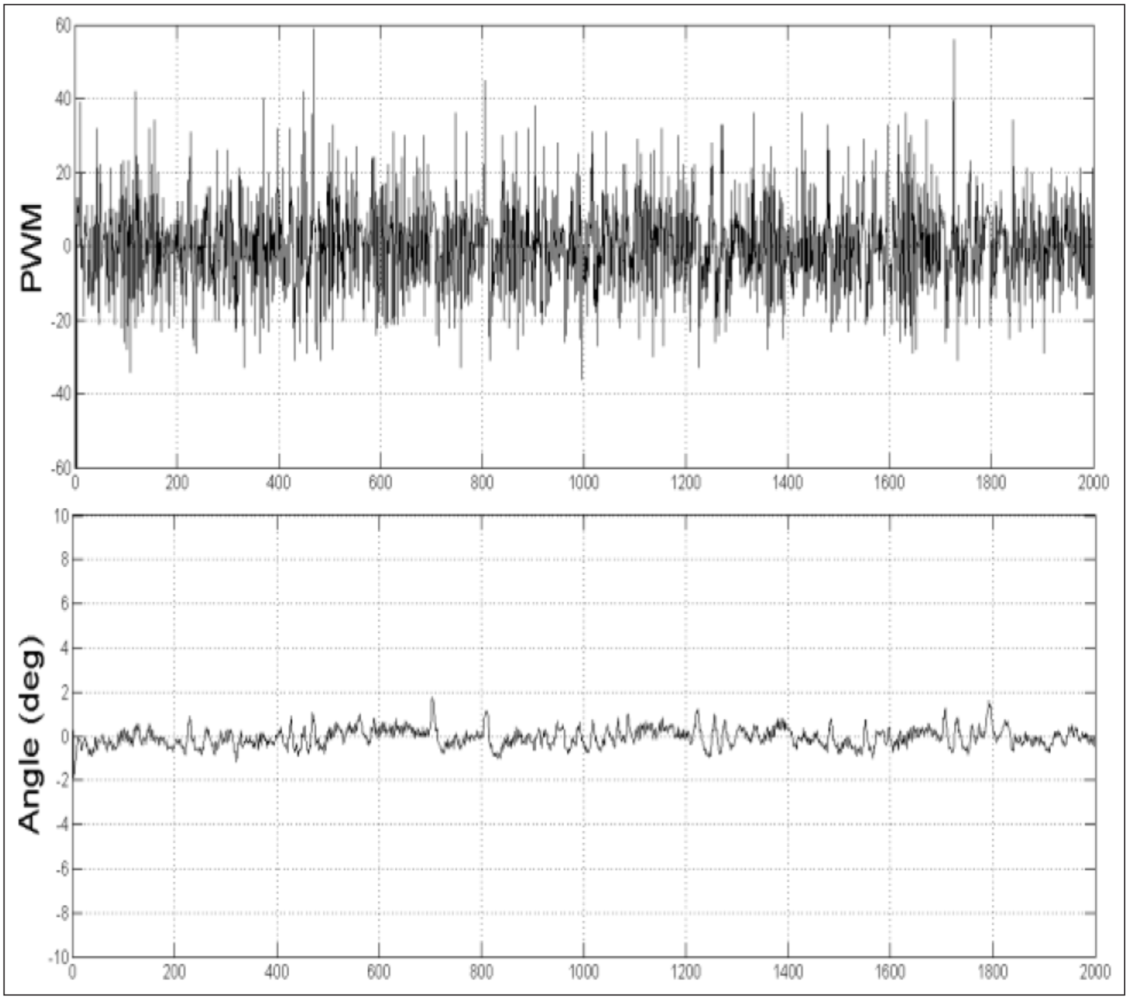

Figure 15. Response of Adaptive Control Error and PWM Plot

As from the graph it can be seen that there is still distortion in the system. In order to make the system more stable and to overcome the above drawback of PD controller, an adaptive technique was implemented.

**A Two Level Adaptive Control**

Adaptive control is the control method where the control parameters are varied according to the set points of the system. For example, at starting motor has high tuning parameters and after the motor is in its stable state the parameters are varied according to the desired condition by control system. In adaptive control the tuning factors are varied depending on the different conditions to make the system more stable. Here two level adaptive control refers to the parameters that are defined for only two ranges of reference angles.

The adaptive control was implemented on the angle in the balancing loop. The parameters $k_p$ *and* $k_p$ were designed to have different values depending on the reference angles.

**Results of Two Level Adaptive Control**

Figure 15 shows the response of the two level adaptive control which minimizes the oscillations in the system which were still present in the PD Controller implementation. The parameters were further tuned and the results are shown in the Figure 16. Response of the controller after the implementation of the adaptive control with tuned parameters further improves the stability.

Numbers of oscillations of the robot as shown in Figure 16 are decreased and the error is reduced to an extent that was not quite possible without adaptive control in the designed system, with the following features:

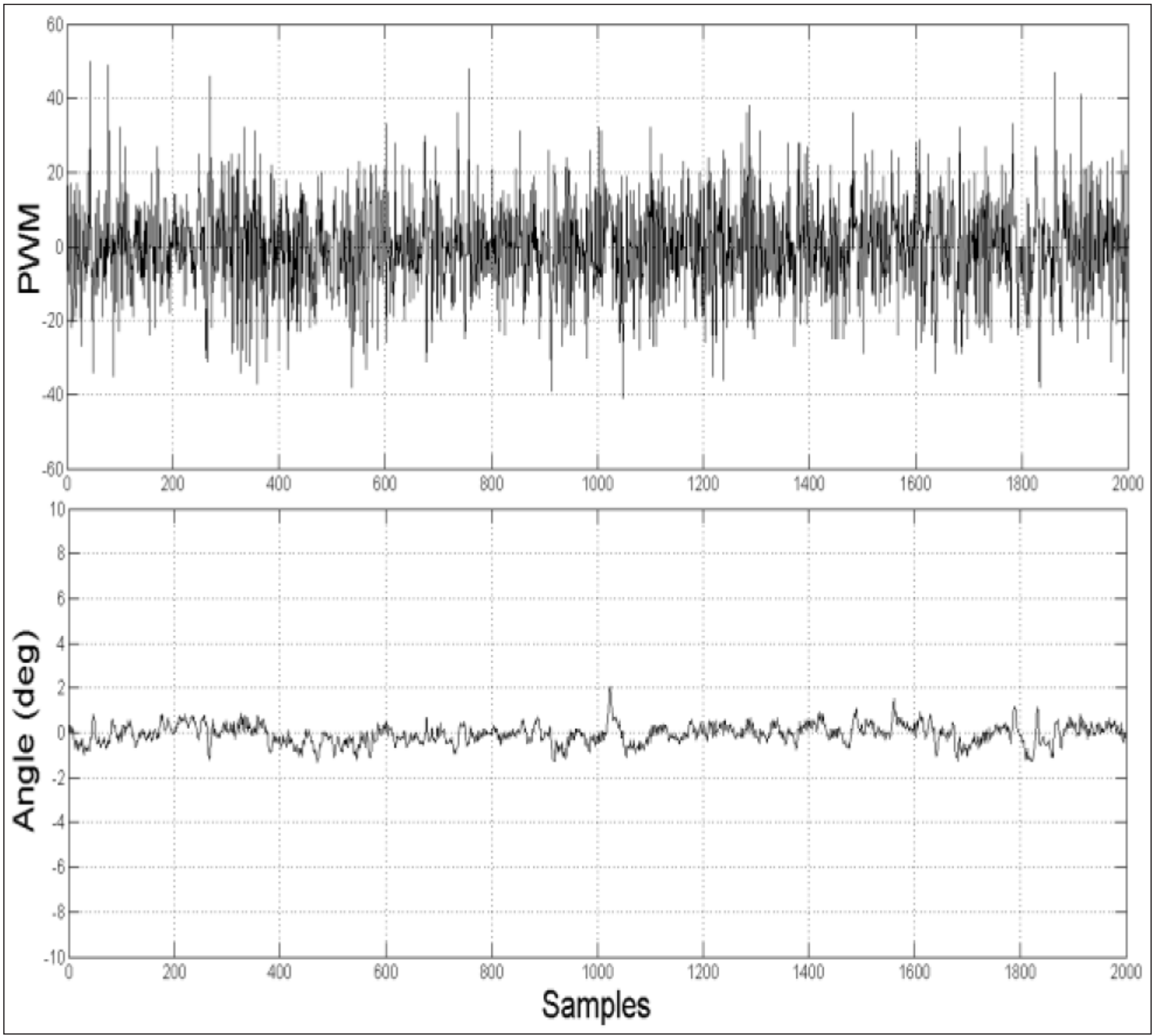

Figure 16. Response of Adaptive Control Error and PWM Plot with tuned parameters

| CONTROLLER | Robot Balance | Overshoots | Resistance against opposite forces |
|---|---|---|---|
| **P** | **Yes** | **Present** | **Yes** |
| **PD** | **Yes** | **Reduced** | **No** |
| **PI** | **No** | **-** | **-** |
| **PID** | **Yes** | **Increases** | **Yes** |
| **2-Level Adaptive PD (Un tuned)** | **Yes** | **Reduced** | **Yes** |
| **2-Level Adaptive PD (Tuned)** | **Yes** | **Minimized** | **Yes** |

Table 1. System Response on Different Parameters

- Reduced the amount of oscillation in error.
- Robot becomes more rigid against the disturbances
- Robot becomes more stable

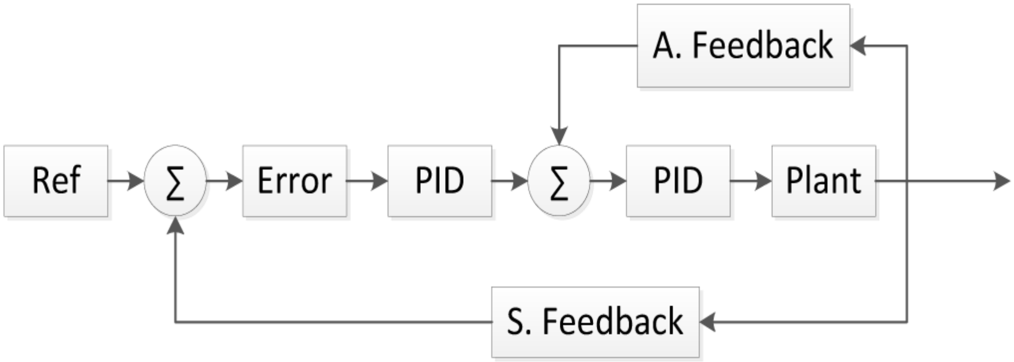

Figure 17. Block Diagram of Forward Movement Strategy

**Robot Movement**

For moving forward the control strategy implemented is shown in Figure 17 Ref is the Reference Speed, A. Feedback is the angle feedback taken from complimentary filter and *S*. Feedback is the speed feedback taken from encoder.

In Figure 17, an outer loop has been implemented which takes speed as feedback. This calculated speed is compared with the reference speed and error is generated. The error limit block saturates the error to a specified range to limit the effect of the speed loop on the balancing. This is done because in case large speed error occurs, the balancing control loop will be disturbed and may cause the robot to fall or make it highly unstable. This speed control loop generates the reference for the position control loop or balancing loop. If the robot is to be only balanced at its position the speed reference is set to zero. If the robot has to be moved forward, the speed control loop generates a ramp reference for the position control loop, which is tracked by it, thus making the robot move.

## 5. Conclusion

In this paper the control system has been implemented for the balancing of a two wheel robot. The control system comprised of a two level adaptive PD controller. Complementary filter was designed and implemented to remove the distortion in the angle measurement. In future the design of control system needs further improvement to make it more robust for different environmental conditions. And also the structure designed which should be fabricated properly to make it more flexible for different conditions and applications.

## 6. Acknowledgment

This project would not have been possible without the guidance and the help of several individuals who in one way or the other contributed and extended their valuable assistance. First and foremost our utmost gratitude to Dr. Shoaib Zaidi, Dean ECE NED University of Engineering & Technology who helped us formulate the idea. We would also like to extend our gratitude to Ms. Hira Mariam, Lecturer NED University of Engineering & Technology who helped us throughout the execution of the project.

## References

[1] NguyenGia Minh Thao, Duong HoaiNghia, Nguyen HuuPhuc. (2010). A PID Backstepping Controller for Two-Wheeled Self-Balancing Robot, IFOST 2010 Proceedings.

[2] Junfeng Wu, Yuxin Liang, Zhe Wang. (2011). A Robust Control Method of Two-Wheeled Self-Balancing Robot, The 6th International Forum on Strategic Technology.

[3] Abeygunawardhana, P. K.W., Michael Defoort, Toshiyuki Murakami. (2010). Self-Sustaining Control of Two-Wheel Mobile Manipulator using Sliding Mode Control, The 11th IEEE International Workshop on Advanced Motion Control, March 21-24, Nagaoka, Japan.

[4] SeongheeJeong, Takayuki Takahashi. (2008). Wheeled inverted pendulum type assistant robot: design concept and mobile control, *Intel Serv Robotics*.

[5] XiaogangRuan, JianxianCai. (2009). Fuzzy Backstepping controllers for Two-Wheeled Self-Balancing Robot, International Asia Conference on Informatics in Control, *Automation and Robotics*.

[6] Goher, K. M., Tokhi, M. O., Siddique, N. H. (2011). Dynamic Modeling and Control of a Two Wheeled Robotic Vehicle with a virtual payload, *ARPN Journal of Engineering and Applied Sciences*, 6 (3), March.

[7] Wu Junfeng, Zhang Wanying. (2011). Reasearch on Control Method of Two-wheeled Self-balancing Robot, Fourth International onference on Intelligent Computation Technology and Automation.

[8] Ching-Chih Tsai, Cheng-Kai Chan, Yen-Hau Fan. (2008). Planned Navigation of a Self-balancing Autonomous Service Robot , IEEE International Conference on Advanced Robotics and its Social Impacts, Taipei, Taiwan, Aug. 23-25.

[9] Shane Colton. (2007). The Balance Filter, Rev.1: Submitted as a Cheif Delphi white paper – June 25.

[10] Charles Young, ChiewFoongKwong. (2011). Wireless Controlled Two Wheel Balancing Robot, *International Journal of Network and Mobile Technologies*, 2 (2), May.