Indian J.Sci.Res. 17(2): 556 - 563, 2018 ISSN: 0976-2876 (Print)
ISSN: 2250-0138(Online)

DESIGN AND IMPLEMENTATION OF SELF-BALANCED ROBOT USING PROTEUS DESIGN TOOL AND ARDUINO-UNO.

[1] Niranjan L, [2] Suhas A R, [3] Chandrakumar H S
[1,2,3]Electronics and Communication Engineering, R R Institute of Technology, Bengaluru

Abstract- This paper explores the self-balancing robot, which is a kind of robot where it is nonlinear and unstable system. The system is stabilized using three axis gyroscope sensor and accelerometer which is the heart of the robot and is based on gyroscope effect. Here we use two stepper motor instead of a regular DC motor. The reason behind this is to have a precise and good performance to have no loss when the battery voltage drops. The entire system is controlled by the Arduino-Uno controller. The angle estimation is done via the complementary filters, LQR and LQR-ANN of the system to stabilize and to synchronize both the motors. This will improve the stability performance significantly. The same is implemented using the Proteus design tool and Mat-Lab

Keywords -Roboarm, Servomotor, FlexSensor, Arduinouno, RF Module, L298N module, MATLAB, proteus

I.Introduction

The model of two-wheel automaton (front and rear wheel) is comparable to the model of motorcycles and bicycles, its body is long and slim, that is appropriate for high-speed operation within the slender path and may adapt to the advanced conditions. Bike could be a robust nonlinear coupling model, where research students are around the dynamic model and control formula for the bikes for studies. Jones planned that steering direction of the front wheel was within the same direction with the body leaning once the bike returns to equilibrium, therefore on additional illustrate the direction of control bike balance and pole-zero placement is designed. Wang L When a metal object moves into the inductive proximity sensor field of detection, Eddy circuits build upon the metallic object, magnetically push back, and finally reduce the inductive sensors, own oscillations field. The sensors detection circuit monitors the oscillator strength and triggers an output from the output circuitry when the oscillator becomes reduced to a sufficient level. [1] studied bicycle inverted pendulum model, dynamic equations of the remote-controlled bicycles were designed with the momentum conservation, models were designed below the constant forward speed higher than the predefined value, it is connected to a centralized hub as a gateway from which a system is controller with a user interface. The devices status is monitored throughout the functioning of the system and the same information is indicated and displayed on the LCD screen for monitoring purpose in case of any changes in the status of the devices. [3]. Guo L. et al. Later the system can be charged wireless using induction coupling system, the pads can conveniently be placed under the table and inside the ceiling so there are no visible wires that could ruin the aesthetic feel of the room. The ease of installation and convenience of this device would make the marketability of this product quite large and if finished could be seen in thousands of robotic devices. If we increase the efficiency of coupling unit the it will increase the power slightly further, wireless power transmission could become a standard means for charging a robotic device [3]. established dynamic equations of unmanned bicycle with Lagrange method, or so calculate the kinetic and mechanical energy of the bicycle according to the middle of mass of every rigid body within the based frame. Everett X. Wang et al. [5] established nonlinear dynamic equation with holonomic and non-holonomic constraints, the model contains all the state variables of the bicycles, however only is that the complicated symbol expression, the controller of the mode is tough to be designed. A preliminary dynamic mode of machine self-balancing robot was established with momentum conservation by Li Y.et al. [7], however the model only contains one degree of freedom. This study can regard the simple machine self-balancing robot as amix of 4 rigid bodies, the nonlinear dynamic model of the robot is obtained by using the Euler-Lagrange technique and image software system like Maple [6]. This model has been improved for the modeling technique in [4], cut back the quality of the model, not only can be used for the particular controller design, also can be widely used for a bicycle and machine mobile mechanism. At the top of this paper, sliding mode controller is designed and simulated by MATLAB. The simulation results show that the designed controller is ready tostabilize the machine self-balancing robot model around equilibrium with great initial deviated roll angle up to 45°, the steering management has result on the stability of the mechanism. self-balancing robot as a mix of 4 rigid bodies, the nonlineardynamic model of the robot is obtained by using theEuler-Lagrange technique and image software system like Maple[2]. This model has been improved for the modeling technique in, cut back the quality of the model, not only can beused for the particular controller design,

[1]Corresponding Author

also can be widely used fora bicycle and machine mobile mechanism. At the top of thispaper, sliding mode controller is designed and simulated by MATLAB[8]. The simulation results show that the designed controller is ready to stabilize the machine self-balancingrobot model around equilibrium with great initial deviated rollangle upto 45°, the steering management has result on the stabilityof the mechanism. A Supervisory Control & Data Acquisition (SCADA) system provides users with a Human Machine Interface (HMI) which can be used for controlling, monitoring and protection of devices.

II.Modeling

The mobile robot may be a removable initial order inverted pendulum where it works as the heart of the whole device. It is a mobile platform, which may attain balance by its self-regulation. For cognizing the mobile robot additional and precisely, we need to analysis its mathematical model. The establishment of additional actual model is that the premise of planning control system and management rule. The robotic systems have a mechanical part, controlsystem and sensing system. Here we use Arduino Uno board for the implementation of the system and as a data acquisition system. Itcontains one embedded system sensors for the balancing the system and two stepper motor for the system as wheels. The platform structured by metallic element bars containsthree layers. within the 1st layer, there are a unit motors, reducers, batteries, 2 wheels of car and 2 little safety wheelsin front and back. within the middle layer, there are accelerometer and gyroscope as the heart of the unit.

III. System Design

To keep the robot balanced, the motors should counteract the autumn of the robot. This action needs a feedback and a correcting part. The feedback component is that the MPU6050 gyroscope plus accelerometer, which provides each acceleration and rotation in all 3 axes that is employed by the Arduino to understand this orientation of the robot. The correcting component is that the motor and wheel combination[9]. The Connection of the sensor to the Arduino and testing the connection are done using proteus design tool before implementing on the hardware. The code is analyzed using the Arduino IDE codes. The L298N module can provide the +5V needed by the Arduino as long as its input voltage is +7 V or greater. However, we have chosen to have separate power sources for the motor and the circuit for isolation. If the system us Note that if you are planning to use a supply voltage of more than +12V for the L298N module, you need to remove the jumper just above the +12V input.

The self-balancing robot is not new, but when we started this research project we found a lot of information, but never in the same place, we had to search a lot to join all information in a single research project. The materials and electronics used in the research project, was a servo motor for a good results and also we can user stepper motor for the same. The chassis is made from the plastic with three flours each flour has different components which has to be in the same order for better results. We user L293 for driving the motors[10].The physics behind this is very simple, the robot stand in two points lined, the wheel, and it tends to fall and lose his verticality, the movement of the wheel in the direction of the falling rises the robot for recover the vertical position.

A Segway-type vehicle is a classic inverted pendulum control problem that is solvable in two degrees of freedom for the simplest models. The vehicle attempts to correct for an induced lean angle by moving forward or backwards, and the goal is to return itself to vertical. Or at least not fall over.For that objective we have two things to do, in one side of it we have to measure the angle of inclination (Roll) that have the vehicle, and in the other hand we have to control the motors for going forward or backwards to make that angle 0, maintaining his verticality. The concept is implemented using the radio transceivermodule [4]. Emphasis being given to presentation of idea. The implementation is based around the well-known microcontroller originally designed by Intel but the chips that we are using were manufactured by the Atmel corp. Although any other microcontroller could be used without any major change. The only direct impact will be only in the software or the assembly code written for the particular microcontroller. The system uses the simple transmitter section designed using the USART [3].

IV. Angle Measurement

Here for measure the angle we have two sensors, accelerometer and gyroscope, both have his advantages and disadvantages. The accelerometer can measure the force of the gravity, and with that information we can obtain the angle of the robot, the problem as we have seen in the accelerometer is that it can also measure the rest of the forces the vehicle is summited, so it has lot of error and noise. The gyroscope is used to measure the angular velocity, so if we integrate this measure we can obtain the angle the robot is moved, the problem of this measure is

that is not perfect and the integration has a deviation, that means that in short time the measure is so good, but for long time terms the angle will deviate much form the real angle.Those problems can be resolved be the combination of both sensors, that's called sensor fusion, and there are a lot of methods to combine it. In this research project we have tried two of them: Kalman Filter, and complementary filter.

The Kalman filter is an algorithm very extended in robotics, and offers a good result with low computational cost. The library is used for arduino that implements this method.

The Complementary filter is a combination of two or more filters that combines the information from different sources and gets the best value you want.

angle = A * (angle + gyro * dt) + (1 - A) * acceleration;where A is normally equals to 0.98.

First we tried to use Kalman filter but the results obtained was not up to the mark, the angle was calculated with a little delay and it affect the control.

The Kalman filter has three variables and can be changed based on the parameter of your sensor, and varying this can obtain better result, the values were changed to get the better results but was not good and we tried to implement the same using complementary filter. The main structure of the robot was created using some plastics materials for chassis with some nuts and screws, two wheels with two motors, one battery socket, one caster wheel, and even 2 little wheels for encoders.

The two motor were placed in the lower part of the structure and closed it with the two plastic parts, keeping it together with the screws. The electronics and the battery creates a tower in the upper part of the structure.The electronics were used here are arduino UNO, a motor driver, in this case a L298 was used, here we use a commercial motor driver based on the chip L298, maybe much powerful that we need for these motors but we have it and it works fine. If a DC motor has to be used, then a DC motor driver that is a H-bridge or use a L293. For the IMU we used 10DOF that is GY-80 with 3-axis accelerometer, 3-axis gyroscope, magnetometer, barometer and temperature sensors. We use only accelerometer and gyro. The IMU is connected to the Arduino using I2C bus, so we need 2 wires for communication (SDA and SCL) and 2 wires for power, it uses 3.3V so we need 3.3V wire and GND. The motor driver take power directly form the battery so don't have to connect arduino's power to it, but we need 6 wires to control it, 3 for each motor, one for send the PWM signal for control the motor velocity, and for indicate the direction we want the motor to spin. The code has 4 files: one the main code, a second one for the motors, the third is for the PID, and the last one is for the sensor code.

In the main code is to initialize the entire robot which in turn communicates with the sensors and other devices connected to it. The sensor errors are calculated and modified to get the better results. The initial angle is found and made to zero, it means that the sensor has an initial deviation, when we place the robot vertically the sensor doesn't show that the angle is zero, instead send a deviation angle, this initial angle is used to subtract it from the posterior measurements of the sensors, to obtain the real angle. So when we initiate the robot we have to maintain it vertically until it starts to move the wheels.

The next part of code is the loop where we take the sensor values every 10 milliseconds, the frequency of sampling is 100Hz, this value is chosen as very low and very high frequencies could not work, and we calculate the angle of the robot using the complementary filter previously explained. The angle is known and now we can use that information to control the motors, this uses an intermediate PID, the simplest way to control things efficiently. In order to use the accelerometer, in this case the ADXL345, the relative library is used for this case.

V.System Analysis

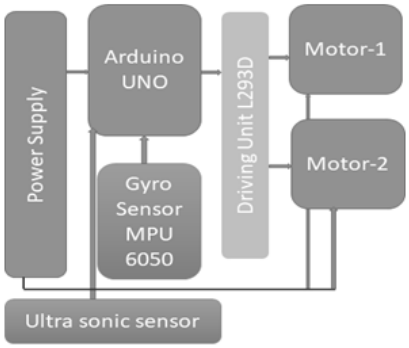

Fig. 1: Block diagram ofSelf balanced robot

To keep the robot balanced, the motor should offset the fall of the robot. This action requires a feedback and a revision section. The feedback component is the MPU6050 Gyroscope + Accelerometer, which provides each of the three axes used by the Arduino for each acceleration and

rotation to understand this direction of the robot. Correction part is a combination of motor and wheel.

The sensor connection to the Arduino and the test connection were done using the proteus design tool before using the hardware implementation. Arduino is a joint open supply component and packaging company, project and user community used to change and manufacture single-board microcontrollers and microcontroller suites for building digital devices and interactive objects for sensing and managing objects in the physical world. The project's products are released in ASCII text file hardware and software packages licensed under the GNU General Public License (LGPL) or the GNU General Public License (GPL), allowing anyone to make Arduino boards and software distributions. The Arduino board comes preinstalled.

The Arduino board style uses a series of microprocessors and controllers. These boards come with a set of digital and analog input / output (I / O) pins that will connect multiple expansion boards (shields) and different circuits. These boards have a serial communication interface, as well as Universal Serial Bus (USB) on some models, and are also used to load programs from a personal computer. Microcontrollers often use functional programming from the programming languages C and C ++. In addition to leveraging the traditional compilation toolchain, the Arduino project provides an integrated development environment (IDE)

VI.Basic Concept &The Implementation Of The System

Using an accelerometer to measure dip, the MPU6050 has a 3-axis accelerometer and a 3-axis gyroscope. The accelerometer measures the acceleration of three axes and the gyroscope measures the angular velocity of the three axes. In order to measure the tilt angle of the robot, we need the acceleration values along the y-axis and the z-axis. The atan2 (y, z) function gives the angle of curvature between the positive z-axis of the plane and the point given by the coordinates (z, y) on the plane, with the counterclockwise angle being a positive sign), The minus sign indicates the clockwise angle (left half plane, y <0). We use the library written by Jeff Rowberg to read the MPU6050's data. Upload the code given below to see how the dip changes.

Using a gyroscope to measure the tilt angle, the MPU6050's position is zero tilt when the program starts running. The angle of inclination is measured relative to this point. Keep the robot stable at a fixed angle, you will find the angle will gradually increase or decrease. It will not stay stable. This is due to the inherent drift of the gyroscope. In the code given above, the cycle time is calculated using the millis () function built into the Arduino IDE. In later steps, we will use timer interrupts to create accurate sample intervals. The sampling period will also be used to generate the output using the PID controller.

VII. The Result Is Combined With A Supplemental Filter

We have two metrics from the perspective of two different sources. The measurement from the accelerometer is affected by a sudden horizontal movement and the measurement from the gyro gradually deviates from the actual value. In other words, accelerometer readings are affected by short duration signals and the gyroscopes read signals over long periods of time. These books complement each other to some extent. Using complementary filters to combine them, we get stable, accurate angular measurements. The complementary filter is essentially a high-pass filter acting on the gyroscope and a low-pass filter acting on the accelerometer to filter out the drift and noise in the measurement. currentAngle = 0.9934 * (previousAngle + gyroAngle) + 0.0066 * (accAngle) 0.9934 and 0.0066 are filter coefficients with a filter time constant of 0.75s.

A low-pass filter allows any signal longer than this duration to pass, and a high-pass filter allows any signal that is shorter than this duration to pass. The filter's response can be adjusted by choosing the correct time constant. Lowering the time constant will allow more horizontal acceleration to pass. Eliminate accelerometer and gyroscope offset errors. By defining the offset value in the setup () routine, any errors due to offsets can be eliminated.

VIII. Experimental Analysis

Here the PID stands for Proportional, Integral and Derivative. Each of these terms provides a unique answer to our self-balancing robot. As the name suggests, the proportional term produces a response proportional to the error. For our system, the error is the robot's tilt angle. The integral term generates a response based on the cumulative error. This is essentially the sum of all errors multiplied by the sampling period. This is a response based on past

system behavior. Derivative terms are proportional to the derivative of the error. This is the difference between the current error and the previous error divided by the sampling period.

This serves as a predictive term that responds to the robot's behavior in the next sampling cycle. Multiplying these terms by their corresponding constants (ie, Kp, Ki, and Kd) and adding the results yields an output that is then sent as a command to drive the motor. The ultrasonic distance sensor we use is US-020. It has four pins, Vcc, Trig, Echo and Gnd. It is powered by a 5V power supply. The trigger and echo pins are connected to the Arduino's digital pins 9 and 8, respectively. We will use the NewPing library to get the sensor's distance value. We will read the distance every 100 milliseconds, and if the value is between 0 and 20 centimeters, we will order the robot to perform a rotation. This should be enough to keep the robot away from obstacles.

The translation of kinetic energy of wheels in the form as shown below,

$$T_1 = \frac{1}{2} M_{RL} X^2_{RL} + \frac{1}{2} M_{RR} X^2_{RR} \quad \text{....(1)}$$

The rotation kinetic energy of two wheels is in the form of

$$T_2 = \frac{1}{2} J_{RL}\theta^2_{RL} + \frac{1}{2} J_{RR}\theta^2_{RR} \quad \text{.....(2)}$$

The translation kinetic energy of body's center of mass is

$$T_3 = \frac{1}{2} M_P ((\theta_P L \cos\theta_P + X_{RM})^2 + (-\theta_P L \sin\theta_P)^2) \quad \text{.....(3)}$$

The rotation kinetic energy of body round the axis which is through the center of mass and parallel to z axis is

$$T_4 = \frac{1}{2} J_{P\theta}\, \theta_P^2 \quad \text{.....(4)}$$

The rotation kinetic energy of body round y axis is shown in

$$T_5 = \frac{1}{2} J_{P\theta}\, \delta^2 \quad \text{.... (5)}$$

As a result of total kinetic energy of the entire system is calculated as

$$T = T_1 + T_2 + T_3 + T_4 + T_5 = \frac{1}{2} M_{RL} X^2_{RL} + \frac{1}{2} M_{RR} X^2_{RR} + \frac{1}{2} J_{RL}\theta^2_{RL} + \frac{1}{2} J_{RR}\theta^2_{RR} + \frac{1}{2} M_P ((\theta_P L \cos\theta_P + X_{RM})^2 + (-\theta_P L \sin\theta_P)^2) + \frac{1}{2} J_{P\theta}\, \theta_P^2 + \frac{1}{2} J_{P\theta}\, \delta^2 \quad (6)$$

Then the mathematical model of two wheels and body system is obtained by subtracting the total kinetic energy, generalized coordinate and generalized force into the Lagrange's equation. Then the equations will be in the form of

$$(M_{RL} R^2 + J_{RL} + \frac{1}{4} M_P R^2 + \frac{R^2}{D^2} J_{P\delta})\, \theta_{RL} + (\frac{1}{4} M_P R^2 + \frac{R^2}{D^2} J_{P\delta})\, \theta_{RR} + \frac{1}{2} M_P R L \theta_P = C_L \quad (7)$$

$$(M_{RR} R^2 + J_{RR} + \frac{1}{4} M_P R^2 + \frac{R^2}{D^2} J_{P\delta})\, \theta_{RR} + (\frac{1}{4} M_P R^2 + \frac{R^2}{D^2} J_{P\delta})\, \theta_{RL} + \frac{1}{2} M_P R L \theta_P = C_R \quad (8)$$

$$\frac{1}{2} M_P R L \theta_{RL} + \frac{1}{2} M_P R L \theta_{RR} + (J_{P\theta} + M_P L^2)\, \theta_P = M_P g L \theta_P - (C_L + C_R) \quad (9)$$

XI. Simulation Results

Fig. 2: Simulation result for two motors.

Fig. 3: Simulation result output for a motor.

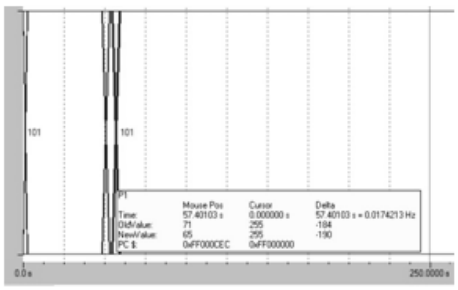

Fig. 4: Simulation result for sensor.

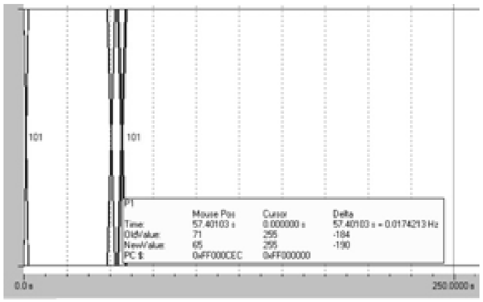

Fig. 5: Simulation result output for sensor.

The robot with LQR maintains the robot in a balanced way, the energy consumption is

$J=\int_0^{\infty}(X^T Qx + u^T Ru)d_t$(10)

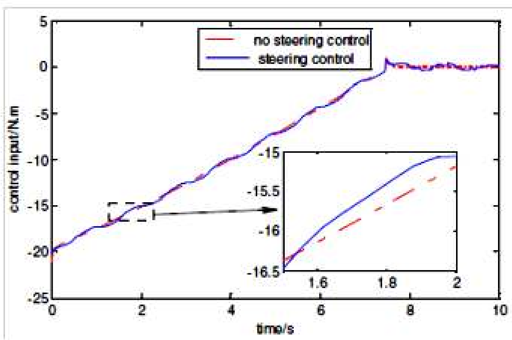

Fig. 6: Response cure of the robot

The response cure shows how much the system deviates from the present value when the it moves from the present state to the next state.

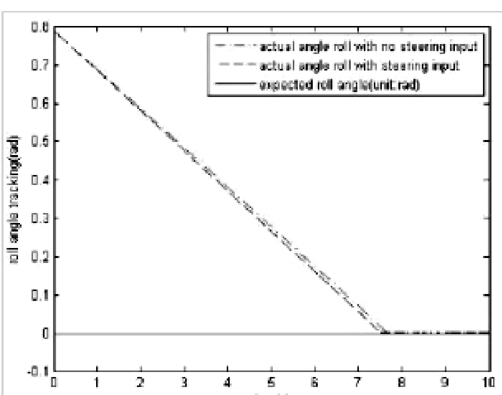

Fig. 7: Time response of the angle

The low pass filter on acceleration,

$\theta_{accel(filt)}=\frac{\tau}{\Delta t+\tau}\ \theta_{accel(filt)(t-1)}+\frac{\Delta t}{\Delta t+\tau}\ \theta_{accel}$....(11)

let $\frac{\tau}{\Delta t+\tau}=\alpha$ and $\frac{\Delta t}{\Delta t+\tau}=1-\alpha$

$\theta_{accel(filt)}=\ \alpha\ (\theta_{accel(filt)(t-1)}) + (1-\alpha)\ \theta_{accel}$ ....(12)

Fig. 7: Time response of the angle.

$\theta_{gyro(filt)} = \frac{\tau}{\tau+\Delta t}\ (\theta_{gyro}-\ \theta_{gyro(t-1)}) +$

$\frac{\tau}{\tau+\Delta t}\theta_{gyro(filt)(t-1)}$.............................(13)

$\theta_{gyro(filt)} =\ \alpha\ (\theta_{gyro}-\ \theta_{gyro(t-1)}) + \alpha\ \theta_{gyro(filt)(t-1)}$

The filtered values obtained from both sensors in (12) and (13) are summed together to provide more reliable estimation of the robot tilt angle which yields:

$\theta_{filt} =\ \theta_{accel(filt)}+ \theta_{gyro(filt)}$.......................(14)

$\theta_{filt}=\ \alpha\ (\theta_{accel(filt)(t-1)}+ \theta_{gyro(filt)(t-1)}+ \theta_{gyro}.\Delta t)$

$+ (1-\alpha)\ \theta_{accel}$..(15)

$\theta_{filt}= \alpha(\theta_{filt(t-1)}+ \theta_{gyro}.\ \Delta t) + (1-\alpha)\ \theta_{accel}$

For the error of pitch and roll angle as follows:

$\varphi_{err}= \varphi_{meas} - 0^0$ and $\theta_{err}= \theta_{meas} - 0^0$

The algorithm for the control system is hence:

$V_{\varphi}=\ V_{\varphi(t-1)} + K_p\ (\varphi - \varphi_{t-1}) + K_I\ (\varphi) + K_D\ (\varphi\ -2\ \varphi_{t-1} + \varphi_{t-2})$ ....(16)

$V_{\theta} = V_{\theta(t-1)} + K_p\ (\theta - \theta_{t-1}) + K_I\ (\theta) + K_D\ (\theta\ -2\ \theta_{t-1} + \theta_{t-2})$ ....(17)

ө expected

ө actual

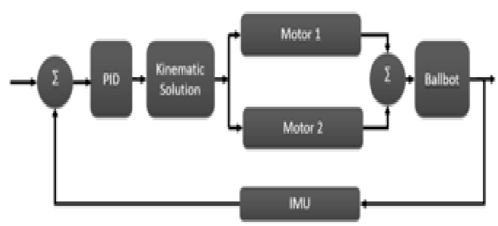

Fig. 8: Robot balancing control block diagram.

ω expected

ω actual

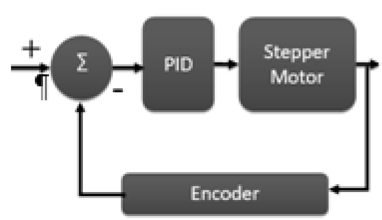

Fig. 9: Robot motor speed control block diagram.

Fig. 10: Prototype Model

X. Conclusion

Experimental results show that the developed control strategy allows the robot to maintain its balance dynamically. In addition, robots are also able to recover from the tilt caused by external disturbances. By using PID and improved LQR control algorithm, the improved LQR has better performance than PID in robot self-balancing. Future work will focus on higher speed motions and higher spatial precision motion control. More and more Will develop more useful features.

References

[1]. Niranjan L, Sushma S, Shreekanth R, Santhoshilakshmi G S "Design and Implementation of Security Based Automatic Teller Machine against Intruders", in International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE), ISSN 2394-6849, Vol 4, Issue 6, Page 260-266, June 2017.

[2]. Suhas A.R, H L Viswanath "A Comparative Study between Extended Kalman filter and Unscented Kalman filter for Traffic State Estimation", published in IJIET Journal, volume 4 Issue 1-June 2014, pp- 188-197.

[3]. Niranjan L, Nethravathi V, Bhavya Shree G, Nethravathi G, Rithu Shah "Home Automation using SCADA & IOT" in International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE), ISSN 2394-6849, Vol 4, Issue 6, Page 335-344, June 2017.

[4]. Suhas A.R, Asha K K, Satyanarayana "Multi-Stage Auv-Based Localization For Underwater Wireless Sensor Network", published in JEST-M Journal, volume 4 Issue 4, Jan 2015, pp-14-19.

[5]. Niranjan L, Charmila G, Bindhu V, Jyothi M, Meghana M "A Novel Approach to Wireless Power Transmission in Non-Radiative Field" in International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE), ISSN 2394-6849, Vol 4, Issue 6, Page 304-310, June 2017.

[6]. Suhas A.R, Arun T V "Substation Automation Systems", published in IJSER Journal, volume 7 Issue 05, May 2016, pp-215-218.

[7]. Niranjan L, Sreekanth B, Suhas A R, Mohan Kumar B N "Advanced System for Driving Assistance in Multi-Zones using RF and GSM Technology" in International Journal of Engineering Research & Technology (IJERT), ISSN: 2278-0181, Vol. 3 Issue 6, June – 2014.

[8]. Chandrakumar H S, Tejaswini M R, "Robust face detection and person recognition using gabor texture and multi block LBP features", published in IJSER Journal, volume 7 Issue 05, pp-210-214, May-2015

[9]. Chandrakumar H S, Sadika, Supritha, Meenu Gaur "Establishing High Transmission Bandwidth from Service Providers through SDH" in International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE), ISSN 2394-6849, Vol 4, Issue 6, Page 314-318, June 2017.

[10]. Sreekanth B, [2] MeesalaPuneeth, [3] Divya A T, [4] Savitha J S, [5] Pavithra R "Robotic Four Wheeler for Health Monitoring" in International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE), ISSN 2394-6849, Vol 4, Issue 6, Page 280-287, June 2017.