2014 International Conference on Robotics and
Emerging Allied Technologies in Engineering (iCREATE)
Islamabad, Pakistan, April 22-24, 2014

Modeling, Control of a Two-Wheeled Self-Balancing Robot

Osama Jamil*, Mohsin Jamil*, Yasar Ayaz*, Khubab Ahmad**

*School of Mechanical and Manufacturing engineering
National University of Sciences and Technology
Islamabad, Pakistan
osama.jamil@live.com
mohsin@smme.nust.edu.pk
yasar@smme.nust.edu.pk
**University of Engineering and Technology
Taxila, Pakistan
khubab@hotmail.com

***Abstract*— Two wheeled self-balancing robot is based on an inverted pendulum system. This paper describes the development of kinematics model of a two wheeled self-balancing robot and its control using various control techniques. The analysis of the developed kinematic model can be divided into wheel and frame structure, after one is done with design and development of mechatronics system is finalized. The said test platform is two Degree of Freedom highly non-linear and unstable system. Main focus of this research is to develop an efficient controller required to enable the robot to act in real time and test different controllers to analyze their performance by the vertical balance, position, its control signal and disturbance rejection capabilities.**

***Index Terms*—Self-balancing robot, PID, LQR.**

I. INTRODUCTION

Optimization of Control System for a two-wheeled self-balancing robot has been a hot area of research for the past few years, and the main reason for that is its nonlinear dynamics. The inherent complexity associated with the control of this platform makes it the most appropriate test platform for the design and development of control systems of automobiles, missiles, space crafts, robot, and even military transport. The design can also be used to develop a personal transport vehicle and would prove to be very useful as it would allow people to travel small distances on a small mode of transport and would help reduce air pollution. One such machine developed by Dean Kamen is known as SEGWAY, now commercially available in the market. A cost effective solution is presented using Kalman filtering and PID algorithm for a two wheeled car [1].

Scientists and engineers are working to develop techniques to make a dynamically stable and robust solution; various techniques are applied and tested on this platform. Pole placement yields less overshoots but the settling time is high [2]. LQR yields good settling time but more overshoots than pole placement [2]. Another attempt for mechanically stable chassis made using LEGO bricks which reduces mechanical design complexity considerably and tested for PD and PI controllers where PI control is showing overshoots and oscillations, this sort of solution eases the design complexity but is not suitable for a real time application [3]. Hybrid controllers are also designed to achieve the desired performance; the pros of PID and LQR are combined to develop a solution [4]. Wireless control added to move it in four directions using real time tuned kalman filter and PID [5].

The kinematics model divided into body and wheel model has been designed using velocity decomposition method, tested by ADAMS simulation analysis [6].

Analysis has been made to study the performance of various controllers, using robust control [7], simple solution for an inverted pendulum mounted on a car using PD controller and single potentiometer as a sensor [8], LQR and Dual PID for input tracking and disturbance rejection [9]. [10] presents a balancing robot with PID algorithm to vertically balance itself only.

Nbot, a two wheeled self-balancing robot built by David P. Anderson uses inertial sensor and senses position using the built-in motor encoders, Legway a balancing robot was created successfully by Steven Hassenplug using the Lego Mindstorm kit.

With the increase in percentage of aged people over young ones in some countries like Japan has provide the researchers to aid them in moving around using wheel chairs when it passes a road curb or takes a small step, control system for inverted pendulum is required, this was made by Takahashi [11].

In this paper, first of all inverted pendulum system modeling is done as it resembles the system. Then, system mathematical equations are equated and linearized. Dual-PID and LQR control techniques are designed and tested in Simulink and analyzed for vertical balance, control signal, position control and disturbance rejection capabilities of both the controllers the necessary requirements for making this system a mobile robot.

II. INVERTED PENDULUM SYSTEM

The Inverted pendulum is a highly unstable and non-linear system; it's one of the complex systems to be controlled in control engineering due to its non-linear dynamics. Inverted pendulum system is studied in detail. Following is the model of an inverted pendulum system with cart.

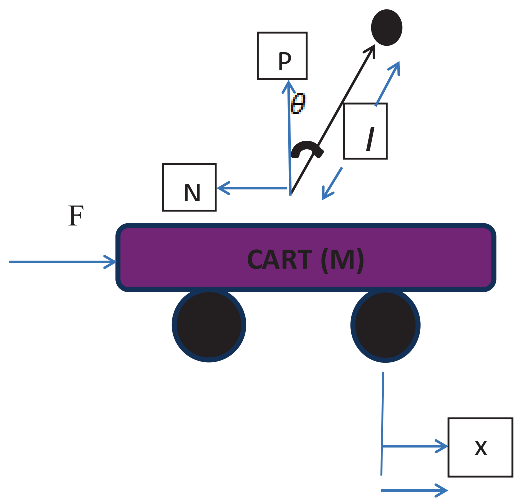


Fig. 1. Inverted pendulum system with cart

The pendulum will simply fall over the cart is not moved to balance it. The mathematical model obtained is in the form of non-linear equations

$$M\ddot{x} + b\dot{x} + m\ddot{x} + ml\ddot{\theta}\cos\theta - ml\dot{\theta}^2\sin\theta = F \qquad (1)$$

$$I\ddot{\theta} + ml^2\ddot{\theta} + mgl\sin\theta = -ml\ddot{x}\cos\theta \qquad (2)$$

The non-linear equations in the linear form

$$(I+ml^2)\ddot{\varphi} - mgl\varphi = ml\ddot{x} \qquad (3)$$

$$(M+m)\ddot{x} + b\dot{x} - ml\ddot{\varphi} = u \qquad (4)$$

State space model was obtained from the equations

$$\begin{bmatrix}\dot{x}\\ \ddot{x}\\ \dot{\varphi}\\ \ddot{\varphi}\end{bmatrix} = \begin{bmatrix}0 & 1 & 0 & 0\\ 0 & \frac{-(I+ml^2)b}{I(M+m)+Mml^2} & \frac{m^2gl^2}{I(M+m)+Mml^2} & 0\\ 0 & 0 & 0 & 1\\ 0 & \frac{-mlb}{I(M+m)+Mml^2} & \frac{mgl(M+m)}{I(M+m)+Mml^2} & 0\end{bmatrix}\begin{bmatrix}x\\ \dot{x}\\ \varphi\\ \dot{\varphi}\end{bmatrix} + \begin{bmatrix}0\\ \frac{I+ml^2}{I(M+m)+Mml^2}\\ \frac{ml}{I(M+m)+Mml^2}\\ 0\end{bmatrix}u \qquad (5)$$

$$y = \begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 0 & 1 & 0\end{bmatrix}\begin{bmatrix}x\\ \dot{x}\\ \varphi\\ \dot{\varphi}\end{bmatrix} \qquad (6)$$

Using all the equations and the state space model, a MATLAB simulation yielded a result which we were aware about that the system is unstable without any control

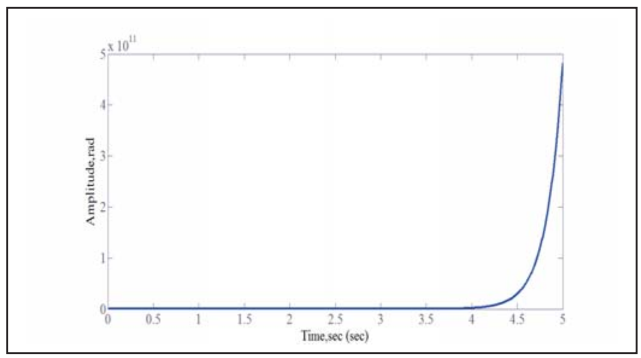


Fig.2 Impulse response of Inverted pendulum without any control

III. SYSTEM MODELING

The dynamics of the robot are described by the mathematical model in order to facilitate the development of a control system for balancing the robot.

In order to develop an efficient control for this system mathematical model is very important.

System modeling is divided into main three parts as follows:

- Linear Model of DC motor.
- Mathematical modeling and linearization of two wheeled inverted pendulum (robot).
- Integration with dynamic model of DC motor

A. Linear model of a DC motor:

The robot uses two DC motors to drive the wheels.

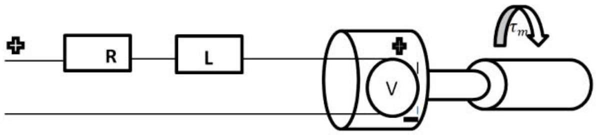


Fig.3 Circuit of DC Motor

The equation obtained

$$\frac{d\omega}{dt} = \frac{k_m}{I_R}\left(-\frac{k_e\omega}{R} + \frac{V_a}{R}\right) - \frac{\tau_a}{I_R} \tag{7}$$

Equations were also obtained in the matrix form

$$\begin{bmatrix} \dot{\theta} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & \frac{-k_m k_e}{I_R R} \end{bmatrix} \begin{bmatrix} \theta \\ \omega \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \frac{k_m}{I_R R} & -\frac{1}{I_R} \end{bmatrix} \begin{bmatrix} V_a \\ \tau_a \end{bmatrix} \tag{8}$$

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \omega \end{bmatrix} + \begin{bmatrix} 0 & 0 \end{bmatrix} \begin{bmatrix} V_a \\ \tau_a \end{bmatrix} \tag{9}$$

This model is integrated with the dynamic model of the balancing robot to provide us with input to output relationship, which is used to obtain the state space model.

B. Model for the robot:

The two wheeled inverted pendulum albeit more complex in system dynamics has sort of the similar behavior with a pendulum on a cart. The pendulum and the wheel dynamics are analyzed separately at the beginning but this will eventually lead to two equations which will describe the behavior of the robot.

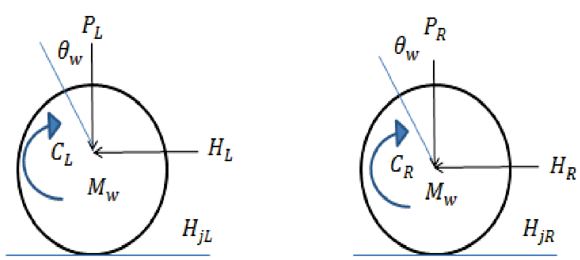


Fig.4. Wheels of the robot

Equations for the left and right wheels are determined by using the above figure.

For Left Wheel:

$$M_\omega \ddot{x} = \frac{-k_m k_e}{Rr^2}\dot{x} + \frac{k_m}{Rr}V_a - \frac{I_\omega}{r^2}\ddot{x} - H_L \tag{10}$$

For Right Wheel:

$$M_\omega \ddot{x} = \frac{-k_m k_e}{Rr^2}\dot{x} + \frac{k_m}{Rr}V_a - \frac{I_\omega}{r^2}\ddot{x} - H_R \tag{11}$$

Both equations are combined

$$2\left(M_\omega + \frac{I_\omega}{r^2}\right)\ddot{x} = \frac{-2k_m k_e}{Rr^2}\dot{x} + \frac{2k_m}{Rr}V_a - (H_L + H_R) \tag{12}$$

C. Chassis model:

The robot's chassis is similar to the inverted pendulum design so it has been modeled accordingly; figure below is the free body diagram of chassis design.

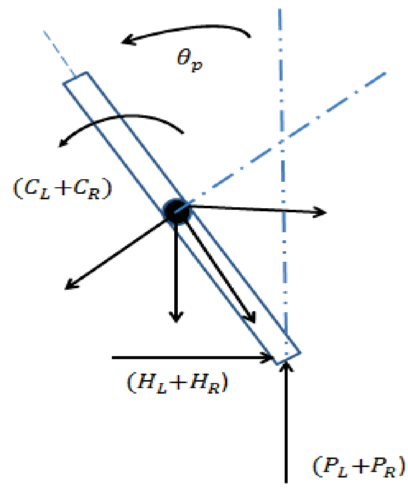


Fig.5. Free body diagram of the chassis

After calculation of all the equations of the chassis , this model is integrated with the DC motor model to get the following equations

$$(I_p + l^2 M_p)\ddot{\theta}_p - \frac{2k_m k_e}{Rr}\dot{x} + \frac{2k_m}{R}V_a + M_p g l \sin\theta_p = -M_p l \ddot{x}\cos\theta_p \tag{13}$$

$$\frac{2k_m}{Rr}V_a = (2M_w + \frac{2I_w}{r^2} + M_p)\ddot{x} + \frac{2k_m k_e}{Rr^2}\dot{x} + M_p l \ddot{\theta}_p \cos\theta_p - M_p l \theta_p^2 \sin\theta_p \tag{14}$$

The equation obtained are non-linear so the equations must be linearized. assume that $\varphi_p = \pi + \theta$, hence linearized equations are obtained

$$\ddot{\varphi} = \frac{M_p l}{(I_p + M_p l^2)}\ddot{x} + \frac{2k_m k_e}{Rr(I_p + M_p l^2)}\dot{x} - \frac{2k_m}{R(I_p + M_p l^2)}V_a + \frac{M_p g l}{(I_p + M_p l^2)}\varphi \tag{15}$$

$$\ddot{x} = \frac{M_p l}{\left(\frac{2I_w}{r^2} + M_p + 2M_w\right)}\ddot{\varphi} - \frac{2k_m k_e}{Rr^2\left(\frac{2I_w}{r^2} + M_p + 2M_w\right)}\dot{x} + \frac{2k_m}{R\left(\frac{2I_w}{r^2} + M_p + 2M_w\right)}V_a \tag{16}$$

The above equations are the two main governing equations of the system.

D. State Space model:

State space model is obtained using the above two main governing equations

$$\begin{bmatrix}\dot{x}\\ \ddot{x}\\ \dot{\varphi}\\ \ddot{\varphi}\end{bmatrix} = \begin{bmatrix}0 & 1 & 0 & 0\\ 0 & \frac{2k_m k_e(M_p lr - I_p - M_p l^2)}{Rr^2\alpha} & \frac{M_p g l^2}{\alpha} & 0\\ 0 & 0 & 0 & 1\\ 0 & \frac{2k_m k_e(r\beta - M_p l)}{Rr^2\alpha} & \frac{M_p g l\beta}{\alpha} & 0\end{bmatrix}\begin{bmatrix}x\\ \dot{x}\\ \varphi\\ \dot{\varphi}\end{bmatrix} + \begin{bmatrix}0\\ \frac{2k_m(-M_p lr + I_p + M_p l^2)}{Rr\alpha}\\ 0\\ \frac{2k_m(-r\beta + M_p l)}{Rr\alpha}\end{bmatrix}u \tag{17}$$

$$y = \begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 0 & 1 & 0\end{bmatrix}\begin{bmatrix}x\\ \dot{x}\\ \varphi\\ \ddot{\varphi}\end{bmatrix} \tag{18}$$

$$\beta = 2M_w + \frac{2I_w}{r^2} + M_p$$

$$\alpha = I_p\beta + M_p l^2\left(M_w + \frac{I_w}{r^2}\right)$$

IV. Control Techniques

To stabilize this system implementation of a control technique is required. Using the control technique (PID,LQR etc) the controller will be synthesized.

A. Proportional integral derivative (PID)

PID stands for proportional integral derivative. In PID, error value is calculated as the difference between a reference point and the actual output. The error is minimized by adjusting the process control inputs.

The PID algorithm involves three constant parameters. The proportional, the integral and derivative, denoted by P, I, and D.The functions P,I and D can be described as:

'P' Factor

1. Based on current rate of change, depends on present or current errors.
2. The product of gain and measured error.
3. large gain has fast response time and small steady state error
4. Causes overshoots.

'I' factor:

1. Based on current rate of change, depends on the history of errors.
2. Eliminates the steady state error.
3. Product of gain and summation of past errors.
4. Causes overshoots

'D' factor:

1. Based on current rate of change depends on the prediction of future errors.
2. Product of gain and rate of change of error.
3. Reduces the overshoots caused by proportional and integral factor.
4. Increases noise.
5. Too high gain will cause the system unstable.

The figure below describes a PID loop:

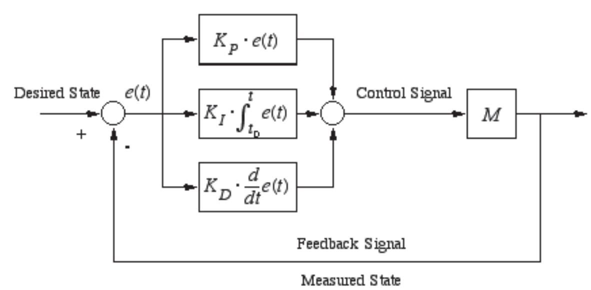


Fig.6. PID loop

B. Linear Quadratic Regulator (LQR)

The problem of linear quadratic regulator control restricts our attention to linear systems or linearized non-linear systems and selection of a cost function which is a quadratic function of states and controls. The formulation of problem follows.

The linear system is given by its state space representation

$$\dot{x} = Ax + Bu$$
$$y = Cx \quad (19)$$

Find a control function u that will minimize the cost function J

$$J = \frac{1}{2}\int_{0}^{\infty}\left(x^T Qx + u^T Ru\right)dt \quad (20)$$

Q matrix includes the states and their weights whereas R is a scalar matrix.

Algebraic Ricatti Equations (ARE) to calculate the gains and minimizing the cost function

$$A^T P - PA + Q - PBR^{-1}B^T P = 0 \quad (21)$$

$$u = -Kx \quad (22)$$

$$K = R^{-1}B^T P \quad (23)$$

V. SIMULATIONS

A close-loop system is also called a feedback system, as it computes its input into a system using previous state and its model of system. This type of system has check on its output whether the results are achieved or not, any errors are present. This system corrects errors and can tolerate any disturbance occurred in the system. Controller efficiency will be tested on its vertical balance, position control and disturbance rejection capability.

Table I shows the specification of the robot that will be used for simulations.

TABLE I

1	Gravitational acceleration (meters/sec^2)	g =9.81
2	Wheel radius (meters)	r=0.077
3	Wheel mass (kilogram)	Mw=0.3
4	Body mass (kilogram)	Mp= 6
5	Wheel inertia (kilogram*meter^2)	Iw=0.0017
6	Body inertia (kilogram*meter^2)	Ip=0.29
7	Distance from body's center of mass (meter)	l=0.2
8	Motor torque (Nm/Amp)	Km = 0.0458
9	Back EMF (Vs/radians)	Ke = 0.0458
10	Terminal Resistance (Ω)	R = 2.49

A. Dual PID control

Dual PID controller is used to control this system i.e. one PID for balance control and another PID to control the motor speed and position of the robot.

The system is given by the following figure

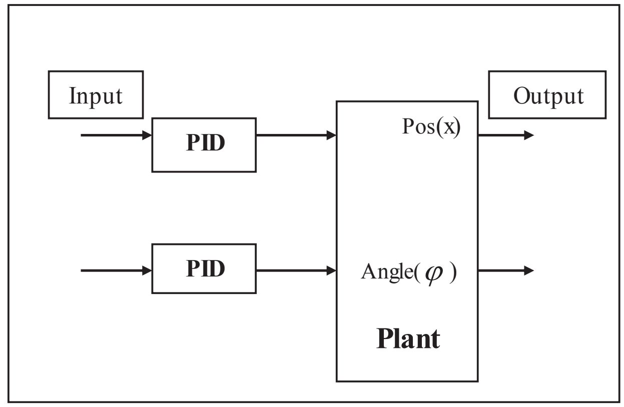


Fig.7. Dual PID control system

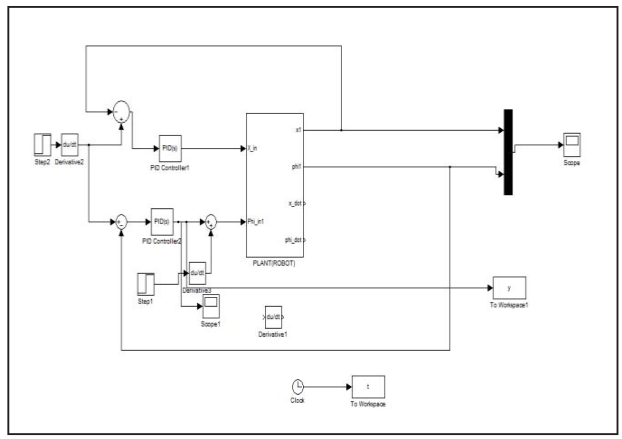


Fig. 8. Simulink model for PID-PID control

Fig 8 shows the Simulink model for the PID-PID control simulation.

First, P-P controller is applied to the closed loop system. Proportional controller is a linear feedback controller. In this the controller output is proportional to the error signal. In simple words the output is the multiplicative product of error signal and proportional gain.

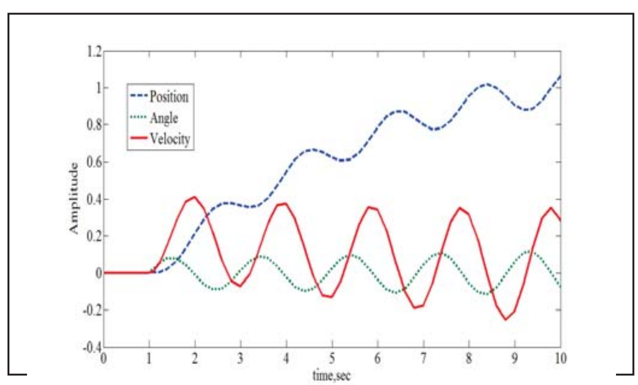


Fig.9. Output with P-P controller

As it can be observed from the figure the response, after application of proportional (P-P) controller, the output is now oscillating and the dynamics of the system does not allow it to stabilize which means we need to add another factor into the PID controller.
The next controller applied is P-PI

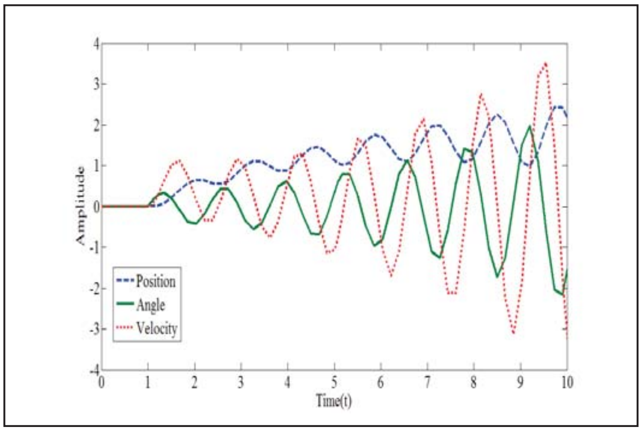


Fig. 10. Output with P-PI controller

Now the output response is still not desirable and overshoots increase overtime which is not a very favorable result for our system. Integral controller eliminates steady state error but also causes overshoots which is highly undesirable for this system.

Now, we will remove the I (integral) factor which is causing the overshoots to increase over time. Testing the system with P-PD controller

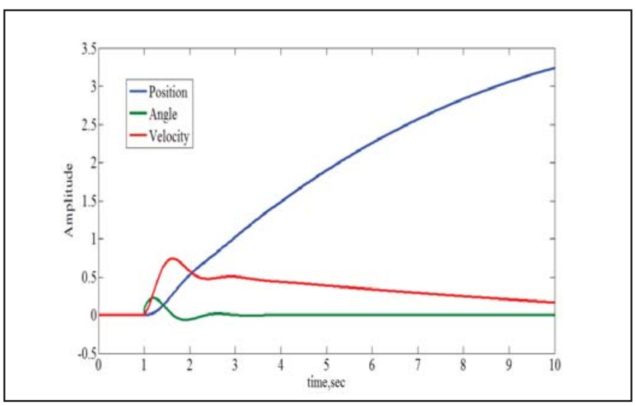
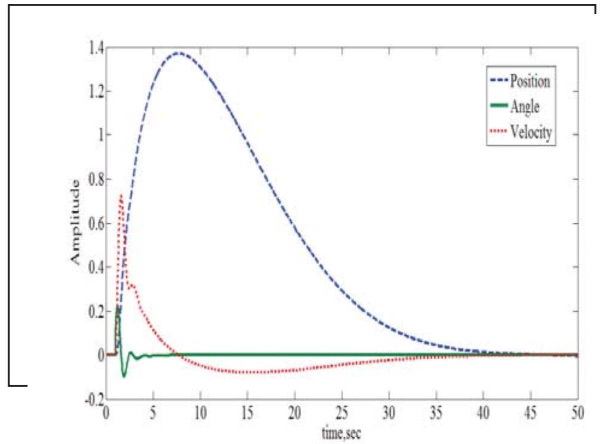


Fig.11. Output with P-PD controller

Observing the behavior of the system using PD controller from the graph above that the overshoots are vanished and the system is stable hence a the PD controller is showing some promising results in terms of balancing it vertically but the position control is still not achieved.

In an attempt to control the position we will now test the system with PID-PID controller.



From all the simulated controllers this one showed the best performance compared with all the controllers applied. But analyzing the output we can observe that the response is very slow for the position control.

Now, testing the system with impulse disturbance to test the disturbance rejection capabilities of the controller.

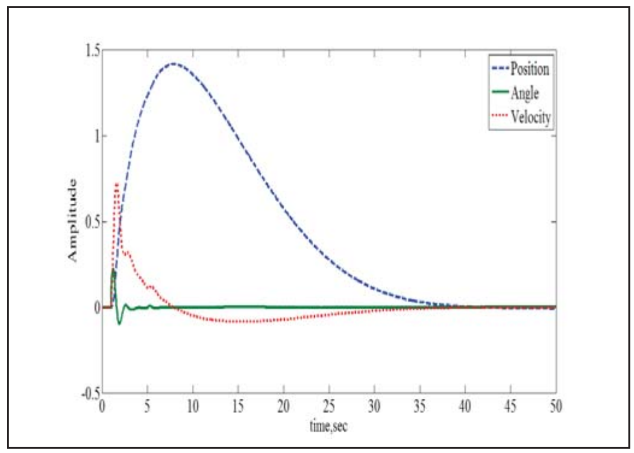


Fig.13 Output with Impulse disturbance

The response is not much affected with the addition of disturbance at 5 sec.

Now checking the control signal provided to the system.

The control signal shows an impulse of very high amplitude, for which there might not be a real time solution available.

PID control is easy to implement but it has its limitations. The high gains of P, I and D factors results in a non-practical solution.

B. Linear Quadratic Regulator (LQR)

Now we will simulate system for LQR control, the gain matrix K was calculated using MATLAB and then it was simulated.

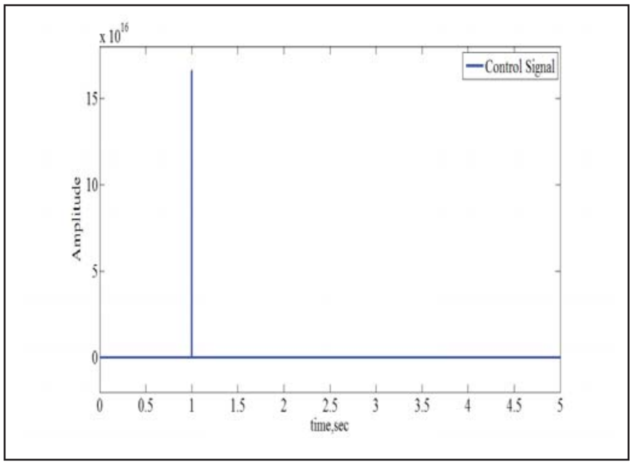


Fig.14. Control signal of PID (angle control)

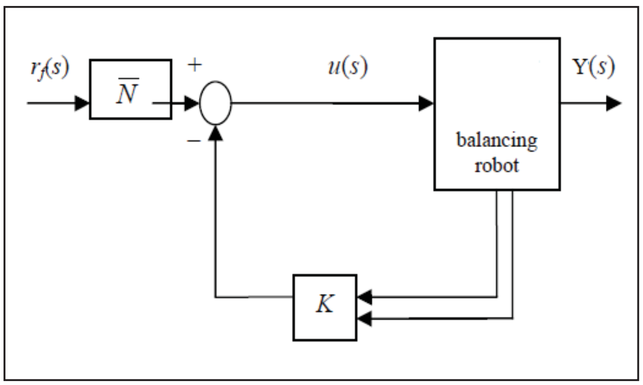


Fig. 15. LQR control system[9]

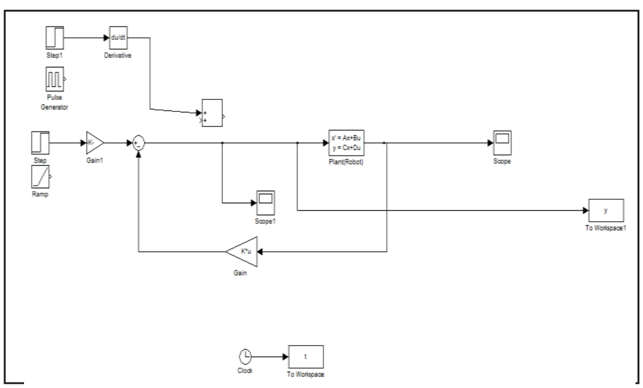


Fig. 16. Simulink model for LQR control.

The fig. 16 shows the Simulink model for LQR control simulation.

For

$$Q = \begin{bmatrix} x & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (24)$$

x and y are the weights of the states, these values determine the response of the system.

The initial start point of angle is set to 0.35 radians (20 degree). And the position is set to 0.

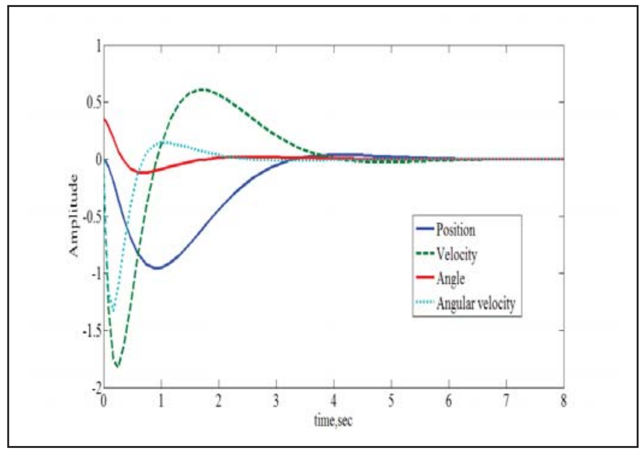


Fig.17. Output response with LQR control

.

The result are good in every aspect the position is also controlled and the robot is vertically stable.

To test the disturbance rejection capability of LQR controller, impulse disturbance is added at 5sec.

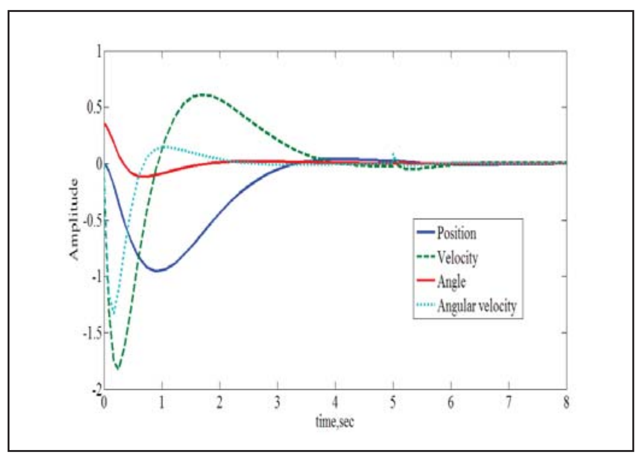


Fig.18. Output of LQR with impulse disturbance

The output shows us that the system rejects an impulse disturbance without any compromise in the position and vertical angle.

Now testing the system with impulse train disturbance at 5 sec.

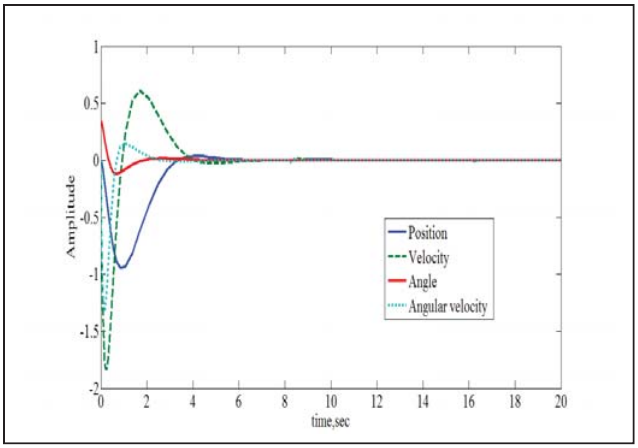


Fig.19. Output of LQR with impulse train disturbance

The system is also rejecting train of impulses (every 4 seconds) thus LQR control is capable of rejecting disturbances quite good.

Now the system is simulated for step and ramp input with impulse disturbance respectively.

Testing for these inputs will simulate if the robot is given a command for a specific position (step input) or to move forward continuously (ramp input) . The output response shows how system behaves with these inputs and if this robot system is good for moving around.

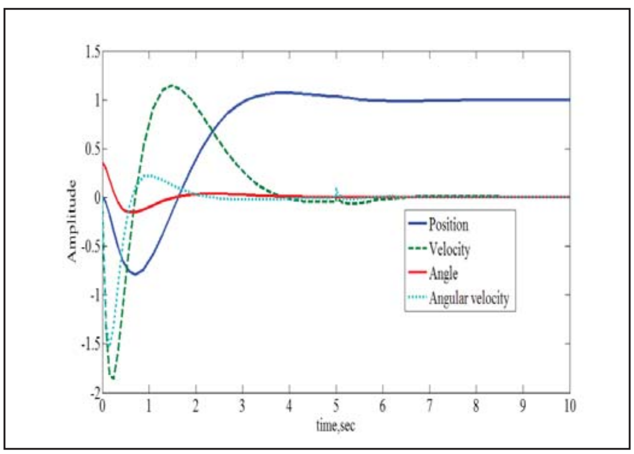


Fig.20. Output with step input and impulse disturbance

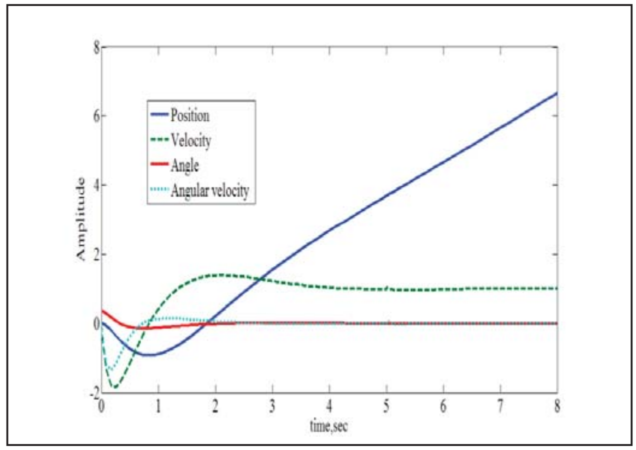


Fig.21. Output with ramp input and impulse disturbance

Now checking the control signal of the LQR control.

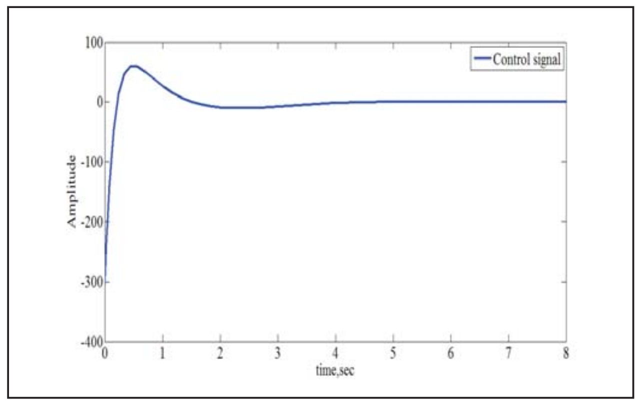


Fig.22. Control signal provided by LQR controller

VI. CONCLUSION AND FUTURE WORK

The complete system mathematical model in the form of equations was obtained including wheels model, dc motor model, chassis and platform model. The matrix obtained for linearized mathematical model of an inverted pendulum system. It was simulated using Simulink and MATLAB.

Dual-PID and LQR control were tested for position control, upright (vertical) balance angle, their control signals and their disturbance rejection capabilities.

PID controller is easier to implement but the main concern is the overshoots created by the 'I' (integral) factor in PID which makes the system prone to disturbance and is highly undesirable for this sort of non-linear system. Another major downside of PID is that it was not able to control the position and its high gains yielded a control signal too high for the system to be practically implemented.

LQR controller yielded results which were more practical alongside it was also stabilizing the system, it showed good results with the addition of disturbances and then it was tested with step and ramp input for which the result was again good only downside was the rise time which can be improved according to the systems specifications (actuators) and increasing the weights accordingly in Q matrix.

It is better to use LQR controller for self-balancing robot if the system is to include change in its position or the robot is to move around while maintaining its balance. PID can be used if position and speed are not a concern and the system is to balance itself vertically only.

The next step is to verify all these results in a real time system and practically implement the system to validate these results.

TABLE II.

Dual PID		
Vertical balance	***Position Control***	***Disturbance Rejection***
yes	no	yes
LQR		
Vertical Balance	***Position Control***	**Disturbance Rejection**
yes	**yes**	**yes**

VI. REFERENCES

[1] Liu Kun, Bai Ming, Ni Yuhua, "Two-wheel self-balanced car based on Kalman filtering and PID algorithm" 2011 IEEE 18Th International Conference on Industrial Engineering and Engineering Management (IE&EM), vol.Part 1, Sept. 2011.

[2] J. Wu and W. Zhang, "Research on control method of two-wheeled self-balancing robot," Proceedings of the 4th International Conference on Intelligent Computation Technology and Automation (ICICTA'11), pp. 476–479, Shenzhen, China, March 2011.

[3] H.Ferdinando, H.Khoswanto, S.Tjokro, "Design and evaluation of two-wheeled balancing robot chassis", Communications, Computing and Control Applications (CCCA), 2011 International Conference on , vol., no., pp.1-6. , March. 2011.

[4] Liang Sun, Jiafei Gan "Researching of Two-Wheeled Self-Balancing Robot Base on LQR Combined with PID, Intelligent Systems and Applications" (ISA), 2010 2nd International Workshop on , vol., no., pp.1-5., , May. 2010.

[5] Charles Yong and Chiew Foong Kwong, "Wireless Controlled Two Wheel Balancing Robot", International Journal Network and Mobile Technologies , Volume 2, May,2011.

[6] H. Bin, L.W. Zhen, L.H. Feng: "The Kinematics Model of a Two-Wheeled Self-Balancing Autonomous Mobile Robot and Its Simulation." ICCEA '10 Proceedings of the 2010 Second International Conference on Computer Engineering and Applications - Volume 02, 2010.

[7] N. M. Abdul Ghani, N. I. Mat Yatim, N. A. Azmi: "Comparative Assessment for Two Wheels Inverted Pendulum Mobile Robot Using Robust Control", International Conference on Control, Automation and System 2010, KINTEX, Gyeonggi-do, Korea, Oct 27-30, 2010.

[8] Mahadi Hassan, Chanchal Saha, Md. Mostafizur rahman, Md. Rabiual Islam Sarker an Subrata K. Aditya, "Balancing Of An Inverted Pendulum Using PD controller", 20th August 2011.

[9] A.N.K. Nasir, M.A. Ahmad, R.M.T. Raja Ismail, "The control of a highly non-linear two-wheels balancing robot: A comparative assessment between LQR and PID-PID control schemes". World Academy of Science, Engineering and Technology, v-46 – 44, 2010.

[10] U. Adeel, K.S.Alimgeer, O. Inam, Ayesha H., M Qureshi, M. Ashraf, "Autonomous Dual Wheel Self Balancing Robot Based on Microcontroller", Journal of Basic and Applied Scientific Research , 2013.

[11] Hidetoshi Ikeda, ZhiDong Wang, Takayuki Takahashi, Eiji Nakano, "Stable Step Climbing and Descending for Tandem Wheelchairs Connected by a Passive Link", IEEE International conference on complex medical engineering, 2007.