Implementation of a Two Wheel Self-Balanced Robot using MATLAB Simscape Multibody

Shubham Mohapatra
Electrical Engineering
Delhi Technological University
New Delhi, India
shubham.mohapatra1996@gmail.com

Rachit Srivastava
Electrical Engineering
Delhi Technological University
New Delhi, India
rachitsrivastava@outlook.com

Rupesh Khera
Electrical Engineering
Delhi Technological University
New Delhi, India
rupeshkhera9@gmail.com

***Abstract*— This paper introduces a method to design hardware systems of a 2-wheeled self-balancing robot using software tools. The implementation phases include design simulation, hardware design, signal processing and control algorithms. This paper studies a CAD based methodology in designing the robot with set physical properties and accordingly the calculation of the torque, speed, direction of motion requirements to balance the robot is done. The robot is subjected to an environment similar to a real environment with gravitational, friction and normal forces acting on it. Therefore, we can simulate the torque and speed required to adjust the robot based on an external force acting on it. Through this simulation we can optimize the hardware design of the robot knowing its mechanical features.**

***Keywords*— CAD**

I. Introduction

The two-wheel self-balanced robot is based on a classical control system problem known as the Inverted Pendulum. This system uses an accelerometer plus gyro sensor to measure the inclination of the robot. The complementary filter combines the result of the accelerometer and gyroscope to give the tilt angle of the robot.

The tilt again is maintained by control algorithms like Proportional-Integral-Derivative, Linear Quadratic Regulator, Fuzzy logic and Pole Placement. This system uses a PID controller due to its reliability and ease of implementation.

The CAD model of the robot is designed using SolidWorks. The dimensions, material and structure is determined as per the required payload capacity for transportation. The CAD model is imported to MATLAB Simscape Multibody and simulated using a motion study test bench. The objective of the simulation tests is to determine the angular velocity and torque requirements of the robot and to determine the parameters of Kp, Ki and Kd for the PID closed loop.

The hardware design and software implementation are guided by the results of the simulation. The results of the simulation and physical implementation are shown in section V.

II. System Design

A. *Mechanical specifications*

The payload capacity of the robot is 0.4 kg with the maximum dimension of 10x10 cm^2. The dimensions of the robot were estimated based on this desired load out. The drive mechanism of a motor and wheel is used for movement of chassis.

The chassis material is of softwood due to its yield strength and high availability. An approximate design of the robot was made using SolidWorks including the density and weight of materials being used.

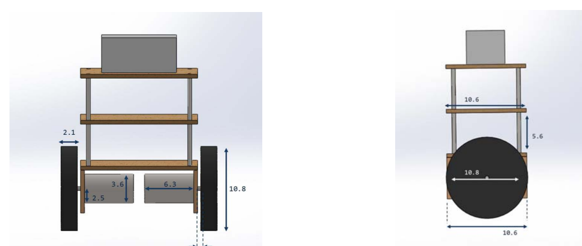

Figure 1: SolidWorks design of the robot

B. *Simscape Multibody Model*

The SolidWorks model is imported into SimScape Multibody. The connection of each mechanical part is done through a type of joint defining the motion between them. The fixed and rigid 3D transformation between two frames is set using the rotation transforms.

The figure 2 shows how each plate is connected to the bolts and the revolute joint is used to connect the shaft with the wheels. Similarly, a prismatic joint is used for the connection between the track and the wheels.

The revolute joint is input with a S-PS converter to mimic the action of a motor. And the prismatic joints are connected to a PS-S converter to measure the linear displacement and torque.

III. Control System

The robot control system is designed on MATLAB Simulink where the robot's multibody block is integrated

978-1-5386-7989-0/19/$31.00 ©2019 IEEE

with the PID block for closed loop control and an input signal is fed to the robot.

The torque requirement is estimated through an open loop system through a sinusoidal input signal through which the counter-torque required to balance the robot to a stable equilibrium is determined. Similarly, the angular velocity is determined for a closed loop PID system with an impulse input signal.

A. Determining the Torque

To estimate the torque required to recover the back to its stable equilibrium position (upright) a test motion signal of an impulse force was given to the revolute motor shaft joints of the robot. The required torque is then calculated for the open loop system and given as shown in Figure

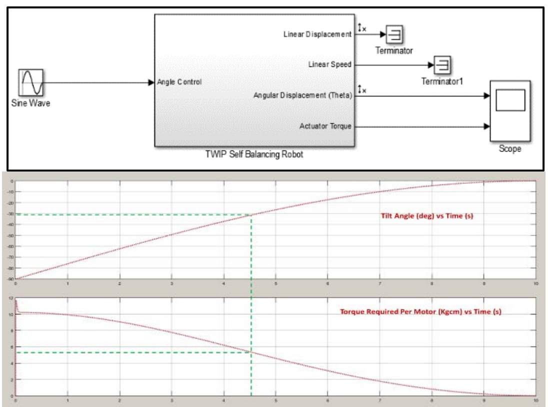

Figure 2: Open Loop Control System for torque

From the graph it can be seen that the torque required per actuator to bring the robot back to 0 degrees from a tilt angle of about 30 degrees is approximately 5.4 kg.cm - per motor.

B. Determering the Angular Velocity

Next, to estimate the required maximum rpm (for a wheel diameter of 10.8 cm), and to check the viability of close loop control using angle measurement a closed loop system

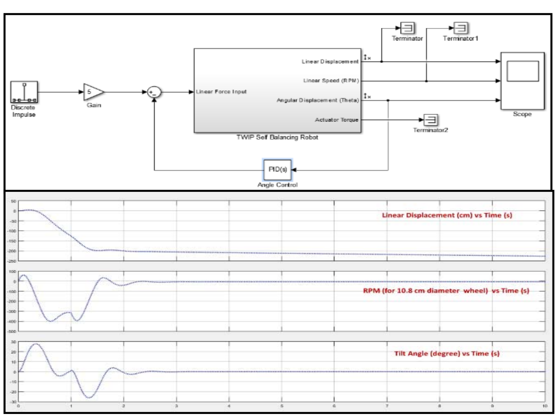

Figure 3: PID control system for angular velocity

was controlled using a Simulink PID block as shown. The plant was tuned using the PID block and the impulse response of the system was analysed to determine the required max rpm. As can be seen from the figure 4, the robot undergoes linear displacement and settles to a new steady state value of linear displacement.

When max deflection of tilt angle is about 30 degrees, the max angular velocity reached by the motors is around 400 RPM.

IV. PID TUNING

The optimal response is designed by setting the parameters of Kp, Ki and Kd. These are the theoretical values of the parameters later reviewed for tuning the actual PID parameters of the robot. Optimal response for balancing of the robot was seen at Kp=0.563. The best anticipatory control was observed at Kd= 0.02583 and the minimum steady state error was seen for Ki = 1.552.

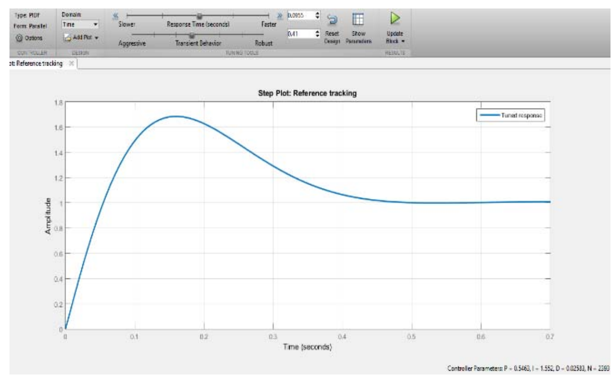

Figure 4: PID tuning of the robot in MATLAB

V. RESULT

The simulation tests and robot modelling provided with deep insights into the system behaviour without bothering about solving complex differential equations on our own. The required rated torque for each motor was calculated as 6-7 kg.cm and the rated angular velocity as 500 rpm.

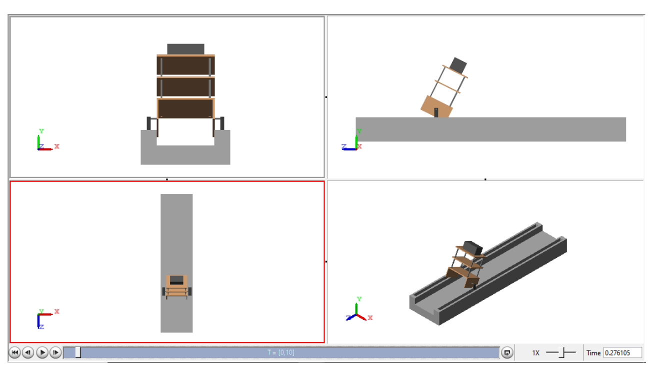

Figure 5: Performance test of the robot in a test bench

The construction of robot was done as per the specifications of the CAD model. As per the rpm, torque and power calculations obtained from the simulation results the motor and battery were selected. PID controller of the robot was tuned as per the simulation results using a mobile interface shown in figure 6.

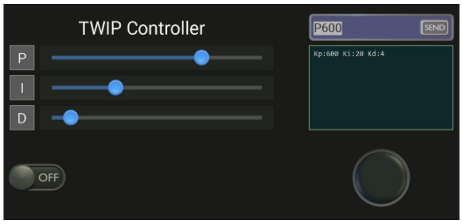

Figure 6: Mobile application to tune the robot

The simulation phase using multibody and solidworks can be followed for any such robotic application to determine the hardware ratings and design the control system model of the robot. The performance of the final ready was quite stable and adaptive to external forces. This methodology helped build a more theoretical approach and minimized chances of any estimation errors.

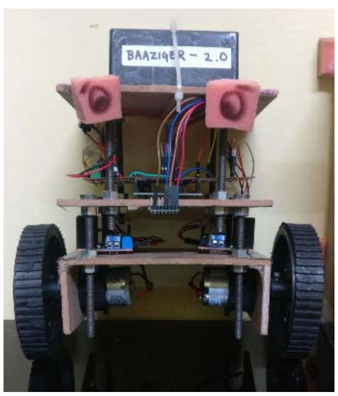

Figure 7: Final Robot

ACKNOWLEDGMENT

This research is supported by the guidance of Dr. Pragati Kumar, Delhi Technological University and rest of the faculty of Electrical Engineering Department at Delhi Technological University.

REFERENCES

[1] Fengxin Sun, Zhen Yu and Haijiao Yang, "A Design for Two-Wheeled Self-Balancing Robot Based on Kalman Filter and LQR," in International Conference on Mechatronics and Control, pp. 612-615,2014.

[2] Liang Sun and Jiafei Gan, "Researching of Two-Wheeled Self-Balancing Robot Based On LQR Combined with PID," in International Workshop on Intelligent Systems and Applications,2010

[3] M. Baloh and M. Parent, "Modeling and model verification of an intelligent self-balancing two-wheeled vehicle for an autonomous urban transportation system," in The conference on computational intelligence, robotics, and autonomous systems, pp. 1-7,2003.

[4] J. Juan Rincon Pasaye, J. Alberto Bonales Valencia, and F. Jimenez Perez, "Tilt measurement based on an accelerometer, a gyro and a kalman filter to control a self-balancing vehicle," IEEE International Autumn Meeting, pp. 1–5,2013.

[5] A. M. Mohtasib and M. H. Shawar, "Self-balancing two-wheel electric vehicle (steve)," Industrial Electronics Society, IECON 2013-39th Annual Conference of the IEEE, pp. 4198-4203,2005.

[6] A. Salerno and J. Angeles, "Nonlinear controllability of a quasiholonomic mobile robot," in Proc. IEEE ICRA, pp. 3379-3384,2005.

[7] B. Friedland, "Control system design: an introduction to state-space methods", Dover Publications, Inc., pp.30-32,2005.