Development of Robotic Arm Manipulator mounted on Self Balancing Two Wheeled Mobile Robot

Pranay Junare
College of Engineering Pune
Pune
junarepg18.extc@coep.ac.in

Shaunak Mahajan
College of Engineering Pune
Pune

PrithvishTaukari
College of Engineering Pune
Pune

Anirudh Nallawar
College of Engineering Pune
Pune

Dr. S. S. Ohol
College of Engineering Pune
Pune

Abstract— **Self-balancing robots are becoming increasingly popular now a days. They have better agility and are compact in size when compared to four-wheeled mobile robots. On the other hand, robotic arms are widely used in manufacturing units and industrial automation processes. But most of these robotic arms lack the navigation capabilities. Thus, to overcome these issues of mobile arm manipulation and performing swift locomotion in narrow dense areas, we have built a robotic system. It has two wheeled self-balancing mobile base and two 5DOF arms driven by servo actuators capable of mimicking the humanoid robot movement. For accurate state estimation of the robot, angles computed by implementation of MEMS sensor fusion algorithms is used. The Cascaded PID Controller is designed to realize the movement of the robot in forward and backward direction. Thus, the complete methodology used to maintain the robot upright throughout the unprecedented disturbance along with mechanical system design and mathematical modeling is mentioned in this paper. Later the simulation of the robotic system is done using Gazebo and ROS and finally verification of the simulation results is done by actual hardware implementation.**

1. Introduction

In today's world the application of two wheeled robots is difficult to limit to specific applications, since they can be used in numerous applications such as Segway, humanoid wheeled robots, service robots and surveillance in hazardous places. In the defense sectors, these robots can be used to train the soldiers as they can be made to act as insurgents, hostages or civilians. They can be used for visual surveillance in sensitive high-density areas as these robots are capable of moving on the streets just like humans. The design of the Self-balancing robot is based on the principle of the inverted pendulum which maintains the balance about zero moment point. Main challenge in these robots is to maintain the position of the robot upright. On the other hand, existing robotic arms are widely used in industries, warehouses, defense, manufacturing and hence combining it with mobile platforms which are compact and are capable of navigating in dense environments makes it one of the important areas of research. This paper presents the development of the Robotic arm manipulator mounted on the two-wheeled self-balancing mobile robot. The main advantage of using a two-wheeled mobile robot over the legged robot or four-wheeled robot is better mobility as well as the ability to rotate in small space. These robots are capable of moving faster than the legged robots. Two-wheeled self-balancing robots are intended to balance themselves by controlling the rotation of the wheel governed by the immediate response from the encoded motors. Also, this system computes the tilt angle of the robot and provides an optimal controlled feedback to motors using the efficient control algorithm.

The remaining sections of the paper are organized as follows: In Section 2 literature survey is done. Section 3 includes detailed Mechanical Specification and Mathematical modelling. Section 4 describes the methodology followed by us to control the built Robot system design by simulating it in the Gazebo environment. Section 5 describes the experimental results observed on the actual hardware implementation. Lastly section 6 includes the conclusions derived from the research work.

2. Related Work

Over the years, researchers have widely studied modelling and dynamic control of Two wheeled self-balancing robot. Robots such as JOE [1]- a mobile inverted pendulum robot developed in 2002, nBot[2] - which was featured as NASA's Cool Robot of the Week in 2003 &iBot[3] - a wheelchair capable of balancing on two co-axial wheels developed by Dean Kamen are some of the pioneering work. Multiple control strategies were proposed such as Pole Placement [1], Linear Quadratic Regulator (LQR)[4], Proportional Integral Derivative(PID)[5][6], Fuzzy control [7]. Since none of the single controller that performs efficiently on all the Self balancing robots so multiple control algorithms have been and will be proposed over the years. There have also been several attempts at developing two wheeled robots which had robotic arms mounted on them. Some of them include NASA's Robonaut and MIT's Cardea robot, both based on SEGWAY robotic mobile platform. One of the most recent and a notable mention is Handle Robot by Boston Dynamics, which is a two wheeled robot having robotic manipulator arm mounted on it.

3. System Design

A. Mechanical Structure

The mechanical model mainly consists of two robotic arm manipulators controlled by servos, namely MG996 for rotation of main joints and micro servo for 2 finger gripper. The arm is designed in such a way that it can mimic human hand movements and is mounted onto the torso which in turn is mounted on the main body of a two wheeled balancing robot. The basic test-based model was designed on CAD modelling software and was designed in accordance with the 3D prototyping capabilities. Hence, the material chosen was ABS which is a commonly used material for 3D prototyping. The wheels used were standard wheels available in the market and a 12V high torque encoded motor of 400 rpm was used for precise control of wheels to balance the robot. The torso and the lower body are acrylic laser cut and are joined together using bond glue and slots. The robotic arm mainly consists of 4 links to imitate human hand motion. The first link is attached to the torso which mimics the shoulder’s rotational movement in the vertical circle. The next link mimics the lateral motion of the robotic arm. Link 3 mimics the elbow movement of the arm and the last link mimics the wrist rotation. The last link is then connected to a servo actuated 2 fingered grippers. This gripper consists of a servo horn which is connected to two links at the end which has a restricted linear motion. The entire model has a maximum span of 1300 mm. It has a height of 800mm and has a width of 200mm.

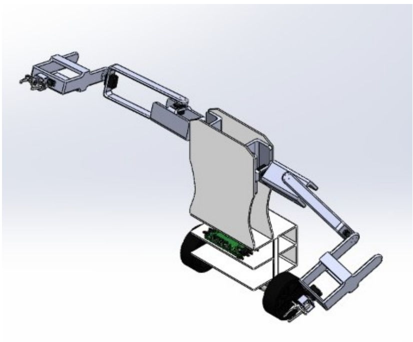

Fig. 1. CAD Design

B. Mathematical Model

Assumptions:

(i) The system is assumed to be rigid.
(ii) There is no net slip in wheels and ground.
(iii) Radius of both wheels are constant and don't vary.
(iv) Mass distribution in the body is uniform.

Table 1

Symbol	**Name of the parameter**	**Values**	**Unit**
r	Wheel radius	0.15	m
M_R	Mass of rotating system	0.52	kg
I_R	Moment of Inertia	0.0117	kgm^2
F_R	Friction Coefficient	0.2	$Nrad^{-1}s^{-1}$
M_B	Mass of main body	3.00	kg
I_B	MOI of main body	0.82	kgm^2
F_B	Friction coefficient of main body	0.002	$Nrad^{-1}s^{-1}$
l	Length measured from the wheel axis to COM	0.112	m
g	Gravity	9.8	ms^{-2}

The motion of the two-wheel-balancing robot can be obtained using Lagrangian dynamics.

The kinetic energy of the rotation system due to angular displacement is:

$$K.E. = \frac{1}{2} m_R v^2 \# \ (1)$$

Here, x_i and y_i coordinates indicate the position of the center of gravity of the main body:

$$x_i = x + l\sin(\theta_b) \#(2)$$
$$y_i = -l sin(\theta_b) \#(3)$$

Here θ_b is the angle subtended by the vertical axis and the longitudinal axis of the two-wheel-robot.

Differentiating x_i and y_i with respect to time we have:

$$v_x = v + l cos(\theta_b); \#(4)$$
$$v_y = -l sin(\theta_b); \#(5)$$

Calculating the scalar quantity of total velocity;

$$|v^2| = v^2 + +2lv\omega\cos(\theta_b) + l^2\omega^2 \#(6)$$

Now the kinetic energy of the main body due to linear motion is:

$$K.E._B = \frac{1}{2} m_B v^2 + m_B l\omega v cos(\theta_b) + \frac{1}{2} m_b l^2 {\theta_b}^2 \#(7)$$

Hence, by adding all the K.E. we get the total energy as:

$$E_T = \frac{1}{2}(m_B + m_R)v^2 + m_B lv\omega\cos(\theta_b) + \frac{1}{2}(J_B + m_B l^2)\omega^2 \#(8)$$

Using Lagrangian equations [5], we find angular acceleration and linear acceleration for the linearized model as:

$$a = -\frac{m_B l}{m_B + m_R}\omega + \frac{1}{m_B + m_R}(T_w - f_R v) \#(9)$$

$$\alpha = \frac{m_B l}{J_B + m_B l^2} a + \frac{m_B l g}{(J_B + m_B l^2)}\theta_b \ \#(10)$$

4. Simulation in Gazebo

For verification, efficient testing, algorithm implementation and feasibility of the proposed robot model we first need to simulate it in a virtual environment. Thus, for this purpose out of numerous Robotics Simulator platforms such as Webots, Gazebo, V-REP, CARLA, AirSim, USARSim we choose Gazebo software due to its enormous features, flexibility and convenient integration with ROS. Gazebo is an Open-source 3D simulator which has an inbuilt physics engine, supports sensor manipulation and has a rich visualization. On the other hand, Robot Operating System (ROS) is a middleware that has numerous collections of libraries, drivers, and tools for building robotic systems.[9]

A. Simulation world

Gazebo was used to simulate the entire real-world environment. It allows us to create a 3D Virtual world with characteristics such as length, mass, joints, links, friction coefficients, etc[10,11]. Other parameters such as simulation step size, camera pose, lighting was configured according to our need. Finally, in this experiment for simulating the environment, a custom world file was written by specifying different parameters. (Fig. 2)

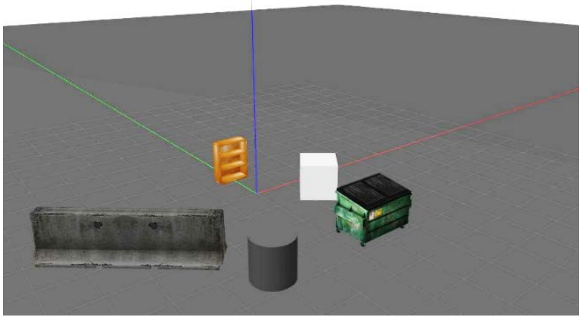

Fig. 2. The simulation world

Table 2: Parameters in Gazebo

Name of the parameter	**Values**
Gravity(m/s^2)	9.8
Density (kg/m^3)	1.1
Mass (kg)	3.6
Length of frame(mm)	1300
Width of frame(mm)	200
Height of frame(mm)	800

B. Modelling the robot

For building the robot model, SolidWorks was used and the built model was converted to URDF (Universal Robotic Description Format) using SolidWorks to URDF Exporter. URDF file only specifies the dynamic and kinematic [12] properties and hence further it was modified to add additional properties and simulation specific information to make it compatible with Gazebo. Some of the important parameters are listed in TABLE II. Our robot model has two robotic arm manipulators, two wheels, upper torso and a lower body chassis frame. Finally importing of robot model in the previously built Gazebo virtual environment was done. (Fig. 3)

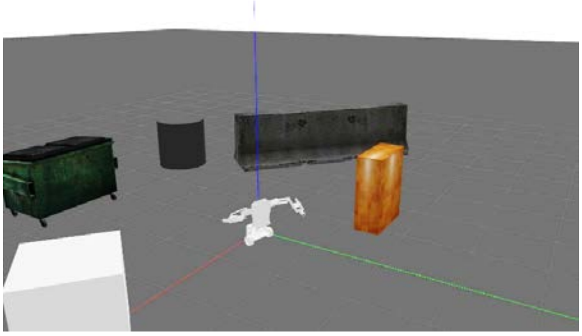

Fig. 3. The simulated robot model

C. State Estimation

To apply the control strategy on a Self-Balancing robot it is very important to estimate the dynamic current state of the robot. The state of the robot at any time t is represented by y(t) and is represented by the following eq 1.

$$y(t) = [\phi \;\; \theta \;\; x \;\; y] \,\#(11)$$

Where:
ϕ: pitch angle (Tilt angle)
θ: yaw angle
x: Position x of the robot
y: Position y of the robot

For measuring the pitch and yaw angles of the robot MEMS motion sensor is used. Further the raw values of accelerometer, gyroscope and magnetometer are fused using Sensor Fusion algorithms such as Complementary and Madgwick filters.

D. Cascade PID Control

Proportional-Integral-Derivative (PID) Controller is a feedback control mechanism used for controlling the process variable. It is used when the control systems are closed loops. PID Controller has three parameters: Proportional Gain, Integral Gain and Derivative Gain. Proportional component determines the system response to the current error value, Integral component determines the accumulation of overtime error value whereas Derivative Component is used to predict future error in the system. PID controller is represented by the following equation:

$$u(t) = K_p * e(t) + K_i \int_0^t e(t)\,dt + K_d * \#(12)$$

Where:
u: controller output.
E: Error = Setpoint – Process Variable
K_p: proportional constant gain.
K_i: integral constant gain.
Ki: derivative constant gain.

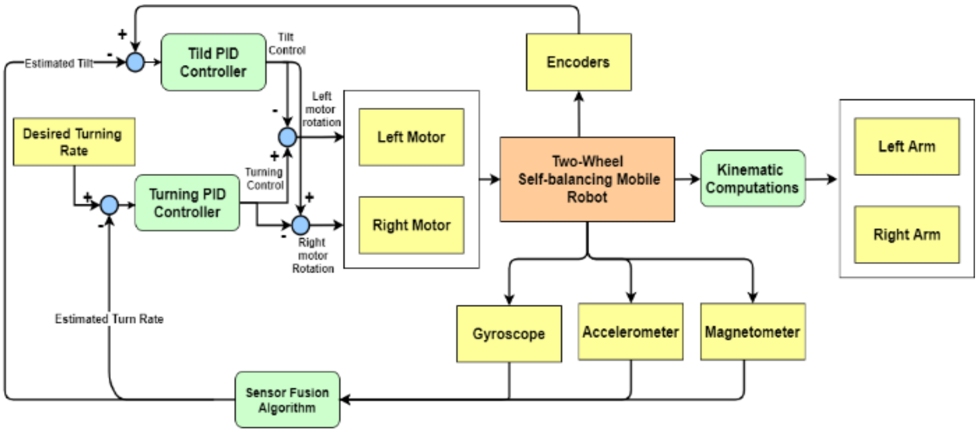

Fig. 4. Control System block Diagram

We implemented the cascaded PID controller in order to control the robot state. Cascaded PID controller is a control algorithm where output of the first controller provides the setpoint for the second controller. Such systems are able to give improved response to non-linear gains and disturbances. In our controller the inner PID loop is used to precisely control the encoded motors whereas the outer PID loop maintains the robot upright by controlling the tilt angle. The complete block diagram of the implemented Control System is as shown in Fig. 4.

5. Experimental Results

The proposed robot after mathematical analysis and simulation was implemented in hardware prototype. The developed robot having two robotic arm manipulators and balancing on two-wheels is shown in the Fig. 5. Thus, the proposed theory was experimentally verified.

6. Conclusion

In this paper, mathematical modelling and analysis of Self balancing mobile robot having Robotic arm mounted on it is done. Further the proposed model was simulated in Gazebo to test the feasibility and finally it was implemented in hardware. The experimental results show that for the robot having mechanical specifications as mentioned in the paper, Implemented Cascaded PID Controller is able to achieve stability.

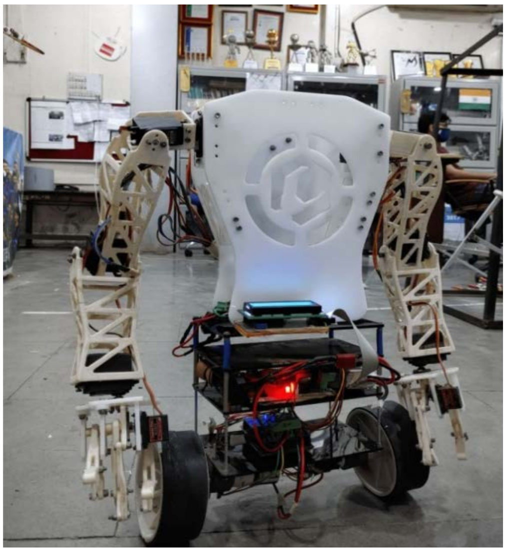

Fig. 5. Hardware Implementation

7. Acknowledgment

The authors would like to thank all the members of Robotics & Automation Laboratory COEP, for their valuable inputs. This research is supported by Robotics and Automation Laboratory of College of Engineering Pune.

8. References

[1] Grasser, A. D’arrrigo, S. Colombi, And A. C. Rufer, “Joe: A Mobile, Inverted Pendulum,” Ieee Transactions on Industrial Electronics. 49, No. 1, Pp. 107–14, 2002
[2] P. Anderson. (2003, Aug.) nBot balancing robot. Online. [Online]. Available: http://www.geology.smu.edu/ dpa-www/robo/nbot
[3] L. Kamen, R. R. Ambrogi, R. K. Heinzman, C. C. Langenfeld, M. A. Nisbet, S. B. Smith (III), and T. A. Brindley, “Medical improvements to a personal vehicle”, U.S. Patent 6,405,816, June 18, 2002
[4] S. Ahmad and M. O. Tokhi, “Linear Quadratic Regulator (LQR) Approach for Lifting and Stabilizing of Two-Wheeled Wheelchair,” 2011 4th Int. Conf. Mechatronics, no. May, pp. 1–6, May 2011.
[5] T.-J. Ren, T.-C. Chen, and C.-J. Chen, “Motion control for a two-wheeled vehicle using a self-tuning PID controller,” Control Eng. Pract., vol. 16, no. 3, pp. 365– 375, Mar. 2008.
[6] H. -S. Juang and K. -Y. Lum, "Design and control of a two-wheel self-balancing robot using the arduino microcontroller board," 2013 10th IEEE International Conference on Control and Automation (ICCA),2013, pp. 634-639, doi: 10.1109/ICCA.2013.6565146.
[7] Huang, W. Wang, and C. Chiu, “Design and Implementation of Fuzzy Control on a Two-Wheel Inverted Pendulum,” IEEE Trans. Ind. Electron., vol. 58, no. 7, pp. 2988–3001, 2011.
[8] Fabián R. Jiménez, Ilber A. Ruge, and Andrés F. Jiménez, “Modeling and Control of a Two Wheeled Auto-Balancing Robot: a didactic platform for control engineering education”, 18th LACCEI International Multi-Conference for Engineering, Education, and Technology:
[9] Player/Stage project. (2016) The Player Project. [Online]. Available: http://playerstage.sourceforge.net/
[10] http://wiki.ros.org/gazebo_ros_pkgs
[11] http://gazebosim.org/tutorials?tut=ros_plugins&cat=connect_ros
[12] Takaya, Kenta, et al. "Simulation environment for mobile robots testing using ROS and Gazebo." 2016 20th International Conference on System Theory, Control and Computing (ICSTCC). IEEE, 2016.