*i*Journals: International Journal of Software & Hardware Research in Engineering (IJSHRE)
ISSN-2347-4890
Volume 11 Issue 10 October 2023

Effect of Wheel Materials on the Balancing of a Self- Balancing Robot

Diti Chhaproo
Prabhavati Padamshi Soni International Junior College
Mumbai,India
ditichhaproo@gmail.com

Reetu Jain
Chief-Mentor and Founder
On My Own Technology Pvt. Ltd
Mumbai, India
reetu.jain@onmyowntechnology.com

***Abstract*— A self-balancing robot model was first built by Professor Kazuo Yamafuji in the 1980's when he suggested that it could follow the principal of an inverted pendulum. It uses a feedback control system and PID algorithms to adjust motor speeds and directions which helps it stay upright. The wheels of a Self-Balancing Robot are rudimentary factors in this process as the correction has to be instantaneous to prevent the robot from falling down. The purpose of this paper is to test various wheel materials and make observations using concepts of Physics - including Centre- of- Mass, Torque, Friction, and advanced mechanics- including the vector method, the Lagrangian dynamics, and the screw theory to conclude the ideal material for the functionality of this bot.**

***Index Terms*— Self- Balancing, Wheel Material, Mechanics, Friction, Lagrangian**

I. INTRODUCTION

The topic of research is varying the materials of the wheels of a self-balancing robot to identify and expatiate the prominent differences in the speed of error correction and adjustability to changing resistance conditions. A self-balancing robot is a highly unstable, high-order, multi-variable, strong-coupled, and non-linear system. To maintain balance, the bot would require a control input system to constantly alter the torque in the wheels. These can be used to stabilize the system by keeping the robot's tilt angle close to vertical and its center of mass over the pivot. Wheels play a fundamental role in this system as the material of the wheels will change their weight and coefficient of friction. This will subsequently affect the inertia and adaptability of its balancing. This robot will have a narrow base, but its mass will be concentrated around the chassis where the motors and components i.e., the Arduino Nano Microcontroller, MPU6050 gyro sensor, and TB6612FNG motor driver are soldered. This robot also has a low center of mass since the battery is on the base chassis with the heavy motors- these aid in the stability.

The four materials chosen for this paper are Rubber, Hard plastic, PLA(Polylactic acid), and TPU(Thermoplastic Polyurethane). Ideally, the tire traction would have maximum contact with the ground and minimal resistance through the drive system. The entire working principle of the Self-Balancing Robot comes from the inverted pendulum. This system works with delicate control, which means the control of output should be precise on the order of the precision of the input of the system- a theoretically impossible dilemma in our surroundings with factors of resistance and gravity. Ideally, the inverted pendulum concept works perfectly in the functioning of this robot because its delicate control system syncs with the PID control. The Proportional-Integral-Derivative calculates Kp, Ki, and Kd values which are a linear combination of the value proportional to the difference between its correct value and desired value, an integral of the amalgamation of its past errors to remove any steady-state error, and a derivative to anticipate any future errors. The self-balancing robot can be used to maneuver in compact, remote scenarios on rough terrains with changing friction coefficients. The wheel materials could play a crucial role in such situations, especially if it's in a dangerous territory, and this paper aims to help make its speed of

adjustment more seamless and time-efficient by experimenting with this factor.

II. LITERATURE REVIEW

Bang-gui Zheng et al[1] scrutinized that PID and LQR were used for controlling the robot and to provide a viable feedback alternative for the robot by mitigating the accumulated residual errors and autonomous oscillations. Using concepts of Newtonian mechanics like sliding and rolling friction and digital simulators like MATLAB, they tested out the variations in angular displacement for a PID, an LQR and a PID+LQR system and observed a lesser variation per unit time for the latter. The system successfully reduced the frequency and amplitude of the self-oscillations, giving a stable smooth operation, making the duo more efficient at feedback control- as delineated in their paper.

Using the concept of a learning bike and the idea that a more skilled rider would need lesser support on a self-balancing system when multiple disturbances are applied, Mamta M. Barapatre and V N. Sahare[2] assessed various balancing techniques. Lagrangian mechanics was used by the authors to depict the stability of the bike by using electric motors to control its handles, inputting values from a gyro sensor, and controlling this stability by commanding the steering torque or the steering angle to use in equations of motion. They found that accurate calculations of the center of gravity and the tilt angle along with minimizing collisions by testing the model in different unstable and stable dynamics would reduce the cost and increase the safety of a learning bike, as highlighted in their paper.

In their seminal work, Junfeng Wu and Wanying Zhang[3] designed fuzzy logic and feedback controller simulations at a set disturbance force. Using Newtonian mechanics and frictional coefficients in an effort to calculate a more versatile and dynamic performance metric for a self-balancing robot, and MATLAB to illustrate 3D surface plots and correlation diagrams for the same. The simulation results, as is ultimately asserted in their research, have demonstrated that the fuzzy logic control method used in the robot system yields superior dynamic performance compared to the pole placement technique.

Linyuan Guo et al[4] proposed using a reinforcement Q-learning to solve the LQR for the bot. The issue identified that solving the linear Riccati equation would require optimal system parameter information. A system that would control the robot without precise readings would be desirable. Utilizing the mathematics of Newtonian mechanics to design an optimal control system and using an online resource to input decoupling and the pre-feedback law, their research unveiled a successful algorithm overcoming obstacles like complexity and instability and uncovered a more apprehensible self-balancing mechanism.

Nguyen Gia Minh Thao et al [5] aimed to design a self-balancing robot that would focus on hardware specifications, a Kalman filter algorithm, and a PID backstepping control system. They aimed to help the bot maintain stability as it caught the reference signal which would implement the three PID loops on it to aid in direction control, stability, and position management. Their results showed a triumphant system and more efficient control of the self-balancing bot.

This paper used Raspberry Pi instead of the traditional Arduino IDE to code the PID control system. The main objective that Yadav Ashwini Ramchandra et al [6] undertook was finding a cheaper alternative to build a self-balancing robot. Error detection in pre-set Kp,Ki,Kd values was detected and corrected, the robot achieved independent motion with the Raspberry Pi-coded PID and trajectory control mechanisms. They were able to create the code successfully and develop a cost-effective bot, as advocated by their analysis.

Aditi Supekar et al [7] sought to review the various software, technologies(such as PID, fuzzy logic etc.) and navigation systems that are used to build self-balancing robots as well as forage for newer applications for this autonomous machine. Constructing these robots with various amalgamations of software like LiDAR or building them with the inverted pendulum concept, they constricted the parameters and conducted tests in different conditions, and showed their results, showcasing through their study the advancement in this niche field over the past few years in terms of the control systems and navigation and how it could be useful in hotels and hospitals(especially during COVID-19).

Jianhai Han et al. [8] described the Sensor Fusion Algorithm and searched for a cost-effective, MEMS sensor, working system. Using the Kalman filter to gather the fused angle from the sensors and then geometric equations helped calculate the angle. Derivatives were used to find the control algorithm calculations and get values for Kp,Ki, and Kd. The authors experimented with these different ways to get the inclination angle and concluded that the complementary filter gave the most accurate value- thus successfully designing the sensor fusion mechanism for a Self-balancing robot.

Using distant sensors, Gothe Shahid Shams et al [9] aimed to design their robot that would balance the two chassis using Arduino IDE. They use the inverted pendulum as a reference to conduct calculations on this system- stating that the pseudo-force's torque in the accelerating robot's frame would oppose gravity's torque; with sufficient strength, it could level the robot's platform. Summarising that concrete results are often impossible to attain with this type of robot, they proposed the future utilization of a rotating disc for precision.

Sulabh Kumra and Shilpa Mehta [10] identified the problem as a Self-balancing robot isn't able to interact with dynamic conditions like rough terrains. They used a single axis and a single accelerometer and faced the issue of large fluctuations in the acceleration due to the quick adjustment of the entire machine. They didn't make use of a gyro sensor which caused problems with balancing the robot and used spirit levels instead. It has been substantiated in their paper that the bot was unstable and thus the system was rather unsuccessful.

Jingang Yi [11] delved into constructing a sensing system for the deformation of tires in various robotic automobiles. Using the Darmstadt, Surface acoustic wave(SAW), Strain and capacitance, and PVDF sensors for experimentation, he constructed kinematic schematics for the frame, considering the center of gravity to be the robot's geometric center. Choosing PVDF sensors due to their low cost, high flexibility, and strong compatibility with wheel tread materials, he successfully created the 'Smart-tire' system after monitoring various wheel-ground interactions like friction and slip.

Using neural networks and converting non-linear dynamics to a linear system, Isaac Gandarilla et al [12] established an alternative control system -an adaptive neural network or ANN - which they tested via comparison to other feedback/delicate control systems. They referred to the inverted pendulum concept and used the Lagrangian framework to quantify this dynamic. As delineated in their paper, the ANN system proved to be more robust, accurate, and less error-prone than a PID control mechanism.

This self-balancing bot, named Hominid-3, was used in an obstacle detection scenario equipped with ultrasonic sensors and a fuzzy control system. Xiaogang Ruan and Wangbo Li [13] used MATLAB to simulate the workings of this control system and placed the Hominid-3 in uncharted territory with 24 fuzzy rules. MATLAB and Newtonian mechanics(using the inclination angle and frictional coefficients) both showed in the tables and equations underscored in their study, that the obstacle avoidance control mechanism worked with the fuzzy logic control and the bot was successful in an unknown environment at dodging the obstacles.

This autonomous system was modeled by Umar Adeel et al [14] in PROTEUS, MATLAB, and VM lab to simulate the control system verification and monitor the robot as it successfully acquired its stable system. In their influential analysis of the data, Lagrangian mechanics is used to model a linear continuous state space representation of the bot.

Tomislav Tomašić et al [15] used a control system(PID), system characterization, and electronics. Performance limits in mobile vehicles is increasing, they claimed as they successfully balanced the bot from their computer and phone. An LQR system is implemented, and they propose a future scope of using a vision system to help in object detection.

III. PROPOSED METHODOLOGY

A. Electric components

The TB6612FNG motor driver pairs well with the gear motors with a supply range of 2.5V to 13.5V. It has 4 input pins to connect the two motors and control their direction (HIGH/LOW) and two

PWM pins to control the speed (in rpm) of the motor. The 16 connections to the motor driver connect it to the Arduino Nano which would give it instantaneous feedback from the PID to control the motor speed and direction. The driver is easy, affordable, and powerful enough to control the two high-speed motors(100-150rpm) and draw enough current for the motors to use, which, in our case is 1.2A continuous current and 3.2A peak current-compatible with most DC motors.

The MPU6050 is a gyroscope accelerometer sensor. The gyroscope measures angular velocity meaning the change in yaw, pitch, and roll in the three axes. The three-axis accelerometer measures velocity. It works on a power supply range of 3V to 5V. It uses the Inter-Integrated circuit to connect multiple devices on a bus, usually sensors and microcontrollers. The two bidirectional lines of the I2C are SDA (which conducts data communication between components) and SCL (which streamlines said transmission of data). The SDA connects to the A4 pin and the SCL connects to the A5 pin so that if the robots tilt and angle moves away from the preset condition, then the PID values are calculated and give feedback to the Nano microcontroller, and the motors are moved accordingly.

Arduino Nano is a smaller microcontroller design with the same functionality as the larger Arduino Uno board and is a more convenient successor of the same. It has PWM pins to help control the motor speeds with the driver and it also has analog pins to connect with the MPU sensor. It can be coded through Arduino IDE. Its compactness helps as it can be soldered onto this self-balancing robot's minimalistic circuit board(100mm x 120 mm). Its versatility and ease of coding make it an ideal fit for this project.

Fusion 360 was a tool used to design the entire model of this robot prior to its construction. A model can help as it can cut costs if we calculate the exact dimensions required of the clamps and studs. First, the chassis was designed with four circular holes on each corner for stud placement(which would connect the two chassis) and gaps for the wires to go through to give it a cleaner look. This was then acrylic cut and four 50mm studs connected the layers. The DC motors were also designed to fit in the clamps via various features of the CAD software.

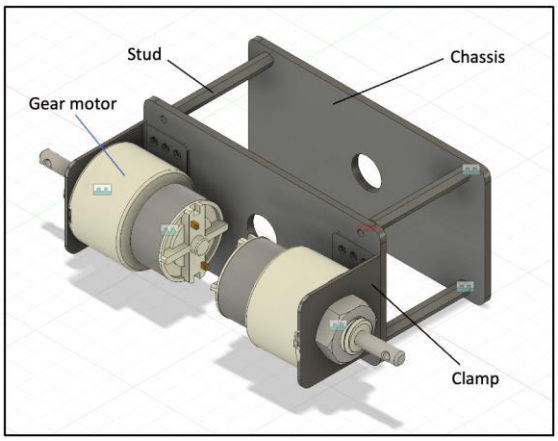

Figure 1 3D CAD model of the Self-Balancing Robot

B. Wheel material choices:

Since the magnifying glass is on the materials of the wheels in this paper, they were chosen based on their rigidity and composition. Rubber was chosen because of its traction and elasticity- it is one of the most popular wheel materials used widely in robotics. Hard plastic was chosen as it is lightweight and has low rolling resistance. Wheels of TPU- a thermoplastic elastomer- were 3D printed for this experiment; it was chosen because of its shock-absorbing and flexible properties. PLA is a biodegradable elastomer that is rigid and robust-making it a better finishing material, however, it has low tensile strength.

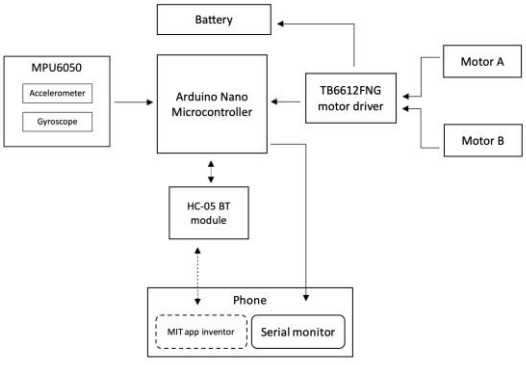

Figure 2 Block diagram of the Self Balancing Robot

The block diagram of the Self-Balancing Robot entails a basic representation of the connection of the components either via connecting wires or Bluetooth. The change in the inclination angle of the robot is sensed by the gyro sensor i.e.,

MPU6050. It sends a signal to the Arduino Nano which is connected to the phone by Bluetooth to the MIT app inventor with the PID code. The Kp,Ki, Kd values are calculated and if they don't conform to the pre-set, an alert is sent back to the microcontroller to send a signal to the motors to move according to that change. The Nano board sends a signal to the motor driver which then sends signals to the motors to move to control that change and revert the robot back to a stable and upright position.

C. Inverted pendulum(Theory behind the Self-Balancing Robot):

The inverted pendulum is the primary concept of the working principle of a self-balancing robot. When the pendulum is on a moving cart that has velocity on a linear plane, it moves with rotational acceleration and so the cart moves in that direction to ensure the pendulum bob stays upright at its setpoint inclination angle i.e., 0 degrees. If the pendulum bob moves left, for example, the moving cart will move left too and the center of gravity for the entire mass will adjust to the right so that the rotating mass doesn't topple. This concept models in the self-balancing robot as the two chassis and motors act as the pendulum bob- the rotating mass and the wheels which are also the pivots of the system are in a linear motion on the ground. In our case, Lagrangian dynamics aid in increasing the simplicity of calculations by eliminating force constraints that Newtonian mechanics may adopt. This system also works with energies, rather than forces- producing the derivation of the equations of motion without the complexities of vectors.

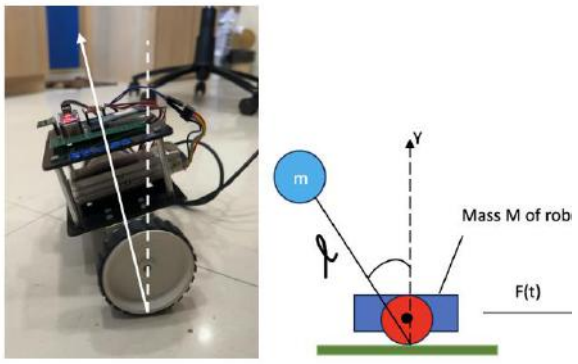

Figure 4 Mathematical expressions of Inverted Pendulum

$X_m X_m$ is the displacement of rotating mass in the X-axis.

$Y_m Y_m$ is the displacement of rotating mass in the Y-axis.

XX represents the displacement of the mass M in the X- axis.

First, we establish two equations of motion of the rotating mass using trigonometry.

$$X_m = X - l\,sin\,sin\,\theta$$
$$Y_m = l\,cos\,cos\,\theta$$

Now, differentiate both these equations.

$$\frac{d}{dt}(l) = l\,\dot{\theta}$$
$$\dot{X}_m = \dot{X} - l\,\dot{\theta}\,cos\,cos\,\theta$$

$$\dot{Y}_m = -l\,\dot{\theta}\,sin\,sin\,\theta$$

The Lagrangian equation is: $L = T - VL = T - V$

Potential Energy, V

$$V = mgh$$
$$V = mg\,Y_m$$
$$Y_m = l\,cos\,cos\,\theta$$
$$V = mgl\,cos\,cos\,\theta$$

Kinetic Energy, T

$$T = \frac{1}{2}M\dot{X}^2 + \frac{1}{2}m\left(\sqrt{\dot{X_m}^2 + \dot{Y_m}^2}\right)^2$$
$$T = \frac{1}{2}M\dot{X}^2 + \frac{1}{2}m\left(\dot{X_m}^2 + \dot{Y_m}^2\right)$$

$$T = \frac{1}{2}M\dot{X}^2 + \frac{1}{2}m\left[\left(\dot{X} - l\,\dot{\theta}\,cos\,cos\,\theta\right)^2 + \left(-l\,\dot{\theta}\,sin\,sin\,\theta\right)^2\right]$$

$$T = \frac{1}{2}M\dot{X}^2 + \frac{1}{2}m\left(\dot{X}^2 - 2\dot{X}l\,\dot{\theta}\,cos\,cos\,\theta + l^2\dot{\theta}^2cos^2\theta + l^2\dot{\theta}^2sin^2\theta\right)$$

$$T = \frac{1}{2}M\dot{X}^2 + \frac{1}{2}m\left[\dot{X}^2 - 2\dot{X}l\,\dot{\theta}\,cos\,cos\,\theta + l^2\dot{\theta}^2(cos^2\theta + sin^2\theta)\right]$$

$$T = \frac{1}{2}M\dot{X}^2 + \frac{1}{2}m\left[\dot{X}^2 - 2\dot{X}l\,\dot{\theta}\,cos\,cos\,\theta + l^2\dot{\theta}^2(cos^2\theta + sin^2\theta)\right]$$

$$T = \frac{1}{2}M\dot{X}^2 + \frac{1}{2}m\dot{X}^2 - ml\dot{\theta}\dot{X}\,cos\,cos\,\theta + \frac{1}{2}ml^2\dot{\theta}^2$$

$$T = \frac{1}{2}\dot{X}^2(M+m) - ml\dot{\theta}\dot{X}\,cos\,cos\,\theta + \frac{1}{2}ml^2\dot{\theta}^2$$

Lagrange's equations

$$\frac{d}{dt}\left(\frac{\partial L}{\partial q_i}\right) - \frac{\partial L}{\partial q_i} = Q_i$$

$$L = T - V$$

According to D'Alembert's Principle, the sum of the force differences and derivatives of momentum with respect to time become zero in any virtual displacement scenario.
Thus F(t) is put on the right-hand-side of the Euler-Lagrange equation.

According to Lagrange, take the derivative of L with respect to $\dot{X}\dot{X}$
Differentiating only the terms with XX

$$(M+m)\ddot{X} - ml\left(\ddot{\theta}\, cos\, cos\, \theta - \dot{\theta}^2 sin\theta\right) = F(t)$$

This equation includes acceleration(both linear and rotational) and mass.
It also demonstrates Newton's second law i.e., F=ma

Next, we differentiate the L equation with respect to $\theta\theta$
Once again, only differentiating the terms with $\theta\theta$

$$ml^2\ddot{\theta} - ml\ddot{X}\, cos\, cos\, \theta + ml\dot{X}\dot{\theta}\, sin\, sin\, \theta - ml\dot{\theta}\dot{X}\, sir$$

There is no external moment, so this equation equals zero.

$$ml^2\ddot{\theta} - ml\ddot{X}\, cos\, cos\, \theta - mgl\, sin\, sin\, \theta = 0$$

$$l\ddot{\theta} - \ddot{X}\, cos\, cos\, \theta - g\, sin\, sin\, \theta = 0$$

Thus, the two equations of motion for the inverted pendulum system become:

$$(M+m)\ddot{X} - ml\left(\ddot{\theta}\, cos\, cos\, \theta - \dot{\theta}^2 sin\theta\right) = F(t)$$
$$l\ddot{\theta} - \ddot{X}\, cos\, cos\, \theta - g\, sin\, sin\, \theta = 0$$

These are equations that could be further used to model the acceleration variation with inclination angle in the self-balancing robot. However, in this paper, the analysis will be based on the MPU6050 readings.

.

IV. RESULT AND DISCUSSION

A. Statistical Study

To conduct an in-depth statistical analysis of the MPU6050 readings prior to producing FFT graphs, a bell curve was made for the four materials. To model the normal distribution of the acceleration values, MS Excel functions were used with a custom-made dataset from the Serial monitor values. For comparison purposes, they were transposed onto one graph.

The standard deviation, $\sigma\sigma$, was found by finding the deviation of the average accelerometer values and average acceleration for each material- which came to the set point **11.1** ms^{-2}.

To find average accelerometer values $\sqrt{a_x{}^2 + a_y{}^2 + a_z{}^2}\sqrt{a_x{}^2 + a_y{}^2 + a_z{}^2}$ is the expression used.

Since the MPU6050 gives accelerometer readings in quanta of 'g', they have to be converted to ms^{-2} because the values are very large and incoherent.
This conversion was done with the equation:

$$\frac{a_{average}}{16384} * 9.81$$

Where, 9.81 is the acceleration of free-fall, g, and 16384 relates to the full scale of this gyroscope and to convert to raw g values, it needs to be divided out.

To make this graph, certain anomalous values had to be eliminated to minimize skewness and maintain the kurtosis of all four plots at approximately 3(for example, values at the beginning of motion when power delays for the motors were zero had to be removed.)

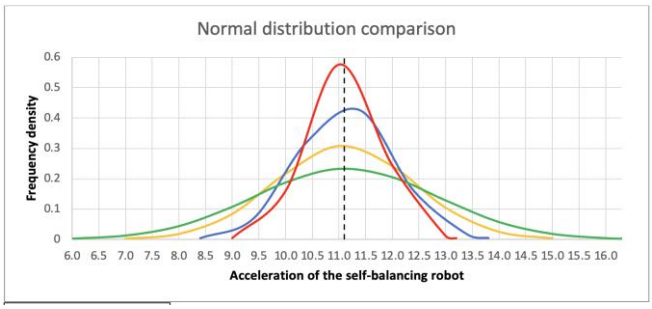

Figure 5 Normal Distribution curves of the four materials

Legend	
Colour	Material
	TPU
	Hard Plastic
	Rubber
	PLA

Figure 6 Legend for the Normal distribution and Cumulative frequency graphs

From Figure 5, TPU doesn't deviate much from its average value of 11.1. It has the lowest standard deviation value of 0.684 which illustrates that this flexible material has higher precision and lower risk of outliers. It means that the bot is more stable and lies within a smaller acceleration range.

Hard plastic with a standard deviation of 0.894 has a lower peak and wider range meaning there is a greater chance of uncertainty and outliers, making it less predictable. Another interesting observation from this plot is that it is not a symmetric bell curve- it is a skewed distribution. Although the average acceleration is 11.1, the 0.5 percentile is at 11.4. This skewed graph could be a result of weak shock absorption and wear and tear. This could reduce the traction and the bot may have to accelerate more to maintain the inclination angle setpoint(from the inverted pendulum concept).

Rubber had a standard deviation of 1.30 indicating that acceleration spreads from the mean more since it has higher elasticity than TPU and thus it absorbs more energy, and its deformation follows a broader distribution. TPU is a synthetic, so it has a more controlled distribution leading to its narrow curve.

PLA has the broadest bell curve. It has a standard deviation of 1.71 and this variance in the acceleration values could be a result of the unevenness of the tire material since it was 3D penned upon a round template. It could also be due to PLA's amorphous nature which could cause deformation to be disordered and random.

Another graph to draw interesting conclusions from is the cumulative frequency of these materials. It follows the same legend as the normal distribution graph.

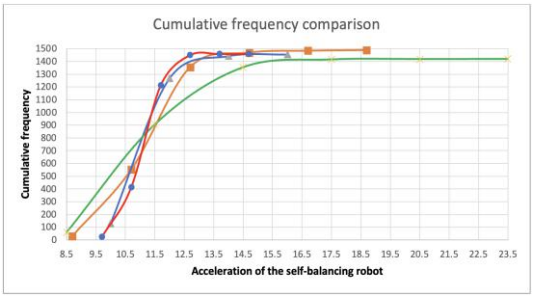

Figure 7 Cumulative frequency curves of the four materials

Scrutinizing Figure 7, the elastic materials i.e., Rubber and TPU have curved cumulative lines. PLA and hard plastic have lower tensile strength and thus more linear increases in the CDF.

The largest increase in cumulative frequency for each material is in the class interval with the mean acceleration i.e. 11.1- which is also the mode.

B. Vibrational Study

From the output data of the Serial Monitor, the acceleration values in the three axes were analyzed using an FFT algorithm. Fast Fourier Transform essentially converts the accelerometer readings into vibrational frequency by plotting the magnitude against the frequency. The graphs of acceleration and motor speeds show the variation in these values over a period of time. These values have been plotted from a database where each of the four materials was tested out on dry concrete and the time period of their motion readings was extracted with 100 readings per second.

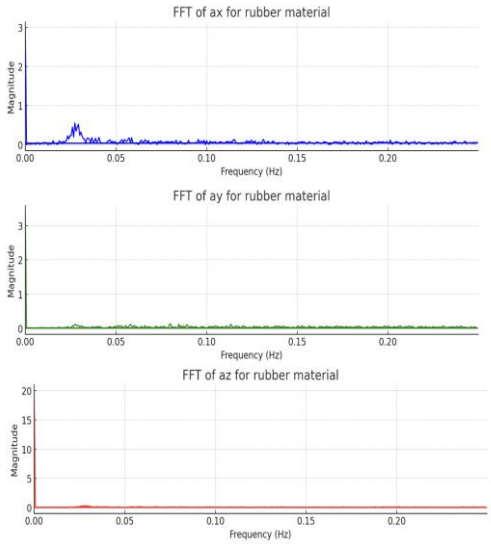

Figure 8 Fast Fourier Transform graph of accelerometer values for Rubber

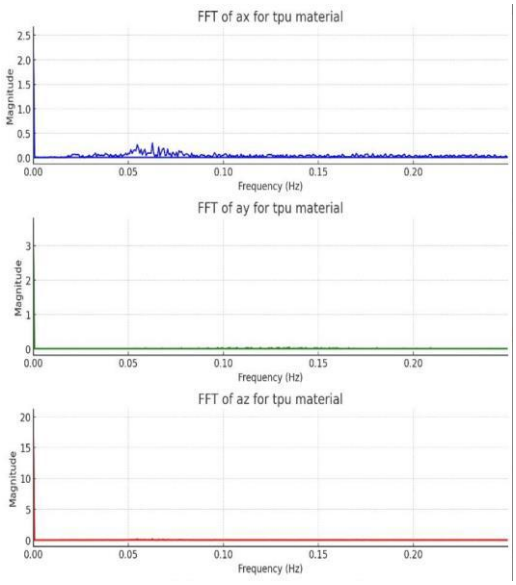

Figure 9 Fast Fourier Transform graph of accelerometer values for TPU

The accelerations start at a high magnitude at 0 meaning that the robot wasn't level at the beginning. The values for rubber remain peak for $a_x a_x$ at about 0.025Hz indicating that there were quick oscillations with rubber in correcting the robot's position. Disturbance for TPU is at higher frequencies for $a_x a_x$ but for $a_y a_y$ and $a_z a_z$, is minimal compared to all materials. This pattern is consistent for all four materials showing that they can successfully balance the robot.

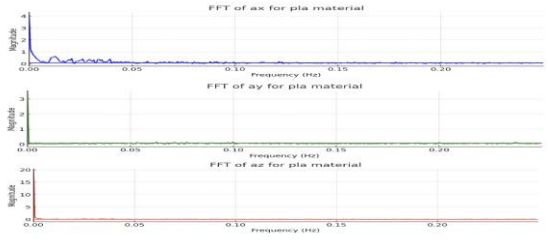

Figure 10 Fast Fourier Transform graph of a_x for PLA

The $a_x a_x$ value for PLA has small peaks at lower frequencies showing that it has weak oscillations and drifts over various frequencies and has slower responsiveness to alterations in the inclination angle.

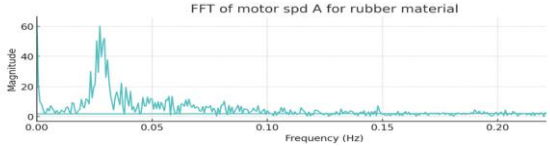

Figure 11 Fast Fourier Transform graph of Motor speed Rubber tire A

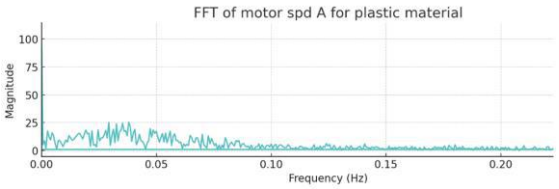

Figure 12 Fast Fourier Transform graph of Motor speed plastic tire A

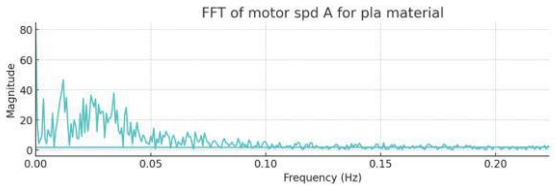

Figure 13 Fast Fourier Transform graph of Motor speed PLA tire A

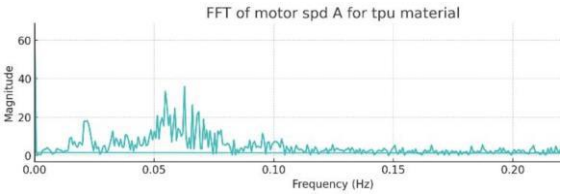

Figure 14 Fast Fourier Transform graph of Motor speed TPU tire A

Rubber displays high frequencies for the motor speed of motor A at approximately 0.025Hz. This must be a resonant frequency i.e., when the natural frequency of rubber is equal to the forced frequency of it. The motor speed of TPU graphs also exhibits a resonant frequency after 0.05Hz. PLA exhibits higher magnitudes at lower frequencies.

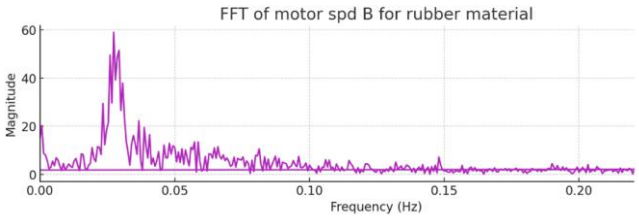

Figure 15 Fast Fourier Transform graph of Motor speed Rubber tire B

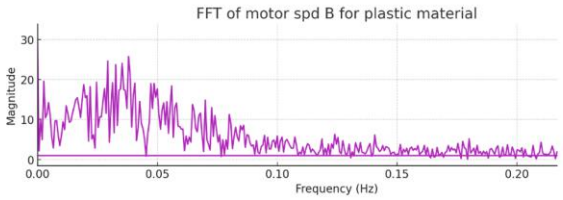

Figure 16 Fast Fourier Transform graph of Motor speed plastic tire B

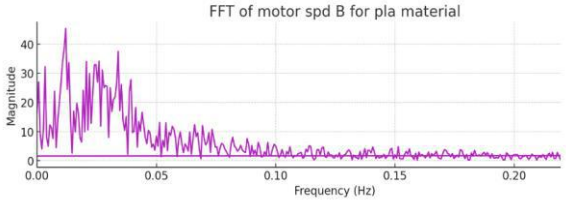

Figure 17 Fast Fourier Transform graph of Motor speed PLA tire B

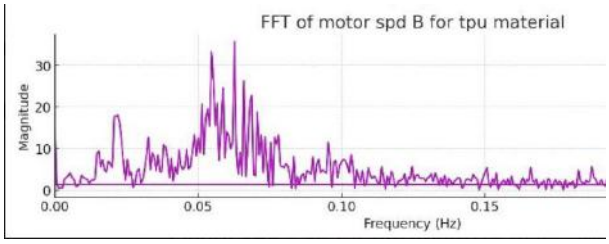

Figure 18 Fast Fourier Transform graph of Motor speed TPU tire B

The FFT graphs for Motor speed of Motor B gave further insight. Once again, the peak of rubber and TPU is prominent. However here, the magnitudes of frequencies for both hard plastic and PLA are high at low frequencies indicating the control mechanisms for both the materials to stabilise the robot against various disturbances. Plastic shows mild disturbance in the higher frequency range too showing more disturbance.

V. CONCLUSION

From the graphical analysis, TPU shows the least deviation from its mean acceleration, and it can be 3D printed easily into selective patterns and sizes if further analysis had to be done, however, it did show high sensitivity to minute changes in inclination angle. Rubber is also a suitable wheel material because although its bell curve was broader, it didn't show disarray as plastic did. Hard plastic and PLA both did help the robot balance but due to the brittleness and lack of shock absorption for the materials, they faced challenges and had slow responsiveness to error correction, which is crucial for the self-balancing robot. The mass distribution was also a rudimentary factor for this paper, as shown by the Lagrangian equations of motion. The battery pack being on the bottom chassis served purposefully to give the robot a lower center of gravity and keeping the MPU6050 gyro-sensor on top helped in detecting small angle differences better and thus making the entire process faster. In the future, focus can be put on the size of the tires too, or expanding the types of materials. Testing can be carried out on different factors that would alter the coefficient of friction- such as lubrication of the surface. The robot can be tested in various conditions and on rough surfaces to observe the robot's adaptability and stability with high friction. On the contrary, they can also be tested on almost frictionless surfaces. That range in results will be deemed to produce an ameliorated determinant of the ideal tire material.

ACKNOWLEDGMENT

I would like to express my gratitude to the mentors of On My Own Technology Pvt. Ltd. for extending their help in carrying out this project.

REFERENCES

1. Zheng, Bang Gui, Yi Bin Huang, and Cong Ying Qiu. "LQR+ PID control and implementation of two-wheeled self-balancing robot." Applied Mechanics and Materials 590 (2014): 399-406
2. Barapatre, Mamta M., and V. N. Sahare. "A Review on Various Methods for Self Balancing." IJCSN International Journal of Computer Science and Network 4.1 (2015)
3. Wu, Junfeng, and Wanying Zhang. "Design of fuzzy logic controller for two-wheeled self-balancing robot." Proceedings of 2011 6th international forum on strategic technology.

Vol. 2. IEEE, 2011.

4. Guo, Linyuan, Syed Ali Asad Rizvi, and Zongli Lin. "Optimal control of a two-wheeled self-balancing robot by reinforcement learning." International Journal of Robust and Nonlinear Control 31.6 (2021): 1885-1904.
5. Thao, Nguyen Gia Minh, Duong Hoai Nghia, and Nguyen Huu Phuc. "A PID backstepping controller for two-wheeled self-balancing robot." International Forum on Strategic Technology 2010. IEEE, 2010.
6. Ramchandra, Y., et al. "Self-balancing robot using Raspberry Pi and PID controller." International Journal of Innovative Science and Research Technology 6.4 (2021): 321-322.
7. Supekar, Aditi, et al. "Review on Self-balancing Robot Navigation."
8. Han, Jianhai, Xiangpan Li, and Qi Qin. "Design of two-wheeled self-balancing robot based on sensor fusion algorithm." International journal of automation technology 8.2 (2014): 216-221.
9. Kotiyal, Bandanawaz, et al. "Self balancing robot." (2015).
10. Kumra, Sulabh, and Shilpa Mehta. "Singular Axis Self Balancing Robot." Journal of Image Processing and Vision Science: Vol 1.4 (2012): 9.
11. Yi, Jingang. "A piezo-sensor-based "smart tire" system for mobile robots and vehicles." IEEE/ASME transactions on mechatronics 13.1 (2008): 95-103.
12. Gandarilla, Isaac, et al. "Trajectory tracking control of a self-balancing robot via adaptive neural networks." Engineering Science and Technology, an International Journal 35 (2022): 101259.
13. Ruan, Xiaogang, and Wangbo Li. "Ultrasonic sensor based two-wheeled self-balancing robot obstacle avoidance control system." 2014 IEEE International Conference on Mechatronics and Automation. IEEE, 2014.
14. Adeel, Umar, et al. "Autonomous dual wheel self balancing robot based on microcontroller." Journal of Basic and Applied Scientific Research 3.1 (2013): 843-848.
15. Tomašić, Tomislav, Andrea Demetlika, and Mladen Crneković. "Self-balancing mobile robot tilter." Transactions of FAMENA 36.3 (2012): 23-32.