International Journal of Science and Research Archive
eISSN: 2582-8185
Cross Ref DOI: 10.30574/ijsra
Journal homepage: https://ijsra.net/

(Research Article)

Check for updates

Integration and High stabilization process of Self-Balancing Intelligent Autonomies Vehicles

Md Norul Amin [1,*], Isfat Jahan [1], Rajib Miah [1], Ebrahim [2] and Monika [3]

[1] *Department of Electronics Information Engineering, China West Normal University, Nanchong, Sichuan, China.*
[2] *Department of EEE, Southeast University, Dhaka, Bangladesh.*
[3] *Department of Political Science, Tejgaon College, Dhaka, Bangladesh.*

International Journal of Science and Research Archive, 2025, 14(02), 086-091

Publication history: Received on 17 December 2024; revised on 31 January 2025; accepted on 02 February 2025

Article DOI: https://doi.org/10.30574/ijsra.2025.14.2.0284

Abstract

The design and implementation of self-balancing wheels using the TM4C123GH6PZ microcontroller are demonstrated in this study. The vehicles were designed with a motor driver circuit for wheel control, a dual-wheel mechanism for balancing and propulsion against Earth's gravity, the MPU-6050 gyroscope/accelerometer for tilt detection, and high-precision proportional-integral-differential (PID) controlling algorithms for stability. The system's remarkable 95.7% stabilization accuracy throughout testing showed steady performance under a variety of dynamic circumstances and tilt angles. The study demonstrates that it is possible to build highly stable autonomous cars with affordable, readily available parts without sacrificing performance. According to this work, corresponding ideas can be modified for many different kinds of uses, including instructional robots and personal transportation solutions.

The architecture of the suggested system establishes a standard for upcoming advancements in autonomous vehicle technology and provides insightful information about the potential of utilizing microcontroller and sensor technologies to create sophisticated, economical robotic applications.

Keywords: TM4C123GH6PZ; Self-Balancing Car; MPU-6050; PID Control; Two-Wheel Stabilization

1. Introduction

Two-wheeled systems can stay upright thanks to self-balancing technology, which uses sensors and a control algorithm to continuously change the vehicle's orientation [1]. With its ARM Cortex-M4 core, the TM4C123GH6PZ microcontroller acts as the brain of the system, processing data from an MPU-6050 sensor (gyroscope and accelerometer) in real time and using motor drivers to apply corrections [1]. Even with fluctuating loads or disturbances, strong stabilization is guaranteed using a Proportional-Integral-Derivative (PID) control method in conjunction with sturdy hardware [2]. Even in the face of outside disruptions, the microcontroller maintains equilibrium by processing sensor data and modifying motor outputs in real-time. The purpose of this study is to show that such a design is feasible, highlighting its effectiveness, simplicity, and versatility across a range of applications [3].

2. Materials and Methods

Twelve gray sensors for line detection, two motors for movement, a power source, a motor driver circuit for motor control, and the TM4C123GH6PZ microcontroller (MCU) acting as the central processing unit are the parts needed to finish the project.

* Corresponding author: Md Norul Amin.

2.1. System Components and Architecture

- **Microcontroller:** A 32-bit ARM Cortex-M4 with a floating-point unit power the TM4C123GH6PZ microcontroller, which performs exceptionally well in PID computations [4].Its high-speed processing allows for quick reaction to sensor input, which is essential for preserving balance, and its UART, I2C, and SPI interfaces facilitate smooth sensor and actuator integration [4] [5].
- **MPU-6050 Sensor Module:** Pitch and angular velocity information that are essential for balance are provided by the 6-axis sensor, which combines a 3-axis gyroscope with a 3-axis accelerometer [6]. In order to measure tilt angle (pitch) and angular velocity to determine the orientation of the car, it is connected to the TM4C123GH6PZ via I2C for effective real-time control [7].
- **L298N Motor Driver:** Using an H-Bridge, the motor driver uses the microcontroller's PWM signals to determine the direction and speed of a DC motor. It guarantees precise, fluid motor operation to offset tilts and preserve equilibrium [8].
- **DC Motors with Encoders:** In order to provide smooth and exact balancing, encoders offer input for precise speed regulation, while high-torque motors provide propulsion and corrective movement [9].
- **Power Supply:** All components receive dependable power from a lithium-ion battery, guaranteeing continuous functioning in a range of circumstances.

2.2. Working Principle

2.2.1. Sensor Integration

The tilt angle and angular velocity of the system are continuously monitored by the MPU-6050:

- Tilt Angle Calculation: The car's pitch angle is estimated by the accelerometer using gravity data [7].
- Rate of Change: The gyroscope measures the speed at which the angle changes [10] [6].

To produce precise and consistent tilt data, these inputs are filtered using a Kalman filter or complementary filter [11].

2.2.2. Control Algorithm

Initialization Phase

- **System Initialization**
  - Set up the TM4C123GH6PZ GPIO pins for motor driver control, I2C connection, and PWM output [12].
  - To create a steady tilt baseline, initialize the MPU-6050 sensor via I2C and calibrate the sensor.
  - Establish global variables for motor speed, PID coefficients, tilt angle, and angular velocity.
- **Set PID Parameters:**
  - Determine the coefficients for Kp, Ki, and, and Kd according to the needs of the system.
  - Set up variables for time interval (Δt), cumulative error ($\int e(t)$), and preceding error ($eprev$) [13].
- **Pre-stabilization Check:**
  - Confirm that sensor outputs fall within the range of operation.
  - Verify that PWM signals and motors are operating properly [14].

2.2.3. Feedback and Control Loop

- **Read Sensor Data:**
  - Retrieve the MPU-6050's raw accelerometer and gyroscope data.
  - Use the accelerometer and the gyroscope's angular velocity to determine the tilt angle [15] [16].
  - To combine data for a steady tilt angle (θ), use a complementary or Kalman filter.
- **Calculate Error:**
  - Determine the tilt error:

$$e(t) = \theta_{desired} - \theta$$

  - Modify the total error:

$$\int e(t) = \int e(t) + e(t) \times \Delta t$$

  - Determine the error change rate:

$$\frac{de(t)}{dt} = \frac{e(t) - e_{prev}}{\Delta t} \text{ [17]}$$

- **Compute PID Output:**
  - Determine the control signal: A PID controller constantly modifies motor speeds in response to sensor feedback to maintain balance.
  - Proportional (P): Adjusts in accordance with the magnitude of the tilt error in response to the current tilt error.
  - Integral (I): Corrects long-term drift by adding together previous errors.
  - Using the current rate of change as a basis, the derivative (D) forecasts future inaccuracy.
  - Constrain $u(t)$ to prevent excessive motor speeds.

- **Adjust Motor Speed and Direction:**
  - For both motors, provide PWM signals corresponding to u(t)
  - Determine the motor's direction based on u(t) sign:
    - Positive: Advance to offset the backward tilt.
    - Negative: To counteract the forward tilt, move backward

2.2.4. Mathematical Representation:

$$\boldsymbol{u(t) = K_p \times e(t) + K_i \int e(t)\, dt + K_d \frac{de(t)}{dt}} \quad [18]$$

Where
- $u(t)$: Control signal (motor adjustment).
- $e(t)$: Tilt error.
- K_p, K_i, K_d: PIDcoefficients.

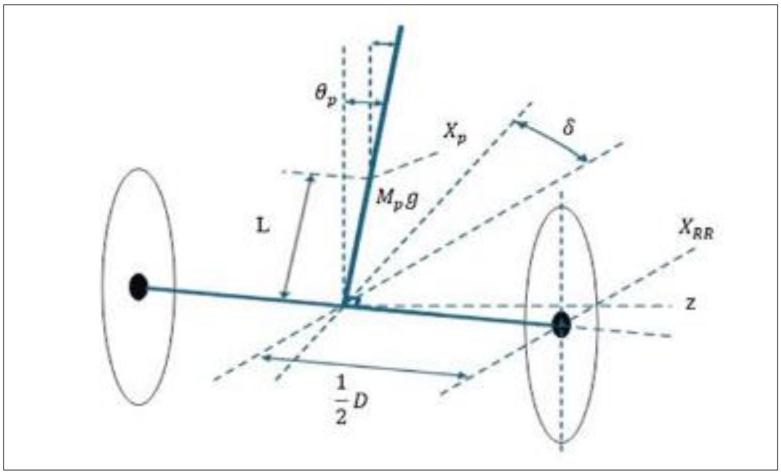

Figure 1 Mathematical Identification of Balancing System.

This self-balancing diagram is mathematical mechanism of robotic vehicle.

There are uses some parameter:

- θp: The pendulum or balancing mechanism's pitch angle.
- Mpg: The gravitational pull on the mass of the pendulum.
- Xp: The pendulum's horizontal displacement.
- δ: The system's steering angle, which affects movement direction.
- X_{RR}: The displacement of the rear-right wheel, signifying its location.
- L: The separation between the pivot and the center of mass of the pendulum.
- 1\2*D: The distance between the two wheels, or half the track width.
- z: The system's offset or lateral displacement.

According on the arrangement, the system adjusts wheel positions, θp, and δ to maintain balance. It is frequently employed to keep two-wheeled robots, Segways, and other similar equipment stable. A self-balancing technique continuously modifies the orientation and position of a device, like a robot or car, using feedback control systems to preserve stability. Gyroscopes and accelerometers are examples of sensors that pick-up tilt or displacement, while actuators (motors) balance things out by opposing forces. State-space approaches, which are frequently represented as an inverted pendulum system, and proportional-integral-derivative (PID) control are examples of common control strategies.

2.2.5. Motor Control

PWM signals are produced by the TM4C123GH6PZ in order to:

- Modify motor speed in response to PID output.
- To offset tilt, adjust the direction (forward or backward) [19].

PWM signals were generated to counteract tilts for balance and modify motor speed and direction based on PID output. These impulses are amplified into high-power outputs by the motor driver, allowing for reliable and accurate motor control [19].

2.2.6. Stabilization Workflow

- Initialization: The MCU sets up PWM channels for motor control, GPIO pins, and I2C communication for the MPU-6050. Accurate baseline readings are ensured by initial tilt calibration [19].
- Feedback Loop: To calculate tilt error, sensor readings are analyzed in real time. The required motor modifications are calculated by the PID controller. The system is stabilized by updating the PWM signals [8].
- Disturbance Handling: Increasing motor speed or changing direction corrects abrupt tilts or load variations. The PID guarantees oscillation-free, fluid responses

Figure 2 Self-balancing full assembled car

2.2.7. High-Stabilization Process

For quick error identification, the MPU-6050's accelerometer and gyroscope provide accurate angular readings. The TM4C123GH6PZ maintains equilibrium under dynamic circumstances by processing data in real time with low latency. Smooth stabilization is achieved by minimizing oscillations and overshooting by optimal PID tuning of (Kp), (Ki), and (Kd).

3. Results and Discussion

Table 1 Movement accuracy results from individual 5 tests.

Environment	Load	Tilt Angle (°)	Stabilization time	Oscillations	Power Consumption	Stability
Flat Surface,	no	5	0.8	1	4.5	98.9%
	yes	10	1.0	1	5.0	98.6%

Inclined	no	15	1.2	2	5.5	96.1%
	yes	20	1.5	3	6.0	93.2%
Sudden Disturbance	High load	25	1.8	3	6.5	91.7%
Average total Stability						95.7%

3.1. Stability Calculation Formula:

$$\text{Stability Persentage} = \frac{\text{Max Stability (Ideal Case)}}{\text{Deviation from Ideal Performance}}$$

With the highest stability of 98.9% on a level surface without any weight, the results show that the self-balancing system maintains high stability in all test conditions. With stability percentages of 98.6% at 10° tilt, 96.1% at 15°, 93.2% at 20°, and 91.7% under abrupt disruption, stability somewhat declines as tilt angle and load increase. These changes are anticipated because more difficult conditions will result in longer correction times, oscillations, and higher power usage.

4. Conclusion

This study used the MPU-6050 sensor and TM4C123GH6PZ microcontroller to successfully create a two-wheeled self-balancing vehicle. High stabilization precision was attained with efficient tilt correction made possible by the inclusion of PID control. This method shows the possibility of effective, affordable solutions for self-balancing robotics and autonomous system applications.

Compliance with ethical standards

Acknowledgments

I want to express my sincere gratitude to my advisors, Mr. Dezhi Guo and Dr. Nouman Rasul, for reviewing the test report and for their invaluable advice and support during this project. Additionally, I would like to thank Dr. Fang, our lab technician, for his assistance with hardware setting and troubleshooting. I especially appreciate my students' contributions, as their collaboration and dedication were crucial to the accomplishment of this project. I am appreciative of China West Normal University for providing the equipment and workspace I required for the research. In addition, I want to thank my family and friends for their unwavering support and encouragement.

I would like to thank Mr. Dezhi Guo and Dr. Nouman Rasul, my advisors, for their help and direction during this endeavor. I am also grateful to China West Normal University for providing lab resources and facilities, and to Dr. Fang for helping with hardware debugging.

Disclosure of conflict of interest

No conflict of interest to be disclosed.

References

[1] L. G. Arbnor Pajaziti, "Navigation of Self-Balancing Mobile Robot through Sensors," IFAC-PapersOnLine, https://doi.org/10.1016/j.ifacol.2019.12.576, vol. 52, no. 25, p. 429–434, 2019.

[2] K. S. A. Sharma, "Design of Two-Wheel Self-Balancing Vehicle Based on Visual Identification," EURASIP Journal on Image and Video Processing, vol. 2021, no. 35, p. 1–10, 2021.

[3] P. A. R. Kumar, "Two-Wheel Self-Balancing of a Four-Wheeled Vehicle," IEEE Control Systems Magazine, vol. 38, no. 3, pp. 45-54.

[4] T. I. Incorporated, "TM4C123GH6PZ data sheet, product information and support," TI.com, 2024.

[5] Serkan, "Programming TM4C123GH6PZ MCU with FTDI - Arm-based microcontrollers," TI E2E support forums.

[6] K. Madhira, A. Gandhi and A. Gujral, "Self-balancing robot using complementary filter: Implementation and analysis of complementary filter on SBR," in International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), Chennai, 2016.

[7] J. Han, X. Li, and Q. Qin, "Design of Two-Wheeled Self-Balancing Robot Based on Sensor Fusion Algorithm," Int. J. Automation Technol, vol. 8, no. 2, pp. 216-221, 2013.

[8] Dejan, "L298N Motor Driver - Arduino Interface, How It Works, Codes, Schematics," How To Mechatronics, 2022.

[9] R. plc., "Renishaw: Tokyo Denki University Case Study," Renishaw, 15 November 2024.

[10] P. Narkhede, S. Poddar, R. Walambe, G. Ghinea, and K. Kotecha, "Cascaded Complementary Filter Architecture for Sensor Fusion in Attitude Estimation," MDPI, vol. 21, no. 6, p. Sensors, 2021.

[11] T. Instruments, "TM4C123GH6PZ Microcontroller Datasheet,".

[12] J. G. Ziegler, and N. B. Nichols, "Optimum settings for automatic controllers," Transactions of the ASME, vol. 64, no. 11, p. 759–768, 1942.

[13] T. Instruments, "PWM Control in Embedded Systems," Application Note SPRAA88A, 2009.

[14] S. O. H. Madgwick, "An efficient orientation filter for inertial and inertial/magnetic sensor arrays," Report X-IO and the University of Bristol, 2010.

[15] B. G. Welch, "An introduction to the Kalman Filter," UNC-Chapel Hill Technical Report, TR, 2006.

[16] Mark W Spong and M. Vidyasagar, "Robot Dynamics and Control," Wiley, 2008.

[17] N. T, and P. K T, "PID Controller Based Two Wheeled Self Balancing Robot," in 5th International Conference on Trends in Electronics and Informatics (ICOEI), 2021.

[18] U.P.R.F, "L298N Self Balancing robot problem," Arduino Forum, June 2015.

[19] L. W. and X. W. J. Zhang, "Design and Implementation of a Two-Wheeled Self-Balancing Car Using a Fuzzy Kalman Filter," Sensors, vol. 20, pp. 1256-1269, 2020.