AL-Bahir
Journal for Engineering and Pure Sciences
BJEPS

Volume 5 | Issue 2 Article 3

Review of Hardware Implementation for the Two-Wheeled Self-Balancing Robot

Ghaidaa Hadi Salih Elias
College of Computer Science and Information Technology, Kerbala University, Kerbala, Iraq

Recommended Citation
Elias, Ghaidaa Hadi Salih (2024) "Review of Hardware Implementation for the Two-Wheeled Self-Balancing Robot," *Al-Bahir*: Vol. 5: Iss. 2, Article 3.
Available at: https://doi.org/10.55810/2313-0083.1073

Review of Hardware Implementation for the Two-Wheeled Self-Balancing Robot

Source of Funding
Not found funding

Conflict of Interest
The authors declare no conflict of interest

Data Availability
Publicly available data

Author Contributions
by assist researchers in building and designing two-wheeled mobile robots in the future

REVIEW

Review of Hardware Implementation for the Two-wheeled Self-balancing Robot

Ghaidaa H. Salih Elias

College of Computer Science and Information Technology, Kerbala University, Kerbala, Iraq

Abstract

The working principle of a self-balancing robot is similar to that of an inverted pendulum, with the mobile robot's controller playing a crucial part in both self-balancing and stabilization. It is the kind that constantly modifies itself to keep balance when it rides on two wheels. This review will center on the basic construction of the suggested robot, summarize the recent studies relative to control methods and the building of the two-wheeled self-balancing robot, and discuss the outcomes of control experiments conducted on hardware systems. It will assist researchers in building and designing two-wheeled mobile robots in the future.

Keywords: Arduino Uno, DC motor driver, DC gearmotor, Two-wheeled robot

1. Introduction

Robotic systems are being developed quickly to assist humans in their professions nowadays. So, many studies are being conducted to create a wide variety of robots with varying dimensions, forms, and motions. From a fixed robot that remains in one spot to a mobile robot that can move freely in any direction, one of these robotics' applications that this paper concerns is the two-wheeled self-balancing robot.

In the scientific community, the control and design of the two-wheeled self-balancing robot have gained wide popularity. It is an unstable system due to the fact that it has a center of gravity above the center point. Besides, it is a nonlinear system, and it is considered an underactuated system (less control inputs than degrees of freedom, which need to be controlled). According to this problem, classical control strategies make it difficult to apply them to that type of system. Therefore, the two-wheeled self-balancing robot is a good application for researchers to validate the power of different control techniques [1].

Many researchers have built different prototypes of a self-balancing robot so it can balance itself. The contribution of this review is represented in presenting recent studies relative to the implementations of the two-wheel balancing robot in real time, in order to help future researchers know the basic components of building a self-balancing robot and the best control methods that were conducted on the proposed robot to keep it balanced on the floor with the expectation that it would be under external disturbances [2].

The review paper is arranged as below: The architecture of the robot and the comparison table of the literature are detailed in section 2. Finally, the review conclusion is involved in section 3.

2. Robot constructing

The suggested robot has been constructed using both hardware and software components. The hardware components involve a microcontroller, DC motors, a DC motor driver, and sensors. In return, software components include the code related to the control system, the motor driver, and sensor devices, which is written with a suitable programming language, and then it is uploaded to the microcontroller. The architecture of the two-wheel self-balancing robot is represented in Fig. 1.

The microcontroller is like the heart of the robot, and it can be an Arduino Uno, Arduino Mega,

Received 26 May 2024; revised 15 August 2024; accepted 16 August 2024.
Available online 11 September 2024
E-mail address: gadahadi6@gmail.com.

https://doi.org/10.55810/2313-0083.1073

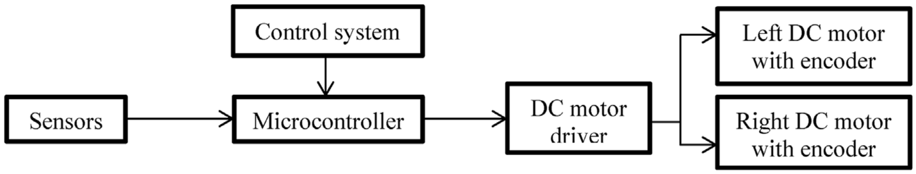

Fig. 1. Two-whell self-balancing robot construction.

Arduino Nano, Field Programmable Gate Array (FBGA), or any integrated circuit that can be programmed. DC motors are used to generate the motion for the robot, and it is controlled by the driver through a Pulse Width Modulation signal (PWM) to drive the wheels with different speeds and directions. As that, the encoder is employed in the feedback control system for providing the driver with DC motor speed. Beside, the sensors provide the robot with information from the robot body itself as well as the environment. One of the common sensors used in the two-wheel self-balancing robot

Table 1. Literature of two-wheel robot configuration compontes (Microcontroller, motors, motor driver, sensors, and communication module).

Ref/Year	Proposed microcontroller	Motors type	DC motor driver type	Sensors used	Communication module	Other componts
[3]/2017	Arduino Uno	DC gearmotor	Not determine its type	MPU6050	USB connection	Not found
[4]/2018	Atmega 328	DC gearmotor	L298	MPU6050	USB connection	Not found
[5]/2018	32-bit ARM7 microprocessor	DC motor with encoder DC servo motor	Not determine its type	MPU6050	HC-05 bluetooth module is used as a communication tool between the computer and the robot in order to transmit data in real time. Beside that, the robot is also controlled remotely via the installed program on an Android phone	A voltage regulator circuit is employed to generate different levels of voltage for the mobile robot
[6]/2018	Open FPGA (Icezum Alhambra II) as well as Arduino Nano	DC gearmotor	MC33926	MPU6050	USB connection	Not found
[7]/2019	Arduino Uno	DC motor with gearbox	Not determine its type	MPU6050	USB connection	Not found
[8]/2020	ESP8266 NodeMCU 1.0	DC gearmotor	L298N	MPU6050	A web server is created to control the robot via a web page and mobile application	Not found
[9]/2020	Arduino Uno	DC gearmotor	L298N and ULN2003	MPU6050	HC-05 bluetooth module	Not found
[10]/2020	ESP32 control card	Stepper bipolar motor	DRV8825	IMPU6050	Remote control is created through a WiFi connection between a control unit and a mobile phone or PC	Not found
[11]/2020	Arduino Nano	Stepper motor	A4988	MPU6050	HC-05 bluetooth module	Voltage regulator, capacitors
[12]/2021	Arduino Mega	DC motor with encoder	L298N	MPU6050, ultrasonic sensor	USB connection	LCD 1602 display, potentiometer, buck modules
[13]/2021	Arduino Mega	DC motor with gearbox and encoder	L298N	MPU6050	USB connection	Not found
[14]/2021	Arduino Uno	DC motor with encoder	L298N	MPU6050	USB connection	Not found
[15]/2022	Arduino Uno	DC motor with encoder and gearbox	L298P	MPU6050	USB connection	Not found
[16]/2023	Arduino Mega	DC motor with encoder and gearbox	LM298	Not found	HC-05 bluetooth module	LM2596 DC step down converter

is the Inertial Measurement Unit (IMU) MPU6050. That sensor inculdes the accelerometer and gyroscope, where the accelerometer is used to measure acceleration along the robot body axis, while the other is used to measure angular velocity along the robot body axis, too. There are different control systems implemented to control the two-wheel self-balancing robot, such as the Proportional Integral Derivative (PID), Linear Quadratic Regulator (LQR), Model Reference Adaptive Control (MRAC), so that the suggested robot could balance itself. The review has been divided into two parts: the components that formed the robot (the microcontroller type, motors, sensors, and communication module), which are inserted in Table 1. On the other hand, the second part provides control methods used, comparison control methods, and discussion of the outcomes of the control experiments of the hardware systems of the two-wheeled balancing robot, as clarified in Table 2.

Table 2. Literature on the application of two-wheel robot control methods.

Ref/Year	Proposed controller	Compared methods	External disturbances test	Results
[3]/2017	LQR with precompensation	PID, LQR	Not found	The suggested controller and the comparison controller are modeled using the inverted pendulum theory and simulated with the MATLAB program Simulink. In practical tests, they prove that the system response of the LQR controller with precompensation gave better performance with the pendulum's vertical angle and a settling time of less than 5 radians and 1 s, respectively.
[4]/2018	PID	LQR, pole placement, PID	Not found	The LQR, pole placement, and PID controllers are simulated with the MATLAB program, and the comparative results between them proved that the steady-state response of the LQR and pole placement is superior to that of the PID controller. However, the researchers found it difficult to implement them in real time as they required representing the system in a state space model. On the other hand, a PID controller is durable and simple to implement when compared to other controllers. When implemented in hardware, it has a high ability to balance and respond quickly, although the integral control system causes the overshoot issue. However, the expected design algorithm effectively regulated the robot's balance.
[5]/2018	PD controller for position and adaptation fuzzy controller based on relation models for balance of the two-wheeled self-balancing robot	fuzzy controller	Yes	The tests of the suggested robot were performed with simulation and hardware implementation. The simulation outcomes demonstrate that the robot's position approaches the equilibrium point and remains stable after 1.3 s for the non-adaptive fuzzy control system and 0.7 s for the fuzzy control system. Beside, the experimental results explain that the adaptive fuzzy controller performs well in terms of rapid reaction, good balance, and robustness against an external disturbance, which is represented by pushing the robot slightly rearward. Before and after the external disruption, both the non-adaptive and adaptive fuzzy controllers are capable of maintaining the robot in equilibrium, but when the adaptive fuzzy controller is used, the robot's pitch angle and position change over narrower ranges. The robot's regulation time is limited to 0.4 s when employing the non-adaptive fuzzy controller and to 0.3 s while utilizing the adaptation fuzzy controller.
[6]/2018	Proportional Derivative (PD)	Nan	Not found	In this study, the researchers applied the PD controller for the suggested robot. Thus, the control implementation, which was designed for a two-wheeled robot system, has succeeded in keeping its balance even in the face of external disturbances.
[7]/2019	PID with fuzzy logic	Nan	Not found	The fuzzy logic control system with the PID controller achieved control accuracy, balance, and flexibility for the two-wheeled balancing robot system. Reducing the proportional gain and adding in the integral gain allowed the oscillation amplitude to be dampened, and the robot would balance with minimal jitter of less than 1 cm. Also, the overshoot was dampened by adding the derivative gain. The suggested robot also replied to an external disturbance effect, which is represented by pushing the robot with a finger. The two-wheel robot recovered balance fast within 1 s; however, the extreme external disturbance caused it to vacillate wildly and then drop over. In this state, the study found that the robot can follow an external disturbance with less force.

(*continued on next page*)

Table 2. (continued)

Ref/Year	Proposed controller	Compared methods	External disturbances test	Results
[8]/2020	PID	Nan	Not found	The two-wheeled self-balancing robot system has been successfully implemented. The most crucial aspect of this project is to build the control wireless for the robot by implementing a control system application in C# as well as to create a webserver to control the robot via web page and mobile application.
[9]/2020	PID	Nan	Not found	The control system application for the robot was built through a wireless connection (bluetooth) and an Android device application, as the two-wheel robot's ability to balance itself and carry out required commands was experimentally confirmed, and the robot's response time was less than 2 s. Furthermore, the tracking error of the pitch angle response was less than 5 deg during the movement of the robot.
[10]/2020	PID	LQR	Not found	In this study, two control systems, the PID and LQR, have been simulated by the Matlab program Simulink and implemented wirelessly through the wifi network for connecting the microcontroller node with the computer system to make the two-wheel balance robot stable irrespective of disturbances. The response of the LQR control was improved for controlling the robot's vertical position, which is resistant to disturbances. It has less overshoot and can reach the control design specifications in 0.2 s, and the vertical tilt angle for the self-balancing wheel robot is less than 5 deg.
[11]/2020	PID	Nan	Not found	The control system application for the robot was built through a wireless connection (bluetooth) and an Android device application as the researchers succeeded in implementing the control for the two-wheeled self-balancing robot system. They concluded that these applications would add excitement to the learning process and serve as a key differentiator from specialized robots that perform these jobs using four wheels.
[12]/2021	PID	Nan	Not found	The PID control system was designed and tuned through the trial and error method. As well, it was implemented through the Arduino Mega to minimize the balance error relative to the proposed robot. As the researchers found, in the case of operating motors with different speeds, the adjusting of the PID controller coefficients was changed continuously to make the robot regain its balance.
[13]/2021	Adaptive controller by using the radial basis function (RBF) neural networks and the sliding mode control technique	Sliding mode control (SMC)	Not found	The proposed control system was proven to be stable by using the Lyapunov theory, and then it was tested with simulation and real-time. Experimental results of the SMC with the RBF response provide better performances with a percentage of 19.76%, 47.55% of the tilt and heading angle errors, respectively, and 89.013% of the distance error compared to the SMC approach.
[14]/2021	MRAC	Nan	Yes	The experiment outcomes demonstrate that the suggested control response could track the desired tilt angle with settling and rise times of 0.87 s and 0.27 s, respectively. Furthermore, the suggested controller could overcome the external disturbance effect that is represented by forcing the plant system by hand, which leads to its slope of 6°. So, the system takes 4.72 s to return to its original position.
[15]/2022	PID	Nan	No	The tuned PID controller is implemented for stabilizing the two-wheeled self-balancing robot. An optimization problem is used to estimate the parameters of the model for matching the measurements in real time accurately. The results demonstrated that the robot's measured parameters were adjacent to the robot's estimated parameters.
[16]/2023	LQR	Nan	No	The LQR control system is simulated for the robot via Matlab (Simulink), and it is applied in hardware implementation. It was tested in keeping on balancing, forward moving, backward moving, turning left, and turning right. In most tests, the suggested control system proves effective in stabilizing the system.

3. Conclusion

This review presents a group of modern studies relative to two-wheel self-balancing robot applications in hardware implementation. The review concludes that most of the researchers suggested Arduino as a microcontroller due to its low cost and ease of programming. Besides that, most researchers suggested DC motors with their driver (L298N) for consuming a low amount of power. All papers mentioned in the review used a MPU6050 sensor to achieve high stability and reliability for the control performance of the robot. As well, the review concluded that most studies suggested the PID controller for keeping the robot balanced due to the simplicity of its design and programming on a microcontroller board. Finally, this review will be considered in the future as a key reference for the researchers relative to the components of the two-wheel self-balancing robot's construction and control theory implementation experimentally for the proposed robot.

Data availability

Not applicable.

Code availability

Not applicable.

References

[1] Guo L, Rizvi SAA, Lin Z. Optimal control of a two-wheeled self-balancing robot by reinforcement learning. Int J Robust Nonlinear Control 2021;31(6):1885–904. https://doi.org/10.1002/rnc.5058.
[2] Gul F, Rahiman W. Mathematical modeling of self balancing robot and hardware implementation. In: Proceedings of the 11th international conference on robotics, vision, signal processing and power applications: enhancing research and innovation through the fourth industrial revolution; 2022. p. 20–6. https://doi.org/10.1007/978-981-16-8129-5_4.
[3] Johnson AM, N A. Design and development of a two wheeled self balancing robot for an application of object carrying. Int J Eng Res Technol 2017;6(7):580–7. https://doi.org/10.17577/IJERTV6IS070240.
[4] Imtiaz MA, Naveed M, Bibi N, Aziz S, Naqvi ZSH. Control system design, analysis & implementation of two wheeled self balancing robot (TWSBR). 2019. https://doi.org/10.1109/IEMCON.2018.8614858.
[5] Mai TA, Anisimov DN, Dang TS, Dinh VN. Development of a microcontroller-based adaptive fuzzy controller for a two-wheeled self-balancing robot. Microsyst Technol 2018;24(9):3677–87. https://doi.org/10.1007/s00542-018-3825-2.
[6] Cerezo JO, Morales EC, Plaza JMC. Control system in open-source FPGA for a self-balancing robot. Electronics 2019;8(2): 1–19. https://doi.org/10.3390/electronics8020198.
[7] S NMV, Srimani SG, Reddy GAK. Design and development of self balancing robot. Int J Res Eng IT Soc Sci 2019;9(2): 20–32.
[8] Molnár J, Gans Š, Slavko O. Design and implementation self-balancing robot. 2020. https://doi.org/10.1109/PAEP49887.2020.9240815.
[9] Aldhalemi AA, Chlaihawi AA, Al-Ghanimi A. Design and implementation of a remotely controlled two-wheel self-balancing robot. IOP Conf Ser Mater Sci Eng 2021; 1067(1):012132. https://doi.org/10.1088/1757-899x/1067/1/012132.
[10] Jiménez L FR, Ruge R IA, Jiménez L AF. Modeling and control of a two wheeled auto- balancing robot: a didactic platform for control engineering education. 2020. https://doi.org/10.18687/LACCEI2020.1.1.556.
[11] Rohilla Y, Shekhawat AS. Design and control of two-wheeled self-balancing robot using Arduino. In: Proceedings - international conference on smart electronics and communication, ICOSEC 2020; 2020, no. October. p. 1025–30. https://doi.org/10.1109/ICOSEC49089.2020.9215421.
[12] Rahman M, Anam MR, Adib AR. Self-balancing robot with PID controller. 2021 [Online]. Available: https://www.researchgate.net/publication/369833840.
[13] Nghia VBV, Van Thien T, Son NN, Long MT. Adaptive neural sliding mode control for two wheel self balancing robot. Int J Dyn Control 2022;10(3):771–84. https://doi.org/10.1007/s40435-021-00832-1.
[14] Isdaryani F, Salam R, Feriyonika. Design and implementation of two-wheeled robot control using MRAC. J Telecommun Electron Comput Eng 2021;13(3):25–30 [Online]. Available: https://api.semanticscholar.org/CorpusID:240148974.
[15] Hassan M, Moustafa AM, Moness M. Modeling and parameters estimation of a self-balancing two-wheeled vehicle. J Adv Eng Trends 2024;43(1):373–81. https://doi.org/10.21608/jaet.2022.144785.1197.
[16] Do Q, Tran V, Ngo M, Tran M, Thiem Q, Nguyen H. Modeling and optimal control for two-wheeled self-balancing robot. J Fuzzy Syst Control 2024;2(1):22–8. https://doi.org/10.59247/jfsc.v2i1.162.