Periodicals of Engineering and Natural Sciences ISSN 2303-4521
Vol. 9, No. 3, July 2021, pp.409-417 *Original Research*

Design and implementation control system for a self-balancing robot based on internet of things by using Arduino microcontroller

[1]Ehsan Qahtan Ahmed, [2]Ibtisam A. Aljazaery, [3]Azhar F. Al-zubidi, and [4]Haider TH. Salim ALRikabi
[1,3] Department of Computer Science, AL Nahrain University, Baghdad, Iraq
[2]Department of Electrical Engineering, College of Engineering, University of Babylon, Babylon, Iraq
[4]Department of Electrical Engineering, College of Engineering, Wasit University, Wasit, Iraq

ABSTRACT

This project is designed for attempting on developing an autonomous self-balancing robot. In this work, the two-wheel robotic system consists of a microcontroller (Arduino), Dc motor, and sensor. The Arduino is used to read the sensor data and gives the order of the motor based on the control algorithm to remaine the system is stable at different impediment. The robot is drive with Dc motor and the Arduino cannot drive. A motor driver (L298 type) is used to provide a sufficient current. The Ultrasonic sensor (used to sense impediment during the movement) and 3-axis gyroscope accelerometer sensor (To measure the robot inclination angle) to control the two-wheel robot. The controller laws allow reaching static or moving targets based on three structured IOT interactions between the elementary controllers and the sensor with actuator via Cloud environment. Regarding the technical detail must be designed based on the mathematical model. The mathematical model is used based on the model of some references, after that, the transfer function of the system is found. In this work, the MATLAB Simulink is used in the design of the controller, and the PID controller is used due to the simplicity and good activity in central systems. The PID tuner package Simulink is used to obtain the controller parameter (kp, ki, kd) that gives fast and good system response and stability. The result of the designed controller shows that the system has remained stable (remained vertically) and very fast (less than 1sec) until the system reaches the desired output.

Keywords: IOT, Microcontroller (Arduino), MATLAB Simulink, PID, self-balancing robot.

Corresponding Author:

Ehsan Qahtan Ahmed
Department of Computer Science, Al Nahrain University
Baghdad, Iraq
ehsan.qahtan@nahrainuniv.edu.iq

1. Introduction

The modern technology has created a new reality in computer and communication in which networked computing systems in many forms are gaining popularity all the time. pervading our environment and attempting to improve all areas of our daily activities systems and services that go beyond the standard (human-to machine). The Internet of Things (IoT) is a term used to describe thise interconnected devices [1]. In other wise the process of controlling the balance of a portable two-wheeled robot is one of the most important topics that have been popular in the scientific community and the field of computer engineering in particular [2, 3]. The purpose of studying such robots is the possibility of finding the ideal mathematics model and its typical characteristics that contribute to its structure. Some previous studies used the same principle of traceability, the so-called, automatic Wheelchair [4-7]. So, the Segway became the most popular two-wheeled robot. With different uses, it has such as in / out movement, industrial automation, and defense the application became the personal transportation method in use at present. The Segway Personal Transporter stands for highly self-used robots drafting vehicles available on the market. The methods for their use were established by various controllers. Silva et al, 2015 by means of the state model and linear quadratic regulator (LQR) joint with proportional and integration and as a derivative control device (PID) succeeded in building two wheels of stabilization robot [8-14]. This type of self-balancing robot has a comparable application to the inverted pendulum that is taught in academic universities.

The study used these robots extensively in search of a "perfect" and "fully well-matched" algorithm. Several attempts used classic, linear, multivariate algorithms, nonlinear backward controls, and fuzzy variable controls, then groupings of these methods [15, 16]. There are major reasons that this research has enabled it to be found in academia, with the potential to increase the affordability of commercial costs outside the bandwidth of microprocessor sensors and panels (COTS). With the advent of JOE, the digital signal processor board and control panel, using microprocessors like 68HC11 and ARM, became a complementary series to build the ATmega from ATmel Engineering [17, 18].The main factor in establishing robots in recent years. One such prototype is Arduino, an open-platform prototype based on AT mega-style C-like processors, and by connecting them to the software development environment, we have made it possible to connect them with various COTS sensors [19, 20]. Contribute to an effective and fast role for the transformation into a common stage for teaching and transformational creation, with applications extending from robotics to manipulation and control, to be interconnected network control. In this study, an academic project was conducted on how to design, build and control a two-wheel self-balancing robot. We ran the robot using two DC motors, equipped with an Arduino Mega board with ATmega2560 processor and installed a uniaxial gyroscope and a two-axis accelerometer on the board to determine the position [9, 21]. COTS sensors compensated for gyroscopic drifts, to implement the problem of having a single axis in such a balance for the robot using a supplemental filter, and to simplify the complementary filtering approach from the Kalman filter, two of the control designs with linear motion equations were used for this purpose, namely the project: PID). The differential proportional integral control element based on a design-based linear quadratic regulator (LQR). Where he found it is the strength of the modeling errors that can occur during the tentative determining of these electrical and kinetic parameters as moments of inertia and kinetic gain. Various IoT interactions try to overcome these difficulties, while standardization projects try to ensure interoperability by deploying the suggested methods widely and in an organized manner. This article highlights three major classes for schematic representation of the two-wheel balancing robot control systems, which are motivated by the above issues and efforts [1, 7, 22, 23].

2. Structure of the two-wheel balancing

This section defines all the components that would allow the robot to perform the actions required for different tasks, self-balancing robotic anatomy can be built with sensors, actuators, and actuators control, and a development and upgrade board. Figure (1) shows how the sensors are applied and utilized in the design of the robot's balance, and how these sensors are used to obtain measurements of acceleration, distance traveled, and angle of inclination of the robot.

Figure 1. It shows the organizational structure of the robot

2.1. PID

One of the frequently employed controllers is the proportional-integral-derivative control system, that represents a control unit if an error occurs because it does not need to be given the typical functions and equations required for its control. Difference between the present fault and previous (component derivative) is considered based on (distance to target) along with instantaneous error (proportional component), and past errors (integral component). It is feasible to evaluate each term throughout tuning from the proportional, integral, and derivative control unit parameters of the gain Kp, Ki, and Kd respectively. Figure (2) demonstrations a block graph of the continuous-time PID control closed-loop configuration

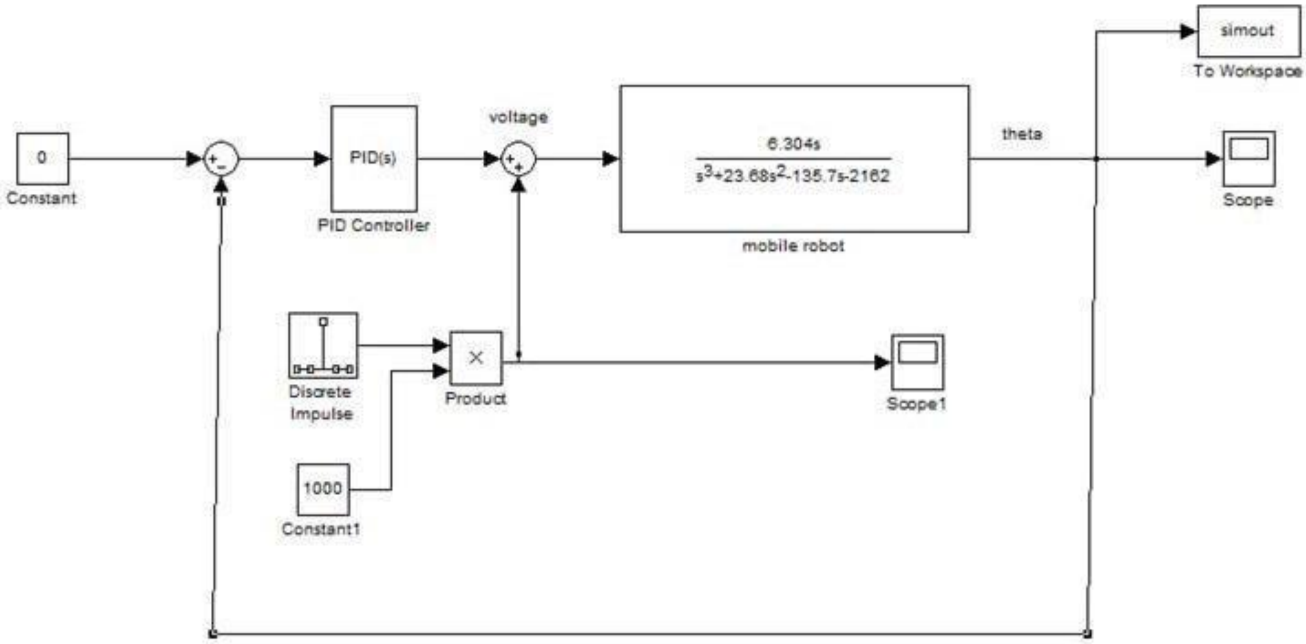

Figure 2. PID control closed-loop configuration

2.2. Position control

Attempting for overcoming a complexity of the project model, and with the search for different alternatives, some solutions emerged resulting from the dynamic tuning of a controller reference, by means of linear place and rapidity as input and creating compensation as an output (this displacement is then added to the controller source). Then following the experimental error method, the study dealt with conducting many experiments, including by means of only linear positioning, which made a steadier performance because it was unable to stimulate the response to high velocity. Later it Swas used velocity to reduce the angular error and when mixing both approaches. It was concluded there were no typical or formal studies to follow such a study because it depends only on experimental work. Figure (3) shows the method has only two requirements, first: a controller, a kind of dimension (current robot angle as input, PWM magnitude as output and required angle as reference), second: a location sensor, such as an encoder.

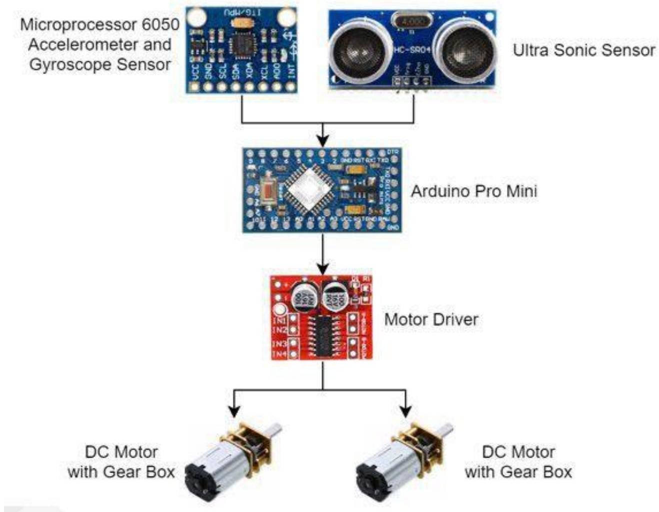

Figure 3. Design hardware requirements

3. IOT control system discribtion

For the schematic representation of robot control systems, there are three main types. We proposed the following three versions for continuing study on the usage of a two-wheel self-balancing robot for the aim of identifying, monitoring, and intervening in danger areas: (a) direct or manual control; (b) mixed control; (c) supervised control. These sensors send and receive signals from the controller via Cloud environment.
Direct or manual control: The robot has no artificial intelligence or autonomy, and the operator controls the robot's movement directly without the assistance of an automated system; the principle of operation is very simple and is based on four bits (forward-backward and left-right), in which case signal identification is a priority, as the microcontroller timer is used as an internal counter so that it can count the number of times the robot has moved.
Mixed control: The robot has some artificial intelligence and/or autonomy, and the operator assists it in performing certain tasks. When the robot switches to autonomous mode, the system of orientation sensors (gyroscope) and distance measurement sensors (odometer) makes predictions about the next path to take. The longer the distance between the operator and the robot, the more difficult teleoperation becomes, especially due to the visual impression on the operator caused by the robot's motions.
Supervised control: The robot is intelligent and/or autonomous, with the user only intervening at a high decision level. Control architectures are built to allow for the application of a single control rule in each context: the path to the objective, obstacle avoidance, and so on. Multi-agent systems, which may self-organize based on the development of phenomena, are used to manage the interactions amongst controllers depending on the situation [24, 25]. Figure (4) presents the three types of remote-control systems for robots connecting with the sensor via Cloud environment.

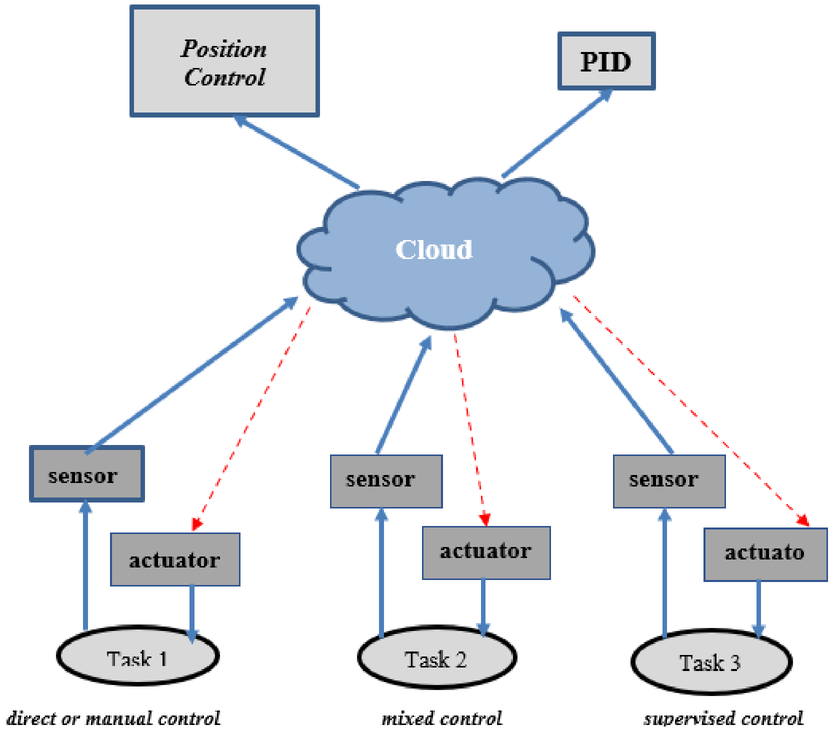

Figure 4. Interactions among the basic controllers

4. The proposed work

In this section, the mechanics, the software, and procedures of building a bimbo are all explained.

4.1. Physical properties

Figure (5) shows the physical makeup of the Bimbo, Lego-based robot, with particular engineering and durability. As a chassis contains four main units: (1) a lower part that holds the motors (2) the 2nd unit for storing electronic materials (sensors, Arduino, H-bridge, among others); (Iii) carrying the 3rd battery; And (4) the latter, at the topmost, in which a user can set things to test balance or employ them to mount more electronics. Bimbo consists of a 4,800 mAh battery, and the Arduino Pro Mini, 5V / 16MHz, Pololu L298 Dual H-Bridge, the intervention robot's control architecture is built on the dual Pololu 30: 1 Metal, 12v 500rpm gear motor with 64 CPR encoder motor for autonomous navigation., Pololu 6, DOF IMU-MPU6050 sensor, HC-06 Bluetooth module, 3 voltmeter number display to indicate battery voltage level. The whole amassed system is depicted in Figure(5). Regarding the physical dimensions, the Bimbo is 15.8 x 16.1 x 7.1 cm, it weighs 1062 g and has two wheels that are 9 cm in diameter.

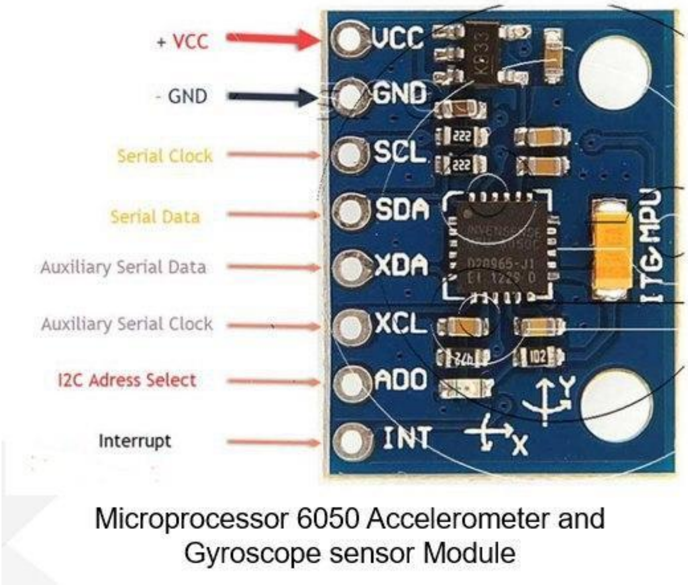

Figure 5. A fully assembled bimbo robot.

A number of mainly motor control and reading standard libraries along with different sensors are existing. It is a sort of "software laboratory" design. That allow control units with the necessary details to be tested. The three types of control system robot moves are simple to use and flexible enough.

4.2. Experiments: The design includes four libraries:

Engine: Multiple engines are allowed to be declared; May each one. It has different variables (such as the lowest stimulus value; or Extreme stimulus value - must be adjusted to stop the case of Battery off). Figure (6) shows some functions for controlling Motors (rotation and direction).

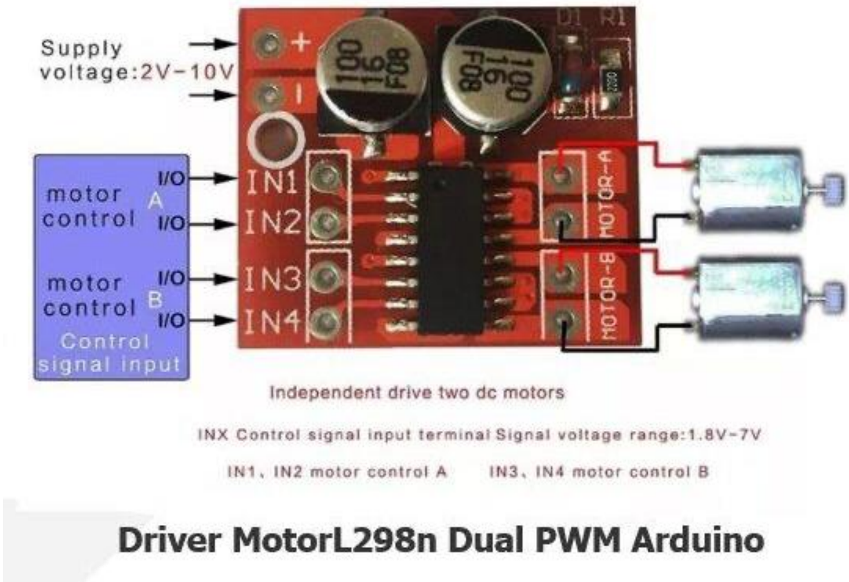

Figure 6. Driver motor, PWM Arduino

Encoder: The motors possess a 30: 1 gearbox, with 64 CPR encoders, namely single shaft rotation must go up 30 x 64 = 1920 events. For Arduino interrupts based on the Arduino library, it's intolerable to deal with it. Nevertheless, this project is ok and counting is not necessary. This allows the standard to be used Arduino functions, 320 clicks per Wheel rotation as in Figure (7).

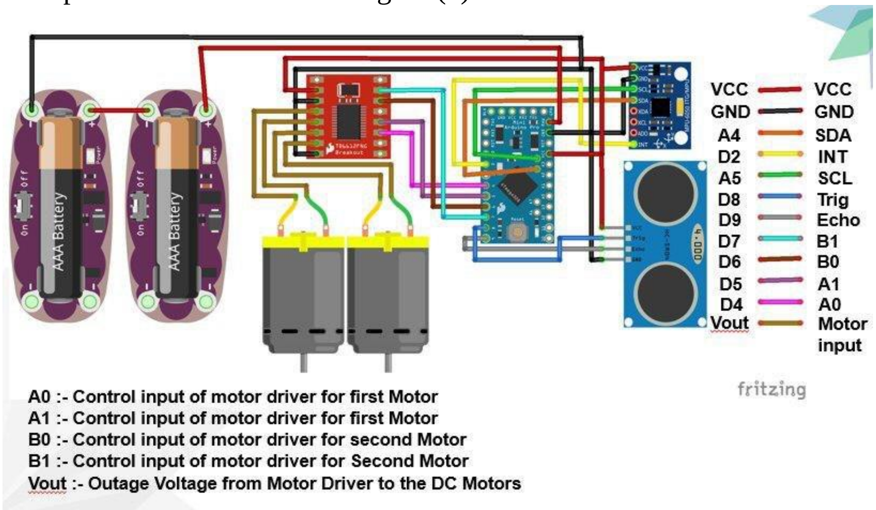

Figure 7. Full hardware design

MPU 6050: It is employed for communicating with acceleration sensor and gyroscope. This sensor is very noisy and requires a filter. Each balancing filter and Kalman work well. This is kind of app, but since Kalman has a response, it was considered as MPU uses this library 6050. The carrying out of this filter is based on Christian Lauszus solution[26]. Figure (8) shows suitably suited controller to Bimbo.

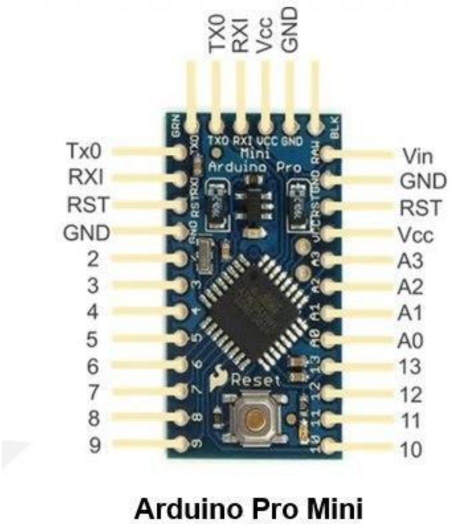

Figure 8. Arduino pro mini

Each library was tested individually after the library was implemented. A mechanism was established with Matlab as a surveillance tool and Bluetooth interface for in-time variables to read robotic variables. On the part of the MATLAB, a time/signal variation graph was developed to continuously display. This algorithm is limited by the same axis scale, but allows for a separate input scale feature as several variables the user wants to see. All library companies have been tested to show the desired answer, but three limitations have been identified. In non-linear fashion, the first motor turns sooner than the other. It was modified by reading the values of the encoder and forcing an engine to move slower. The second, the battery voltage varies for the different PWM values. This problem is because the output of ordinary controllers is supposed to be linear in composition 2 (where Vin is the battery voltage and Vout is the voltage used for the motors) and Vin is supposed to be constant. Through the display we could observe that various PWM values force Vin to change. Moreover, the battery status changes the response, making the creation of a linearization process more difficult. The above-mentioned "Position Control" flowchart resolved this problem.

4.3. Main algorithm (flowchart)

Figure (9) represents the flowchart explaining a main applied procedure.

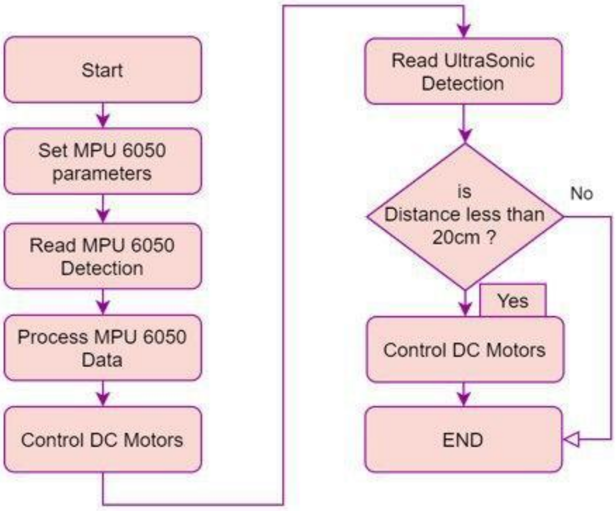

Figure 9. Flowchart of the developed system

Navigation functions were implemented to ensure Bimbo stability: (i) moving forward; (ii) moving back, (iii) turning right, (iv) turning left and (v) retaining the immobile unit without any of the above commands. Functionally, the target angle controls operations I and (ii). Given that the output is "absolute," the addition of symmetric values before the final values are transmitted to the motors will force rotation without losing balance.

5. Results and discussion

The main two objectives, balancing a two-wheel robost and finding a simple, multi-infrastructure controller, were considered to have been achieved through this project. This new approach can be employed for controlling different types of robots. Bimbo's stability has been tested to cause disturbances in the form of outside forces and to change the mass of the robot (carrying cellphones, student cases, and toolboxes). The time it takes Bimbo to reach the initial position is associated with the parameters of the "position control." Gains that lead to a quicker recovery time, however, will lead to more oscillation. Below (10,11), show the real position of Bimbo. Figures below. Figure (10) shows the Bimbo response time when stable and at the starting point, where the time response of the disturbance is induced and is removed from the initial position in Figure (11).

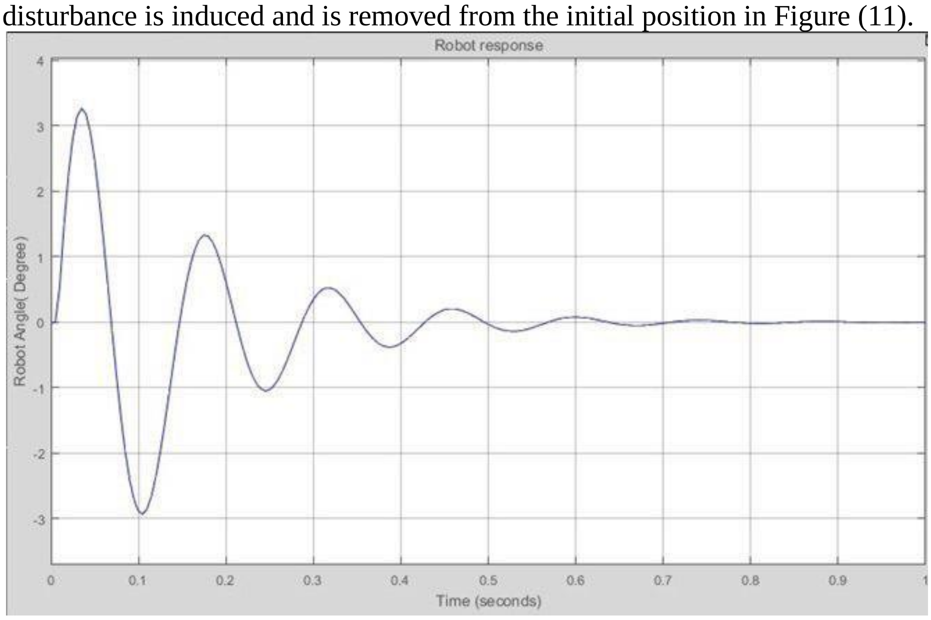

Figure 10. Time response of Bimbo where is stable and at initial position

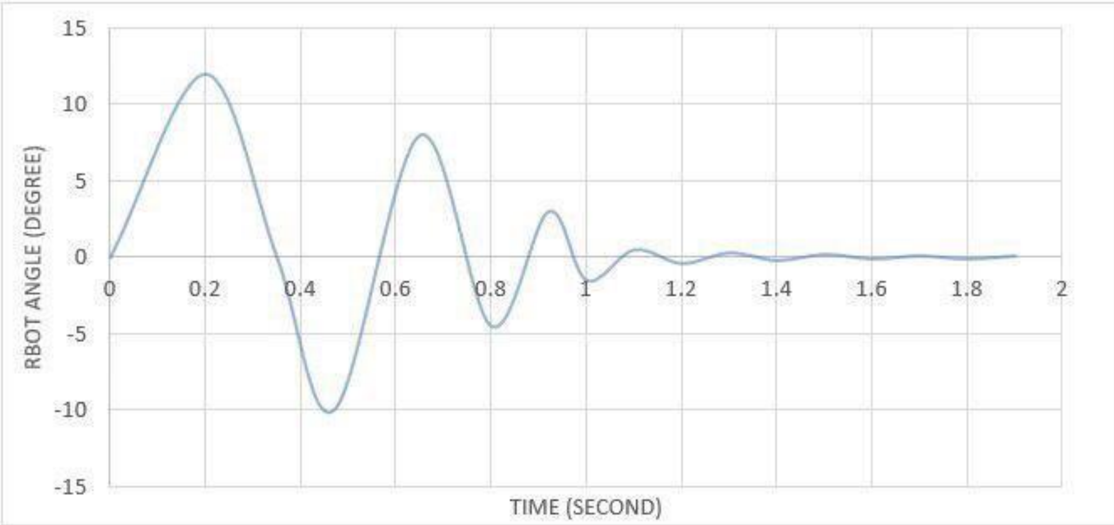

Figure 11. Time response of Bimbo where is unstable

6. Conclusions

A simple self-equilibrating robot solution has been developed as a key objective of this project. We have built a two-wheel self-equilibrating robot with low-cost components and we have implemented a tilt-angle calculator and a PID stabilizer for balance movement. The controller rules allow for the reach of static or moving targets based on three organized IOT interactions between the sensor/actuator and the primary controls using the cloud environment. They must be designed using a mathematical model with regard to technical details. A dynamic reference adjustment process ("position control") was used for obtaining the final solution. In addition, the PID reference can also be controlled dynamically by the robot movement. The solution described is general and easy to use for various robot types. However, this hypothesis needs additional testing, which involves experimental work with various robots. The "position control" techniques, as stated earlier, require some adjustment. These were tested and optimized in this project.

References

[1] K. Fysarakis, I. Askoxylakis, O. Soultatos, I. Papaefstathiou, C. Manifavas, and V. Katos, "Which IoT protocol? Comparing standardized approaches over a common M2M application," in 2016 IEEE Global Communications Conference (GLOBECOM), 2016, pp. 1-7: IEEE.

[2] R. S. Martins and F. Nunes, "Control system for a self-balancing robot," in 2017 4th Experiment@ International Conference (exp. at'17), 2017, pp. 297-302: IEEE.

[3] A. Zdešar, S. Blažic, and G. J. I.-P. Klančar, "Engineering education in wheeled mobile robotics," IFAC-PapersOnLine, vol. 50, no. 1, pp. 12173-12178, 2017.

[4] A. M. Chheda, R. K. Mandava, and P. undavilli, "Design and development of two-wheeled self-balancing robot and its controller," International Journal of Mechatronics and Automation, vol. 8, no. 1, pp. 1-8, 2021.

[5] A. Alaidi, I. Aljazaery, H. Alrikabi, I. Mahmood, and F. Abed, "Design and Implementation of a Smart Traffic Light Management System Controlled Wirelessly by Arduino," international Journal of Interactive Mobile Technologies, vol. 14, no. 7, pp. 32-40, 2020.

[6] N. F. AL-Bakri, A. F. Al-zubidi, A. B. Alnajjar, and E. Qahtan, "Multi label restaurant classification using support vector machine," Periodicals of Engineering and Natural Sciences (PEN),vol. 9, no. 2, pp. 774-783, 2021.

[7] A. Ouannas and I. Aljazaery, "A new method to generate a discrete chaotic dynamical systems using synchronization technique," Far East J. Dyn. Syst, vol. 24, no. 1-2, pp. 15-24, 2014.

[8] V. C. A. Silva, P. Leite, F. Soares, G. Lopes, J. S. Esteves, and P. Garrido, "Controlling an equilibrist Lego robot," 2015.
[9] W. K. Meteab, H. T. Salim, S. A. H. Al Sultani, and I. A. Aljazaery, "Controlling and Monitoring a Robot-Car Based on Smart Phone Applications," in IOP Conference Series: Materials Science and Engineering, 2021, vol. 1094, no. 1, p. 012096: IOP Publishing.
[10] R. M. Al_airaji, I. A. Aljazaery, S. K. Al_dulaimi, and H. T. S. Alrikabi, "Generation of high dynamic range for enhancing the panorama environment," Bulletin of Electrical Engineering and Informatics, Article vol. 10, no. 1, pp. 138-147, 2021.
[11] N. F. A.-B. A. F. Al-zubidi, R. K. Hasoun, S. H. Hashim, and H. TH. S. Alrikabi "Mobile Application to Detect Covid-19 pandemic by using Classification Techniques: Proposed System," International Journal of Interactive Mobile Technologies, vol. 15, no. 16, 2021.
[12] N. F. AL-Bakri and S. Hashim, "A Study on the Accuracy of Prediction in Recommendation System Based on Similarity Measures," Baghdad Science Journal, vol. 16, no. 1 Supplement, 2019.
[13] A. F. Hassan, "Using Unlabeled Data Set for Mining Knowledge from DDB," Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI), vol. 7, no. 1, pp. 1-8, 2021.
[14] I. A Aljazaery, "General Control Method for Complete Synchronization of Two-Arbitrary 2-D Chaotic Systems in Discrete-Time," journal of kerbala university, vol. 11, no. 2, pp. 253-258, 2015.
[15] D. W. Scott, Multivariate density estimation: theory, practice, and visualization. John Wiley & Sons, 2015.
[16] O. H. Yahya, H. T. ALRikabi, R. a. M. Al_airaji, and M. Faezipour, "Using Internet of Things Application for Disposing of Solid Waste," International Journal of Interactive Mobile Technologies, vol. 14, no. 13, pp. 4-18, 2020.
[17] S. Bergbreiter and K. S. Pister, "Cotsbots: An off-the-shelf platform for distributed robotics," in Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453), 2003, vol. 2, pp. 1632-1637: IEEE.
[18] H. A. Naman, M. Al-dabag, H. Salim, "Encryption System for Hiding Information Based on Internet of Things," International Journal of Interactive Mobile Technologies (iJIM), vol. 15, no. 2, 2021.
[19] J. Clark and R. Fierro, "Mobile robotic sensors for perimeter detection and tracking," ISA transactions, vol. 46, no. 1, pp. 3-13, 2007.
[20] N. A. jassim, and H. Alrikabi, "Design and Implementation of Smart City Applications Based on the Internet of Things," iJIM, vol. 15, no. 3, 2021.
[21] N. F. AL-Bakri and S. Hassan, "A Proposed Model to Solve Cold Start Problem using Fuzzy User-Based Clustering," in 2019 2nd Scientific Conference of Computer Sciences (SCCS), 2019, pp. 121-125: IEEE.
[22] A. H. T. A.-G. Ali Abdulwahhab Mohammed, Azhar F. Al-zubidi, Harith M. Saeed, "Unsupervised Classification and Analysis of Istanbul-Turkey Satellite Image Utilizing the Remote Sensing," Second International Scientific Conference (Sisc2021), AIP Conference Proceedings,2021.
[23] A. F. A.-z. Teaba Wala Aldeen Khairi, Ehsan Qahtan Ahmed, "Modified Multipath Routing Protocol Applied On Ns3 Dcell Network Simulation System," International Journal of Interactive Mobile Technologies (iJIM), 2021.
[24] L. Ș. Grigore, D. Gorgoteanu, C. Molder, O. Alexa, I. Oncioiu, A. Ștefan, D. Constantin, M. Lupoae, and R.-I. J. S. Bălașa, "A dynamic motion analysis of a six-wheel ground vehicle for emergency intervention actions," Sensors, vol. 21, no. 5, p. 1618, 2021.
[25] B. Mohammed, R. Chisab, and H. Alrikabi, "Efficient RTS and CTS Mechanism Which Save Time and System Resources," international Journal of Interactive Mobile Technologies, vol. 14, no. 4, pp. 204-211, 2020.
[26] E. Eriksson, "Self-Balancing Robot Control System in CODESYS for Raspberry Pi: Design and Construction of a Self-Balancing Robot using PLC-programming tools," ed, 2016.