INTENSIF, Vol.5 No.2 August 2021
ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v5i2.15296

Semi-Adaptive Control Systems on Self-Balancing Robot using Artificial Neural Networks

Sistem Kendali Semi-Adaptif pada Robot self-balancing menggunakan jaringan saraf Tiruan

Received:
25 November 2020
Revised:
11 May 2021
Published:
8 August 2021

[1]Eko Setiawan, [2*]Dahnial Syauqy
[1,2]S2Teknik Komputer, Universitas Brawijaya
[1,2]Malang, Indonesia
E-mail: [1]ekosetiawan@ub.ac.id, [2]dahnial87@ub.ac.id
*Corresponding Author

Abstract— A self-balancing type of robot works on the principle of maintaining the balance of the load's position to remains in the center. As a consequence of this principle, the driver can go forward reverse the vehicle by leaning in a particular direction. One of the factors affecting the control model is the weight of the driver. A control system that has been designed will not be able to balance the system if the driver using the vehicle exceeds or less than the predetermined weight value. The main objective of the study is to develop a semi-adaptive control system by implementing an Artificial Neural Network (ANN) algorithm that can estimate the driver's weight and use this information to reset the gain used in the control system. The experimental results show that the Artificial Neural Network can be used to estimate the weight of the driver's body by using 50-ms-duration of tilt sensor data to categorize into three defined classes that have been set. The ANN algorithm provides a high accuracy given by the results of the confusion matrix and the precision calculations, which show 99%.

Keyword— semi-adaptive control system, self-balancing robot, artificial neural network

Abstrak— *Robot self balancing bekerja dengan prinsip menjaga keseimbangan posisi beban agar tetap berada di tengah. Sebagai konsekuensi dari prinsip ini, pengemudi dapat maju-mundur dengan bersandar ke arah tertentu. Salah satu faktor yang mempengaruhi model kendali adalah bobot pengemudi.. Sistem kendali yang telah dirancang tidak akan dapat menyeimbangkan sistem jika pengemudi yang menggunakan kendaraan melebihi atau kurang dari nilai bobot yang telah ditentukan. Tujuan utama penelitian ini adalah mengembangkan sistem kendali semi-adaptif dengan mengimplementasikan algoritma Jaringan saraf Tiruan yang dapat memperkirakan bobot pengemudi dan menggunakan informasi tersebut untuk mengatur ulang penguatan yang digunakan pada sistem kendali. Dari hasil percobaan didapatkan bahwa Jaringan Saraf Tiruan dapat digunakan untuk mengestimasi bobot badan pengemudi dengan menggunakan data sensor kemiringan berdurasi 50 ms untuk dikategorikan menjadi 3 kelas yang telah ditetapkan. Algoritma ANN memberikan akurasi yang cukup tinggi yang ditunjukkan dengan hasil confusion matrix dan ketepatan perhitungan yang menunjukkan 99%.*

Kata Kunci— *sistem kendali semi-adaptif, robot self-balancing, jaringan saraf tiruan*

ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v5i2.15296

I. INTRODUCTION

These days, many electric vehicles replace fuel vehicles sold in the market [1]. Personal electric vehicles are generally used to transport one driver without other passengers. These personal vehicles include self-balancing vehicles, electric scooters, electric bicycles, electric wheelchairs, and electric skateboards. A survey conducted by Shin shows that self-balancing vehicles are more commonly used than other personal vehicles in indoor, outdoor, or semi-outdoor environments [2].

Segway Inc. first introduced Self-balancing personal vehicles at the end of 2001, named Segway personal transporter (PT). It went on public sale in 2002 and was promoted as a new method of transportation [3]. When it was introduced to the market, Segway PT was able to sell around 80,000 units which were used for the tour, personal, police, military, and emergency handling units [4]. A self-balancing vehicle has two wheels placed on one axis, and each of them is powered by a battery-powered electric motor. This vehicle can balance itself and keep the driver's position on it so as not to fall. This capability is known as self-balancing. Forward control is examined by tilting the driver's position forward to balance by following the driver's movements. Turning is done by tilting the steering lever to the right or left, which causes the vehicle to change the speed between the two motors, resulting in a turning motion.

The most important part of a self-balancing vehicle is a control system that keeps the vehicle's wheels to be able to follow the driver's body tilt. The first control system used in self-balancing vehicles is a linear control system including PID [5], [6], and LQR [7], [8]. PID control system stands for Proportional-Integral-Derivative, which provides a control loop feedback mechanism to control the process and variation of output. The Linear-Quadratic-Regulator (LQR) control is also a feedback controller in which the engineer is required to specify cost function parameters and eventually compare the result and the design. The linear control system was often chosen because it is simple. Based on the simulation results, the LQR control system can return the vehicle to its initial condition in less than 2 seconds when a disturbance is given [9]. Linear control system requires linearization of the nonlinear model, which means that the linear control system only works well at a limited working point.

Several nonlinear control systems have also been proposed to overcome the limitations of this linear system. A fuzzy control system, a nonlinear control system, can be used to maintain the balance of the vehicle [10], [11]. According to the simulation findings, the fuzzy control system can restore vehicle equilibrium in less than ten seconds [11]. Even the combined Fuzzy and PID simulated using Simulink showed that the error could be reduced to 60% [12]. Several variants of the sliding-mode control system are also used to maintain balance. Hierarchical sliding-mode is the best type of sliding mode which can return the system to a stable state in about 5 seconds [13].

The control system that has been previously proposed can control well, but the determination of the control parameters uses the assumption of fixed vehicle and driver weights. It is not following the actual conditions because the weight of the driver may vary so that a control system needs to be reset. The weight and location of the driver's center-of-mass will affect the gain value in the control system [14]. Drivers who are not too tall and light will make it easier for self-balancing vehicles to remain stable.

This study proposes a semi-adaptive mechanism in the vehicle control system that can change the control system parameters based on the driver's weight. The linear control system will be chosen as the primary control system because of the ability of the control system to respond quickly to changes. The driver's weight will be estimated using Artificial Neural Network (ANN) on changes in driver tilt angle. The weight information aimed to change the gain in the control system.

II. RESEARCH METHOD

This study proposes a semi-adaptive method to the control system as described in Figure 1. The initial step that will be carried out is to construct a mathematical model of the self-balancing vehicle. After the mathematical model is verified, the next step is to arrange a filter to reduce noise on the sensor [15]. The PID control system will be applied to the system after ensuring that the sensor reading noise can be adequately suppressed. The artificial neural network (ANN) will then be designed and arranged to determine the optimal configuration in estimating the user's weight. The entire system will be combined according to Figure 1 and tested to find out how effective the method is in solving problems in personal self-balancing vehicles.

The mathematical equation will design a control system that keeps the driver upright on the vehicle and does not fall. The mathematical equation of the vehicle is derived using the laws of physics.

ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v5i2.15296

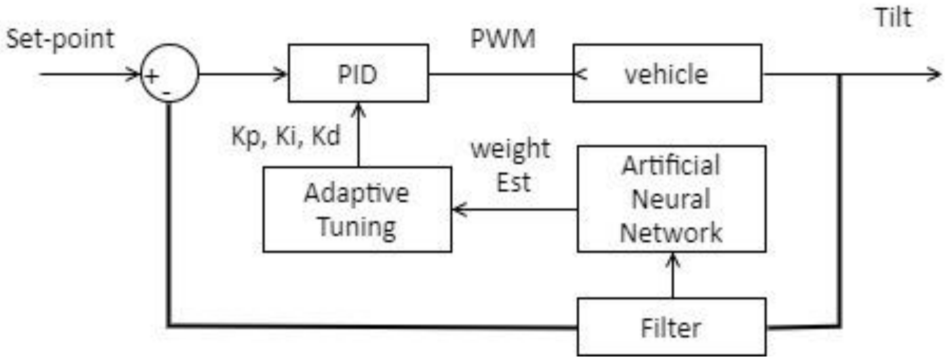
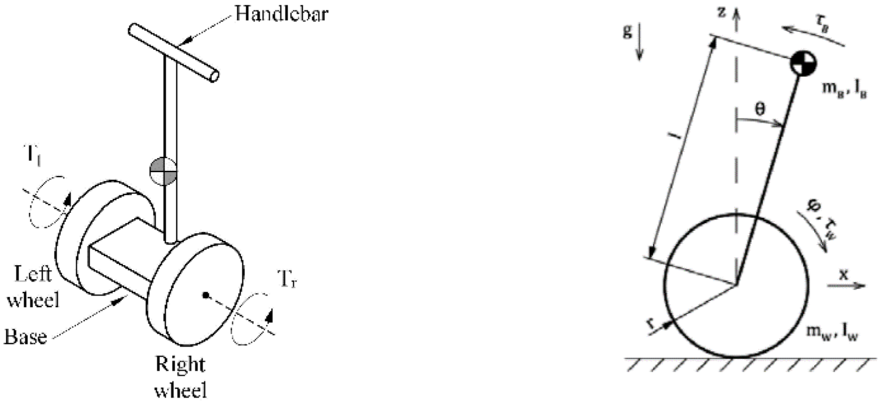


Figure 1. CONTROL SYSTEM BLOCK DIAGRAM



(a) (b)

Figure 2 (A) SELF-BALANCING VEHICLE SKETCH [14] (B) SIDE VIEW OF VEHICLE SKETCH [16]

Figure 2 (a) is a sketch of a personal self-balancing vehicle. Vehicles can be divided into two parts that are connected. The top is the driver with the footing and steering. The bottom is a wheel that rotates freely and is permanently attached to the ground. The wheels of self-balancing vehicles must always touch the ground to ensure the vehicle can be controlled. Any movement that keeps one of the wheels off the ground will result in system instability and the driver falling. One technique can change the wheel axle mechanism [17] or change the driver's position [18].

The mathematical equation that applies to vehicles is derived by assuming that the wheels are permanently attached to the ground or no wheel slip. The friction between the wheels and the ground is neglected. The sketch of the vehicle, when viewed from the side, is shown in Figure 2(b). The right direction represents the front direction of the vehicle in the sketch. A pendulum represents the center of gravity with mass mB and inertia IB located at a distance l with the axle. Based on the sketch, the top of the vehicle can lean forward and backward freely. The degree of skew is denoted by θ. The following equation can calculate the position of the upper center of gravity.

$$x_B = x + l\sin\theta$$
$$z_B = l\cos\theta \qquad (1)$$

The x symbol is the deviation of the vehicle from the starting position. The movement speed of the upper vB can be calculated using the following equation.

$$v_B = \sqrt{\dot{x}_B{}^2 + \dot{z}_B{}^2} = \sqrt{\left(\dot{x} + l\dot{\theta}\cos\theta\right)^2 + \left(-l\dot{\theta}\sin\theta\right)^2} = \sqrt{\dot{x}^2 + 2\dot{x}l\dot{\theta}\cos\theta + l^2\dot{\theta}^2(\cos^2\theta + \sin^2\theta)} \qquad (2)$$

The mathematical equation of the vehicle can be derived by observing the principle of the law of conservation of energy, namely kinetic energy and potential energy. The kinetic energy of the top of the EKB vehicle consists of translational kinetic energy and rotational kinetic energy. The kinetic energy of the top of the vehicle, which includes the driver, footing, and lever, can be formulated in the following equation.

$$E_{KB} = \frac{1}{2}m_B v_B{}^2 + \frac{1}{2}I_B\dot{\theta}^2$$
$$= \frac{1}{2}m_B\left(\dot{x}^2 + 2\dot{x}l\dot{\theta}\cos\theta + l^2\dot{\theta}^2\right) + \frac{1}{2}\left(\frac{1}{3}m_B l^2\right)\dot{\theta}^2$$
$$= \frac{1}{2}m_B\dot{x}^2 + m_B\dot{x}l\dot{\theta}\cos\theta + \frac{4}{6}m_B l^2\dot{\theta}^2 \qquad (3)$$

The IB's upper inertia can be compared to a homogeneous cylinder or rod rotating at one end. Therefore, rotational kinetic energy and translational kinetic energy make up the kinetic energy of the bottom or both wheels. The E_{KW}'s lower kinetic energy may be expressed as follows:

$$E_{KW} = \frac{1}{2}m_W\dot{x}^2 + \frac{1}{2}I_W\dot{\varphi}^2$$
$$= \frac{1}{2}m_W\dot{x}^2 + \frac{1}{2}\left(\frac{1}{2}m_W r^2\right)\left(\frac{\dot{x}}{r}\right)^2$$
$$= \frac{3}{4}m_W\dot{x}^2 \qquad (4)$$

The mass of the two wheels is denoted by mW with the radius of the wheel r. the wheel's inertia resembles the inertia of a cylinder rotating in the center. The angle of the wheel φ can be converted into a translational form using x and r. The potential energy produced by the top of the vehicle can be calculated using the following equation.

$$E_P = m_B g l\cos\theta \qquad (5)$$

The mathematical model of the equations of motion of a vehicle can be found using Lagrange's rule.

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right) - \frac{\partial L}{\partial q} = Q \qquad (6)$$

L denotes the Lagrange equation. The coordinate system of the Lagrange q will use the position x and the angle of inclination θ. The force acting on the system is denoted by Q. The Lagrange equation is derived by using all the kinetic energy minus the working potential energy so that the equation becomes as follows.

$$L = E_{KB} + 2E_{KW} - E_P$$

$$= \frac{1}{2}m_B\dot{x}^2 + m_B\dot{x}l\dot{\theta}\cos\theta + \frac{4}{6}m_B l^2\dot{\theta}^2 + \frac{3}{2}m_W\dot{x}^2 - m_B g l\cos\theta$$

$$= \left(\frac{m_B}{2} + \frac{3m_W}{2}\right)\dot{x}^2 + m_B\dot{x}l\dot{\theta}\cos\theta + \frac{4}{6}m_B l^2\dot{\theta}^2 - m_B g l\cos\theta \qquad (7)$$

Personal vehicles have two types of forces that work, namely translational force and rotational force. Both forces come from the torque τ generated by the right-motorbike and the left-motorbike. The translational force is converted from torque using wheel radius r as follows.

$$F = \frac{\tau_R + \tau_L}{r} \qquad (8)$$

The x position used as the q coordinate system will produce the following equation for motion.

$$\frac{d}{dt}\left[(m_B + 3m_W)\dot{x} + m_B l\dot{\theta}\cos\theta\right] - 0 = F$$

$$(m_B + 3m_W)\ddot{x} + m_B l\ddot{\theta}\cos\theta - m_B l\dot{\theta}^2\sin\theta = \frac{\tau_R + \tau_L}{r} \qquad (9)$$

The second coordinate used to find the equation of motion is the angle of inclination θ. The resulting motion equation is then shown as follows.

$$\frac{d}{dt}\left[m_B\dot{x}l\cos\theta + \frac{4}{6}m_B l^2 2\dot{\theta}\right] - m_B\dot{x}l\dot{\theta}(-\sin\theta) - m_B g l(-\sin\theta) = \tau_R + \tau_L$$

$$\left[m_B\ddot{x}l\cos\theta + m_B\dot{x}l(-\sin\theta)\dot{\theta} + \frac{4}{3}m_B l^2\ddot{\theta}\right] + m_B\dot{x}l\dot{\theta}\sin\theta + m_B g l\sin\theta = \tau_R + \tau_L$$

$$m_B\ddot{x}l\cos\theta - m_B\dot{x}l\dot{\theta}\sin\theta + \frac{4}{3}m_B l^2\ddot{\theta} + m_B\dot{x}l\dot{\theta}\sin\theta + m_B g l\sin\theta = \tau_R + \tau_L$$

$$m_B\ddot{x}l\cos\theta + \frac{4}{3}m_B l^2\ddot{\theta} + m_B g l\sin\theta = \tau_R + \tau_L \qquad (10)$$

Based on these two coordinates, the equation of motion for personal self-balancing vehicles can be written as follows.

$$(m_B + 3m_W)\ddot{x} + m_B l\ddot{\theta}\cos\theta - m_B l\dot{\theta}^2\sin\theta = \frac{\tau_R + \tau_L}{r}$$

$$m_B\ddot{x}l\cos\theta + \frac{4}{3}m_B l^2\ddot{\theta} + m_B gl\sin\theta = \tau_R + \tau_L \qquad (11)$$

The torque of a motor τ is determined by the motor voltage v so that the torque of the two motors can be assumed as follows.

$$\tau_R + \tau_L = 2\tau = 2Kv_M \qquad (12)$$

The value of K is a constant between the voltage and the resulting torque. By substituting the torque value, the equation of motion of a personal self-balancing vehicle can be written as follows.

$$(m_B + 3m_W)\ddot{x} + m_B l\ddot{\theta}\cos\theta - m_B l\dot{\theta}^2\sin\theta = \frac{2Kv_M}{r}$$
$$m_B\ddot{x}l\cos\theta + \frac{4}{3}m_B l^2\ddot{\theta} + m_B gl\sin\theta = 2Kv_M \qquad (13)$$

A. State Space

The equation of motion produced in the previous chapter is nonlinear. This equation can be approximated by linearization at the equilibrium point θ = 0. The Taylor series will be used to find the linear equation of the sin and cos functions to produce the following equation.

$$\cos\theta \approx 1$$
$$\sin\theta \approx \theta$$
$$\dot{\theta}^2 \approx 0 \qquad (14)$$

By substituting linearization for the equations of vehicle motion, the linear equation at the equilibrium point can be written as follows.

$$(m_B + 3m_W)\ddot{x} + m_B l\ddot{\theta} = \frac{2Kv_M}{r}$$
$$m_B\ddot{x}l + \frac{4}{3}m_B l^2\ddot{\theta} + m_B gl\theta = 2Kv_M \qquad (15)$$

The overall mathematical model of this vehicle can be represented in a matrix form as follows.

$$\begin{bmatrix} m_B + 3m_W & m_B l \\ m_B l & \frac{4}{3}m_B l^2 \end{bmatrix}\begin{bmatrix}\ddot{x}\\ \ddot{\theta}\end{bmatrix} + \begin{bmatrix}0 & 0\\ 0 & 0\end{bmatrix}\begin{bmatrix}\dot{x}\\ \dot{\theta}\end{bmatrix} + \begin{bmatrix}0 & 0\\ 0 & m_B gl\end{bmatrix}\begin{bmatrix}x\\ \theta\end{bmatrix} = \begin{bmatrix}\frac{2K}{r}\\ 2K\end{bmatrix}v_M$$
$$\boldsymbol{E}\begin{bmatrix}\ddot{x}\\ \ddot{\theta}\end{bmatrix} + \boldsymbol{F}\begin{bmatrix}\dot{x}\\ \dot{\theta}\end{bmatrix} + \boldsymbol{G}\begin{bmatrix}x\\ \theta\end{bmatrix} = \boldsymbol{H}v_M$$
$$\begin{bmatrix}\ddot{x}\\ \ddot{\theta}\end{bmatrix} = -\boldsymbol{E}^{-1}\boldsymbol{F}\begin{bmatrix}\dot{x}\\ \dot{\theta}\end{bmatrix} - \boldsymbol{E}^{-1}\boldsymbol{G}\begin{bmatrix}x\\ \theta\end{bmatrix} + \boldsymbol{E}^{-1}\boldsymbol{H}v_M \qquad (16)$$

The inverse of the matrix E is as follows.

ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v5i2.15296

$$\boldsymbol{E}^{-1} = \begin{bmatrix} \frac{4}{m_B + 12m_W} & \frac{-3}{m_B l + 12m_W l} \\ \frac{-3}{m_B l + 12m_W l} & \frac{3(m_B + 3m_W)}{m_B l^2 (m_B + 12m_W)} \end{bmatrix} \qquad (17)$$

The mathematical model of personal vehicles can be represented in state-space through matrix manipulation to produce the following matrix form.

$$\frac{d}{dt}\begin{bmatrix} x \\ \theta \\ \dot{x} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -\boldsymbol{E}^{-1}\boldsymbol{G} & & -\boldsymbol{E}^{-1}\boldsymbol{F} & \end{bmatrix}\begin{bmatrix} x \\ \theta \\ \dot{x} \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \boldsymbol{E}^{-1}\boldsymbol{H} \end{bmatrix} v_M \qquad (18)$$

The state-space form of the personal self-balancing vehicle system is represented in the following matrix equation.

$$\frac{d}{dt}\boldsymbol{X} = \boldsymbol{AX} + \boldsymbol{BU}$$

$$\frac{d}{dt}\begin{bmatrix} x \\ \theta \\ \dot{x} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{3m_B g}{m_B + 12m_W} & 0 & 0 \\ 0 & \frac{-3g(m_B + 3m_W)}{(m_B + 12m_W)l} & 0 & 0 \end{bmatrix}\begin{bmatrix} x \\ \theta \\ \dot{x} \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{8Kl - 6Kr}{m_B rl + 12m_W rl} \\ \frac{-6K(m_B l - m_B r - 3m_W r)}{m_B(m_B + 12m_W)rl^2} \end{bmatrix} v_M \qquad (19)$$

Matrix X is a state. The U matrix is the input from the system. The Y output of a vehicle is the tilt angle, one of the system states. So, the state-space equation for the output is as follows.

$$\boldsymbol{Y} = \boldsymbol{CX} + \boldsymbol{DU}$$

$$[\theta] = \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix}\begin{bmatrix} x \\ \theta \\ \dot{x} \\ \dot{\theta} \end{bmatrix} + [0]v_M \qquad (20)$$

B. Transfer Function of the System

The simulation and design of control systems require a transfer function of a system. The transfer function can be derived from the state-space equation that was formed in the previous chapter. The derivation is done by finding a solution of state X to the input U through the following Laplace transform.

$$\begin{aligned} s\boldsymbol{X}(s) &= \boldsymbol{AX}(s) + \boldsymbol{BU}(s) \\ [s\boldsymbol{I} - \boldsymbol{A}]\boldsymbol{X}(s) &= \boldsymbol{BU}(s) \\ \boldsymbol{X}(s) &= [s\boldsymbol{I} - \boldsymbol{A}]^{-1}\boldsymbol{BU}(s) \end{aligned} \qquad (21)$$

Based on the solution of state X, the Y output can be calculated using the following equation.

$$\boldsymbol{Y}(s) = \boldsymbol{C}\boldsymbol{X}(s) + \boldsymbol{D}\boldsymbol{U}(s)$$

$$\boldsymbol{Y}(s) = \boldsymbol{C}\{[s\boldsymbol{I} - \boldsymbol{A}]^{-1}\boldsymbol{B}\boldsymbol{U}(s)\} + \boldsymbol{D}\boldsymbol{U}(s)$$

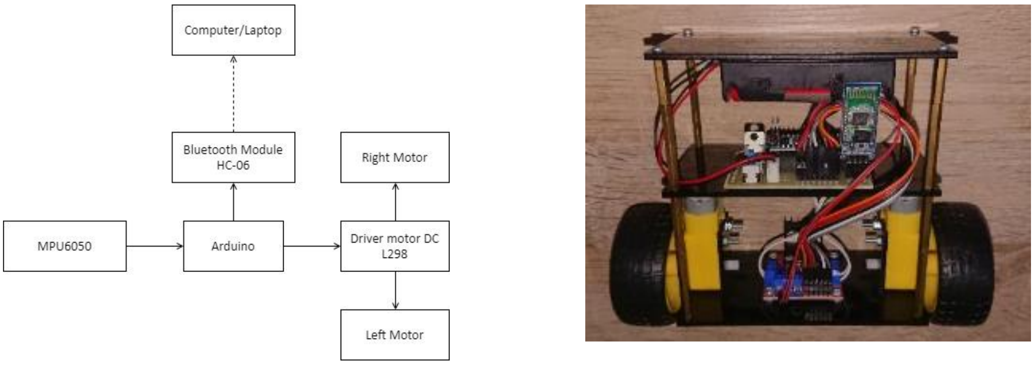
$$Y(s) = \frac{6K\big((m_B + 3m_W)r - m_B l\big)}{m_B r l(3g(m_B + 3m_W) + (m_B + 12m_W)ls^2)} \tag{22}$$

C. Hardware Design

Figure 3 illustrates the block diagram of the self-balancing robot. The MPU6050 accelerometer sensor and gyroscope will be used as input from the system to read the vehicle's tilt. The readings from the sensor will be forwarded to the microcontroller to be processed using the PID control system and the Artificial Neural Network adaptive system. The output from the PID control system will then be forwarded to the L298 DC motor driver. The motor driver will control the right and left motors of the vehicle at the same speed. The HC-06 Bluetooth module is also embedded in the vehicle prototype for data retrieval. Data retrieval is carried out wirelessly so as not to affect system performance in balancing the vehicle. The Bluetooth module will send tilt data to the computer continuously. The data received by the computer will then be used in analyzing system performance.

The prototype implementation of the personal self-balancing vehicle is shown in Figure 3(b). It uses three 18650 batteries connected in series as the primary power source. The battery is placed into the motor driver located at the bottom, and the microcontroller, sensors, and Bluetooth module are located at the center of the vehicle.

This study used additional weights as a simulation for riders who have different weights. The load will be placed on the vehicle when the robot is running. There are three types of vehicle load scenarios that will be used in this study. The three types of vehicle loads were examined to analyze the effect of the load on the vehicle's response.



(a) (b)

Figure 3. (A) BLOCK DIAGRAM OF A PERSONAL SELF-BALANCING VEHICLE (B) PERSONAL SELF-BALANCING VEHICLE PROTOTYPE

D. PID Tuning

The PID parameters applied in the system need to be tuned first to keep the vehicle in an upright position. The first step, taken before the tuning process, is to determine the vehicle's parameters that affect the transfer function as listed in the previous section. Parameter measurement is done by weighing each component and measuring the radius of the wheel and vehicle height, as shown in Figure 4. The complete measurement results are shown in Table 1.

Table 1. VEHICLE PARAMETERS

No.	Parameter	Symbol	Value
1	Wheel weight	m_W	36.5 gr
2	Vehicle and load weight	w_B	1) 406 gr 2) 406 + 95 = 501 gr 3) 406 + 187 = 593 gr
3	Gravity	g	9.8 m/s^2
4	Vehicle height	l	12.5 cm
5	Wheel radius	r	3.25 cm

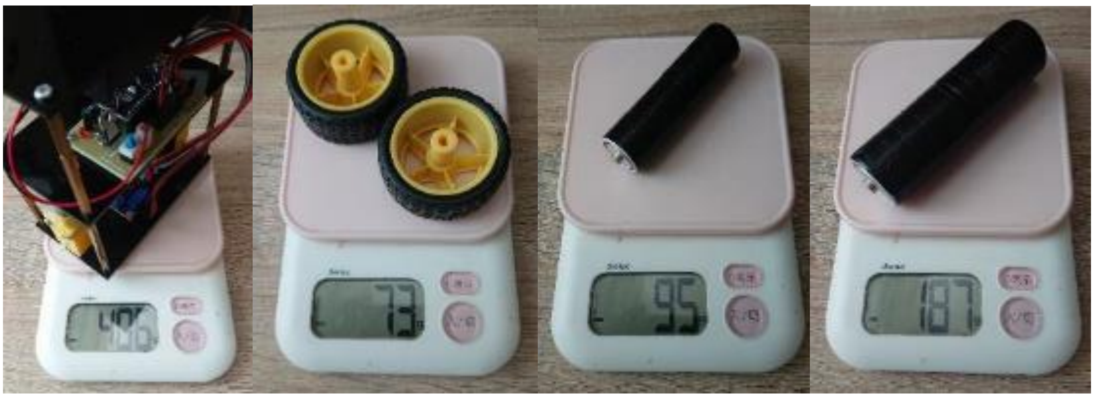


Figure 4. MEASUREMENT OF PERSONAL SELF-BALANCING VEHICLE PARAMETERS

The tuning technique increases the Kp value gradually and gives the Ki and Kd values to 0. The Kp value is increased so that the vehicle can move against the leaning direction of the vehicle by quick-moving so that it can change the leaning direction of the vehicle. When the vehicle can reply against the leaning direction quickly, the Kp value can compensate for the speed of slope changes.

After the Kp value can oppose the direction of the vehicle's slope, the Kd value is increased to reduce the response speed of the Kp parameter. The value of Kd continues to be increased so that the vehicle does not fall but continues to move along the slope. After the vehicle continues to move along the slope while continuing to try to make the vehicle position upright, the addition of the Kd value is stopped.

The next step is to add Kp and Kd values simultaneously so that the vehicle can continue moving back and forth to balance the position. The addition of these two parameters will probably result in oscillations between the vertical positions. When the vehicle can oscillate to maintain an upright position, the addition of Kp and Kd is stopped.

The final step requires a large reduction in oscillating motion to keep the vehicle in an upright position. This reduction in oscillation is made by increasing the value of Ki until the oscillation, or alternating movement is only around the upright position. The vehicle does not turn too far. This tuning process is carried out for the three types of loads, namely "load 1", "load 2," and "load 3".

Artificial Neural Network

This section describes how to implement ANN in classifying the signal shape of the accelerometer sensor readings on the self-balancing robot to determine the appropriate control parameters. In general, in the process of implementing ANN (Artificial Neural Network), there are three sub-processes, namely a) class b) sensor signal acquisition and training data formulation, c) training process and algorithm evaluation

1. Class

There are three classes used in this system, namely

- No-load (load 1)
- With battery load C (load 2)
- With battery load D (load 3)

The general working principle of the system is to use the PID parameter for "no-load" regardless of the type of load when it is on the first run. Over time, the accelerometer sensor will acquire the value of the robot's balance stability. The correct PID parameter for the suitable class will make the robot stable in its balanced pose. However, the mismatch of the PID parameter will cause the robot to tend to be less stable in balancing itself. Thus, it is necessary for the three classes to each have their respective PID parameters tuned and tested that these parameters are stable enough for each class.

2. Sensor signal acquisition and training data formulation

After determining the correct class and PID parameters for each class, the accelerometer sensor data (tilt) was collected from the robot to get how stable the robot is in its balance pose. As already explained, the robot initially used the "no-load" PID parameter. Then, the robot was conditioned in no-load, battery load C; and battery load D. Furthermore, the robot was run for each condition. The sensor was acquired and then sent to the programming computer via Bluetooth for data logging for a certain period.

ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v5i2.15296

The shape of the signal shows that the no-load condition, with the PID parameter at no-load, shows a signal that tends to be stable, indicating that the robot tends to be stable in balancing itself. Furthermore, it is necessary to crop the signal data so that the training data is balanced and focused during self-balancing conditions. In this case, each signal would be truncated for nearly 900 milliseconds, at which point the robot has started trying to balance itself. Figure 5 shows the results for cutting nearly 900 milliseconds in each condition.

In the cropping process, a total of 888 value reading data were generated for each class. Furthermore, to be able to predict class, it is not enough just to input one value. Therefore, 50 data sequences would be used as classification input. The 50 data represent the reading of the sensor value for 50 ms. It is considered sufficient as a characteristic of each condition. Thus, 50 data framing/windowing would be carried out on a total of 888 data for each class. The results of the framing extraction are used as training data for each class. The illustration of the 50 data framing running on a total of 888 data is shown in Figure 6.

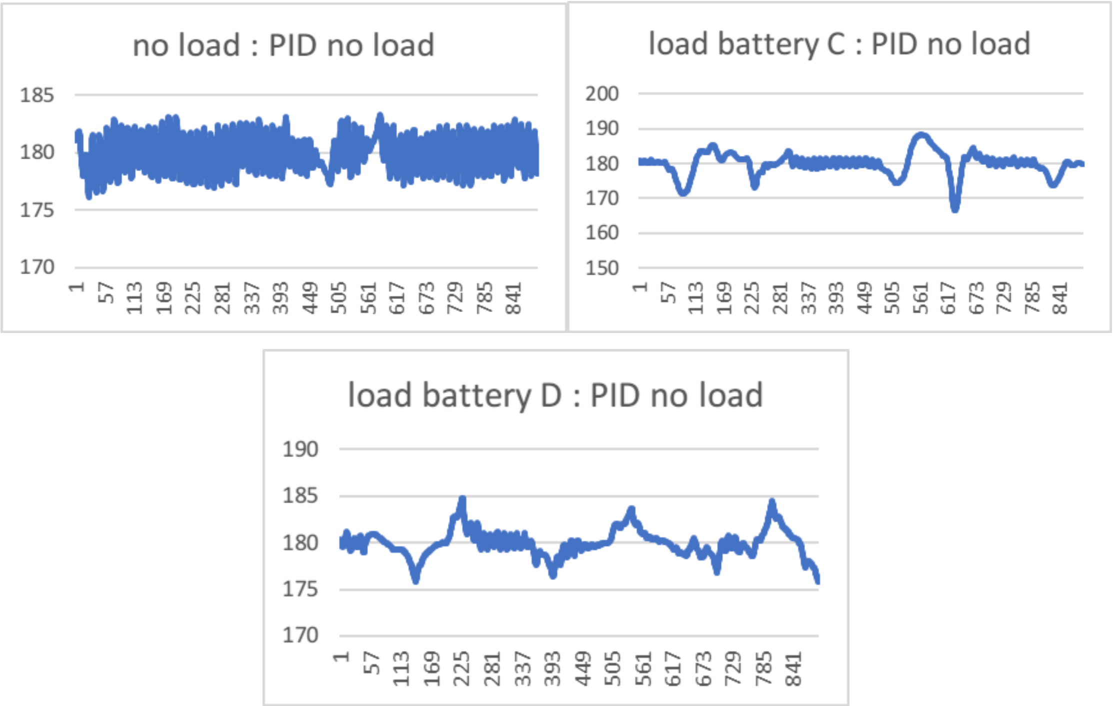


Figure 5. APPROXIMATELY 900 MILLISECONDS OF FILTERED DATA WHEN BALANCED (SELF-BALANCING)

From the framing process of each class, a total of 839 rows of data were obtained for each class, which means that there are a total of 2517 data for all classes. Of this number, 20% will be taken as test data to evaluate the performance of ANN classification.

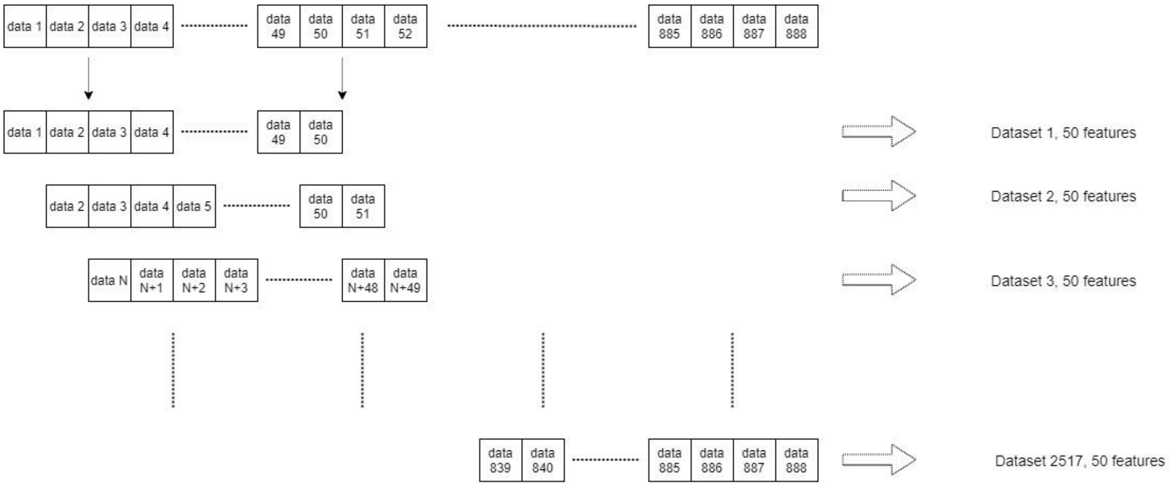


Figure 6. THE PROCESS OF FRAMING 50 DATA RUNNING ON A TOTAL OF 888 DATA AS A DATASET

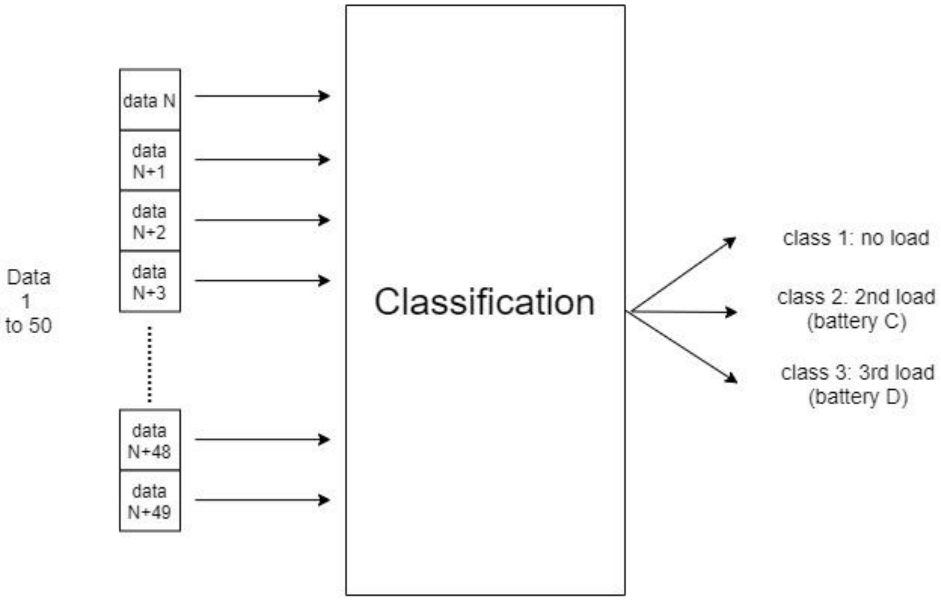
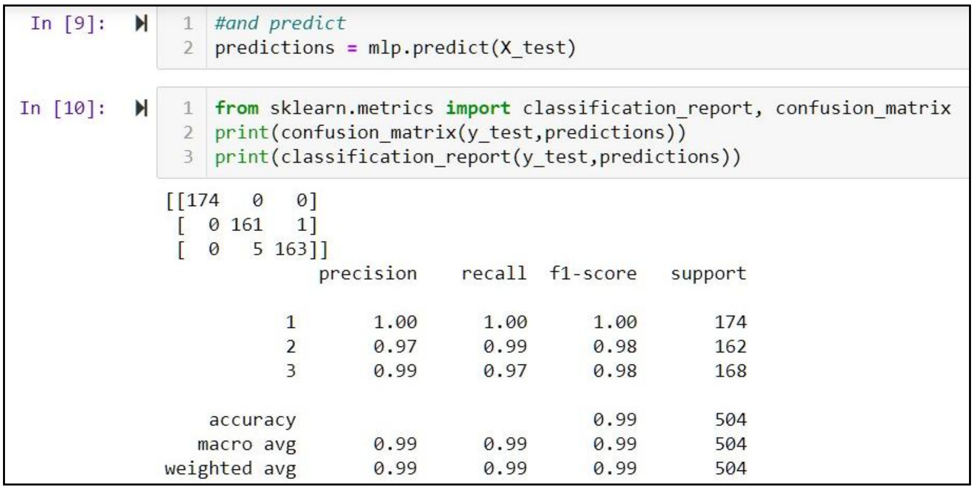


Figure 7. ILLUSTRATION OF FEATURES AND CLASSES FOR CLASSIFICATION

3. Training process and evaluation

The Scikit-learn tool [19] used the Python programming language through the Jupyter Notebook to train the ANN algorithm [20]. First, the data extracted from the sensor signal readings (2517 lines of data) was converted into a CSV file. Next, the process of preparing the dataset into training data and test data is carried out. In this case, 20% (504 data) of the data will be used as test data, the remaining 80% (2013 data) was used as training data. Furthermore, before performing machine learning algorithms, the standardization process is first carried out using the Scikit library, namely the standard scaler. The algorithm does not work correctly if the dataset feature does not have a standard normal distribution. If a feature has much more significant variance, there is dominance so that the algorithm cannot learn as expected.



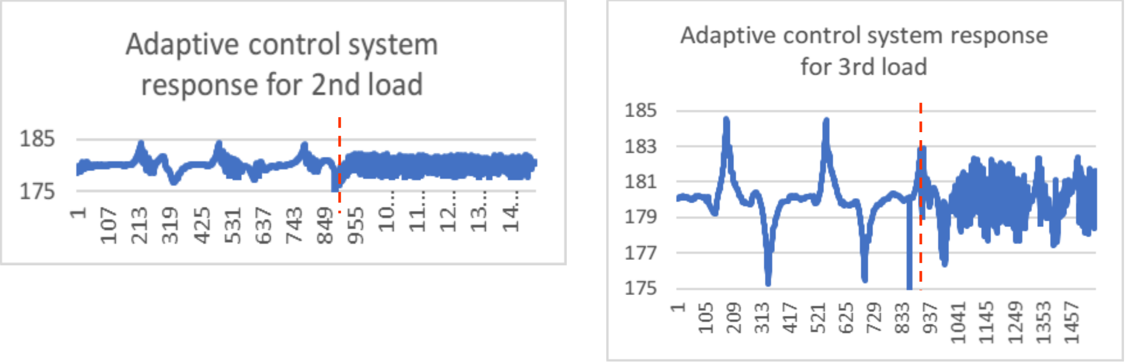
```
In [9]:   #and predict
          predictions = mlp.predict(X_test)

In [10]:  from sklearn.metrics import classification_report, confusion_matrix
          print(confusion_matrix(y_test,predictions))
          print(classification_report(y_test,predictions))

[[174   0   0]
 [  0 161   1]
 [  0   5 163]]
              precision    recall  f1-score   support

           1       1.00      1.00      1.00       174
           2       0.97      0.99      0.98       162
           3       0.99      0.97      0.98       168

    accuracy                           0.99       504
   macro avg       0.99      0.99      0.99       504
weighted avg       0.99      0.99      0.99       504
```

Figure 8. THE RESULT OF ALGORITHM EVALUATION IS CONFUSION MATRIX AND ITS ACCURACY

Next is the implementation of Artificial Neural Networks using MLPclassifier. The parameters used are the default Scikit parameter, using the ReLU activation function (rectified linear unit function, f (x) = max (0, x)), the learning rate is constant with a value of 0.001, 3000 iterations, with three hidden layers 50 each. A built-in tool, a classification report, and a confusion matrix aimed to evaluate the performance of the algorithms that have been applied. The confusion matrix and precision indicate three calculation results. Load 1 (no load) has 0 misclassifications, load 2 (battery C) has one misclassification, and load 3 (battery D) has only five misclassifications.

III. RESULT AND DISCUSSION

The purpose of this experimental test is to determine the system response to load changes, whether it can adaptively change the PID parameters according to the installed load. Since the robot was run, the robot would be set to balance itself with the PID parameter at no-load for approximately 850 ms, which would then apply the classification from 50 ms data so that at 900 ms, the correct PID parameter is used according to the load. The result of plotting the signal for "load 2" is shown in Figure 9(a). From the image, after 900 ms, the robot had started to stabilize in balancing itself and did not oscillate like before. It can be seen from the oscillation value, which was in the range of 178o-182o. Besides, when the parameters are incorrect, the oscillations were in the range of 175o-184o with a massive swing.



(a) (b)

Figure 9. (A) SEMI-ADAPTIVE CONTROL SYSTEM RESPONSE AT THE SECOND LOAD (B) SEMI-ADAPTIVE CONTROL SYSTEM RESPONSE AT THE THIRD LOAD

The result of plotting the signal for "load 3" is shown in Figure 9(b). From the image, after 900 ms, the robot has started to stabilize in balancing itself and does not oscillate like before. It shows that the oscillation value is in the range of 178^{o} -182^{o}, while when the parameters are not correct, the oscillations are in the range of 175^{o} -185^{o} with a giant swing.

Based on both figures, each of them was marked using vertical, red-dotted lines; the system can directly change the control parameters. The oscillation of the balancing robot can be reduced drastically depending on the load provided. Even though in the final state of the tilt signal, it still rapidly oscillated, but the degree of the swing had been significantly reduced. It implies that there is still space for development in terms of lowering the minimum fast oscillation. In addition, it also is interesting if the system can fully adaptively tune the control system based on the weight directly, which eliminates the need for classification of load's weight.

IV. CONCLUSION

This research aims to develop a method to semi-adaptively tune and control a self-balancing robot using an Artificial Neural Network to estimate the type of current load on the robot. The experiment result shows that the Artificial Neural Network can estimate the weight of the driver's body by using the value of the tilt sensor in the form of a sequence for a duration of 50 ms as the input parameter to classify the classes that had been set. However, we limit only three weight classes to be used as prototypes to show the semi-adaptive concept. According to the class, the classification results from the Artificial Neural Network can be used as a determining indicator to select the correct PID parameters. Of course, it is required that prior proper tuning for each class is done so that the control system responds as expected.

According to the paper, the confusion matrix and precision of the ANN algorithm confirm high accuracy (99 percent), with 0 misclassifications of load 1 (no load), one misclassification of load 2 (battery C), and only five misclassifications of load 3 (battery D). The final test also shows that the system response is also good in switching the PID parameters indicated by the tilt signal on the self-balancing robot.

Currently, there are only three predefined classes of robot load. In the future, it is suggested to add more categories for robot load. If possible, tune the PID parameters directly from their weights, eliminating the need for class division. In addition, the ANN parameter is set by default, so related research can compare performance by changing ANN parameters. It is recommended to implement it in the actual self-balancing vehicle with a human driver so that the performance can be accurate in actual conditions.

ACKNOWLEDGMENTS

The authors would like to thank the Faculty of Computer Science, Universitas Brawijaya, for providing financial support for this research in the DIPA 2020 scheme.

REFERENCES

[1] J. Y. Yong, V. K. Ramachandaramurthy, K. M. Tan, and N. Mithulananthan, "A review on the state-of-the-art technologies of electric vehicle, its impacts, and prospects," Renewable and Sustainable Energy Reviews. 2015, DOI: 10.1016/j.rser.2015.04.130.

[2] G. W. Shin, K. J. Lee, D. Park, J. H. Lee, and M. H. Yuri, "Personal mobility device and user experience: A state-of-the-art literature review," 2018, DOI: 10.1177/1541931218621305.

[3] A. Pourmand, J. Liao, J. M. Pines, and M. Mazer-Amirshahi, "Segway® Personal Transporter-Related Injuries: A Systematic Literature Review and Implications for Acute and Emergency Care," J. Emerg. Med., vol. 54, no. 5, 2018, DOI: 10.1016/j.jemermed.2017.12.019.

[4] K. Boniface, M. P. McKay, R. Lucas, A. Shaffer, and N. Sikka, "Serious injuries related to the Segway® personal transporter: A case series," Ann. Emerg. Med., 2011, doi: 10.1016/j.annemergmed.2010.06.551.

[5] S. C. & V. V. Shubhank Sondhia, Ranjith Pillai. R, Sharat S. Hegde, "Development of Self Balancing Robot With PID Control," Int. J. Robot. Res. Dev. (IJRRD, 2017.

[6] R. S. Martins and F. Nunes, "Control system for a self-balancing robot," 2017, DOI: 10.1109/EXPAT.2017.7984388.

[7] S. Trimpe, A. Millane, S. Doessegger, and R. D'Andrea, "A self-tuning LQR approach demonstrated on an inverted pendulum," 2014, DOI: 10.3182/20140824-6-za-1003.01455.

[8] Y. Gong, X. Wu, and H. Ma, "Research on Control Strategy of Two-Wheeled Self-Balancing Robot," 2016, DOI: 10.1109/CSMA.2015.63.

[9] R. Babazadeh, A. G. Khiabani, and H. Azmi, “Optimal control of Segway personal transporter,” 2016, doi: 10.1109/ICCIAutom.2016.7483129.

[10] J. Huang, M. Ri, D. Wu, and S. Ri, "Interval type-2 fuzzy logic modeling and control of a mobile two-wheeled inverted pendulum," IEEE Trans. Fuzzy Syst., 2018, DOI:

10.1109/TFUZZ.2017.2760283.

[11] C. H. Huang, W. J. Wang, and C. H. Chiu, "Design and implementation of fuzzy control on a two-wheel inverted pendulum," IEEE Trans. Ind. Electron., 2011, DOI: 10.1109/TIE.2010.2069076.

[12] A. S. Wardoyo, S. Hendi, D. Sebayang, I. Hidayat, and A. Adriansyah, "An investigation on the application of fuzzy and PID algorithm in the two-wheeled robot with the self-balancing system using microcontroller," 2015, DOI: 10.1109/ICCAR.2015.7166003.

[13] L. Chen et al., "Robust hierarchical sliding mode control of a two-wheeled self-balancing vehicle using perturbation estimation," Mech. Syst. Signal Process., 2020, DOI: 10.1016/j.ymssp.2019.106584.

[14] A. Castro, C. Adams, and W. Singhose, "Dynamic response characteristics of a two-wheeled inverted-pendulum transporter," 2013, DOI: 10.1109/CDC.2013.6760100.

[15] G. Bobby, E. Susanto, and F. Y. Suratman, “Implementasi Robot Keseimbangan Beroda Dua Berbasis Mikrokontroler,” ELKOMIKA J. Tek. Energi Elektr. Tek. Telekomun. Tek. Elektron., vol. 3, no. 2, p. 142, 2015, doi: 10.26760/elkomika.v3i2.142.

[16] P. Frankovský, L. Dominik, A. Gmiterko, I. Virgala, P. Kurylo, and O. Perminova, "Modeling of two-wheeled self-balancing robot driven by DC gearmotors," Int. J. Appl. Mech. Eng., 2017, DOI: 10.1515/ijame-2017-0046.

[17] S. Kwon, S. Kim, and J. Yu, "Tilting-type balancing mobile robot platform for enhancing lateral stability," IEEE/ASME Trans. Mechatronics, 2015, DOI: 10.1109/TMECH.2014.2364204.

[18] F. Dai, J. Li, J. Peng, Z. Zhu, S. Jiang, and X. Gao, "Design and Control of a Multi-DOF Two-Wheeled Inverted Pendulum Robot," 2014, DOI: 10.1109/WCICA.2014.7052763.

[19] G. Varoquaux, L. Buitinck, G. Louppe, O. Grisel, F. Pedregosa, and A. Mueller, “Scikit-learn,” GetMobile Mob. Comput. Commun., 2015, DOI: 10.1145/2786984.2786995.

[20] F. P'erez and B. E. Granger, "Jupyter Notebook Documentation," Comput. Sci. Eng., 2018.