Available online at www.ijournalse.org

Emerging Science Journal

Vol. 4, Special Issue "IoT, IoV, and Blockchain", (2020, 2021)

Using a Combination of PID Control and Kalman Filter to Design of IoT-based Telepresence Self-balancing Robots during COVID-19 Pandemic

Iswanto Iswanto [1*], Alfian Ma'arif [2], Nia Maharani Raharja [3], Tony Khristanto Hariadi [1], Muhammad Abdus Shomad [1]

[1] *Universitas Muhammadiyah Yogyakarta, Yogyakarta 55183, Indonesia*

[2] *Universitas Ahmad Dahlan, Yogyakarta 55191, Indonesia*

[3] *Universitas Islam Negeri Sunan Kalijaga, Yogyakarta 55281, Indonesia*

Abstract

COVID-19 is a very dangerous respiratory disease that can spread quickly through the air. Doctors, nurses, and medical personnel need protective clothing and are very careful in treating COVID-19 patients to avoid getting infected with the COVID-19 virus. Hence, a medical telepresence robot, which resembles a humanoid robot, is necessary to treat COVID-19 patients. The proposed self-balancing COVID-19 medical telepresence robot is a medical robot that handles COVID-19 patients, which resembles a stand-alone humanoid soccer robot with two wheels that can maneuver freely in hospital hallways. The proposed robot design has some control problems; it requires steady body positioning and is subjected to disturbance. A control method that functions to find the stability value such that the system response can reach the set-point is required to control the robot's stability and repel disturbances; this is known as disturbance rejection control. This study aimed to control the robot using a combination of Proportional-Integral-Derivative (PID) control and a Kalman filter. Mathematical equations were required to obtain a model of the robot's characteristics. The state-space model was derived from the self-balancing robot's mathematical equation. Since a PID control technique was used to keep the robot balanced, this state-space model was converted into a transfer function model. The second Ziegler-Nichols's rule oscillation method was used to tune the PID parameters. The values of the amplifier constants obtained were K_p=31.002, K_i=5.167, and K_d=125.992128. The robot was designed to be able to maintain its balance for more than one hour by using constant tuning, even when an external disturbance is applied to it.

Keywords:

Self-Balancing Robot;
Telepresence Robot;
Kalman Filter;
COVID-19:
Mathematical Modeling;
PID;
Zigler-Nichols.

Article History:

Received:	08	September	2021
Revised:	20	January	2022
Accepted:	03	March	2022
Published:	12	March	2022

1- Introduction

The transmission route of COVID-19 in humans mainly originates from person-to-person transmission [1, 2]. The first human-to-human transmission was known based on a medical case of the first patient in Vietnam and the US [3] who was infected while in Wuhan [4, 5] without ever having physical contact with the Seafood Wholesale Market, which was the beginning of the spread of the SARS-CoV-2 virus [6]. Since the number of COVID-19 cases emerged worldwide [7], the Centers for Disease Control and Prevention (CDC) has developed many health procedures and

* **CONTACT**: iswanto_ppi_umy@ieee.org

DOI: http://dx.doi.org/10.28991/esj-2021-SP1-016

guidelines for health professionals and the public in general [8]. Doctors, nurses, and medical personnel must wear protective clothing while treating patients due to the risk of cross-transmission [9, 10]. Publics need to minimize physical touch and direct human contact [11]. This new regulation greatly affects healthcare management systems and many other systems in general [12–14]. For example, rehabilitation robots can be used to correct weakened hand function [15, 16] so that rehabilitation is no longer entirely dependent on doctors and health workers. However, although physical contact still needs to be limited, health professionals still must be 'present' in many cases. Hence, telepresence becomes essential to minimize direct contact to reduce the transmission rate of the disease while preserving face-to-face, engaging interactions.

Telepresence robots can empower people to work, play, and learn with others, regardless of geographical distance [17]. Telepresence in robotics is a subfield of robotic-assisted routes, where humans act as operators, sending high-level instructions to the assistive robot while receiving sensory feedback [18]. The sensory feedback can be experienced in various ways. For example, Rhee et al. [19] discovered a telecollaboration that involves teleporting long-distance collaborators to another real-world environment. Similarly, Sun and Yan [20] proposed a mixed reality-based teleoperation system. Convenient Virtual Reality (VR) planning optimized human perception for telepresence was researched by Becerra et al. [21]. As telepresence continues to develop, researchers also proposed and studied many fundamental algorithms and methods to deliver these sensory feedbacks. Deep 3-D segmentation of Point Cloud was researched by Zhong et al. [22]. Spatial presence, performance, and behavior between real, distant, and virtual immersive environments were studied by Khenak et al. [23]. Meanwhile, Erat et al. [24] proposed real-time view planning of unstructured Lumigraph modeling.

In the meantime, studies and research regarding data communication stability in systems with the Internet of Things (IoT) also continued. For example, a study regarding Similarity Meter and Distance Meter for Set Neutrosophic Interval with IoT Industry Evaluation was researched by Peng [25]. A more sophisticated, reliant, and robust predictive control mechanism also has been developed for a networked teleoperation space robot [26]. The stability of data communication with Xbee S2b Zigbee used on an Arduino-based sumo robot was investigated [27]. These findings implied that a medical telepresence robot could establish stable and secure real-time data communication using one of these technologies so that doctors can engage with patients or monitor patients' health conditions more safely.

Other challenges in designing a medical telepresence robot are related to its mechanical configuration and control system. A medical telepresence robot should represent human presence to make engaging interactions possible. Its height should be an average height of a human and should be able to maneuver freely and autonomously. Moreover, it requires to be battery-operated with less energy consumption so that it can also be used for monitoring patients' conditions.

A self-balancing robot is a two-wheel robot with a working principle similar to an inverted pendulum that balances itself vertically on a horizontal surface [27]. An inverted pendulum has a center of gravity above the fulcrum [28]. The inverted pendulum's basic concept is to find and stabilize the stick's position above the robot [29]. This robot configuration design is similar to human feet. In addition, two-wheeled robots consume less power with stability-enhancing behavior than four-wheeled robots. Also, they have better maneuverability since using two wheels reduces the turn radius to zero.

Some researchers used the inverted pendulum balancing method for stabilizing other robots. For example, the combined sliding mode control method used for swing-up control and stabilization was investigated by Park and Chwa [30]. The virtual linear inverted pendulum model for bipedal propulsion planning was investigated by Motoi [31]. An efficient 3-D three-legged walker stabilization and directional control based on PDAC was proposed by Aoyama [32]. This robot configuration requires a strong and stable control system to keep a balance and maintain its position and condition without outside assistance when it becomes subject to disturbance [33]. Meanwhile, several methods have been used to maintain the robot's balance. For example, Sliding Mode Control (SMC) has been used as a nonlinear controller [34–36]. It can also be combined with a high-level observer for better control performance [37]. A synchronized stabilization platform was also proposed to control the inverted pendulum [38]. Another nonlinear controller proposed for controlling inverted pendulum was Feedback Linearization [39, 40]. Predictive control techniques such as Model Predictive Control (MPC) [41] were also reported. Another filtering method was also used, such as Kalman filter [42, 43]. LQR-based controllers were also made [44–48]. Moreover, intelligent control techniques were also proposed: Neural Network [49–52], and Fuzzy [53–55].

However, those aforementioned methods are unsuitable for disturbance rejection control in medical telepresence robots since they require complex calculations that consume more power and memory. Another popular controller in the control system is the proportional-integral-derived (PID) control [56, 57]. Simplicity, durability, and near-optimal performance are the reasons why PID control is so popular in the industry [58]. It is simple, functionally easy to understand, and practical [59, 60]. Since it only requires simple calculations, PID will consume less power and memory, making it the most suitable controller for medical telepresence robots.

PID algorithm requires parameter setting to get performance with minimum error, thus, its tuning process must be properly done to achieve the correct parameters [61, 62]. The parameters to be obtained from PID control system are K_p, K_i, and K_d [63, 64]. Most researchers used sophisticated PID tuning methods to set the K_p, K_i, and K_d configuration values, such as the Fuzzy logic algorithm [65, 66]. Others even combined it with the Kidney-Inspired Algorithm [67] and PSO [68], making it more complex. However, the Ziegler-Nichols method is one of the standard tuning methods used to set the K_p, K_i, and K_d configuration values. There are two methods of Ziegler-Nichols's tuning, the first method [69] and the second method [70]. Ziegler-Nichols is one of the oldest tuning methods designed for PID controllers. Ziegler Nichols tuning method may provide less performance for complex or advanced systems; it can be the simplest method yet sufficient to give good system control performances for less complicated systems.

In disturbance rejection control, measurement noise profoundly affects the control system performance. Noise from the sensors makes less accurate system measurements. Bad system measurement can make the controller too aggressive or even too tolerant to error, making it unable to correct errors in the system. Then, the Kalman Filter algorithm can be used for sensors with highly sensitive response characteristics to changes in reading values of the object being measured. The filter can reduce noise in output data readings while enhancing efficiency [71, 72].

Disturbance rejection control is highly essential in designing medical telepresence robots since the proposed design is two-wheeled, self-balancing mobile robots. The selfie sticks where the mobile phone/tablet will be hung need to be balanced by the robot so that it can do its function properly. Many studies have analyzed self-balancing robots in detail, but only one approach has been used, and it was not specifically designed for medical telepresence robots. Therefore, this study aimed to propose a disturbance rejection control method and implementation, equipped with noise filtering by Kalman Filter, which can perform in a robust and stable manner, specifically designed for medical telepresence robots. Hopefully, the result of this study can help health professionals to overcome the overwhelming COVID-19 pandemic by making them treat patients more safely with low-cost and easy-to-be-designed medical telepresence robots.

The paper will be organized in the following structure. The first section is an introduction to the problem and research problem formulation. Next, the design of the self-balancing robot will be explained, and the model of the robot will be presented. Then, the conversion of the transfer function model into state-space representation is explained in the next section. The fourth section will contain fundamental theories regarding PID control design, while the fifth will consist of the Ziegler Nichols tuning rules. The fundamental theory of Kalman Filter will be described in the sixth section. Finally, the results and discussion will be presented in the eighth section, and the conclusion will be presented in the last section.

2- Methodology

The research method is shown in Figure 1. The first is system modeling using the Lagrange equation. The obtained model was in state-space representation, but it was then converted to a transfer function to implement the PID. The next step is PID controller design and Ziegler Nichols tuning. After that, the Kalman filter was designed. The next is simulation and hardware design. Then, examinations were done to get the data research. The result of the examination will be analyzed and evaluated. The research is considered finished if the proposed controller can give good results. Vice versa, the examination will be repeated if it cannot provide good performance results.

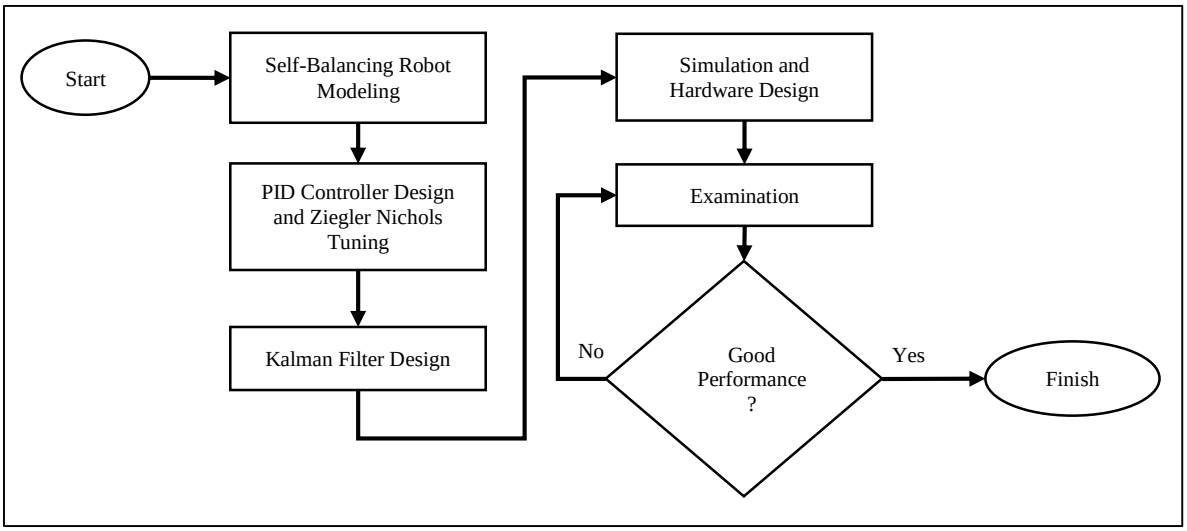

Figure 1. Research Methodology

3- Self-Balancing Robot Modeling

Self-balancing medical telepresence robot modeling was one of the phases in robot design [73, 74]. It was designed to understand the characteristics of the robot's dynamic behavior [75–77]. The first step was to determine the coordinates, the force, the energy, and the Lagrangian functions of the system, and the next step was to determine the motion equation using the Lagrange equation [78–80].

The self-balancing device had two degrees of freedom, allowing it to be represented using two general coordinates [81]. The coordinates used for the analysis of this robot are the horizontal shift from the robot p and the angle shift of the inverted pendulum (θ). p will be positive if the motion is to the right and (θ) positive if the movement is clockwise, calculated from its reverse position as illustrated in Figure 2 [82].

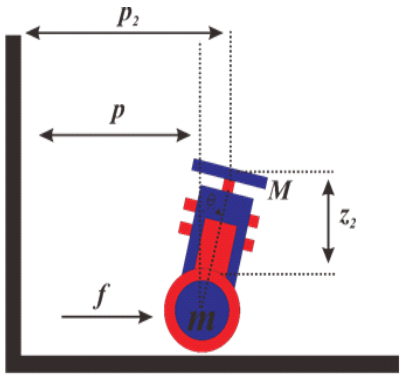

Figure 2. Robot physical system

3-1- Kinetic Energy and Potential Energy

The kinetic energy on the wheel (ER) is the sum of the wheel rotational kinetic energy and the wheel translation, as expressed in Equation 1, m is the mass of the wheel, and p is the rotational speed of the wheel towards the horizontal:

$$E_R = E_{Rotation} + E_{Translation} \quad (1)$$

$$E_R = \frac{1}{2} I\omega^2 + \frac{1}{2} mp^2 \quad (2)$$

$$E_R = \frac{1}{2}\left(\frac{1}{2} mR^2\right)\frac{p^2}{R^2} + \frac{1}{2} mp^2 = \frac{3}{4} mp^2 \quad (3)$$

$$E_B = \frac{1}{2} Mv^2 \quad (4)$$

$$p_2 = p + l\, sin\theta \text{ then } \dot{p}_2 = \dot{p} + l\dot{\theta}\, \cos\theta \quad (5)$$

$$z_2 = l\, \cos\theta \text{ then } \dot{z_2} = -l\dot{\theta} \sin\theta \quad (6)$$

The translation of Equation 4 yields Equation 7;

$$E_B = \frac{1}{2} M\, (\dot{p}_2 + \dot{z}_2)^2 \quad (7)$$

$$E_B = \frac{1}{2} M \left(\dot{p}_2^2 + 2\dot{p}\dot{\theta} l \cos\theta + \frac{1}{2} Ml^2\, \dot{\theta}^2\right) \quad (8)$$

Thus, the total kinetic energy T is expressed in equation

$$T = \frac{3}{4} m\dot{p}^2 + \frac{1}{2} m\dot{p}^2 + Ml \cos\theta\ \dot{p}\dot{\theta} + \frac{1}{2}\ Ml\dot{\theta}^2 \quad (9)$$

The robot moves along the horizontal axis, and the potential energy V of the system was determined entirely by the angle of the inverted pendulum written in Equation 10. g is the acceleration of the earth's gravity, and l is the length of the inverted pendulum

$$V = Mgl \cos\theta \quad (10)$$

3-2- Lagrangian

At this phase, a dynamic system analysis was performed using the Lagrangian method written in Equation 11.

$$L = T - V \tag{11}$$

Substituting Equations 9 and 10 into Equation 11 obtained the Lagrangian Equation 12.

$$L = \left(\frac{3}{4}\, m + \frac{1}{2}\, m\right) \dot{p}^2 + Ml\, cos\, \theta \dot{p}\dot{\theta} + \frac{1}{2} Ml^2 \dot{\theta}^2 - Mgl\, cos\, \theta \tag{12}$$

3-3- The Lagrange Equation

Lagrange analysis on p dan θ was performed after p and θ as the general coordinates had been determined. The completion of the Lagrange equation for p was written in Equation 13.

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{p}}\right) - \frac{\partial L}{\partial p} = f \tag{13}$$

Equation 12 was used to calculate the partial derivative expressed in Equation 13, which yielded Equation 14.

$$\frac{d}{dt}\left\{\left(\frac{3}{2}m + M\right)\dot{p} + Ml\dot{\theta}\cos\theta\right\} - 0 = f \tag{14}$$

$$\left(\frac{3}{2}m + M\right)\ddot{p} + Ml\ddot{\theta}\cos\theta - Ml\dot{\theta}^2 \sin\theta = f \tag{15}$$

The completion of the Lagrange equation for θ was written in Equation 16.

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}}\right) - \frac{\partial L}{\partial \theta} = 0 \tag{16}$$

Equation 14 was used to calculate the partial derivative expressed in Equation 16, which yielded Equation 17.

$$\frac{d}{dt}\left(Ml\dot{p}\ \cos\theta + Ml^2\dot{\theta}\right) - Mgl\ \sin\theta = 0 \tag{17}$$

$$Ml\ddot{p}\cos\theta - ml\dot{p}\sin\theta + Ml^2\ddot{\theta} - Mgl\sin\theta = 0 \tag{18}$$

The self-balancing robot's system equation was then formulated as in Equation 19.

$$\left(\frac{3}{2}m + \ M\right)\ddot{p} + Ml\ddot{\theta}\cos\theta - Ml\dot{\theta}^2\sin\theta = f \tag{19}$$

$$Ml\ddot{p}\cos\theta - Ml\dot{p}\ \sin\theta + \ Ml^2\ddot{\theta} - Mgl\sin\theta = 0$$

3-4- Linearization

Equation 14 is a nonlinear equation. The control system was aimed to keep the inverted pendulum upright. The equation was linearized and applicable for $\sin 0\ \theta \approx$ dan $cos\ 1\ \theta \approx$ as formulated in Equation 20.

$$\left(\frac{3}{2}\, m + M\right)\ \ddot{p} + Ml\ddot{\theta} = f \tag{20}$$

$$Mp + Ml\ddot{\theta} - Mg\theta = 0 \tag{21}$$

By completing Equation 15, $\ddot{p}$ dan $\ddot{\theta}$ were obtained, as written in Equations 22 and 23.

$$\ddot{p} = \frac{2f}{3m} - \frac{2Mg}{3m}\ \theta \tag{22}$$

$$\ddot{\theta} = \frac{2f}{3ml} + \left(\frac{3m + 2M}{3ml}\right) g\theta \tag{23}$$

Equation 23 was expressed in a state-space equation to obtain Equation 24, u denoted the force acting on the robot self-balancing system x.

$$\dot{x} = Ax + Bu \tag{24}$$

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \left(\frac{3m+2M}{3ml}\right)g & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{2M}{3g}g & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \\ p \\ \dot{p} \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{2}{3ml} \\ 0 \\ \frac{2}{3m} \end{bmatrix} u \tag{25}$$

4- State-Space to Transfer Function

The self-balancing model's state space was converted to a transfer function model to be controlled by the PID control. The following are state-space equations.

$$\dot{x} = Ax + Bu \tag{26}$$

$$y = Cx + Du \tag{27}$$

By taking Laplace transformation (with zero value at the initial conditions), the equations turned into:

$$sX(s) = AX(s) + BU(s) \tag{28}$$

$$Y(s) = CX(s) + DU(s) \tag{29}$$

Then, Equation 28 can be rewritten as:

$$(sI - A)X(s) = BU(s) \tag{30}$$

Both sides were then pre-multiplied with $(sI - A)^{-1}$ so that the following equation can be obtained.

$$(s) = (sI - A)^{-1}BU(s) \tag{31}$$

By substituting X (s) in the output equation, Equation 29 can be expressed as:

$$Y(s) = (C(sI - A)^{-1}B + D)U(s) \tag{32}$$

Finally, the transfer function equation can be formulated as:

$$\frac{Y(s)}{U(s)} = C(sI - A)^{-1}B + D \tag{33}$$

5- PID Control Design

The PID control characteristics were strongly influenced by combining each control action Proportional K_p, Integral K_i And Derivative (K_d) [83]. The regulation of the K_p, K_i, and K_d constant values showed the prominent nature of each control action [84]. When one or two of the control actions were more prominently set out, they would significantly contribute to the overall system. Control action parameters were not independent, so if one of the values was changed, there would be a change in the system response [85]. Figure 3 shows the block diagram for the combination of proportional (P), integral (I), and derivative (D) control action.

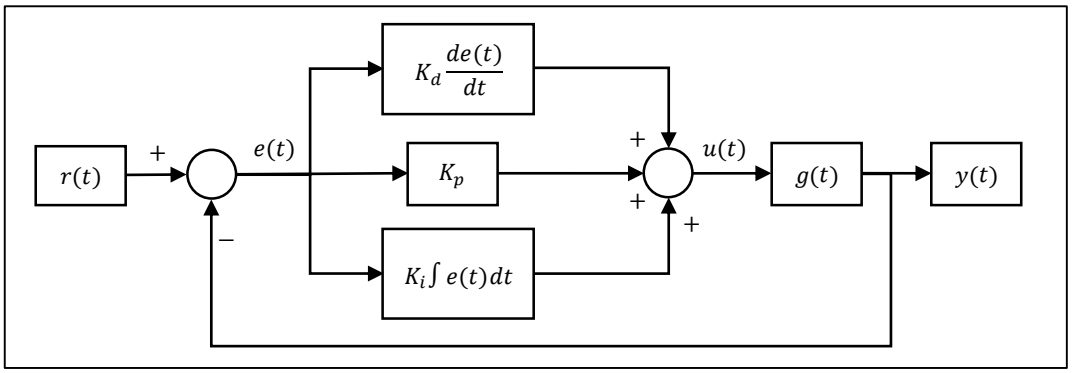

Figure 3. PID control block diagram

The following equations are the mathematical equation for the parallel PID control action:

$$u(t) = K_p e(t) + K_i \int_0^t e(t)dt + K_d \frac{de(t)}{dt} \quad (34)$$

with

$$K_d = K_p T_d \quad (35)$$

$$K_i = \frac{K_p}{T_i} \quad (36)$$

6- Ziegler Nichols Tuning Rules

The Ziegler-Nichols rules related the Plant transient response characteristics to the values of K_p, T_i, and T_d [70]. The parameter tuning in this research was conducted based on the second method of the Ziegler Nichols rules, in which the values $T_i = \infty$ and $T_d = 0$ must be set first; the response from the plant can only be seen if the value of K_p is set (Figure 4). This method involved giving the value of K_p ranging from 0 to the critical value K_{cr} before the plant responds with an oscillating curve. This approach could not be used if the response did not display an oscillating curve.

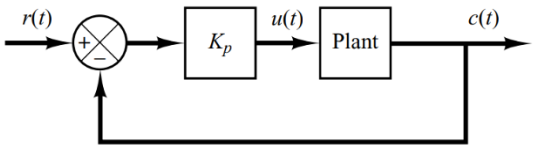

Figure 4. Giving K_p Value to Plant

This method was performed by providing K_p value to the plant periodically from 0 to critical value of K_{cr} until the plant responds with an oscillating curve. Thus, the critical value of K_{cr} and period value of P_{cr} were determined using the experimental method. Figure 5 illustrates the oscillating response expected from the system. Table 1 presents the PID formula using the second method of Ziegler-Nichols rules. P, PI, and PID are three different types of control.

Table 1. PID formulation based on the second method of Zigler-Nichols

Control Type	K_p	T_i	T_d
P	$0.5K_{cr}$	∞	0
PI	$0.45K_{cr}$	$\frac{1}{1.2}P_{cr}$	0
PID	$0.6K_{cr}$	$0.5P_{cr}$	$0.5P_{cr}$

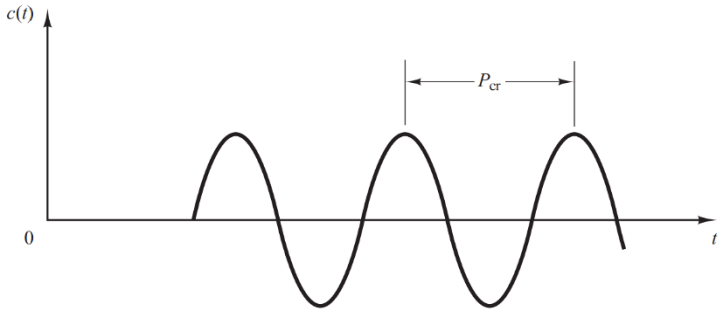

Figure 5. The oscillating response

7- Kalman Filter

The noise in a signal containing readings or measurements was reduced or eliminated using the Kalman Filter [86]. It estimated the process state in a system, minimizing the mean square error value before, now, or in the future [87, 88]. The two processes in Kalman Filter were time update and measurement update [89]. Time update was the process of predicting future readings, and measurement update was the process of correcting the predicted value. The noise in the measurement results was used to calculate time update and measurement update to obtain a more accurate estimation value [90]. The two processes in the Kalman filter were the prediction stage and the correction stage [91].

7-1- Prediction Stage (Time Update)

A prediction stage is a process in which future readings are predicted to obtain state and covariance values ranging from $k-1$ to k [92]. Two equations at the Prediction Stage were State Prediction and Error Covariance Prediction [93, 94].

The state Prediction equation is

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_k \quad (37)$$

where A is state transition matrix and B is the control matrix.

The Error Covariance Prediction equation is;

$$P_k^- = AP_{k-1}A^T + Q \quad (38)$$

where P is the state variance matriks and Q is the process variance matrix.

7-2- Correction Stage

A correction stage is a process of correcting the predicted value [95]. The first step was to calculate the Kalman gain K_k (39). Next, the actual process value z_k, which would be used for post-estimation state calculations (40), was determined. Finally, the post-estimation error covariance values (41) were obtained using Kalman Gain equation, Estimation, and Covariance Error in the correction stage.

The Kalman Gain equation is;

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1} \quad (39)$$

where H is the measurement matrix and R is measurement variance matrix.

The Estimation equation is;

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) \quad (40)$$

The covariance error equation is

$$P_k = (I - K_k H)\, P_k^- \quad (41)$$

The prediction and correction stages were repeated several times, with the post-estimation values used to calculate the next new pre-estimation value.

8- Hardware Design

8-1- Electrical Design

Figure 5 shows some of the electronic components used to build the robot. It consisted of Arduino nano components, DC motors with encoders, L298N motor drivers, MPU sensors 6050, a 16×2 LCD, and a push-button. The components were assembled and simulated to be integrated into a robot-enabled system.

8-2- Robot mechanical design

The robot comprises three chassis arranged in three levels, as shown in Figure 6. The top and bottom chassis were made of 2 mm black acrylic material. The middle chassis was produced using a PCB and control center consisting of an Arduino microcontroller, an L298N driver, and a GY-521 MPU 6050 sensor. The robot chassis included switches, push buttons, 16×2 LCD, and the bottom chassis included batteries and two DC encoders. It had two wheels as a driving force. The robot's dimensions and weight were 16×16×6 cm and 642 grams, and the wheels' diameter and weight were 6.5 cm and 36 grams.

The robot was designed in a two-wheel configuration to increase its power efficiency, maneuverability, and stability-enhancing behavior. It was also battery-operated with a Lithium rechargeable battery. Figure 7 shows the mechanical design of the prototype. A height-adjustable selfie sticks with a rounded base made of durable plastic will be added to the robot using 3M industrial double-sided tape. Meanwhile, the universal phone/tablet holder on the tip of the selfie stick was equipped with a mechanical, adjustable neck-angle gripper.

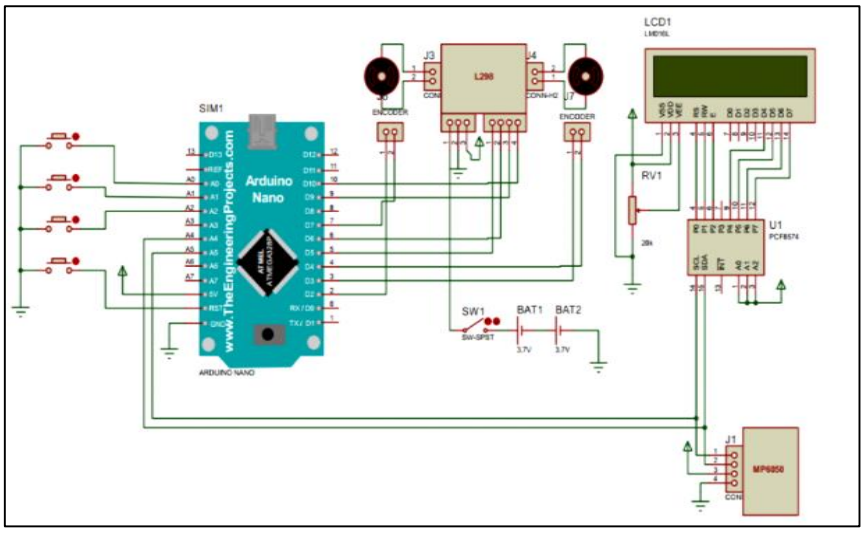

Figure 6. Electrical design

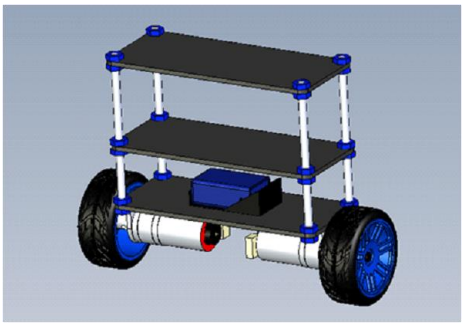

Figure 7. Self-balancing robot prototype design

8-3- Control System Design

Figure 8 shows the design of the self-balancing robot control system. The PID control and the Kalman filter were two methods to stabilize the robot. The tuning method to find the PID parameter values was the second method of Ziegler-Nichols's rules. The Kalman filter was combined with an Inertial Measurement Unit (IMU) sensor to eliminate noise.

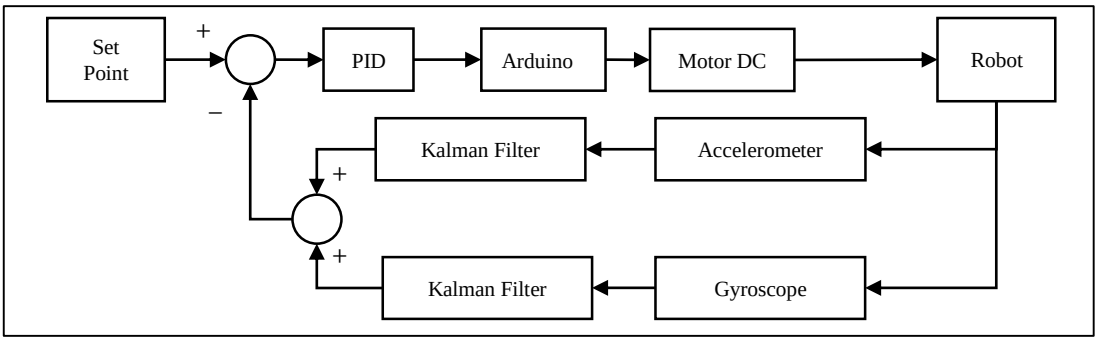

Figure 8. Design of the proposed disturbance rejection control

A system modeling for the self-balancing robot was required to design a PID control system. Table 2 presents a set of parameters to determine the transfer function values of the robot mass, wheel mass, robot length, and gravitational acceleration values.

Table 2. Parameters to find transfer function values

Notation	Parameter	Value	Unit
M	Robot Mass	0.642	Kg
m	Wheel Mass	0.036	Kg
l	Robot Length	0.16	Meter
g	Gravitation	9.8	m/s^2

The parameter values used for state-space modeling were written in Equation 19. The model was then converted into a transfer function using Equation 25 to be more easily controlled by the PID. The transfer function equation was written in Equation 42.

$$G = \frac{0.024^2 + 0.1572}{s^4 - 2.776^{-17}s^3 - 0.02551s^2} \tag{42}$$

After obtaining the transfer function value from the robot system, tuning the PID parameter as the control value of the robot was completed.

8-4- Communication and Connectivity Design

In order to provide telepresence and automatic function to the robot, the robot will be equipped with a DTMF decoder module. DTMF was chosen due to its reasonably small size and dimension (37×25×12 mm, 10g) while enabling motor control through Skype/WhatsApp communication. The proposed connectivity design is described briefly as follows. A tablet/smartphone with an internet connection is attached to the robot and connected to the DTMF module. When the tablet/smartphone receives a video call through Skype/WhatsApp, it will automatically answer the call. The caller will type a specific number as code to move the robot to the right/left; the specific number will emit a sound with a particular frequency, and DTMF will decode it. Therefore, the robot can be remotely controlled based on the specific number that the caller typed.

8-5- Determining the K_{cr} and P_{cr}

The K_{cr} value of 51.667 generated stronger oscillating and constant graphs. Figure 9 shows the graph response. It illustrates how to get the values $(x_1, y_1) = (29.6, 1.985)$ and $(x_2, y_2) = (30.46, 1.983)$.

The Equation 43 is the value of P_{cr} obtained from the graph:

$$P_{cr} = x_2 - x_1$$
$$P_{cr} = 30.46s - 29.76s = 0.7 \tag{43}$$

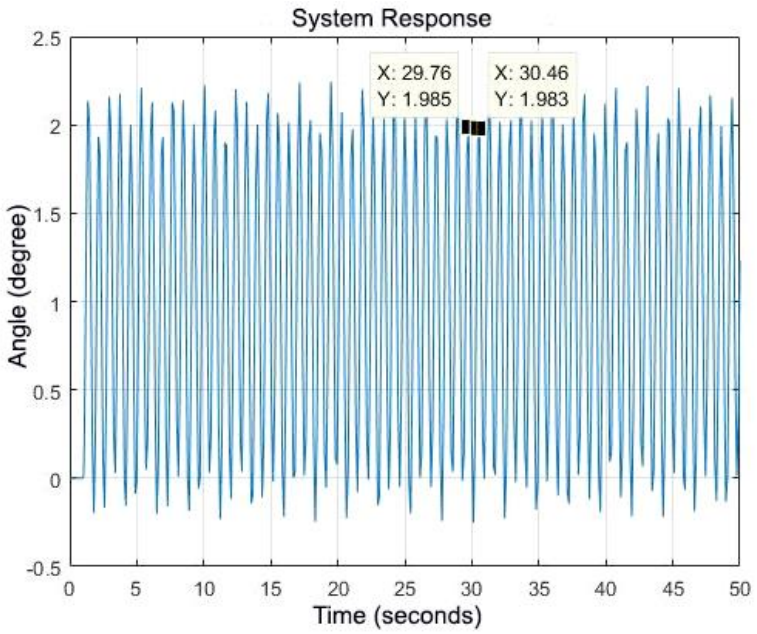

Figure 9. Test Results for K_{cr} Value

8-6- Determining K_p, T_i and T_d values

After obtaining the K_{cr} and P_{cr} values, the second Ziegler-Nichols rules were used to calculate the K_p , T_i and T_d values and the results were formulated in Equations 39, 40, and 41.

$$K_p = 0.6 \text{ x } K_{cr}$$
$$K_p = 0.6 \text{ x } 51.667 = 31.002 \tag{44}$$

$$T_i = 0.5 \text{ x } P_{cr} \tag{45}$$

$$T_i = 0.5 \text{ x } 0.7 = 0.35$$

$$T_d = 0.125 \text{ x } P_{cr} \tag{46}$$

$$T_d = 0.125 \text{ x } 0.7 = 0.0875$$

The T_i and T_d values were modified with Equations 45 and 46 to adjust the oscillation acceleration and noise according to system requirements by computing the values of certain variables as needed. The value of those variables was searched experimentally.

$$T_iM = T_i \times T_iV \tag{47}$$

$$T_iM = 0.35 \times 17 = 5.95$$

$$T_dM = T_d \times T_dV \tag{48}$$

$$T_dM = 0.0875 \times 46.5 = 4.06875$$

where T_im is T_i modification, T_dm is T_d modification, T_iV is the variable of T_i, T_dV is the variable of T_d. Equations 49 and 50 formulate the parameter values conversion from T_iM and T_dM to K_i dan K_d.

$$K_i = \frac{K_p}{T_iM} \tag{49}$$

$$K_i = \frac{31.002}{5.95} = 5.2104201$$

$$K_d = K_p \times T_dM \tag{50}$$

$$K_d = 31.002 \times 4.375 = 126.1393875$$

9- Results and Discussion

Figure 10 displays the three-level self-balancing robots created, controlled by PID control and Kalman filter. The robot used two DC motor drives. The first and third level chassis were made of Acrylic, while the second level chassis was made of PCB installed with the electrical parts. The PID control used Ziegler Nichols's rules and was tested using simulations and an actual robot implementation, whereas the Kalman filter was tested using an actual robot implementation.

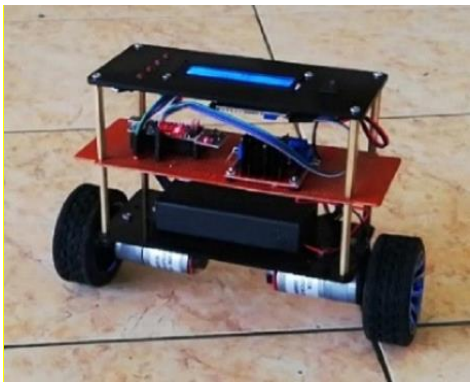

Figure 10. Self-balancing robot prototype

9-1- System Simulation

Simulations on this robot system were performed on the Simulink in Matlab software. It was conducted when the transfer function value and the PID tuning value of the system had been obtained using a step test signal. It aimed to observe the transient response of the system.

9-2- System Simulation without PID

Figure 11 illustrates the system simulation without PID application. The self-balancing robot was modeled using the transfer function with a step signal. Figure 12 shows that the system's initial response was stationary while an upward movement was shown an infinite period later. It proves that the system without a PID controller is unstable.

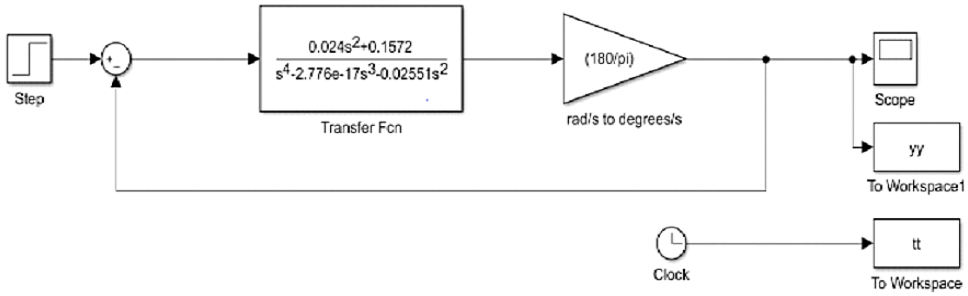

Figure 11. **System simulation without PID**

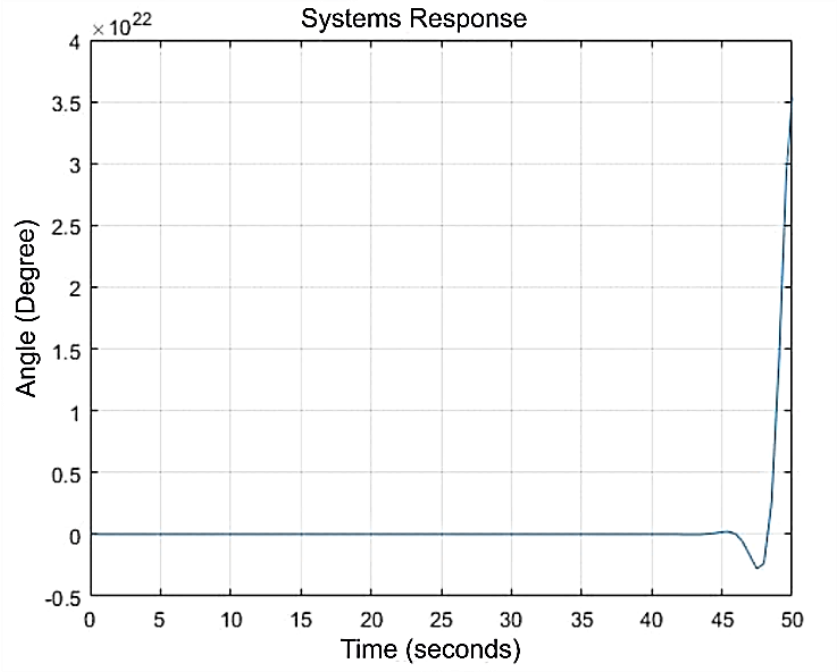

Figure 12. **System simulation results in PID**

9-3- System Simulation with PID

Figure 13 illustrates the block diagram of system simulation with PID. The input was a step function and PID control with K_p, K_i, and K_d of 31.002, 5.2104201, and 126.1393875, respectively. The step response resulted is displayed in Figure 13.

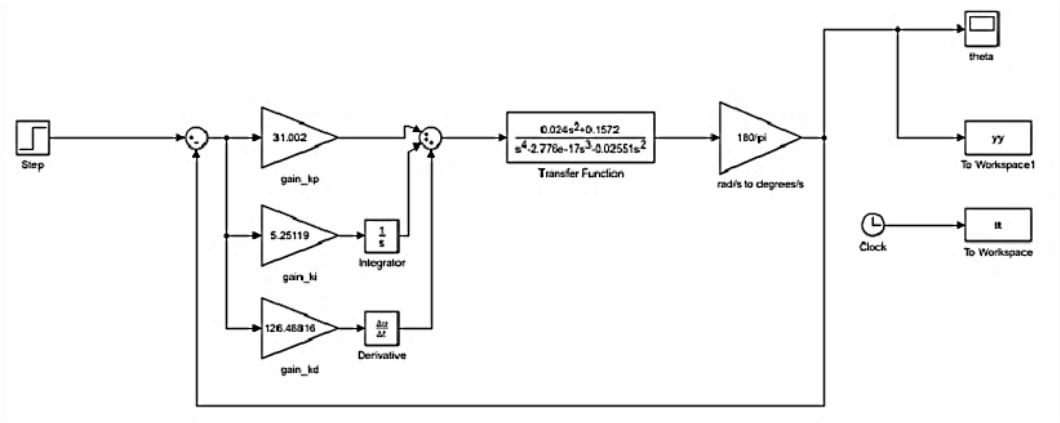

Figure 13. **Block diagram of the system with PID**

It demonstrates that the PID controller improves the system's reaction (Figure 14). It is proven by the transition period as presented in the table. The time sampling in this system testing was 50 seconds with a rise-time value of 4.448s, pre-shoot 0.625%, overshoot 24.375%, and undershoot 1.696%.

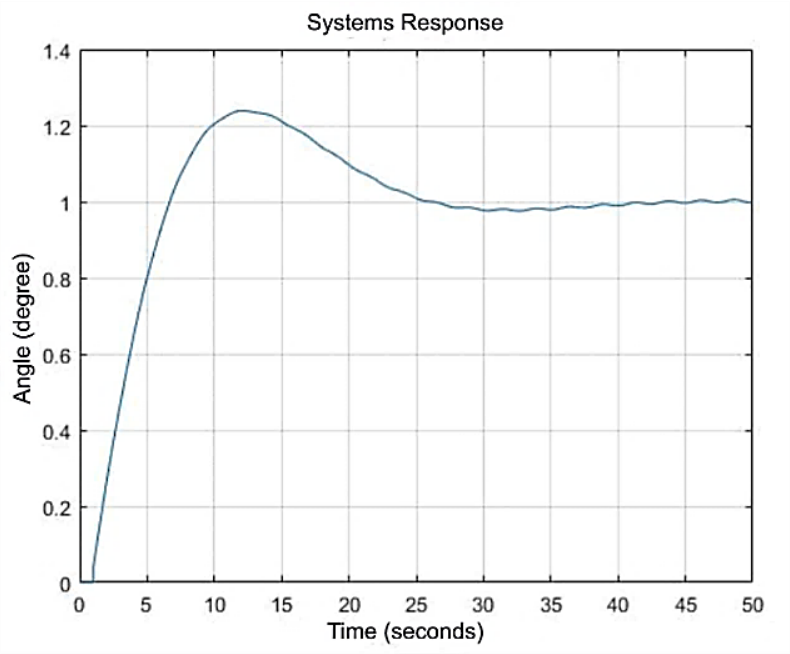

Figure 14. System simulation result with PID

9-4- Testing for Real Robot System without PID

Figure 15 illustrates the comparison of angle readings of the system with and without PID. Three graph curves show the set-point value, roll angle value without Kalman filter, and roll angle value with Kalman filter colored in black, blue, and red, respectively. The red-colored gravitational curve shows that the Kalman filter can reduce the noise signal from the IMU. The self-balancing robot cannot be stabilized without implementing any controller.

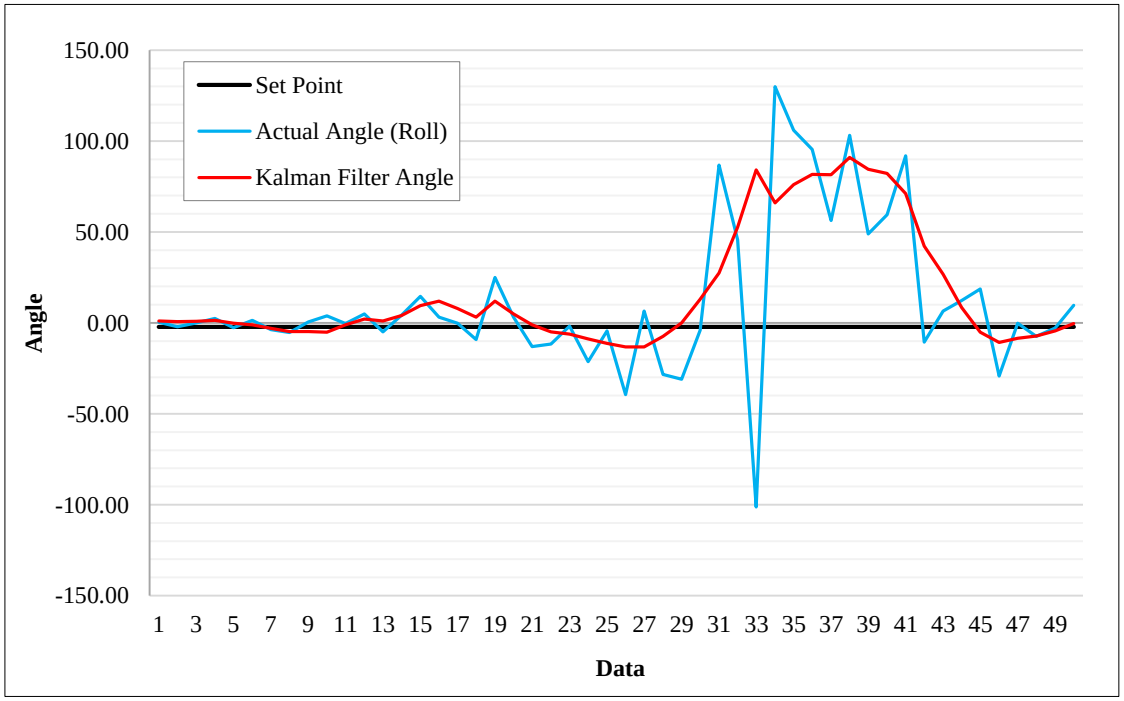

Figure 15. Comparison of angle reading values without PID

Figure 16 displays the motor speed response without PID. It shows that the movement of the two motors is constant, and there is a condition in which the motor moves backward and then quickly moves forward at a relatively fast speed. When the graph shows an increasing line, the motor moves forward; while the graph exposes a decreasing line, the motor moves backward. At the beginning of the movement, the robot's response was slow, but later it moved fast. Based on the motor movement, the robot has no feedback to the motor to be used as a reference to stay balanced.

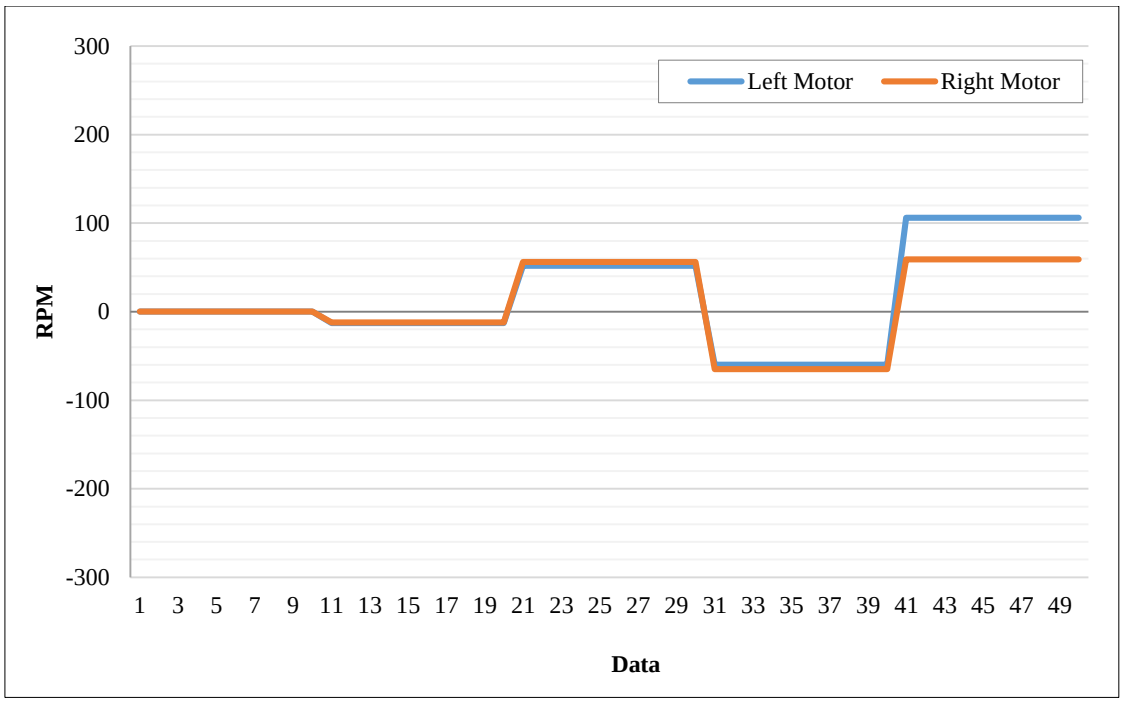

Figure 16. System's motor speed response without PID

9-5- Testing for Real Robot System Testing with PID

Figure 17 illustrates the comparison of angle readings of the system PID. Three graph curves are set-point curves, roll angle curves without Kalman filter, and roll angle curves with Kalman filters in black, blue, and red lines. The figure shows that the self-balancing robot is stable. The disturbances can be eliminated by using the filter.

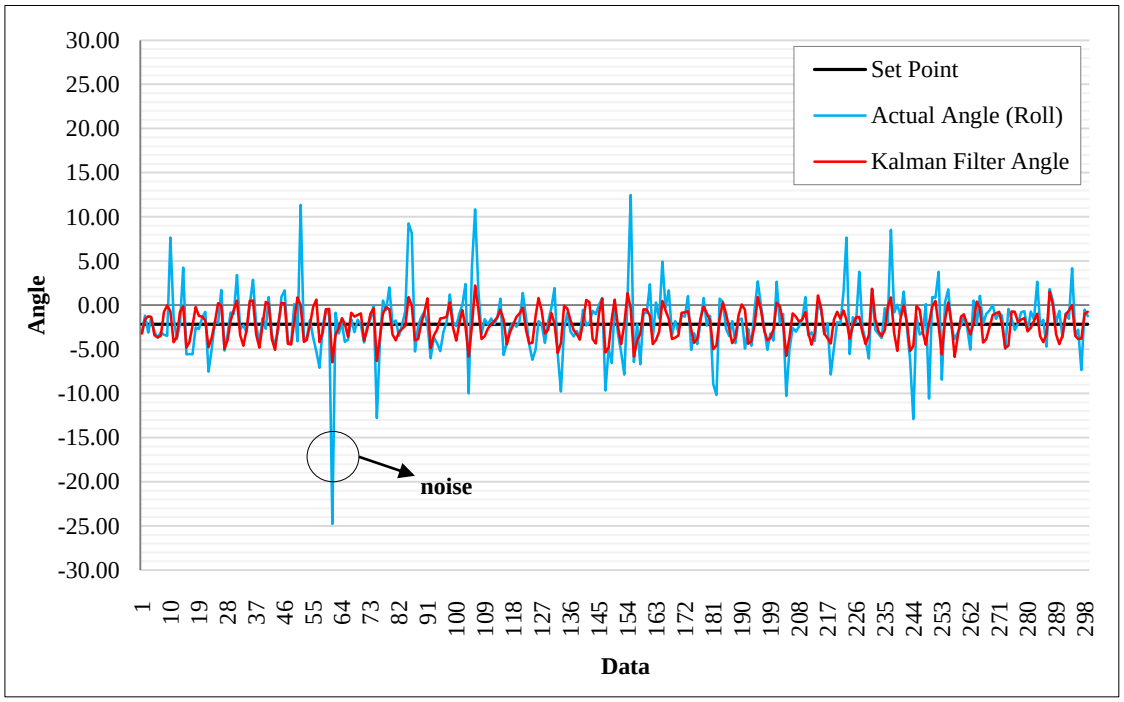

Figure 17. Comparison of angle reading values with PID

Figure 18 illustrates the motor speed response system with PID. The movement direction was the same, but the left motor was more dominant than the right because they had different responses when carrying loads even though they had similar specifications. The graph exposes that the motor movement oscillates due to balancing the position by moving forward and backward. The motor movement referred to robot angle changes. When the robot was about to fall forward, the motor responded by moving forward or CW with a certain speed to return to a balanced position. Conversely, when the robot was about to fall backward, the motor responded by moving backward or CCW.

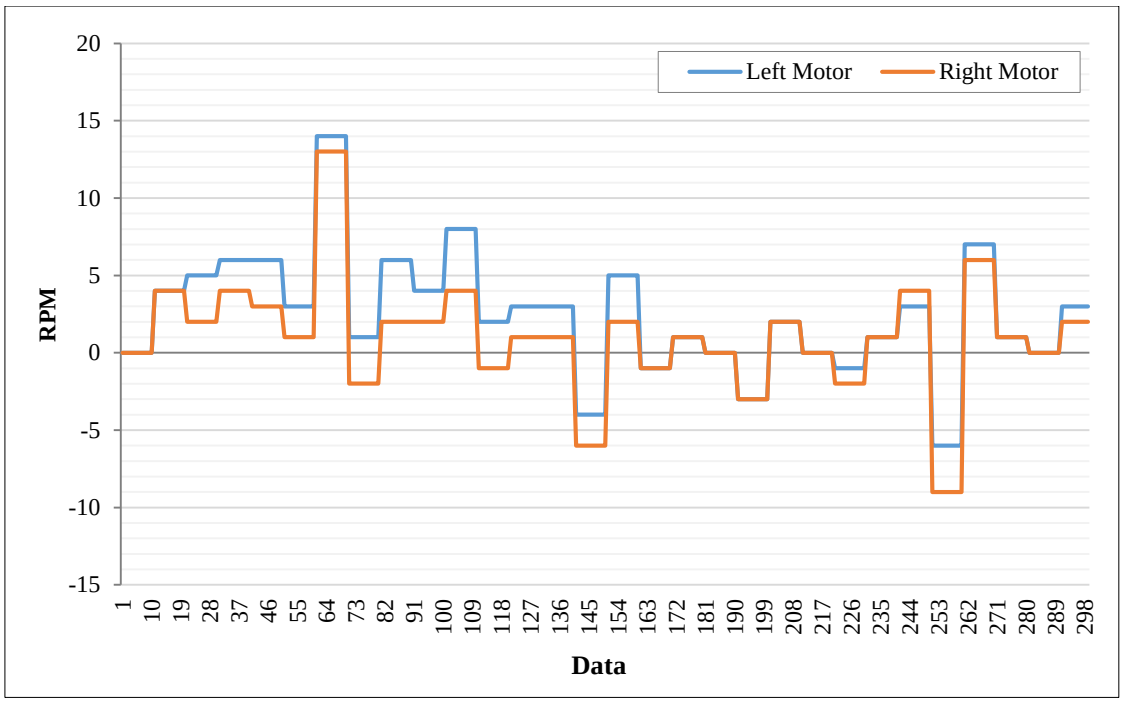

Figure 18. Motor speed response system with PID

9-6- Testing for Real Robot System for Disturbance

The test was performed after it was found that a more stable system could be applied to this robot system by using PID control. The robot was pushed to observe whether it could maintain its balance or not. Figure 19 reveals the system response to disturbance. The response given by the system to the robot appears to be satisfactory. The robot was pushed two times. The roll angle reading showed an overshoot, but it was not the noise generated by the sensor reading but an external disturbance. The Kalman filter, used as an algorithm to filter the actual angle readings, could reduce and even eliminate noise caused by external disturbance for the robot to remain stable to reach or approach the set-point. It enabled the robot to return to its balance without falling.

Figure 20 shows the motor speed response to the disturbance. The robot was firstly stable, then moved backward or CCW when pushed. The robot retreat was the response. When the motor remained stable, it indicated that the system improved the robot's state and gave feedback to the motor to stabilize so that the robot did not fall.

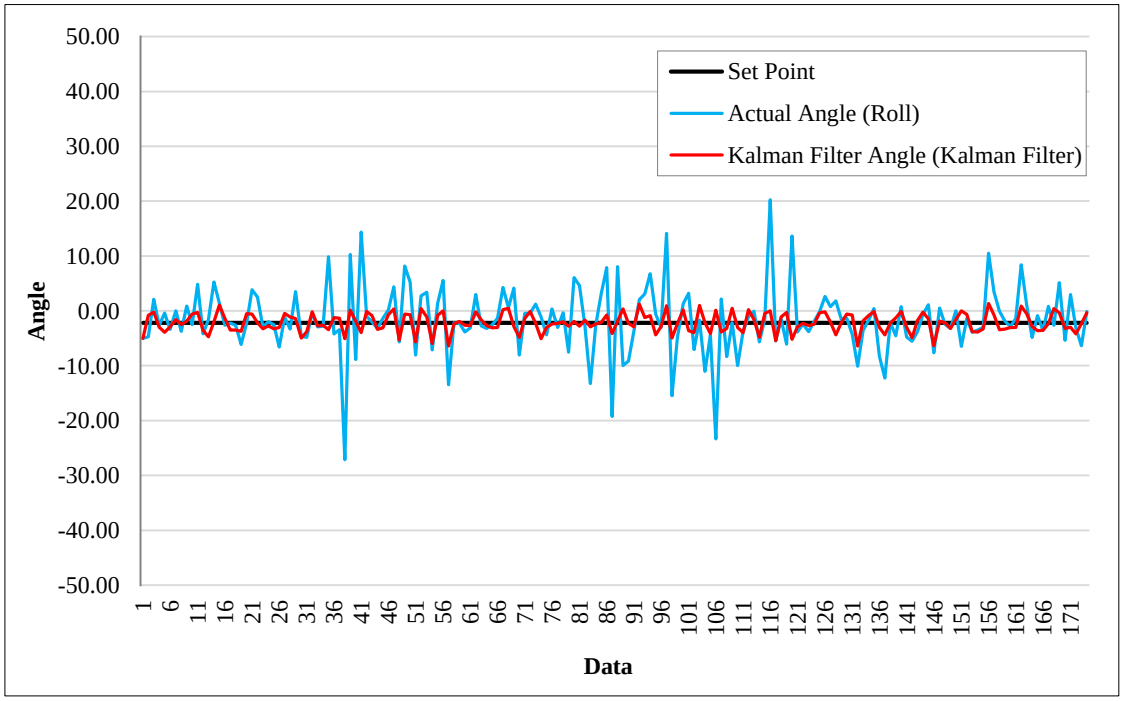

Figure 19. System response to disturbance

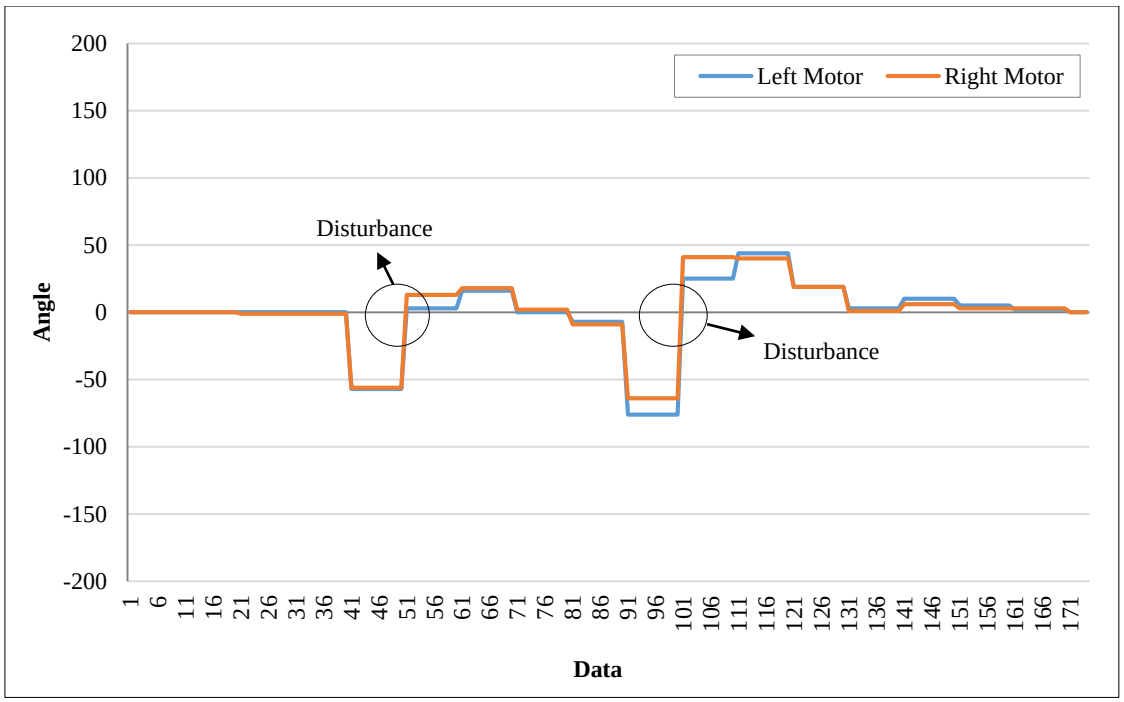

Figure 20. Motor speed response to disturbance

10- Conclusion

A robust and stable disturbance rejection control was successfully designed specifically for the proposed medical telepresence robot. The PID parameters were tuned using the second method of Ziegler Nichols rules, resulting in $K_p = 31.002$, $K_i = 5.167$ and $K_d = 125.992128$ for a prototype of a self-balancing robot with a size of 16×16×16 cm. The combination of GY-521 MPU 6050 sensor reading and the Kalman Filter aimed to get an accurate reading value. The robot system was tested using simulation and real robots under three different conditions: without a PID controller, with a PID controller, and when given a disturbance. The measurement evaluations were performed by comparing the Kalman filter with the GY-521 MPU 6050 sensor readings. According to the simulation and real-time implementation results, the robot maintained balance for approximately one hour before the system was turned off and re-balanced itself when applied with external disturbances. The Kalman filter provides the best for the robot's system performance because it can reduce and eliminate noise in the readings when the robot is either stationary or subjected to a disturbance.

11- Declarations

11-1-Author Contributions

I.I., A.M., N.M.R., T.K.H., and M.A.S. contributed to the design and implementation of the research, to the analysis of the results and to the writing of the manuscript. All authors have read and agreed to the published version of the manuscript.

11-2-Data Availability Statement

The data presented in this study are available in article.

11-3-Funding and Acknowledgements

This research was supported by a WCR Project grant from DIKTI through Research Directorate, Universitas Muhammadiyah Yogyakarta, with the contract number: 1867/E4/AK.04/2021 dated 07 June 2021 and Agreement /Contract Number: 165/E4.1/AK.04.PT/2021 dated 12 July 2021; the contract between LLDIKTI with PTS UMY was registered with Number 3279.4/LL5/PG/2021, and the contract between PTS/LPPM and Researcher was registered with Number: 008/SP.LRI/VIII/2021, awarded to A/P Dr.,Ir., Iswanto., S.T., M. Eng., IPM.

11-4-Conflicts of Interest

The authors declare that there is no conflict of interests regarding the publication of this manuscript. In addition, the ethical issues, including plagiarism, informed consent, misconduct, data fabrication and/or falsification, double publication and/or submission, and redundancies have been completely observed by the authors.

12- References

[1] Jones, D. L., Baluja, M. Q., Graham, D. W., Corbishley, A., McDonald, J. E., Malham, S. K., Hillary, L. S., Connor, T. R., Gaze, W. H., Moura, I. B., Wilcox, M. H., & Farkas, K. (2020). Shedding of SARS-CoV-2 in feces and urine and its potential role in person-to-person transmission and the environment-based spread of COVID-19. Science of the Total Environment, 749, 141364. doi:10.1016/j.scitotenv.2020.141364.

[2] Widyasmoro, W., Suwarno, I., Surahmat, I., Nugraha, T. A., & Al_barazanchi, I. (2022). Dissemination of technology utilization of FM community radio as a means to support teaching learning activities for students during the covid-19 pandemic at Muhammadiyah Elementary School Tlogolelo, Hargomulyo, Kokap District, Kulon Progo, DIY. Jurnal Pengabdian Dan Pemberdayaan Masyarakat Indonesia, 2(1), 34–42.

[3] Suwarno, I., Ma'arif, A., Maharani Raharja, N., Nurjanah, A., Ikhsan, J., & Mutiarin, D. (2021). IoT-based Lava Flood Early Warning System with Rainfall Intensity Monitoring and Disaster Communication Technology. Emerging Science Journal, 4, 154–166. doi:10.28991/esj-2021-sp1-011.

[4] Ralph, R., Lew, J., Zeng, T., Francis, M., Xue, B., Roux, M., Ostadgavahi, A. T., Rubino, S., Dawe, N. J., Al-Ahdal, M. N., Kelvin, D. J., Richardson, C. D., Kindrachuk, J., Falzarano, D., & Kelvin, A. A. (2020). 2019-nCoV (Wuhan virus), a novel Coronavirus: Human-to-human transmission, travel-related cases, and vaccine readiness. Journal of Infection in Developing Countries, 14(1), 3–17. doi:10.3855/jidc.12425.

[5] Iswanto, I., Raharja, N. M., Maarif, A., Supangkat, G., Pandey, A., Deniz, C., Nurjanah, A., Rijalusalam, D. U., Sánchez-López, C., & Ahmad, I. (2021). Empowerment of mosque communities to increase body immune with aroma therapy robots. Jurnal Pengabdian Dan Pemberdayaan Masyarakat Indonesia, 1(4), 127–135.

[6] Wang, D., Hu, B., Hu, C., Zhu, F., Liu, X., Zhang, J., Wang, B., Xiang, H., Cheng, Z., Xiong, Y., Zhao, Y., Li, Y., Wang, X., & Peng, Z. (2020). Clinical Characteristics of 138 Hospitalized Patients with 2019 Novel Coronavirus-Infected Pneumonia in Wuhan, China. JAMA - Journal of the American Medical Association, 323(11), 1061–1069. doi:10.1001/jama.2020.1585.

[7] Ziafati Bagherzadeh, S. H., & Toosizadeh, S. (2022). Eye Tracking Algorithm Based on Multi Model Kalman Filter. HighTech and Innovation Journal, 3(1), 15–27. doi:10.28991/hij-2022-03-01-02.

[8] Sofia, S., Ariani, M., & Sa, Z. (2021). Socialization of covid-19 prevention for children at kaye aceh village, southwest aceh regency, aceh province, indonesia. Jurnal Pengabdian Dan Pemberdayaan Masyarakat Indonesia, 1(4), 142–148.

[9] Huang, L., Lin, G., Tang, L., Yu, L., & Zhou, Z. (2020). Special attention to nurses' protection during the COVID-19 epidemic. Critical Care, 24(1), 1–3,. doi:10.1186/s13054-020-2841-7.

[10] Hasan, N. A., Fauzi, R., Lestari, Y., Siti, F., & Alfiana, R. D. (2021). Technology Dissemination of Blood type Checking and Health Examination by the Thematic Community Service Program (KKN) of Alma Ata University in Hamlet of Kejambon Kidul. Jurnal Pengabdian Dan Pemberdayaan Masyarakat Indonesia, 1(9), 371–378.

[11] Wang, M., Pan, C., & Ray, P. K. (2021). Technology Entrepreneurship in Developing Countries: Role of Telepresence Robots in Healthcare. IEEE Engineering Management Review, 49(1), 20–26. doi:10.1109/emr.2021.3053258.

[12] Riduwan, & Ma'ruf, F. (2021). Dissemination of Sharia Cooperative Research, Solutions during a Pandemic. Jurnal Pengabdian Dan Pemberdayaan Masyarakat Indonesia, 1(11), 459–468.

[13] Shamblaw, A. L., Rumas, R. L., & Best, M. W. (2021). Coping during the COVID-19 pandemic: Relations with mental health and quality of life. Canadian Psychology/Psychologie Canadienne, 62(1), 92–100. doi:10.1037/cap0000263.

[14] Al Firdaus, A. A., Muafiah, E., Heriyudanta, M., & Al_barazanchi, I. (2021). Empowerment of Marketing Strategies of Angkringan Traders through Social Media during Covid-19 Time in Ponorogo. Jurnal Pengabdian Dan Pemberdayaan Masyarakat Indonesia, 1(3), 84–94.

[15] Zhu, X., & He, B. (2021). Underactuated rehabilitation robotics for hand function. Journal of Robotics and Control (JRC), 2(5), 337–341. doi:10.18196/jrc.25103.

[16] Shalal, N. S., & Aboud, W. S. (2021). Smart robotic exoskeleton: A 3-dof for wrist-forearm rehabilitation. Journal of Robotics and Control (JRC), 2(6), 476–483. doi:10.18196/jrc.26125.

[17] Fitter, N. T., Raghunath, N., Cha, E., Sanchez, C. A., Takayama, L., & Mataric, M. J. (2020). Are We There Yet? Comparing Remote Learning Technologies in the University Classroom. IEEE Robotics and Automation Letters, 5(2), 2706–2713. doi:10.1109/LRA.2020.2970939.

[18] Abibullaev, B., Zollanvari, A., Saduanov, B., & Alizadeh, T. (2019). Design and Optimization of a BCI-Driven Telepresence Robot through Programming by Demonstration. IEEE Access, 7, 111625–111636. doi:10.1109/ACCESS.2019.2933268.

[19] Rhee, T., Thompson, S., Medeiros, D., Dos Anjos, R., & Chalmers, A. (2020). Augmented Virtual Teleportation for High-Fidelity Telecollaboration. IEEE Transactions on Visualization and Computer Graphics, 26(5), 1923–1933. doi:10.1109/TVCG.2020.2973065.

[20] Sun, D., Kiselev, A., Liao, Q., Stoyanov, T., & Loutfi, A. (2020). A New Mixed-Reality-Based Teleoperation System for Telepresence and Maneuverability Enhancement. IEEE Transactions on Human-Machine Systems, 50(1), 55–67. doi:10.1109/THMS.2019.2960676.

[21] Becerra, I., Suomalainen, M., Lozano, E., Mimnaugh, K. J., Murrieta-Cid, R., & Lavalle, S. M. (2020). Human Perception-Optimized Planning for Comfortable VR-Based Telepresence. IEEE Robotics and Automation Letters, 5(4), 6489–6496. doi:10.1109/LRA.2020.3015191.

[22] Zhong, M., Li, C., Liu, L., Wen, J., Ma, J., & Yu, X. (2020). Fuzzy Neighborhood Learning for Deep 3-D Segmentation of Point Cloud. IEEE Transactions on Fuzzy Systems, 28(12), 3181–3192. doi:10.1109/TFUZZ.2020.2992611.

[23] Khenak, N., Vezien, J., & Bourdot, P. (2020). Spatial Presence, Performance, and Behavior between Real, Remote, and Virtual Immersive Environments. IEEE Transactions on Visualization and Computer Graphics, 26(12), 3467–3478. doi:10.1109/TVCG.2020.3023574.

[24] Erat, O., Hoell, M., Haubenwallner, K., Pirchheim, C., & Schmalstieg, D. (2019). Real-Time View Planning for Unstructured Lumigraph Modeling. IEEE Transactions on Visualization and Computer Graphics, 25(11), 3063–3072. doi:10.1109/TVCG.2019.2932237.

[25] Peng, X. (2019). New multiparametric similarity measure and distance measure for interval neutrosophic set with IoT industry evaluation. IEEE Access, 7(c), 28258–28280. doi:10.1109/ACCESS.2019.2902148.

[26] Chen, H., Huang, P., & Liu, Z. (2019). Mode Switching-Based Symmetric Predictive Control Mechanism for Networked Teleoperation Space Robot System. IEEE/ASME Transactions on Mechatronics, 24(6), 2706–2717. doi:10.1109/TMECH.2019.2946197.

[27] Albert, K., Phogat, K. S., Anhalt, F., Banavar, R. N., Chatterjee, D., & Lohmann, B. (2020). Structure-Preserving Constrained Optimal Trajectory Planning of a Wheeled Inverted Pendulum. IEEE Transactions on Robotics, 36(3), 910–923. doi:10.1109/TRO.2020.2985579.

[28] Pathak, K., Franch, J., & Agrawal, S. K. (2005). Velocity and position control of a wheeled inverted pendulum by partial feedback linearization. IEEE Transactions on Robotics, 21(3), 505–513. doi:10.1109/TRO.2004.840905.

[29] Rizal, Y., Wahyu, M., Noor, I., Riadi, J., Feriyadi, F., & Mantala, R. (2021). Design of an Adaptive Super-Twisting Control for the Cart-Pole Inverted Pendulum System. Jurnal Ilmiah Teknik Elektro Komputer Dan Informatika, 7(1), 161. doi:10.26555/jiteki.v7i1.20420.

[30] Park, M. S., & Chwa, D. (2009). Swing-up and stabilization control of inverted-pendulum systems via coupled sliding-mode control method. IEEE Transactions on Industrial Electronics, 56(9), 3541–3555. doi:10.1109/TIE.2009.2012452.

[31] Motoi, N., Motoi, N., Suzuki, T., & Ohnishi, K. (2009). A Bipedal Locomotion Planning Based on Virtual Linear Inverted Pendulum Mode. IEEE Transactions on Industrial Electronics, 56(1), 54–61. doi:10.1109/TIE.2008.2004663.

[32] Aoyama, T., Hasegawa, Y., Sekiyama, K., & Fukuda, T. (2009). Stabilizing and direction control of efficient 3-D biped walking based on PDAC. IEEE/ASME Transactions on Mechatronics, 14(6), 712–718. doi:10.1109/TMECH.2009.2032777.

[33] Huang, J., Guan, Z. H., Matsuno, T., Fukuda, T., & Sekiyama, K. (2010). Sliding-mode velocity control of mobile-wheeled inverted-pendulum systems. IEEE Transactions on Robotics, 26(4), 750–758. doi:10.1109/TRO.2010.2053732.

[34] Irfan, S., Mehmood, A., Razzaq, M. T., & Iqbal, J. (2018). Advanced sliding mode control techniques for Inverted Pendulum: Modelling and simulation. Engineering Science and Technology, an International Journal, 21(4), 753–759. doi:10.1016/j.jestch.2018.06.010.

[35] Kim, S., & Kwon, S. (2017). Nonlinear optimal control design for underactuated two-wheeled inverted pendulum mobile platform. IEEE/ASME Transactions on Mechatronics, 22(6), 2803–2808. doi:10.1109/TMECH.2017.2767085.

[36] Ma'arif, A., Vera, M. A. M., Mahmoud, M. S., Ladaci, S., Çakan, A., & Parada, J. N. (2022). Backstepping Sliding Mode Control for Inverted Pendulum System with Disturbance and Parameter Uncertainty. Journal of Robotics and Control (JRC), 3(1), 86–92. doi:10.18196/jrc.v3i1.12739.

[37] Huang, J., Zhang, M., Ri, S., Xiong, C., Li, Z., & Kang, Y. (2020). High-Order Disturbance-Observer-Based Sliding Mode Control for Mobile Wheeled Inverted Pendulum Systems. IEEE Transactions on Industrial Electronics, 67(3), 2030–2041. doi:10.1109/TIE.2019.2903778.

[38] Islam1, M. R., Hossain2, M. R. T., & Banik, S. C. (2021). Synchronizing of stabilizing platform mounted on a two-wheeled robot. Journal of Robotics and Control (JRC), 2(6), 552–558. doi:10.18196/jrc.26136.

[39] Van Lam, P., & Fujimoto, Y. (2019). A Robotic Cane for Balance Maintenance Assistance. IEEE Transactions on Industrial Informatics, 15(7), 3998–4009. doi:10.1109/TII.2019.2903893.

[40] Sahnehsaraei, M. A., & Mahmoodabadi, M. J. (2021). Approximate feedback linearization based optimal robust control for an inverted pendulum system with time-varying uncertainties. International Journal of Dynamics and Control, 9(1), 160–172. doi:10.1007/s40435-020-00651-w.

[41] Watson, M. T., Gladwin, D. T., Prescott, T. J., & Conran, S. O. (2019). Dual-Mode Model Predictive Control of an Omnidirectional Wheeled Inverted Pendulum. IEEE/ASME Transactions on Mechatronics, 24(6), 2964–2975. doi:10.1109/TMECH.2019.2943708.

[42] Iwendi, C., Alqarni, M. A., Anajemba, J. H., Alfakeeh, A. S., Zhang, Z., & Bashir, A. K. (2019). Robust Navigational Control of a Two-Wheeled Self-Balancing Robot in a Sensed Environment. IEEE Access, 7, 82337–82348. doi:10.1109/ACCESS.2019.2923916.

[43] Wu, L. F., & Li, T. H. S. (2020). Fuzzy dynamic gait pattern generation for real-time push recovery control of a teen-sized humanoid robot. IEEE Access, 8, 36441–36453. doi:10.1109/ACCESS.2020.2975041.

[44] Mahmoud, M. S., & Nasir, M. T. (2017). Robust control design of wheeled inverted pendulum assistant robot. IEEE/CAA Journal of Automatica Sinica, 4(4), 628–638. doi:10.1109/JAS.2017.7510613.

[45] Lin, L. G., & Xin, M. (2020). Nonlinear Control of Two-Wheeled Robot Based on Novel Analysis and Design of SDRE Scheme. IEEE Transactions on Control Systems Technology, 28(3), 1140–1148. doi:10.1109/TCST.2019.2899802.

[46] Akhond, S., Herzig, N., Wegiriya, H., & Nanayakkara, T. (2019). A method to guide local physical adaptations in a robot based on phase portraits. IEEE Access, 7, 78830–78841. doi:10.1109/ACCESS.2019.2923144.

[47] Ramos, J., & Kim, S. (2018). Dynamic bilateral teleoperation of the cart-pole: A study toward the synchronization of human operator and legged robot. IEEE Robotics and Automation Letters, 3(4), 3293–3299. doi:10.1109/LRA.2018.2852840.

[48] Iacob, C. G. (2020). Linear and angular position control of a custom built stepper motor driven self-balancing robot. 24th International Conference on System Theory, Control and Computing, ICSTCC 2020, Sinaia, Romania, 648–653. doi:10.1109/ICSTCC50638.2020.9259706.

[49] Yuan, Y., Li, Z., Zhao, T., & Gan, D. (2020). DMP-Based Motion Generation for a Walking Exoskeleton Robot Using Reinforcement Learning. IEEE Transactions on Industrial Electronics, 67(5), 3830–3839. doi:10.1109/TIE.2019.2916396.

[50] Carpentier, J., & Mansard, N. (2018). Multicontact Locomotion of Legged Robots. IEEE Transactions on Robotics, 34(6), 1441–1460. doi:10.1109/TRO.2018.2862902.

[51] Li, Z., Ren, Z., Zhao, K., Deng, C., & Feng, Y. (2020). Human-Cooperative Control Design of a Walking Exoskeleton for Body Weight Support. IEEE Transactions on Industrial Informatics, 16(5), 2985–2996. doi:10.1109/TII.2019.2900121.

[52] Puriel Gil, G., Yu, W., & Sossa, H. (2019). Reinforcement Learning Compensation based PD Control for a Double Inverted Pendulum. IEEE Latin America Transactions, 17(2), 323–329. doi:10.1109/TLA.2019.8863179.

[53] Huang, J., Ri, M., Wu, D., & Ri, S. (2018). Interval type-2 fuzzy logic modeling and control of a mobile two-wheeled inverted pendulum. IEEE Transactions on Fuzzy Systems, 26(4), 2030–2038. doi:10.1109/TFUZZ.2017.2760283.

[54] Bounemeur, A., & Chemachema, M. (2021). Adaptive Fuzzy Fault-Tolerant Control for a Class of Nonlinear Systems under Actuator Faults: Application to an Inverted Pendulum. International Journal of Robotics and Control Systems, 1(2), 102–115. doi:10.31763/ijrcs.v1i2.306.

[55] Gong, D., Wang, P., Zhao, S., Du, L., & Duan, Y. (2018). Bionic quadruped robot dynamic gait control strategy based on twenty degrees of freedom. IEEE/CAA Journal of Automatica Sinica, 5(1), 382–388. doi:10.1109/JAS.2017.7510790.

[56] Saad, M., Amhedb, A. H., & Al Sharqawi, M. (2021). Real time DC motor position control using PID controller in LabVIEW. Journal of Robotics and Control (JRC), 2(5), 342–348. doi:10.18196/jrc.25104.

[57] Handaya, D., & Fauziah, R. (2021). Proportional-integral-derivative and linear quadratic regulator control of direct current motor position using multi-turn based on LabView. Journal of Robotics and Control (JRC), 2(4), 332–336. doi:10.18196/jrc.24102.

[58] Kadry, S., & Rajinikanth, V. (2021). Design of PID Controller for Magnetic Levitation System using Harris Hawks Optimization. Jurnal Ilmiah Teknik Elektro Komputer Dan Informatika, 6(2), 70. doi:10.26555/jiteki.v6i2.19167.

[59] Syafeeq Lone, S., Zainul Azlan, N., & Kamarudzaman, N. (2021). Soft Pneumatic Exoskeleton for Wrist and Thumb Rehabilitation. International Journal of Robotics and Control Systems, 1(4), 440–452. doi:10.31763/ijrcs.v1i4.447.

[60] Hussein Mohammed Al-Almoodi, H., Zainul Azlan, N., Shahdad, I., & Kamarudzaman, N. (2021). Continuous Passive Motion Machine for Elbow Rehabilitation. International Journal of Robotics and Control Systems, 1(3), 402–415. doi:10.31763/ijrcs.v1i3.446.

[61] Jabeur, C. Ben, & Seddik, H. (2022). Optimized Neural Networks-PID Controller with Wind Rejection Strategy for a Quad-Rotor. Journal of Robotics and Control (JRC), 3(1), 62–72. doi:10.18196/jrc.v3i1.11660.

[62] Rodriguez-Abreo, O., Rodriguez-Resendiz, J., Fuentes-Silva, C., Hernandez-Alvarado, R., & Falcon, M. D. C. P. T. (2021). Self-Tuning Neural Network PID with Dynamic Response Control. IEEE Access, 9, 65206–65215. doi:10.1109/ACCESS.2021.3075452.

[63] Widya Suseno, E., & Ma'arif, A. (2021). Tuning of PID Controller Parameters with Genetic Algorithm Method on DC Motor. International Journal of Robotics and Control Systems, 1(1), 41–53. doi:10.31763/ijrcs.v1i1.249.

[64] Cabré, T. P., Vela, A. S., Ribes, M. T., Blanc, J. M., Pablo, J. R., & Sancho, F. C. (2021). Didactic platform for DC motor speed and position control in Z-plane. ISA Transactions, 118, 116–132. doi:10.1016/j.isatra.2021.02.020.

[65] Kristiyono, R., & Wiyono. (2021). Autotuning fuzzy PID controller for speed control of BLDC motor. Journal of Robotics and Control (JRC), 2(5), 400–407. doi:10.18196/jrc.25114.

[66] Maghfiroh, H., Ahmad, M., Ramelan, A., & Adriyanto, F. (2022). Fuzzy-PID in BLDC Motor Speed Control Using MATLAB/Simulink. Journal of Robotics and Control (JRC), 3(1), 8–13. doi:10.18196/jrc.v3i1.10964.

[67] Ekinci, S., & Hekimoglu, B. (2019). Improved Kidney-Inspired Algorithm Approach for Tuning of PID Controller in AVR System. IEEE Access, 7, 39935–39947. doi:10.1109/ACCESS.2019.2906980.

[68] Kashyap, A. K., & Parhi, D. R. (2021). Particle Swarm Optimization aided PID gait controller design for a humanoid robot. ISA Transactions, 114, 306–330. doi:10.1016/j.isatra.2020.12.033.

[69] Patel, V. V. (2020). Ziegler-Nichols Tuning Method: Understanding the PID Controller. Resonance, 25(10), 1385–1397. doi:10.1007/s12045-020-1058-z.

[70] Ogata, K. (2010). Modern control engineering (5th Ed.). Upper Saddle River, NJ: Prentice-Hall, New Jersey, United States.

[71] Ma'arif, A., Iswanto, I., Nuryono, A. A., & Alfian, R. I. (2019). Kalman Filter for Noise Reducer on Sensor Readings. Signal and Image Processing Letters, 1(2), 11–22. doi:10.31763/simple.v1i2.2.

[72] Alfian, R. I., Ma'Arif, A., & Sunardi, S. (2021). Noise reduction in the accelerometer and gyroscope sensor with the Kalman filter algorithm. Journal of Robotics and Control (JRC), 2(3), 180–189. doi:10.18196/jrc.2375.

[73] Qu, D., Zheng, Y., Guo, J., & Song, R. (2020). A Control Scheme for Single Legged Hopping Robot Based on Fuzzy PD Algorithm. Proceedings of 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference, ITNEC 2020, Itnec, 384–388. doi:10.1109/ITNEC48623.2020.9084676.

[74] Lakatos, D., Ploeger, K., Loeffl, F., Seidel, D., Schmidt, F., Gumpert, T., John, F., Bertram, T., & Albu-Schaffer, A. (2018). Dynamic Locomotion Gaits of a Compliantly Actuated Quadruped with SLIP-Like Articulated Legs Embodied in the Mechanical Design. IEEE Robotics and Automation Letters, 3(4), 3908–3915. doi:10.1109/LRA.2018.2857511.

[75] Tzorakoleftherakis, E., Ansari, A., Wilson, A., Schultz, J., & Murphey, T. D. (2016). Model-Based Reactive Control for Hybrid and High-Dimensional Robotic Systems. IEEE Robotics and Automation Letters, 1(1), 431–438. doi:10.1109/LRA.2016.2522078.

[76] Erkol, H. O. (2018). Optimal PIλ Dμ controller design for two wheeled inverted pendulum. IEEE Access, 6, 75709–75717. doi:10.1109/ACCESS.2018.2883504.

[77] Zaytsev, P., Wolfslag, W., & Ruina, A. (2018). The Boundaries of Walking Stability: Viability and Controllability of Simple Models. IEEE Transactions on Robotics, 34(2), 336–352. doi:10.1109/TRO.2017.2782818.

[78] Zhang, L., Ren, X., & Guo, Q. (2020). Balance Control of a Wheeled Hopping Robot. Chinese Control Conference, CCC, 2020-July, 3801–3805. doi:10.23919/CCC50068.2020.9189592.

[79] Dong, S., Yuan, Z., Yu, X., Zhang, J., Sadiq, M. T., & Zhang, F. (2019). On-Line Gait Adjustment for Humanoid Robot Robust Walking Based on Divergence Component of Motion. IEEE Access, 7, 159507–159518. doi:10.1109/ACCESS.2019.2949747.

[80] Mohan, A., Sivaprakasam, M., George, B., & Kumar, V. J. (2018). Self-balancing signal conditioning circuit for a floating-wiper resistive displacement sensor. IEEE Sensors Journal, 18(18), 7544–7550. doi:10.1109/JSEN.2018.2858824.

[81] Su, Y., Wang, T., Zhang, K., Yao, C., & Wang, Z. (2020). Adaptive Nonlinear Control Algorithm for a Self-Balancing Robot. IEEE Access, 8, 3751–3760. doi:10.1109/ACCESS.2019.2963110.

[82] Maghfiroh, H., & Santoso, H. P. (2021). Self-balancing robot navigation. Journal of Robotics and Control (JRC), 2(5), 408–412. doi:10.18196/jrc.25115.

[83] Borase, R. P., Maghade, D. K., Sondkar, S. Y., & Pawar, S. N. (2020). A review of PID control, tuning methods and applications. International Journal of Dynamics and Control, 9(2), 818–827. doi:10.1007/s40435-020-00665-4.

[84] Okelola, M. O., Aborisade, D. O., & Adewuyi, P. A. (2021). Performance and Configuration Analysis of Tracking Time Anti-Windup PID Controllers. Jurnal Ilmiah Teknik Elektro Komputer Dan Informatika, 6(2), 20. doi:10.26555/jiteki.v6i2.18867.

[85] Li, P., & Zhu, G. (2019). IMC-based PID control of servo motors with extended state observer. Mechatronics, 62(June), 102252. doi:10.1016/j.mechatronics.2019.102252.

[86] Sun, X., & Yan, G. (2018). Multi-sensor optimal weighted fusion incremental Kalman smoother. Journal of Systems Engineering and Electronics, 29(2), 262–268. doi:10.21629/JSEE.2018.02.06.

[87] Rong, H., Peng, C., Chen, Y., Zou, L., Zhu, Y., & Lv, J. (2018). Adaptive-Gain Regulation of Extended Kalman Filter for Use in Inertial and Magnetic Units Based on Hidden Markov Model. IEEE Sensors Journal, 18(7), 3016–3027. doi:10.1109/JSEN.2018.2806932.

[88] Wang, H., Li, H., Fang, J., & Wang, H. (2018). Robust Gaussian Kalman filter with outlier detection. IEEE Signal Processing Letters, 25(8), 1236–1240. doi:10.1109/LSP.2018.2851156.

[89] Zhang, Q., Yang, Y., Xiang, Q., He, Q., Zhou, Z., & Yao, Y. (2018). Noise Adaptive Kalman Filter for Joint Polarization Tracking and Channel Equalization Using Cascaded Covariance Matching. IEEE Photonics Journal, 10(1), 1–1. doi:10.1109/JPHOT.2018.2797050.

[90] Zahraoui, Y., Akherraz, M., & Ma'arif, A. (2021). A Comparative Study of Nonlinear Control Schemes for Induction Motor Operation Improvement. International Journal of Robotics and Control Systems, 2(1), 1–17. doi:10.31763/ijrcs.v2i1.521.

[91] Zhao, S., Shmaliy, Y. S., Ahn, C. K., & Liu, F. (2018). Adaptive-Horizon Iterative UFIR Filtering Algorithm with Applications. IEEE Transactions on Industrial Electronics, 65(8), 6393–6402. doi:10.1109/TIE.2017.2784405.

[92] Dionelis, N., & Brookes, M. (2018). Phase-aware single-channel speech enhancement with modulation-domain Kalman filtering. IEEE/ACM Transactions on Audio Speech and Language Processing, 26(5), 937–950. doi:10.1109/TASLP.2018.2800525.

[93] Luo, J., & Qin, S. (2018). A Fast Algorithm of SLAM Based on Combinatorial Interval Filters. IEEE Access, 6, 28174–28192. doi:10.1109/ACCESS.2018.2838112.

[94] Rahmaniar, W., & Rakhmania, A. E. (2021). Online digital image stabilization for an unmanned aerial vehicle (UAV). Journal of Robotics and Control (JRC), 2(4), 234–239. doi:10.18196/jrc.2484.

[95] Yin, L., Deng, Z., Huo, B., Xia, Y., & Li, C. (2018). Robust Derivative Unscented Kalman Filter under Non-Gaussian Noise. IEEE Access, 6, 33129–33136. doi:10.1109/ACCESS.2018.2846752.