ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

KANDÓ KÁLMÁN FACULTY OF ELECTRICAL ENGINEERING
ELECTRONIC AND COMMUNICATION SYSTEMS INSTITUTE
INSTRUMENTATION AND AUTOMATION DEPARTMENT

THESIS WORK

USES OF MOBILE ROBOTS FOR EDUCATIONAL PURPOSES

OE-KVK Student's Name **Ali Amina Hassan-**[redacted]

Registration number: T009071/FI12904/K

2023/2024

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Óbuda University
Kandó Kálmán Faculty of Electrical Engineering

Institute of Instrumentation and Automation

THESIS ASSIGNMENT

Student's surname, forename (s): Amina Hassan Ali
Thesis number: SZD2309040923441238[redacted]
Registration number: T009071/FI12904/K **Neptun code:** [redacted]

Branch of study:
Specialization: Instrumentation and automation specialisation

Thesis title: Uses of mobile robot for educational purpose
Task description: The thesis will ideally explain the uses of different commercially available cost-effective mobile robot kits in educational environments (secondary school and university). The goal for the users of this kits is to discover and test how a robot can navigate its environment, making decisions and execute some simple tasks. Besides that, it will be shown how these sets can be used to introduce basic engineering concepts such as control theory or PID compensation.

Institutional consultant's name: Árpád Varga

The limitation period of the theme issued: 31.12.2025.
Deadline for submission of Thesis: 15.12.2023.

The thesis is: Not secreted.

Issued: Budapest, 24.10.2023.

P.H ....
Director of Institute

The Thesis is suitable for submission:

Varga Árpád
....
Institutional consultant

....
External consultant

STUDENT STATEMENT

I hereby declare that the present final thesis paper is solely my own work and that the references (written and electronic literature and tools used during the process) have been acknowledged and fully cited. I hereby grant the university and the task assigning institution permission to use free of charge the final results of this paper for their own purposes, abiding by the pertaining rules of encryption.

Budapest, 2023, December 15

....
Signature of student

Acknowledgement

With heartfelt gratitude, I would like to begin by expressing my gratitude to Almighty God for granting me the strength and wisdom to embark on and complete this research thesis.

I extend my sincere appreciation to all the individuals who played a pivotal role in the successful completion of this research thesis. First, my gratitude goes to my supervisor, Mr. Arpad Varga, whose invaluable guidance, support, and insightful feedback were instrumental in shaping the direction and scope of this study. Their expertise has enriched the research process and contributed significantly to its success.

I am deeply thankful to educators, researchers, and industry professionals who generously shared their time, knowledge, and perspectives, thus enhancing the depth and breadth of this study. Their contributions have greatly enriched the findings and recommendations, making this study more comprehensive and insightful.

Acknowledgments are also due to the authors of relevant literature and studies that laid the foundation for this research. Their groundbreaking work and insights have been crucial in shaping our understanding of mobile robots in educational settings.

Finally, I express my heartfelt appreciation to my family and friends for their unwavering support and encouragement throughout this research journey. Their belief in me has been a constant source of motivation and I am truly grateful for their presence in my life.

While it is not feasible to mention everyone individually, please know that your contributions, whether big or small, are deeply valued and have played a significant role in the successful completion of this research. Thank you all.

Abstract

Introduction: The incorporation of mobile robots into STEM education is gaining prominence. This study addresses the integration of mobile robots in educational settings, specifically focusing on Science, Technology, Engineering, and Mathematics (STEM) education. The aim of this thesis is to set the parameters of the proportional, integral, and derivative (PID) controller tuner in MATLAB for optimal control outcomes and to assess the overall effectiveness of mobile robots for educational purposes.

Methods and Materials: This research utilized a comprehensive methodology, including the comparison and evaluation of various mobile robots based on compatibility, features, and programming languages. It involves fine-tuning the PID controller parameters in MATLAB to optimize the control outcomes. The methodology assessed the overall effectiveness of mobile robots in enhancing educational outcomes.

Results: This study suggests that mobile robots have the potential to significantly enhance STEM education. After comparing various robots, MBOT Ranger was found to be the best option, with a considerable 38.6% advantage in terms of compatibility, affordability, design, programming language support, and features/power source. The implementation of the PID controller in MATLAB improved the performance in terms of the distance measurement accuracy and stability, particularly in situations where holding and balancing are required.

Conclusion: The research found that mobile robots in education improve STEM education and critical thinking. To choose the right robot, consider features, programming languages, and long-term impacts. Manufacturers and developers should keep innovating to meet educators' and students' needs. Educators should carefully assess and adopt robots to enhance STEM education, focusing on optimizing control outcomes with PID controllers in MATLAB.

Table of Contents

List of tables

List of Figures

INTRODUCTION

1.1 Background

In recent years, mobile robots have gained significant attention for their potential use in educational settings, particularly in the fields of Science, Technology, Engineering, and Mathematics (STEM) education. These robots offer a distinctive and interactive learning experience, fostering student engagement with complex concepts and the development of critical thinking skills. However, to unlock the full potential of mobile robots in education, it is imperative to acknowledge existing challenges and explore innovative approaches to enhance their effectiveness[1].

The integration of mobile robots into educational settings presents several challenges, with a primary focus on three critical aspects: The need for standardized guidelines and frameworks for seamlessly incorporating mobile robots into the curriculum [2]. Educators grapple with aligning the use of robots with learning objectives and assessing their impact on students' academic performance. The high cost of mobile robots and their limited availability in educational institutions further hinder their widespread adoption. Moreover, there is a pressing need to address the training and support required for educators to effectively utilize mobile robots in classrooms.

The lack of consensus regarding the most effective programming languages and platforms for teaching robotics is another significant challenge. Different robots support various programming languages, such as Scratch, Python, or Arduino C, making it challenging for educators to select the most suitable option. Additionally, the limited availability of comprehensive educational resources and lesson plans for mobile robots restricts educators' ability to integrate them effectively into their teaching[3]

To fine-tune the Proportional, Integral, and Derivative (PID) controller in MATLAB to achieve the desired control outcomes is a crucial aspect that demands attention. The optimization of control outcomes through MATLAB is essential for ensuring a seamless learning experience with mobile robots. Existing studies have aptly identified these challenges and proposed insightful solutions. Hudson [2] highlighted the need for clear guidelines and pedagogical frameworks for integrating mobile robots into educational settings, emphasizing the alignment of robot activities with curriculum goals. Researchers such as Kargar Tasooji and Marquez [4] have discussed the importance of providing educators with comprehensive training in programming languages and platforms to enhance their confidence in using mobile robots. They emphasize the need for the development of standardized educational resources that align with curriculum standards. Moreover, Fumagalli, Huber, Dymczyk, Siegwart and Dubé [5] explored the effectiveness of different programming languages in teaching robotics, suggesting the integration of multiple languages to cater to diverse student needs.

The primary objective of this study is to explore and evaluate the effectiveness of this collaborative network approach in enhancing the utilization of mobile robots for educational purposes. By addressing existing challenges and providing educators with the necessary resources and support, this approach aims to promote the widespread adoption of mobile robots in education and improve students' learning outcomes in STEM subjects. The emphasis on fine-tuning the PID controller in MATLAB aligns with the broader goal of ensuring optimal control outcomes for a holistic and effective learning experience.

1.2 Problem Statement

The integration of mobile robots into education presents numerous challenges and opportunities that require thorough exploration and understanding. Educators and researchers must understand the potential benefits and limitations of using mobile robots for educational purposes to make informed decisions and effectively address these challenges[6-8]. The investigation involves a multifaceted approach, encompassing the comparison of different mobile robots used in educational settings, the evaluation of their compatibility, features, and programming languages, the assessment of the overall effectiveness of utilizing mobile robots for educational purposes[5, 9-12], and the nuanced task of fine-tuning the Proportional, Integral, and Derivative(PID) controller in MATLAB to achieve the desired control outcomes[13, 14]. The research seeks to gain a profound understanding of the imperative need for comprehensive training and support for educators, and the intricate process of fine-tuning the PID controller in MATLAB.

1.3 Research Objectives

1.3.1 Main objective

The main objective of this thesis is to improve educational practices by comparing various mobile robots, refining the PID controller in MATLAB, and evaluating the overall utility of mobile robots in educational environments.

1.3.2 Specific objectives

i) To compare the different mobile robots used in educational settings.
ii) To fine-tun the PID controller in MATLAB to achieve the desired control outcomes.
iii) To assess the overall effectiveness of mobile robots for educational purposes.

1.3.3 Research questions

i) How do different mobile robots compare in terms of their compatibility, features, and programming languages?
ii) How can the PID controller of the MATLAB be fine turned to achieve the desired control outcomes?
iii) How effective are mobile robots for enhancing STEM education and fostering critical thinking skills?

1.4 Significance of the research

This study holds significance in advancing the current understanding of the use of mobile robots as educational tools, aligning with specific objectives. By systematically comparing various mobile robots and fine-tuning PID controllers in MATLAB, the research aims to provide educators and researchers with valuable insights into the effectiveness of these technologies. This will empower decision-makers to make informed choices when selecting and implementing mobile robot programs in educational settings.

1.5 Scope of the research

This study investigates the use of mobile robots as instrumental tools for fostering educational development, with a focused emphasis on STEM education. Aligned with specific objectives, the research delves into the comparison of various mobile robots used in educational settings, scrutinizing their conformity, attributes, and programming languages. Furthermore, the study extends its scope to include the fine-tuning of PID controllers in MATLAB to achieve desired control outcomes, a crucial aspect for optimizing the performance of mobile robots in educational environments.

LITERATURE REVIEW

2.1 Introduction

This literature review provides an overview of previous studies and research related to the use of mobile robots in educational settings. It explores the definitions of the key terms and establishes a foundation for understanding the concepts and applications of mobile robots for educational purposes.

2.2 Definitions of key terms

Table 1:Definitions of key terms

Keyword	Explanations
Mobile Robots	Robotic devices designed to move and operate autonomously or semi-autonomously[4-6].
Educational Purposes	Utilization of resources and tools for learning, teaching, and educational activities[10, 12, 15-19].
PID	A PID (Proportional – Integral – Derivative) controller is an instrument used by control engineers to regulate temperature, flow, pressure, speed, and other process variables in industrial control systems[13].

2.3 Theoretical literature review

2.3.1 A Comparative Analysis of Mobile Robotics in Education

Mobile robotics in education has gained reputation as an effective tool for engaging students in STEM (Science, Technology, Engineering, and Mathematics) subjects. This literature review aims to compare the features of four prominent mobile robots used in educational settings: MBOT Ranger, LEGO Mindstorms EV3, Arduino Robot, and Ozobot Evo. The analysis was based on several key factors, including compatibility, price, configuration and design, programming language use, features, and power sources[17].

Table 2:Comparison of the different Mobile robotic in education use purposes

NAME OF THE ROBOT	Compatibility	Price	Configuration and Design	Programming language use	Features And power source
MBOT RANGER [17]	It is fully compatible with mBlock which used for the scratching the graphical programming	It is cheap and more affordable.	It is designed in the three-in-one assembly design of the robot.	Use the block based graphical programming (scratch), Arduino C, Python and Matlab.	It is flexible and included with various sensors and module with power rechargeable batteries.
LEGO MINDSTORMS EVE3 [10, 11, 17, 20-23]	It is compact with EV3 bricks Mindstorm Firmware	It is very expensive but offer the extensive ability in uses.	It is free building design which can build variety robots	Python programming, and scratch.	It is included with various sensors with multiple motors and has AA batteries type.
ARDUINO ROBOT [17, 20, 22]	Has compact with visual programming which is called Blocky.	It has a low level of effective cost and affordability	It is flexible depend upon the kit or component used.	Uses an Arduino IDE and C/C++, scratch and Python	It has communication module; actuator and its power source depend on the work chosen.
OZOBOT EVO ROBOT [17]	It is compact with an Oz blocky and colour code for programming	It is cheap and affordable,	It is one way designing with precise colour code design.	It is used an evo blocky program as same as scratch.	It is merged with sensors, colour codding and rechargeable batteries source.

SPHERO ROBOT [17]	It is compatible with Edu application software	Expensive, it varies according to the robotic functions and features.	Has less configuration in physical form.	It is Edu application for the scratch and can be programmed by java script.	It is design with spherical design, come with various sensor and rechargeable batteries power source

2.4 Selection of the best mobile robots for education use

The comparison analysis presented in this study involves two main components: priority trade-offs and pairwise comparisons of the options for each criterion.

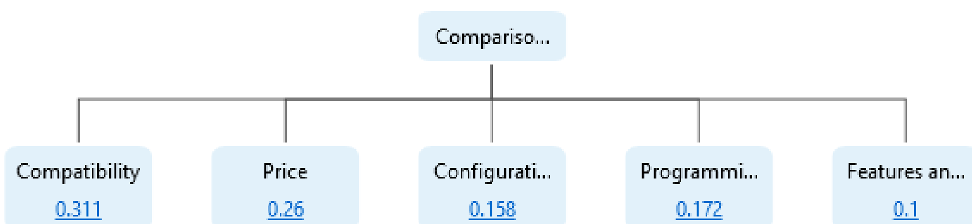

Figure 1:Comparison Analysis in tree diagram

2.5 Priority Trade-offs

The initial step involves comparing the importance of each criterion against one another. The results revealed the prioritization of criteria in the following order: Compatibility, Price, Configuration and Design, Programming Language Use, and Features and Power Source. The consistency ratio was calculated as 0.085, indicating a satisfactory level of consistency in the judgments[24-26].

Table 3:Priority Trade-offs of criteria

	Compatibility	Price	Configuration and Design	Programming language use	Features and Power Source	Priorities
Compatibility	1	2	2	2	2	0.310662671
Price	0.5	1	2	3	2	0.25974692
Configuration and Design	0.5	0.5	1	0.5	3	0.157968698
Programming language use	0.5	0.333333333	2	1	2	0.171834832
Features and Power Source	0.5	0.5	0.333333333	0.5	1	0.09978688
*** Consistency Ratio calculated as**	**0.085**					

2.6 Pairwise Comparisons

The subsequent phase involved comparing options within each criterion, utilizing a scale from 1 to 9. The priorities derived from these comparisons provide the overall ranking of mobile robotics platforms.

Compatibility: The priorities for mobile robots (MBOT Ranger, LEGO Mindstorms EV3, Arduino Robot, Ozobot Evo Robot, and Sphero Robot) were calculated, indicating MBOT Ranger as the preferred choice.

Price: The priorities for price comparisons place the MBOT Ranger at the top, closely followed by LEGO Mindstorms EV3.

Configuration and Design: MBOT anger emerges as the preferred choice in terms of configuration and design.

Programming Language Use: MBOT Ranger maintains its lead in programming language use.

Features and Power Source: MBOT Ranger again leads to features and power source considerations.

Multi-Criteria Utility Function and Metrics

This study employs a multi-criteria utility function to amalgamate the priorities assigned to each criterion. The resulting metrics indicate the overall ranking of mobile robotics platforms. The MBOT Ranger emerged as the preferred option, followed by the LEGO Mindstorms EV3, Arduino Robot, Ozobot Evo Robot, and Sphero Robot.

Table 4:Multi-Criteria Utility Function and Metrics

Option Name	Priorities
MBOT RANGER	0.386
LEGO MINDSTORMS EVE3	0.212
ARDUINO ROBOT	0.136
OZOBOT EVO ROBOT	0.174
SPHERO ROBOT	0.091

The bar graph depicts the priorities assigned to various mobile robotics platforms based on the critical criteria, with taller bars indicating higher preferences. MBOT Ranger emerged as the top choice, with a substantial 38.6%, showcasing its exceptional standing in compatibility, affordability, design, programming language support, and features/power source. LEGO Mindstorms EVE3 follows closely at 21.2%, displaying strong educational capabilities. The Arduino Robot secures the third position at 13.6%, noted for its affordability and flexibility. The Ozobot Evo Robot claims a moderate 17.4%, while the Sphero Robot holds the lowest priority at 9.1%. In essence, the MBOT Ranger is the most suitable mobile robotics platform for educational contexts, providing valuable guidance for educators and institutions in making informed decisions regarding robotic integration into educational programs.

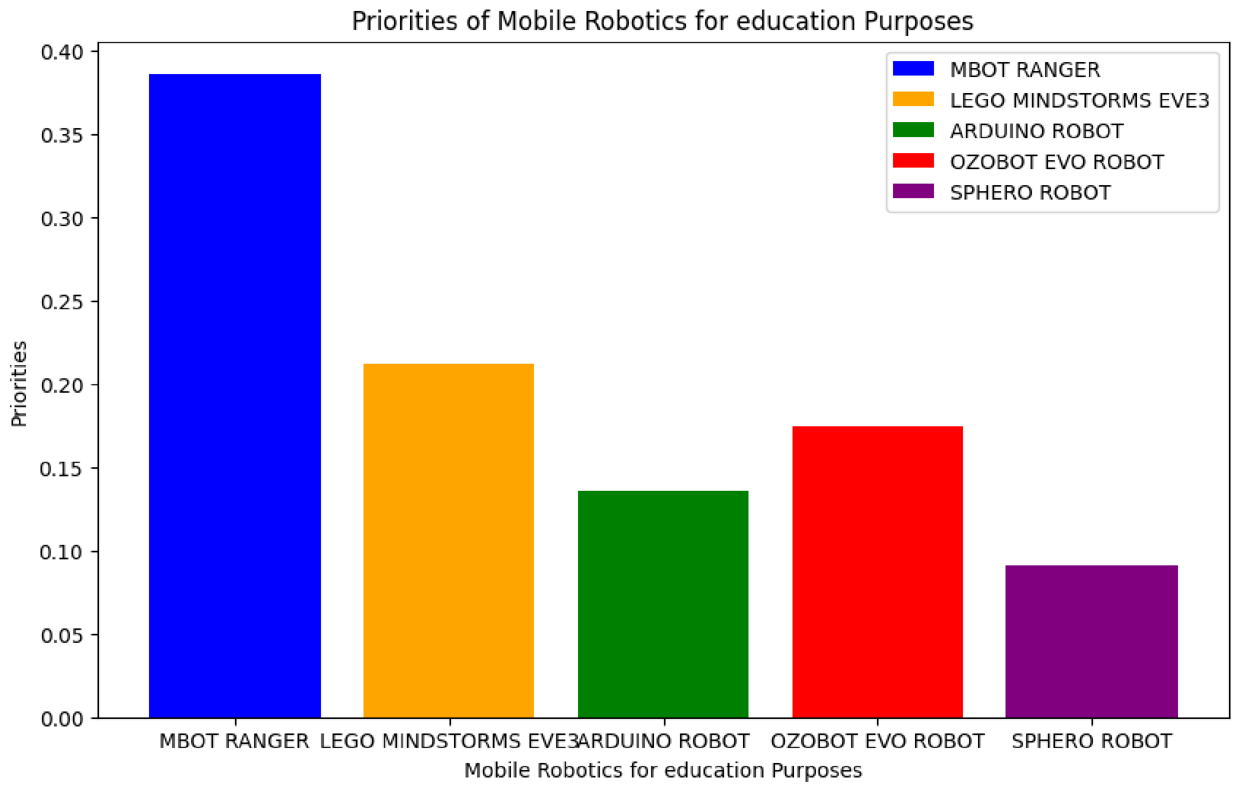

Figure 2:Priorities assigned to various mobile robotics platforms.

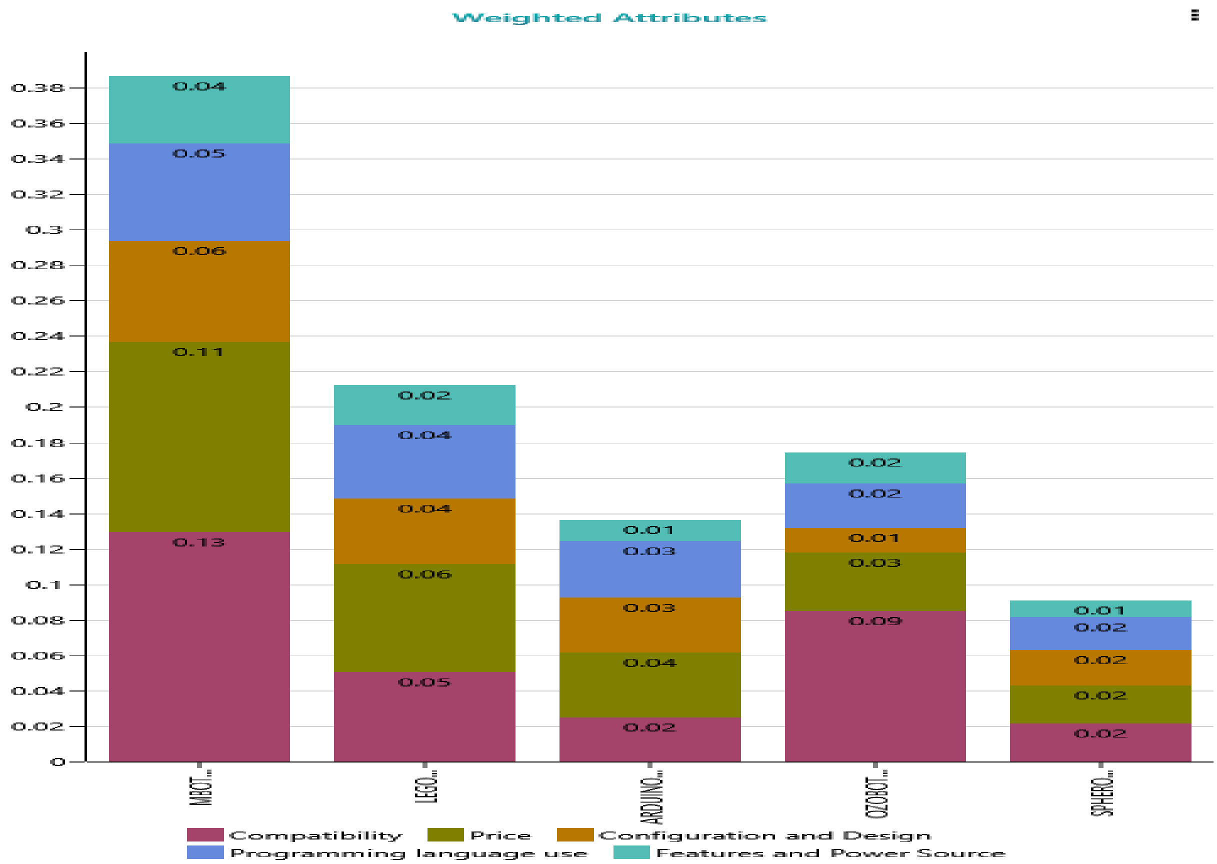

Figure 3: Weighted Attributes

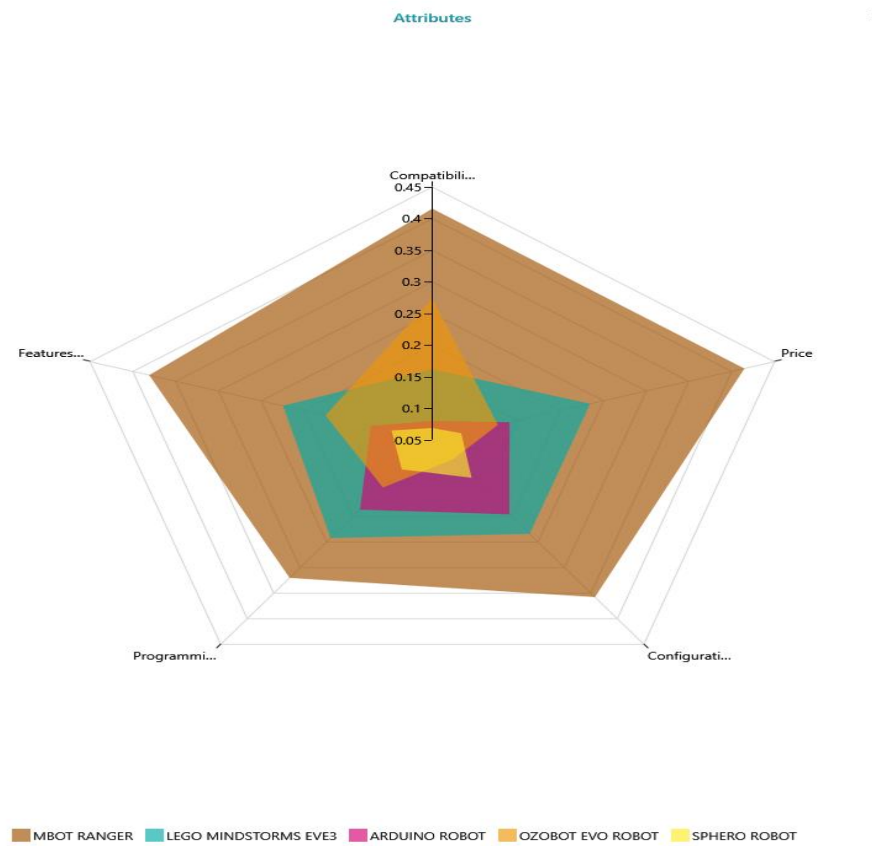

Figure 4:Web of Attributes

Through comparison analysis, the systematic comparison and prioritization of mobile robotics platforms illuminate MBOT Ranger as a standout choice for educational contexts. Its consistent lead across compatibility, price, design, programming language support, features, and power source makes it an ideal candidate for educators seeking an effective and versatile robotic solution. This structured decision-making approach, bolstered by comparison, offers valuable guidance to educators and institutions aiming to improve STEM education through mobile robotics integration.

2.7 A PID (Proportional – Integral – Derivative) Controller

The Proportional Integral and Derivative (PID) controller serves as a closed-loop control system extensively utilized across diverse industries. This controller manages process field variables by responding to measurements, finding widespread application in industrial control systems for overseeing parameters such as speed, temperature, and pressure. Its versatility extends to various domains, including process control, motor drives, magnetic and optic memories, automotive, flight control, and instrumentation. By integrating the proportional, integral, and derivative components, the PID[14] controller calculates the output signal based on the process response variable. The primary goal is to minimize the discrepancy between the output and the desired setpoint of a system, making it an effective tool for precision control in various applications[13].

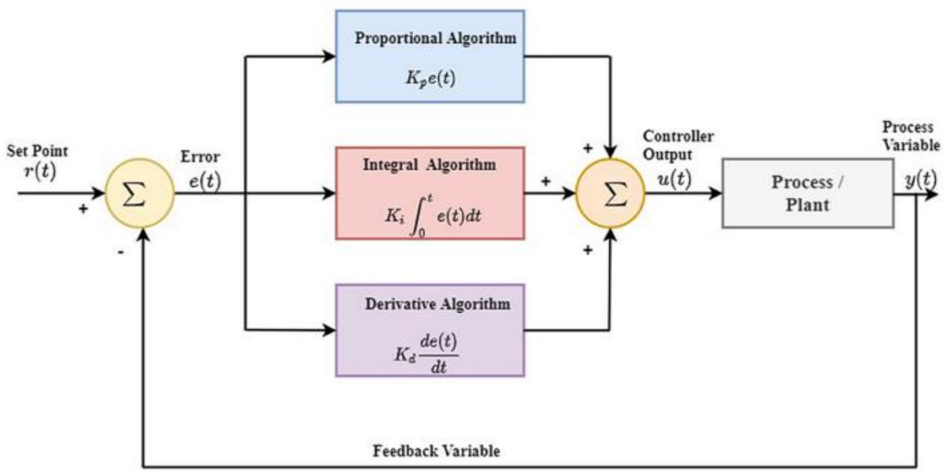

Figure 5:Block diagram of process control using PID[13].

The Figure 5 illustrates the configuration of the PID controller, comprising various parameters. Starting from the controller input, denoted as the reference [r(t)] or setpoint, a comparator, driven by the reference input element, calculates the error signal [e(t)]. This error signal, defined as the difference between the reference signal and the system output signal [y(t)], is expressed mathematically as e(t) = r(t) - y(t). The loop filter, housing three parameters, optimizes this error signal. The output of the system is then generated from the control signal (u[t]), initiating the closed-loop control operation. Following operational steps continuously reduces the error, with the controller aiming to minimize it over time, relying on the adjustment of the integral, derivative, and proportional parameters.

2.7.1 Steps in Designing the Control Loop

i) Determining the Transfer Function

The initial step involves determining the transfer function, which approximates the model of the system concerning the transfer function. Before delving into the design specifics, it is crucial to identify the plant in the process field that requires the design of a compensator block. The knowledge of the process field may not always be available, so in the zero step of designing, finding the approximate transfer function becomes essential, representing the behavior of the process field.

ii) Choosing the PID Configuration

Several channels, such as P, PI, PIDT1, or others, may be applicable, depending on the application and system. The choice of the channel to be activated in the PID compensator block is determined by the desired function. Additionally, selecting the structure of the channel depends on whether the American or European structure is preferred. While the design choice is not paramount, the crucial decision lies in selecting the active channel within these PID controller structures.

iii) Setting Parameters Inside the Compensator Block

The final step involves configuring the parameters within the compensator block, fine-tuning the PID controller to achieve the desired control outcomes. This step is integral to ensuring the effective performance of the control loop in regulating the system variables based on the setpoint. These parameters can be the Proportional coefficient control of the P block, the Integral time constant of the I block, the TD derivative time constant inside of the DT1 block, first order time constant inside of the DT1 block.

iv) Check the stability requirement.

It is known that stability is very important in designing the system of PID compensator block. It is decided upon the parameters of the compensator block where the simulation can be done to prevent the system from destruction or blowing out. It is good to check the stability of the system to prevent this kind of problem from happening in the system or process field.

v) Checking the quality measurement requirement.

If the system is stable then quality should be applicable where the given process should check if the system shows overshoot to the plant but should be less than 5% (<5%) which is good for the quality of the system measurement nevertheless if it shows the 10% the quality measurement required is not good which give the consequence to redo the process and do another tuning for the PID compensator.

2.7.2 Realizing of the Integral Channel

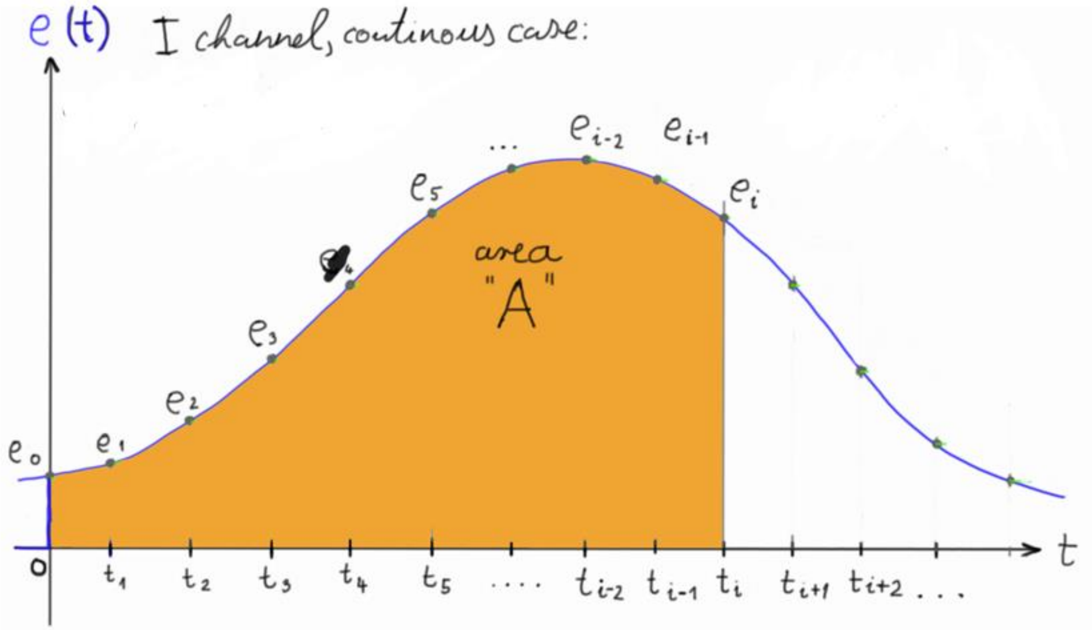

Figure 6:Integral channel graph

The output of the integral channel is denoted by.

$$y_{I_i}(t_i) = \frac{1}{T_I} \int_{0}^{ti} e(t)\, dt \qquad (1)$$

This is the output of the digitalize integral channel at the (t_i) time instant. It is an integral if we have the continuous signal, the first need to integral the error signal from zero time instant (t_i) time instant and multiply with $\frac{1}{T_I}$ for the continuous case. In addition, we must estimate the continuous situation using a method that can be easily elaborated by calculation, which can be performed by adding, subtracting, and multiplying the digital values.

Having the continuous signal the integral (e_i) up to the (t_i) time instant is the area the plot of the actual function as it can be seen from the Figure 7 the yellow area is the integral value which also need to multiply by $\frac{1}{T_I}$. The simple way to approximate the value of the denoted yellow area using the incoming digital error signal which are (e_0), (e_1) and some other error values is to use the property of zero order value[27].

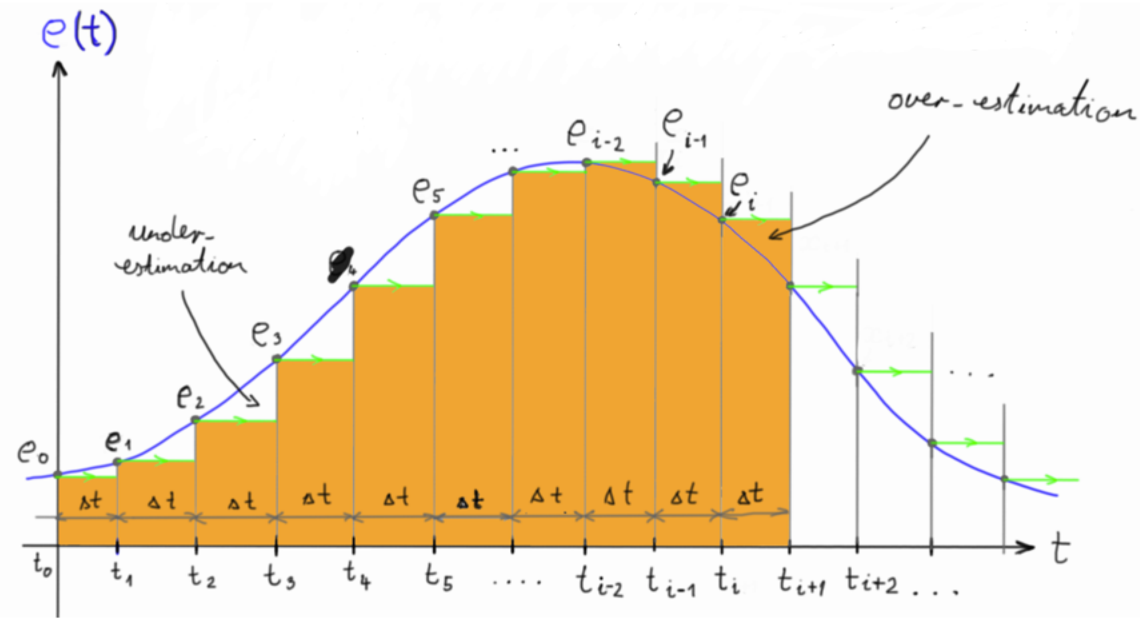

Figure 7:Derivative channel graph

However, they can be considered as error values (e) as they are kept constant up to the next incoming error value, and for that, a small rectangle can be built from the error values. It can be seen in the Figure 7 there are small rectangle which has boarder aside with the length of Δt and the other length is the side of the (e_0) similarly the dimension length of the small rectangle is Δt and the height is (e_1) value and can continue in this way as in the Figure 7 shown. It can be estimated that the area Figure 7 the plot of the function signal is the sum of rectangles. Of course, it will not be the perfect solution because the actual area of the rectangle will underestimate the precise area in Figure 7 the plot and the other side of the actual area of the $e_1 - 2$ and $e_i - 3$ will be overestimated the plot but somehow the over estimation and underestimate will be balance by each other. The integral of the e(t) signal can be approximate as the sum of the small area of the rectangle which is $e_i \Delta t$ where the e_i is the actual incoming error value and then the calculation of the sum of the rectangle up to give time instant must be done. The sampling time is the constant we can put the sampling time before the sum notation which will result the actual value of the integral will be Δt as the sum of the incoming error values up to a specific $t(i)$. The eq.(2) can be used to increase the precision of the estimation.

$$y_{I_i}(t_i) = \Delta t \cdot \frac{1}{T_I} \cdot \sum_{n=0}^{i} e_n \tag{2}$$

The precision of the integral channel can be improved in two ways:

1) Decreasing the Δt sampling time.

If we increase the Δt sampling time, the precision of the estimated of the integral value will be better. It can be seen in the Figure 8 it is used the smaller Δt value, the sum of the rectangle will be much closer to the original area as in   Figure 8 the graph.

If $\Delta t \rightarrow 0$

$$= \Delta t \cdot \sum_{n=0}^{i} e_n \tag{3}$$

$$= \int_{0}^{ti} e(t)dt \tag{4}$$

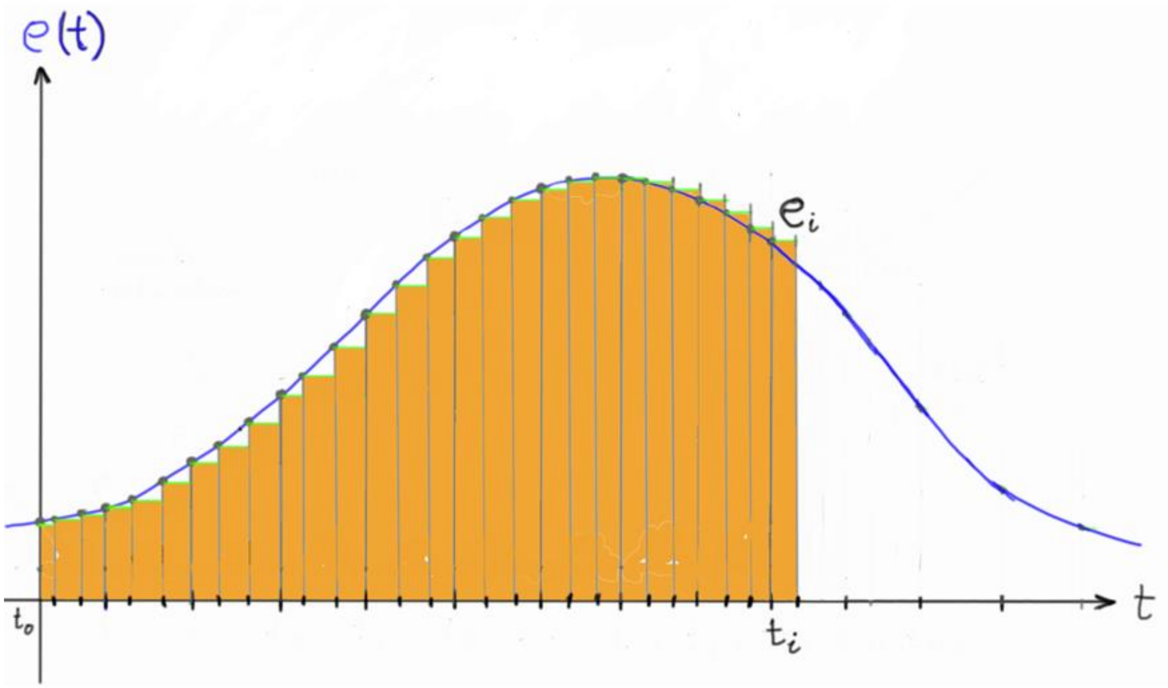

Figure 8:Integral channel for sampling time

2) Numerical Integration Estimation [28].

This numerical estimation is known as the trapezoidal rule. In this case the area as shown Figure 9 in the function up to $t(i)$ time instant can be calculated as the sum of the area of the small trapezoid. The area of trapezoid can be calculated by the following eq.(5),

$$Area\ of\ trapezoid = h\left(\frac{a+b}{2}\right) \tag{5}$$

Due to the formular eq. (5) it can be then easily to apply for the territory when area A_i of the trapezoid can be deal with the value of the area of the actual signal (e_1) and $e_1 - 1$ as the area of the previous signal which will be need to calculate and stored the area of each trapezoid, if we have small Δt there is no significant difference between the rectangular methos and trapezoid method.

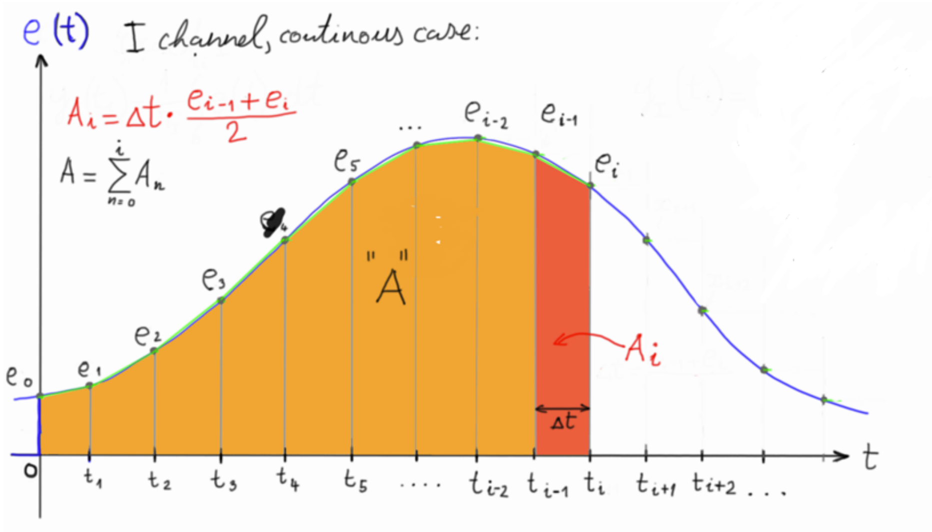

Figure 9:Integral channel for the numerical trapezoid rule

However, the integral signal is also corrupted by noise, but because of the approximation of the integral signal, it contains some territory where the error values will balance each other. In some cases, because of noise, it will have many territories to the rectangle, but in some cases, it will have a small number of territories, but for a long time, there will be a balance between them.

2.7.3 Realization of Derivative Channel

In this case, the first step is to determine how to realize the approximate value of the derivative channel using the discrete value of the error signal value. The exact definition of the derivative is shown in eq.(6) mathematical form:

$$\frac{de(t)}{dt} = \lim_{\Delta t \to 0} \frac{e(t + \Delta t) - e(t)}{\Delta t} \tag{6}$$

Where e is the error signal, for the limit value will give the precise error value when Δt sampling time tends to zero. If there is a very small enough Δt of the sampling time value, the formular in eq.(6) can be simplify as shown in eq.(7).

$$\left.\frac{de(t)}{\mathrm{dt}}\right|_{t_i} \approx \frac{e_{i-}e_{i-1}}{\Delta t} = \frac{\Delta e}{\Delta t} \tag{7}$$

Where.

e_i is the actual error value.

e_{i-1} where denotes the previous error.

Δt sample value.

t is the time.

Δe is the variation in the error.

It can also be used to visualize the value of the derivative by drawing a slope of the intersection between two points, the higher the slip in the slope in the derivative value. If the slope is at a positive angle, then we will have a positive derivative value, and if the slope has a negative angle, then we will have a negative derivative. In this case what has been drawn to the figure shows the negative derivative value because the previous error value is larger than the current one therefore, the Δe is the negative number and the slope angle relative to the horizontal line can be calculated by

$$\propto = arctg\,\frac{\Delta e}{\Delta t} \tag{8}$$

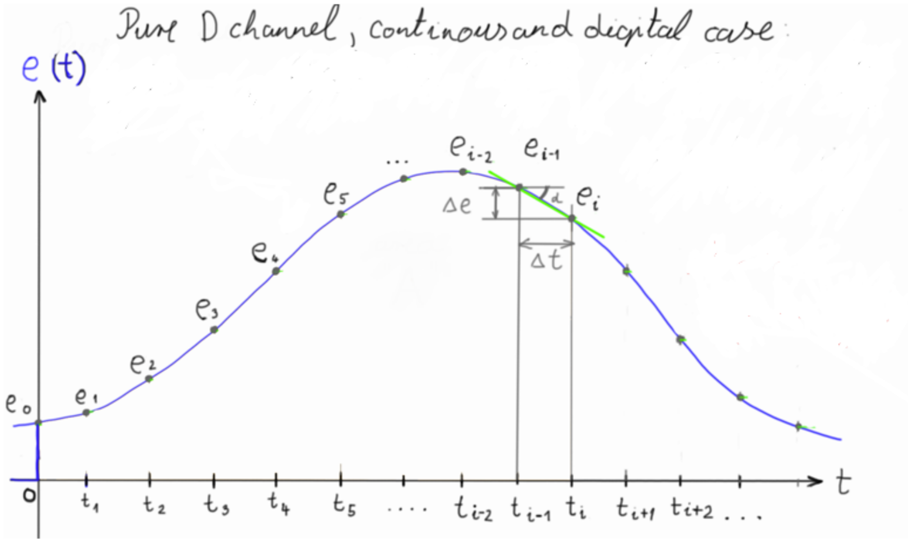

Figure 10:Realization of derivative channel graph

2.7.4 Realization of the First Order T1 Channel

It is crucial to perform a numerical analysis of the first-order channel because of the presence of measurement noise in real-world scenarios. This noise affects the measurement signal and generates an error signal, which is the difference between the reference and measurement signals. In certain real-world measurements, the error signal may exhibit rapid and erratic behavior with small oscillations relative to the original signal, rather than a smooth curve shape. The Figure 11 shows that the blue and green colors represent the original and true signals, respectively. We can visualize the value of the actual derivative by connecting the previous value which are e_i_1 *and* e_i The derivative value, which is determined by the noise signal, can be graphically represented. There is a considerable difference between the derivative value, which is calculated based on the original or true signal, and that which is calculated based on the noise signal. The pure derivation channel will be very sensitive to noise as the derivative calculation is based on the previous value and actual value; they will cause a large disturbance relative to the original or pure derivative value. Therefore, we need something to filter out the noise from the measured signal, and for this purpose, we can use a filter.

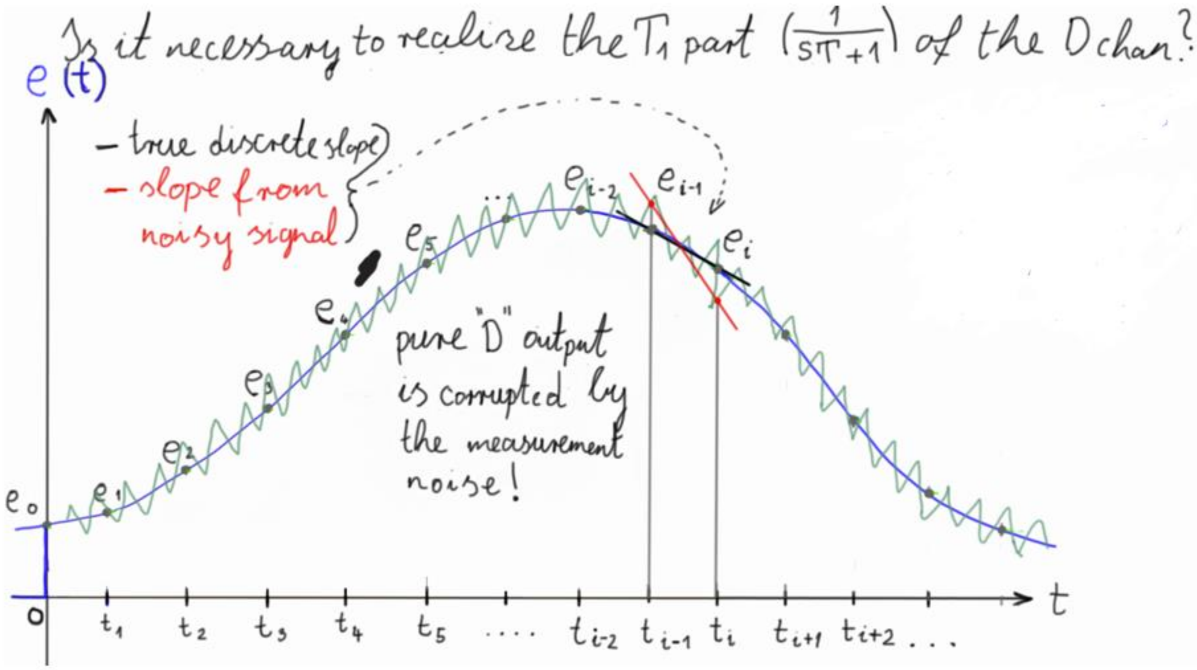

Figure 11:Realization of the first-order channel graph.

2.8 System Identification

A technique for constructing mathematical representations of dynamic systems based on observable data related to the system's input and output signals is proposed. Instead of relying on mathematical computations, this method involves deriving a dynamic system model from empirical data. To complete the system identification process, it is imperative to measure both input and output signals in either the time or frequency domain. By introducing input signals, which subsequently influence the output, we establish connections between the input and output data. Assuming the capability to sufficiently excite all crucial dynamics of the system, a mathematical model can be derived from the data, mirroring the observed relationship. The selection of an appropriate model structure is necessary, representing the behaviour under observation, followed by the adjustment of model parameters to best fit the data. Accurate system identification is crucial for predicting the system's response to various stimulation in the context of designing a control model. Robotics, signal processing, and control systems can all leverage the advantages of system identification. Effective experimental design is integral to generating informative data for model fitting, simplifying the representation of real systems. Noteworthy benefits of these models include formal analysis using models to ensure performance or safety, integration of model predictions with sensor measurements to enhance state estimation or predict maintenance requirements and replacement of physical systems and environments through simulation, with the possibility of simplifying models for faster simulations. However, regardless of simplification, a model must accurately capture the essential system features relevant to its intended application. The system identification has two types of the model approach which are explaining below.

2.8.1 Black Box Model

This approach involves constructing a model based on the step response of the transfer function, specifically for an input step function. The conceptual idea is akin to dealing with a black box, where the internal workings are unknown or invisible. This signifies a complete lack of knowledge about the system, and the aim is to identify the system solely through input and output data. Unlike traditional methods relying on first principles or physical modelling programs, this approach operates under the premise of minimal prior information about the system dynamics.

Practically, it implies a scenario where there is no preliminary understanding of the process field, but measurements are feasible for both the input and output of the process field, whether it's an actuator, transducer, or plant. The emphasis lies on having measurement data without any a priori knowledge about the system. The objective is to derive a mathematical model that aligns well with the measured data, without delving into the intricacies of how the system functions. The primary goal is to identify a model that exhibits similar behaviour.

In the context of a previous experiment, the obtained data includes measurements of both input and output signals, and various models are considered, such as the IT1 block or other alternatives. The key criterion is a good fit between the IT1 model and the measured data. This black-box approach commonly employs approximate models for the process field. Within this approach, two methods exist for representing the process field: explaining through drawings or selecting a graphical model. In both cases, the focus is on effectively capturing the system's behavior based on the available input and output measurements, without relying on a detailed understanding of the internal workings of the system.

1. Self-settling process field

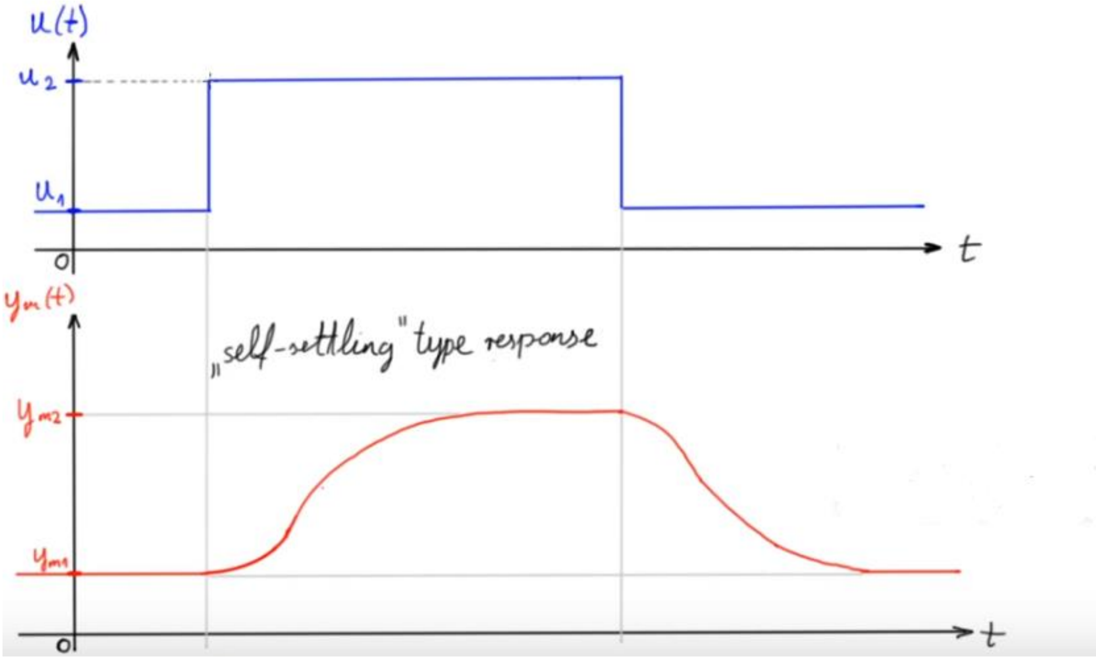

Figure 12:Self-settling process field

The graph in Figure 12 shows that the when the input of the process field (U1) is set to a certain level the output also will be at rest of the ym value, it is shows that for a very short of time the input signal can be increased from U1to a U2 value and then sustain from u2 value for a while and then go back to original value. The output of the process field will start to increase but after an interval of time the slope will reach a finite value which then starts to decrease and then go back to its original value. In this process, the HPT1 model is used which is formular can be shown,

$$GE\ approx\ (s) = \frac{K_P.e^{-s\tau}}{sT_1 + 1} \quad (9)$$

Where,

K_P is the proportional coefficient.

T_1 is the first order time constant and

τ is the time inside the delta block.

2. Integral-type process field

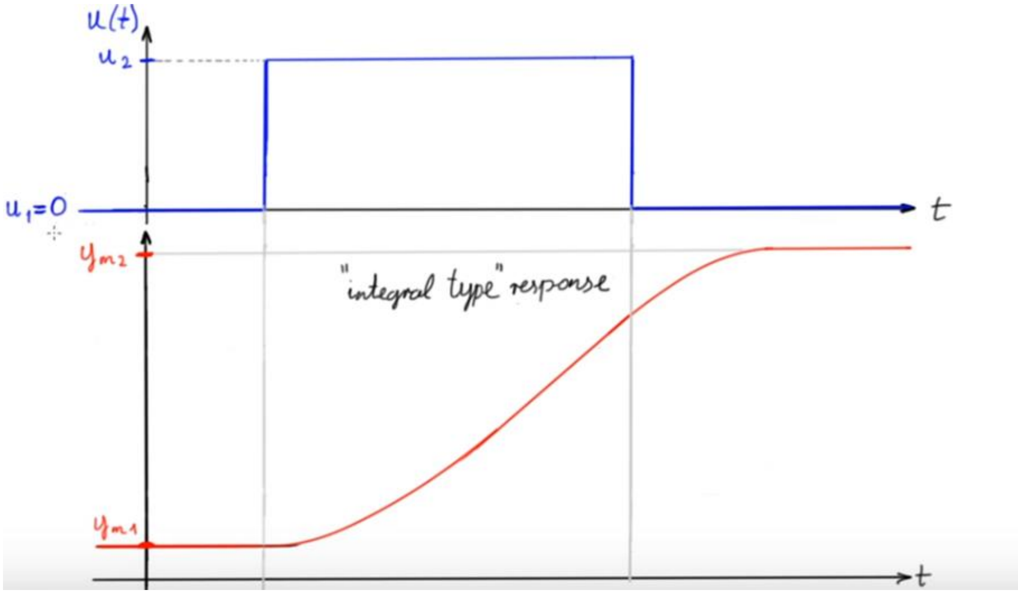

Figure 13:Integral-type process field

In this graph as indicated in Figure 13, you can see how the output of the process field keeps changing when the input is not zero. To maintain a steady output, it's a good idea to set the u1 value to zero. But when there's a sudden change in the input, you'll notice the output initially increases. After a while, though, it stabilizes, forming a smooth, consistent slope that looks like a straight line. Even if the input eventually goes back to zero, the output slope keeps rising and settles at a finite value, never returning to zero. This trend continues as long as the input signal remains non-zero, and you can track it by observing the position on the graph.

In this process, the IT_I model is used.

$$GEapprox(s) = \frac{1}{sT_I} \cdot \frac{1}{sT_1} \quad (10)$$

The parameters are TI and T1.

Where,

TI is the integral time constant and

T1 is the first order time constant.

2.8.2 Grey Box Model

This model type involves theoretical considerations regarding the process field. While the differential of the process field is understood, the coefficients associated with it remain unknown, providing only the form of the process field. Unlike the previous scenario, there exists some knowledge about the system, and a partial mathematical model is already established. However, certain parameters, notably the coefficients of the process field (GE(s)), remain elusive.

Practically, this approach combines a detailed understanding of the mathematical aspects of the system to formulate an initial problem. Despite having knowledge of the system, there are specific parameters that are not known. The strategy here is to use available data to augment this existing knowledge and refine the model structure or parameter set. Consequently, even though the form of the process field is known, the coefficients of GE(s) are still undetermined.

Mathematically, this can be expressed as follows:

$$GE(s) = \frac{K_P}{a_2 S^2 + a1_S + a_0} \quad (11)$$

Where K_P, a_1, a_2, a_0 are the coefficients with unknown values.

The eq.(11) illustrates that the transfer function's structure is identifiable, yet the specific values of the coefficients are undetermined. This analogy is comparable to being able to perceive the inner workings, akin to understanding the differential equation governing the process field, without precisely knowing the values associated with eq.(11). In essence, it allows for a glimpse into the system's internal dynamics, revealing the form of the transfer function, while leaving the coefficients unspecified.

METHODS AND MATERIALS

3.1 Chapter overview

The methods and materials used in this research involved a literature review and comparative analysis of existing studies and research on the use of mobile robots for educational purposes. The analysis examined compatibility, features, programming languages, and overall effectiveness. The materials used included academic papers, research articles, and relevant literature in the fields of mobile robotics and education.

3.2 Introduction

This thesis has been carried out to show the exact way and different methods used to commercialize the available cost effect of mobile robots in educational environments, whether in secondary schools or universities. Introducing basic engineering concepts is one of the major aims of this thesis because it can challenge students to discover and test how a robot can demonstrate its environment, make decisions, and execute tasks. The other option in using the educational kits is to show how the robotics can show some balancing action according to the programmable applied, reading the values measured from one distance to another, which can make the proper curve and the meaning of the thesis's goal. The kit (Mbot Ranger) is regarded as the main educational robot kit used for the thesis, as it provides pictures of the full components on the way of analysing the system from the assembling, scratching, programming, and infinite extensibility in developing more robotic projects. Therefore, Mbot Ranger was a device that helped us reach the target of the thesis. To reach many diverse targets, observation and performance assessments were performed using kits in the laboratory to obtain proper values.

3.3 Data Collection

This step was performed by conducting observation and performance assessments to obtain proper results according to the expected needs. Therefore, the observation was to observe the robot by setting to obtain the balancing values according to the distance measurement and distance travelled by the robot because of the time consumed setting. The operation was conducted by setting different values to obtain the desired performance of the robot, which provides the exact conducting operation desired. The performance assessments were correctly impressed after the desired implemented values and operations were obtained. The system for obtaining the values was collected and observed in different ways, which are described.

3.4 Graphical programming

In the realm of computing, visual programming languages, also known as block coding, enable users to create programs by manipulating graphical elements, rather than textually stating them. These languages provide an alternative to traditional text-based programming, which utilizes lines of code to replace text with pictures or symbols that represent physical objects. Consequently, programming languages that incorporate graphics support a distinct coding methodology.

In contrast to procedural languages such as C and object-oriented languages such as C++ or Java, which rely on high-level statements, graphical languages enable programming through the selection of objects, their linking, and addition of functionality. This is demonstrated in Figure 14 which illustrates the practical application of while loops and for loops in this stage of students' learning. This step explains how the method was used to obtain the proper values for collecting data through graphical programming(scratch). The method of setting and operating the robot is shown in the Figure 14. This shows that the setting of the power motor is generally 250; however, owing to the measurement, the setting should be 100 to -100, meaning that one motor will be operated in the reverse direction compared to the other one.

The following statement has been employed as a while loop, serving as a control-flow mechanism to execute an action repeatedly for multiple instances until a specific criterion is fulfilled. Its configuration stipulates that its execution ought to occur when the elapsed time is less than one second. Upon fulfilment of this condition, the motor will function differently in terms of direction, but the motor value will remain zero. It is crucial to properly direct the ultrasonic sensor ports to ensure effective detection and measurement of an object's distance, based on the operation of the program. Consequently, the ultrasonic sensor will perform accordingly and provide a value that allows for the generation of a graph demonstrating the relationship between the distance travelled, object distance, and time. The set time before the program's execution is 0.5 seconds, which serves as a waiting period for the measurement of distance. This waiting time is vital to the operation of a robotic or automotive system because it ensures that the vehicle moves appropriately and provides accurate value readings that can be used to generate graphs in a separate graphing application. The graphical measurement was calculated from the start of the wait period until the measurement was detected and up until the final measurement according to the program's settings[29]. The values are defined within the Mblock software code and then transmitted to Arduino C, which provides the appropriate values for generating the graph[30].

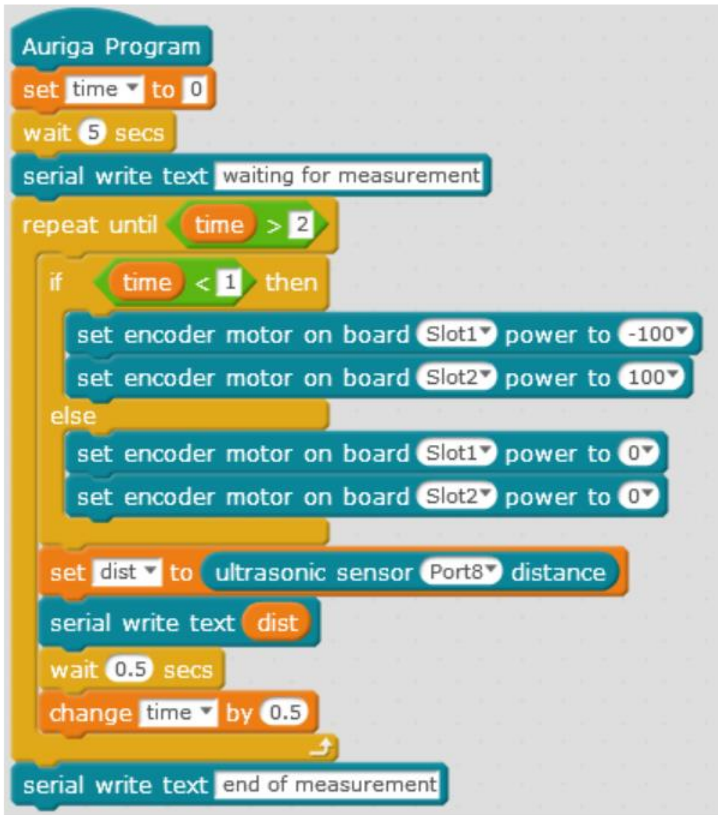

Figure 14:Graphical programming

3.4.1 Excel Data Measurement

Graphical programming has demonstrated significant value in balancing the measurement of the distance travelled, distance of the object, and time. This demonstrates the relationship between an object's distance and its speed over time. The graph depicts the movement of the robotic object in relation to the obstacle object over time, with one line representing the distance travelled by the robot, and the other representing the distance of the object collected during the experiment. The slope of each line indicates the speed rate of the object. The distance travelled can be determined by the total displacement of the object from its starting point to its final point, which is measured over time. Practically, this section focuses on the measured distance and how it changes over time.

The distance travelled and the distance of the object can be used to calculate the distance travelled by the robot:

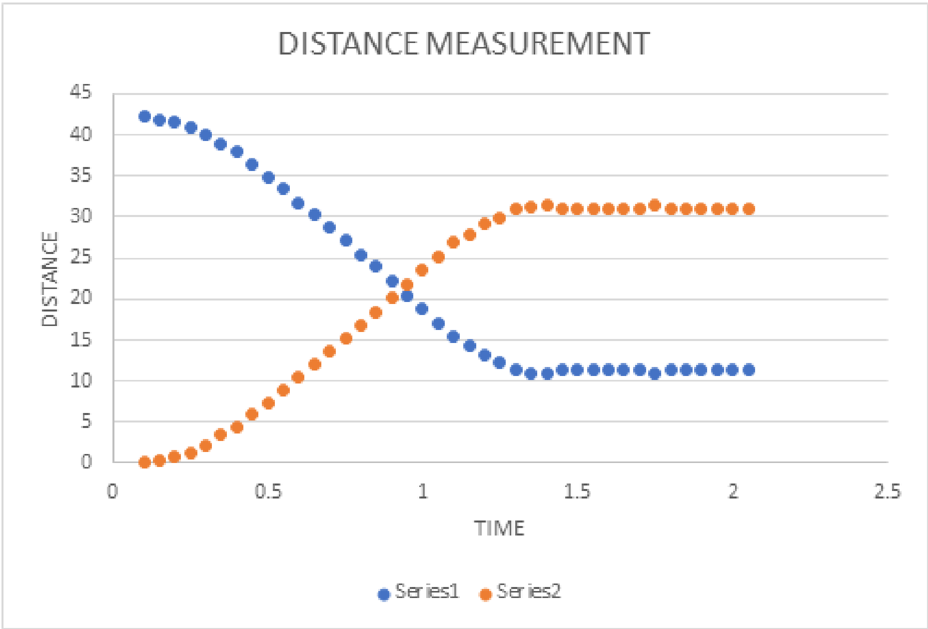

Figure 15:Distance travelled.

3.4.2 MATLAB Measured Data

The interactive MATLAB interface simplifies numerous tasks. From a practical standpoint, the graphical and data-exporting capabilities of MATLAB facilitated the drawing of graphs, thereby reducing the workload. In addition, MATLAB's graphical output is optimized for interaction, and its functionality can be significantly expanded by the addition of toolboxes[31]. As shown in the Figure 16 , the transfer of data enabled clear identification of the input and output signals.

When the input signal is not zero, the output signal changes constantly. In such cases, it is advisable to set the input signal to zero, which results in a constant output signal. If the input signal is increased, the output signal also increases, and after some time, the slope becomes constant, producing a straight line. Conversely, if the input signal approaches zero, the output signal continues to increase, but with a decreasing slope, eventually reaching a finite value and not returning to its original value. It can clearly be seen when the motor power is 100% and the position of the measure distance is actually constant, this is a step function at zero time instant where the value jump from zero to 100 and then go back to zero as the output will not go back to previous value but will remain to the value it was before, in this case the input is increasing for a distance due to the curve of the travelled distance shows. These observations suggest that the model is a black box approach model as there is no knowledge of the system but only the measurements.

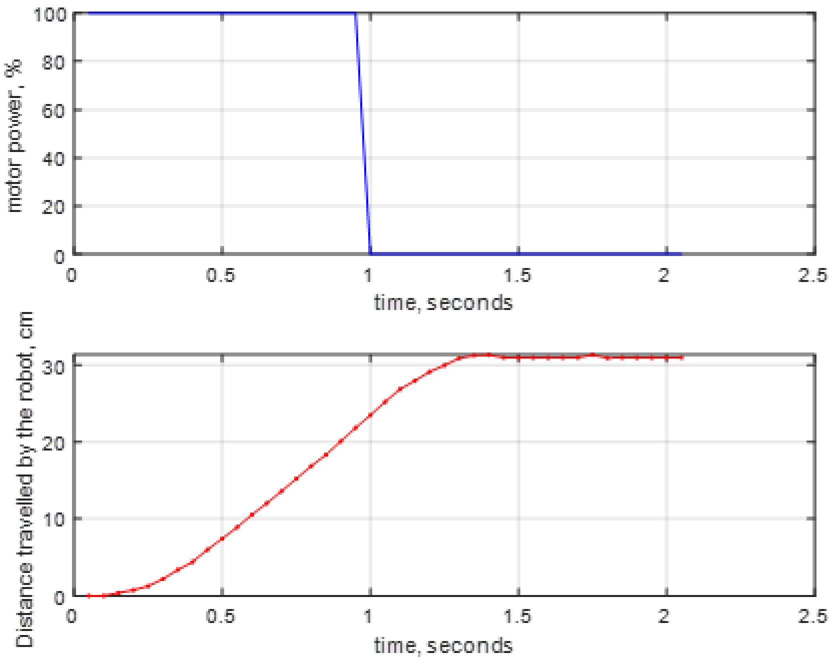

Figure 16:Distance travelled from MATLAB.

3.4.3 Distance PID Tuning

Another method employed to identify the behaviour of the model involves using a PID tuner. In cases where the mathematical model of the plant is unknown, but input and output data are available, it is possible to identify the plant model from the data and utilize it for control design purposes. The PID tuner offers a system identification toolbox that can identify and edit the plant model, as well as import step response functions, impulse responses, or arbitrary input/output data.

In the Figure 17, the imported data specify the name of the signal, which identifies the input and output signal values and sets certain parameters, such as the sample and start time. The system can estimate model parameters, which can be graphically adjusted and fixed based on desired readings, such as the system gain or tangent, to ensure that the measured data are as accurate as possible[32]. The PID tuner can also specify whether the tool estimates or ignores the parameters to achieve the best fit. Automatic estimations often provide the best fit.

This process is employed to adjust the PID response to setpoint changes and unmeasured disturbances, such as variability in control error, with the primary objective of maintaining the measured process value at a setpoint or desired level. In this case the integrator model of one pole has been used which is actually an $IT1$(Integrator first order) approximate module.

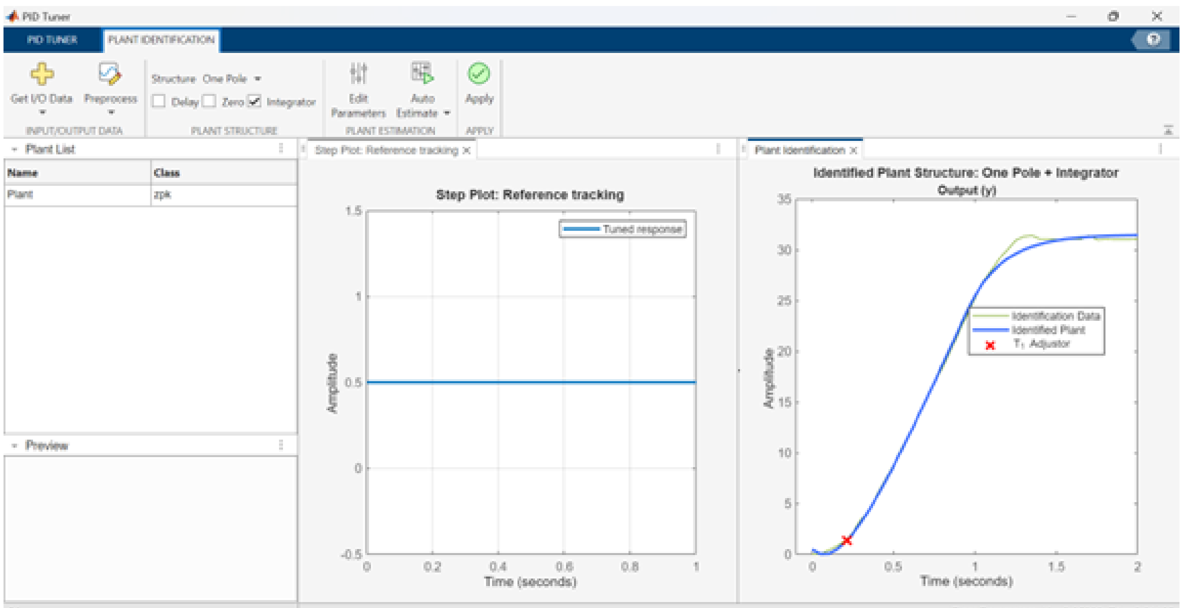

Figure 17: PID tuning.

RESULTS AND DISCCUSION

4.1 Chapter overview

The results and discussion section presents the findings of the comparative analysis, highlighting the similarities, differences, and effectiveness of different mobile robots for educational purposes. It discusses the implications of the findings and provides insights for educators and researchers interested in utilizing mobile robots in educational settings. MBOT Ranger robot and its design

After representing all preliminary information about the robot and its design with the measurements found, the final step is to build up the balancing and distance-holding of the program in the software to obtain the desired results. During the creation of the program, three basic coefficients were considered: the derivative gain, integral gain, and proportion gain were used to test at the same time for the scratch by putting the value that was finally made to achieve good results, as expected. This graphical programming has been divided into three parts. The graphical programming is segmented into three sections. The first part serves as the initialization phase involves input values for the PID controller Figure 18, while the second part constitutes the main, highlighting the PID Controller data Figure 19. The third part marks the end of the program Figure 20. The achievable can be seen as follows:

4.2 Graphical Programming for The PID Data Holding

The first step of the program considered the program structure. Before executing the sequential blocks, the program must wait for one second for a specific waiting period. After the waiting period, the program provides a sound to indicate that it is ready for a command from the user. The program waits for five second before executing the sequential block(s) again. The time interval of the distance between the robot's point and the object's location point was 20 second. The delta time was set to determine the period based on the measurement of the system. The int_sum variable was set to zero for value declaration. The error_Prev variable was also set to zero. The three coefficients the proportional gain (KP), integral gain (KI), and derivative gain (KD) were set with their chosen values for the desired set point or reference value. This will help with the overall control action during execution. To improve the stability of the system performance, the error derivative filter should be set to zero to differentiate the noise frequency for the graphical interface of the program. T_filter should be set to 0.05 seconds to determine the time interval for filtering the controlling signal, where the time should be set to zero.

Figure 21:Graphical Programming for the PID Data Holding

The program was designed to utilize the repeat block function, which enables the implementation of all included blocks, and then stops when the specified state condition is met. To achieve this, the "repeat until" function has been employed to set a time value greater than 20 seconds, which will trigger the execution of the program when the time exceeds this threshold. The delta time and time values were added to provide an additional value, which is useful in determining the system measurements and position settings.

It is essential to establish a connection between the ultrasonic sensor and Auriga board, which requires inserting the pin into port 8 to ensure proper tracking. The distance setting was identified, and the measured error calculation was determined by the difference between the desired position of the robot and the measured distance of an object. This will help correct the base value in the three channels of the proportional, integral, and derivative functions. Mathematically, the summation and multiplication of the error value set are applied to achieve the desired outcome, int_sum and delta time to obtain the total value of the time applied. This step must be performed to find the correct value of the error that will be used for the measurement and then set the error_derivative_filter to filter out the value of the error occurring to the measured distance with the value of the delta time and T_filetr. The program then sets three channels to three different coefficient gains together with the error_derivative_filter, int _sum, which shows the accumulation of the signal over time when the integral gain can observe the balancing of the integral sum and obtain a fast and accurate response to our system. The proportional gain is set to observe the behavior response of the error signal, whereas the derivative gain is set to show the rate of change of the error signal concerning the filtering of the error derivative of the signal. Subsequently, the control signal was set to the three-parameter channel, which was proportional, derivative, and integral, to synchronize the operation of the program and attain the actual value.

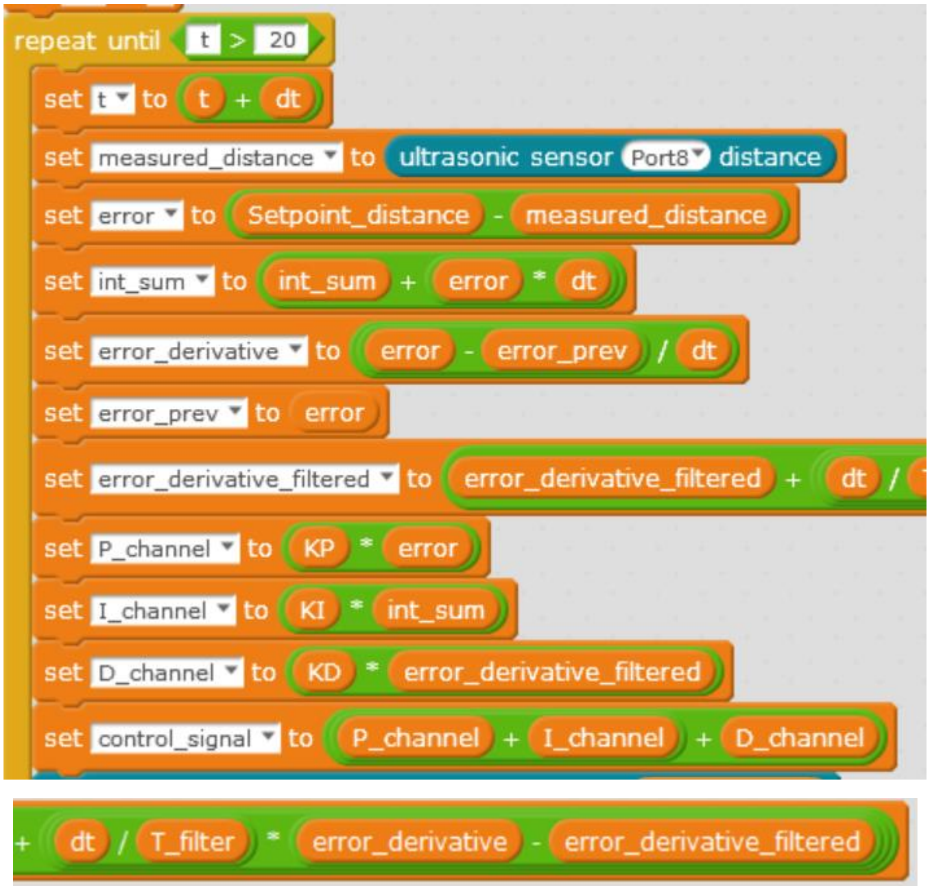

Figure 22:Graphical Programming for The PID Data

The encoder motor in slot one was configured to rotate in a clockwise direction upon reception of the control signal, signifying clockwise rotation of the motor. The second encoder motor in slot two was set to rotate in an anticlockwise direction upon reception of the control signal, indicating a reverse direction for the motor. The robot executes commands based on the object being held and moves either forward or backward based on the execution of the program. The serial-written text enabled the robot and object to measure the distance between them. After waiting for a certain period, the program commands the robot to produce a sound, indicating completion of the measurement. Consequently, the encoder motor in slot one will be set to zero power, and the encoder motor in slot two will also be set to zero power, signifying the conclusion of the program and distance measurement for the robot.

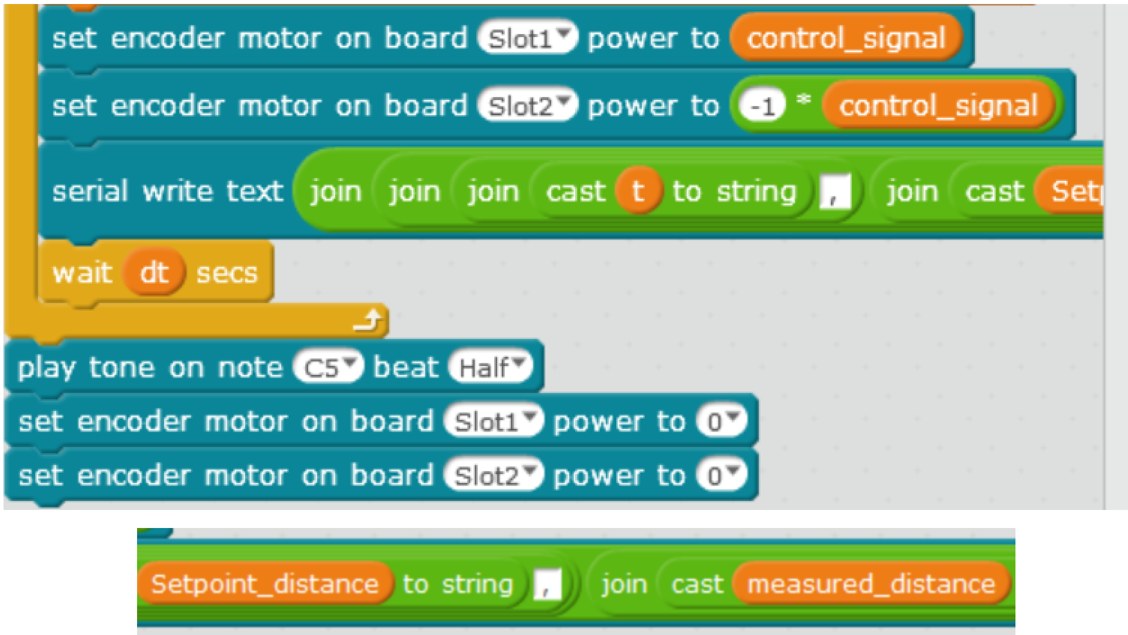

Figure 23:PID controller's measured data holding.

The graphical programming representation in Figure 24 depicts the PID controller's measured data holding for the time, measured distance, and control signal. The graph consists of two lines, with the blue line representing the input and red line representing the output. The system graph illustrates the implementation of KP, KD, and KI, showing the moving line that represents the system overshoot. This line starts below the setpoint distance and overshoots it but eventually reaches the setpoint after some time. In the middle of the graph, noise is filtered out as the signal approaches the desired value. The output signal of the graph starts in Figure 24 the control signal setting and shows the system overshooting for a certain amount before reaching the desired value, and then filtering out according to the measurement setting. The system continued in this direction until the values changed from scratch.

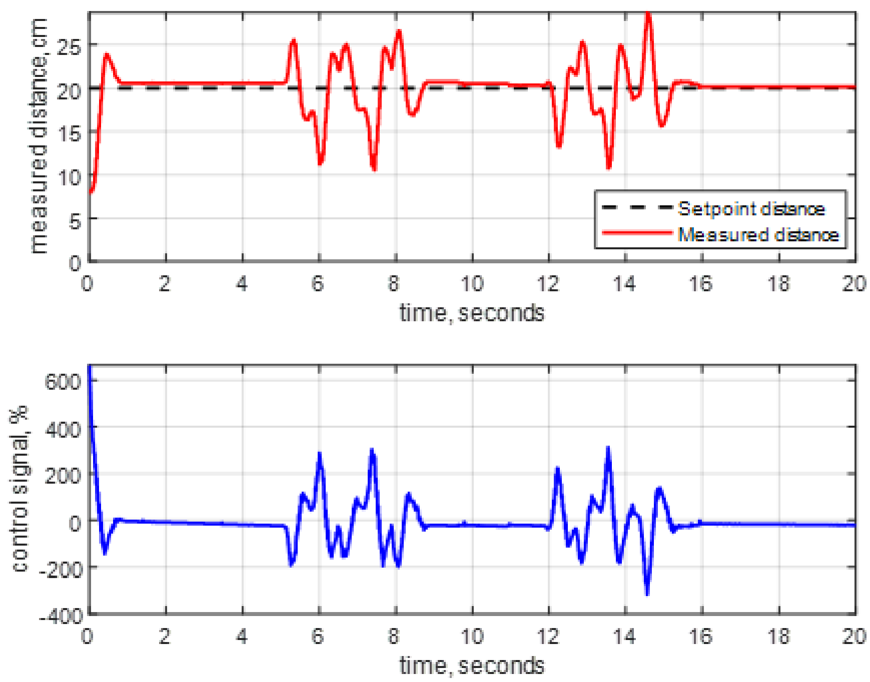

Figure 24:PID controller for time, measured distance, and control signal graphs.

The graph in the Figure 25 depicts the aggressive behaviour of the signal as the value of KD is increased, resulting in a rapid change owing to the error signal. The graph shows the variation in the signal as the value changes, which also demonstrates the amplification of noise in the system control, leading to a loss of damping oscillation. The graph begins with the measured signal in Figure 25 the desired value, but after some time, it reaches the set point of twenty-two seconds as set value. However, it overshoots for a certain period before settling at the set-point value.

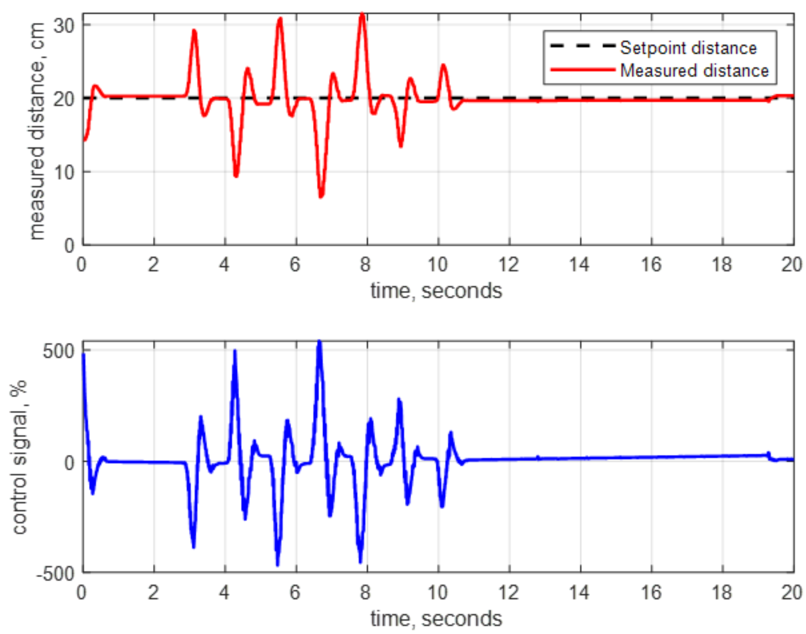

Figure 25:The variation of the signal as the value change

The graph in Figure 26 illustrates how the graph changes as the integral gain increases to a higher value. The line in the graph represents the desired value of the system, which is achieved through rapid response and overshoot, this overshoot causes undesirable oscillations in the graph. Improperly tuned PID parameters can result in steady-state errors and instability, which can disrupt the system balance. Therefore, careful adjustment is necessary when selecting values to measure the control of the system. Poorly set parameters may lead to undesirable behavior, as shown in the graph.

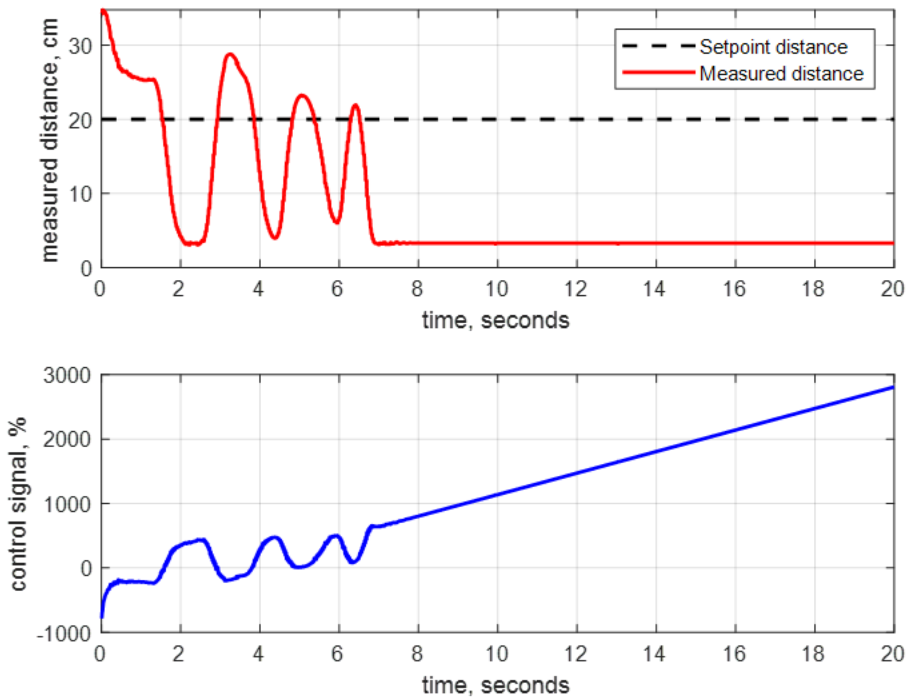

Figure 26:Changing of the graph as the value of the integral gain has been changed to a higher

CONCLUSION AND RECOMMENDATION

5.1 Conclusion

This study provides a comprehensive analysis of the utilization of mobile robots for educational purposes, addressing the specific objectives of comparing different mobile robots, fine-tuning PID controllers in MATLAB for optimal control outcomes, and assessing the overall effectiveness of these robots. By evaluating compatibility, features, programming languages, and control outcomes, valuable insights were obtained, significantly contributing to existing knowledge in the field and holding important implications for educators and researchers.

The study includes addressing the challenges identified in fine-tuning PID controllers in MATLAB, ensuring that educators can optimize control outcomes for a seamless learning experience. By doing so, students gain access to enhanced learning opportunities, and the development of critical thinking skills is fostered.

This thesis has been successful achieved on what has been outline from over all explanations, due to this achievement it is possible to use this kind of education robot (mbot Ranger) to the different parts of the educational environment which can allow the students to practice and apply the PID controller as the part of the curriculum outline, which can help them not only in understanding of the PID controller in educational robotics but also to engage on the practical applications and adapting robotic platforms in more experiential learning.

5.2 Recommendations

Building upon the analysis and findings, several recommendations are proposed to further enhance the utilization of mobile robots in educational settings.

a) Educators and institutions are encouraged to accurately assess the compatibility, features, and programming languages of mobile robots during the selection and implementation process. Attention should be given to the fine-tuning of PID controllers in MATLAB to achieve desired control outcomes, addressing any challenges identified during this process.

b) Manufacturers and developers are urged to sustain innovation and improvement in mobile robot technologies to meet the evolving needs of educators and students in educational settings. This involves advancing features, programming languages, affordability, and user-friendliness. Continuous enhancement of mobile robot technologies not only supports educators in delivering engaging and effective learning experiences but also addresses the nuanced requirements of educational environments, including the optimization of PID controllers for precise control outcomes.

REFERENCES

[1] R. Raj, and A. Kos, "A Comprehensive Study of Mobile Robot: History, Developments, Applications, and Future Research Perspectives," *Applied Sciences,* vol. 12, no. 14, 2022.

[2] M. Hudson, "Using a Multifaceted Robotics-Based Intervention to Increase Student Interest in STEM and Computational Thinking Skills," 2019.

[3] B. Nasirian, M. Mehrandezh, and F. Janabi-Sharifi, "Efficient Coverage Path Planning for Mobile Disinfecting Robots Using Graph-Based Representation of Environment," *Frontiers in Robotics and AI,* vol. 8, 2021.

[4] T. Kargar Tasooji, and H. J. Marquez, "Cooperative Localization in Mobile Robots Using Event-Triggered Mechanism: Theory and Experiments," *IEEE Transactions on Automation Science and Engineering,* vol. 19, pp. 3246-3258, 2022.

[5] G. Fumagalli, Y. Huber, M. Dymczyk, R. Y. Siegwart, and R. Dubé, "Fast Image-Anomaly Mitigation for Autonomous Mobile Robots," *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS),* pp. 2436-2442, 2021.

[6] A. Erdoğmuş, and U. Yayan, "Virtual Robotic Laboratory Compatible Mobile Robots for Education and Research," *2021 International Conference on INnovations in Intelligent SysTems and Applications (INISTA),* pp. 1-6, 2021.

[7] C. Connolly, J. C. Walsh, H. Worlikar, L. Ryan, A. Murray, S. O'Connor, J. Kelly, S. Coleman, V. Vyas Vadhira, E. Newell, and D. T. O'Keeffe, "Exploring new frontiers of education using humanoid robots – a case study of patient centred innovation in digital health education," *Irish Educational Studies,* vol. 41, pp. 107 - 115, 2022.

[8] J. Li, R. Suzuki, and K. Nakagaki, "Physica: Interactive Tangible Physics Simulation based on Tabletop Mobile Robots Towards Explorable Physics Education," *Proceedings of the 2023 ACM Designing Interactive Systems Conference,* 2023.

[9] S. Amri, C. Budiyanto, and R. Yuana, "Beyond computational thinking: Investigating CT roles in the 21st century skill efficacy," *THE 2ND INTERNATIONAL CONFERENCE ON SCIENCE, MATHEMATICS, ENVIRONMENT, AND EDUCATION,* vol. null, pp. null-null, 2019.

[10] S. Anwar, N. Bascou, M. Menekse, and A. Kardgar, "A Systematic Review of Studies on Educational Robotics," *Journal of Pre-College Engineering Education Research (J-PEER),* vol. null, pp. null-null, 2019.

[11] L. Cesaretti, L. Screpanti, D. Scaradozzi, and E. Mangina, “Analysis of Educational Robotics Activities Using a Machine Learning Approach,” *Makers at School, Educational Robotics and Innovative Learning Environments,* vol. null, pp. null-null, 2021.

[12] P. Christodoulou, A. Andreou, and Z. Zinonos, “skillsChain: A Decentralized Application That Uses Educational Robotics and Blockchain to Disrupt the Educational Process,” *Sensors (Basel, Switzerland),* vol. 21, pp. null-null, 2021.

[13] P. R. Jeyaraj, and E. R. S. Nadar, “Real-time data-driven PID controller for multivariable process employing deep neural network,” *Asian Journal of Control,* vol. 24, pp. 3240 - 3251, 2022.

[14] J. Wang, M. Li, W. Jiang, Y. Huang, and R. Lin, “A Design of FPGA-Based Neural Network PID Controller for Motion Control System,” *Sensors (Basel, Switzerland),* vol. 22, 2022.

[15] M. Cartagena, C. Bonilla, J. Menendez, and M. Cardona, "Design and Development of a Modular Robotic Kit “CubeBot” for Educational Purposes." pp. 1-7.

[16] C. Chronis, and I. Varlamis, “FOSSBot: An Open Source and Open Design Educational Robot,” *Electronics,* vol. null, pp. null-null, 2022.

[17] S. Evripidou, K. Georgiou, L. Doitsidis, A. Amanatiadis, Z. Zinonos, and S. Chatzichristofis, “Educational Robotics: Platforms, Competitions and Expected Learning Outcomes,” *IEEE Access,* vol. 8, pp. 219534-219562, 2020.

[18] A. Sophokleous, P. Christodoulou, L. Doitsidis, and S. Chatzichristofis, “Computer Vision Meets Educational Robotics,” *Electronics,* vol. null, pp. null-null, 2021.

[19] N. A. Uslu, G. Yavuz, and Y. K. Usluel, “A systematic review study on educational robotics and robots,” *Interactive Learning Environments,* vol. 31, pp. 5874-5898, 2022.

[20] G. Karalekas, S. Vologiannidis, and J. Kalomiros, “EUROPA: A Case Study for Teaching Sensors, Data Acquisition and Robotics via a ROS-Based Educational Robot,” *Sensors (Basel, Switzerland),* vol. 20, pp. null-null, 2020.

[21] D. Scaradozzi, L. Cesaretti, L. Screpanti, and E. Mangina, “Identification of the Students Learning Process During Education Robotics Activities,” *Frontiers in Robotics and AI,* vol. 7, pp. null-null, 2020.

[22] D. Scaradozzi, D. Scaradozzi, L. Screpanti, L. Cesaretti, M. Storti, and E. Mazzieri, “Implementation and Assessment Methodologies of Teachers’ Training Courses for STEM Activities,” *Technology, Knowledge and Learning,* vol. 24, pp. 247-268, 2019.

[23] M. Valzano, C. Vergine, L. Cesaretti, L. Screpanti, and D. Scaradozzi, “Ten years of Educational Robotics in a Primary School,” *Makers at School, Educational Robotics and Innovative Learning Environments*, vol. null, pp. null-null, 2021.

[24] S. Kumar, and T. Raj, "A multi attribute selection of mobile robot using AHP/M-GRA Technique: A multi attribute selection of mobile robot using AHP/M-GRA technique," *Robotic systems: Concepts, methodologies, tools, and applications*, pp. 262-284: IGI Global, 2020.

[25] S.-G. Racz, M. Crenganiș, R.-E. Breaz, A. Maroșan, A. Bârsan, C.-E. Gîrjob, C.-M. Biriș, and M. Tera, “Mobile Robots—AHP-Based Actuation Solution Selection and Comparison between Mecanum Wheel Drive and Differential Drive with Regard to Dynamic Loads,” *Machines,* vol. 10, no. 10, pp. 886, 2022.

[26] M. Simion, L. Socaciu, O. Giurgiu, and S. M. PETRIŞOR, “The selection of industrial robots for military industry using AHP method: a case study,” *Acta Technica Napocensis-Series: Applied Mathematics, Mechanics, and Engineering,* vol. 61, no. 2, 2018.

[27] A. Maddi, A. Guesssoum, and D. Berkani, "Applying a Technique of Identification for Estimating Definite Integrals."

[28] W. Shubo, X. Ren, N. Jing, and D. Li, “Robust Integral of Sign of Error and Neural Network Control for Servo System with Continuous Friction,” *2016 35th Chinese Control Conference (CCC),* pp. 3531-3536, 2016.

[29] T. Wu, Y. Zheng, L. Qipu, S. Guoqiang, X. Wang, and L. Xindong, “Continuous Commutation Failure Suppression Method Based on Self-adaptive Auto-disturbance Rejection Proportional-integral Controller for HVDC Transmission System,” *Journal of Modern Power Systems and Clean Energy,* 2020.

[30] m. Ranger, “mBot Ranger Instruction PDF size,” *Makeblock Forum,* 2016.

[31] C. Jobst, P. Ferrari, S. L. Isabella, and D. O. Cheyne, “BrainWave: A Matlab Toolbox for Beamformer Source Analysis of MEG Data,” *Frontiers in Neuroscience,* vol. 12, 2018.

[32] Q. Li, D. Gong, J. Shen, C. Rao, L. Ni, and H. Zhang, "SF-MVPA: A from raw data to statistical results and surface space-based MVPA toolbox," *Frontiers in Neuroscience,* vol. 16, 2022.