Date: 28.04.2023

ME304 - Motion Control Project Progress: Motion Control of a Differential Drive Robot For a Predetermined Path

Nilsu Atlan, Okan Arif Güvenkaya, Ege Kırdar
Sabanci University, Istanbul, Turkey
e-mail: natlan@sabanciuniv.edu, okanarif@sabanciuniv.edu, egekirdar@sabanciuniv.edu

I. MATERIALS AND METHODS

A. MECHANICAL DESIGN OF THE ROBOT

a) Differential Drive Robot Design

Differential drive robots are mobile which moves with two separate motor driven wheels which are placed on the side of the chassis (body of the robot) [6]. Chassis of the differential driven robots can be design in various shapes such as rectangular or circular depending on the application. By the chassis design of the robot, as the result kinematics of the robot changes. The mathematical model and the differential robot kinematics should be kept as simple as possible to make a simpler control algorithm design [6]. In addition to mathematical model, the chassis design is essential component for the dynamic model of the differential robot [3]. The dynamics of differential drive robots are non-linear and includes non-holonomic wheel constraints which makes difficulties in modeling and control algorithms [3].

In this project, the chassis of the differential driven robot was designed in SolidWorks CAD program which allows to visualize the model of the chassis design and make corresponding scaling processes according to the components will be used. In this project, a circular based chassis design was preferred to make the mathematical modelling (kinematics), and in the further steps controlling of the system easier.

Along with the mathematical modeling of the robot chassis, the chassis should involve essential components to move, power and control such as battery, motors, motor driver, microcontroller, breadboard, and other components that vary depending on the application. By considering these essential components, design may become more complex while including all the components on the chassis and the kinematic may show differ. To prevent that, instead expanding horizontally, adding an extra level to the chassis which is expanding in vertically helps to sustain the kinematics of the robot.

In this project, chassis of the robot includes three batteries, two motors, a motor driver, microcontroller as Raspberry PI, a breadboard to connect all the connections in one place and a QR code which 21x21-sized printed on a square paper to calibrate the camera. Along with the components, there are many components involves on the chassis so to maintain the strength of the chassis, Plexi was chosen as the key material of the chassis. In Plexi material which is a transparent thermoplastic homopolymer that satisfies high strength with a lower weight. There are 3-, 5- and 7-mm thickness options in Plexi, among these options 3 mm thickness was picked since the length of the chassis is 40 cm long, to not make the chassis heavier 3 mm thickness will be sufficient to maintain enough strength for the chassis. In this sense the upper and lower bodies of the chassis was made of 3-mm thickness and the bodies relate to a screw between them. Depending on the wheel size, the screw heights are adjustable.

b) Wheels

Wheels are highly important in differential drive robots. There are two wheels and one ball caster on the chassis. Wheels are controlled by the motors and ball caster helps the system to stay stable in the initial position and during the motion, so it doesn’t be driven any motors.

As mentioned in the chassis design, the differential drive robots involve non-holonomic wheels. In this project there are two different wheels are planning to be connected to the chassis. Depending on the connections with the motor motors of each wheel, whichever provides an easier connection, it will be implemented on the chassis. In addition to wheel selection, depending on the selection, height of the wheel and motor connections should also change since the distance from the ground of each wheel (ball caster and non-holonomic wheels) should be equal to stabilize the system.

There are two options in wheels in our project which are small and large wheels. Small wheel has 7.5-cm height, 3-cm thickness, and the large wheel has 12-cm height, 6.5-cm thickness. Also, the ball caster width is 47-mm, and the radius of the ball is 44-mm.

B. ROBOT HARDWARE DETAILS

a) Raspberry Pi Model Details

For the robot control, the system requires a microcontroller. There were several options for microcontrollers for the project such as Arduino boards. For coding the robot in Python 3 and large-scale usability scope, Raspberry Pi 4 B 2GB RAM decided to be used as a main control unit in the robot.

The Raspberry Pi is a small computer, so it requires an operating system such as Linux or Windows. Since it has 2 GB ram and as mentioned it is kind of a small computer, there is a specific operating system that is suggested for beginners. It is Raspian.

In total, there are 40 pins in the model for electronic connection and communication. The pin configuration of raspberry pi 4B is shown below figure: [12]

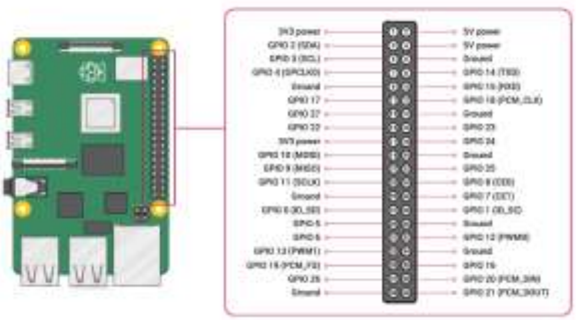

FIGURE 1. **Raspberry Pi 4 Pin Configuration**

Required Pin Information:

GPIO: These pins stand for general-purpose input/output. GPIO pins provide data exchange with other hardware elements. Digital or PWM-type signals can be used throughout these pins for communication and control. The voltage range of the GPIO pins is 3.3V (HIGH) to 0V (LOW). [12]

- Digital Signals: These signals are binary signals as low and high.
- PWM: It stands for pulse width modulation. By arranging the duty cycle of the signal, a kind of analog signal is created. By PWM, any voltage can be given from GPIO pins between 3.3V-0V. [13]

Ground: For electrical safety, common ground is an important issue. These pins represent the ground of raspberry pi. [12]

b) Motor and Encoder Details

For the actuation of the wheels of the robot wheels, a motor is required. By considering load conditions as the mass of the vehicle the motor selection was completed. 12V brushed DC motor was determined to use. The model of the motor is 37D Metal Gearmotors belonging to the brand Pololu. Another cause to select this motor model was that the encoder is already mounted on the motor itself. Encoders have 2 hall effect sensor and these quadrature encoders provides a resolution of 64 counts per revolution. Two motors are used in the project to control the wheels of the robot independently. [8]

c) Motor Driver Details

To control the speed and rotation direction of the motor and arrange the voltage between the microcontroller and motor, a motor driver is required. Based on the selected 12V DC motor a motor driver was selected. The selected motor driver model is L298N. By 1 L298N module 2 motors can be controlled. With the help of the H-Bridge, the motor rotation direction is controlled. The driver also regulates the motor speed by using PWM. [4] Pin configuration of the L298N is shown in the following figure.

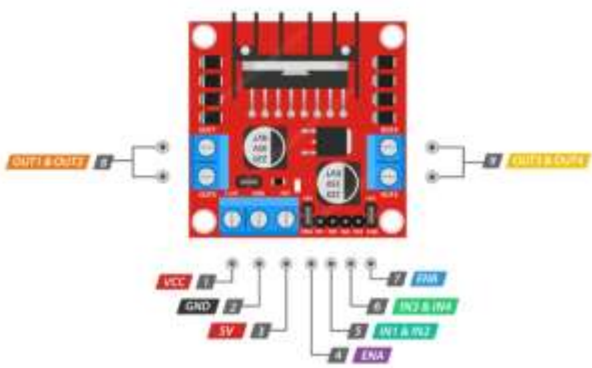

FIGURE 2. **Driver Pin Configuration**

Pin Information of Motor Driver:

- VS: pin powers the IC’s internal H-Bridge, which drives the motors. This pin accepts input voltages ranging from 5 to 12V

- GND: is the common ground pin.
- OUT1, OUT2 and OUT3, OUT4: Output channels of the motor driver for motors A and B.
  - 5-12V DC motors can be connected to these terminals
  - Each channel on the module can supply up to 2A to the DC motor.
- IN1, IN2, and IN3, IN4: pins control the spinning direction of motors A and B

Input1	Input2	Spinning Direction
Low(0)	Low(0)	Motor OFF
High(1)	Low(0)	Forward
Low(0)	High(1)	Backward
High(1)	High(1)	Motor OFF

FIGURE 3. Motor Direction Control Pin Configuration

- ENA and ENB: used to turn on/off the motors and control their speed.
  - Pulling these pins HIGH will cause the motors to spin while pulling them LOW will stop them. However, with Pulse Width Modulation (PWM), the speed of the motors can be controlled.

d) Battery

Energy is one of the essential issues to be deal with in the robot. During the implementation of the robot, since the this is a differential drive robot, it moves. Throughout the movement, it is important to use a reliable that ensures us that it will last till the end of the process also the process time is unknown in the project, the battery should last long time in the movement. Therewithal, the energy source shouldn't be connected to an electric power unit which will create constraints in the movement.

By considering all these issues, it was decided to use either a "Lipobatt" or a power bank which can provide 12V for the motor drivers. The reason of these options in the battery selection were made because of their long-time charging capabilities and their re-chargeable properties.

In the case of "Lipobatt", there are various types of connections such as either connecting a battery that provides 12V DC directly or using batteries that each provides 3.7 V DC and connecting them serially in the breadboard to get 12V. In the other side, power bank can be anything provides the requirements.

Among the batteries, there are advantages and disadvantages while in the power bank the charging capacity is visible, in "Lipobatt" it is not visible which crates hard times in the low char of the battery.

e) Aerial View Camera

Throughout the control of the differential drive robot, to compensate for the errors that may occur from the feedback retrieved from the encoders an aerial view camera implementation is crucial for this project. This camera is integrated into the system utilizing an ArUco Marker Detection algorithm which is further explained in the continuation of the Methods part of this report.

There are two options explored for the aerial view camera selection:

1. AEE Magicam
2. GoPro Hero 10

The compatibility of the cameras with Python and their performance are deciding elements for this selection. Both action cameras are operable with Python such that the information can be retrieved to be utilized in controlling of the robot.

C. MATHEMATICAL MODEL

a) Robot Kinematics

The differential drive is one of the common locomotion-types of autonomous robots. Two controllable wheels help the motion of a differential drive robot [10]. In addition to driving wheels there can be a passive wheel to keep the robot in static stability by providing the third contact point. This configuration ensures that the center of gravity will be within the triangle formed by the ground contact points of the wheels [11]. In the mathematical model of the differential drive robot, there are 3 important parameters. These 3 parameters represent the position and orientation of the robot. The representation point is on the middle point of the wheel's rotation axes. This point is accepted as the centroid of the robot and mathematical modeling is created based on that point. The parameters are as shown in the following:

- $\boldsymbol{\theta}$: It is the directional angle from the reference coordinate frame to the robot, or it is called the heading angle

- x: position of the robot along the x-axis with respect to the reference frame
- y: position of the robot along the y-axis with respect to the reference frame

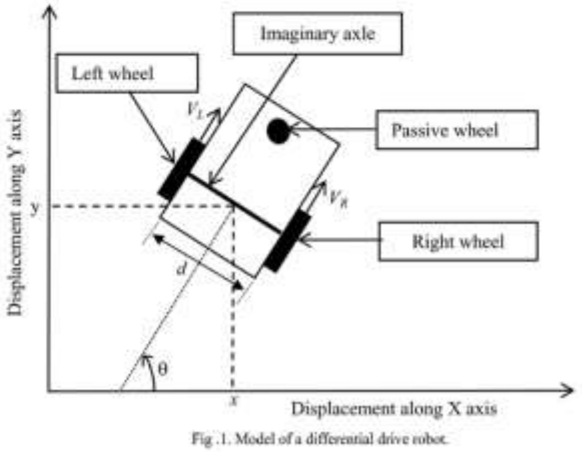

FIGURE 4. **Model of a differential drive robot.**

The robot moves with the motion of the left and right wheels. The maneuvers that the robot can do is as follows:

1. Linear Motions:

    a. Move Forward

    b. Move Backward

2. Rotational Motion

    a. Turn Left

    b. Turn Right

Linear motion is provided by the rotation of the wheels in the same direction. The mathematical representation of linear motion is as follows:

$$V = \frac{(V_R + V_L)}{2}$$

Rotational motion is provided by rotating the wheels in the reverse direction. The mathematical representation of rotational motion is as follows:

$$W = \frac{(V_R - V_L)}{d}$$

There is a relationship between the linear velocity of the wheels, the angular velocity of the motors, and radius of the wheels. Therefore, the above equations can be represented in terms of angular velocities and radius of the wheels [6].

$$V = \frac{R(w_R + w_L)}{2}$$

$$w = \frac{R(w_R - w_L)}{L}$$

Where w_R and $w_L$are the angular velocity of the right and left wheels of the robot, and R is the radius of the wheels.

Modeling of the robot is done by accepting the following assumptions:

1. Rolling Contact: The wheel must roll when the motion happens, and no slippage occurs.

2. No Lateral Motion: During the motion of the robot, the wheel must not slide through the rotation axis.

The rate of change of position of the robot along the x and y directions can be represented as the x dot and y dot. And the angular velocity of the robot is represented as theta dot or omega(w).

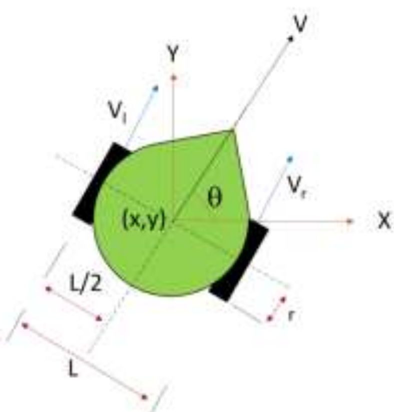

FIGURE 5. **The mathematical formulation.**

$$\dot{x} = V * \cos\theta$$

$$\dot{y} = V * \sin\theta$$

$$\dot{\theta} = \frac{(V_R - V_L)}{L}$$

[11]

In the end, the kinematic equation of the differential drive robot becomes the following equation:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} V \\ \omega \end{bmatrix}$$

a) Control System

The differential drive robot is controlled with its corresponding feedback from the encoders embedded in the motors and the aerial view camera selected.

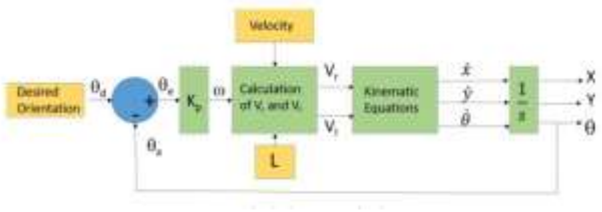

FIGURE 6. Block Diagram of The Control [N5]

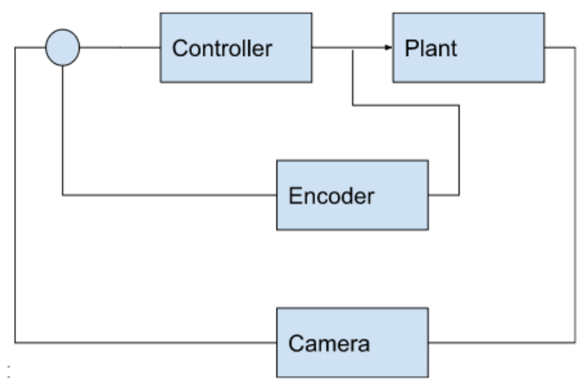

FIGURE 7. Block Diagram of The Control with Feedback

The most crucial element of the control of a differential drive robot is the orientation error. To compensate for the error coming from the encoders the camera is introduced to this system. An if condition is integrated to the control blocks. The information is normally retrieved from and tracked by the encoder. Whenever the ArUco marker detection algorithm detects an error larger than the given small angle limit the feedback of the camera and the orientation of the detected error from the camera are enabled. The x and y axes are virtually placed on the differential drive robot and the ArUco marker detection algorithm's features are used to calculate the orientation error which is the amount of deviation from the denoted axes.

B. ROBOT SOFTWARE DETAILS

a) Remote Connection to Raspberry

Robot motions are controlled with Raspberry PI 4, to control the robot, first the controlling of Raspberry PI should be made. Raspberry PI is a controller microcomputer sized computer which operates in the same manner as regular PC that involves its own operating system, RAM, CPU, and ports that a regular computer has [14]. In addition to these hardware properties, it can also connect to internet by WIFI or Ethernet.

In this project all the robot controllers were built in Raspberry PI to control the motor movements and to make trajectory following predictions, it is the brain of the robot. To control Raspberry PI, since it operates as a PC, it should be opened and from its own operating system, the corresponding control applications should be written inside it. As a regular PC it has its own desktop inside which can open essential programs like internet or programming IDE's. There are several connection techniques in the literature to connect to raspberry PI's desktop such as SSH, VNC, TTL. However, before applying these techniques, Raspberry PI should be opened with a monitor, and keyboard to make the first time set up.

SSH also known as "Secure Shell", is a network protocol that provides remote access to processors of a robot, from the external computer [9]. By using it, programs that already exist can be executed or it allows you to open a terminal window on Raspberry Pi from the command line of the host computer to login to the server [9]. Also, it makes connections either with wireless or wired [9], and raspberry pi already satisfies these preferences. SSH operates in Linux environment, which makes it possible to make connections with ROS (Robotic Operating System) that provides utilities for users in robotic control [9]. A lot of programs can be run by using SSH that makes it an essential technique in robot control [9].

One other way to communicate is VNC as known as "Virtual Network Computing", communication with VNC provides for remote communication to access Raspberry Pi graphical desktop. Difference between SSH and VNC is that VNC provides a connection with the host computing environment from anywhere around the world [15]. On the other hand, SSH provides communication with the host computer within a server. VNC allows an important flexibility of mobile access that provides connection to the host environment simultaneously without requiring to carry the specific hardware [15]. It is a "thin client" system that is designed to require less requirements from the end-user [15]. Unlike the regular remote-control applications, VNC is a protocol that is totally independent of the operating system, it operates over a reliable transport like TCP/IP, so it doesn't have proprietary protocols that makes it difficult to display reliable information [15].

The last method is TTL, which has the same purpose with VNC, but it connects with a cable, so the user should be next to Raspberry PI to reach its desktop from user's own laptop.

Overall, among these connection techniques to the desktop of Raspberry PI, VNC was considered as the most efficient among the methods since without any cable limitations, with wireless techniques we can connect to the desktop of Raspberry PI.

b) IDE and Python

Main language of Raspberry PI is Python to control the microcontroller. In this project Raspberry PI 4 has been using and it supports Python version 3.

Python is an efficient object-oriented programming language which has a dynamic syntax that makes it preferable for scripting to control components [2]. Also, python supports a comprehensive open-source library and modules capacity that can be used in most of the environments [2]. Moreover, since it is a level programming language, computation in python is very fast [2]. Additionally, as it is, built-in language, there is no further requirement to download a script in python which makes it rational that commands for the robot can be written in python.

In Raspberry PI, the supported IDE is "Geany" which provides to code in Python. Geany is pre-installed IDE in Raspbian which is the operating system of Raspberry PI. If it is not visible in the default set up in Raspberry, it can be easily downloaded from the control terminal by writing a simple command as "sudo apt install geany".

C. ARUCO MARKER DETECTION

Initially, an ArUco marker needs to be generated specific to this project. Before their detection, markers also need to be printed and placed in the environment which for the context of this project is on top of the differential drive robot. Marker images can be generated using the "generateImageMarker()" function [7].

The ArUco marker is placed on top of the robot, visible by the aerial view camera. The ArUco module is employed for pose estimation [7] via vision-based localization. "The encoding of information and the monochromatic design of these markers make the detection and localization of the markers within a frame quite fast and extremely efficient" [1]. The changes in terms of orientation of the direction reference vector and the position of the marker are utilized in reorienting and repositioning the robot while recalculating a path for the differential drive robot to follow. The analysis of the marker is done by the camera and the computer pair.

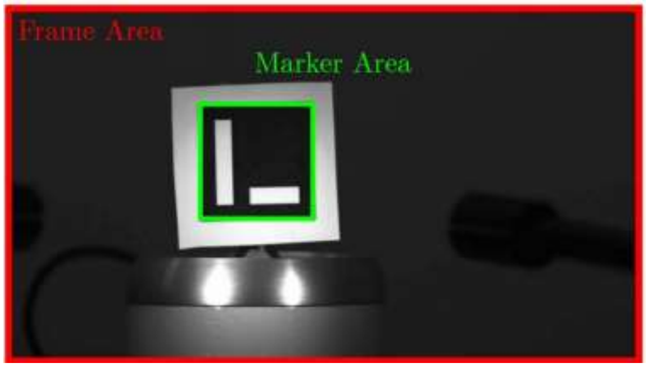

FIGURE 8. ArUco Marker Detection Example

II. RESULTS

A. MECHANICAL DESIGN

a) Chassis Design

The final design of the chassis was decided on to be a circular shape to keep the mathematical as simple as possible with the beneficial of circular based chassis. However, to adjust the changes of the wheels, side of the chassis were cut from the sides where the wheels are placed. So, it is not a full circular shape in this form.

Without facing placement problems in the chassis, the chassis is designed in a larger scale for the components. By this way if needed, additional components will be able to fit in a comfortable way.

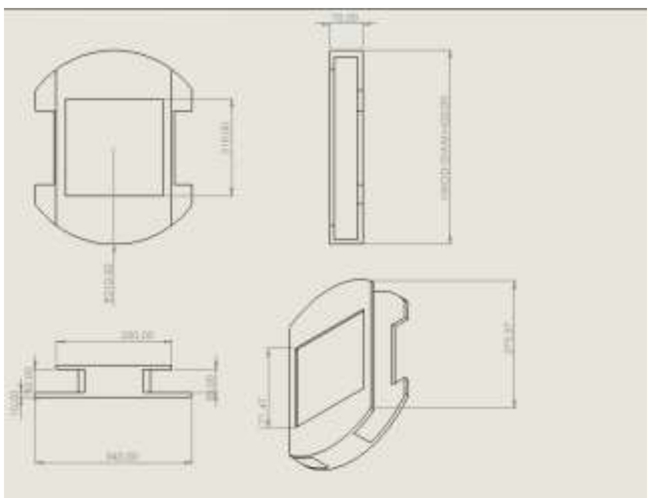

FIGURE 9. Technical Drawing from Top of the Chassis

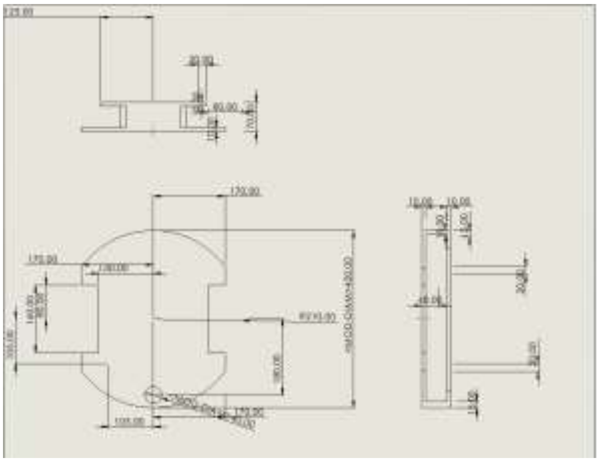

FIGURE 10. Technical Drawing from Bottom of the Chassis

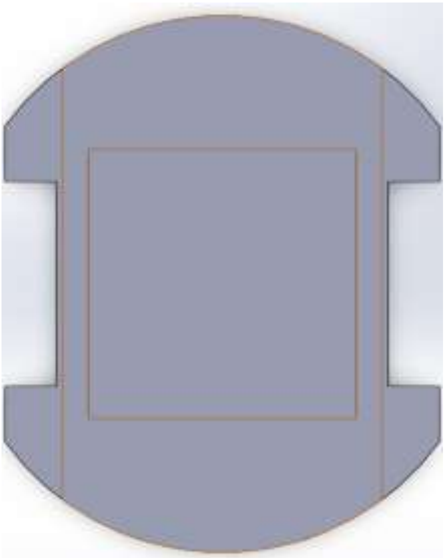

FIGURE 11. Top View of the Chassis

FIGURE 12. Bottom View of the Chassis

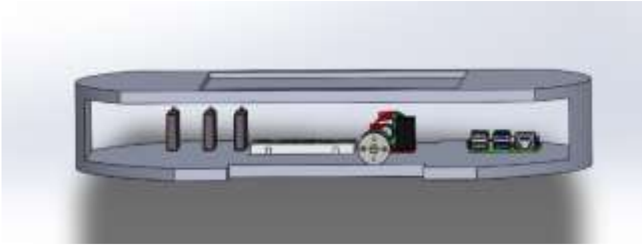

FIGURE 13. Side View of the Chassis with Components

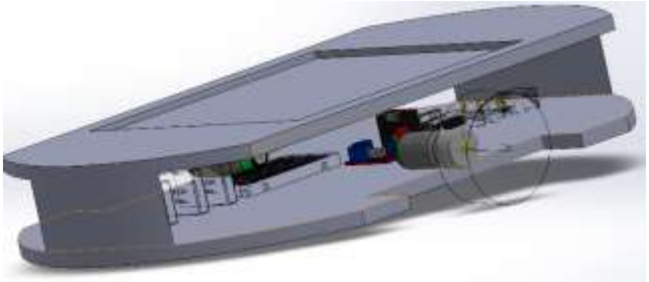

FIGURE 14. View of the Chassis with Small Wheel

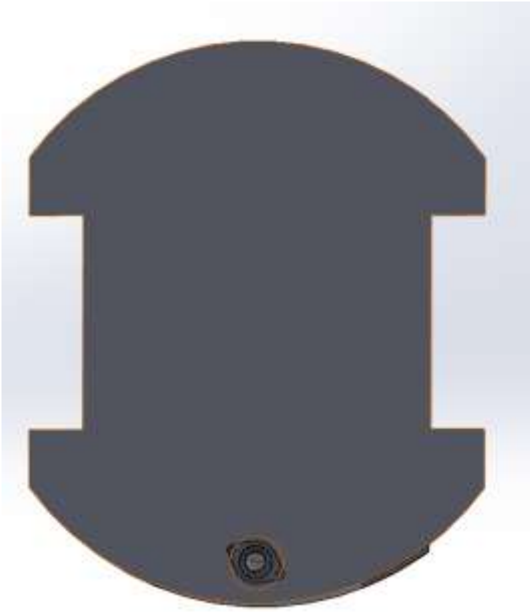

FIGURE 15. Bottom View of the Chassis with Ball Caster

b) Wheel

In this project there are two options of the wheels. One option is small wheel with 7.5-cm height and the other option is the large wheel with 12-cm height. Depending on the connections of the motor, one wheel among them will be selected to be used in the chassis. These are the wheels that spined by the motors, along with them there is a ball caster wheel which provides the stability of the system.

FIGURE 16. Small Wheel

FIGURE 17. Large Wheel

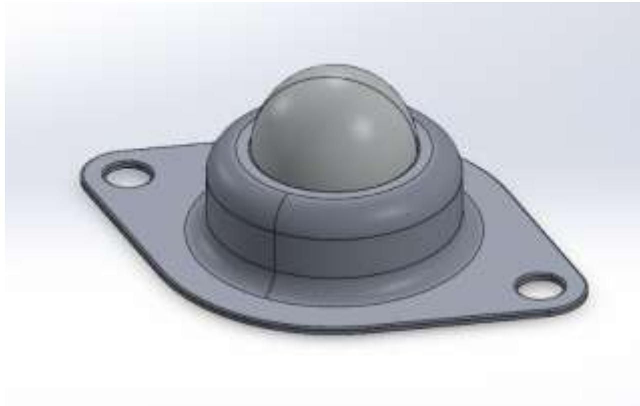

FIGURE 18. Ball Caster

B. CIRCUIT DIAGRAM

According to the pin configurations and power information of the terminals of the Raspberry Pi 4B module, driver, power supply, and motors, the circuit diagram was built. The drawn version of the circuit diagram of the motor control system is shown in the following figure.

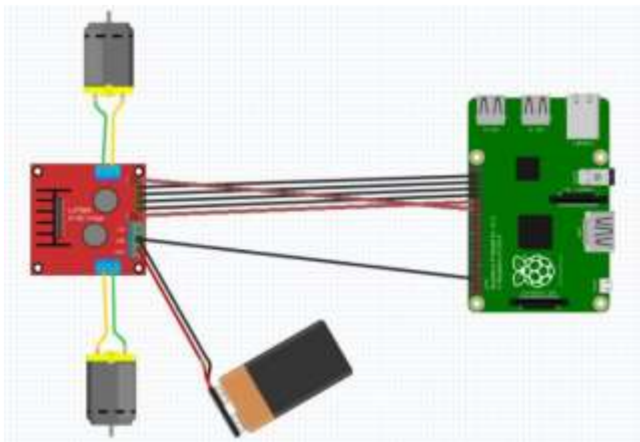

FIGURE 19. Circuit Diagram of System

C. RASPBERRY PI REMOTE CONNECTION

Since Raspberry Pi 4B can connect to Wi-Fi, the VNC method works for remote control of Raspberry.

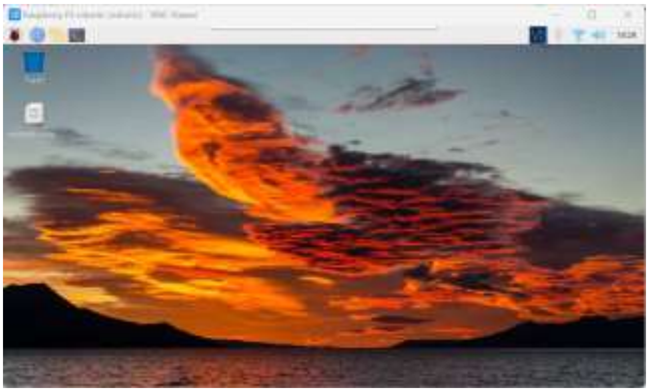

FIGURE 20. **Raspberry Pi Screen**

FIGURE 21. **VNC Method: Raspberry Pi Screen from Windows Computer**

D. GOOGLE COLLAB CODE

As mentioned in the materials section, normally the code run in raspberry pi via an ide called Geany. However, connecting raspberry pi with VNC can cause some delays. Therefore coding instead of connecting Raspberry remotely, the code was written in the Google collab and the final version of it copy pasted into Geany.

To code was written in Python 3. Written code control the 2 motors with input given from the keyboard. In the beginning, to use the GPIO pins on raspberry pi, RPI.GPIO library was imported. Then in the main part of the code, the pin numbers were defined. After that input-output arrangements were done for pins. In other words, GPIO pins were labeled as input or output. For motor speed control, as mentioned in the material and method part PWM is used. The ENA and ENB pins are labeled as PWM. After these pin initializations and labeling, some functions were written to write clean and readable code. Also, these functions will be useful for further progress and implementations. 4 Functions were written, these are:

- SingleMotorForwardMotion(inPin1, inPin2): This function takes the IN1 and IN2 pins of the corresponding motor as input. Then by giving the right digital voltage to pins, it makes the rotation direction of the motor to be forward.
- SingleMotorBackwardMotion(inPin1, inPin2): This function takes the IN1 and IN2 pins of corresponding motor as an input. Then by giving the right digital voltage to pins, it makes the rotation direction of the motor to be backward.
- SingleMotorIncreaseSpeed(powerPin, initialSpeed, increament): This function takes the PWM pin of the motor, initial speed, and increment amount of speed as input. Then by changing the duty cycle of this pin, it increases the speed of the rotation by the PWM duty cycle.
- SingleMotorDecreaseSpeed(powerPin, initialSpeed, increament): This function takes the PWM pin of the motor, initial speed, and decrement amount of speed as input. Then by changing the duty cycle of this pin, it decreases the speed of the rotation by the PWM duty cycle.

For the trial, the motor direction and speed were controlled by the keyboard. The button and corresponding missions were as follows:

- W: Rotate both motors in forward direction
- S: Rotate both motors in backward direction
- A: Rotate motors in different directions
- D: Rotate motors in the reverse direction of the case pressed A
- U: Increase the rotation speed of the motors
- Y: Decrease the rotation speed of the motors
- T: Stop both motors
- E: Stop the code

By clicking here, you can reach the video.

REFERENCES

[1] A. Sampathkrishna, "ArUco Maker based localization and Node graph approach to mapping." Accessed: Mar. 27, 2023. [Online]. Available: https://arxiv.org/pdf/2208.09355.pdf

[2] DANIEL SEKYERE-ASIEDU, "A RASPBERRY PI BASED SYSTEM TO HELP THE VISUALLY IMPAIRED AT HOME", M.S thesis, College of Eng. and Sc., Near East University, Lefkosa, 2018. [Online]. Available: http://docs.neu.edu.tr/library/6712561709.pdf

[3] Dhaouadi R, Hatab AA (2013) Dynamic Modelling of Differential-Drive Mobile Robots using Lagrange and Newton-Euler Methodologies: A Unified Framework. Adv Robot Autom 2: 107. doi: 10.4172/2168-9695.1000107

[4]: Habil, H. J., Al-Jarwany, Q. A., Hawas, M. N., & Mnati, M. J. (2022, June). Raspberry Pi 4 and Python Based on Speed and Direction of DC Motor. In 2022 4th Global Power, Energy and Communication Conference (GPECOM) (pp. 541-545). IEEE.

[5]: "In-Depth: Interface L298N DC Motor Driver Module with Arduino." Last Minute Engineers, 28 Nov. 2018, lastminuteengineers.com/l298n-dc-stepper-driver-arduino-tutorial/.

[6] J. CERKALA and A. JADLOVSKÁ, "Methodology for the Modelling of Mobile Robot with Differential Chassis," 16th Scientific Conference of Young Researchers – FEI TU of Košice, vol. 15, no. 3, May 2016.

[7] "OpenCV: ArUco marker detection (aruco module)," docs.opencv.org. https://docs.opencv.org/4.x/d9/d6d/tutorial_table_of_content_aruco.html#:~:text=ArUco%20markers%20are%20binary%20square

[8] "Pololu - 12V 37D Metal Gearmotors." Www.pololu.com, www.pololu.com/category/271/12v-37d-metal-gearmotors.

[9] R. K. Megalingam, S. Tantravahi, H. S. S. K. Tammana, N. Thokala, H. S. R. Puram, and N. Samudrala, "Robot Operating System Integrated robot control through Secure Shell(SSH)," IEEE.

[10] R. Mathew and S. S. Hiremath, "Trajectory Tracking and Control of Differential Drive Robot for Predefined Regular Geometrical Path," *Procedia Technology*, vol. 25, pp. 1273–1280, 2016, doi: https://doi.org/10.1016/j.protcy.2016.08.221.

[11] R. Siegwart, Illah Reza Nourbakhsh, and Davide Scaramuzza, Introduction to autonomous mobile robots. Cambridge, Mass.: Mit Press, 2011.

[12]:"Raspberry Pi Documentation - Raspberry Pi Hardware." Www.raspberrypi.com, www.raspberrypi.com/documentation/computers/raspberry-pi.html.

[13]: Raza, K. M., Kamil, M., & Kumar, P. (2016). Speed Control of DC Motor by using PWM. International Journal of Advanced Research in Computer and Communication Engineering, 5(4).

[14] V. Vujovic, B. Perisic, N. Davidović, M. Maksimovic, and V. Milosevic, "Raspberry Pi as Internet of Things hardware: Performances and Constraints," Jul. 2014, Available: https://www.researchgate.net/publication/272175660_Raspberry_Pi_as_Internet_of_Things_hardware_Performances_and_Constraints

[15] Virtual network computing. IEEE Internet Computing, 2(1), 33–38 | 10.1109/4236.656066 Richardson, T., Stafford-Fraser, Q., Wood, K. R., & Hopper, A. (1998). Virtual network computing. IEEE Internet Computing, 2(1), 33–38. doi:10.1109/4236.656066