

# XÂY DỰNG THƯ VIỆN CHO MOBILE ROBOTIC BÁM QUỠ ĐẠO TRONG MÔI TRƯỜNG CÓ VẬT CẢN BUILDING A LIBRARY FOR MOBILE ROBOT TRACTION IN ENVIRONMENTS WITH OBSTACLES

LÊ HỮU TOÀN<sup>1,a</sup>, CHÂU HUỖNH MINH<sup>2</sup>, NGUYỄN BÁCH<sup>2</sup>,  
TRẦN VĨNH PHÚC<sup>1</sup>

<sup>1</sup>Trường Đại học Sư phạm Kỹ thuật Vĩnh Long

<sup>2</sup>Sinh viên khoa KTCN Cơ khí

<sup>a</sup>Tác giả liên hệ: toanlh@vlute.edu.vn

*Nhận bài(Received): 15/4/2025; Phản biện (Reviewed):07/5/2025; Chấp nhận (Accepted):03/6/2025*

## TÓM TẮT

Lập kế hoạch đường đi và tránh va chạm là hai lý thuyết cốt lõi, đóng vai trò quan trọng trong việc thiết kế và phát triển các hệ thống robot di động. Trong nghiên cứu này, phần mềm NI LabVIEW được lựa chọn để triển khai bộ xử lý điều khiển thời gian thực, nhằm thực hiện chức năng tránh chướng ngại vật một cách hiệu quả và linh hoạt. Bài báo đề xuất một ứng dụng điều khiển tự động robot di động, với chương trình điều khiển được xây dựng trên nền tảng LabVIEW. Hệ thống được thiết kế nhằm thực hiện các chức năng điều khiển và giám sát chuyển động của robot theo thời gian thực, đáp ứng yêu cầu vận hành trong môi trường có chướng ngại vật. Hiệu quả của giải pháp được đánh giá thông qua các thí nghiệm thực tế. Kết quả thực nghiệm cho thấy hệ thống vận hành ổn định, đạt độ chính xác cao với sai số nhỏ, qua đó thấy được hiệu quả của phương pháp được đề xuất.

**Từ khóa:** Robot di động, Labview, lập kế hoạch đường đi

## ABSTRACT

*Path planning and obstacle avoidance are fundamental topics in the field of mobile robotics, forming the basis for autonomous navigation in dynamic environments. This paper presents a real-time control application developed using NI LabVIEW to execute effective obstacle avoidance for a mobile robot. The control system enables real-time monitoring and autonomous navigation by integrating sensor feedback into the motion planning process. Experimental results demonstrate the stability, reliability, and high precision of the system with minimal error, validating the feasibility and efficiency of the proposed method.*

**Keywords:** Mobile Robotics, Labview, path planning

## 1. ĐẶT VẤN ĐỀ

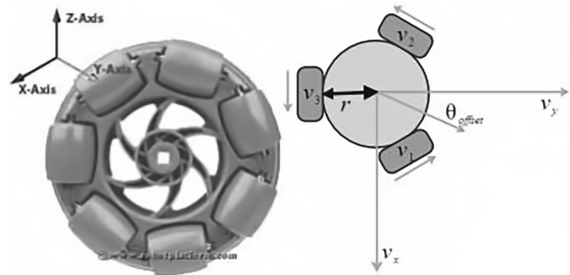
Lĩnh vực robot ngày càng phát triển mở ra xu hướng tiếp cận mới trong tương lai. Các robot có khả năng hoạt động độc lập hoặc lấy con người làm trung tâm, hỗ

trợ và tương tác với con người. Để đạt được điều đó, robot cần có khả năng học hỏi từ môi trường và phản hồi linh hoạt với các tín hiệu từ người dùng [1]. Một trong những yêu cầu quan trọng của robot di

động là khả năng tự động điều hướng đến vị trí mục tiêu, đồng thời tránh va chạm với các chướng ngại vật trong môi trường hoạt động cần sự tích hợp của nhiều cảm biến, tính toán và điều khiển [2]. Các yếu tố cơ bản như đặc tính cơ cấu chấp hành, điều khiển và lập kế hoạch đường đi là những nền tảng cốt lõi trong thiết kế hệ thống robot [3]. Bộ công cụ LabVIEW Robotics cùng với LabVIEW Vision Toolkit của hãng National Instruments cung cấp nền tảng phần mềm và phần cứng hỗ trợ các kỹ sư và nhà nghiên cứu trong việc thiết kế, mô phỏng và triển khai các hệ thống robot nhúng cho mục đích điều khiển, thử nghiệm và giám sát [4]. Và việc ứng dụng cảm biến Sharp IR, Ultrasonic trong robotic ngày càng rộng rãi [5]. Các robot hiện đại thường được trang bị cảm biến siêu âm nhằm phát hiện chướng ngại vật và điều chỉnh hướng đi để hoàn thành nhiệm vụ [6]. Bài báo trình bày ứng dụng phần mềm NI LabVIEW để xây dựng hệ thống điều khiển robot có khả năng xác định và duy trì khoảng cách với chướng ngại vật dựa trên tín hiệu cảm biến nhằm tránh va chạm và hỗ trợ điều hướng robot trong môi trường có chướng ngại vật.

## 2. Tính toán và thiết kế Mobile robotic

Robot được thiết kế với 3 bánh xe omni và các động cơ DC 12V của Studica, mỗi bánh xe cách nhau 120 độ, như trong Hình 1. Tốc độ động cơ được điều khiển bằng phương pháp PID để bám theo tường. Robot di chuyển dọc theo viền của đối tượng với một khoảng cách nhất định. OMR được sử dụng với ba bánh xe omni. Phương trình động học hệ thống Omni 3 bánh nhằm mục đích hiểu đặc điểm chuyển động của robot omni [7]



Hình 1: Mô hình để robot

Xét vận tốc  $V$  gồm 2 thành phần  $V_x$  và  $V_y$ . Chiếu lần lượt 2 thành phần này lên tọa độ của bánh xe ta có

Phương  $V_1$  ta có:

$$V_1 = -V_x \cos(30^\circ) - V_y \sin(30^\circ) \quad (1)$$

Phương  $V_2$  ta có:

$$V_2 = V_x \cos(30^\circ) - V_y \sin(30^\circ) \quad (2)$$

Phương  $V_3$  ta có:

$$V_3 = V_y \quad (3)$$

Thay giá trị lượng giác của  $\sin(30^\circ)$ ,  $\cos(30^\circ)$  vào (1), (2) và (3) ta được hệ phương trình (4)

$$\begin{cases} V_1 = -\frac{\sqrt{3}}{2}V_x - \frac{1}{2}V_y \\ V_2 = \frac{\sqrt{3}}{2}V_x - \frac{1}{2}V_y \\ V_3 = V_y \end{cases} \quad (4)$$

Qui ước chiều góc dương của Robot là ngược chiều kim đồng hồ. Góc 0 độ nằm trên trục  $Ox$ . Ta có khi Robot di chuyển theo phương  $V$  (góc  $\alpha$ ).

$$\begin{aligned} V_x &= \cos(\alpha) \cdot V \\ V_y &= \sin(\alpha) \cdot V \end{aligned} \quad (5)$$

Từ (4) và (5) ta có hệ phương trình (6)

$$\begin{cases} V_1 = -\frac{\sqrt{3}}{2}\cos(\alpha) - \frac{1}{2}\sin(\alpha) \\ V_2 = \frac{\sqrt{3}}{2}\cos(\alpha) - \frac{1}{2}\sin(\alpha) \\ V_3 = \sin(\alpha) \end{cases} \quad (6)$$

Áp dụng bộ điều khiển PID vào hệ thống điều khiển giúp giảm giá trị sai số trong điều khiển. Phương trình (7) biểu diễn tín hiệu lỗi từ điều khiển PID dưới dạng tín hiệu PWM để điều chỉnh tốc độ quay của động cơ. [8]

$$u = Kp(e)t + Ki \int_0^t e(t)dt + Kd \frac{de(t)}{dt} \quad (7)$$

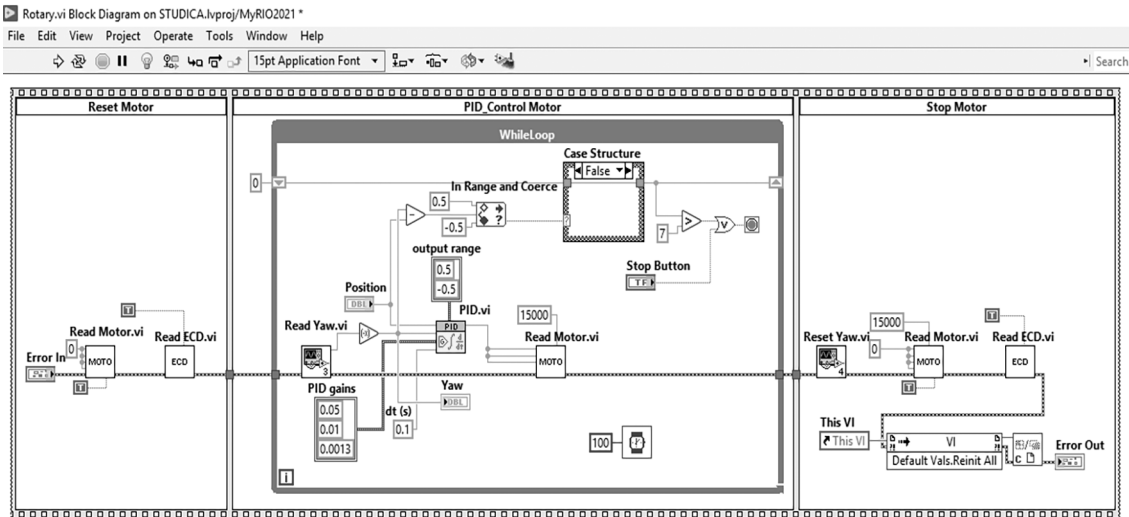
Trong đó,  $u$  là tín hiệu điều khiển,  $Kp$  là hằng số tỉ lệ,  $Ki$  là hằng số tích phân,  $Kd$  là hằng số đạo hàm và  $e(t)$  là tín hiệu lỗi. Tín hiệu lỗi được thu nhận từ sự chênh lệch giữa giá trị đặt và kết quả đọc từ cảm biến. Phương pháp này được sử dụng để khắc phục lỗi từ kết quả vị trí, của hệ tọa độ  $x$ ,  $y$  và góc của hệ thống thu được từ cảm biến góc quay.

### 3. Xây dựng chương trình Mobile Robotic

#### 3.1 Chương trình xoay góc

Nền tảng NI LabView là ngôn ngữ lập trình đồ họa bao gồm hai cửa sổ là sơ đồ khối để viết tất cả các mã trong chương trình và bảng điều khiển phía trước để hiển thị tất cả đầu ra bằng cách cung cấp quyền kiểm soát của người dùng. Cảm biến được kết nối vật lý với myRIO để truyền dữ liệu đến FPGA. Phần mềm cấu hình FPGA để truyền dữ liệu qua bus myRIO đến bộ xử

lý thời gian thực. Trên robot chương trình mong muốn được triển khai trên FPGA để xem dữ liệu từ các cảm biến khoảng cách và điều chỉnh động cơ truyền động để robot tránh chướng ngại vật. Để chứng minh chức năng của robot tránh chướng ngại vật, chúng tôi đã đề xuất một sơ đồ khối được hiển thị trong hình 2. Cảm biến siêu âm liên tục thu thập dữ liệu về môi trường xung quanh chúng và tạo ra tín hiệu đầu ra. Từ những hàm ReadMotor, hàm ReadYaw, ... ta tiến hành đấu nối như (hình 2). Chương trình sẽ chạy tuần tự từng Flat (Reset motor, Rotary, Stop Motor). Flat Reset motor: Reset Encoder về 0 (tất cả mọi giá trị của động cơ về giá trị 0). Flat PID Control Motor: Tổng hợp những khối lệnh kết nối với nhau để điều khiển Robot xoay đúng góc quy định. Hàm ReadYaw.vi: để đọc giá trị cảm biến góc Yaw. Giá trị sẽ được đọc và cập nhật liên tục theo chu kỳ của vòng lặp While (100ms). Giá trị truyền ra sẽ được đưa vào Process variable của PID. Tần số hoạt động của động cơ là 15000 hz do không cần momen lớn mà cần độ chính xác cao nên tần số cao giúp động cơ quay với độ mịn và chính xác hơn. Tín hiệu dừng: khi giá trị Setpoint đưa vào và giá trị cảm biến đọc được thông qua In Range and Coerce (giới hạn của độ lệch là từ -0.5 đến 0.5). Khi độ lệch được thỏa điều kiện thì tín hiệu sẽ truyền vào Case Structure và sẽ đếm dần đến khoảng thời gian (700ms) và Stop While Loop. Flat Stop Motor: Sau khi có tín hiệu dừng vòng lặp While thì chương trình sẽ chuyển qua Flat Stop motor. Nhiệm vụ Flat này là Reset góc Yaw, dừng động cơ và Reset Encoder. Đồng thời xóa tất cả dữ liệu đưa vào, ra cũng như thông số cảm biến của chương trình.

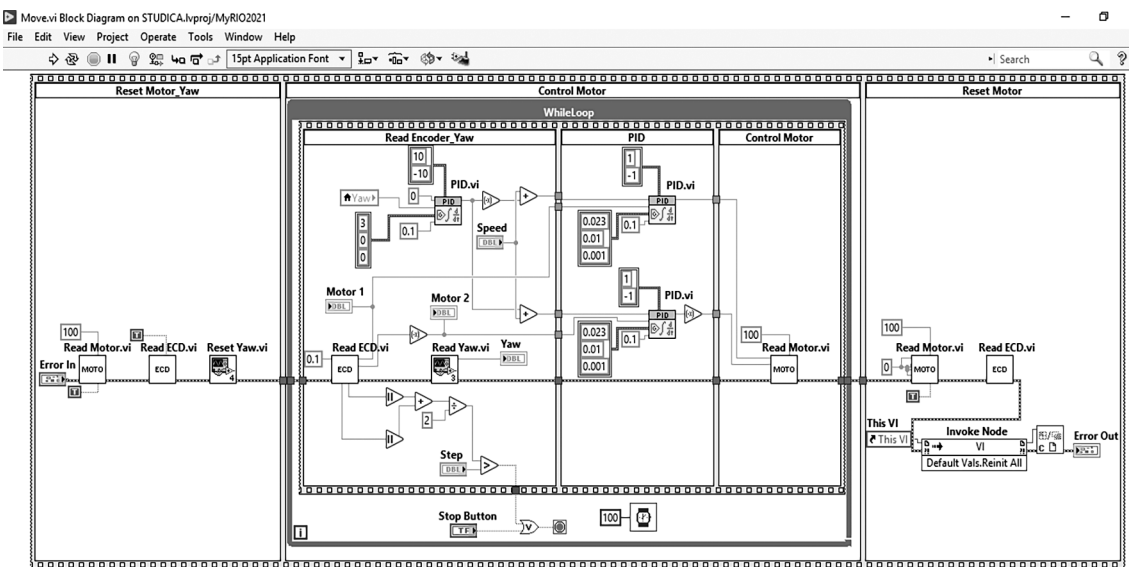


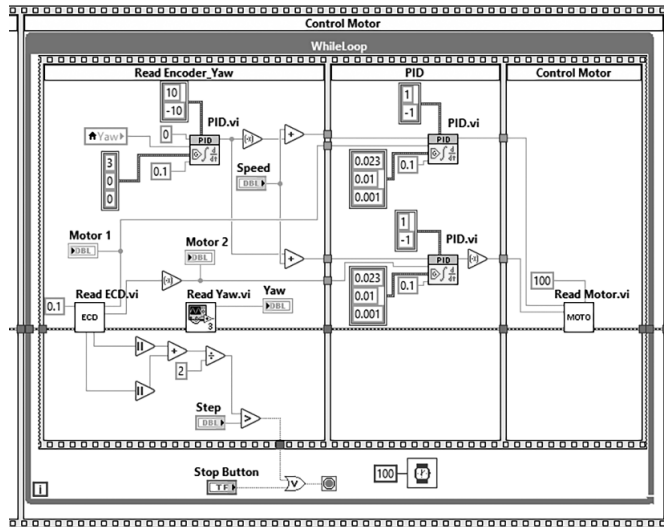
Hình 2: Input\_Output trên Block Diagram và Front Panel

### 3.2 Chương trình di chuyển theo quỹ đạo thẳng

Tương tự hàm Rotary chương trình sẽ hoạt động tuần tự từng Flat từ trái sang phải (hình 3). (Reset Motor\_Yaw, Control Motor, Reset Motor). Flat Reset Motor\_Yaw: Reset Encoder và cảm biến góc quay về giá trị 0. Flat Control Motor: sẽ lặp lại liên tục cho đến khi có tín hiệu dừng nhờ vào vòng lặp While. Trong vòng lặp While có thêm Flat Sequence Structure gồm 3 Flat sẽ hoạt động tuần tự từ trái sang phải (Read

Encoder\_Yaw, PID, Read Motor). Read ECD Ở khối Read ECD.vi sẽ trả về 2 dữ liệu đó là vận tốc của động cơ trong khoảng thời gian dt và tổng quãng đường mà Robot đi được từ lúc bắt đầu di chuyển đến khi dừng Robot. Qua đó ta sẽ lấy trung bình cộng của hai động cơ 1 và 2 để làm tín hiệu dừng. Nếu quãng đường trung bình Robot di chuyển lớn hơn quãng đường mà ta xét thì sẽ thoát khỏi vòng lặp While Loop. Sau khi thoát khỏi vòng lặp While Loop sẽ đến Flat Reser Motor.



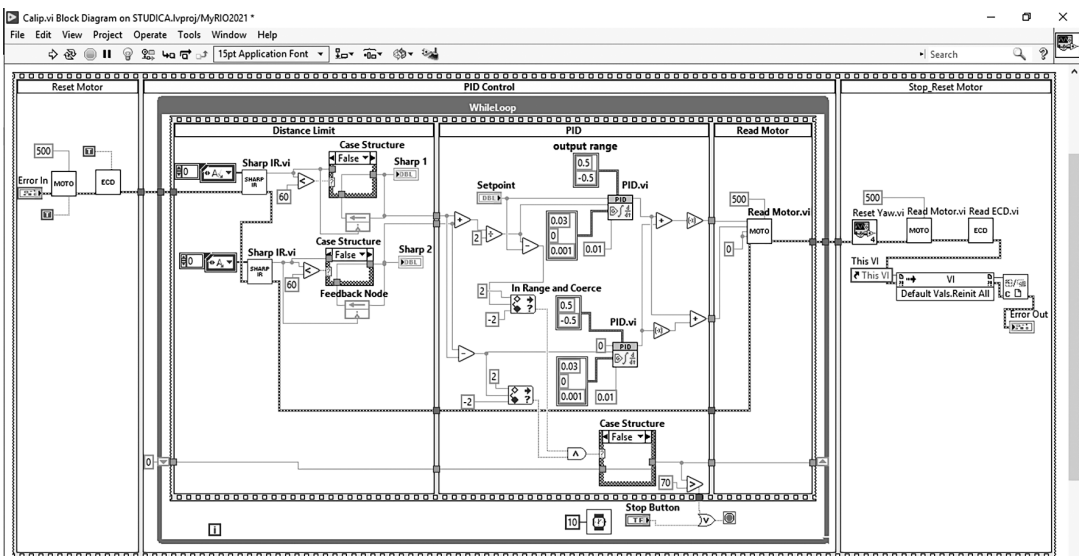


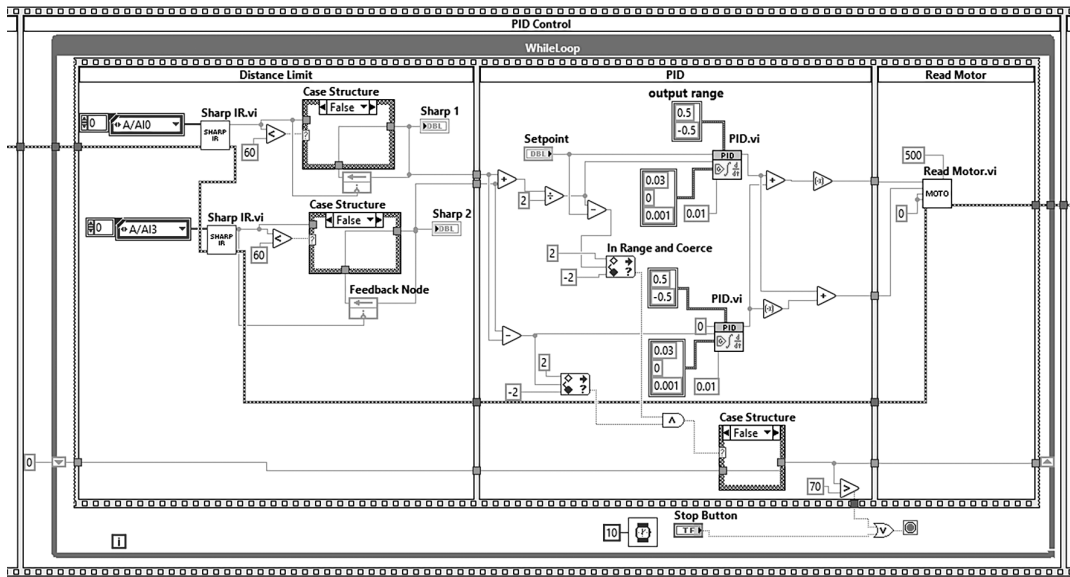
Hình 3: Input\_Output hàm di chuyển quỹ đạo thẳng trên Block Diagram

### 3.3 Chương trình calip với mặt phẳng

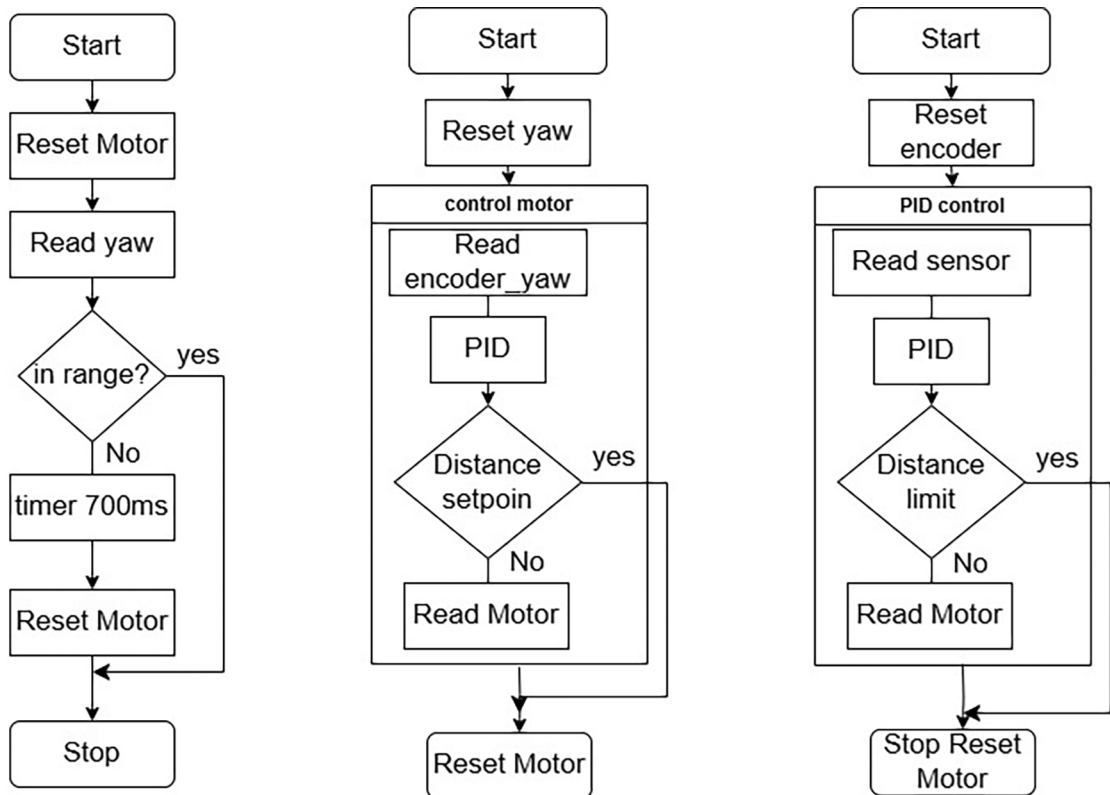
Hàm Calip được thiết kế giúp căn chỉnh lại góc của Robot so với mặt phẳng tường. Đặt giá trị cho Setpoint khi đó Robot sẽ tính toán khoảng cách hiện tại so với mặt phẳng tường và tính toán hệ số sai lệch giữa khoảng cách tường và Setpoint mà cho ra tốc độ và dừng Robot tương đối đúng so với khoảng cách mà ta đặt (hình 4). Hoạt động của hàm Calip cũng tương tự như hàm Move thực hiện tuần tự từng Flat từ trái sang phải gồm có 3 Flat chính: Reset

Motor, PID Control, Stop\_Reset Motor. Flat Reset Motor: Reset Motor và Encoder về 0. Do hàm Calip không sử dụng cảm biến góc quay nên không cần Reset cảm biến. Flat PID Control: có vòng lặp While sẽ lặp lại liên tục với thời gian là 10ms. Trong vòng lặp While Loop có 3 Flat thực hiện tuần tự từ trái sang phải là: Distance Limt, PID, Read Motor. Flat Stop\_Reset Motor: Khi Robot đã đạt được khoảng cách và bộ đếm lớn hơn giá trị ta đặt trước thì sẽ thoát vòng lặp While Loop và tiến hành Stop động cơ và Reset Encoder và góc Yaw.





Hình 4: Input\_Output hàm Calip mặt phẳng trên Block Diagram



flow chart chương trình xoay góc

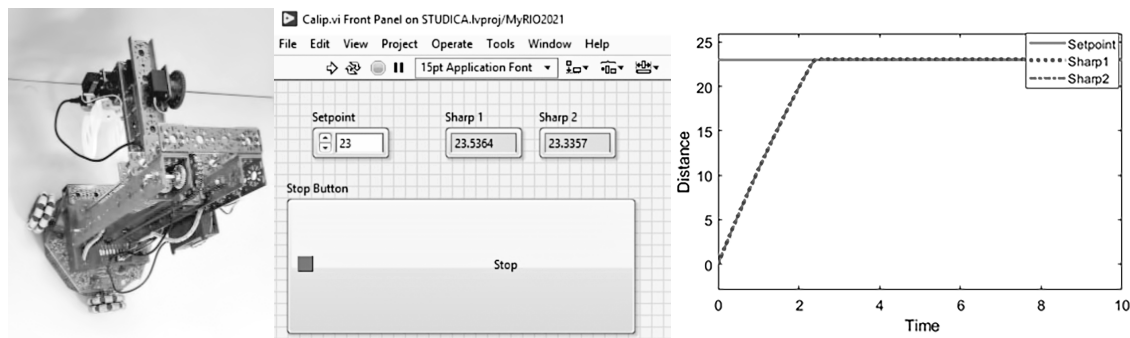
flow chart chương trình chạy thẳng

flow chart chương trình calip với mặt phẳng

Hình 5 : giải thuật chương trình xoay, chạy thẳng, calip

## 4. Kết quả thực nghiệm

### 4.1 Khảo sát khả năng Calip của robot với mặt phẳng phía trước

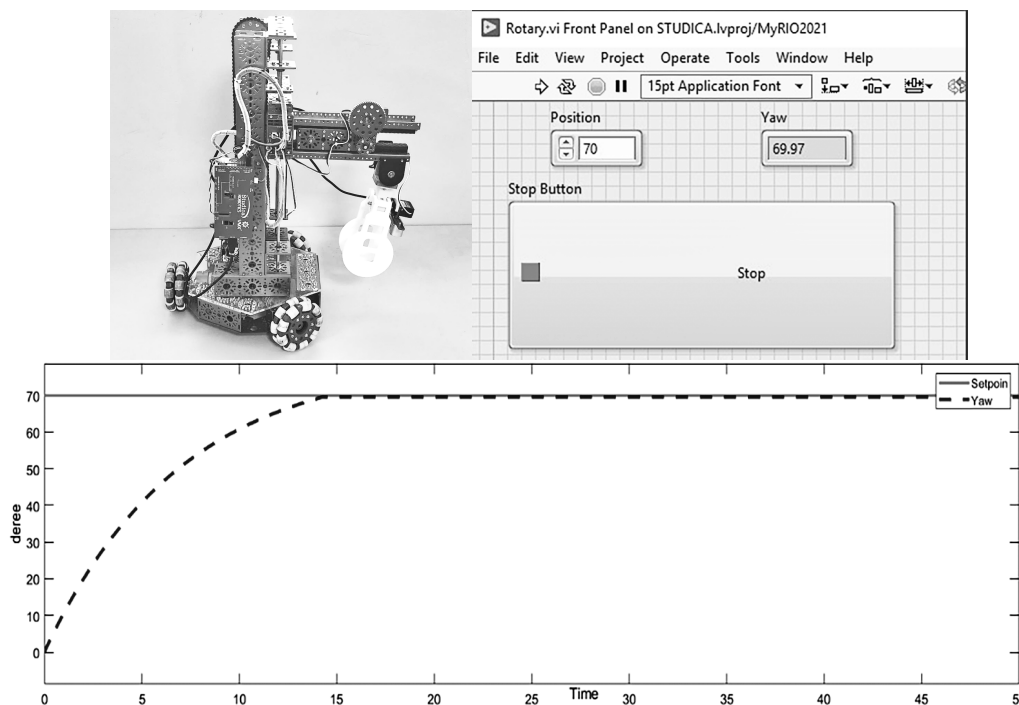


Hình 6: Thực nghiệm hàm Calip mặt phẳng

Đặt Setpoint là 23cm, xe tự động Calip với khoảng cách tương đối bằng 23cm. Sai số mà cảm biến nhận được hiện tại là gần bằng 0.5cm. độ lệch giữa 2 cảm biến khoảng gần bằng 0.3cm. Sai số cảm biến

nhận được không quá lớn, trong khoảng cho phép từ 0,5 đến 1cm.

### 4.2 Khảo sát robot quay theo góc alpha

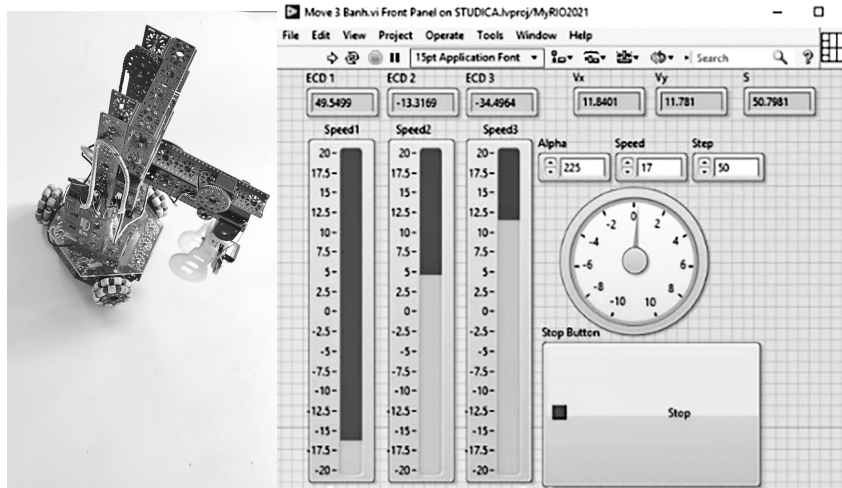


Hình 7: Thực nghiệm hàm Rotary

Nhập Setpoint: góc quay của Robot là 70 độ, sau khi thực hiện cảm biến đọc được góc Yaw khi dừng xe là 69.97 độ. Kết quả cho ta thấy sai số của cảm biến nhận trả về

so với Setpoint nhỏ hơn 0.5 độ.

### 4.3 Khảo sát robot di chuyển theo quỹ đạo chạy thẳng



**Hình 8: Thực nghiệm hàm di chuyển thẳng**

Yêu cầu đặt ra ở hàm này là đảm bảo Robot di chuyển đúng góc độ ta xét, di chuyển theo quỹ đạo thẳng và dừng đúng khoảng cách (quãng đường) ta đặt. Truyền vào thông số  $\text{Alpha} = 225^\circ$ ,  $\text{Speed} = 17$  và  $\text{Step} = 50$  cm. Qua thực nghiệm thì robot đã đi đúng góc độ xét, tương đối đúng tốc độ cũng như quãng đường sai lệch không quá vượt quá 3cm.

## 5. KẾT LUẬN

Nghiên cứu đã xây dựng thành công

chương trình, thư viện điều khiển robot di động sử dụng phần mềm NI LabVIEW, cho phép điều hướng và tránh chướng ngại vật theo thời gian thực với độ chính xác cao và sai số nhỏ. Kết quả thực nghiệm khẳng định tính khả thi, ổn định và tiềm năng ứng dụng của giải pháp trong môi trường thực tế. Hướng phát triển tiếp theo là tích hợp các thuật toán trí tuệ nhân tạo nhằm nâng cao khả năng nhận dạng và ra quyết định điều hướng.

## TÀI LIỆU THAM KHẢO

- [1]. Nahavandi, Saeid, et al. "A comprehensive review on autonomous navigation". ACM Computing Surveys, 2022.
- [2]. Dirik, M., Kocamaz, A. F., & Donmez, E. "Implementation of fuzzy controller for mobile robot navigation on NI's embedded-FPGA robotic platform". Computer Science, 4(2), 80-87. 2019
- [3]. Xia, X., Li, T., Sang, S., Cheng, Y., Ma, H., Zhang, Q., & Yang. "Path planning for obstacle avoidance of robot arm based on improved potential field method". Sensors, 23(7), 3754. 2023.
- [4]. Zhang, Y., Li, Y., Gu, X., Liu, H., Zhang, Y., & Hu, W. "Laser spot image acquisition and processing based on LabVIEW". Optik, 185, 505-509. 2019
- [5] Mary B. Alatise, Student, and Gerhard P. Hancke, Senior Member, "Navigation of Mobile Robots Using Infrared Sensors". Journal of Robotics and Automation, vol. 11, no. 4, pp. 123-130, Dec 2020.

- [6]. Pandey, A., Pandey, S., & Parhi, D. ‘’Mobile robot navigation and obstacle avoidance techniques: A review’’. *Int Rob Auto J*, 2(3), 00022. 2017.
- [7]. Ma Soe Soe Htay, Khaing Win Phyu, Su Yadanar, Wut Yi Win. ‘‘Kinematics and Control A Three-wheeled Mobile Robot with Omnidirectional Wheels’’. *ICCR International Conference on Communication and Computer Research* October 23 (Mon.), 2023: Seoul, Korea.
- [8] Hendi Purnata, Syahrul Ramadan, Muhammad Arif Hidayat, Irvan Maulana, ‘‘PID Control Schematic Design for Omnidirectional Wheel Mobile Robot Cilacap State of Polytechnic’’. *Journal of Telecommunication Network (Jurnal Jaringan Telekomunikasi)* Vol.12, No.2 (2022).