

# Đánh giá sai số điều khiển và sai số định vị trong quá trình xây dựng bản đồ 2D của robot di động

1<sup>st</sup> Nguyễn Quốc Huy

*School of Electrical and Electronic Engineering  
Hanoi University of Science and Technology  
Hanoi, Vietnam  
huy.nq223709@sis.hust.edu.vn*

2<sup>nd</sup> Trần Hồng Long

*School of Electrical and Electronic Engineering  
Hanoi University of Science and Technology  
Hanoi, Vietnam  
long.th233511@sis.hust.edu.vn*

3<sup>rd</sup> Đào Tuấn Anh

*School of Electrical and Electronic Engineering  
Hanoi University of Science and Technology  
Hanoi, Vietnam  
anh.dt223748@sis.hust.edu.vn*

4<sup>th</sup> Lưu Tùng Hải

*Department of Control Engineering and Information Technology  
Budapest University of Technology and Economics  
Budapest, Hungary  
haitungluu@edu.bme.hu*

5<sup>th</sup> Nguyễn Hữu Phát

*School of Electrical and Electronic Engineering  
Hanoi University of Science and Technology  
Hanoi, Vietnam  
phat.nguyenhuu@hust.edu.vn*

**Abstract**—Trong các hệ thống robot di động tự hành, độ chính xác của điều khiển và định vị quyết định trực tiếp đến chất lượng bản đồ và khả năng xác định vị trí của robot. Trong quá trình thực nghiệm với robot sử dụng động cơ DC Encoder, cảm biến quán tính IMU và hoạt động trên framework ROS2, ghi nhận sai lệch đáng kể giữa vận tốc điều khiển và vận tốc thực tế, cùng với sai số góc quay và sai số tích lũy về vị trí, dẫn đến bản đồ bị méo và giảm hiệu quả của các thuật toán SLAM và xác định vị trí. Bài báo đề xuất một phương pháp đánh giá và hiệu chỉnh sai số trong hệ thống định vị của robot di động trước khi sử dụng trong bài toán xây dựng bản đồ 2D. Phương pháp này kết hợp hiệu chỉnh vận tốc tuyến tính với giảm sai số góc, giúp giảm đáng kể sai lệch vị trí và hướng, qua đó nâng cao độ chính xác bản đồ và độ tin cậy của quá trình định vị trong môi trường thực.

**Keywords**—Mobile Robot, Odometry, IMU Calibration, PID, ROS2, 2D Mapping, Localization

## I. GIỚI THIỆU

Trong các ứng dụng robot di động, việc định vị chính xác đóng vai trò then chốt để đảm bảo khả năng điều hướng và thực hiện nhiệm vụ. Thông thường, robot sử dụng bộ mã hóa vòng quay gắn trên bánh xe (encoder) và cảm biến quán tính (IMU) cho phép ước lượng quỹ đạo dựa trên phương pháp dead reckoning. Tuy nhiên, dữ liệu từ các cảm biến này chịu sai số tích lũy theo thời gian. Cụ thể, tín hiệu encoder bị ảnh hưởng bởi hiện tượng trượt bánh, hao mòn lốp và thay đổi bán kính bánh xe do tải trọng thay đổi, khiến khoảng cách di chuyển ước tính không phản ánh chính xác thực tế [1]. Tương tự, IMU tích hợp gia tốc và góc quay cũng không hoàn hảo

do đo nhiễu ngẫu nhiên và bias, dẫn đến sai số vị trí và góc ngày càng lớn khi tích phân dài hạn [2].

Trong nhiều ứng dụng, robot di chuyển trong môi trường thiếu tín hiệu định vị tuyệt đối (ví dụ trong nhà hoặc khu vực GPS-denied), do đó SLAM phải dựa vào các đo tương đối từ encoder, IMU, camera hoặc LiDAR. Trong ngữ cảnh này, các sai số cảm biến và odometry sẽ tích lũy, làm giảm độ chính xác của cả bản đồ và vị trí ước lượng nếu không có cơ chế sửa lỗi. Nhiều công trình chỉ ra rằng drift trong SLAM chủ yếu do trượt bánh xe và nhiễu IMU gây ra [3]. Để giảm drift, hệ thống SLAM thường áp dụng kỹ thuật phát hiện vòng lặp (loop closure) và tối ưu hóa toàn cục dựa trên đồ thị hoặc bộ lọc Kalman mở rộng để điều chỉnh sai số tích lũy.

ROS2, thế hệ mới của hệ điều hành robot, ngày càng được ứng dụng trong xử lý dữ liệu cảm biến và kiểm soát robot. So với ROS đời đầu, ROS2 sử dụng chuẩn giao tiếp DDS, hỗ trợ thời gian thực và khả năng phân tán tốt hơn, từ đó đáp ứng yêu cầu về hiệu năng trong các hệ thống tự chủ hiện đại [4]. Nhiều nghiên cứu cho thấy ROS2 phù hợp cho các kiến trúc robot mô-đun, cho phép tích hợp linh hoạt các thành phần cảm biến và thuật toán (ví dụ encoder, IMU, camera) vào hệ thống thống nhất.

Vì vậy, để đảm bảo độ chính xác cao trong SLAM và định vị, cần thiết phải áp dụng các phương pháp hiệu chỉnh và xử lý trước dữ liệu cảm biến. Đối với encoder, các bước hiệu chuẩn như xác định chính xác bán kính bánh xe và khoảng cách giữa các bánh (track width) là cơ bản. Đối với IMU, phân tích Allan deviation giúp xác định các thành phần nhiễu và bias để thiết lập bộ lọc phù hợp, từ đó giảm thiểu drift [2].

Từ đó, đóng góp của bài nghiên cứu này bao gồm:

- **Hiệu chuẩn đáp ứng vận tốc:** Bài báo đề xuất quy trình hiệu chỉnh vận tốc tuyến tính của robot thông qua bộ điều khiển PID (Proportional–Integral–Derivative) [?]. Phương pháp này cho phép giảm thiểu đáng kể sai lệch giữa vận tốc đặt và vận tốc thực tế, nhờ khả năng bù sai số theo thời gian và hạn chế dao động trong đáp ứng động học.
- **Phân tích và hiệu chuẩn IMU:** Bài báo nghiên cứu sử dụng phân tích độ lệch Allan (Allan deviation) để xác định nhiễu và bias của cảm biến IMU. Dựa trên kết quả này, quá trình hiệu chuẩn (Calibration) được thực hiện nhằm bù trừ sai lệch và giảm trôi góc (drift), qua đó cải thiện độ ổn định và tăng độ chính xác trong xác định vị trí của robot.
- **Đánh giá sai số định vị:** Cuối cùng, các phương pháp được tích hợp nhằm đánh giá tính khả thi của hệ thống định vị trong môi trường trong nhà, nơi tín hiệu GPS bị hạn chế. Hệ thống sử dụng dữ liệu từ encoder và IMU để ước lượng vị trí và hướng chuyển động của robot. Kết quả cho thấy hệ thống đạt độ tin cậy và độ chính xác cao, tạo nền tảng cho các ứng dụng định vị trong nhà và xây dựng bản đồ 2D tự động.

## II. NGHIÊN CỨU LIÊN QUAN

Nhiều công trình đã tập trung vào sai số của odometry và IMU cho robot di động. Các nghiên cứu về odometry thường phân tích sai số hệ thống và ngẫu nhiên của encoder. Ví dụ, Fazekas [5] đề xuất một mô hình động của odometry có tính đến sự thay đổi động của bán kính bánh xe và kết hợp lọc Kalman để ước lượng tham số; kết quả cho thấy sau khi hiệu chỉnh mô hình, sai số vị trí giảm gấp 5 lần. Tương tự, He [1] đã tổng hợp các yếu tố gây sai số odometry và sử dụng các phương pháp học máy để dự đoán và bù đắp sai số đó.

Về IMU, Suvorkin [2] phân tích đặc tính nhiễu và bias của các cảm biến MEMS-IMU ở nhiều cấp độ, sử dụng biểu đồ Allan để xác định các thành phần noise và bias. Kết quả cho thấy IMU chất lượng cao có mức nhiễu thấp hơn và độ trôi ổn định hơn, từ đó cải thiện độ chính xác của định vị kết hợp GNSS/IMU [2]. Một số kỹ thuật xử lý khác, chẳng hạn Zero-Velocity Update, cũng thường được sử dụng để giảm thiểu drift khi thiết bị đứng yên.

Về SLAM, Ghadimzadeh [3] tổng hợp các phương pháp SLAM đa cảm biến và nhấn mạnh tác động của sai số cảm biến đến drift. Họ ghi nhận rằng SLAM tích hợp IMU và LiDAR cần cơ chế phát hiện vòng lặp và tối ưu hóa đồ thị để hạn chế sai số tích lũy.

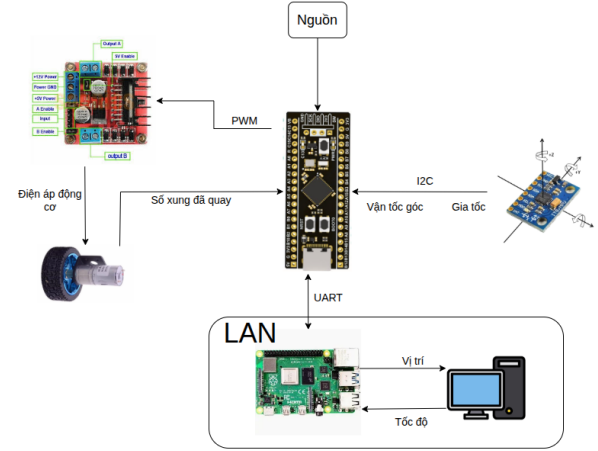
ROS2 cũng là chủ đề được quan tâm trong các nghiên cứu về hệ thống phân tán và tự động hóa robot. Khảo sát của Bonci [4] nêu bật khả năng của ROS2 trong tích hợp linh hoạt phần cứng và phần mềm đa dạng, đặc biệt trong các ứng dụng công nghiệp đòi hỏi thời gian thực cao. Nhiều nghiên cứu khác cho thấy ROS2 giảm đáng kể độ trễ truyền thông so với ROS1 và hỗ trợ tốt cho các hệ thống đa tác nhân.

Các công trình trước đây cho thấy sai số từ encoder và IMU có thể được giảm thiểu thông qua việc hiệu chuẩn và áp dụng bộ lọc thích hợp trong phương pháp [1], [2]. Đồng thời, nền

tảng ROS2 cung cấp khung xử lý dữ liệu và điều khiển linh hoạt, hỗ trợ tích hợp cảm biến và thuật toán SLAM ở mức cao. Nhiều phương pháp hiệu chỉnh cả extrinsic và intrinsic cho IMU và encoder đã được đề xuất nhằm nâng cao độ chính xác của hệ thống định vị.

## III. PHƯƠNG PHÁP NGHIÊN CỨU

### A. Cấu hình hệ thống thí nghiệm



Hình 1: Mô hình tổng quan hệ thống.

Hình 1 mô tả robot di động được thiết kế sử dụng hai động cơ DC có gắn encoder để đo số xung quay và tính toán vận tốc của từng bánh xe. Bộ cảm biến IMU MPU6050 được tích hợp để thu thập dữ liệu gia tốc tuyến tính và vận tốc góc của robot. Vi điều khiển STM32 đóng vai trò là bộ xử lý trung gian, thực hiện điều khiển động cơ thông qua tín hiệu PWM, đồng thời đọc tín hiệu encoder và giá trị gia tốc tuyến tính vận tốc góc từ IMU.

Các dữ liệu về vận tốc và góc quay được truyền tới Raspberry Pi thông qua giao tiếp UART để truyền tiếp thông tin đến máy tính chủ qua cơ chế DDS (Data Distribution Service) trong ROS2 [6]. Máy tính chủ đóng vai trò giám sát, hiển thị vị trí, vận tốc, và thực hiện gửi dữ liệu tốc độ phục vụ cho việc điều khiển từ xa.

Vận tốc tuyến tính của bánh xe được xác định dựa trên số xung encoder thu được trong một chu kỳ lấy mẫu [7]:

$$v = \frac{2\pi R \Delta n}{N \Delta t} \quad (1)$$

Trong đó:

- $v$ : Vận tốc tuyến tính của bánh xe (m/s)
- $R$ : Bán kính bánh xe (m)
- $\Delta n$ : Số xung đếm được trong chu kỳ lấy mẫu
- $N$ : Số xung encoder cho một vòng quay
- $\Delta t$ : Khoảng thời gian lấy mẫu (s)

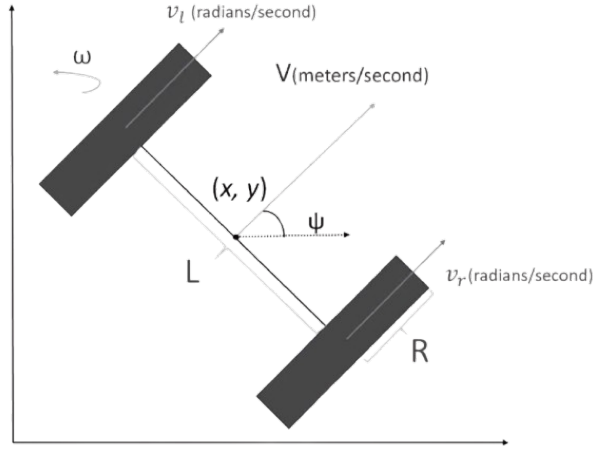
Cảm biến quán tính IMU (Inertial Measurement Unit) cung cấp dữ liệu bao gồm gia tốc tuyến tính ( $a_x, a_y, a_z$ ) và vận tốc góc ( $\phi, \theta, \psi$ ). Dựa trên mối quan hệ hình học giữa các trục cảm biến và vectơ trọng lực, góc nghiêng roll ( $\phi$ ), pitch ( $\theta$ )

và yaw ( $\psi$ ) có thể được ước lượng từ các dữ liệu gia tốc và vận tốc góc theo các công thức [8]:

$$\text{Pitch} = \theta = \arcsin\left(\frac{a_x}{g}\right), \quad \text{Roll} = \phi = \arctan\left(\frac{a_y}{a_z}\right) \quad (2)$$

$$\text{Yaw} = \psi(t) = \int_0^t \dot{\psi}(\tau) d\tau \quad (3)$$

Hệ thống robot di động sử dụng cấu trúc Differential-drive (hai bánh dẫn động độc lập), trong đó mỗi bánh có vận tốc lần lượt là  $v_r$  và  $v_l$ . Khi robot di chuyển, vị trí của robot trong mặt phẳng  $(x, y)$  và góc định hướng  $\psi$ , như hình dưới đây:



Hình 2: Mô hình động học của robot vi sai (Differential Drive Kinematic Model)

Từ giá trị vận tốc tuyến tính mỗi bánh  $v_r$  và  $v_l$  và giá trị góc quay quanh trục z ( $\psi$ ) được phản hồi về khi xe di chuyển, vị trí robot được cập nhật theo thời gian bằng mô hình động học vi sai sau [9]:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \frac{R}{2}(v_r + v_l) \cos \psi \\ \frac{R}{2}(v_r + v_l) \sin \psi \\ \frac{R}{L}(v_r - v_l) \end{bmatrix} \quad (4)$$

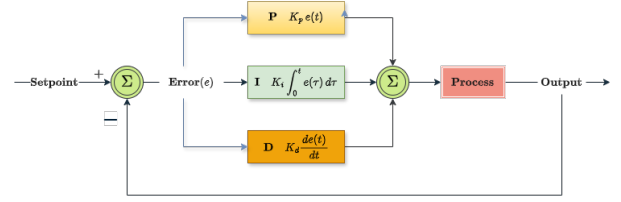
Trong đó:

- $\dot{x}, \dot{y}$ : Vận tốc theo trục  $x$  và trục  $y$
- $\dot{\psi}$ : Tốc độ quay theo trục  $z$
- $v_r, v_l$ : Vận tốc tương ứng của mỗi bánh
- $R$ : Bán kính của bánh xe
- $L$ : Khoảng cách giữa 2 bánh

Từ công thức trên có thể thấy bất kỳ sai lệch nào trong phép đo vận tốc và góc quay đều sẽ được tích lũy theo thời gian, dẫn đến sai số tích lũy trong ước lượng vị trí  $(x, y)$  của robot. Sai số này ảnh hưởng trực tiếp đến độ chính xác của quá trình định vị và xây dựng bản đồ (mapping). Để giảm thiểu sự lan truyền của sai số tích lũy, bài báo sẽ trình bày phương pháp trong việc hiệu chỉnh vận tốc tuyến tính và góc quay của robot thông qua hai hướng chính: bộ điều khiển PID nhằm ổn định

đáp ứng vận tốc [10], và hiệu chuẩn (calibration) cảm biến IMU [11] nhằm giảm sai lệch góc đo tích lũy theo thời gian.

### B. Phương pháp hiệu chỉnh vận tốc tuyến tính và góc quay



Hình 3: Sơ đồ bộ điều khiển PID (PID controller diagram)

1) *Phương pháp hiệu chỉnh vận tốc tuyến tính*: Trong mô hình điều khiển chuyển động cho robot điều khiển bằng hai bánh, giảm sai số và ổn định động học của hệ là một bước quan trọng trong những nhiệm vụ kế tiếp. Do đó, mục tiêu ở bước này là tìm được bộ số  $K_p, K_i, K_d$  tối ưu cho hệ thống robot. Ban đầu giữ cố định  $K_i=0, K_d=0$  sau đó tăng dần giá trị  $K_p$  cho đến khi tốc độ thực tế của hệ thống đáp ứng đủ giá trị tốc độ mục tiêu. Khi  $K_p$  đã đưa hệ về vùng đáp ứng đủ nhanh nhưng vẫn tồn tại sai số bền vững cần hiệu chỉnh hệ số tích phân  $K_i$  để loại bỏ sai số này, tăng  $K_i$  từ từ cho đến khi sai số bền vững giảm về mức có thể chấp nhận, đồng thời phải tránh được hiện tượng tích phân quá mức. Cuối cùng thêm thành phần đạo hàm  $K_d$  để giảm vượt quá do quán tính của xe và cải thiện khả năng dập dao động. Toàn bộ quy trình kèm theo các biện pháp thực nghiệm với các bộ số khác nhau để so sánh thời gian đáp ứng của tốc độ bánh cùng với sai số của hệ thống. Sau khi hiệu chỉnh bộ ba tham số trên sẽ cho ra một hệ thống robot ở trong môi trường thực đáp ứng tốc độ mỗi bánh ổn định.

2) *Phương pháp hiệu chỉnh góc quay*: Khi sử dụng con quay (gyroscope) để đo tốc độ góc quay quanh trục Z và tích phân giá trị đó để thu được góc Yaw, bất kỳ offset tịnh tiến (bias) nào trong tín hiệu tốc độ sẽ tích lũy theo thời gian thành một lỗi góc ngày càng lớn (tức là trôi). Phép hiệu chuẩn calibration mà bài báo sử dụng tuy đơn giản nhưng rất hiệu quả, đó là ước lượng bias này trong trạng thái thiết bị đứng yên (tức là tốc độ thực bằng 0) bằng cách lấy trung bình mẫu của một số lượng lớn mẫu  $G_z$  đo cụ thể ở đây là  $N = 1000$  mẫu.

$$G_z^{(\text{bias})} = \frac{1}{N} \sum_{i=1}^N \widetilde{G}_z[i].$$

Sau đó trừ giá trị bias ước lượng khỏi mọi mẫu thu được trong vận hành bình thường từ đó làm giảm phần lớn thành phần trôi của hệ thống bắt nguồn từ offset tịnh tiến.

$$\psi(t) = \int_0^t (\dot{\psi}(\tau) - G_z^{(\text{bias})}) d\tau.$$

### C. Kịch bản đánh giá sai số

Nhằm đánh giá mức độ cải thiện độ chính xác khi áp dụng các phương pháp trên, bài báo tiến hành các thí nghiệm dưới đây:

### 1) Đánh giá sai số vận tốc trong điều khiển:

- Sai số vận tốc đáp ứng: Cho robot chạy thẳng trong khoảng thời gian  $t$  cố định, đo quãng đường  $s$  thực tế và tính  $v = s/t$  so sánh với giá trị vận tốc  $v$  đặt trước nhằm đánh giá độ khả năng đáp ứng và độ ổn định của vận tốc.
- Sai số quãng đường: Đặt mốc 1.2 m, robot chạy ở các vận tốc khác nhau (0.1–1.0 m/s), ghi  $s_{real}$  và so sánh  $s_{theory} = v \cdot t$ . Kết quả thu được cho phép đánh giá mức độ chính xác của phép ước lượng quãng đường theo dải vận tốc, cũng như khả năng phản ứng của bộ điều khiển PID trong quá trình giảm tốc hoặc dừng khẩn cấp.

### 2) Đánh giá sai số góc:

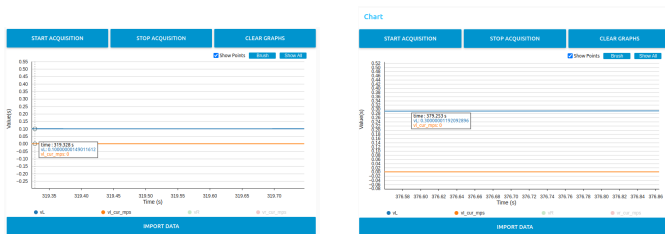
- Sai số góc tích lũy: Robot được giữ một góc cố định trong khi ghi nhận liên tục giá trị góc phản hồi từ cảm biến IMU theo thời gian. Dữ liệu thu được trước và sau khi hiệu chuẩn (Calibration) được so sánh nhằm đánh giá mức độ trôi góc (drift) và khả năng ổn định của cảm biến.
- Sai số góc định vị: Robot được điều khiển xoay theo các góc tùy ý và giá trị góc thực tế đo đạc được, sau đó so sánh với góc phản hồi thu được cảm biến thông qua cơ chế DDS trên máy tính chủ. Kết quả giúp xác định độ chính xác của phép đo góc quay và hiệu quả của quá trình hiệu chuẩn trong việc giảm sai lệch góc định vị.

### 3) Đánh giá sai số định vị tọa độ:

- Robot được điều khiển di chuyển theo các quỹ đạo khác nhau trong môi trường thực. Tại từng điểm mốc, các tọa độ thực tế  $(x, y)$  được ghi nhận và so sánh với tọa độ phản hồi thu được từ hệ thống cảm biến [12]. Qua đó định lượng sai số định vị tổng thể của robot trong môi trường thực.

## IV. KẾT QUẢ NGHIÊN CỨU

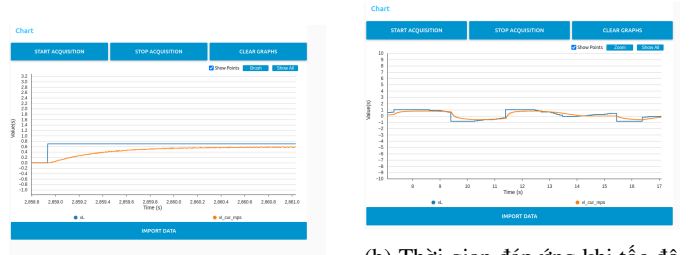
### A. Đánh giá sai số vận tốc trong điều khiển



(a) Vận tốc đáp ứng 0.1 (m/s) (b) Vận tốc đáp ứng 0.3 (m/s)

Hình 4: Đồ thị đáp ứng vận tốc ở vùng vận tốc thấp khi không có PID

1) Sai số vận tốc đáp ứng: Từ Hình 4a và Hình 4b có thể thấy rằng ở vùng vận tốc thấp, do thiếu cơ chế bù sai số của bộ điều khiển PID, vận tốc thực tế của bánh xe (đường màu cam) không đạt được giá trị đặt mong muốn (đường màu xanh).

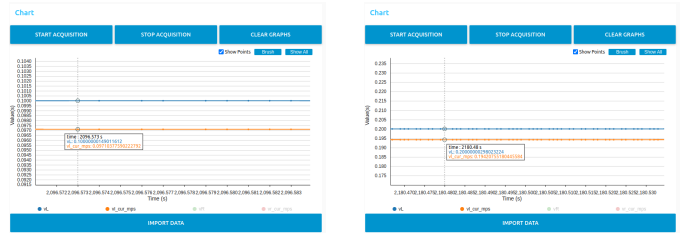


(a) Vận tốc đáp ứng 0.7 (m/s)

(b) Thời gian đáp ứng khi tốc độ thay đổi

Hình 5: Đồ thị đáp ứng vận tốc ở vùng vận tốc cao khi không có PID

Từ Hình 5a có thể nhận thấy rằng, ở vùng vận tốc cao, tốc độ bánh thực tế (đường màu cam) bám theo xu hướng vận tốc đặt (đường màu xanh). Tuy nhiên thay gian đáp ứng vẫn còn chậm khi tốc độ thay đổi như Hình 5b

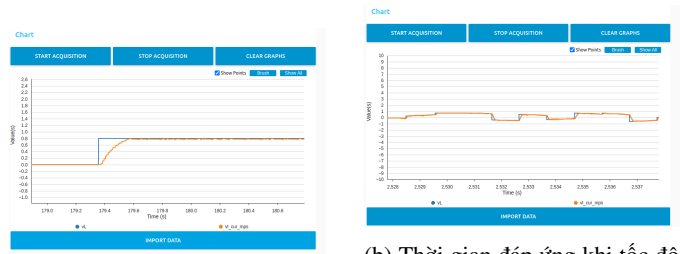


(a) Vận tốc đáp ứng 0.1 (m/s)

(b) Vận tốc đáp ứng 0.2 (m/s)

Hình 6: Đồ thị đáp ứng vận tốc ở vùng vận tốc thấp khi có PID

Từ Hình 6 có thể nhận thấy rằng, ở vùng vận tốc thấp nhờ cơ chế bù sai số của bộ điều PID vận tốc thực tế của bánh xe (đường màu cam) đã bám sát và đạt được giá trị mong muốn (đường màu xanh).



(a) Vận tốc đáp ứng 0.7 (m/s)

(b) Thời gian đáp ứng khi tốc độ thay đổi

Hình 7: Đồ thị đáp ứng vận tốc ở vùng vận tốc cao khi có PID

Từ Hình 7a có thể thấy thời gian đáp ứng của bộ điều khiển PID là 0.22s khi từ vận tốc 0(m/s) lên 0.7(m/s). Bên cạnh đó khi tốc độ mong muốn thay đổi liên tục thì tốc độ thực tế cũng đáp ứng nhanh hơn theo như ở Hình 7b

2) Sai số tốc độ: Trước khi sử dụng bộ điều khiển PID (bảng I) có thể dễ dàng nhận ra sai số của tốc độ đáp ứng so với giá trị mong muốn bị lệch rất lớn với giá trị truyền vào( sai số tỉ đối trung bình cỡ 50%), tốc độ càng tăng thì sai số

Bảng I: Kết quả đo sai số tốc độ thực tế và tốc độ mong muốn trước khi PID

Tốc độ (m/s) Lần	0.5	0.6	0.7	0.8	1.0
1	0.121	0.251	0.359	0.446	0.686
2	0.110	0.242	0.359	0.454	0.682
3	0.100	0.242	0.358	0.440	0.697
4	0.098	0.247	0.363	0.430	0.681
5	0.118	0.234	0.367	0.442	0.686
Trung bình	0.109	0.243	0.361	0.442	0.686
Sai số tuyệt đối	0.391	0.357	0.339	0.358	0.314
Sai số tỉ đối (%)	78.12	59.47	48.40	44.70	31.35

Bảng II: Kết quả đo sai số tốc độ thực tế và tốc độ mong muốn sau khi PID

Tốc độ (m/s) Lần	0.1	0.2	0.3	0.5	0.7
1	0.084	0.176	0.274	0.447	0.620
2	0.084	0.170	0.268	0.447	0.616
3	0.084	0.170	0.259	0.448	0.615
4	0.083	0.174	0.271	0.460	0.617
5	0.082	0.176	0.264	0.456	0.624
Trung bình	0.083	0.173	0.267	0.452	0.618
Sai số tuyệt đối	0.017	0.027	0.033	0.048	0.082
Sai số tỉ đối (%)	16.60	13.40	10.93	9.68	11.66

càng nhỏ đi bởi vì với tốc độ nhỏ thì xe cần một lực ban đầu lớn để thắng được lực ma sát nghỉ nên xe đáp ứng vận tốc ổn định rất chậm. Khi tăng dần tốc độ xe thì sai số đã được giảm bớt nhưng tổng quan mức sai số về tốc độ vẫn rất lớn. Điều này làm ảnh hưởng vô cùng nghiêm trọng tới việc cập nhật vị trí của robot lên bản đồ sau này. Sau khi sử dụng bộ điều khiển PID (bảng II) có thể thấy ngay sai số của tốc độ đáp ứng đã giảm đáng kể( sai số tỷ đối trung bình cỡ 12%), từ đó việc cập nhật vị trí của robot lên bản đồ trở nên chính xác hơn.

3) *Sai số quãng đường:* Kết quả thực nghiệm ở bảng III cho thấy sai số quãng đường của robot sẽ tăng dần nếu tốc độ tăng bởi vì vấn đề quán tính (sai số tỉ đối trung bình khoảng 12%). Đối với bảng IV có thể nhận ra rõ sai số quãng đường đã giảm đáng kể so với bảng III( sai số tỉ đối trung bình khoảng 2.7% ) nhờ bộ điều khiển PID trong quá trình giảm tốc.

Bảng III: Kết quả đo sai số quãng đường thực tế và quãng đường mong muốn trước khi PID

Tốc độ (m/s) Lần	0.5	0.6	0.7	0.9	1.0
1	117.7	124.0	130.0	142.5	152.5
2	118.8	123.0	131.0	141.0	153.0
3	106.5	123.0	132.5	142.5	153.0
4	115.5	123.0	131.0	142.0	150.5
5	112.0	125.0	129.0	141.0	154.0
Trung bình	114.1	123.6	130.7	141.8	152.6
Sai số tuyệt đối	5.9	3.6	10.7	21.8	32.6
Sai số tỉ đối (%)	4.92	3.00	8.92	18.17	27.17

## B. Đánh giá sai số góc

1) *Sai số góc tích lũy:* Quan sát bảng V cho thấy sau khi áp dụng quá trình hiệu chuẩn (calibration), hiện tượng trôi (drift) của góc yaw gần như được triệt tiêu hoàn toàn, giá trị góc dao

Bảng IV: Kết quả đo sai số quãng đường thực tế và quãng đường mong muốn sau khi PID

Tốc độ (m/s) Lần	0.5	0.6	0.7	0.9	1.0
1	119.5	120.0	122.0	126.0	127.0
2	119.5	119.5	121.0	126.0	127.0
3	119.5	121.0	121.5	125.5	127.5
4	119.0	121.5	123.0	126.0	127.5
5	119.0	121.5	122.0	124.5	128.0
Trung bình	119.3	120.7	121.9	125.6	127.4
Sai số tuyệt đối	0.7	0.7	1.9	5.6	7.4
Sai số tỉ đối (%)	0.58	0.58	1.58	4.67	6.17

Bảng V: So sánh kết quả góc quay theo thời gian trước và sau khi hiệu chuẩn Calibration

Thời gian đo (s)	Trước Calibration (°)	Sau Calibration(°)
0	2.63	0.03
30	6.85	0.05
60	16.95	-0.04
90	26.75	-0.07
120	35.51	-0.05
150	44.29	0.06
180	53.14	0.07
210	65.36	0.13
240	76.17	-0.04
270	84.69	-0.12

động nhỏ quanh 0°, phản ánh chính xác trạng thái thực tế của robot khi đứng yên.

Bảng VI: Sai số góc quay thực tế và phản hồi khi hiệu chuẩn Calibration

Lần đo	Góc thực tế (°)	Góc phản hồi (°)	Sai số tuyệt đối (°)	Sai số tỉ đối (%)
1	65.44	68.88	3.44	4.99
2	109.41	121.29	11.88	9.79
3	122.25	141.00	18.75	13.30
4	124.87	122.51	2.36	1.93
5	61.79	65.07	3.28	5.04
6	142.15	158.13	15.98	10.11
7	46.48	46.61	0.13	0.28
8	259.73	243.19	16.54	6.80
9	85.86	91.68	5.82	6.35
10	329.64	320.52	9.12	2.85
Trung bình	-	-	8.73	6.14

2) *Sai số góc định vị:* Kết quả trong Bảng VI cho thấy sai số góc quay sau khi hiệu chuẩn Calibration tương đối ổn định, với sai số tuyệt đối trung bình khoảng 8.73° và sai số tỉ đối trung bình 6.14 %.

## C. Sai số định vị tọa độ

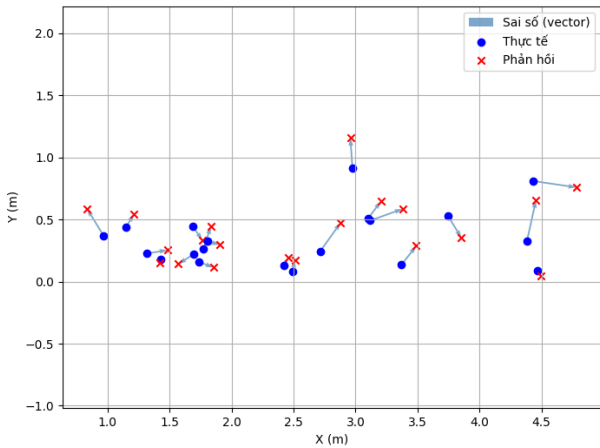
Bảng VII thể hiện kết quả so sánh giữa tọa độ thực tế  $(x_{real}, y_{real})$  và tọa độ phản hồi  $(x_{map}, y_{map})$  thu được từ hệ thống cảm biến trong các lần thí nghiệm ở các vận tốc khác nhau và góc quay khác nhau. Sai số định vị được đánh giá bằng hai chỉ số là sai số tuyệt đối trung bình (MAE) và căn sai số bình phương trung bình (RMSE) [13]:

$$MAE = \frac{1}{n} \sum_{i=1}^n \left| \sqrt{(x_{real,i} - x_{map,i})^2 + (y_{real,i} - y_{map,i})^2} \right| \quad (5)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n [(x_{real,i} - x_{map,i})^2 + (y_{real,i} - y_{map,i})^2]} \quad (6)$$

Bảng VII: Kết quả đo sai số định vị vị trí giữa thực tế và phản hồi từ hệ thống cảm biến

Tốc độ (m/s)	Hoànch độ đo	Hoànch độ MAP	Tung độ đo	Tung độ MAP
0.1	0.225	0.254	1.315	1.484
	0.265	0.446	1.775	1.834
	0.220	0.142	1.695	1.571
	0.177	0.154	1.430	1.421
	0.370	0.588	0.968	0.835
0.2	0.444	0.333	1.689	1.765
	0.324	0.296	1.810	1.906
	0.160	0.113	1.736	1.854
	0.440	0.544	1.149	1.215
	0.810	0.759	4.435	4.780
0.3	0.242	0.475	2.714	2.881
	0.510	0.647	3.103	3.205
	0.491	0.585	3.117	3.380
	0.914	1.157	2.975	2.959
	0.088	0.045	4.468	4.499
0.5	0.329	0.656	4.383	4.456
	0.531	0.351	3.745	3.854
	0.127	0.193	2.420	2.459
	0.135	0.288	3.370	3.491
	0.080	0.175	2.494	2.511



Hình 8: So sánh tọa độ thực tế và phản hồi từ hệ thống cảm biến

Kết quả tính toán cho thấy:

$$MAE = 0.178 \text{ m}, \quad RMSE = 0.175 \text{ m}$$

Giá trị MAE và RMSE gần tương đương cho thấy sai số định vị có phân bố ổn định, không xuất hiện các sai lệch đột biến.

## V. KẾT LUẬN

Sau khi áp dụng bộ điều khiển PID, vận tốc của robot bám theo giá trị đặt tốt hơn ngay cả ở vùng tốc độ thấp (0.1 m/s đến 0.3 m/s), đồng thời đáp ứng nhanh hơn khi đầu vào thay đổi (Hình 7b). Sai số vận tốc giảm đáng kể, từ **78.12%** xuống **9.68%** ở 0.5 (m/s) và từ **59.47%** xuống **11.66%** ở 0.7 (m/s), trong khi sai số quãng đường giảm từ **27.17%** xuống **6.17%**.

Sau hiệu chuẩn Calibration, hiện tượng trôi góc (drift) được triệt tiêu hoàn toàn (Bảng V), với sai số góc chỉ **6.14°**.

Sai lệch định vị trung bình đạt khoảng 17–18 cm so với vị trí thực, cho thấy sự hiệu quả của quá trình hiệu chỉnh cảm biến và bộ điều khiển PID. Kết quả này là cơ sở quan trọng cho việc triển khai các ứng dụng định vị trong môi trường trong nhà, nơi tín hiệu GPS bị hạn chế.

## TÀI LIỆU THAM KHẢO

- [1] K. He, H. Ding, N. Xu, and K. Guo, “Wheel odometry with deep learning-based error prediction model for vehicle localization,” *Applied Sciences*, vol. 13, no. 9, p. 5588, 2023.
- [2] V. Suvorkin, M. Garcia-Fernandez, G. Gonzalez-Casado, M. Li, and A. Rovira-Garcia, “Assessment of noise of mems imu sensors of different grades for gnss/imu navigation,” *Sensors*, vol. 24, no. 6, p. 1953, 2024.
- [3] A. Ghadimzadeh Alamdari, F. Azizi Zade, and A. Ebrahimehkhani, “A review of simultaneous localization and mapping for the robotic-based nondestructive evaluation of infrastructures,” *Sensors*, vol. 25, no. 3, p. 712, 2025.
- [4] A. Bonci, F. Gaudeni, M. C. Giannini, and S. Longhi, “Robot operating system 2 (ros2)-based frameworks for increasing robot autonomy: A survey,” *Applied Sciences*, vol. 13, no. 23, p. 12796, 2023.
- [5] M. Fazekas, P. Gáspár, and B. Németh, “Calibration and improvement of an odometry model with dynamic wheel and lateral dynamics integration,” *Sensors*, vol. 21, no. 2, p. 337, 2021.
- [6] S. Parra, R. Zunino *et al.*, “Specifying qos requirements and capabilities for ros 2 dds-based systems,” in *ROSE 2021: IEEE Int. Workshop on Robotics Software Engineering*, 2021, pp. 1–8.
- [7] R. Petrella, M. Tursini, L. Peretti, and M. Zigliotto, “Speed measurement algorithms for low-resolution incremental encoder equipped drives: A comparative analysis,” in *Proceedings of the International Aegean Conference on Electrical Machines and Power Electronics (ACEMP)*, 2007.
- [8] M. Wrona, “Roll and pitch angles from accelerometer sensors,” <https://mwrona.com/posts/accel-roll-pitch/>, 2020, accessed 2025-10-29.
- [9] S. K. Malu and J. Majumdar, “Kinematics, localization and control of differential drive mobile robot,” *Global Journals of Research in Engineering*, vol. 14, no. H1, pp. 1–7, 2014.
- [10] K. J. Åström and T. Häggglund, *Automatic Tuning of PID Controllers*. Research Triangle Park, NC: Instrument Society of America (ISA), 1988.
- [11] Z. Wang, X. Cheng, and J. Fu, “Optimized multi-position calibration method with nonlinear scale factor for inertial measurement units,” *Sensors*, vol. 19, no. 16, p. 3568, 2019.
- [12] T. Foote, B. Gerkey, K. Conley, W. Meeussen *et al.*, “Rep-105: Coordinate frames for mobile platforms,” <https://www.ros.org/reps/rep-0105.html>, 2010, accessed 2025-10-29.
- [13] T. Chai and R. R. Draxler, “Root mean square error (rmse) or mean absolute error (mae)—arguments against avoiding rmse in the literature,” *Geoscientific Model Development*, vol. 7, no. 3, pp. 1247–1250, 2014.