

## LỜI CẢM ƠN

Để có thể hoàn thành đề tài luận văn tốt nghiệp một cách hoàn chỉnh, bên cạnh sự nỗ lực cố gắng của nhóm còn có sự hướng dẫn tận tình của quý Thầy Cô, cũng như sự động viên ủng hộ của gia đình và bạn bè trong suốt thời gian học tập nghiên cứu và thực hiện luận văn.

Xin chân thành bày tỏ lòng biết ơn sâu sắc đến thầy Nguyễn Chí Nghĩa và thầy Nguyễn Ngọc Khai người đã hết lòng giúp đỡ và tạo mọi điều kiện tốt nhất cho chúng em hoàn thành luận văn này. Xin gửi lời tri ân nhất của chúng em đối với mọi sự giúp đỡ của các Thầy.

Xin chân thành bày tỏ lòng biết ơn đến toàn thể quý Thầy Cô trong bộ môn khoa Điện – Điện Tử Trường Đại Học Bách Khoa TP.HCM đã tận tình truyền đạt những kiến thức quý báu cũng như tạo mọi điều kiện thuận lợi nhất cho em trong suốt quá trình học tập nghiên cứu và cho đến khi em thực hiện đề tài luận văn này.

Xin chân thành bày tỏ lòng biết ơn đến gia đình, những người đã không ngừng động viên, hỗ trợ và tạo mọi điều kiện tốt nhất cho chúng em trong suốt quá trình học tập và thực hiện luận văn.

Cuối cùng, chúng em cũng xin gửi lời cảm ơn đến các bạn cùng ngành đã luôn cùng em trong suốt thời gian học tập tại trường, đã luôn hỗ trợ, động viên nhau, giúp chúng em hoàn thành luận văn một cách hoàn chỉnh.

Tp. Hồ Chí Minh, tháng 06 năm 2018

VÕ TRẦN THIÊN BẢO

BÙI ANH HUY

## MỤC LỤC

TÓM TẮT LUẬN VĂN .....	1
CHƯƠNG 1. GIỚI THIỆU .....	2
1.1. Xu hướng phát triển của robot hiện đại .....	2
1.2. Robot di động .....	3
1.3. Xe tự hành trên mặt đất AGV .....	3
1.4. Nhược điểm và cách khắc phục của xe tự hành trên mặt đất .....	4
CHƯƠNG 2. MÔ HÌNH ROBOT OMNI.....	6
2.1. Cấu trúc.....	6
2.1.1. Bánh xe omni.....	6
2.1.2. Thân xe .....	7
2.2. Cách di chuyển của robot omni .....	8
CHƯƠNG 3. CÁC PHƯƠNG PHÁP ĐỊNH VỊ CHO ROBOT .....	10
3.1. Giới thiệu .....	10
3.2. Các phương pháp định vị.....	10
3.2.1. Phương pháp Dead-Reckoning.....	10
3.2.2. Hệ thống dẫn đường cột mốc chủ động.....	10
3.2.3. Định vị sử dụng bản đồ .....	11
3.2.4. Phương pháp định vị tuyệt đối (Absolute Positioning Method).....	12
3.3. Ưu điểm, nhược điểm của các phương pháp .....	13
CHƯƠNG 4. CẤU TRÚC HỆ THỐNG ROBOT OMNI.....	14
4.1. Kit STM32F407 Discovery .....	14
4.2. Module cảm biến la bàn số HMC5883L .....	15

4.3. Module Bluetooth HC-06 .....	16
4.4. Module điều khiển động cơ L298N.....	16
CHƯƠNG 5. ĐỘNG CƠ VÀ ENCODER.....	18
5.1. Động cơ DC .....	18
5.2. Động cơ DC servo GA-25 .....	19
5.3. Encoder .....	20
5.4. Giải thuật đọc tín hiệu từ encoder để tính tốc độ động cơ.....	23
5.4.1. Độ phân giải .....	23
5.4.2. Độ chính xác.....	24
5.4.3. Các phương pháp đọc tín hiệu encoder để tính tốc độ và vị trí.....	25
5.4.4. Xử lý nhiễu bằng phần cứng .....	25
5.4.5. Xử lý nhiễu bằng thuật toán .....	27
5.4.6. Giải thuật .....	29
CHƯƠNG 6. ĐIỀU KHIỂN VÀ ĐỊNH VỊ.....	31
6.1. Bộ điều khiển Dynamic Surface Control (DSC) .....	31
6.1.1. Giới thiệu chung .....	31
6.1.2. Thuật toán DSC .....	32
6.1.3. Mô hình hóa robot omni .....	35
6.1.4. Mô hình động lực học.....	36
6.1.5. Thiết kế bộ điều khiển .....	37
6.1.6. Áp dụng cho đối tượng Omnidroid với hai mặt trượt S1, S2 .....	39
6.2. Phương pháp định vị Dead-Reckoning .....	41
6.2.1. Giới thiệu.....	41

6.2.2.	Tầm quan trọng của bài toán dead reckoning.....	42
6.2.3.	Điều kiện để định vị tốt .....	42
6.2.4.	Áp dụng phương pháp định vị vào Omni robot .....	43
6.3.	Sơ đồ khái thuật toán điều khiển .....	43
<b>CHƯƠNG 7.</b>	<b>GIAO DIỆN ĐIỀU KHIỂN VÀ GIÁM SÁT</b> .....	<b>44</b>
7.1.	Thư viện OpenCV .....	44
7.1.1.	Tổng quan.....	44
7.1.2.	Một số module của OpenCV .....	44
7.2.	Giao diện điều khiển và giám sát .....	46
<b>CHƯƠNG 8.</b>	<b>THỰC NGHIỆM VÀ ĐÁNH GIÁ</b> .....	<b>47</b>
8.1.	Thực hiện mô phỏng đáp ứng sau khi qua bộ điều khiển.....	47
8.2.	Thực nghiệm.....	52
8.3.	Khó khăn và giải pháp .....	60
8.4.	Đánh giá chung .....	68
<b>CHƯƠNG 9.</b>	<b>ỨNG DỤNG VÀ HƯỚNG PHÁT TRIỂN</b> .....	<b>69</b>
9.1.	Ứng dụng .....	69
9.2.	Hướng phát triển.....	69
<b>TÀI LIỆU THAM KHẢO</b> .....		<b>70</b>

## DANH MỤC HÌNH VẼ

Hình 1.1: Xe tự hành trên sao hỏa. Ảnh: NASA.....	4
Hình 2.1: Bánh xe omni (trái) và bánh xe omni mecanum (phải) .....	6
Hình 2.2: Robot omni.....	7
Hình 2.3: Chuyển động thẳng, lùi của robot omni.....	8
Hình 2.4: Chuyển động sang trái, phải của robot omni .....	8
Hình 2.5: Chuyển động chéo góc của robot omni.....	9
Hình 2.6: Chuyển động xoay của robot omni .....	9
Hình 3.1: Mô hình định vị cột mốc .....	11
Hình 3.2: Mô hình xử lý thuật toán định vị bằng hình ảnh.....	12
Hình 4.1: Kit STM32F407VG Discovery.....	14
Hình 4.2: Module la bàn số HMC5883L .....	15
Hình 4.3: Sơ đồ chân module HC-06.....	16
Hình 4.4: Module cầu H L298N .....	17
Hình 5.1: Điều chỉnh độ rộng xung PWM .....	18
Hình 5.2: Mối quan hệ giữa dòng và áp trên động cơ DC .....	19
Hình 5.3: Động cơ DC servo GA-25 .....	20
Hình 5.4: Optical Encoder.....	21
Hình 5.5: Hai kênh A và B lệch pha trong encoder .....	22
Hình 5.6: Độ chính xác khi đọc tín hiệu encoder.....	23
Hình 5.7: Nhiều khi đọc tín hiệu encoder .....	24
Hình 5.8: Sơ đồ mạch lọc thông thấp bậc 1 cho encoder.....	25
Hình 5.9: Biểu đồ Bode đáp ứng ngõ ra encoder.....	26
Hình 5.10: Tín hiệu encoder sau khi lọc nhiễu .....	27
Hình 5.11: Bộ lọc Kalman đơn giản .....	28
Hình 5.12: Bốn trạng thái của kênh A, B encoder .....	29
Hình 5.13: Chu trình đọc tín hiệu từ encoder.....	30

Hình 6.1: Mô hình robot omni 4 bánh .....	35
Hình 6.2: Sơ đồ khói thuật toán điều khiển .....	43
Hình 7.1: Giao diện điều khiển .....	46
Hình 8.1. Đáp ứng theo trực tọa độ x với $k_{11} = k_{12} = k_{13} = 3$ .....	47
Hình 8.2. Đáp ứng theo trực tọa độ y với $k_{11} = k_{12} = k_{13} = 3$ .....	48
Hình 8.3. Đáp ứng góc với $k_{11} = k_{12} = k_{13} = 3$ .....	48
Hình 8.4. Đáp ứng theo trực tọa độ x với $k_{11} = k_{12} = k_{13} = 5$ .....	49
Hình 8.5. Đáp ứng theo trực tọa độ y với $k_{11} = k_{12} = k_{13} = 5$ .....	49
Hình 8.6. Đáp ứng góc với $k_{11} = k_{12} = k_{13} = 5$ .....	50
Hình 8.7. Đáp ứng theo trực tọa độ y với $k_{11} = k_{12} = k_{13} = 8$ .....	50
Hình 8.8. Đáp ứng theo trực tọa độ y với $k_{11} = k_{12} = k_{13} = 8$ .....	51
Hình 8.9. Đáp ứng góc với $k_{11} = k_{12} = k_{13} = 8$ .....	51
Hình 8.10. Đáp ứng thực tế theo trực tọa độ x với $k_{11} = k_{12} = k_{13} = 5$ .....	53
Hình 8.11. Đáp ứng thực tế theo trực tọa độ y với $k_{11} = k_{12} = k_{13} = 5$ .....	53
Hình 8.12. Đáp ứng thực tế theo góc với $k_{11} = k_{12} = k_{13} = 5$ .....	54
Hình 8.14. Đáp ứng thực tế theo trực tọa độ x với $k_{11} = k_{12} = k_{13} = 5$ .....	55
Hình 8.15. Đáp ứng thực tế theo góc với $k_{11} = k_{12} = k_{13} = 5$ .....	56
Hình 8.16. Đáp ứng thực tế theo trực tọa độ x với $k_{11} = k_{12} = k_{13} = 8$ .....	56
Hình 8.17. Đáp ứng thực tế theo trực tọa độ y với $k_{11} = k_{12} = k_{13} = 8$ .....	57
Hình 8.18. Đáp ứng thực tế theo góc với $k_{11} = k_{12} = k_{13} = 8$ .....	57
Hình 8.19. Đáp ứng thực tế theo trực x với $k_{11} = k_{12} = k_{13} = 12$ .....	58
Hình 8.20. Đáp ứng thực tế theo trực y với $k_{11} = k_{12} = k_{13} = 12$ .....	58
Hình 8.21. Đáp ứng thực tế theo góc với $k_{11} = k_{12} = k_{13} = 12$ .....	59
Hình 8.22. Đáp ứng thực tế theo trực x sau khi giải quyết hạn chế .....	62
Hình 8.23. Đáp ứng thực tế theo trực y sau khi giải quyết hạn chế .....	62
Hình 8.24. Đáp ứng thực tế theo góc sau khi giải quyết hạn chế .....	63
Hình 8.25. Đáp ứng thực tế theo trực x sau khi giải quyết hạn chế .....	63

Hình 8.26. Đáp ứng thực tế theo trục y sau khi giải quyết hạn chế .....	64
Hình 8.27. Đáp ứng thực tế theo góc sau khi giải quyết hạn chế .....	64
Hình 8.28. Đáp ứng thực tế theo trục x sau khi giải quyết hạn chế .....	65
Hình 8.29. Đáp ứng thực tế theo trục x sau khi giải quyết hạn chế .....	66
Hình 8.30. Đáp ứng thực tế theo trục x sau khi giải quyết hạn chế .....	66
Hình 8.30. Vẽ lại quỹ đạo di chuyển.....	67

## **DANH MỤC BẢNG**

Bảng 8.1: Đánh giá chất lượng đáp ứng mô phỏng .....	52
Bảng 8.2: Đánh giá chất lượng đáp ứng thực tế với điểm đặt $\{1, 1, \pi/4\}$ .....	54
Bảng 8.3: Đánh giá chất lượng đáp ứng thực tế với điểm đặt $\{1, 0, 0\}$ .....	59
Bảng 8.4: Đánh giá chất lượng đáp ứng sau giải pháp với điểm đặt $\{1,1,\pi/4\}$ .....	67

## **DANH MỤC TỪ VIẾT TẮT**

PWM: Pulse Width Modulation

DC: Direct Current

DSC: Dynamic Surface Control

OpenCV: OpenSource Computer Vision

SVM: Support Vector Machine

## TÓM TẮT LUẬN VĂN

*Theo dự đoán trong tương lai, robot sẽ là tâm điểm của một cuộc cách mạng lớn sau Internet. Con người sẽ có nhu cầu sở hữu một robot cá nhân như nhu cầu một máy tính PC như bây giờ. Với xu hướng này, cùng các ứng dụng truyền thống khác của robot trong công nghiệp, y tế, giáo dục đào tạo, giải trí và đặc biệt là trong ngành vũ trụ hàng không và thám hiểm thì thị trường sẽ vô cùng to lớn. Để tài luận văn hướng tới việc ứng dụng công nghệ định vị Dead-Reckoning cho robot tự hành, tạo tiền đề cho một robot dịch vụ hoàn chỉnh, có tính tự hành, tự giải quyết vấn đề về định vị và điều hướng khi có sự cố, có khả năng phục vụ cho đời sống con người.*

*Trong khuôn khổ của luận văn, nhóm sẽ tập trung xây dựng một mô hình mobile robot có khả năng định vị theo quãng đường được định sẵn. Một điểm được nhấn mạnh là khả năng định vị dựa trên tốc độ của robot, số vòng quay của bánh xe và góc quay của cảm biến la bàn để xác định chính xác các thông số cơ bản như: quãng đường đi được, vị trí hiện tại. Từ đó giúp nâng cao khả năng tự hành, đáp ứng được sự chính xác và linh hoạt trong cơ chế vận hành cho giải thuật điều khiển truyền thống cho robot.*

**NHÓM SINH VIÊN THỰC HIỆN**

**BÙI ANH HUY  
VÕ TRẦN THIÊN BẢO**

## CHƯƠNG 1. GIỚI THIỆU

### 1.1. Xu hướng phát triển của robot hiện đại

Theo dự báo trong vòng 20 năm nữa mỗi người sẽ có nhu cầu sử dụng một robot cá nhân như cần một máy tính PC hiện nay và Robot sẽ là tâm điểm của một cuộc cách mạng công nghệ lớn sau Internet. Với xu thế này, cùng với các ứng dụng truyền thống khác của robot trong công nghiệp, y tế, giáo dục đào tạo, giải trí và đặc biệt trong an ninh quốc phòng thì thị trường robot và các dịch vụ ăn theo robot sẽ vô cùng lớn.

Robot tổng hợp trong nó cả khoa học và công nghệ. Để thiết kế và chế tạo được robot, ta cần có các tri thức của toán học, cơ học, vật lý, điện tử, lý thuyết điều khiển, khoa học tính toán và nhiều tri thức khác. Để có thể ứng dụng được robot, ta cần biết rõ về đối tượng ứng dụng. Robot là sản phẩm tích hợp cả khoa học và công nghệ với độ phức tạp cao.

Robot đã có những tiến bộ đáng kể trong hơn nửa thế kỷ qua. Robot đầu tiên được ứng dụng trong công nghiệp vào những năm 60 để thay thế con người làm các công việc nặng nhọc, nguy hiểm trong môi trường độc hại. Do nhu cầu cần sử dụng ngày càng nhiều trong các quá trình sản xuất phức tạp nên robot công nghiệp cần có những khả năng thích ứng linh hoạt và thông minh hơn. Ngày nay, ngoài ứng dụng sơ khai ban đầu của robot trong chế tạo máy thì các ứng dụng khác như trong y tế, chăm sóc sức khỏe, nông nghiệp, đóng tàu, xây dựng, an ninh quốc phòng và gia đình đang có nhu cầu gia tăng đang là động lực cho các robot địa hình và robot dịch vụ phát triển.

Có thể kể đến một số loại robot được quan tâm nhiều thời gian qua là: Tay máy robot (Robot Manipulators), Robot di động (Mobile Robots), Robot phỏng sinh học (Bio Inspired Robots) và Robot cá nhân (Personal Robots). Robot di động được nghiên cứu nhiều như Xe tự hành trên mặt đất AGV (Autonomous Guided Vehicles), Robot tự hành dưới nước AUV (Autonomous Underwater Vehicles), Máy bay không người lái UAV (Unmanned Arial Vehicles). Với Robot phỏng sinh học, các nghiên cứu thời gian qua tập trung vào 2 loại chính là Robot đi (Walking robots) và Robot dáng người (Humanoid Robots). Bên cạnh đó, các loại robot phỏng sinh học dưới nước như robot cá, các cầu

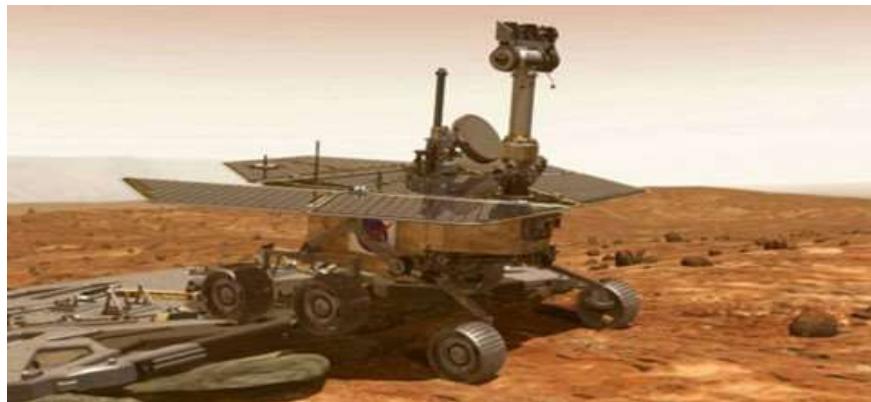
trúc chuyển động phỏng theo sinh vật biển cũng được nhiều nhóm nghiên cứu phát triển. Hiện nay các ứng dụng của robot đang có xu thế chuyển sang các ứng dụng thường nhật như Robot gia đình (home robots) và Robot cá nhân (Personal robots). Mặc dù về cấu trúc của các loại robot có khác nhau nhưng các nghiên cứu hiện nay đều hướng về các ứng dụng dịch vụ và hoạt động của robot trong các môi trường tự nhiên. Với sự phát triển của xã hội và quá trình hiện đại hóa ở các nước phát triển thì nhiều dịch vụ mới được hình thành làm thay đổi quan điểm về robot từ robot phục vụ công nghiệp sang robot phục vụ cho các nhu cầu xã hội và nhu cầu cá nhân của con người. .

### **1.2. Robot di động**

Robot di động là một hệ robot có khả năng thực hiện các nhiệm vụ ở nhiều vị trí khác nhau với khả năng dịch chuyển bằng bánh xe, xích hay bằng chân phụ thuộc vào địa hình. Đối với robot di chuyển dưới nước hay trên không, ta cần có động cơ cánh quạt, chân vịt hay động cơ phản lực để tạo chuyển động cho robot. Khả năng di động làm robot có nhiều ứng dụng và đòi hỏi phải giải quyết nhiều vấn đề mới. Một trong những vấn đề chung cần nghiên cứu ở các loại robot di động là khả năng xác định phương hướng (navigation) của robot. Ta hãy điểm qua sự phát triển của ba loại robot di động chính bao gồm Xe tự hành trên mặt đất AGV, Robot tự hành dưới nước AUV, Máy bay không người lái UAV.

### **1.3. Xe tự hành trên mặt đất AGV**

Xe tự hành AGV bắt đầu được chế tạo để vận chuyển các phôi gia công vào những năm 70. Vấn đề định hướng của xe tự hành là một trong những vấn đề cốt lõi cần phải giải quyết. Bài toán định hướng hiện nay được phân làm 2 loại: định hướng trong nhà (indoor) và định hướng ngoài địa hình (outdoor).



*Hình 1.1: Xe tự hành trên sao hỏa. Ảnh: NASA*

Quá trình định hướng của robot đi động gồm 4 bước: thu nhận cảnh quan môi trường, xác định vị trí, thiết kế quỹ đạo và tạo chuyển động. Với môi trường có cấu trúc, quá trình nhận biết cho phép tạo ra bản đồ hay mô hình không gian phục vụ cho bài toán định vị và thiết kế quỹ đạo robot. Đối với môi trường phi cấu trúc hay thay đổi, robot cần có khả năng tự học quan sát môi trường để xác định được hướng đi của mình. Do đó lĩnh vực xác định hướng đi cho robot di động là một lĩnh vực mà các phương pháp trí khôn nhân tạo như quá trình nhận biết môi trường, suy diễn và tìm hướng đi tối ưu có thể được áp dụng. Vấn đề định vị và tạo bản đồ là những vấn đề nghiên cứu trọng tâm ở robot di động thời gian qua.

Quá trình định vị là quá trình robot xác định được hiện nó ở đâu trong không gian hoạt động. Để đạt được mục tiêu này, cần sử dụng nhiều cảm biến thu nhận các dữ liệu liên quan đến trạng thái của robot và môi trường xung quanh.

#### **1.4. Nhược điểm và cách khắc phục của xe tự hành trên mặt đất**

Các dữ liệu về thông tin định vị thường bị nhiễu và có sai số tích lũy nên cần có các phương pháp lọc động và sử dụng các phương pháp phối hợp cảm biến để có được số liệu đo chính xác hơn. Phương pháp định vị có thể là cục bộ hay toàn cục. Giải pháp đơn giản nhất là định vị cục bộ khi robot thường xuyên cập nhật vị trí của nó so với điểm xuất phát. Ngược lại các phương pháp định vị toàn cục không đòi hỏi biết vị trí của điểm xuất phát.

Để khắc phục độ bát định của các thông tin đo được từ các cảm biến ta cần sử dụng các phương pháp xác suất. Các phương pháp định vị được sử dụng thường dựa trên nguyên lý lọc Bayes kết hợp với một thuật toán đệ quy để ước lượng được vị trí và hướng từ phương trình mô tả chuyển động của robot. Thời gian tính toán của lọc Bayes lâu nên nhiều nghiên cứu gần đây tập trung vào tìm các phương pháp đơn giản hóa để giảm khối lượng tính toán. Quá trình đơn giản hóa này dẫn đến nhiều thuật toán định vị khác nhau phân làm 2 loại tùy thuộc vào cách mô tả độ tin cậy của dữ liệu. Nếu dữ liệu được mô tả bằng các hàm phân bố Gauss, ta có thể sử dụng phương pháp lọc Kalman. Nếu dữ liệu được mô tả bằng nhiều hàm phân bố xác suất khác nhau, ta có thể sử dụng các thuật toán định vị dựa trên quá trình Markov. Phương pháp định vị dựa trên phân bố Gauss và lọc Kalman chỉ ứng dụng hiệu quả cho bài toán định vị cục bộ. Các phương pháp định vị Markov có thể là phương pháp tô pô, phương pháp lưới và phương pháp sử dụng các mẫu rời rạc của các giá trị trạng thái. Khi các dữ liệu robot được mô tả bằng các mẫu rời rạc ngẫu nhiên ta có thể sử dụng các phương pháp lọc phần tử (particle filter) để xác định được vị trí của robot tốt hơn.

Do việc định vị và lập bản đồ cho robot có quan hệ mật thiết với nhau nên từ những năm 90 các nghiên cứu đã tập trung giải quyết hai vấn đề này đồng thời với tên chung là SLAM (Simultaneous Localization And Mapping). Các nghiên cứu gần đây liên quan nhiều đến lập bản đồ cho môi trường động. Lọc Kalman có thể sử dụng được cho bài toán này nhưng không cho độ chính xác cao. Vấn đề tìm được một thuật toán lập bản đồ cho môi trường động còn là một thách thức lớn. Nhiều vấn đề còn bỏ ngỏ như phân biệt các đối tượng tĩnh, đối tượng chuyển động và mô tả chúng trên bản đồ. Ngoài các ứng dụng trong công nghiệp và quân sự, robot di động cũng đã đi vào cuộc sống đời thường như robot hút bụi, lau nhà, cọ bể bơi, lau kính, robot bào tàng, robot dịch vụ văn phòng, bệnh viện và ở các nơi công cộng. Các robot này yêu cầu mức độ nhận thức nhất định, có khả năng tự định hướng, di chuyển trong môi trường biến động và có giao tiếp người-máy thân thiện. Ứng dụng robot di động trong giám sát, gác, cảnh báo về an ninh cũng là một hướng được nhiều công ty bảo vệ an ninh sử dụng.

## CHƯƠNG 2. MÔ HÌNH ROBOT OMNI

### 2.1. Cấu trúc

#### 2.1.1. Bánh xe omni

Bánh xe Omni là loại bánh có các bánh vệ tinh nhỏ đặt xung quanh chu vi bánh, 2 vòng bánh con sole với nhau và vuông góc với trực bánh. Nhờ đó bánh xe có thể di chuyển dọc (do động cơ truyền động) và trượt theo chiều ngang (phụ thuộc vào chuyển động của các bánh còn lại). Kết hợp hai chuyển động này (cộng vector) thì bánh Omni có thể di chuyển theo mọi hướng trong khi vẫn giữ được hướng của đầu robot không đổi trong quá trình di chuyển (do Robot sử dụng bánh omni có thể di chuyển theo bất kỳ hướng nào, ở bất kỳ góc nào mà không cần quay trước khi di chuyển). Đó là sự khác biệt cơ bản nhất so với các loại bánh xe truyền thống hay bánh xích dùng trong xe Tăng, khi đổi hướng thì phải quay đầu rồi đi theo hướng đó.

Vì có khả năng như vậy nên robot omni có khả năng tiết kiệm năng lượng so với robot sử dụng bánh xe thông thường. Để thực hiện khả năng di chuyển đó, robot omni phải có 3 bậc tự do trong mặt phẳng di chuyển gồm: di chuyển độc lập theo các trục x, y và xoay theo trục z. Do đó cần tối thiểu 3 bánh omni với các động cơ điều khiển độc lập, vì mỗi động cơ cho khả năng điều khiển một bậc tự do.



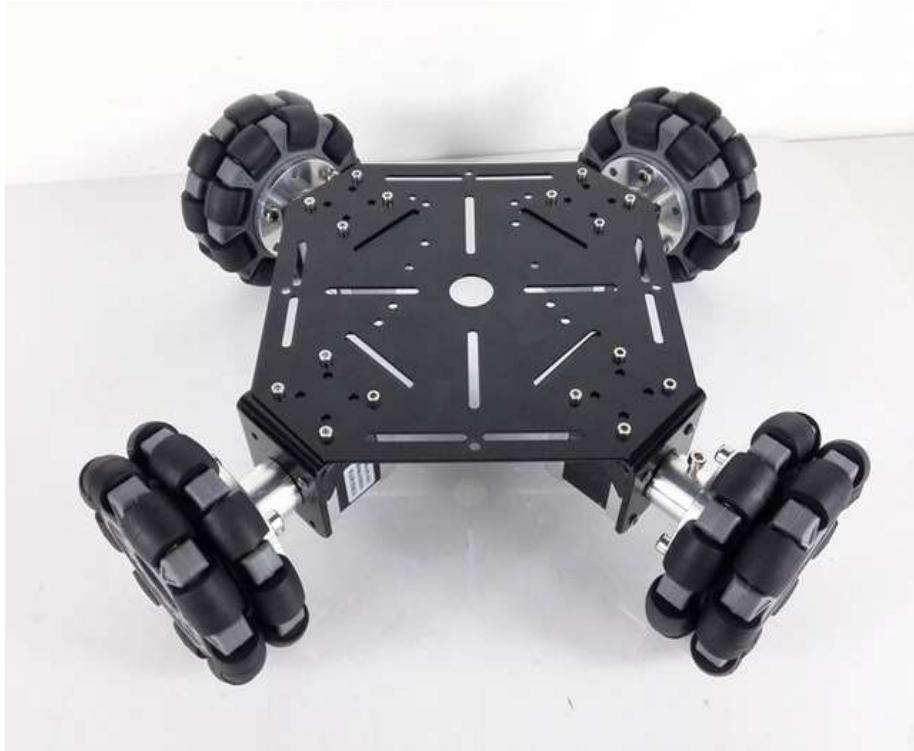
Hình 2.1: Bánh xe omni (trái) và bánh xe omni mecanum (phải)

Bánh xe Omni có thể sử dụng 3 hoặc 4 bánh tùy ý. Ngoài ra bánh Omni cho tốc độ di chuyển đều các hướng cũng như cho kết quả tốt khi di chuyển theo hướng chéo  $45^\circ$ .

Bánh xe Mecanum về cơ bản cũng giống như bánh xe omni nhưng các bánh xe vệ tinh được đặt chêch đi 45 độ. Có 2 loại bánh (trái, phải) khác nhau về hướng của trực bánh xe vệ tinh. Robot cần cả 2 loại trái, phải để có thể di chuyển linh hoạt được.

### 2.1.2. Thân xe

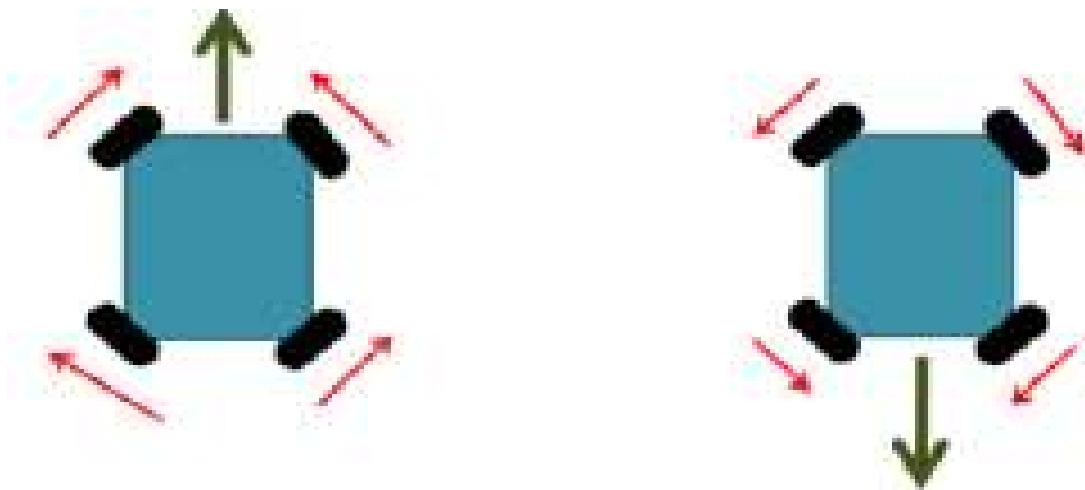
Thân xe ở đây là một giá đỡ kim loại được thiết kế để tối ưu không gian cho các mạch điện tử và động cơ được gắn lên trên nó để điều khiển xe di chuyển.



*Hình 2.2: Robot omni*

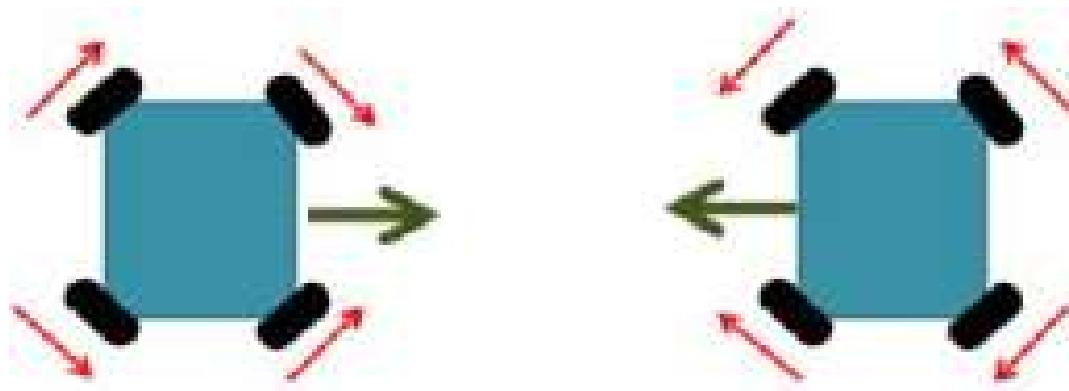
## 2.2. Cách di chuyển của robot omni

- Chuyển động tịnh tiến và chuyển động lùi:



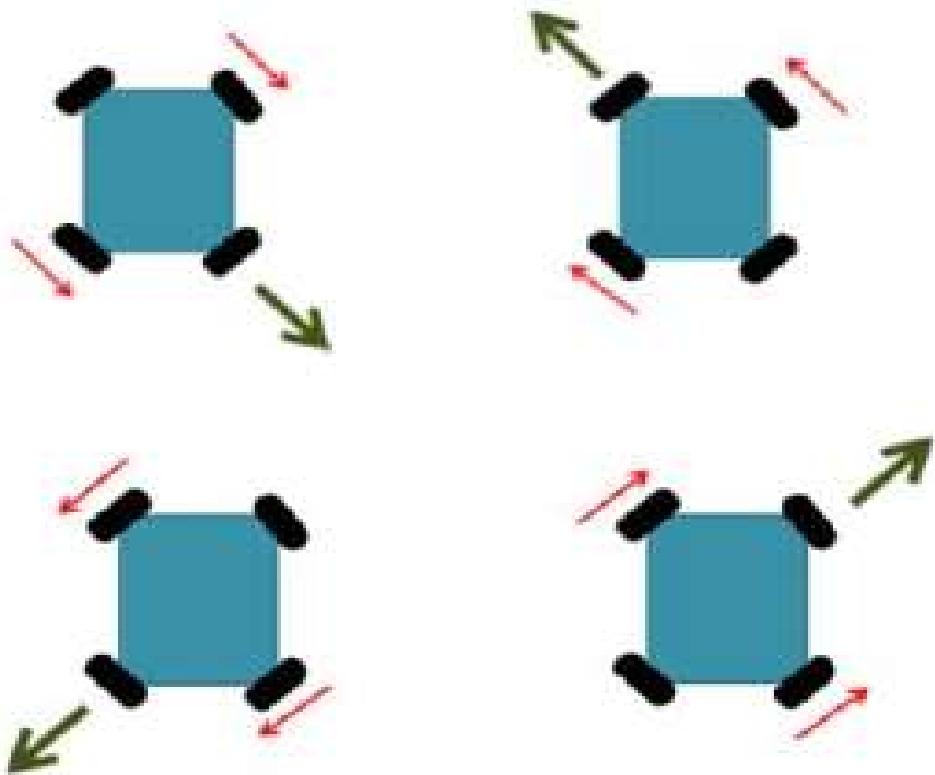
Hình 2.3: Chuyển động thẳng, lùi của robot omni

- Chuyển động sang trái, phải:



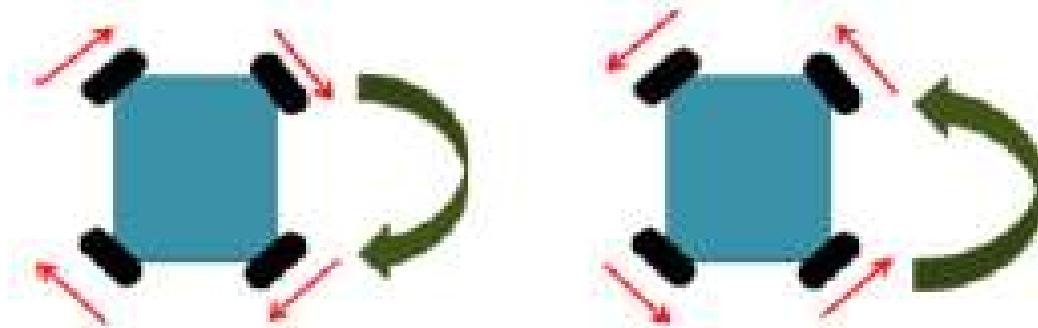
Hình 2.4: Chuyển động sang trái, phải của robot omni

- Chuyển động sang chéo góc:



Hình 2.5: Chuyển động chéo góc của robot omni

- Chuyển động xoay tròn:



Hình 2.6: Chuyển động xoay của robot omni

## CHƯƠNG 3. CÁC PHƯƠNG PHÁP ĐỊNH VỊ CHO ROBOT

### 3.1. Giới thiệu

Một robot di động thông minh phải được điều khiển dẫn đường theo một chiến lược có hiệu quả. Có nhiều nghiên cứu trên thế giới với các thuật giải và phương pháp khác nhau cho dẫn đường rõ bót trong các môi trường trong nhà và ngoài trời. Ta sẽ điểm qua một vài phương pháp như sau:

### 3.2. Các phương pháp định vị

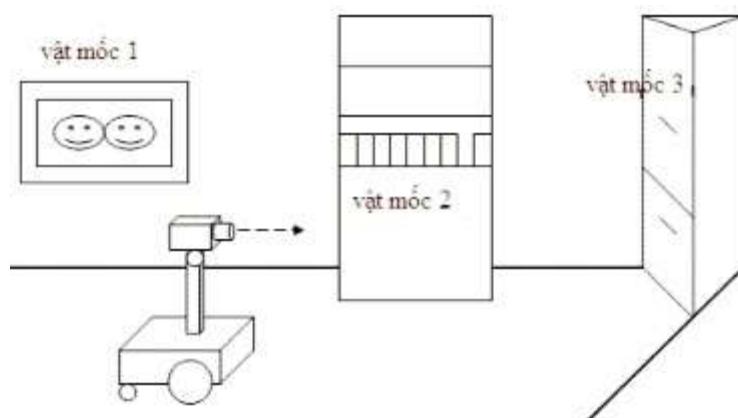
#### 3.2.1. Phương pháp Dead-Reckoning

Dead-reckoning là phương pháp dẫn đường được sử dụng rộng rãi nhất đối với robot di động. Phương pháp này cho độ chính xác trong thời gian ngắn, giá thành thấp và tốc độ lấy mẫu rất cao. Tuy nhiên do nguyên tắc cơ bản của phương pháp dead-reckoning là tích luỹ thông tin về gia tốc chuyển động theo thời gian do đó dẫn tới sự tích luỹ sai số. Sự tích luỹ sai số theo hướng sẽ dẫn đến sai số vị trí lớn tăng tỉ lệ với khoảng cách chuyển động của robot. Tuy nhiên hầu hết các nhà nghiên cứu đều đồng ý rằng dead-reckoning là một phần quan trọng trong hệ thống dẫn đường rõ bót, các lệnh dẫn đường sẽ được đơn giản hóa nếu độ chính xác của phương pháp dead-reckoning được cải thiện. Phương pháp dead-reckoning dựa trên phương trình đơn giản và thực hiện được một cách dễ dàng, sử dụng dữ liệu từ bộ mã hoá số vòng quay bánh xe. Dead-reckoning dựa trên nguyên tắc là chuyển đổi số vòng quay bánh xe thành độ dịch tuyến tính ứng của robot. Nguyên tắc này chỉ đúng với giá trị giới hạn. Có một vài lý do dẫn đến sự không chính xác trong việc chuyển từ số gia vòng quay bánh xe sang chuyển động tuyến tính. Tất cả các nguồn sai số này được chia thành 2 nhóm: sai số hệ thống và sai số không hệ thống. Để giảm sai số dead-reckoning cần phải tăng độ chính xác động học cũng như kích thước tối hạn.

#### 3.2.2. Hệ thống dẫn đường cột mốc chủ động

Hệ thống dẫn đường cột mốc chủ động là hệ thống dẫn đường được sử dụng phổ biến nhất trên tàu biển và máy bay. Hệ thống này cung cấp thông tin vị trí rất chính xác với quá trình xử lý tối thiểu. Hệ thống cho phép tốc độ lấy mẫu và độ tin cậy cao nhưng

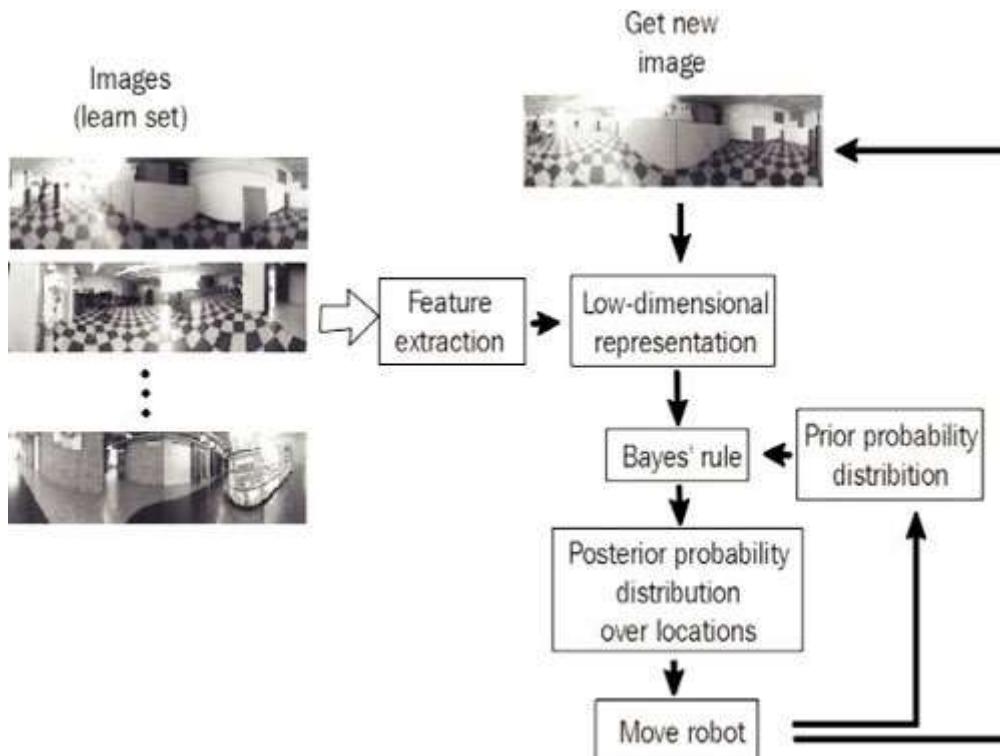
đi kèm với nó là giá thành cao trong việc thiết lập và duy trì. Cột mốc được đặt tại các vị trí chính xác sẽ cho phép xác định tọa độ chính xác của vật thể. Có 2 phương pháp đo dùng trong hệ thống cột mốc chủ động, đó là phép đo 3 cạnh tam giác và phép đo 3 góc tam giác. Phép đo 3 cạnh tam giác. Phép đo 3 cạnh tam giác xác định vị trí vật thể dựa trên khoảng cách đo được tới cột mốc biết trước. Trong hệ thống dẫn đường sử dụng phép đo này thông thường có ít nhất là 3 trạm phát đặt tại các vị trí biết trước ngoài môi trường và 1 trạm nhận đặt trên rô bốt. Hoặc ngược lại có 1 trạm phát đặt trên rô bốt và các trạm nhận đặt ngoài môi trường. Sử dụng thông tin về thời gian truyền của chùm tia hệ thống sẽ tính toán khoảng cách giữa các trạm phát cố định và trạm nhận đặt trên robot. GPS (Global Positioning Systems) - hệ thống định vị toàn cầu hoặc hệ thống cột mốc sử dụng cảm biến siêu âm là các ví dụ khi sử dụng phép đo 3 cạnh tam giác. Phép đo 3 góc tam giác.



Hình 3.1: Mô hình định vị cột mốc

### 3.2.3. Định vị sử dụng bản đồ

Robot sử dụng các cảm biến được trang bị để tạo ra một bản đồ cục bộ môi trường xung quanh. Bản đồ này sau đó so sánh với bản đồ toàn cục lưu trữ sẵn trong bộ nhớ. Nếu tương ứng, rô bốt sẽ tính toán vị trí và góc hướng thực tế của nó trong môi trường.



Hình 3.2: Mô hình xử lý thuật toán định vị bằng hình ảnh

Để nâng cao độ chính xác của việc định vị thì một giải pháp hiện nay là kết hợp hai hoặc nhiều phương pháp định vị nêu trên

### 3.2.4. Phương pháp định vị tuyệt đối (Absolute Positioning Method)

Trong các ứng dụng robot, có hai phương pháp định vị thường được sử dụng là phương pháp định vị tuyệt đối (APM) và phương pháp định vị tương đối (Relative Positioning Methods – RPM).

Phương pháp định vị tương đối chủ yếu dựa trên bài toán dead reckoning (tính số vòng quay của bánh xe để suy ra vị trí tương đối của robot sau một quá trình chuyển động).

Phương pháp này đơn giản, rẻ tiền và hầu như được sử dụng trong tất cả các robot di động, cho dù robot đó sử dụng phương pháp định vị APM. Khuyết điểm của phương pháp dead reckoning là bán kính sai số lớn, và lại là sai số tích luỹ.

Phương pháp định vị tuyệt đối thì lại dùng thêm các cảm biến khác ngoài encoder như dùng beacons, cột mốc, so sánh các bản đồ cục bộ và bản đồ toàn cục, định vị bằng vệ tinh (GPS)... Mỗi phương pháp APM dùng các thuật toán và cảm biến khác nhau, nhưng đều là những phương pháp chủ yếu cho các robot di động ngày nay. Người ta thường dùng phương pháp RPM để giới hạn phạm vi xử lý trước khi dùng phương pháp APM cho việc định vị robot.

### **3.3. Ưu điểm, nhược điểm của các phương pháp**

Chúng ta có thể phân tích một vài ưu khuyết điểm của phương pháp APM để có cái nhìn tổng quát về việc định vị cho robot như sau:

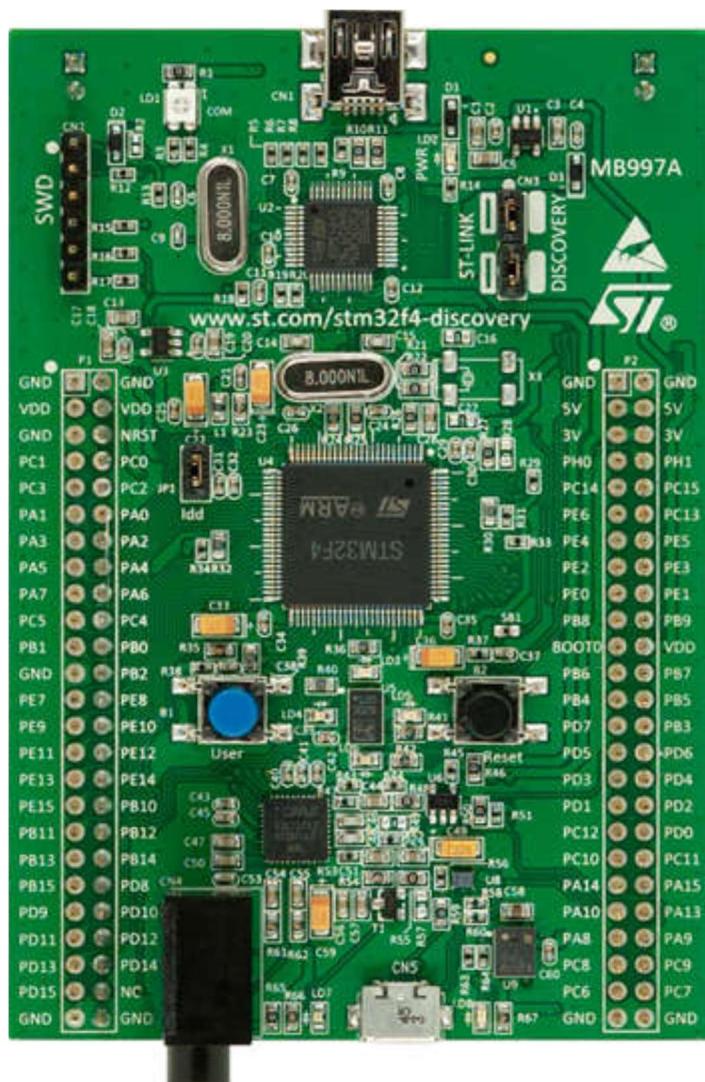
- Phương pháp dùng beacons chỉ thích hợp dùng trong nhà và nó cũng giống như định vị bằng cột mốc.
- Phương pháp so sánh bản đồ thường dùng cho các robot tự trị phức tạp.
- Phương pháp định vị dùng GPS thì chỉ thích hợp dùng ngoài trời, và cũng cho sai số rất lớn (lên đến hàng mét).

Một số phương pháp định vị khác thì dùng cảm biến la bàn hoặc con quay để định hướng cho robot nhờ vào từ trường trái đất...

## CHƯƠNG 4. CẤU TRÚC HỆ THỐNG ROBOT OMNI

### 4.1. Kit STM32F407 Discovery

Bộ kit STM32F4-DISCOVERY với vi điều khiển hiệu suất cao STM32F407VGT6, cho phép người dùng dễ dàng phát triển các ứng dụng xử lý tín hiệu số (hình ảnh, video...). Nó bao gồm một công cụ ST-LINK tích hợp sẵn trên bảng mạch giúp nạp chương trình, gỡ lỗi nhanh chóng.



Hình 4.1: Kit STM32F407VG Discovery

Thông số kỹ thuật chính:

- Vi điều khiển sử dụng nền tảng 32-bit ARM Cortex®-M4
- Xung Clock tối đa: 192 Mhz
- Bộ nhớ Flash: 1 MB
- Header mở rộng cho tất cả LQFP100 I/O.

Ở đê tài lán này, Kit STM32F407 đảm nhiệm vai trò vi xử lý trung tâm cho các tác vụ xử lý, giao tiếp các module, cảm biến và tính toán.

#### **4.2. Module cảm biến la bàn số HMC5883L**

GY-271 HMC5883L là mô đun cảm biến la bàn số dùng để đo từ trường theo 3 trực X, Y, Z, xác định độ lệch của vật thể so với cực Bắc, Nam. Thông số kỹ thuật của module GY271 như sau:

- Chip xử lý: HMC5883L
- Nguồn cung cấp :3-5V
- Cổng giao tiếp: I2C
- Dải đo  $\pm 1.3\text{-}8$  Gaussian



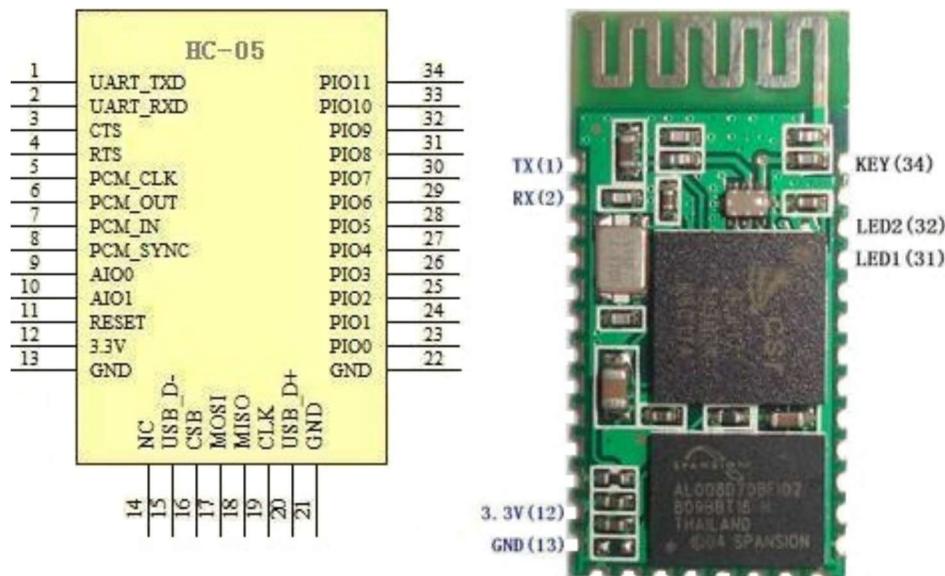
*Hình 4.2: Module la bàn số HMC5883L*

### 4.3. Module Bluetooth HC-06

Module Bluetooth HC-06 được thiết kế để chuyển đổi giao tiếp nối tiếp không đồng bộ và thành giao tiếp không dây Bluetooth và ngược lại.

Thông số kỹ thuật:

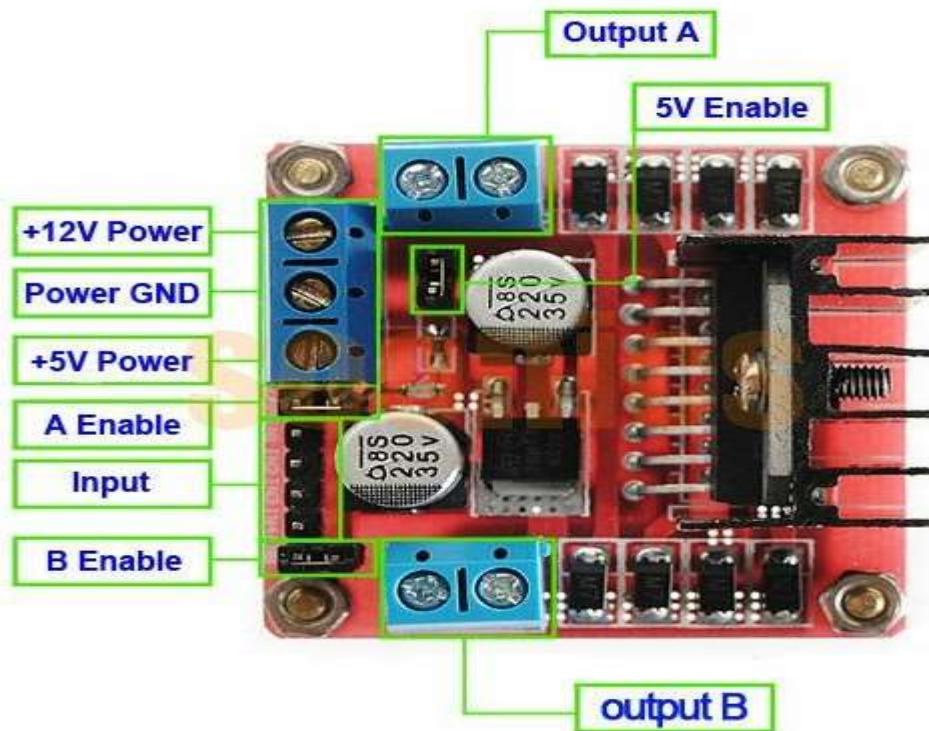
- Chuẩn Bluetooth: V2.0+EDR.
- Điện áp hoạt động: 3.3-5VDC, 30mA.
- Tần số: 2.4GHz.
- Tốc độ: 2.1Mbps (Max)/160kbps
- Tốc độ baudrate mặc định: 9600, 8bit dữ liệu, 1bit Stop. Hỗ trợ tốc độ baud: 9600, 19200, 38400, 57600, 115200, 230400, 460800.



Hình 4.3: Sơ đồ chân module HC-06

### 4.4. Module điều khiển động cơ L298N

Module điều khiển động cơ một chiều (DC Motor Driver) sử dụng chip cầu H L298 giúp điều khiển tốc độ và chiều quay của động cơ DC một cách dễ dàng, ngoài ra module L298 còn điều khiển được 1 động cơ bước lưỡng cực. Mạch cầu H của IC L298 có thể hoạt động ở điện áp từ 5V đến 35V.



Hình 4.4: Module cầu H L298N

Thông số kỹ thuật:

- Điện áp điều khiển: +5 V ~ +35 V
- Dòng tối đa cho mỗi cầu H: 2A
- Điện áp tín hiệu điều khiển: +5 V ~ +7 V
- Dòng tín hiệu điều khiển: 0 ~ 36mA
- Công suất hao phí: 20W (khi nhiệt độ T = 75 °C)
- Nhiệt độ vận hành: -25 °C ~ +130 °C

## CHƯƠNG 5. ĐỘNG CƠ VÀ ENCODER

### 5.1. Động cơ DC

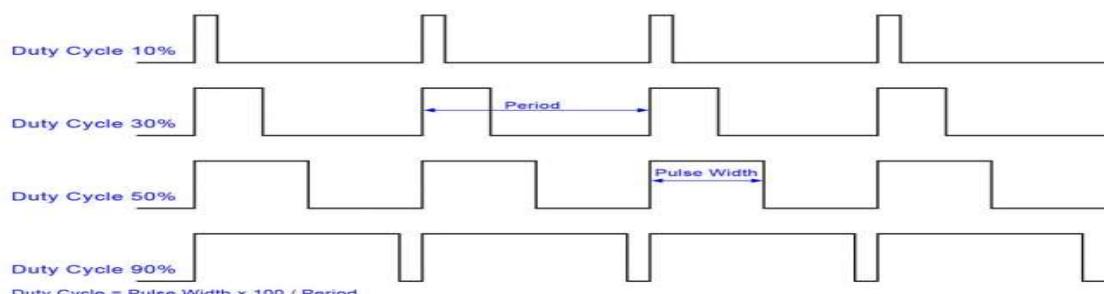
Động cơ DC là động cơ điện hoạt động với dòng điện một chiều. Động cơ điện một chiều được ứng dụng rộng rãi trong các ứng dụng dân dụng cũng như công nghiệp. Cấu tạo của động cơ điện một chiều gồm có hai phần: stator đứng yên và rotor quay so với stator. Phần cảm (phần kích từ-thường đặt trên stator) tạo ra từ trường đi trong mạch từ, xuyên qua các vòng dây quấn của phần ứng (thường đặt trên rotor). Khi có dòng điện chạy trong mạch phần ứng, các thanh dẫn phần ứng sẽ chịu tác động bởi các lực điện từ theo phương tiếp tuyến với mặt trụ rotor, làm cho rotor quay.

Tùy theo cách mắc cuộn dây rotor và stator mà người ta chia thành các loại động cơ sau:

- Động cơ kích từ độc lập: Cuộn dây kích từ (cuộn dây stator) và cuộn dây phần ứng (rotor) mắc riêng rẽ nhau, có thể cấp nguồn riêng biệt.
- Động cơ kích từ nối tiếp: Cuộn dây kích từ mắc nối tiếp với cuộn dây phần ứng

Đối với loại động cơ kích từ động lập, người ta có thể thay thế cuộn dây kích bởi nam châm vĩnh cửu, khi đó ta có loại động cơ điện một chiều dùng nam châm vĩnh cửu

Đối với loại động cơ kích từ độc lập dùng nam châm vĩnh cửu, để thay đổi tốc độ ta thay đổi điện áp cung cấp cho rotor hoạt động. Việc cấp áp một chiều thay đổi thường khó khăn do đó người ta sử dụng phương pháp điều chỉnh chế thay đổi độ rộng xung (Pulse Width Modulation-PWM)

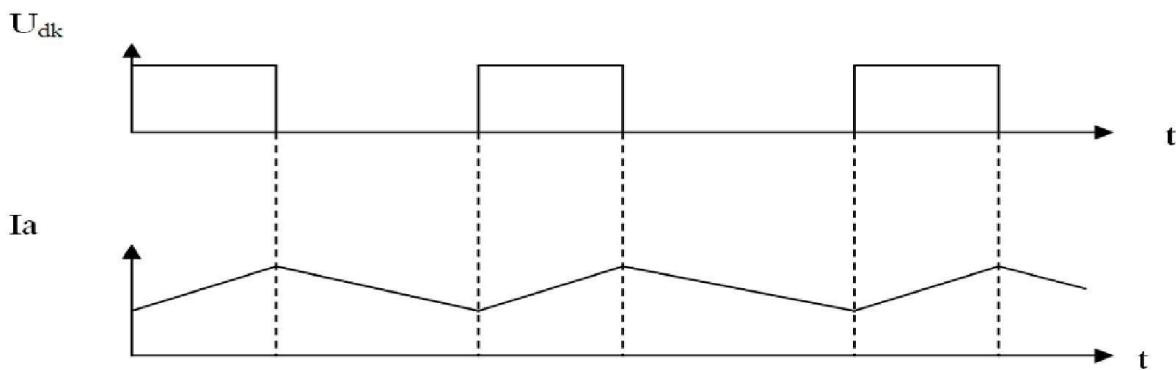


Hình 5.1: Điều chỉnh độ rộng xung PWM

Mạch điều khiển động cơ bằng phương pháp PWM hoạt động dựa theo nguyên tắc cấp nguồn cho động cơ bằng chuỗi xung đóng mở với tốc độ nhanh. Nguồn DC được chuyển đổi thành tín hiệu xung vuông (chỉ gồm hai mức 0 volt và xấp xỉ điện áp hoạt động). Tín hiệu xung vuông này được cấp cho động cơ. Nếu tần số chuyển mạch đủ lớn, động cơ sẽ chạy với một tốc độ đều đặn phụ thuộc vào momen của trực quay.

Với phương pháp PWM, ta điều chỉnh tốc độ động cơ thông qua việc điều chỉnh rộng xung, tức là thời gian mức cao của chuỗi xung vuông cấp cho động cơ trong một chu kỳ. Việc điều chỉnh này sẽ tác động đến công suất trung bình cấp cho động cơ, làm thay đổi tốc độ. Với độ rộng xung có độ rộng nhỏ thì động cơ sẽ chạy chậm và ngược lại nếu độ rộng xung càng lớn thì động cơ DC sẽ chạy nhanh.

Do đặc tính cảm kháng của động cơ, dòng qua động cơ là dòng liên tục có đồ thị gợn sóng phụ thuộc vào điện áp điều khiển như hình sau:

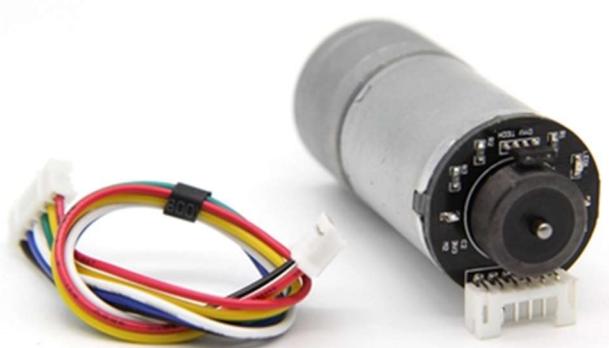


Hình 5.2: Mối quan hệ giữa dòng và áp trên động cơ DC

## 5.2. Động cơ DC servo GA-25

Động cơ DC Servo GA-25 là động cơ không đồng bộ bao gồm hai thành phần chính: động cơ DC và encoder. Ngoài ra, động cơ được gắn thêm hộp số (Gear box) với tỷ số truyền 1:34 có tác dụng làm tăng momen quay và giảm tốc độ động cơ. Động cơ DC servo nói chung được ứng dụng cho việc điều khiển chính xác góc quay, tốc độ, momen. Động cơ GA-25 có các thông số cơ bản như sau:

- Công suất định mức: 13.2W
- Điện áp: 3 – 12VDC
- Momen: 1.88 kgf.cm
- Đĩa Encoder 11 xung với số xung sau giảm tốc: 374 xung / vòng



Hình 5.3: Động cơ DC servo GA-25

### 5.3. Encoder

Encoder là dụng cụ dùng để xác định vị trí góc của đĩa quay hoặc xác định quãng đường di chuyển cho bánh xe, trực động cơ hay bất kỳ thiết bị nào có cơ cấu quay cần xác định độ dịch chuyển.

Có hai loại encoder là encoder tuyệt đối (absolute encoder) và encoder tương đối (incremental encoder). Encoder tuyệt đối cung cấp cho ta chính xác vị trí encoder mà không cần qua xử lý, trong khi encoder tương đối cần đòi hỏi thêm khâu xử lý từ tín hiệu trả về từ encoder trước khi xác định được vị trí quay của động cơ.

Ưu và nhược điểm của:

- Encoder tuyệt đối: Giữ được giá trị tuyệt đối khi encoder mất nguồn tuy nhiên giá thành cao vì chế tạo phức tạp, đọc tín hiệu ngõ ra khó.
- Encoder tương đối: Giá thành rẻ, chế tạo đơn giản, xử lý tín hiệu trả về dễ dàng tuy nhiên dễ bị sai lệch về xung khi trả về. Khi đó nếu hoạt động lâu dài sai số này sẽ tích lũy

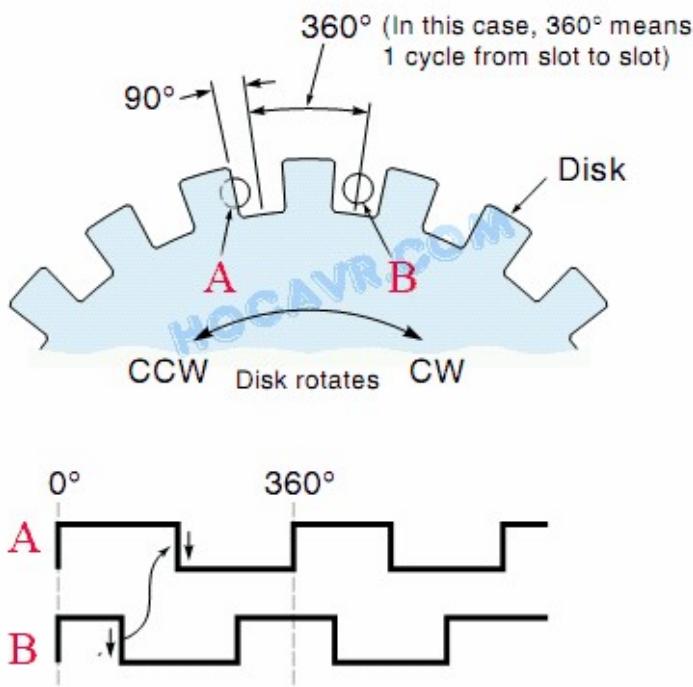
Ở đề tài lần này, ta tập trung nói về *incremental optical encoder* là loại encoder tương đối sử dụng cảm biến quang. Hệ thống optical encoder bao gồm một nguồn phát quang, một cảm biến quang và một đĩa có chia rãnh.



Hình 5.4: *Optical Encoder*

Encoder thường có 3 kênh (3 ngõ ra) bao gồm kênh A, kênh B và kênh I (Index). Trong hình trên ta thấy một lỗ nhỏ bên phía trong của đĩa quay và một cặp phát-thu dành riêng cho lỗ nhỏ này. Đó là kênh I của encoder. Cứ mỗi lần motor quay được một vòng, lỗ nhỏ xuất hiện tại vị trí của cặp phát-thu, hồng ngoại từ nguồn phát sẽ xuyên qua lỗ nhỏ đến cảm biến quang, một tín hiệu xuất hiện trên cảm biến. Như thế kênh I xuất hiện một “xung” mỗi vòng quay của motor. Bên ngoài đĩa quay được chia thành các rãnh nhỏ và một cặp thu-phát khác dành cho các rãnh này. Đây là kênh A của encoder, hoạt động của kênh A cũng tương tự kênh I, điểm khác nhau là trong 1 vòng quay của motor, có N “xung” xuất hiện trên kênh A. N là số rãnh trên đĩa và được gọi là độ phân giải (resolution) của encoder. Mỗi loại encoder có độ phân giải khác nhau, có khi trên mỗi đĩa chỉ có vài rãnh nhưng cũng có trường hợp đến hàng nghìn rãnh được chia. Để điều khiển động cơ, cần phải biết độ phân giải của encoder đang dùng. Độ phân giải ảnh hưởng đến độ chính xác điều khiển và cả phương pháp điều khiển. Không được vẽ trong hình 2, tuy nhiên trên các encoder còn có một cặp thu phát khác được đặt trên cùng đường tròn với kênh A nhưng lệch một chút (lệch M+0,5 rãnh), đây là kênh B của

encoder. Tín hiệu xung từ kênh B có cùng tần số với kênh A nhưng lệch pha 90 độ. Bằng cách phối hợp kênh A và B người đọc sẽ biết chiều quay của động cơ.



Hình 5.5: Hai kênh A và B lệch pha trong encoder

Khi cảm biến A bắt đầu bị che thì cảm biến B hoàn toàn nhận được hồng ngoại xuyên qua, và ngược lại. Hình trên là dạng xung ngõ ra trên 2 kênh. Xét trường hợp motor quay cùng chiều kim đồng hồ, tín hiệu “đi” từ trái sang phải. Khi tín hiệu A chuyển từ mức cao xuống thấp (cạnh xuống) thì kênh B đang ở mức thấp. Ngược lại, nếu động cơ quay ngược chiều kim đồng hồ, tín hiệu “đi” từ phải qua trái. Lúc này, tại cạnh xuống của kênh A thì kênh B đang ở mức cao. Như vậy, bằng cách phối hợp 2 kênh A và B chúng ta không những xác định được góc quay ( thông qua số xung) mà còn biết được chiều quay của động cơ ( thông qua mức của kênh B ở cạnh xuống của kênh A).

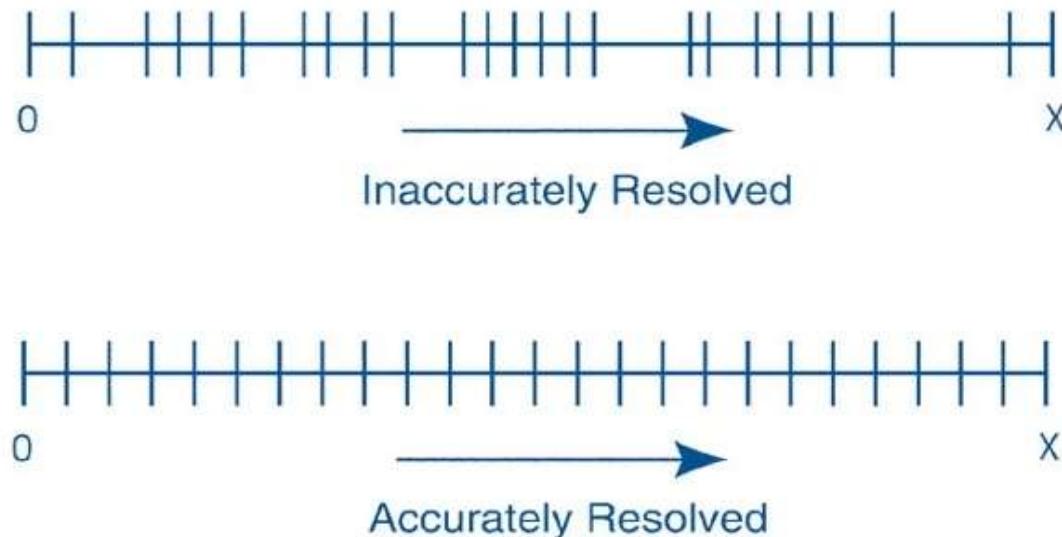
Encoder được sử dụng trong đề tài là loại 374 xung trên một vòng quay với hai kênh A và B.

#### 5.4. Giải thuật đọc tín hiệu từ encoder để tính tốc độ động cơ

Một số khái niệm cần quan tâm khi viết giải thuật đọc tín hiệu encoder: Độ phân giải, độ chính xác, độ lặp lại.

##### 5.4.1. Độ phân giải

Độ phân giải của encoder là số lần đọc của encoder thang đo tuyến tính. Độ phân giải bộ encoder thường được đo bằng xung trên mỗi chu kỳ quay đối với encoder tương đối và bằng bit đối với encoder tuyệt đối. Nói cách khác, Độ phân giải là mức độ chi tiết mà encoder có thể trả về. Trên đĩa encoder quang học, lượng vạch càng dày, độ phân giải càng cao.



Hình 5.6: Độ chính xác khi đọc tín hiệu encoder

Hiểu làm phỏ biến liên quan đến encoder là giả định rằng độ phân giải cao hơn cải thiện độ chính xác của toàn bộ hệ thống. Điều quan trọng cần lưu ý là độ phân giải cao sẽ không bù đắp cho việc giảm sai số. Với độ phân giải cao hơn, khả năng trả về vị trí của encoder được cải thiện nhưng nếu gặp sai số trong quá trình hoạt động, độ phân giải cao sẽ không cải thiện độ chính xác.

Độ phân giải thường sử dụng khi đọc encoder: x1, x2, x4.

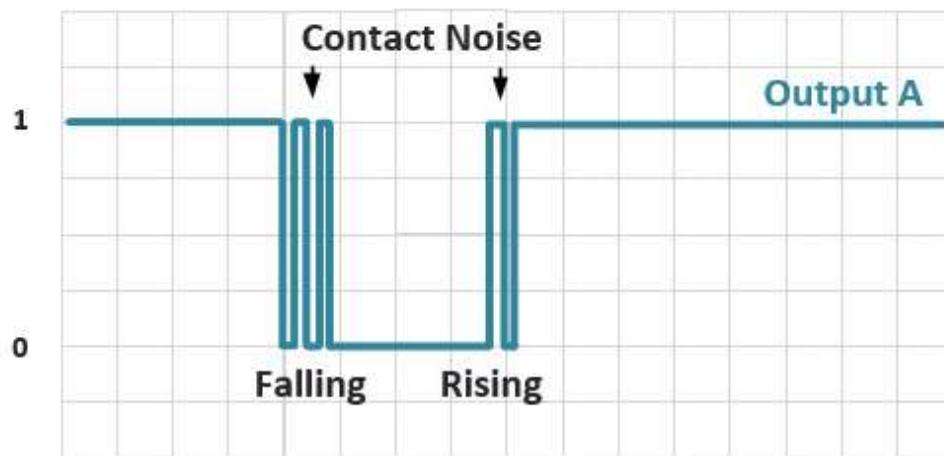
- Ở độ phân giải x1, ta chỉ quan tâm xung cạnh lên (hoặc cạnh xuống) ở một kênh (A hoặc B).
- Ở độ phân giải x2, ta chỉ quan tâm xung cạnh lên và cạnh xuống ở một kênh (A hoặc B).
- Ở độ phân giải x4, ta quan tâm xung cạnh lên và cạnh xuống ở cả hai kênh A và B.

Độ phân giải càng cao, việc đọc tín hiệu càng phức tạp.

Trong đề tài này, xác định vị trí chính xác là vấn đề quan trọng, độ phân giải x4 sẽ phù hợp để giải quyết.

### 5.4.2. Độ chính xác

Độ chính xác biểu thị cho sự sai lệch giữa giá trị trả về và giá trị thực tế của encoder. Độ chính xác ảnh hưởng bởi nhiều yếu tố phần cứng: độ rộng xung trên đĩa quang khác nhau, khe hở giữa các bánh răng của hộp số. Ngoài ra, cũng có những sai số xảy ra trong quá trình đọc tín hiệu encoder: khi trạng thái chân A (hoặc B) thay đổi đột ngột 0 – 5V, sẽ xuất hiện tín hiệu nhiễu (bounce). Những sai số này có thể tối thiểu bằng thuật toán ở mức nào đó.



Hình 5.7: Nhiễu khi đọc tín hiệu encoder

### 5.4.3. Các phương pháp đọc tín hiệu encoder để tính tốc độ và vị trí

Các phương pháp đọc tốc độ động cơ qua encoder đều dựa trên hai nguyên tắc:

- Đếm số xung trả về của encoder trong một chu kỳ:

Sử dụng counter hoặc ngắt ngoài của vi điều khiển để đếm số xung và timer để xác định chu kỳ lấy mẫu. Ưu điểm là đơn giản, ít phụ thuộc vào độ chính xác của encoder, nhược điểm là khả năng có sai số trong giá trị trả về cao.

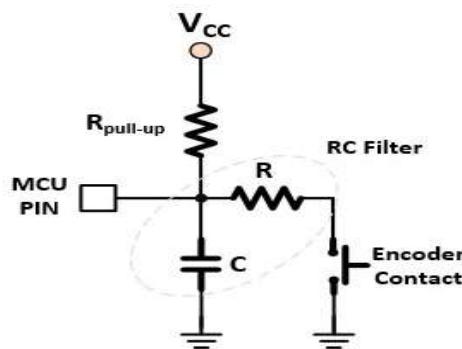
- Đo thời gian cần thiết để các kênh encoder thay đổi trạng thái.

Sử dụng input capture của vi điều khiển, bằng cách đo thời gian giữa hai lần thay đổi trạng thái của kênh encoder. Nói cách khác, phương pháp này đo độ rộng xung của hai kênh A và B. Ưu điểm là xác định chính xác được tốc độ hoặc vị trí động cơ do không bỏ sót xung trả về, nhược điểm là phức tạp, dễ ảnh hưởng bởi nhiễu, và phụ thuộc vào độ chính xác của encoder.

Trong đề tài này, sử dụng input capture sẽ là hợp lý hơn.

### 5.4.4. Xử lý nhiễu bằng phần cứng

Việc xử lý nhiễu trực tiếp bằng phần cứng sẽ cho kết quả chính xác mà không bị trễ. Bộ lọc thông thấp bậc 1 là hợp lý để xử lý nhiễu bounce vì tính đơn giản và hiệu quả của nó. Điện trở kéo lên để chuẩn hóa tín hiệu encoder.



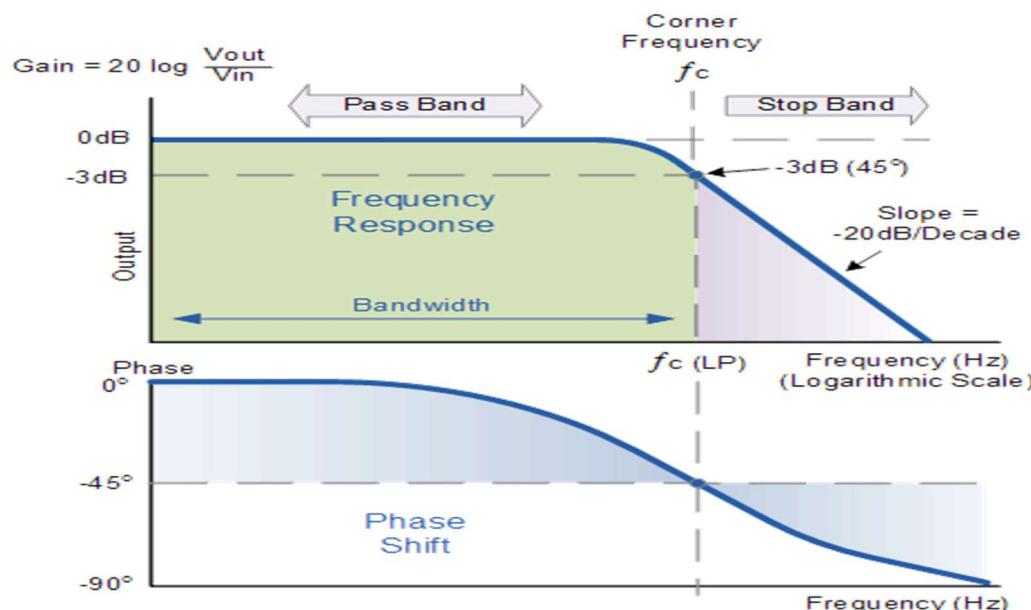
Hình 5.8: Sơ đồ mạch lọc thông thấp bậc 1 cho encoder

Đối với encoder của GA25, do điện trở kéo lên đã được tích hợp nên không cần thiết phải sử dụng thêm. Tính toán giá trị R, C:

Điện áp ngõ ra bộ lọc thông thấp có thể hiểu đơn giản như công thức chia áp:

$$U_c = U \frac{Z_c}{Z} = \sqrt{1 + (2\pi RC)^2}$$

$$\text{Độ lợi } G = 20 \log \left( \frac{U_c}{U} \right) = -10 \log(1 + (2\pi RC)^2)$$



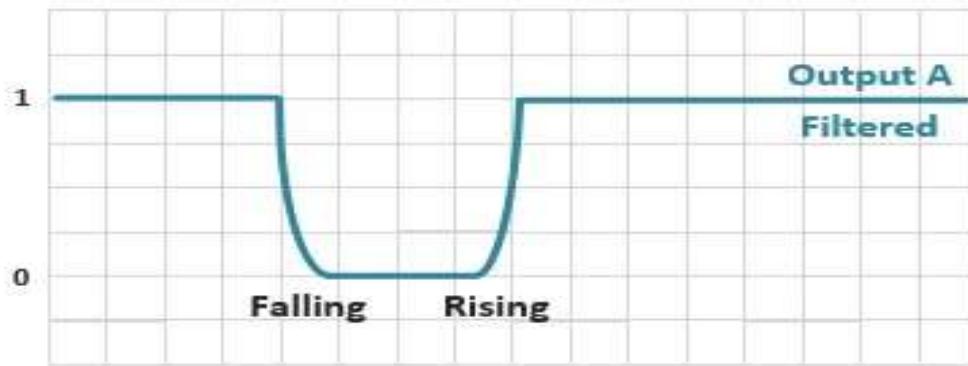
Hình 5.9: Biểu đồ Bode đáp ứng ngõ ra encoder

Biểu đồ plot cho thấy đáp ứng ngõ ra càng nhỏ khi tần số càng cao. Tần số截止 tại  $G = -3\text{db}$ ,  $U_c = 70.7\%U$ ;  $f_c = \frac{1}{2\pi}$

Do đề tài giới hạn tốc độ tối đa của động cơ  $\sim 200$  vòng/phút  $= 3,33$  vòng/s, tương ứng với mỗi chân A, B của encoder sẽ có tần số tối đa  $f_{max} = 3,33 * 374 = 1246,66$  Hz

Vậy cần chọn R, C sao cho  $f_{max} < f_c$ . Nói cách khác  $RC < 1,276 * 10^{-4}$

Chọn  $R = 10$  ( $k\Omega$ ). Suy ra  $C = 1.276 * 10^{-8}$  (F), chọn  $C = 10$  (nF)



Hình 5.10: Tín hiệu encoder sau khi lọc nhiễu

#### 5.4.5. Xử lý nhiễu bằng thuật toán

Bộ lọc Kalman ước lượng một quá trình bằng cách sử dụng một dạng của điều khiển phản hồi (feedback control): bộ lọc ước lượng trạng thái của quá trình tại một thời điểm sau đó có được phản hồi từ các đo đạc (có nhiễu). Như vậy, các phương trình của bộ lọc Kalman được chia thành hai nhóm: cập nhật theo thời gian (time update) và cập nhật theo giá trị đo lường (measurement update). Các phương trình cập nhật theo thời gian để dự đoán (projecting forward) trạng thái hiện tại và vecto hiệp phương sai lỗi nhằm ước lượng trạng thái tiền nghiệm cho bước tiếp theo. Các phương trình cập nhật theo giá trị đo lường dùng để cung cấp phản hồi – ví dụ như kết hợp một giá trị đo lường mới với ước lượng tiền nghiệm để có được ước lượng trạng thái hậu nghiệm.

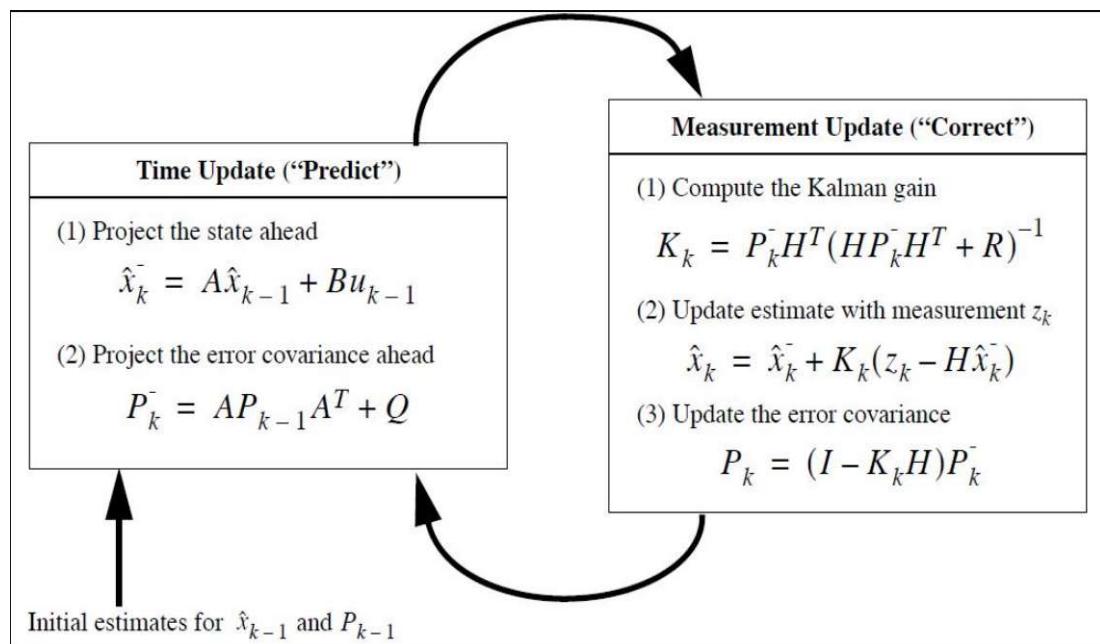
- Các tham số của bộ lọc và điều chỉnh (Filter Parameters and Tuning)

Khi cài đặt bộ lọc Kalman, hiệp phương sai của lỗi đo lường  $R$  thường được đo đạc trước để điều khiển bộ lọc hoạt động. Đo đạc hiệp phương sai của lỗi đo lường  $R$  là hoàn toàn khả thi vì cần đo lường quá trình bằng mọi cách (trong khi bộ lọc hoạt động) do vậy cần đo đạc một số mẫu trước (off-line) để xác định phương sai của sai số đo.

Việc xác định hiệp phương sai của nhiễu quá trình  $Q$  nói chung là khó hơn vì thông thường khó có khả năng quan sát trực tiếp quá trình đang ước lượng. Đôi khi một

mô hình hóa đơn giản của quá trình có thể cho kết quả chấp nhận được nếu như chúng ta “thêm vào” (“injects”) quá trình một sự không chắc chắn thông qua việc chọn Q.

Trong cả hai trường hợp chúng ta có hoặc không chọn được giá trị hợp lý cho các tham số của bộ lọc thì thường sau một khoảng thời gian, chất lượng của bộ lọc sẽ được cải thiện bằng cách điều chỉnh (tuning) các tham số Q và R của bộ lọc. Quá trình điều chỉnh thường thực hiện tĩnh (off-line), thông thường với sự giúp sức của một bộ lọc Kalman khác làm nhiệm vụ xác định hệ thống (system identification).

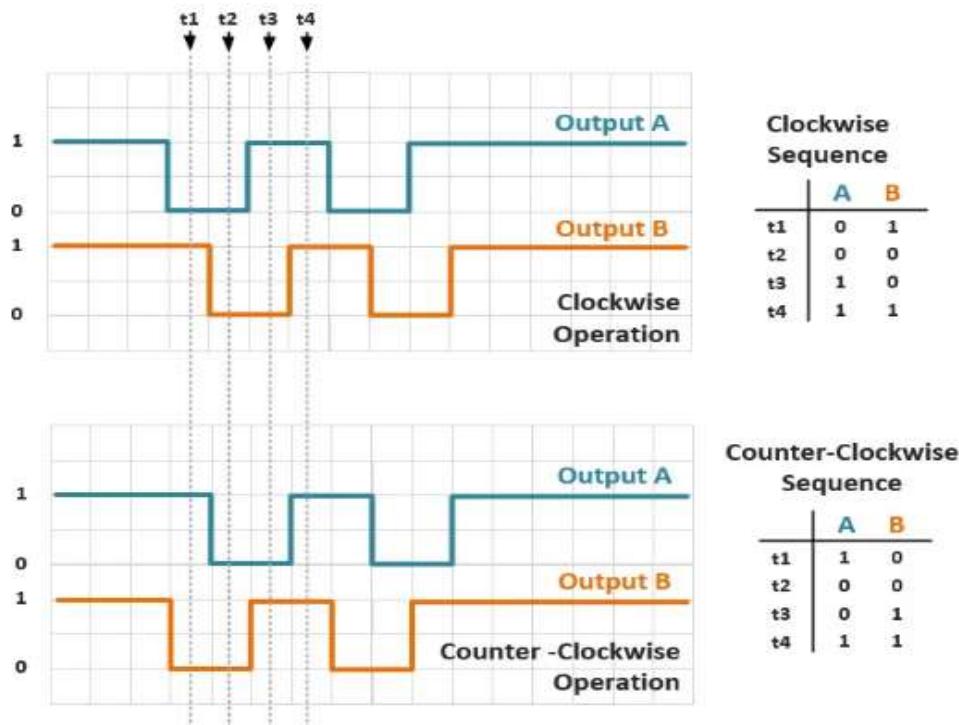


Hình 5.11: Bộ lọc Kalman đơn giản

- Các bước áp dụng bộ lọc Kalman vào việc đọc tín hiệu encoder:
  - ✓ Đọc tín hiệu thô của encoder
  - ✓ Uớc lượng các thông số R, Q
  - ✓ So sánh thay đổi trước và sau bộ lọc để điều chỉnh trong các trường hợp tín hiệu encoder khác nhau

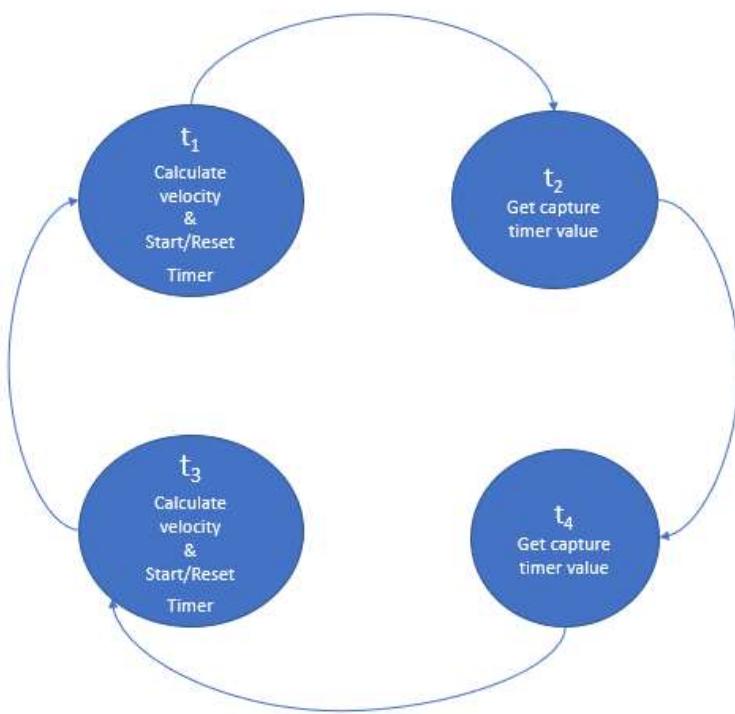
#### 5.4.6. Giải thuật

Xác định 4 trạng thái của hai kênh A và B.



Hình 5.12: Bốn trạng thái của kênh A, B encoder

4 trạng thái sẽ tuần tự lặp lại, t1-t2-t3-t4. Khi có nhiễu xảy ra, tín hiệu chỉ dao động trong trạng thái hiện tại và trạng thái ngay trước đó. Ví dụ: nhiễu xảy ra khi chuyển trạng thái từ 1 sang 2, tín hiệu đọc về sẽ chỉ là t1 hoặc t2. Tương tự ngược lại khi encoder quay ngược chiều.



Hình 5.13: Chu trình đọc tín hiệu từ encoder

Tương tự khi encoder quay ngược chiều

## CHƯƠNG 6. ĐIỀU KHIỂN VÀ ĐỊNH VỊ

### 6.1. Bộ điều khiển Dynamic Surface Control (DSC)

#### 6.1.1. Giới thiệu chung

Các robot được ứng dụng trong đời sống ngày càng nhiều và mở rộng. Các nghiên cứu gần đây đều tập trung vào hướng tăng độ linh hoạt của robot khi hoạt động trong các môi trường, địa hình khác nhau. Các hướng nghiên cứu về cơ khí tập trung vào các cơ cấu, cơ chế thiết bị chấp hành truyền động linh hoạt và thích ứng. Các nghiên cứu về lập trình điều khiển tập trung vào hướng điều khiển thích nghi, thông minh, xử lý ảnh vào tăng độ linh hoạt, tốc độ và thông minh cho robot. Omni Robot chuyển động đa hướng là một dạn robot holonomic, tốc độ và linh hoạt trong di chuyển.

Đã có nhiều công trình nghiên cứu về việc điều khiển hệ robot di động holonomic. Tuy nhiên, có một số các công trình đã bỏ qua độn lực học của robot. Điều này là do các vấn đề thiết kế bộ điều khiển sẽ cực kỳ khó khăn khi sự phức tạp của hệ thốn độn lực tăng.

Kỹ thuật sử dụng bộ điều khiển trượt sliding mode gấp nhược điểm lớn khi điều khiển bám cho robot đó là hiện tượng rung (chattering). Backstepping cũng đã được đề xuất như là một trong các phương pháp đại diện cho việc điều khiển robot di động holonomic có xem xét chuyển động học và động lực học. Tuy nhiên, bộ điều khiển backstepping tồn tại sự ảnh hưởng của nhiễu, khi mà độ phức tạp của hệ thống càng cao thì nhiễu càng lớn. D.Swaroop và C. Gerdes đã đề xuất thuật toán điều khiển DSC để giải quyết vấn đề này bằng cách sử dụng một bộ lọc bậc nhất cho từng bộ điều khiển ảo được tổng hợp ở từng bước của quy trình thiết kế backstepping.

Theo đó, việc sử dụng thuật toán DSC sẽ thích ứng vượt qua mọi thành phần nhiễu và sự không chính xác của các tham số. Dựa trên tiêu chuẩn Lyapunov, tất cả các tín hiệu trong hệ thống vòng kín ổn định và có thể được điều chỉnh nhỏ tùy ý bằng cách điều chỉnh các thông số thiết kế.

### 6.1.2. Thuật toán DSC

Thuật toán DSC được phát triển từ thuật toán MSSC và kĩ thuật Backstepping. Áp dụng cho hệ truyền ngược. Ví dụ sau đây sẽ giúp hiểu rõ về các bước thực hiện của thuật toán.

Xét hệ phi tuyến:

$$\begin{aligned} x_1 &= x_2 + f(x_1) \\ x_2 &= u \end{aligned} \quad (1)$$

Với  $f$  và  $\frac{\partial f}{\partial x_1}$  là hàm liên tục, mục tiêu điều khiển điểm  $x_1$  đến điểm  $x_{1d}$

Đầu tiên ta xây dựng mặt trượt thứ nhất:  $S_1 = x_1 - x_{1d}$ . Đạo hàm  $S_1$  dựa vào (1):

$$\dot{S}_1 = x_2 + f - x_{1d} \quad (2)$$

Tiếp theo ta xét mặt trượt thứ hai:  $S_2 = x_2 - x_{2d}$

Trong đó  $x_{2d}$  gọi là đầu vào ảo được thiết kế để lái  $S_1 \rightarrow 0$ . Đạo hàm  $S_2$ :

$$\dot{S}_2 = u - \dot{x}_{2d} \quad (3)$$

$u$  sẽ được thiết kế để  $S_2 \rightarrow 0$ , tức là  $x_2 \rightarrow x_{2d}$ ,  $x_{2d}$  sẽ được chọn như sau:

$$x_{2d} = \dot{x}_{1d} - f - K_1 S_1 \quad (4)$$

Hàng số dương  $K_1$  sẽ được xác định sau,  $u$  được chọn là:

$$u = \dot{x}_{2d} - K_2 S_2 \quad (5)$$

Từ (4) và (5) ta có:

$$\begin{aligned} \dot{S}_1 &= S_2 - K_1 S_1 \\ \dot{S}_2 &= -K_2 S_2 \end{aligned} \quad (6)$$

Chọn hàm Lyapunov  $V = \frac{S_1^2 + S_2^2}{2}$ . Ta xét tính ổn định của hệ thống, từ (6) ta tính đạo hàm của V:

$$V = S_1 \dot{S}_1 + S_2 \dot{S}_2 = -K_1 S_1^2 - K_2 S_2^2 + S_1 S_2 \quad (7)$$

Ta có bất đẳng thức:

$$S_1 S_2 \leq \frac{S_1^2 + S_2^2}{2} \quad (8)$$

Từ (7) và (8), ta có:

$$\dot{V} \leq -K_1 S_1^2 - K_2 S_2^2 + \frac{S_1^2 + S_2^2}{2}$$

Chọn  $K_1 = K_2 = \frac{1}{2}$  ta có  $\dot{V} \leq 0$ . Theo định lý Lyapunov thì hệ ổn định.

Một nhược điểm của phương pháp này là sự xuất hiện của “nhiều bất thường” trong tính toán đầu vào ảo  $x_{2d}$  dẫn tới tính toán đầu vào thực u đưa ra kết quả không phù hợp. Để khắc phục nhược điểm đó, người ta sử dụng thêm một bộ lọc thông thấp cho các đầu vào ảo. Đây chính là thuật toán điều khiển DSC.

Ta đặt  $\bar{x}_2 = \dot{x}_{1d} - f - K_1 S_1$  và  $x_{2d}$  được tính qua bộ lọc là khâu quan trọng bậc nhất:

$$\tau \dot{x}_{2d} + x_{2d} = \bar{x}_2$$

Đặt  $\xi = x_{2d} - \bar{x}_2$  ta có hệ phương trình:

$$\begin{aligned}\dot{S}_1 &= S_2 + \xi - K_1 S_1 \\ \dot{S}_2 &= -K_2 S_2 \\ \dot{\xi} &= \frac{-\xi}{\tau} + \eta\end{aligned}$$

Lúc này hàm Lyapunov được chọn là:  $V = \frac{S_1^2 + S_2^2 + \xi^2}{2}$

$$\dot{V} = S_1 \dot{S}_1 + S_2 \dot{S}_2 + \xi \dot{\xi} = S_1(S_2 + \xi - K_1 S_1) - K_2 S_2^2 + \xi \left( \frac{-\xi}{\tau} + \eta \right) \quad (9)$$

Sử dụng các bất đẳng thức:

$$S_1 S_2 \leq \frac{S_1^2 + S_2^2}{2}; S_1 \xi \leq \frac{S_1^2 + \xi^2}{2}; \xi \eta \leq \frac{\eta^2 \xi^2 + 1}{2}$$

Từ (9) ta có:

$$\begin{aligned} \dot{V} &\leq -K_1 S_1^2 - K_2 S_2^2 - \frac{\xi^2}{\tau} + \frac{2S_1^2 + S_2^2 + \xi^2}{2} + \frac{\eta^2 \xi^2 + 1}{2} \\ &= (-K_1 + 1) S_1^2 + \left( -K_2 + \frac{1}{2} \right) S_2^2 + \left( \frac{-1}{\tau} + \frac{1}{2} + \frac{M^2}{2} \right) \xi^2 + \frac{\eta^2 \xi^2 + 1}{2} - \frac{M^2 \xi^2}{2} \\ &= (-K_1 + 1) S_1^2 + \left( -K_2 + \frac{1}{2} \right) S_2^2 + \left( \frac{-1}{\tau} + \frac{1}{2} + \frac{M^2}{2} \right) \xi^2 + \frac{1}{2} + \frac{(\eta^2 - M^2) \xi^2}{2} \end{aligned}$$

Chọn  $K_1 = 1 + K; K_2 = \frac{1}{2} + K; \frac{1}{\tau} = K + \frac{1}{2} + \frac{M^2}{2}$

Xét tập B chứa  $\begin{bmatrix} S_1 \\ S_2 \\ \xi \end{bmatrix}$  thỏa mãn  $S_1^2 + S_2^2 + \xi^2 \leq 2p$  với  $p > 0$  là tập kín và lồi. Như vậy luôn

tìm được hằng số dương M để  $\eta \leq M$ . Như vậy  $\dot{V} \leq -2KV + \frac{1}{2}$ . Theo định lý Lyapunov thì hệ ổn định.

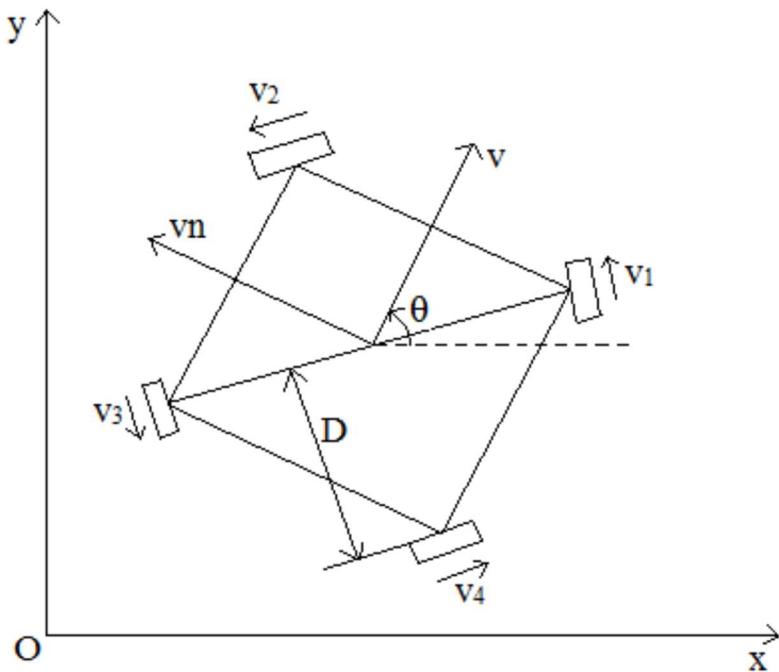
### 6.1.3. Mô hình hóa robot omni

Ở hình dưới ta thấy, bốn bánh xe được đặt lệch nhau  $90^\circ$ , khoảng cách từ bánh đến tâm xe là  $D$  Oxy là trục tọa độ toàn cục,  $v$  là vận tốc thẳng của xe,  $vn$  là vận tốc theo pháp tuyến của xe và  $\omega$  là vận tốc góc của robot omni.

$$v_i = \omega_i r \quad (\text{với } i=1,2,3,4, \dots)$$

$\omega_i$  là vận tốc góc bánh  $i$

$q = [x \ y \ \theta]^T$  là vector tọa độ và hướng của robot trong hệ tọa độ toàn cục.



Hình 6.1: Mô hình robot omni 4 bánh

Chiếu các vector  $v$ ,  $vn$  lên hệ trục tọa độ toàn cục Oxy ta có:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ vn \\ \omega \end{bmatrix} \quad (10)$$

Mặt khác  $[v \ v n \ \omega]^T$  được tính theo vận tốc các bánh như sau :

$$\begin{bmatrix} v \\ v n \\ \omega \end{bmatrix} = \begin{bmatrix} \frac{k}{2} & \frac{-k}{2} & \frac{-k}{2} & \frac{k}{2} \\ \frac{k}{2} & \frac{k}{2} & \frac{-k}{2} & \frac{-k}{2} \\ \frac{1}{4D} & \frac{1}{4D} & \frac{1}{4D} & \frac{1}{4D} \end{bmatrix} \begin{bmatrix} v1 \\ v2 \\ v3 \\ v4 \end{bmatrix}$$

Với  $k = \cos 45^\circ = \frac{\sqrt{2}}{2}$

#### 6.1.4. Mô hình động lực học

Phương trình động lực học của Omni robot dựa trên công thức Euler – Lagrange:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + \tau_d = B(q)\tau \quad (11)$$

Trong đó

- +  $q$  là vector biến khớp.  $q = [x \ y \ \theta]^T$
- +  $\tau_d$  là nhiễu bất định ( $\approx 0$ ).
- +  $\tau$  là vector tín hiệu vào (ở đây là momen lực)
- +  $M(q)$  là ma trận hệ số quán tính và momen quán tính.
- +  $C(q, \dot{q})$  là ma trận hệ số nhót và hướng tâm.
- +  $G(q)$  là ma trận trọng lực.
- +  $B(q)$  là ma trận chuyển đổi đầu vào.

Phân tích phương trình động năng của robot như sau:

$$K = \frac{1}{2}m\dot{x}^2 + \frac{1}{2}m\dot{y}^2 + \frac{1}{2}J\dot{\theta}^2$$

Trong đó:

- +  $m$  là khối lượng xe
- +  $J$  là momen quán tính của xe

Theo công thức Euler – Lagrange ta có  $L = K - P$  (với  $P$  là thế năng của robot = 0)

$$\Rightarrow L = K = \frac{1}{2}m\dot{x}^2 + \frac{1}{2}m\dot{y}^2 + \frac{1}{2}J\dot{\theta}^2$$

Thực hiện một số phép biến đổi phương trình trên với Momen hoặc lực đặt vào biến khớp  $q$  ta có các thông số:

$$C(q, \dot{q})\dot{q} = G(q) = 0$$

$$M(q) = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & J \end{bmatrix}$$

$$B(q) = \frac{1}{r} \begin{bmatrix} \cos \theta_1 & \cos \theta_2 & \cos \theta_3 & \cos \theta_4 \\ \sin \theta_1 & \sin \theta_2 & \sin \theta_3 & \sin \theta_4 \\ D & D & D & D \end{bmatrix}$$

$$\theta_1 = \theta + \frac{\pi}{4}; \theta_2 = \theta + \frac{3\pi}{4}; \theta_3 = \theta - \frac{3\pi}{4}; \theta_4 = \theta - \frac{\pi}{4};$$

Trong đó:

- +  $r$  là bán kính bánh xe
- +  $D$  là khoảng cách từ bánh đến tâm robot

### 6.1.5. Thiết kế bộ điều khiển

Từ phương trình động học

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

Ta có:

$$\begin{aligned}\ddot{x} &= \dot{v} \cos \theta - \dot{v} n \sin \theta - v \dot{\theta} \sin \theta - v n \dot{\theta} \cos \theta \\ \ddot{y} &= \dot{v} \sin \theta + \dot{v} n \cos \theta + v \dot{\theta} \cos \theta - v n \dot{\theta} \sin \theta \\ \ddot{\theta} &= \dot{\omega} \\ \Rightarrow \ddot{x} \cos \theta + \ddot{y} \sin \theta &= \dot{v} - \omega v n - \ddot{x} \sin \theta + \ddot{y} \cos \theta = \dot{v} n - \omega v\end{aligned}$$

Từ phương trình động lực học (11) ta được:

$$\begin{aligned}&\begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & J \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{\theta} \end{bmatrix} \\ &= \frac{1}{r} \begin{bmatrix} \cos(\theta + 45^\circ) & \cos(\theta + 135^\circ) & \cos(\theta - 135^\circ) & \cos(\theta - 45^\circ) \\ \sin(\theta + 45^\circ) & \sin(\theta + 135^\circ) & \sin(\theta - 135^\circ) & \sin(\theta - 45^\circ) \\ D & D & D & D \end{bmatrix} \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \\ \tau_4 \end{bmatrix}\end{aligned}$$

Cũng từ phương trình động lực học, ta có:

$$\begin{aligned}\ddot{x} \cos \theta + \ddot{y} \sin \theta &= \frac{k}{mr} (\tau_1 - \tau_2 - \tau_3 + \tau_4) \\ -\ddot{x} \sin \theta + \ddot{y} \cos \theta &= \frac{k}{mr} (\tau_1 + \tau_2 - \tau_3 - \tau_4) \\ \ddot{\theta} &= \frac{D}{Jr} (\tau_1 + \tau_2 + \tau_3 + \tau_4)\end{aligned}$$

Đặt:

$$\begin{aligned}\tau &= (\tau_1 - \tau_2 - \tau_3 + \tau_4), \tau_n = (\tau_1 + \tau_2 - \tau_3 - \tau_4) \\ \tau_\omega &= (\tau_1 + \tau_2 + \tau_3 + \tau_4)\end{aligned}$$

$$\text{Suy ra: } \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \\ \tau_4 \end{bmatrix} = \frac{1}{4} \begin{bmatrix} 1 & 1 & 1 \\ -1 & 1 & 1 \\ -1 & -1 & 1 \\ 1 & -1 & 1 \end{bmatrix} \begin{bmatrix} \tau_n \\ \tau_\omega \end{bmatrix}$$

Phương trình phía trên sẽ trở thành:

$$\dot{v} - \omega v n = \frac{k}{mr} \tau, \dot{vn} + \omega v = \frac{k}{mr} \tau_n, \dot{\omega} = \frac{D}{Jr} \tau_\omega$$

Ta xác định các biến trạng thái như sau:

$$x_1 = q, x_2 = [v \quad vn \quad \omega]^T$$

$$u = \left[ \frac{k}{mr} \tau \quad \frac{k}{mr} \tau_n \quad \frac{D}{Jr} \tau_\omega \right]$$

Kết hợp với phương trình động học  $\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ vn \\ \omega \end{bmatrix}$  ta có:

$$\dot{x}_1 = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} x_2$$

$$\dot{x}_2 = u + \begin{bmatrix} \omega v n \\ -\omega v \\ 0 \end{bmatrix}$$

#### 6.1.6. Áp dụng cho đối tượng Omnidroid với hai mặt trượt S1, S2

Gọi  $x_d, y_d, \theta_d$  là các giá trị đặt của quỹ đạo robot

Xét hệ:

$$\begin{aligned}\dot{x}_1 &= \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} x_2 = B^{-1} x_2 \\ \dot{x}_2 &= u + \begin{bmatrix} \omega v n \\ -\omega v \\ 0 \end{bmatrix} = u + f\end{aligned}\quad (3.3)$$

Với  $x_1 = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$  và  $x_2 = \begin{bmatrix} v \\ vn \\ \omega \end{bmatrix}$

$$S_1 = \begin{bmatrix} S_{11} \\ S_{12} \\ S_{13} \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x - x_d \\ y - y_d \\ \theta - \theta_d \end{bmatrix} = B(x_1 - x_{1d})$$

- Xét mặt trượt

Đạo hàm  $S_1$  kết hợp với hệ phương trình (3.3) ta có:

$$\begin{aligned}\dot{S}_1 &= \dot{B}(x_1 - x_{1d}) + BB^{-1}x_2 - B\dot{x}_{1d} \\ &= \begin{bmatrix} -\omega \sin \theta & \omega \cos \theta & 0 \\ -\omega \cos \theta & -\omega \sin \theta & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x - x_d \\ y - y_d \\ \theta - \theta_d \end{bmatrix} + \begin{bmatrix} v \\ vn \\ \omega \end{bmatrix} - \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x}_d \\ \dot{y}_d \\ \dot{\theta}_d \end{bmatrix}\end{aligned}$$

Bỏ qua thành phần  $\dot{B}(x_1 - x_{1d})$

$$\text{Chọn } x_{2d} = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x}_d \\ \dot{y}_d \\ \dot{\theta}_d \end{bmatrix} - K_1 S_1 = \begin{bmatrix} \dot{x}_d \cos \theta + \dot{y}_d \sin \theta - k_{11} S_{11} \\ -\dot{x}_d \sin \theta + \dot{y}_d \cos \theta - k_{12} S_{12} \\ \dot{\theta}_d - k_{13} S_{13} \end{bmatrix}$$

$$S_2 = \begin{bmatrix} S_{21} \\ S_{22} \\ S_{23} \end{bmatrix} = x_2 - x_{2d}$$

Xét mặt trượt

Đạo hàm  $S_2$  kết hợp với hệ phương trình (3.3) ta có:

$$\dot{S}_2 = \dot{x}_2 - \dot{x}_{2d} = u + f - \dot{x}_{2d}$$

Chọn  $u = \dot{x}_{2d} - f - k_2 S_2$

Từ phương trình  $\begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \\ \tau_4 \end{bmatrix} = \frac{1}{4} \begin{bmatrix} 1 & 1 & 1 \\ -1 & 1 & 1 \\ -1 & -1 & 1 \\ 1 & -1 & 1 \end{bmatrix} \begin{bmatrix} \tau \\ \tau_n \\ \tau_\omega \end{bmatrix}$  và cách đặt  $u$  ta có:

$$\begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \\ \tau_4 \end{bmatrix} = \frac{1}{4} \begin{bmatrix} 1 & 1 & 1 \\ -1 & 1 & 1 \\ -1 & -1 & 1 \\ 1 & -1 & 1 \end{bmatrix} \begin{bmatrix} \frac{mr}{k} & 0 & 0 \\ 0 & \frac{mr}{k} & 0 \\ 0 & 0 & \frac{Jr}{D} \end{bmatrix} u$$

## 6.2. Phương pháp định vị Dead-Reckoning

### 6.2.1. Giới thiệu

Dead reckoning là một phương pháp phổ biến dùng để xác định vị trí tương đối của robot di động trong mặt phẳng, sau một quá trình di chuyển nào đó, so với vị trí ban đầu. Việc giải bài toán dead reckoning thực ra là giải bài toán động học cho robot di động, nhằm mục đích xác định vị trí và hướng tức thời của robot sau một quá trình chuyển động, trong đó toạ độ gốc là toạ độ gắn với sàn của robot ở thời điểm ban đầu.

Chúng ta cần phân biệt hai loại sai số trong phương pháp dead reckoning là sai số ngẫu nhiên và sai số hệ thống. Trong đó, sai số ngẫu nhiên là các sai số do tương tác của robot với môi trường, ví dụ như ngoại lực, nội lực, những vật cản bất thường, sự trượt của bánh xe... Sai số hệ thống là các sai số do khi ta sử dụng các công thức tinh toán động học lý tưởng, trong khi luôn có những sai lệch không thể quản lý được trong thiết kế như là hai bánh xe không bằng nhau, khoảng cách hai bánh xe không xác định...

### 6.2.2. Tầm quan trọng của bài toán dead reckoning

Như vậy, cho dù dùng bất kỳ phương pháp định vị nào đi nữa, phương pháp dead reckoning dường như là phương pháp luôn luôn cần thiết. Một lý luận khác là muốn điều khiển robot di động thì phải giải bài toán động học cho robot, mà nếu đã giải bài toán động học rồi thì tại sao lại không dùng phương pháp dead reckoning để giới hạn phạm vi xử lý cho các phương pháp định vị tuyệt đối khác?

Tuy nhiên, một điều không ngờ đến là trong cuộc thi các robot sử dụng phương pháp dead reckoning để định vị và di chuyển được tổ chức ở San Jose, California năm 1992, hai robot thắng giải nhất và giải nhì là hai robot dùng phương pháp sửa lỗi UMBmark do nhóm của J. Borenstein thực hiện và nó có thể di chuyển 100 m mà không có sai số. Hay cụ thể hơn là robot có thể “đạo quanh” một phòng ngủ 5m x 5m 5-6 vòng mà không hề có sai số. Với lợi điểm rẻ tiền, đơn giản phương pháp dead reckoning là một giải pháp đầu tiên và cần thiết cho các robot di động.

### 6.2.3. Điều kiện để định vị tốt

Trong thực tế, khi robot hoạt động trên một mặt sàn tương đối tốt, phẳng, thì sai số chủ yếu là các sai số hệ thống. Nhưng nếu mặt sàn lồi lõm thì các sai số ngẫu nhiên lại là vấn đề cần được quan tâm. Trong đề tài này, chúng ta chủ yếu quan tâm đến các sai số hệ thống, bởi vì giả thiết là robot hoạt động trong không gian hai chiều, và điều kiện thí nghiệm là điều kiện của phòng thí nghiệm.

Bài toán đặt ra là làm sao xác định được các sai số của robot trong các ứng dụng thực, bởi vì việc tính toán vị trí dựa vào các thông số ghi nhận từ encoder thành độ dịch chuyển của bánh xe. Nhưng trong thực tế, không bao giờ có sự tuyến tính hoàn hảo trong các ứng dụng robot như vậy. Sự không tuyến tính sẽ gây ra các sai số. Chúng ta cần nhận diện hai loại sai số và một vài nguyên nhân chính dẫn đến các sai số đó.

#### 6.2.4. Áp dụng phương pháp định vị vào Omni robot

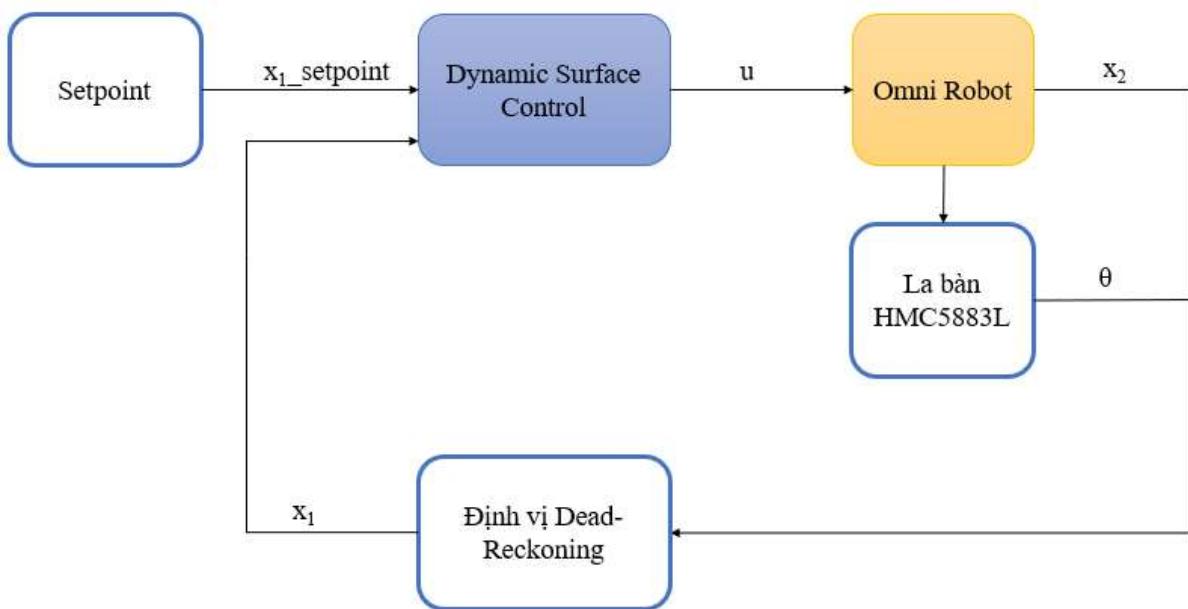
Thông tin tốc độ từ encoders trên từng bánh và mô hình động học được mô tả:

$$x_2 = \begin{bmatrix} v \\ vn \\ \omega \end{bmatrix}$$

Tính toán vị trí tiếp theo của robot

$$\begin{bmatrix} x(k) \\ y(k) \\ \theta(k) \end{bmatrix} = \begin{bmatrix} x(k-1) \\ y(k-1) \\ \theta(k-1) \end{bmatrix} + T \cdot \frac{x_2 + pre\_x_2}{2}$$

### 6.3. Sơ đồ khái thuật toán điều khiển



Hình 6.2: Sơ đồ khái thuật toán điều khiển

## CHƯƠNG 7. GIAO DIỆN ĐIỀU KHIỂN VÀ GIÁM SÁT

### 7.1. Thư viện OpenCV

#### 7.1.1. Tổng quan

OpenCV (OpenSource Computer Vision) là một thư viện mã nguồn mở. OpenCV được phát hành theo giấy phép BSD, do đó nó hoàn toàn miễn phí cho cả học thuật và thương mại. Nó có các interface C++, C, Python, Java và hỗ trợ Windows, Linux, Mac OS, iOS và Android. OpenCV được thiết kế để tính toán hiệu quả và với sự tập trung nhiều vào các ứng dụng thời gian thực. Được viết bằng tối ưu hóa C/C++, thư viện có thể tận dụng lợi thế của xử lý đa lõi. Được sử dụng trên khắp thế giới, OpenCV có cộng đồng hơn 47 nghìn người dùng và số lượng download vượt quá 6 triệu lần. Phạm vi sử dụng từ nghệ thuật tương tác, cho đến lĩnh vực khai thác mỏ, bản đồ trên web hoặc công nghệ robot.

#### 7.1.2. Một số module của OpenCV

- Core

Đây là module chứa các cấu trúc, class cơ bản mà OpenCV sẽ sử dụng trong việc lưu trữ và xử lý hình ảnh như Mat, Scale, Point, Vec... và các phương thức cơ bản sử dụng cho các module khác

- imgproc

Đây là một module xử lý hình ảnh của OpenCV bao gồm các bộ lọc (filter) linear và non-linear và các phép biến đổi hình học (transformation) như resize hoặc các phép biến đổi affine, chuyển đổi hệ màu và các thuật toán liên quan đến histogram (biểu đồ) của hình ảnh.

- highgui

Đây là một module cho phép tương tác với người dùng trên UI (User Interface) như hiển thị hình ảnh, video capturing.

- features2d

Module tìm các đặc trưng (feature) của hình ảnh. Trong module có implement các thuật toán rút trích đặc trưng như PCA...

- calib3d

Hiệu chuẩn máy ảnh và xây dựng lại 3D.

- objdetect

Module cho việc phát hiện các đối tượng như khuôn mặt, đôi mắt, cốc, người, xe hơi, vv trong hình ảnh. Các thuật toán được sử dụng trong module này là Haar-like Features.

- ml

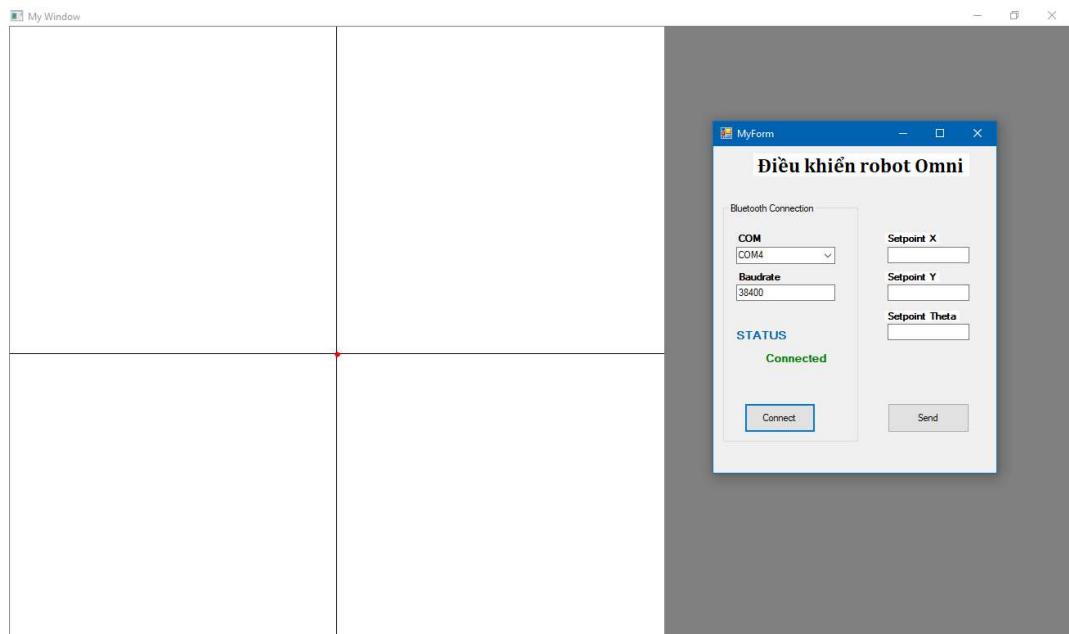
Module này chứa các thuật toán về Machine Learning phục vụ cho các bài toán phân lớp (Classification) và bài toán gom cụm (Clustering). Ví dụ như thuật toán SVM (Support Vector Machine), ANN...

- video

Module phân tích video gồm ước lượng chuyển động, trừ nền, và các thuật toán theo dõi đối tượng (object tracking).

## 7.2. Giao diện điều khiển và giám sát

Giao diện điều khiển được thiết kế bằng Visual Studio và thư viện OpenCV



Hình 7.1: Giao diện điều khiển

Giao tiếp bằng bluetooth sẽ gặp một số nhược điểm:

- Giao tiếp dễ bị nhiễu, dẫn đến truyền sai dữ liệu
- Khoảng cách truyền ngắn
- Giao tiếp nối tiếp cần thời gian để truyền nhận đủ số byte
- Có độ trễ nhất định

## CHƯƠNG 8. THỰC NGHIỆM VÀ ĐÁNH GIÁ

### 8.1. Thực hiện mô phỏng đáp ứng sau khi qua bộ điều khiển

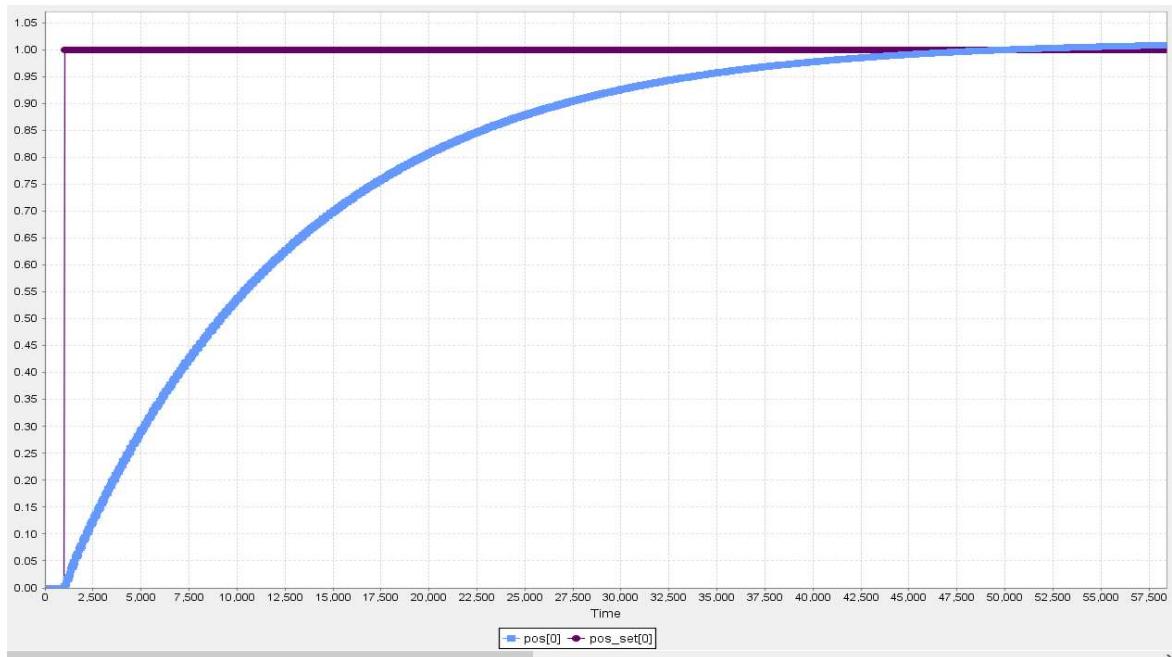
Chu kỳ lấy mẫu  $T = 30\text{ms}$

Thời hằng của khâu quan tính  $t=0.15$

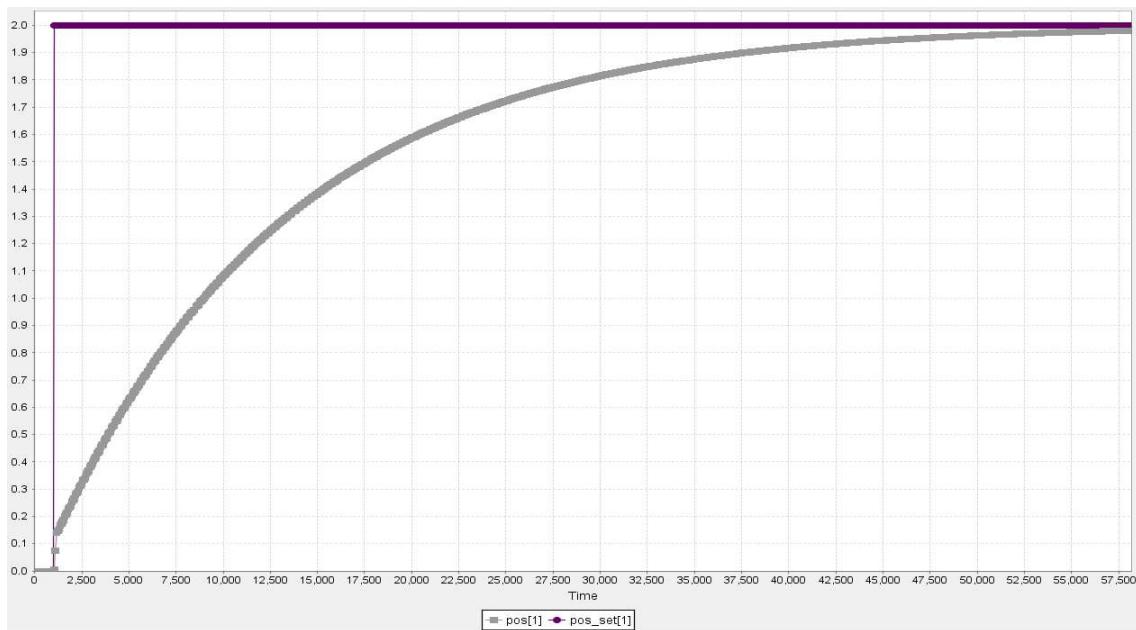
Vị trí đặt là  $\{x, y, \theta\} = \{1, 2, \arctan(2)\}$

Kết quả mô phỏng:

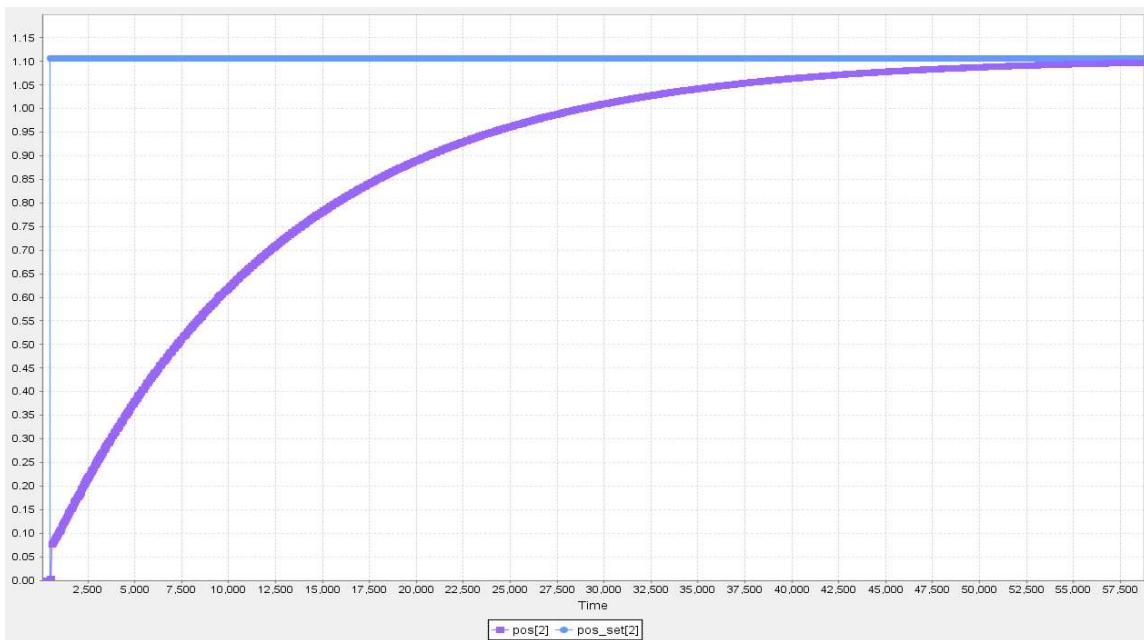
Với  $k_{11} = k_{12} = k_{13} = 3$



Hình 8.1. Đáp ứng theo trục tọa độ x với  $k_{11} = k_{12} = k_{13} = 3$

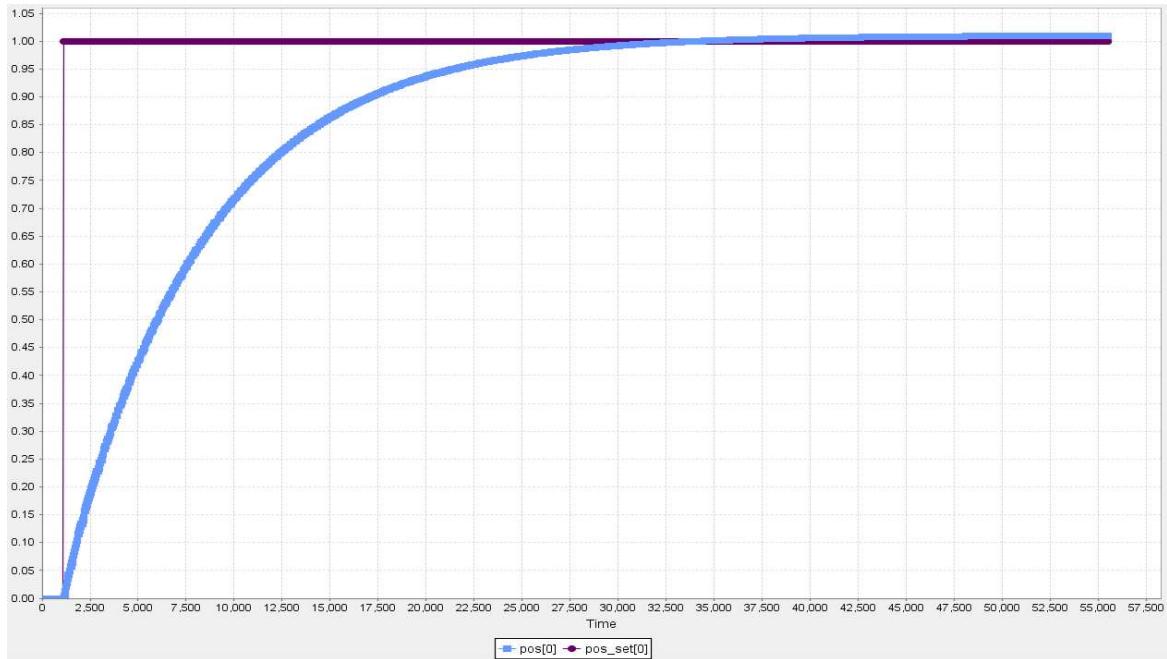


*Hình 8.2. Đáp ứng theo trục tọa độ y với  $k_{11} = k_{12} = k_{13} = 3$*

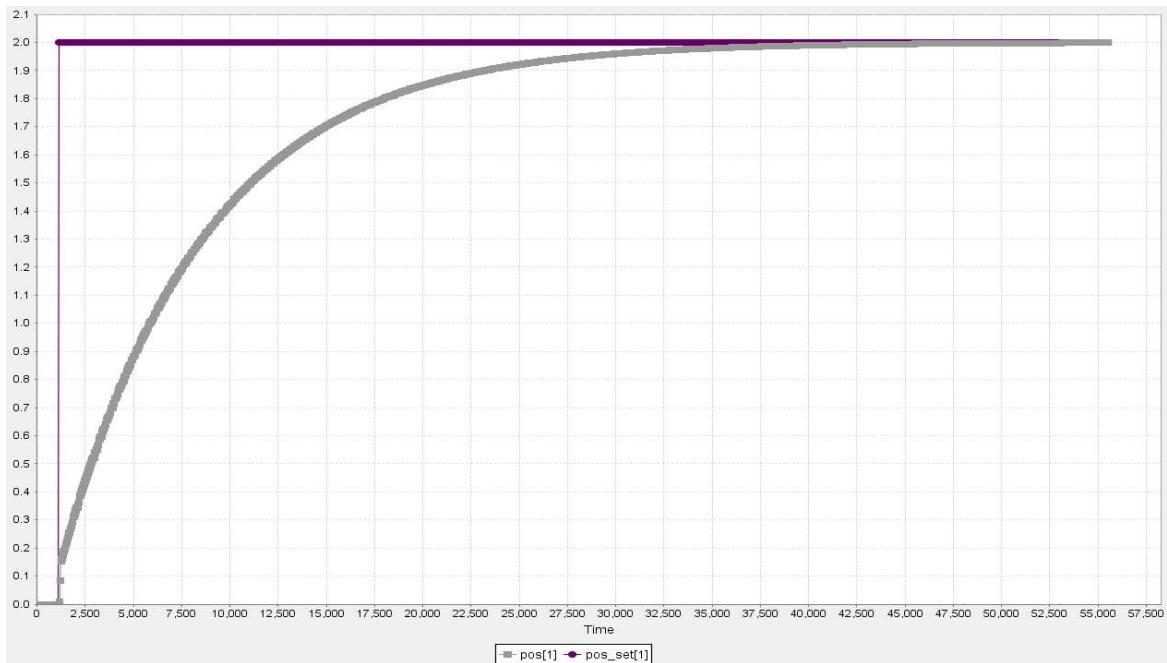


*Hình 8.3. Đáp ứng góc với  $k_{11} = k_{12} = k_{13} = 3$*

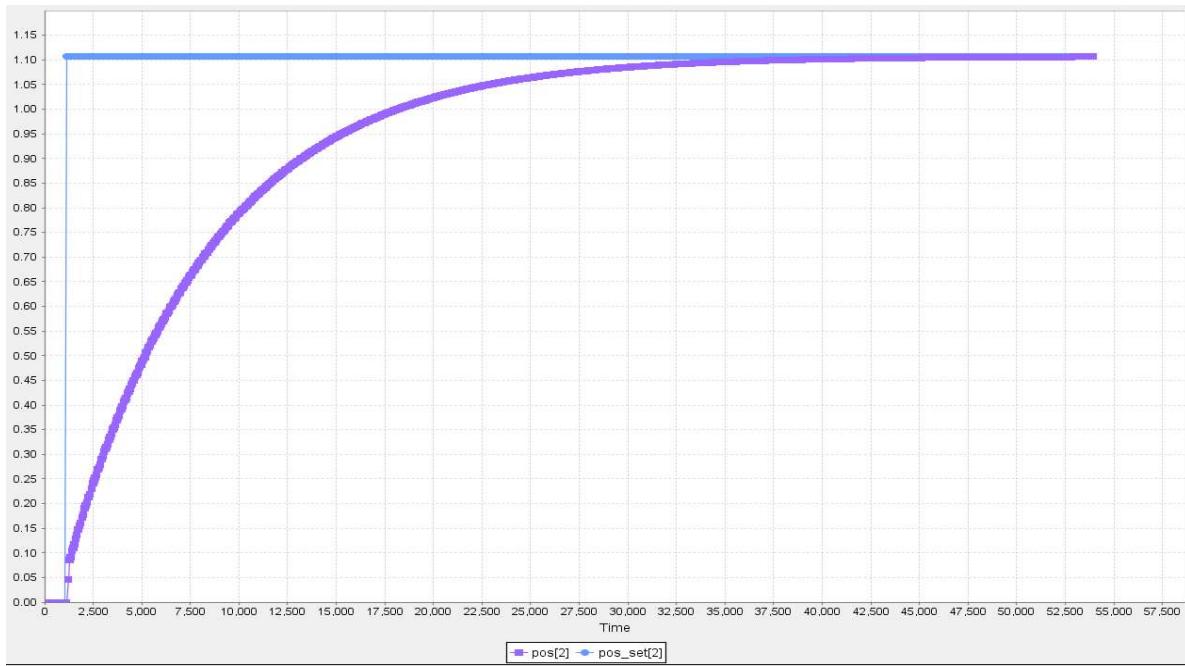
Với  $k_{11} = k_{12} = k_{13} = 5$



Hình 8.4. Đáp ứng theo trục tọa độ  $x$  với  $k_{11} = k_{12} = k_{13} = 5$

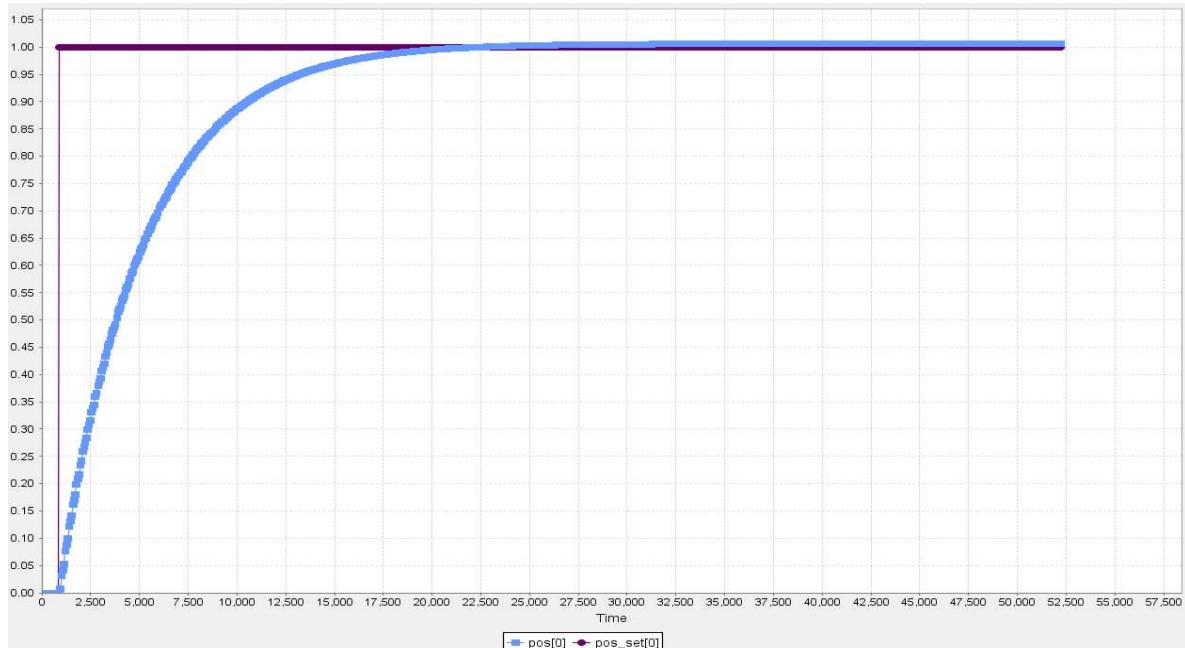


Hình 8.5. Đáp ứng theo trục tọa độ  $y$  với  $k_{11} = k_{12} = k_{13} = 5$

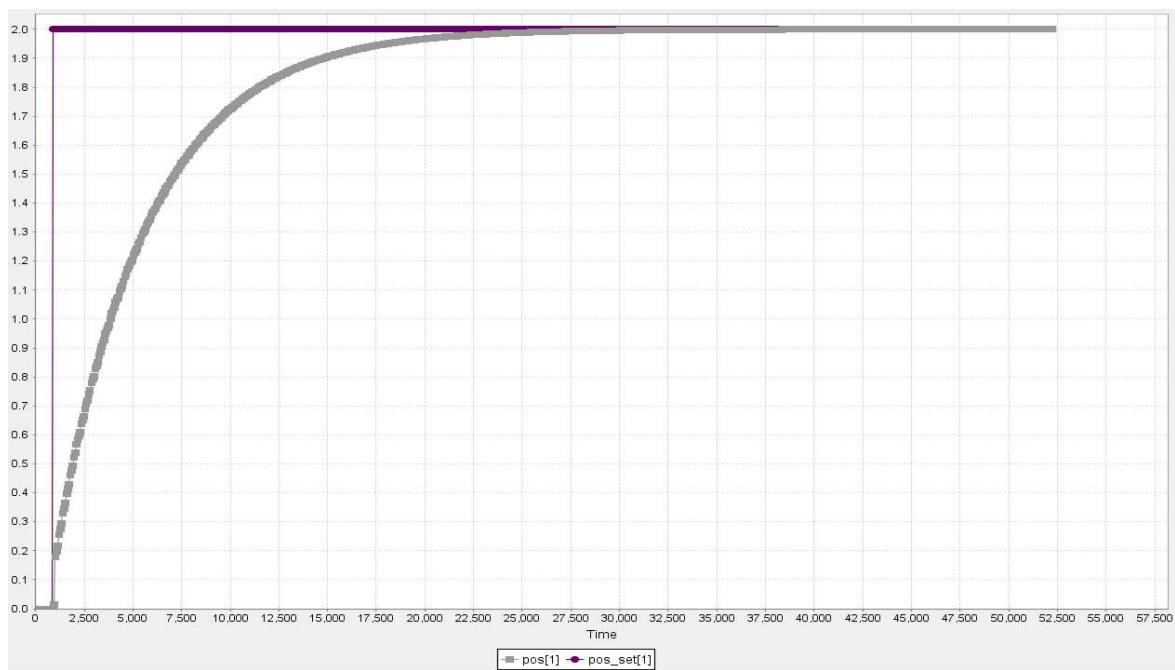


Hình 8.6. Đáp ứng góc với  $k_{11} = k_{12} = k_{13} = 5$

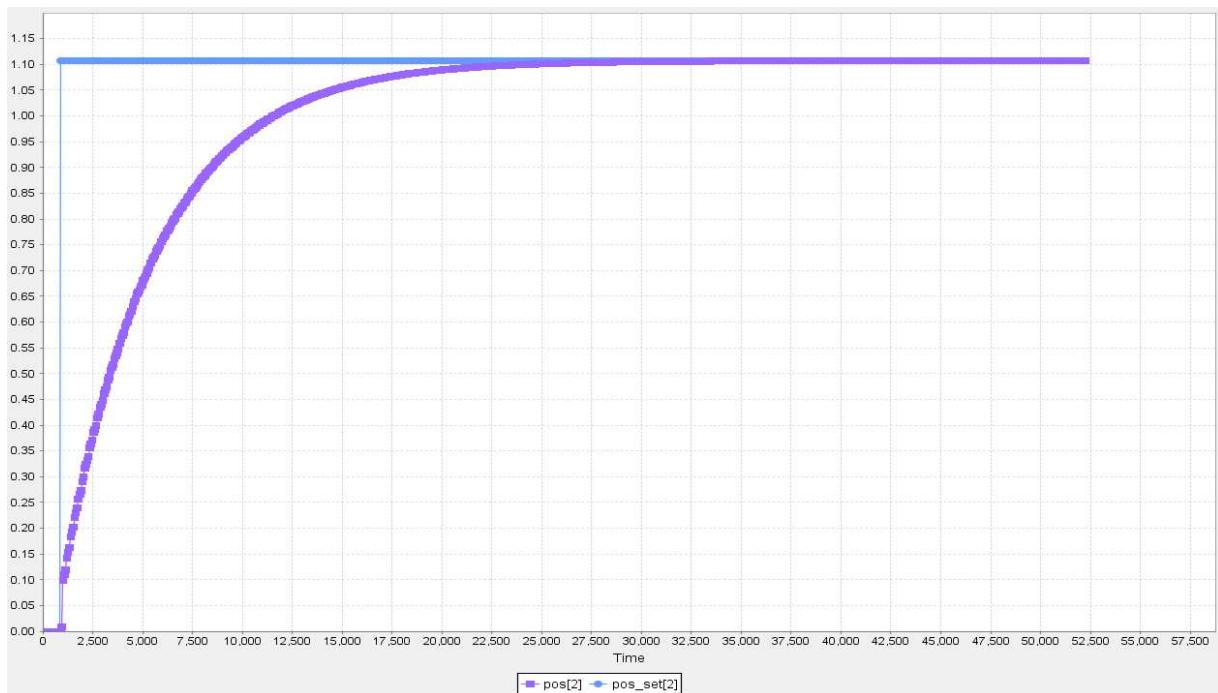
Với  $k_{11} = k_{12} = k_{13} = 8$



Hình 8.7. Đáp ứng theo trục tọa độ y với  $k_{11} = k_{12} = k_{13} = 8$



Hình 8.8. Đáp ứng theo trục tọa độ y với  $k_{11} = k_{12} = k_{13} = 8$



Hình 8.9. Đáp ứng góc với  $k_{11} = k_{12} = k_{13} = 8$

$k_{11} =$ $k_{12} =$ $k_{13}$	x				y				$\theta$			
	Độ vọt lô (%)	Xác lập (m)	Thời gian quá độ (s)	Sai số xác lập (m)	Độ vọt lô (%)	Xác lập (m)	Thời gian quá độ (s)	Sai số xác lập (m)	Độ vọt lô (rad)	Xác lập (rad)	Thời gian quá độ (s)	Sai số xác lập (rad)
3	0	1.017	32.9	0.017	0	1.991	36.6	0.009	0	1.096	36.3	0.011
5	0	1.009	20.3	0.009	0	1.998	23.5	0.002	0	1.107	22.2	0
8	0	1.006	13.3	0.006	0	2.000	15.0	0	0	1.107	14.7	0

Bảng 8.1: Đánh giá chất lượng đáp ứng mô phỏng

Nhận xét:

- Nhờ vào cách chọn tín hiệu điều khiển  $u$ , mặt trượt  $S_1$  và  $S_2$  tiến về 0 theo hàm mũ  $e$ :  $S_1 = e^{-k_1 t}; S_2 = e^{-k_2 t}$   
Vì vậy đáp ứng ngõ ra của hệ thống không có hiện tượng vọt lô. Bù lại, sai số xác lập có tồn tại nhưng rất nhỏ.
- Hệ số K càng lớn, thời gian đáp ứng càng nhanh.
- Hiệu quả của bộ điều khiển DSC đề xuất với sự tồn tại các thành phần bất định. Khi các tham số hệ thống thay đổi, chất lượng bộ điều khiển vẫn được đảm bảo.

## 8.2. Thực nghiệm

Các tham số của hệ thống:

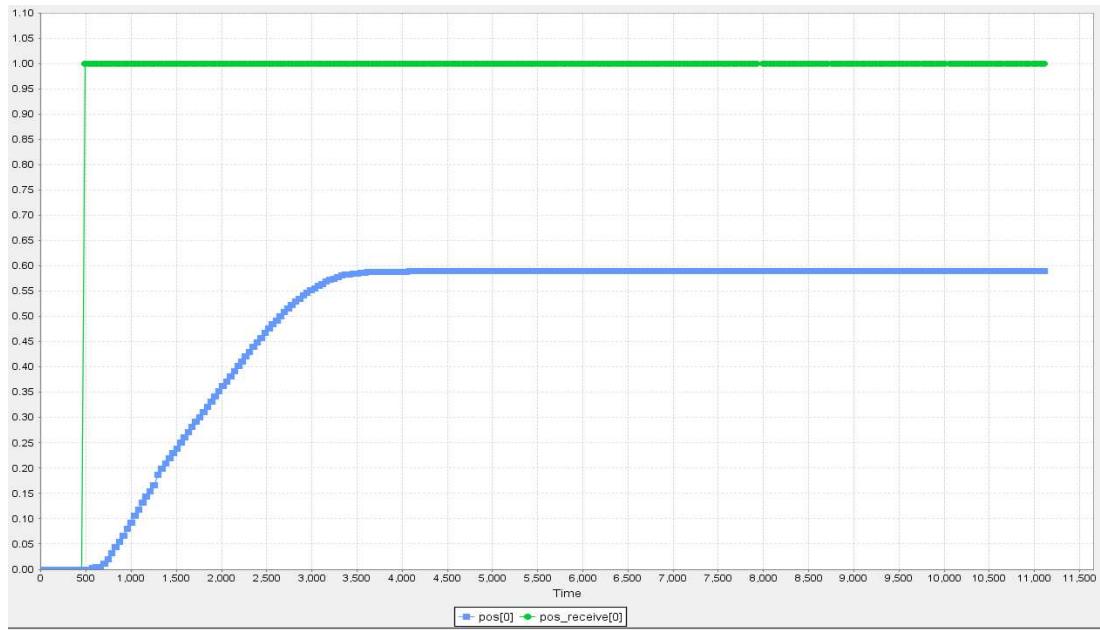
$$m = 1.5 \text{ (kg)};$$

$$D = 0.12 \text{ (m)};$$

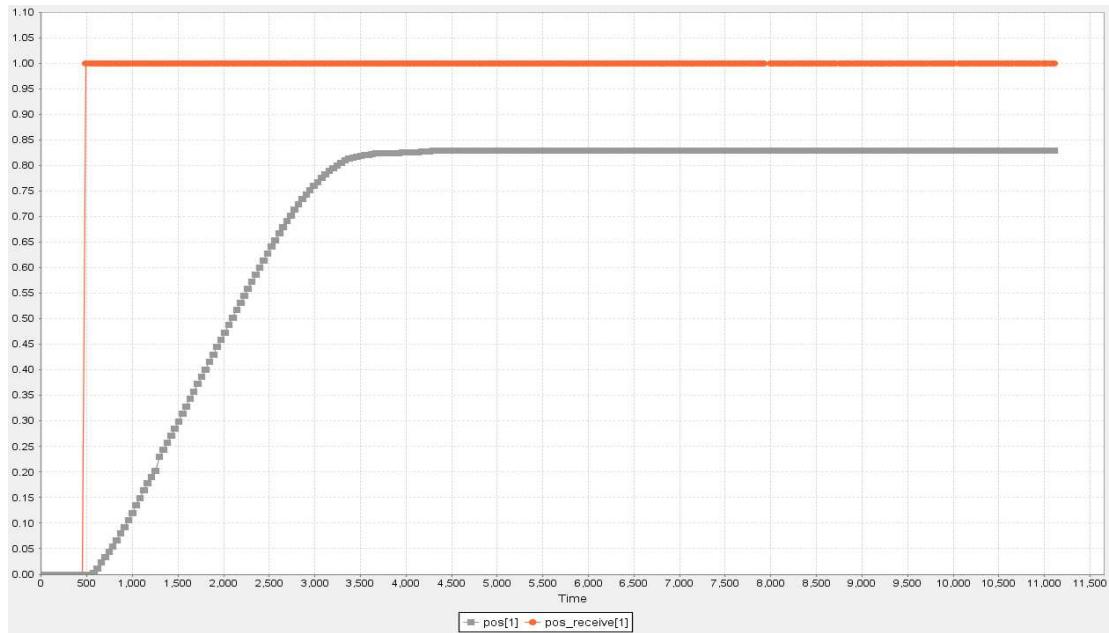
$$J = 0.05 \text{ (N.m)};$$

$$k_{11} = k_{12} = k_{13} = 5;$$

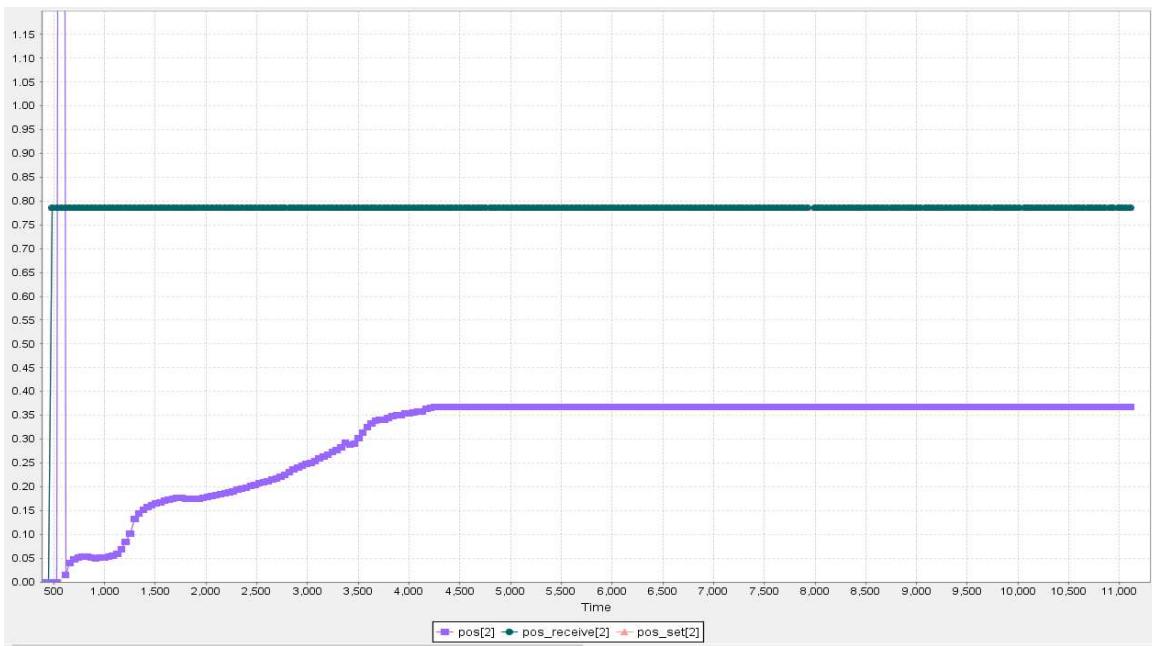
Vị trí đặt là  $\{x, y, \theta\} = \{1, 1, \pi/4\}$



Hình 8.10. Đáp ứng thực tế theo trục tọa độ x với  $k_{11} = k_{12} = k_{13} = 5$



Hình 8.11. Đáp ứng thực tế theo trục tọa độ y với  $k_{11} = k_{12} = k_{13} = 5$



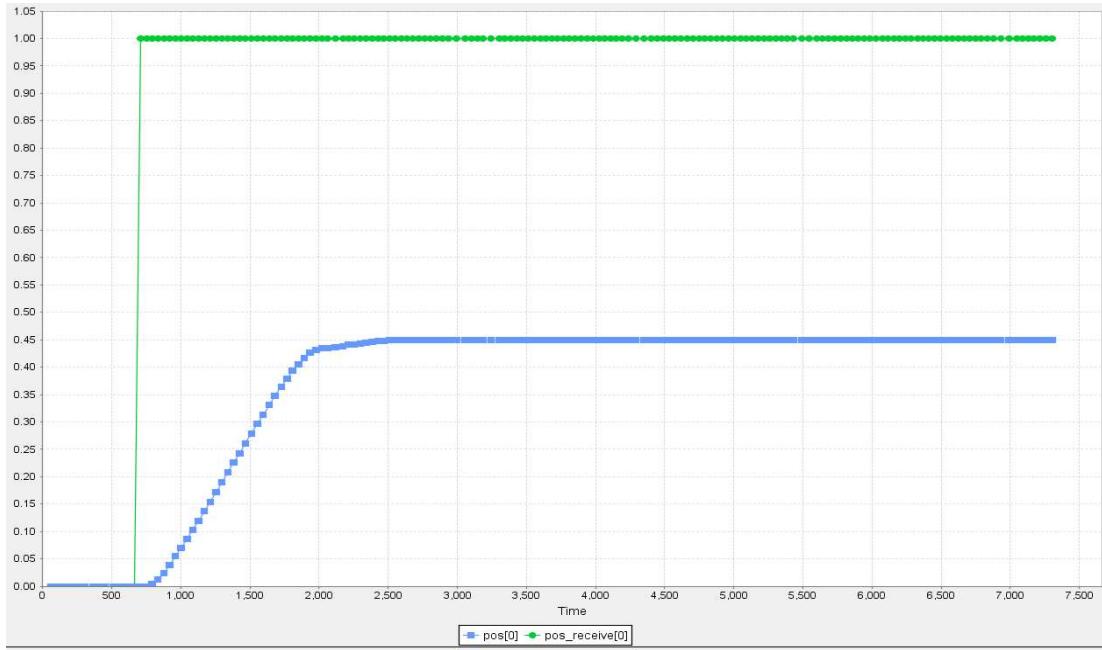
Hình 8.12. Đáp ứng thực tế theo góc với  $k_{11} = k_{12} = k_{13} = 5$

$k_2$	$k_{11} =$ $k_{12} =$ $k_{13}$	x			y			$\theta$		
		Độ vọt lô (%)	Thời gian quá độ (s)	Sai số xác lập (m)	Độ vọt lô (%)	Thời gian quá độ (s)	Sai số xác lập (m)	Độ vọt lô (%)	Thời gian quá độ (s)	Sai số xác lập (rad)
7	5	0	2.01	0.41	0	3.15	0.172	0	3.88	0.4174

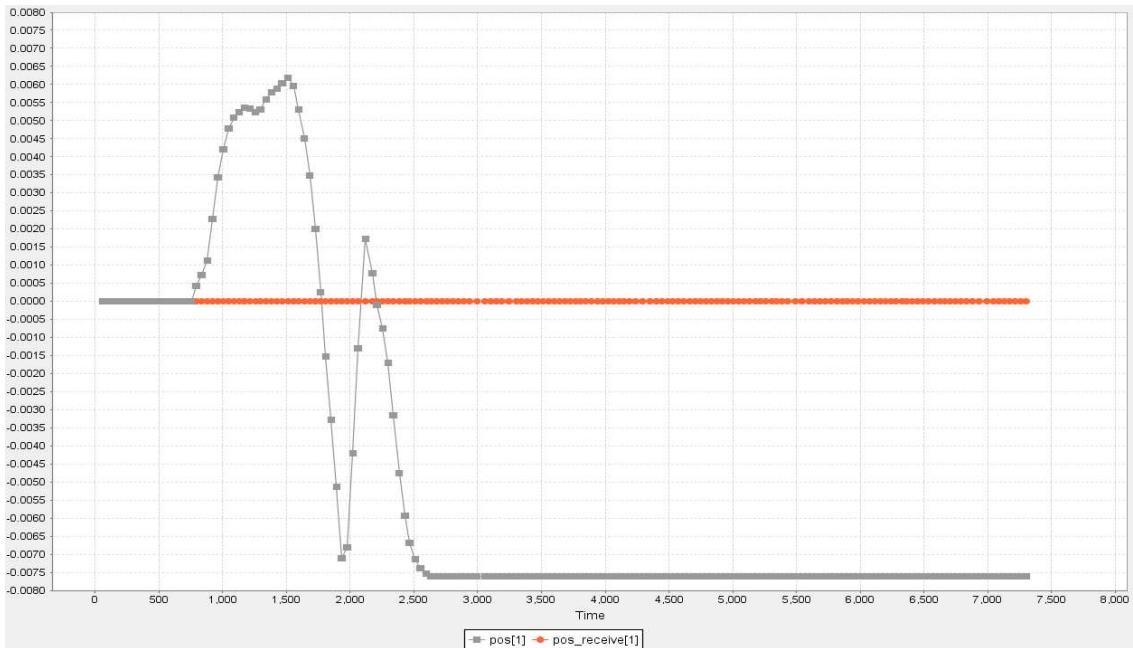
Bảng 8.2: Đánh giá chất lượng đáp ứng thực tế với điểm đặt  $\{1, 1, \pi/4\}$

Đơn giản hơn, ta chỉ đặt setpoint một trục

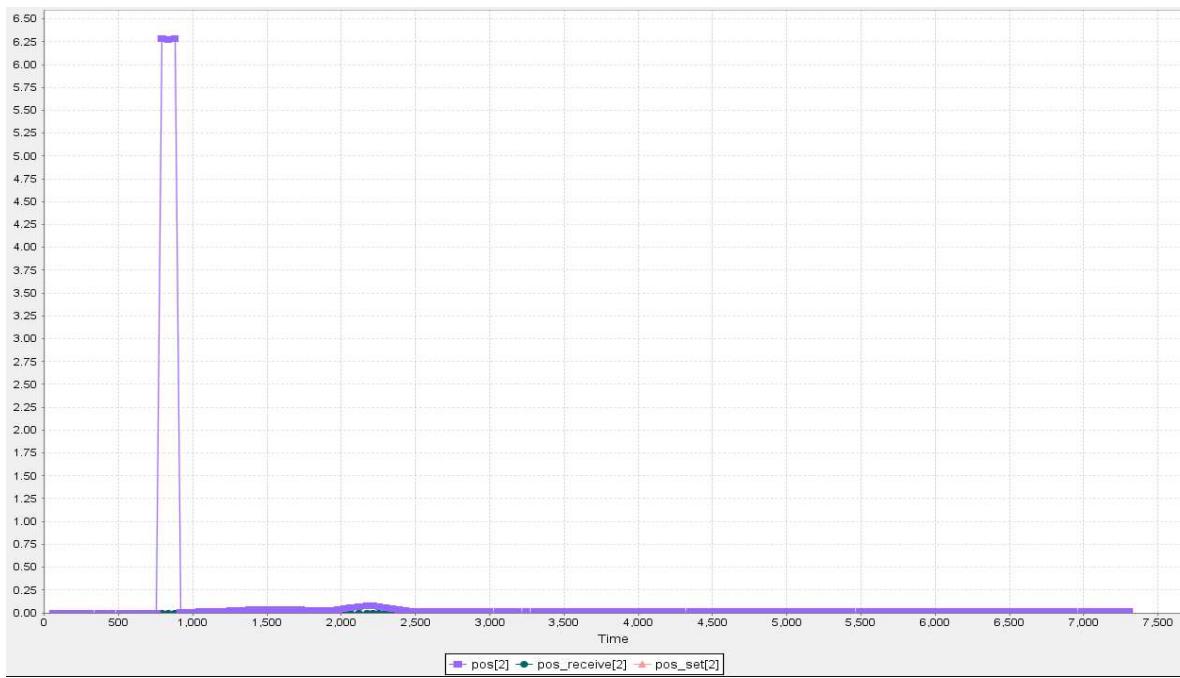
Vị trí đặt là  $\{x, y, \theta\} = \{1, 0, 0\}$



Hình 8.13. Đáp ứng thực tế theo trục tọa độ x với  $k_{11} = k_{12} = k_{13} = 5$

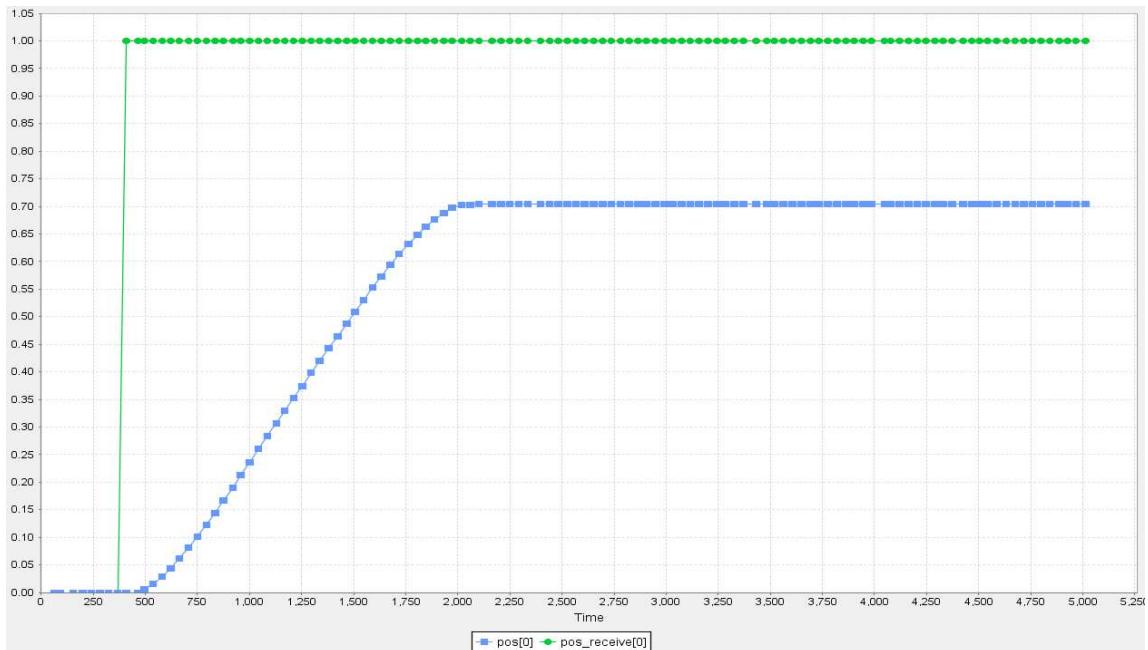


Hình 8.14. Đáp ứng thực tế theo trục tọa độ x với  $k_{11} = k_{12} = k_{13} = 5$

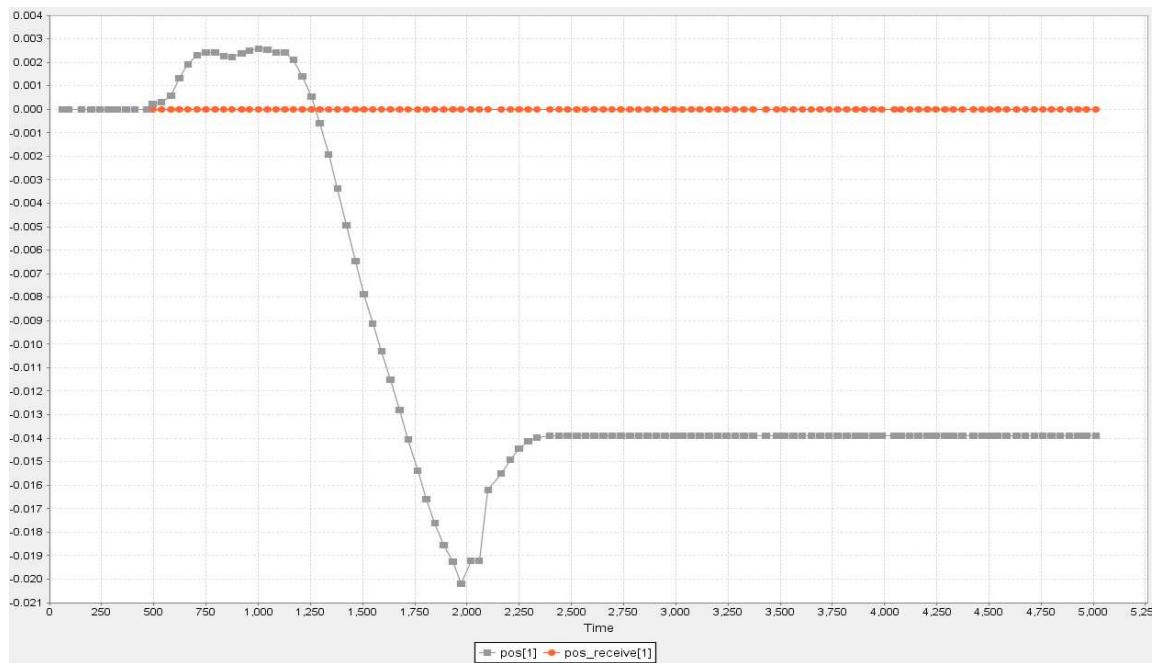


Hình 8.15. Đáp ứng thực tế theo góc với  $k_{11} = k_{12} = k_{13} = 5$

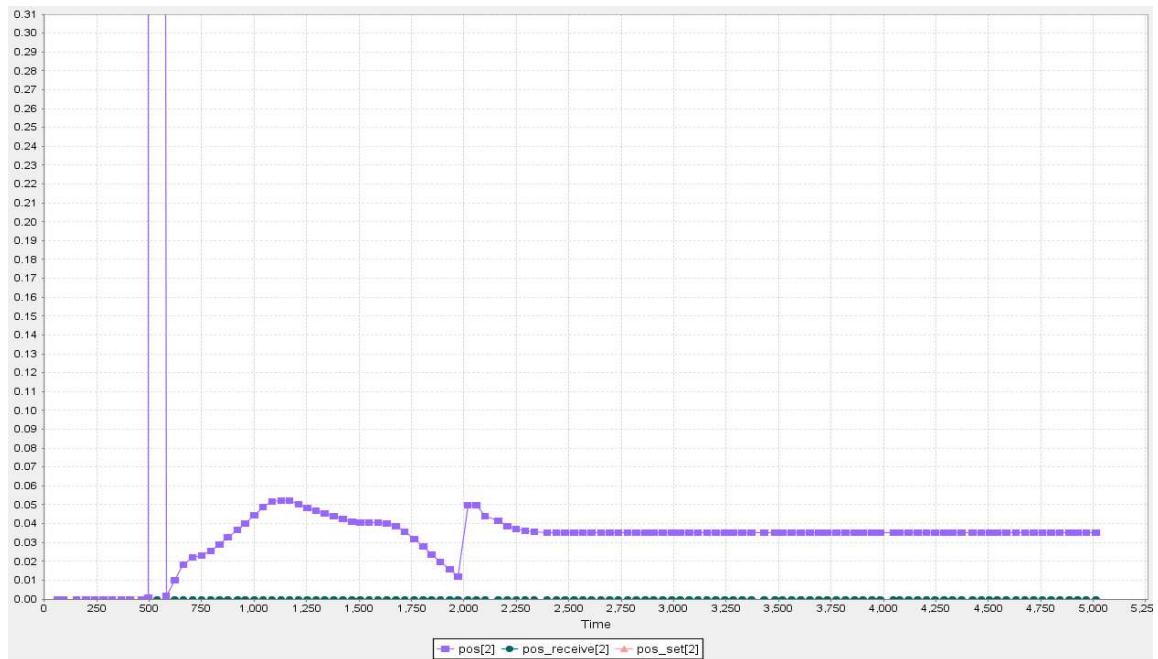
Tăng  $k_{11} = k_{12} = k_{13} = 8$



Hình 8.16. Đáp ứng thực tế theo trục tọa độ x với  $k_{11} = k_{12} = k_{13} = 8$

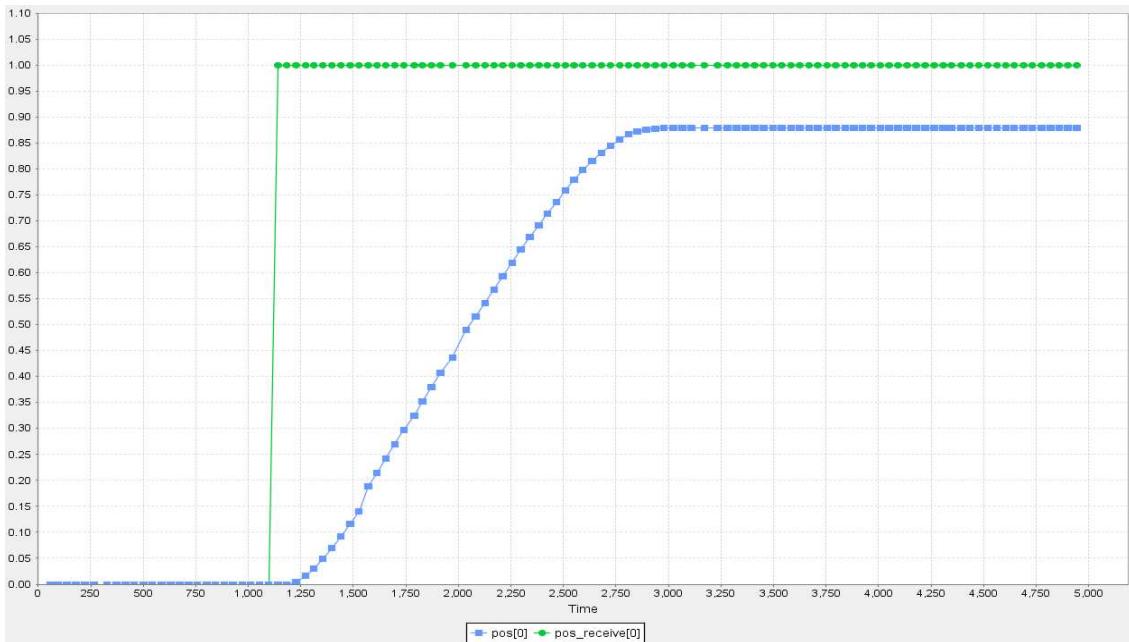


Hình 8.17. Đáp ứng thực tế theo trục tọa độ y với  $k_{11} = k_{12} = k_{13} = 8$

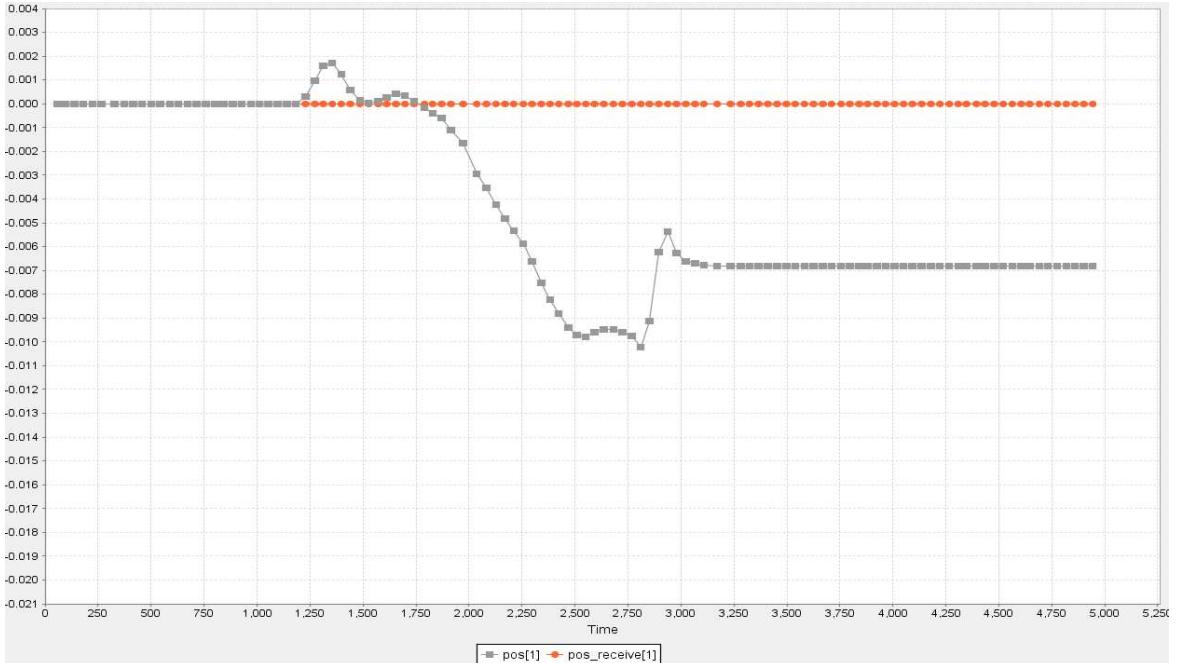


Hình 8.18. Đáp ứng thực tế theo góc với  $k_{11} = k_{12} = k_{13} = 8$

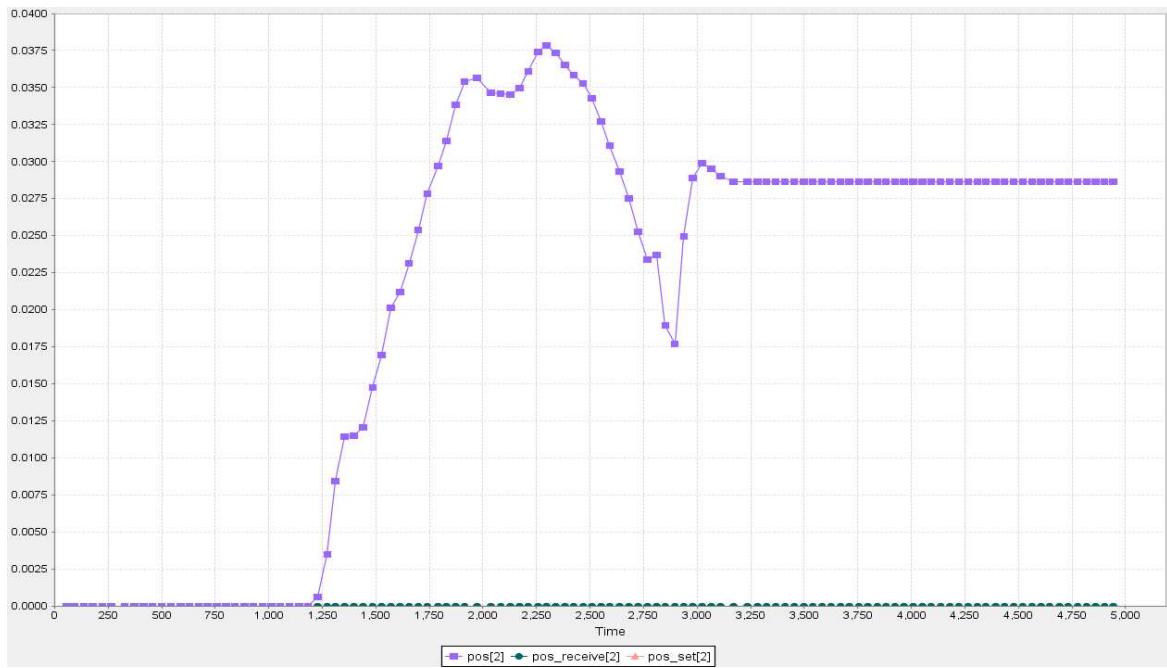
Tiếp tục tăng hệ số  $k_{11} = k_{12} = k_{13} = 12$ ,



Hình 8.19. Dáp ứng thực tế theo trục x với  $k_{11} = k_{12} = k_{13} = 12$



Hình 8.20. Dáp ứng thực tế theo trục y với  $k_{11} = k_{12} = k_{13} = 12$



Hình 8.21. Đáp ứng thực tế theo góc với  $k_{11} = k_{12} = k_{13} = 12$

$k_2$	$k_{11} =$ $k_{12} =$ $k_{13}$	x			y			$\theta$		
		Độ vọt lô (%)	Thời gian quá độ (s)	Sai số xác lập (m)	Độ vọt lô (%)	Thời gian quá độ (s)	Sai số xác lập (m)	Độ vọt lô (%)	Thời gian quá độ (s)	Sai số xác lập (rad)
7	5	0	1.35	0.551	0	1.88	0.008	0	1.91	0.014
7	8	0	1.32	0.295	33	1.82	0.014	17	1.72	0.035
7	12	0	1.37	0.121	41	2.92	0.007	30	3	0.029

Bảng 8.3: Đánh giá chất lượng đáp ứng thực tế với điểm đặt {1, 0, 0}

Nhận xét:

- Vận tốc mỗi bánh thực tế không bám sát vận tốc đặt.
- Sai số xác lập lớn.
- Tăng giá trị  $k_{11}$ ,  $k_{12}$ ,  $k_{13}$  đến ngưỡng nhất định, thời gian xác lập không thể giảm được nữa do giới hạn của phần cứng; có thể làm giảm sai số xác lập nhưng không thể triệt tiêu được hoàn toàn. Ngoài ra giá trị  $k_{11}$ ,  $k_{12}$ ,  $k_{13}$  lớn khi điều khiển một trực nhưng gây sai lệch ở hai thành phần vị trí còn lại. Không thể điều khiển tốt cùng lúc ba thành phần vị trí.

### 8.3. Khó khăn và giải pháp

- Vùng chết khi điều khiển tốc độ động cơ.

Trên thực tế, mỗi động cơ DC khi chạy có tải đều có ngưỡng vận tốc thấp nhất có thể chạy được. Dưới ngưỡng này, động cơ DC không thể chạy được. Khi giá trị vị trí dần tiến đến giá trị đặt, tín hiệu điều khiển mỗi bánh giảm, khi đủ nhỏ, xe không tiếp tục chạy được.

⇒ Dẫn đến sai số xác lập trên trục x và y.

Giải pháp: Không set cố định giá trị đặt ban đầu. Giá trị đặt được tăng dần từ 0 một lượng  $\Delta$ setpoint mỗi chu kỳ, đến khi vị trí của xe còn cách vị trí mong muốn một khoảng cố định, đặt lại giá trị đặt bằng đúng vị trí mong muốn.

```
(init) Setpoint = 0;  
  
if (position expected_position)  
    Setpoint = Setpoint + Δsetpoint;  
  
else  
    Setpoint = expected_position;
```

- Do các động cơ không giống nhau.

Vì vậy giá trị điện áp đưa vào mỗi động cơ đều được tính trung bình. Khi điện áp đưa vào động cơ thấp, có động cơ không quay được, do nằm trong vùng chét, có động cơ mạnh hơn chạy được, làm sai lệch hướng (góc) của xe.

⇒ Dẫn đến sai số xác lập trên trực x, y và góc của xe.

**Giải pháp:** Xác định ngưỡng điện áp nhỏ nhất mà cả 4 bánh xe đều có thể xoay được với tải cho trước. Khi tín hiệu điều khiển dưới ngưỡng, không phát tín hiệu cho cả 4 bánh xe.

if ( $u < \text{limit}$ )

$u = 0;$

- Không xác định được thông số chính xác động cơ.

Việc tính toán điện áp động cơ khi có momen lực và tốc độ chỉ mang tính tương đối. Điều này làm tốc độ mỗi bánh không bám theo tốc độ đặt.

⇒ Dẫn đến sai số xác lập trên trực x, y và góc của xe.

**Giải pháp:** Tăng hoặc giảm tín hiệu điều khiển PWM\_Cycle cho mỗi bánh một lượng  $\Delta \text{PWM\_Cycle}$  tỉ lệ với độ lệch tốc độ tức thời của từng bánh

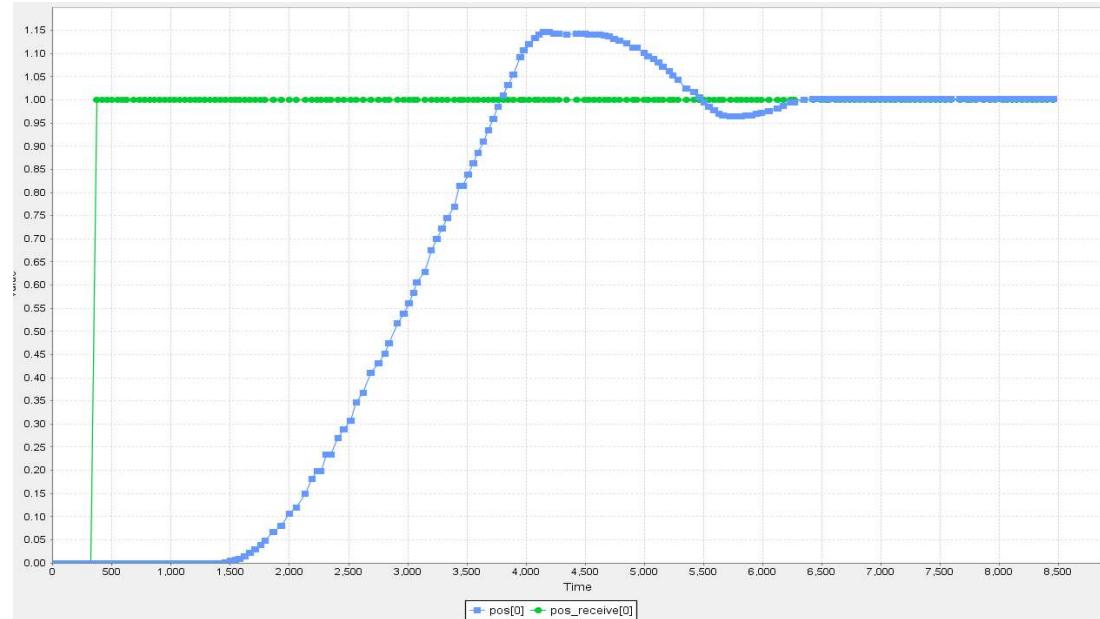
$$\Delta \text{PWM\_Cycle} = k * (v_{\text{set}} - v); // v là vận tốc từng bánh xe$$

$$\text{PWM\_Cycle} = \text{PWM\_Cycle} + \Delta \text{PWM\_Cycle};$$

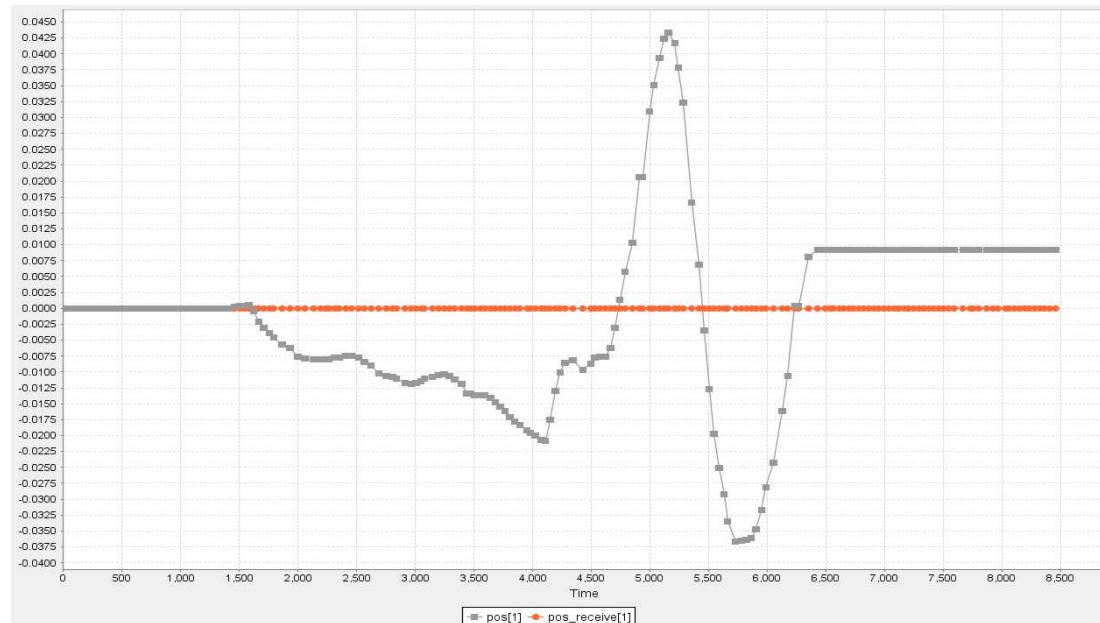
➤ Áp dụng:

Với điểm đặt  $\{x,y,\theta\} = \{1,0,0\}$

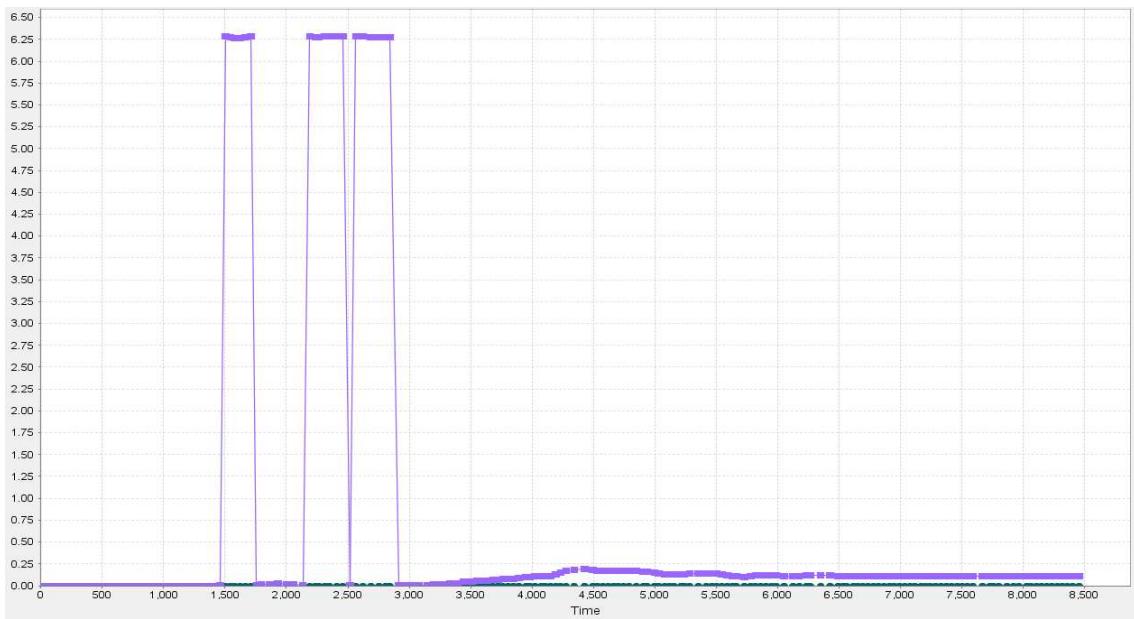
- $k_{11} = k_{12} = k_{13} = 5, k_2 = 3$



Hình 8.22. Đáp ứng thực tế theo trục  $x$  sau khi giải quyết hạn chế



Hình 8.23. Đáp ứng thực tế theo trục  $y$  sau khi giải quyết hạn chế



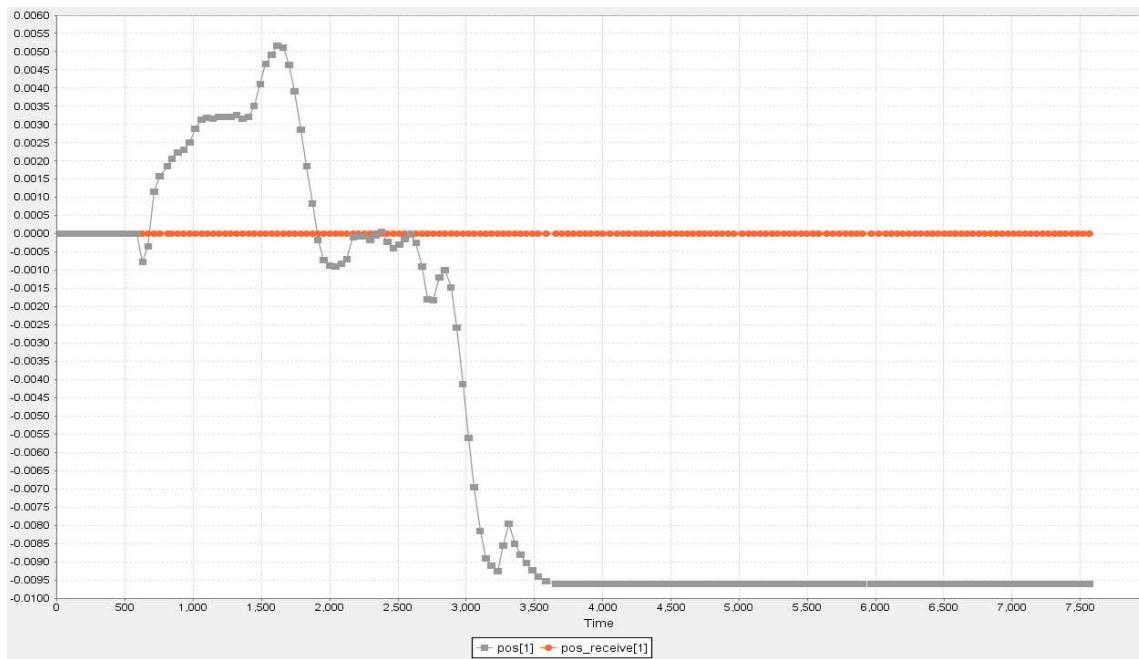
*Hình 8.24. Đáp ứng thực tế theo góc sau khi giải quyết hạn chế*

Thay đổi thông số bộ điều khiển

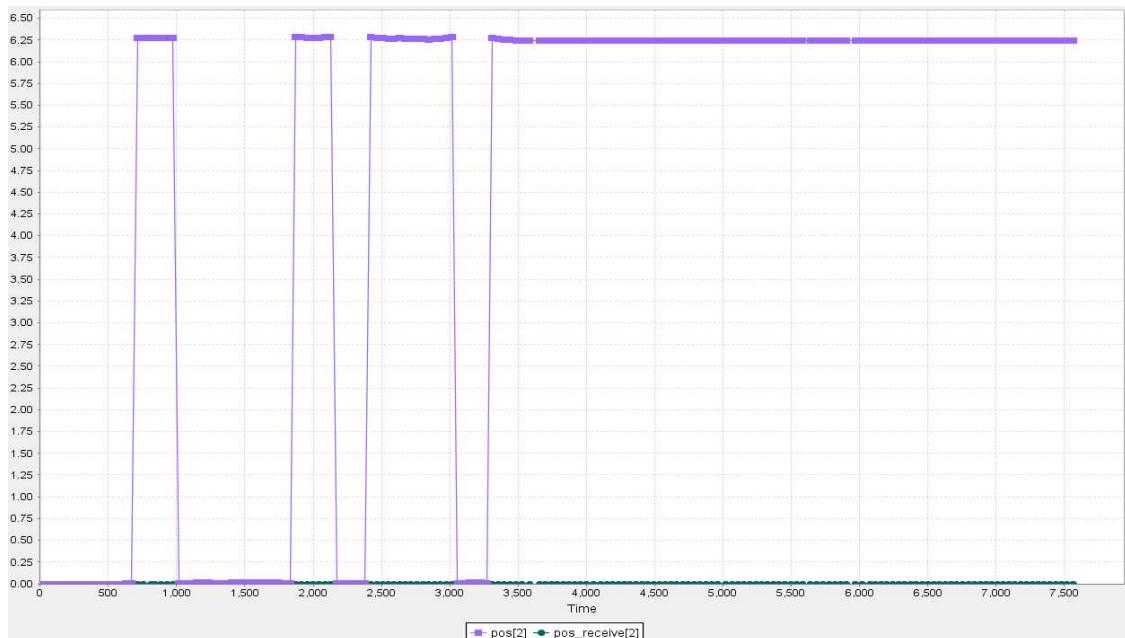
- $k_{11} = k_{12} = k_{13} = 5, k_2 = 7$



*Hình 8.25. Đáp ứng thực tế theo trục x sau khi giải quyết hạn chế*



Hình 8.26. Đáp ứng thực tế theo trục y sau khi giải quyết hạn chế



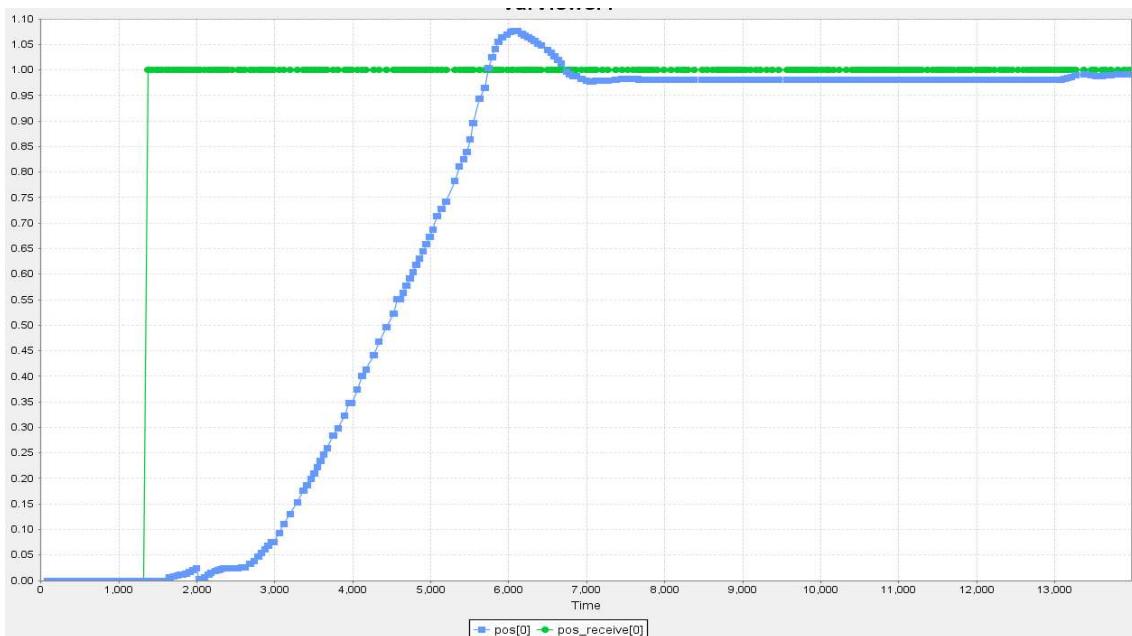
Hình 8.27. Đáp ứng thực tế theo góc sau khi giải quyết hạn chế

$k_2$	$k_{11}$ =	x			y			$\theta$		
		Độ vọt lố (%)	Thời gian quá độ (s)	Sai số xác lập (m)	Độ vọt lố (%)	Thời gian quá độ (s)	Sai số xác lập (m)	Độ vọt lố (%)	Thời gian quá độ (s)	Sai số xác lập (rad)
3	5	14.4	3.721	0.002	367	6.2	0.009	1.5	4.25	0.172
7	5	1.065	3.18	0.016	0	3.5	0.01	0	0	0.037

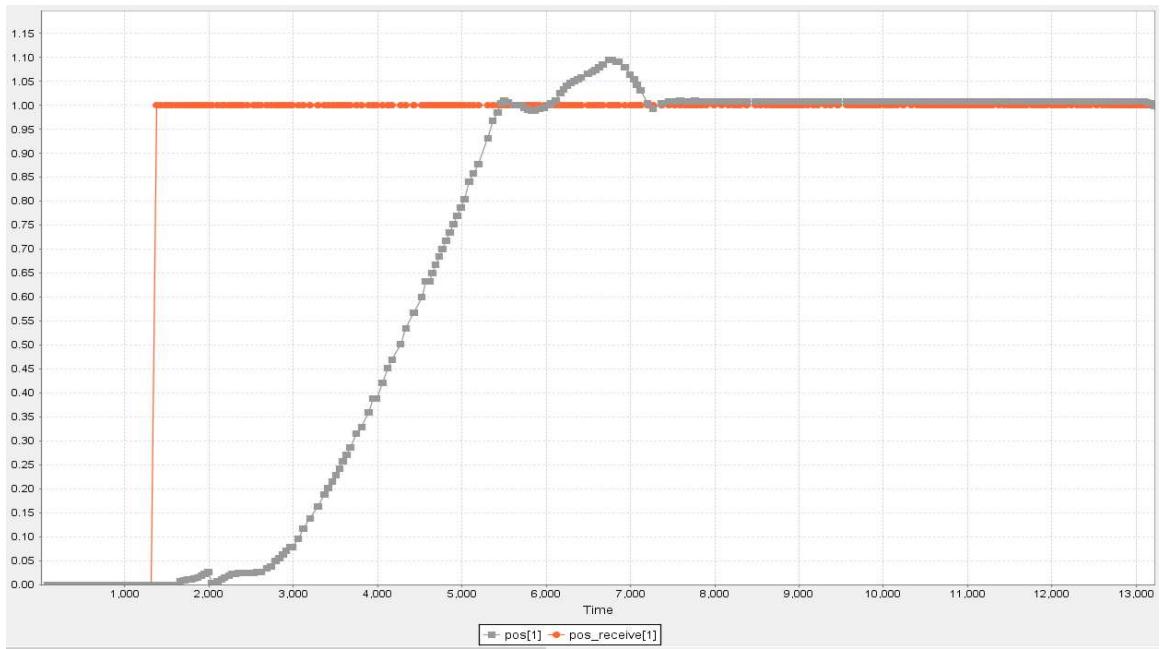
Bảng 8.4: Đánh giá chất lượng đáp ứng sau giải pháp với điểm đặt  $\{1, 0, 0\}$

Với điểm đặt  $\{x, y, \theta\} = \{1, 1, \pi/4\}$

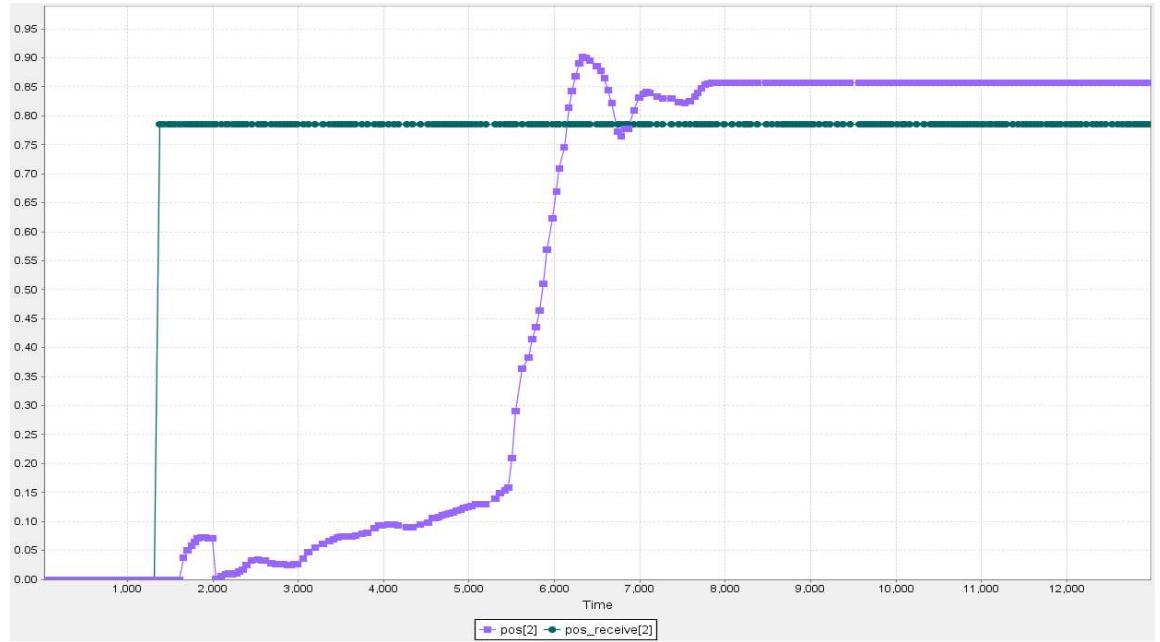
- $k_{11} = k_{12} = k_{13} = 5, k_2 = 3$



Hình 8.28. Đáp ứng thực tế theo trục x sau khi giải quyết hạn chế



Hình 8.29. Đáp ứng thực tế theo trục x sau khi giải quyết hạn chế



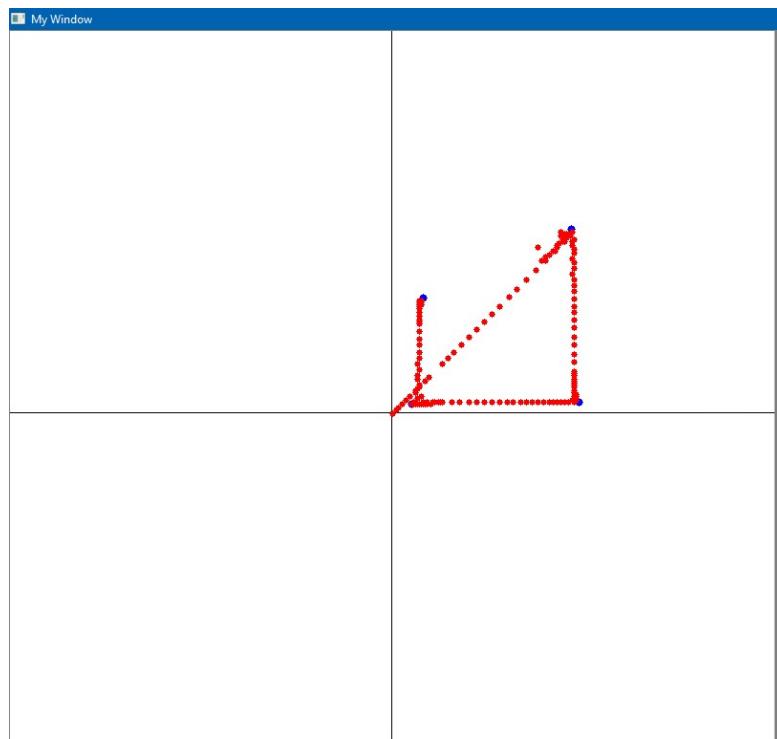
Hình 8.30. Đáp ứng thực tế theo trục x sau khi giải quyết hạn chế

$k_2$	$k_{11} =$ $k_{12} =$ $k_{13}$	x			y			$\theta$		
		Xác lập (m)	Thời gian quá độ (s)	Sai số xác lập (m)	Xác lập (m)	Thời gian quá độ (s)	Sai số xác lập (m)	Xác lập (rad)	Thời gian quá độ (s)	Sai số xác lập (rad)
3	5	1.146	3.721	1.002	0.043	6.2	0.009	0.187	4.25	0.172

Bảng 8.4: Đánh giá chất lượng đáp ứng sau giải pháp với điểm đặt  $\{1, 1, \pi/4\}$

Nhận xét:

- Sai số xác lập nằm trong ngưỡng cho phép, có vọt lố trong giới hạn.
- Quỹ đạo được vẽ lại



Hình 8.30. Vẽ lại quỹ đạo di chuyển

#### 8.4. Đánh giá chung

- **Ưu điểm:**

Bộ điều khiển thể hiện hiệu quả bám tốt, trong điều kiện lý tưởng. Các kết quả mô phỏng cũng cho thấy robot đã đạt được mục tiêu đặt ra, thời gian quá độ ở mức cho phép. Các kết quả chạy thử nghiệm bước đầu cho thấy áp dụng thuật toán DSC để điều khiển robot là hướng đúng đắn, khả thi trong thực tế và có triển vọng cho nhiều ứng dụng khác.

Phương pháp định vị Dead-reckoning thể hiện độ linh động, đơn giản, ít tốn kém, độ chính xác ổn, phù hợp trong những điều kiện nhất định.

- **Hạn chế:**

Để điều khiển tốt, cần lựa chọn đúng thiết bị vận hành hợp lý dựa theo yêu cầu bài toán, nắm rõ những thông số của thiết bị vận hành, độ chính xác và tốc độ đáp ứng phụ thuộc vào phần cứng.

Điều kiện để sử dụng định vị Dead-reckoning còn lý tưởng, tiềm ẩn nhiều khả năng sai số:

Sai số hệ thống: các bánh xe không tròn, các đường kính trung bình của các bánh xe không bằng với đường kính bánh xe mà chúng ta dùng để tính toán, các bánh xe bị lệch trục, độ phân giải giới hạn của encoder...

Sai số ngẫu nhiên: mặt sàn không phẳng, có những chướng ngại vật bất thường, các cảm biến khác không ghi nhận được, và robot vẫn vượt qua chướng ngại vật đó, sự trượt của bánh xe với mặt sàn, tốc độ quá cao và bánh xe chưa đạt đến tốc độ trong một khoảng thời gian tính toán, bánh xe không tiếp xúc với mặt sàn...

Trong thực tế, khi robot hoạt động trên một mặt sàn tương đối tốt, phẳng, thì sai số chủ yếu là các sai số hệ thống. Nhưng nếu mặt sàn lồi lõm thì các sai số ngẫu nhiên lại là vấn đề cần được quan tâm.

## CHƯƠNG 9. ỨNG DỤNG VÀ HƯỚNG PHÁT TRIỂN

### 9.1. Ứng dụng

- Nâng cao tính tự động trong công nghiệp sản xuất và chế biến: robot di chuyển trong nhà máy, xưởng,...
- Thay thế con người hoạt động ở những nơi nguy hiểm hoặc khó khăn.
- Thể hiện tính linh động cao khi robot có thể tự chạy bởi lập trình trước hoặc điều khiển qua vô tuyến.
- Phương pháp định vị dead-reckoning thể hiện được tính đơn giản và tiết kiệm. Ngoài ra tránh được những rủi ro do kết nối không ổn định hoặc bị đứt, robot vẫn có thể định hướng để tiếp tục công việc hoặc trở về vị trí ban đầu.

### 9.2. Hướng phát triển

- Phương pháp định vị bằng dead-reckoning là phương pháp định vị không cột mốc, vậy có thể kết hợp với phương pháp khác để tăng tính chính xác và sửa được sai số tích luỹ: xử lý ảnh, đo cường độ sóng, thời gian truyền sóng, wifi, RF, GPS, ...
- Dùng thuật toán A\* để robot tự tìm đường với input là bản đồ khu vực, giúp nó có thể di chuyển trong nhà, xưởng, từ phòng này qua phòng khác, ...
- Kết hợp với cảm biến tiệm cận để tránh vật cản hoặc vẽ lại đường đi hoặc bản đồ.

## TÀI LIỆU THAM KHẢO

- [1] “Encoder Resolution, Encoder Accuracy and System Repeatability,” Dynapar Company,[Trực tuyến]. Available: [https://www.dynapar.com/knowledge/encoder\\_resolution\\_encoder\\_accuracy\\_repeatability/](https://www.dynapar.com/knowledge/encoder_resolution_encoder_accuracy_repeatability/).
- [2] T. Baede, Motion control of an omnidirectional mobile robot, Eindhoven, 2006.
- [3] G. T. P. T. Cát, “Xu thế phát triển Robot trên thế giới và tình hình nghiên cứu Robot ở Việt Nam hiện nay,” *Tạp chí tự động hóa ngày nay*, 2011.
- [4] K. S. Đường và D. C. K. Từ, “Ứng dụng bộ điều khiển self tuning Fuzzy-PI điều khiển omni-directional mobile robot,” 2012.
- [5] R. R. a. A. G. Forster, “Holonomic Control of a robot with an omnidirectional,” 2006.
- [6] Y. L. D. L. a. C. E. L. T. L.Huang, "Design and Analysis of a Four-wheel Omnidirectional Mobile Robot," 2004.
- [7] M. T. Ngô, “Điều khiển bám quỹ đạo cho Omini Robot bốn bánh sử dụng thuật toán DSC,” *Hội nghị toàn quốc lần thứ 3 về Điều khiển và Tự động hóa - VCCA*, 2015.
- [8] Q. H. Nguyễn và D. T. Trần, “Thiết kế bộ điều khiển cho xe tự hành sử dụng thuật toán DSC,” DHBK Hà Nội.
- [9] R. Rojas, Omnidirectional Control, Berlin, 2005.
- [10] T. Youngblood, “How to Use a Rotary Encoder in an MCU-Based Project,” 2015.