

BỘ GIÁO DỤC
VÀ ĐÀO TẠO

VIỆN HÀN LÂM KHOA HỌC
VÀ CÔNG NGHỆ VIỆT NAM

HỌC VIỆN KHOA HỌC VÀ CÔNG NGHỆ



Nguyễn Minh Đông

**NGHIÊN CỨU XÂY DỰNG THUẬT TOÁN THÍCH NGHI
VÀ HỌC TĂNG CƯỜNG CẤU TRÚC ACTOR-CRITIC
ĐIỀU KHIỂN BÁM QUỲ ĐẠO CHO ROBOT DI ĐỘNG
ĐA HƯỚNG MECANUM**

**LUẬN ÁN TIẾN SĨ
KỸ THUẬT ĐIỀU KHIỂN VÀ TỰ ĐỘNG HÓA**

Hà Nội - 2025

BỘ GIÁO DỤC
VÀ ĐÀO TẠO

VIỆN HÀN LÂM KHOA HỌC
VÀ CÔNG NGHỆ VIỆT NAM

HỌC VIỆN KHOA HỌC VÀ CÔNG NGHỆ

Nguyễn Minh Đông

NGHIÊN CỨU XÂY DỰNG THUẬT TOÁN THÍCH NGHI
VÀ HỌC TĂNG CƯỜNG CẤU TRÚC ACTOR- CRITIC
ĐIỀU KHIỂN BÁM QUỠ ĐẠO CHO ROBOT DI ĐỘNG
ĐA HƯỚNG MECANUM

LUẬN ÁN TIẾN SĨ KỸ THUẬT ĐIỀU KHIỂN VÀ TỰ ĐỘNG HÓA

Mã số: 9 52 02 16

Xác nhận của Học viện

Khoa học và Công nghệ



Nguyễn Thị Trung

Người hướng dẫn 1

(Ký, ghi rõ họ tên)

TS. Ngô Mạnh Tiến

Người hướng dẫn 2

(Ký, ghi rõ họ tên)

PGS. TS Đào Phương Nam

Hà Nội - 2025

LỜI CAM ĐOAN

Tôi xin cam đoan luận án: ***“Nghiên cứu xây dựng thuật toán thích nghi và học tăng cường cấu trúc Actor-Critic điều khiển bám quỹ đạo cho robot di động đa hướng mecanum”*** là công trình nghiên cứu của chính tôi dưới sự hướng dẫn khoa học của tập thể hướng dẫn ***TS. Ngô Mạnh Tiến*** và ***PGS.TS. Đào Phương Nam***. Luận án sử dụng thông tin trích dẫn từ nhiều nguồn tham khảo khác nhau và các thông tin trích dẫn được ghi rõ nguồn gốc. Các kết quả nghiên cứu của tôi được công bố chung với các tác giả khác đã được sự nhất trí của đồng tác giả khi đưa vào luận án. Các số liệu, kết quả được trình bày trong luận án là hoàn toàn trung thực và chưa từng được công bố trong bất kỳ một công trình nào khác ngoài các công trình công bố của tác giả. Luận án được hoàn thành trong thời gian tôi làm nghiên cứu sinh tại Học viện Khoa học và Công nghệ, Viện Hàn lâm Khoa học và Công nghệ Việt Nam.

Hà nội, ngày 10 tháng 06 năm 2025

Tác giả luận án



Nguyễn Minh Đông

LỜI CẢM ƠN

Luận án này được hoàn thành với sự nỗ lực không ngừng của tác giả và sự giúp đỡ hết mình từ tập thể hướng dẫn, bạn bè và người thân.

Đầu tiên, tác giả xin bày tỏ lời tri ân tới **TS. Ngô Mạnh Tiến** và **PGS.TS. Đào Phương Nam** đã tận tình hướng dẫn khoa học để tác giả có thể hoàn thành luận án này.

Tác giả xin gửi lời cảm ơn tới các Thầy giáo, Cô giáo và Cán bộ chuyên môn, Ban Lãnh đạo, phòng Đào tạo, các phòng chức năng của Học viện Khoa học và Công nghệ đã nhiệt tình giúp đỡ và tạo ra môi trường học tập, nghiên cứu tốt để tác giả hoàn thành công trình của mình.

Cảm ơn các Thầy, Cô và các đồng nghiệp ở các nơi mà tác giả tham gia viết bài đã có những góp ý chính xác để tác giả có được những công bố như ngày hôm nay.

Tác giả xin cảm ơn tới Ban giám hiệu trường Đại học Kinh tế- Kỹ thuật Công nghiệp, các đồng nghiệp khoa Điện- Tự động hóa nơi tác giả công tác đã ủng hộ để luận án được hoàn thành đúng thời hạn.

Cuối cùng, tác giả xin gửi tới gia đình, bạn bè, người thân lời cảm ơn chân thành nhất vì đã đồng hành cùng tác giả trong suốt thời gian qua.

Hà Nội, ngày 10 tháng 06 năm 2025

Tác giả luận án



Nguyễn Minh Đông

MỤC LỤC

LỜI CAM ĐOAN	i
LỜI CẢM ƠN	ii
MỤC LỤC.....	iii
DANH MỤC CÁC KÝ HIỆU VÀ CHỮ VIẾT TẮT	v
BẢNG KÝ HIỆU CỦA CÁC THAM SỐ	vii
DANH MỤC CÁC BẢNG BIỂU	ix
DANH MỤC CÁC HÌNH VẼ.....	x
MỞ ĐẦU.....	1
CHƯƠNG 1. TỔNG QUAN VỀ ROBOT DI ĐỘNG ĐA HƯỚNG	6
1.1. Tổng quan về robot di động	6
1.2. Robot di động đa hướng mecanum	8
1.3. Mô hình toán học robot di động đa hướng mecanum	11
1.3.1. Cấu tạo bánh xe mecanum	11
1.3.2. Phương trình động học robot di động bốn bánh xe mecanum	13
1.3.3. Phương trình động lực học robot di động bốn bánh xe mecanum.....	15
1.4. Cấu trúc hệ thống điều khiển robot di động.....	18
1.5. Tổng quan tình hình nghiên cứu trong và ngoài nước	20
1.5.1. Tình hình nghiên cứu trong nước.....	20
1.5.2. Tình hình nghiên cứu ngoài nước	22
1.6. Kết luận chương 1	24
CHƯƠNG 2. THUẬT TOÁN ĐIỀU KHIỂN BÁM QUỸ ĐẠO CHO ROBOT DI ĐỘNG ĐA HƯỚNG MECANUM.....	25
2.1. Bộ điều khiển PID điều khiển bám quỹ đạo cho FMWR	25
2.2. Bộ điều khiển trượt cơ bản cho FMWR.....	26
2.3. Thuật toán điều khiển Backstepping - SMC	28
2.3.1. Thiết kế thuật toán.....	28
2.3.2. Mô phỏng thuật toán	31
2.4. Thuật toán điều khiển mặt trượt động cho FMWR.....	33
2.5. Mô phỏng thuật toán PID, SMC và DSC trên Matlab/Simulink	35
2.6. Đề xuất thuật toán điều khiển mặt trượt động chỉnh định thích nghi mờ cho robot di động đa hướng mecanum (Fuzzy-DSC).....	41
2.6.1. Thiết kế thuật toán Fuzzy-DSC.....	41

2.6.2. Mô phỏng và đánh giá kết quả	43
2.7. Kết luận chương 2	56
CHƯƠNG 3. THUẬT TOÁN HỌC TĂNG CƯỜNG ĐIỀU KHIỂN BÁM QUỠ ĐẠO CHO ROBOT DI ĐỘNG ĐA HƯỚNG MECANUM.....	58
3.1. Cơ sở phương pháp luận nghiên cứu	58
3.1.1. Mạng nơ-ron xấp xỉ hàm.....	58
3.1.2. Ổn định bị chặn tới hạn đều (UUB- Uniformly Ultimate Bounded)	60
3.2. Thuật toán quy hoạch động (ADP- Adaptive Dynamic Programming)	61
3.3. Thiết kế thuật toán học tăng cường điều khiển bám quỹ đạo cho FMWR	70
3.3.1. Biến đổi mô hình động lực học FMWR.....	70
3.3.2. Bộ điều khiển tối ưu và phương trình HJB	71
3.3.3. Thuật toán học tăng cường cấu trúc Actor- Critic	73
3.3.4. Xét tính ổn định cho hệ thống	77
3.3.5. Mô phỏng thuật toán	79
3.4. Kết luận chương 3	90
CHƯƠNG 4. THỰC NGHIỆM VỚI MÔ HÌNH ROBOT DI ĐỘNG ĐA HƯỚNG MECANUM	92
4.1. Thiết kế cơ khí cho FMWR.....	92
4.2. Thiết kế mạch điều khiển cho FMWR	93
4.3. Hệ điều hành ROS.....	95
4.4. Chạy thử nghiệm mô hình trên Gazebo	96
4.5. Thực nghiệm mô hình robot.....	98
4.6. Kết luận chương 4	106
KẾT LUẬN VÀ KIẾN NGHỊ.....	108
DANH MỤC CÔNG TRÌNH KHOA HỌC ĐÃ CÔNG BỐ CỦA TÁC GIẢ	110
TÀI LIỆU THAM KHẢO.....	111
PHỤ LỤC.....	1

DANH MỤC CÁC KÝ HIỆU VÀ CHỮ VIẾT TẮT

Chữ viết tắt	Tiếng Anh	Tiếng việt
AC	<i>Actor- Critic</i>	Cấu trúc Actor – Critic
AC-NNs	<i>Neural Networks Actor Critic</i>	Cấu trúc điều khiển mạng nơ-ron Actor Critic
Actor NN	<i>Actor Neural Network</i>	Mạng nơ-ron Actor
ADP	<i>Adaptive-Dynamic-Programming</i>	Thuật toán thích nghi quy hoạch động
AGV	<i>Automation Guided Vehicle</i>	Robot di động
Critic NN	<i>Critic Neural Network</i>	Mạng nơ-ron Critic
DSC	<i>Dynamic Surface Control</i>	Điều khiển mặt trượt động
FMWR	<i>Four Mecanum Wheeled Robot</i>	Robot di động đa hướng mecanum
FMWR-ME	<i>Four Mecanum Wheel Robot - Mass Eccentricity</i>	Robot di động đa hướng mecanum lệch trọng tâm
Fuzzy-DSC	<i>Adaptive Fuzzy Dynamic Surface Control</i>	Điều khiển mặt trượt động chỉnh định thích nghi mờ
HJB	<i>Hamilton-Jacobi-Bellman</i>	Phương trình Hamilton Jacobi Bellman
HJI	<i>Hamilton Jacobi Isaacs</i>	Phương trình vi phân và đạo hàm riêng
MSSC	<i>Multi Surface Sliding Control</i>	Thuật toán điều khiển đa mặt trượt
NCS		Nghiên cứu sinh
NN	<i>Neural Network</i>	Mạng nơ-ron nhân tạo
NNs	<i>Neural Network</i>	Mạng nơ-ron
Off-Policy IRL	<i>Integral Reinforcement Learning Off-Policy</i>	Luật điều khiển ngoại tuyến với thuật toán học tăng cường
PE	<i>Persistence of Excitation</i>	Điều kiện PE
PI	<i>Policy- Iteration</i>	Thuật toán vòng lặp để xấp xỉ nghiệm tối ưu cho luật điều khiển
PID	<i>Proportional Integral Derivative</i>	Bộ điều khiển vi tích phân tỉ lệ
RISE	<i>Robust Integral of the Sign of the Error</i>	Tích phân bền vững của sai lệch

RL	<i>Reinforcement Learning</i>	Học tăng cường
RL-AC	<i>RL Actor-Critic</i>	Học tăng cường cấu trúc Actor – Critic
ROS	<i>Robot-Operating-System</i>	Hệ điều hành điều khiển robot
SMC	<i>Sliding Mode Control</i>	Thuật toán điều khiển trượt
UB	<i>Ultimate-Bounded</i>	Giới hạn cuối cùng
UUB	<i>Uniformly-Ultimate-Bounded</i>	Giới hạn chặn tới hạn đều

BẢNG KÝ HIỆU CỦA CÁC THAM SỐ

Ký hiệu	Ý nghĩa	Đơn vị
$.x, y, \phi.$	Biểu diễn vị trí trong hệ tọa độ toàn cục	m, rad
$\dot{x}$	Vận tốc của robot theo trục x	m/s
$\dot{y}$	Vận tốc của robot theo trục y	m/s
$\dot{\phi}$	Vận tốc của góc hướng của robot	rad/s
ψ_i	Góc quay của bánh xe thứ i	rad/s
α	Là góc của vector OA so với trục X_R	rad
β	là góc giữa vector OA so với trục chính của bánh xe	rad
l	Khoảng cách từ trọng tâm robot đến bánh xe	m
r	Bán kính bánh xe chính	m
γ	là góc giữa con lăn và trục bánh xe	rad
$\dot{\psi}$	tốc độ quay của bánh xe chính	rad/s
$\dot{\psi}_{sw}$	tốc độ quay của con lăn thụ động	rad/s
J	Ma trận Jacobi	
P	Vị trí tâm hình học của robot	
P'	Vị trí trọng tâm thực tế của robot	
d_1, d_2	Độ lệch trọng tâm	m
m_b	Khối lượng của robot	kg
m_{wi}	Khối lượng của bánh xe thứ i	kg
J_b	Momen quán tính của robot	N/m
J_w	Momen quán tính của bánh xe thứ i so với trục bánh xe	N/m
f_i	là ma sát của bánh xe thứ i với mặt sàn	N
M	Ma trận mô men quán tính và khối lượng	
C	Ma trận giảm chấn và lực ma sát	
D	Ma trận trọng lực	
H	Ma trận chuyển hệ trục tọa độ	

$\mathbf{Q}_i$	Là lực, mô men thứ i	
$\boldsymbol{\eta}$	Vị trí tổng quát của robot so với hệ trục tọa độ	
$\dot{\boldsymbol{\eta}}$	Vận tốc robot	m/s
$\ddot{\boldsymbol{\eta}}$	Gia tốc tổng quát của robot	m / s^2
$\mathbf{e}$	Sai lệch quỹ đạo	m
$\boldsymbol{\tau}_i$	Tín hiệu lực điều khiển tác dụng vào robot	N/m
$\mathbb{R}$	Tập hợp của các số thực	
$\mathbb{R}^n$	Không gian Euclide n chiều.	
$\mathbb{R}^{n \times m}$	Tập chứa các phần tử số thực, ma trận kích thước $n \times m$	
Ω_x	Tập $\Omega_x \subset \mathbb{R}^n$	
$\Psi(\Omega)$	Tập luật điều khiển chấp nhận được	
$\mathbf{W}_a$	Ma trận trọng số của mạng Actor-NNs	
$\mathbf{W}_c$	Ma trận trọng số của mạng Critic- NNs	
$\phi(\mathbf{x})$	Hàm kích hoạt của mạng NN	
$\boldsymbol{\Gamma}$	Ma trận hệ số ước lượng đối xứng	
$J(\mathbf{x}, \mathbf{u})$	Biểu diễn hàm chi phí	
$\mathbf{V}^*(\mathbf{x})$	Biểu diễn hàm Bellman	
ϵ_{Gaus}	Nhiều ngoại lực tác động	N/m

DANH MỤC CÁC BẢNG BIỂU

Bảng 1-1. Thông số của các bánh xe mecanum	14
Bảng 2-1 : Xác định ngôn ngữ tập mờ của biến đầu vào.....	41
Bảng 2-2: Xác định tập mờ của giá trị đầu ra	42
Bảng 2-3: Hệ luật cơ sở suy diễn cho tham số K_1.....	42
Bảng 2-4: Hệ luật cơ sở suy diễn cho tham số K_2.....	43
Bảng 2-5: Bảng so sánh sai lệch quỹ đạo thuật toán DSC và Fuzzy-DSC	50
Bảng 2-6: Bảng đánh giá sai lệch của quỹ đạo robot.....	53

DANH MỤC CÁC HÌNH VẼ

Hình 1.1 Robot di động dạng non-holonomic.....	6
Hình 1.2 Robot di động dạng holonomic: (a) Robot đa hướng bánh xe omni, (b) Robot đa hướng bốn bánh xe mecanum.....	7
Hình 1.3 Các khả năng chuyển động của robot: (a) Robot sử dụng bánh xe truyền thống, (b) Robot đa hướng bốn bánh xe mecanum.....	8
Hình 1.4 Mô hình robot di động bốn bánh xe mecanum	9
Hình 1.5 Các hướng di chuyển của FMWR.....	9
Hình 1.6 Một số ứng dụng của FMWR.....	10
Hình 1.7 Sự thay đổi trọng tâm của robot tích hợp cánh tay cộng tác.....	10
Hình 1.8 Mô hình toán học của bánh xe mecanum.....	11
Hình 1.9 Mô hình robot bốn bánh xe mecanum	13
Hình 1.10 Mô hình FMWR có xét đến độ lệch trọng tâm	16
Hình 1.11 Sơ đồ tổng quan hệ thống điều khiển robot di động	19
Hình 2.1 Quỹ đạo di chuyển của FMWR.....	31
Hình 2.2 Sai lệch quỹ đạo của FMWR	32
Hình 2.3 Quỹ đạo bám của các thuật toán với FMWR có trọng tâm không đổi	36
Hình 2.4 Sai số của quỹ đạo theo trục x.....	37
Hình 2.5 Sai số của quỹ đạo theo trục y.....	37
Hình 2.6 Sai số của quỹ đạo theo góc hướng.....	37
Hình 2.7 Quỹ đạo bám của FMWR-ME khi trọng tâm thay đổi	38
Hình 2.8 Sai lệch quỹ đạo trục x của FMWR-ME.....	38
Hình 2.9 Sai lệch quỹ đạo trục y của FMWR-ME.....	39
Hình 2.10 Sai lệch quỹ đạo theo góc hướng của FMWR-ME	39
Hình 2.11 Quỹ đạo bám của thuật toán DSC với hình số 8.....	40
Hình 2.12 Sơ đồ khối thuật toán điều khiển Fuzzy-DSC.....	41
Hình 2.13 Tập mờ đối với biến đầu vào e_1 , $\dot{e}_1$	42
Hình 2.14 Kết quả mô phỏng thuật toán Fuzzy- DSC với quỹ đạo tròn.....	44
Hình 2.15 Quỹ đạo mô phỏng của robot theo trục x.....	44
Hình 2.16 Quỹ đạo mô phỏng của robot theo trục y.....	44
Hình 2.17 Quỹ đạo mô phỏng của robot theo góc hướng.....	45
Hình 2.18 Sai lệch quỹ đạo bám của robot	45
Hình 2.19 Sai lệch của vận tốc theo giới gian.....	46
Hình 2.20 Các tham số K_1 , K_2 chỉnh định theo thời gian	46
Hình 2.21 Tín hiệu điều khiển Tor với quỹ đạo đường tròn.....	47
Hình 2.22 Chuyển động của FMWR với quỹ đạo hình số 8.....	47
Hình 2.23 Quỹ đạo di chuyển của FMWR theo trục x	48
Hình 2.24 Quỹ đạo di chuyển của FMWR theo trục y	48
Hình 2.25 Quỹ đạo di chuyển của FMWR theo góc hướng.....	48
Hình 2.26 Sai lệch quỹ đạo với trọng tâm robot không đổi.....	49
Hình 2.27 Sai lệch vận tốc với trọng tâm robot không đổi.....	49
Hình 2.28 Chỉnh định các tham số K_1 , K_2	49

Hình 2.29 Tín hiệu điều khiển Tor với quỹ đạo hình số 8	50
Hình 2.30 Mô tả vị trí trọng tâm robot thay đổi biến thiên	51
Hình 2.31 Quỹ đạo di chuyển của FMWR-ME	51
Hình 2.32 Sai lệch của quỹ đạo FMWR-ME	52
Hình 2.33 Sai lệch vận tốc của FMWR-ME	52
Hình 2.34 Chính định các tham số khi lệch trọng tâm	53
Hình 2.35 Nhiễu ngoại lực Gaussian	54
Hình 2.36 Kết quả mô phỏng có nhiễu ngoại	54
Hình 2.37 Sai lệch quỹ đạo khi có nhiễu ngoại	55
Hình 2.38 Chính định tham số K_1 , K_2 khi có nhiễu	55
Hình 3.1 Mạng MLP một lớp ẩn [91]	59
Hình 3.2 Cấu trúc AC-NNs xấp xỉ nghiệm phương trình HJB	67
Hình 3.3 Sơ đồ điều khiển FMWR sử dụng cấu trúc actor-critic	76
Hình 3.4 Kết quả mô phỏng của FMWR với quỹ đạo đường tròn	80
Hình 3.5 Quỹ đạo di chuyển của FMWR theo trục x	80
Hình 3.6 Quỹ đạo di chuyển của FMWR theo trục y	81
Hình 3.7 Quỹ đạo di chuyển của FMWR theo góc hướng	81
Hình 3.8 Sai lệch quỹ đạo của FMWR theo các trục x, y và góc hướng	81
Hình 3.9 Trọng số của mạng nơ-ron Actor	82
Hình 3.10 Trọng số của mạng nơ-ron Critic	82
Hình 3.11 Tín hiệu mô men điều khiển của bánh xe	83
Hình 3.12 So sánh quỹ đạo FMWR và FMWR-ME	84
Hình 3.13 Sai lệch quỹ đạo khi trọng tâm thay đổi	84
Hình 3.14 Sự hội tụ của trọng số nơ-ron Actor- Critic	85
Hình 3.15 Tín hiệu điều khiển Tor của FMWR-ME	85
Hình 3.16 Nhiễu ngoại lực Gaussian	86
Hình 3.17 Quỹ đạo di chuyển khi có nhiễu tác động	86
Hình 3.18 Các sai lệch bám khi có nhiễu tác động vào mô hình	87
Hình 3.19 Các trọng số W_a và W_c khi có nhiễu tác động	87
Hình 3.20 Tín hiệu điều khiển Tor khi có nhiễu tác động vào mô hình	88
Hình 3.21 Quỹ đạo di chuyển thuật toán Fuzzy-DSC và RL-AC	89
Hình 3.22 So sánh sai lệch bám thuật toán Fuzzy-DSC và RL-AC	89
Hình 3.23 So sánh tín hiệu điều khiển Fuzzy-DSC và RL-AC	90
Hình 4.1 Bản vẽ thiết kế cơ khí cho mô hình FMWR	92
Hình 4.2 Mô hình cơ khí FMWR hoàn thiện	93
Hình 4.3 Cấu trúc hệ thống điều khiển của FMWR	93
Hình 4.4 Mô hình robot di động bốn bánh xe mecanum hoàn thiện	94
Hình 4.5 Thiết kế mô hình FMWR trong Gazebo	96
Hình 4.6 Quá trình SLAM bản đồ hóa môi trường	97
Hình 4.7 Quỹ đạo di chuyển của FMWR	97
Hình 4.8 Quá trình điều hướng robot tránh vật cản	98
Hình 4.9 Sơ đồ khối lập trình thực nghiệm thuật toán Fuzzy-DSC	99
Hình 4.10 Camera theo dõi quỹ đạo di chuyển robot	100
Hình 4.11 Thực nghiệm với quỹ đạo đường tròn trọng tâm không đổi	101

Hình 4.12 Thêm tải trọng 5kg và đặt lệch trọng tâm trên robot	101
Hình 4.13 Thực nghiệm với tải trọng 5kg và đặt lệch trọng tâm trên robot	102
Hình 4.14 Thực nghiệm với quỹ đạo hình số 8 trọng tâm không đổi	103
Hình 4.15 Thực nghiệm quỹ đạo hình số 8 trọng tâm thay đổi	104
Hình 4.16 Thực nghiệm mô hình FMWR với quỹ đạo đường zích zắc	105

MỞ ĐẦU

1. Tính cấp thiết của luận án

Trong cuộc cách mạng công nghiệp 4.0 yêu cầu các nhà máy tự động hóa hoàn toàn, trong đó robot di động là một thành phần không thể thiếu. Các robot di động giúp tối ưu hóa quy trình làm việc, giảm thiểu sự can thiệp của con người, tăng hiệu suất và đảm bảo sự chính xác. Trong các loại robot di động phổ biến hiện nay thì robot di động đa hướng có ưu điểm chuyển động linh hoạt theo mọi hướng mà không cần phụ thuộc vào góc quay của robot, có thể làm việc trong không gian hạn chế.

Thông thường, các loại robot di động đa hướng thường sử dụng bánh xe omni hoặc bánh xe mecanum. Trong đó, robot di động đa hướng sử dụng bánh xe mecanum có khả năng chịu tải trọng lớn, có khả năng hoạt động linh hoạt trong các môi trường phức tạp, hẹp và đông đúc, như trong các nhà kho, bệnh viện, khu vực sản xuất công nghiệp hoặc môi trường đô thị. Tuy nhiên, trong thực tế những môi trường này thường có nhiều yếu tố thay đổi liên tục (chướng ngại vật, mặt sàn di chuyển không bằng phẳng, v.v.), đặc biệt là khi tải trọng thay đổi hoặc trong các tình huống gây mất ổn định (ví dụ, khi robot mang theo các vật liệu không đồng đều), khi đó vị trí trọng tâm của robot thay đổi có thể ảnh hưởng lớn đến chuyển động và khả năng bám quỹ đạo. Do đó, việc nghiên cứu các thuật toán điều khiển thích nghi giúp robot hoạt động chính xác và ổn định hơn trong môi trường làm việc thực tế.

Bài toán điều khiển bám quỹ đạo cho robot di động đa hướng bánh xe mecanum là quan trọng và cấp thiết. Yêu cầu các hệ thống có khả năng điều khiển robot duy trì quỹ đạo mà không bị lệch hoặc mất ổn định, đặc biệt khi mô hình robot là phi tuyến có các thành phần bất định chịu ảnh hưởng của môi trường làm việc hoặc khi di chuyển trên địa hình không bằng phẳng. Việc phát triển các thuật toán điều khiển bám quỹ đạo giúp giảm thiểu sai số và ổn định trong quá trình vận hành. Các hướng nghiên cứu chính có thể kể đến như sử dụng thuật toán điều khiển PID thông thường, điều khiển PID mờ hay một số nghiên cứu sử dụng bộ điều khiển mặt trượt. Gần đây các thuật toán điều khiển tiên tiến như MPC (Model Predictive Control), LQR (Linear Quadratic Regulator), và các phương pháp học máy (machine learning) được áp dụng và chứng minh tính hiệu quả.

Trong quá trình nghiên cứu, thách thức lớn cần giải quyết là phải phát triển các thuật toán điều khiển có thể thích nghi với các yếu tố môi trường không ổn định, như trọng tâm bất định của robot khi vận chuyển các vật nặng hoặc khi di chuyển trên địa hình không bằng phẳng. Các thuật toán điều khiển học máy, học tăng cường, học sâu dựa trên mạng nơ-ron nhân tạo cũng được nghiên cứu áp dụng vào điều khiển bám quỹ đạo cho robot, đạt được chất lượng điều khiển tốt hơn.

Với những xu hướng như trên, NCS đã lựa chọn đề tài: ***Nghiên cứu xây dựng thuật toán thích nghi và học tăng cường cấu trúc Actor-Critic điều khiển bám quỹ đạo cho robot di động đa hướng mecanum***. Nội dung nghiên cứu của luận án sẽ tập trung vào xây dựng các thuật toán điều khiển thích nghi và học tăng cường nhằm nâng cao chất lượng điều khiển bám quỹ đạo cho robot di động đa hướng bốn bánh xe mecanum có vị trí trọng tâm thay đổi. Luận án áp dụng các thuật toán điều khiển truyền thống, đến các thuật toán điều khiển học máy, học tăng cường để so sánh kiểm chứng với chất lượng thuật toán điều khiển được đề xuất.

2. Mục tiêu nghiên cứu của luận án

Mục tiêu nghiên cứu thuật toán điều khiển thích nghi và học tăng cường nâng cao chất lượng điều khiển bám quỹ đạo cho robot di động đa hướng mecanum, trong điều kiện chịu ảnh hưởng của trọng tâm thay đổi và có nhiều ngoại tác động. Luận án đã đặt ra các nhiệm vụ nghiên cứu chính sau:

- Nghiên cứu thuật toán điều khiển thích nghi nhằm nâng cao chất lượng điều khiển bám quỹ đạo cho robot di động đa hướng mecanum có trọng tâm thay đổi
- Nghiên cứu thuật toán học tăng cường cấu trúc Actor- Critic áp dụng điều khiển bám quỹ đạo cho robot di động đa hướng mecanum có trọng tâm thay đổi

3. Nội dung nghiên cứu

- Tổng quan về robot di động đa hướng, các nghiên cứu trong và ngoài nước, đưa ra hướng nghiên cứu của luận án.
- Mô hình hóa toán học cho robot di động bốn bánh xe mecanum có trọng tâm thay đổi.
- Áp dụng các thuật toán để điều khiển bám quỹ đạo cho robot mecanum như: PID, SMC, Backstepping-SMC, DSC.

- Nghiên cứu đề xuất thuật toán điều khiển thích nghi dựa trên hệ logic mờ nhằm nâng cao chất lượng bám quỹ đạo của robot di động đa hướng mecanum có trọng tâm thay đổi và nhiều ngoại tác động.
- Nghiên cứu đề xuất thuật toán học tăng cường cấu trúc actor- critic điều khiển bám quỹ đạo cho robot di động đa hướng mecanum chịu ảnh hưởng của nhiều ngoại lực và các tham số bất định của mô hình robot.
- Mô phỏng và thực nghiệm thuật toán trên mô hình robot thực từ đó đánh giá chất lượng và khả năng thực tiễn của thuật toán được đề xuất.

4. Ý nghĩa khoa học và thực tiễn của đề tài

Hiện nay, bài toán điều khiển bám quỹ đạo cho robot di động đa hướng mecanum có tính cấp thiết. Không chỉ phục vụ cho các ngành công nghiệp và sản xuất, nghiên cứu này còn góp phần phát triển công nghệ tự động hóa, tăng cường khả năng ứng dụng robot trong cứu hộ, an ninh, và các lĩnh vực giao thông tự động. Các thuật toán điều khiển bám chính xác quỹ đạo sẽ giúp robot hoạt động ổn định và hiệu quả, đáp ứng được yêu cầu ngày càng cao trong các môi trường phức tạp và ứng dụng thực tế.

Luận án đề xuất phương pháp điều khiển thích nghi, học tăng cường dựa trên luật logic mờ và sử dụng mạng nơ-ron nhân tạo là một hướng đi mới cho bài toán điều khiển bám quỹ đạo robot di động đa hướng bốn bánh xe mecanum để thích nghi với ảnh hưởng của thay đổi tâm khối lượng và các nhiễu bất định của mô hình, nhiễu ngoài.

Các kết quả nghiên cứu của luận án là cơ sở khoa học để áp dụng vào thực tiễn, cùng với việc xây dựng mô hình robot thử nghiệm để kiểm chứng thuật toán, từ đó mở ra khả năng áp dụng vào thực tiễn.

5. Đóng góp mới của luận án

Các đóng góp mới của luận án:

➤ **Đề xuất thuật toán điều khiển mặt trượt động chỉnh định thích nghi mờ để nâng cao chất lượng điều khiển bám quỹ đạo cho robot tự hành đa hướng bốn bánh xe mecanum có trọng tâm thay đổi.** Thuật toán điều khiển được phát triển với bộ điều khiển logic mờ chỉnh định các tham số của thuật toán điều khiển

mặt trượt động, sử dụng đầu vào của bộ chỉnh định mờ là sai lệch vị trí và vận tốc thông qua mô hình mờ sugeno để chỉnh định trực tuyến các tham số điều khiển K_1 , K_2 của thuật toán điều khiển mặt trượt động. Phương pháp điều khiển được đề xuất áp dụng cho FMWR có trọng tâm thay đổi, đảm bảo cho robot bám quỹ đạo tốt hơn ngay cả khi thay đổi khối lượng và độ lệch tâm, thời gian robot tiến đến quỹ đạo nhanh và sai lệch bám cũng nhỏ hơn.

➤ **Đề xuất thuật toán học tăng cường cấu trúc Actor-Critic điều khiển tối ưu bám quỹ đạo cho robot tự hành đa hướng bốn bánh xe mecanum có trọng tâm thay đổi.** Thuật toán học tăng cường được phát triển sử dụng quy hoạch động của Bellman, nghiệm của phương trình HJB được xấp xỉ thông qua giải thuật lặp PI để giải. Thuật toán sử dụng hai mạng nơ-ron với cấu trúc Actor – Critic để giải xấp xỉ nghiệm phương trình HJB. Trong đó mạng Actor dùng để xấp xỉ luật điều khiển tối ưu, mạng Critic dùng để xấp xỉ hàm Bellman. Tín hiệu điều khiển từ Actor sẽ liên tục được cập nhật và đưa vào kích thích hệ thống cho đến khi tín hiệu điều khiển là tối ưu, các luật cập nhật của mạng Critic được thực hiện đồng thời trong một bước lặp dựa theo tối thiểu hóa sai số hàm Bellman tương ứng. Thuật toán được chứng minh tính ổn định theo tiêu chuẩn hàm Lyapunov. Các kết quả mô phỏng của thuật toán được áp dụng điều khiển bám quỹ đạo cho FMWR có trọng tâm thay đổi và ảnh hưởng của nhiễu ngoại. Cho thấy khả năng bám quỹ đạo và ổn định ngay cả khi trọng tâm của robot thay đổi, chịu được ảnh hưởng của các nhiễu ngoại lực tác động với sai lệch nhỏ.

6. Bố cục của luận án

Luận án được trình bày bao gồm 4 chương:

Chương 1: Tổng quan về robot di động đa hướng.

Nội dung nghiên cứu các loại robot di động bánh xe đa hướng, phân tích các ưu điểm, nhược điểm của các công bố liên quan hiện nay từ đó đặt ra hướng nghiên cứu của luận án. Xây dựng mô hình động học, động lực học cho FMWR dựa trên phương trình Euler Lagrange.

Chương 2: Thuật toán điều khiển bám quỹ đạo cho robot di động đa hướng mecanum

Áp dụng các thuật toán điều khiển bám quỹ đạo như: thuật toán PID, điều khiển trượt, điều khiển mặt trượt động, để đánh giá ưu nhược điểm của từng thực toán và từ đó đề xuất thuật toán mới nhằm nâng cao chất lượng điều khiển bám quỹ đạo.

Chương 3: Thuật toán học tăng cường điều khiển bám quỹ đạo cho robot di động đa hướng mecanum

Nghiên cứu các thuật toán học máy, học tăng cường từ đó đề xuất thuật toán điều khiển với cấu trúc Actor-Critic điều khiển FMWR bám quỹ đạo. Thuật toán dựa trên phương trình Hamilton- Jacobi- Bellman (HJB). Sử dụng thuật toán học tăng cường với cấu trúc Actor- Critic để xấp xỉ nghiệm phương trình HJB. Khả năng bám quỹ đạo và ổn định của thuật toán được chứng minh bằng hàm Lyapunov. Mô phỏng thuật toán trên matlab để kiểm chứng tính hiệu quả của thuật toán.

Chương 4: Thực nghiệm với mô hình robot di động đa hướng mecanum

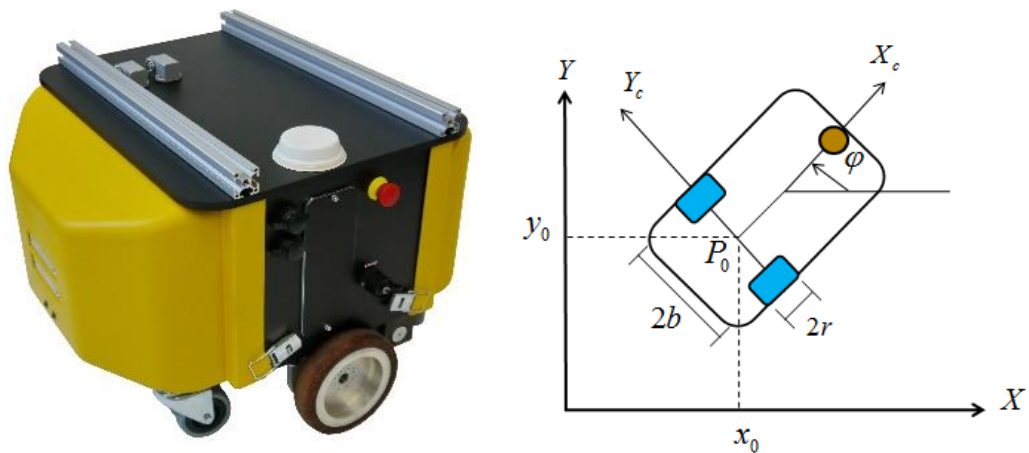
Thiết kế và chế tạo mô hình robot di động bốn bánh xe mecanum với kích thước giống với mô hình toán học đã được nghiên cứu trong chương 2. Mô hình được chạy thử nghiệm với các thuật toán đề xuất, từ đó đánh giá khả năng ứng dụng FMWR vào thực tiễn.

CHƯƠNG 1. TỔNG QUAN VỀ ROBOT DI ĐỘNG ĐA HƯỚNG

1.1. Tổng quan về robot di động

Robot di động (Mobile Robot) là loại robot có khả năng di chuyển và thực hiện các nhiệm vụ mà không cần sự can thiệp trực tiếp của con người. Những robot này thường được trang bị các cảm biến, hệ thống điều khiển và phần mềm để tương tác với môi trường và thực hiện các nhiệm vụ một cách tự động. Chúng có thể di chuyển trên nhiều bề mặt khác nhau và thường được thiết kế để hoạt động trong các môi trường thay đổi, bao gồm cả trong nhà và ngoài trời. Hiện nay, robot di động được ứng dụng rộng rãi trong đời sống xã hội như xe tự lái, robot di động phục vụ, vận chuyển kho hàng ..., trong lĩnh vực sản xuất công nghiệp là các robot vận chuyển vật liệu, hàng hóa trong các nhà máy, xưởng sản xuất, kho bãi; trong lĩnh vực an ninh quốc phòng là các robot trinh sát, dò phá bom mìn, robot chiến trường...; trong hàng không vũ trụ, thám hiểm, khám phá ở các hành tinh ngoài trái đất...

Tùy theo từng yêu cầu nhiệm vụ cụ thể hoạt động và di chuyển khác nhau mà robot di động được thiết kế theo hai loại: robot non-holonomic sử dụng các bánh xe truyền thống có cấu tạo như hình 1.1 và robot đa hướng holonomic được sử dụng bánh xe omni hoặc bánh xe mecanum như hình 1.2.



Hình 1.1 Robot di động dạng non-holonomic

Robot di động truyền thống dạng non-holonomic chỉ xét đến hai bậc tự do điều khiển là: di chuyển tịnh tiến và di chuyển theo góc quay. Robot dạng này có ưu điểm là cấu tạo đơn giản với cơ cấu dẫn động với 2 bánh xe chủ động phù hợp với các quỹ đạo di chuyển tịnh tiến và rẽ hướng đơn giản.



(a)

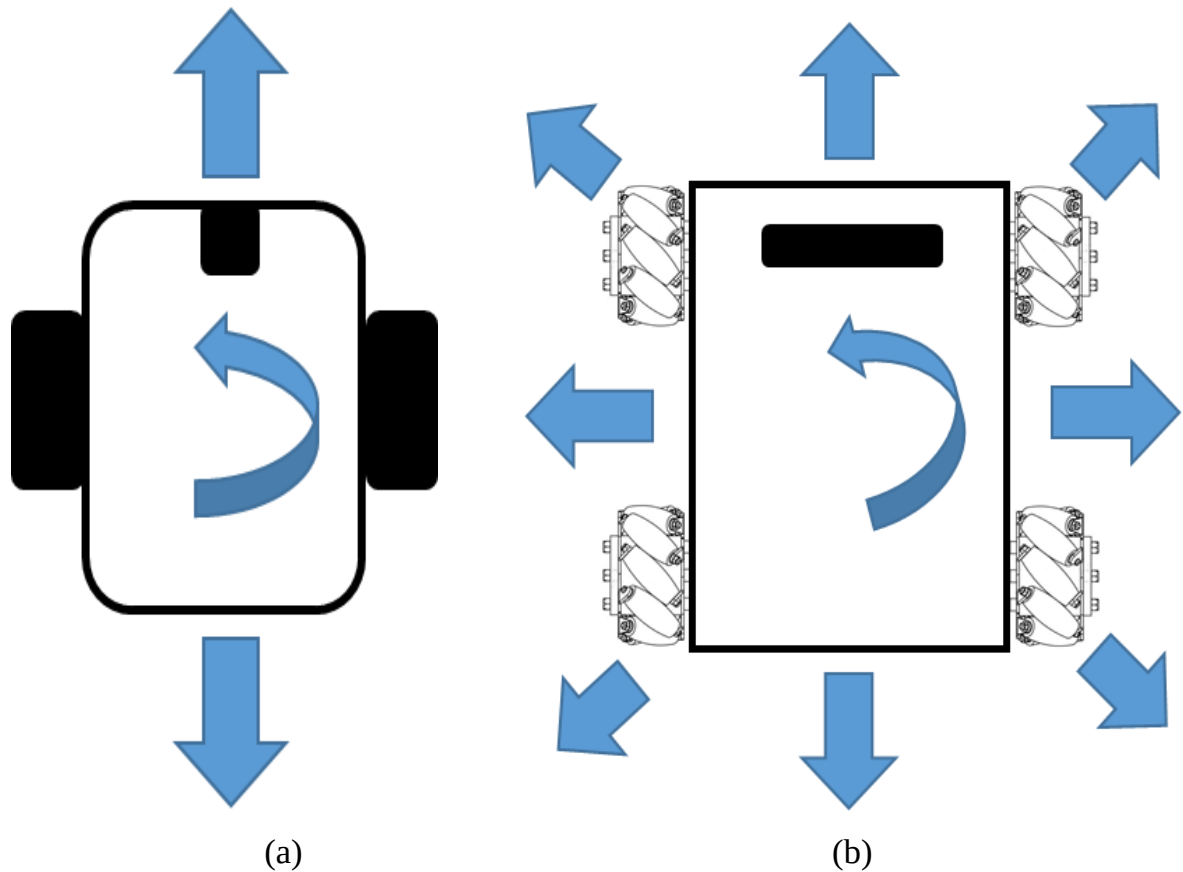


(b)

Hình 1.2 Robot di động dạng holonomic: (a) Robot đa hướng bánh xe omni, (b) Robot đa hướng bốn bánh xe mecanum

Robot di động dạng holonomic hay còn gọi là robot di động đa hướng, được xét đến đầy đủ cả ba phương di chuyển: tịnh tiến, ngang dọc và theo góc quay, do đó nó tăng tính linh hoạt, đặc biệt robot có thể di chuyển theo phương bất kỳ mà góc hướng không đổi. Trong số các loại robot đa hướng thường sử dụng loại bánh xe omni với thiết kế là ba bánh hoặc bốn bánh như hình 1.2(a) và dạng thứ hai là robot sử dụng bánh xe mecanum như hình 1.2(b), các bánh xe của robot được dẫn động độc lập với nhau, trên mỗi bánh xe có các con lăn vệ tinh đặt lệch góc so với trục chính bánh xe giúp chuyển một phần lực tịnh tiến thành lực trượt ngang giúp robot có thể di chuyển đa hướng.

Các khả năng chuyển động của robot được thể hiện trong hình 1.3. Đối với robot sử dụng bánh xe truyền thống thì khả năng di chuyển tịnh tiến và rẽ hướng phụ thuộc vào góc hướng của robot như trong hình 1.3(a), còn đối với robot đa hướng sử dụng bánh xe mecanum thì các hướng di chuyển của robot linh hoạt hơn khi có khả năng di chuyển theo các hướng khác nhau như hình 1.3(b) mà không phụ thuộc đến góc hướng của robot.

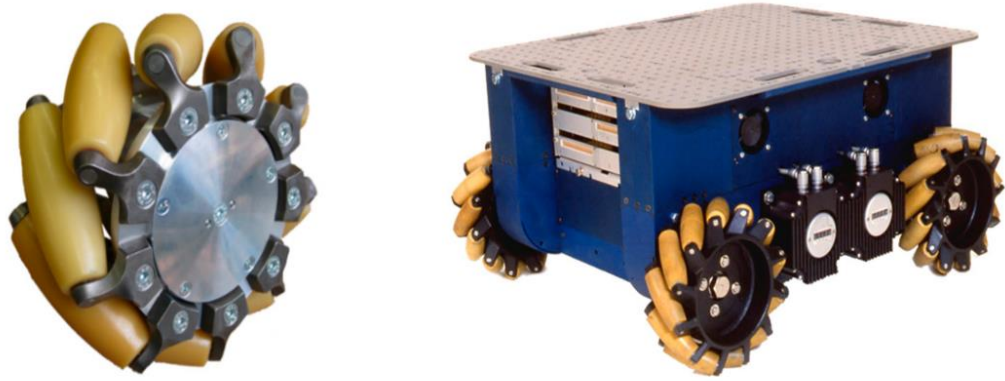


Hình 1.3 Các khả năng chuyển động của robot: (a) Robot sử dụng bánh xe truyền thống, (b) Robot đa hướng bốn bánh xe mecanum

Nhờ vào những ưu điểm như khả năng cơ động tốt hơn, không cần quay hướng, hiệu suất cao trong không gian hạn chế, tiết kiệm không gian và thời gian, do đó robot di động bánh xe đa hướng đã trở thành đối tượng nghiên cứu được nhiều nhà khoa học quan tâm. So với các loại robot di động đa hướng khác có cùng kích thước thì robot đa hướng sử dụng bánh xe mecanum di chuyển mượt mà hơn, có khả năng mang được tải trọng lớn nên được sử dụng nhiều trong công nghiệp và nghiên cứu, đây cũng là đối tượng nghiên cứu chính của luận án.

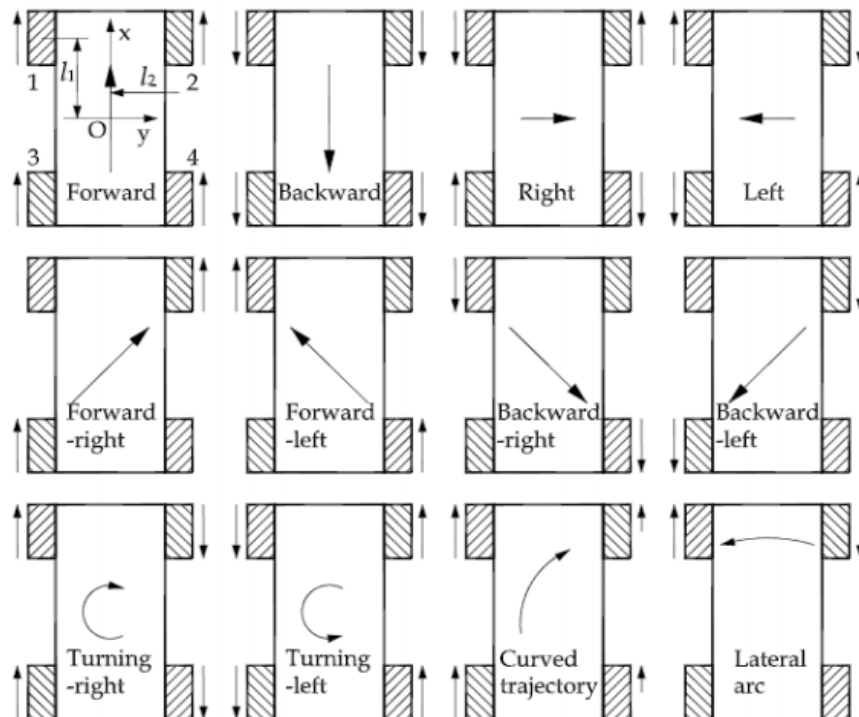
1.2. Robot di động đa hướng mecanum

Robot di động đa hướng mecanum (FMWR) được thiết kế sử dụng bốn bánh xe mecanum được truyền dẫn độc lập với nhau bởi bốn động cơ riêng biệt. Các cặp bánh xe được sắp xếp đối xứng nhau tạo một tư thế cân bằng cho robot và đảm bảo tính động học trên mỗi bánh xe. Trên từng bánh xe mecanum được thiết kế với các con lăn vệ tinh thụ động đặt lệch so với trục chính của bánh xe một góc 45^0 như hình 1.4.



Hình 1.4 Mô hình robot di động bốn bánh xe mecanum

Khi robot hoạt động thì các bánh xe được động cơ truyền động quay theo phương vuông góc với trục truyền động, đồng thời các con lăn thụ động của bánh xe chuyển động biến đổi một phần lực tịnh tiến thành lực trượt ngang, nhờ đó các lực ngang giúp robot di theo phương ngang linh hoạt hay phương bất kỳ mà không phụ thuộc vào góc hướng của robot. Với thiết kế của robot sử dụng bánh xe mecanum thì các hướng chuyển động của robot được thể hiện hình 1.5 [1].



Hình 1.5 Các hướng di chuyển của FMWR

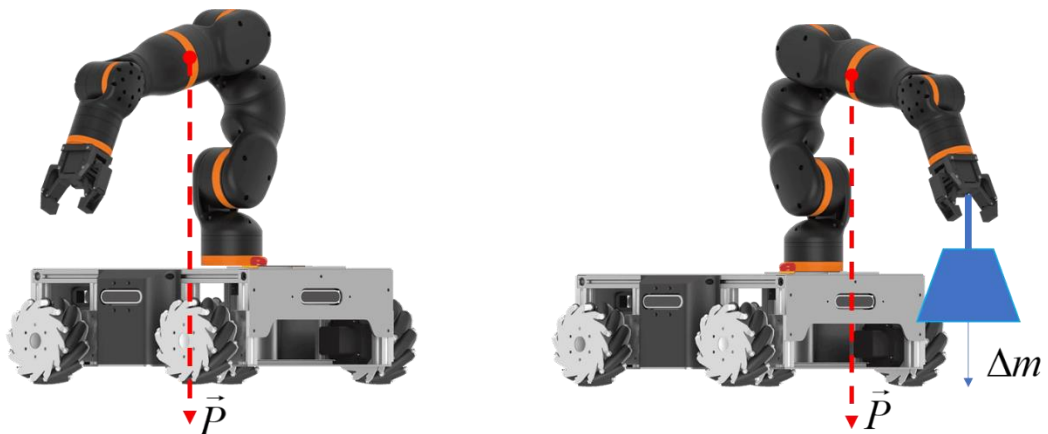
Phương thức di chuyển của FMWR: để cho robot hoạt động thì các bánh xe được di chuyển theo từng cặp và quay hướng khác nhau để tạo thành các hướng di

chuyển mong muốn. Trường hợp robot muốn di chuyển tịnh tiến về phía trước thì các cặp bánh xe 1-3, 2-4 sẽ quay theo chiều thuận và hoạt động cùng tốc độ với nhau, nếu muốn robot di chuyển ngược lại thì các động cơ bánh xe quay đảo chiều. Trường hợp robot di chuyển sang phải mà giữ nguyên góc hướng của robot thì cặp bánh xe 1-4 sẽ quay ngược đồng thời cặp bánh xe 2-3 sẽ quay chiều thuận. Khi muốn góc hướng của robot quay theo chiều sang phải mà vị trí không đổi thì các cặp bánh xe 1-3 quay theo chiều thuận đồng thời cặp bánh xe 2-4 quay theo chiều nghịch. Tương tự như vậy thì các hướng di chuyển khác nhau của robot được thể hiện trong hình 1.5.



Hình 1.6 Một số ứng dụng của FMWR

Nhờ vào các ưu điểm di chuyển linh hoạt và chịu được tải trọng lớn thì FMWR được ứng dụng rộng rãi trong công nghiệp như các robot di động làm nhiệm vụ nâng hạ trong kho hàng, nhà máy, vận chuyển, hay robot được tích hợp tay máy thao tác và chuyển động trong các dây chuyền sản xuất, robot thám hiểm ở trong những môi trường phóng xạ, robot có thể được sử dụng trong các nhiệm vụ khảo sát địa hình khó tiếp cận hoặc nguy hiểm như trong không gian, dưới nước hoặc các vùng đất khắc nghiệt....một số ứng dụng của robot mecanum trong hình 1.6.



Hình 1.7 Sự thay đổi trọng tâm của robot tích hợp cánh tay cộng tác

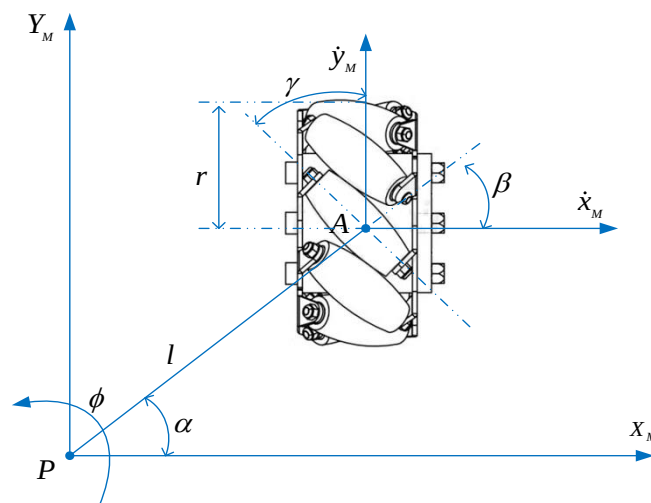
Đối với một số ứng dụng của FMWR như: robot vận chuyển hàng hóa trong kho bãi hoặc robot có tích hợp tay máy cộng tác như hình 1.7. Trong quá trình hoạt động khi robot mang thêm một khối lượng hàng hóa điều đó dẫn đến tổng khối lượng của robot bị thay đổi và trọng tâm của robot cũng thay đổi theo. Khi trọng tâm của robot lệch nhiều về một phía (ví dụ khi robot quay hoặc chuyển động theo đường chéo), có thể dẫn đến việc robot mất thăng bằng và ảnh hưởng đến độ ổn định. Từ đó đặt ra vấn đề, nghiên cứu các thuật toán điều khiển có khả năng thích nghi với các thay đổi đó, và tự động điều chỉnh các tham số (như tốc độ, hướng, và lực tác động lên bánh xe) để đảm bảo robot ổn định trong suốt quá trình di chuyển và bám quỹ đạo một cách chính xác.

1.3. Mô hình toán học robot di động đa hướng mecanum

1.3.1. Cấu tạo bánh xe mecanum

Trong số các loại bánh xe đa hướng, bánh xe mecanum là một loại bánh xe đặc trưng. Với thiết kế của bánh xe này bao gồm các con lăn vệ tinh được gắn nghiêng một góc 45° so với trục chính của bánh xe. Khi trục chính bánh xe được truyền động từ động cơ thì bánh xe chuyển động, đồng thời các con lăn đặt xung quanh bánh xe cũng chuyển động thụ động, biến đổi một phần lực di chuyển tịnh tiến thành lực trượt ngang của mỗi bánh xe, lực truyền động đó giúp bánh xe có thể di chuyển đồng thời theo phương tịnh tiến và phương ngang.

Xét mô hình toán học của bánh xe mecanum [54] được mô tả trong hình 1.8.



Hình 1.8 Mô hình toán học của bánh xe mecanum

Trong hệ trục tọa độ $PX_M Y_M$ trong đó điểm A là tâm của bánh xe, các thông số được xác định trong hệ trục tọa độ như sau:

α - là góc của vector PA so với trục X_M

β - là góc giữa vector PA so với trục chính của bánh xe

l - là khoảng cách giữa tâm P đến tâm bánh xe A

r - là bán kính của bánh xe

γ - là góc giữa con lăn và trục chính bánh xe

ψ - là góc quay của bánh xe quanh trục chính

ϕ - là góc hướng của robot

Vận tốc tức thời của robot là $[\dot{x}_M \quad \dot{y}_M]^T$ và vận tốc quanh trục bánh xe là $\dot{\psi}$, vận tốc dài của bánh xe A có thể được tính bằng tổng véc tơ vận tốc tịnh tiến $\dot{x}_R, \dot{y}_R$ và vận tốc tương đối được tính $l\dot{\phi}$. Giả sử độ trượt dọc của bánh xe $\rho = 0$ thì vận tốc của bánh xe được tính như sau [54]:

$$r\dot{\psi} \cos \gamma = [\sin(\alpha + \beta + \gamma) - \cos(\alpha + \beta + \gamma) - l \cos(\beta + \gamma)][\dot{x}_R \quad \dot{y}_R \quad \dot{\phi}]^T \quad (1.1)$$

Hướng của khung tọa độ robot so với khung tọa độ gốc được biểu diễn bằng ma trận quay:

$$\mathbf{H}(\phi) = \begin{bmatrix} \cos \phi & \sin \phi & 0 \\ -\sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (1.2)$$

trong đó: ϕ - là góc giữa trục X_M của hệ trục tọa độ gắn với robot so với trục X của hệ trục tọa độ gốc.

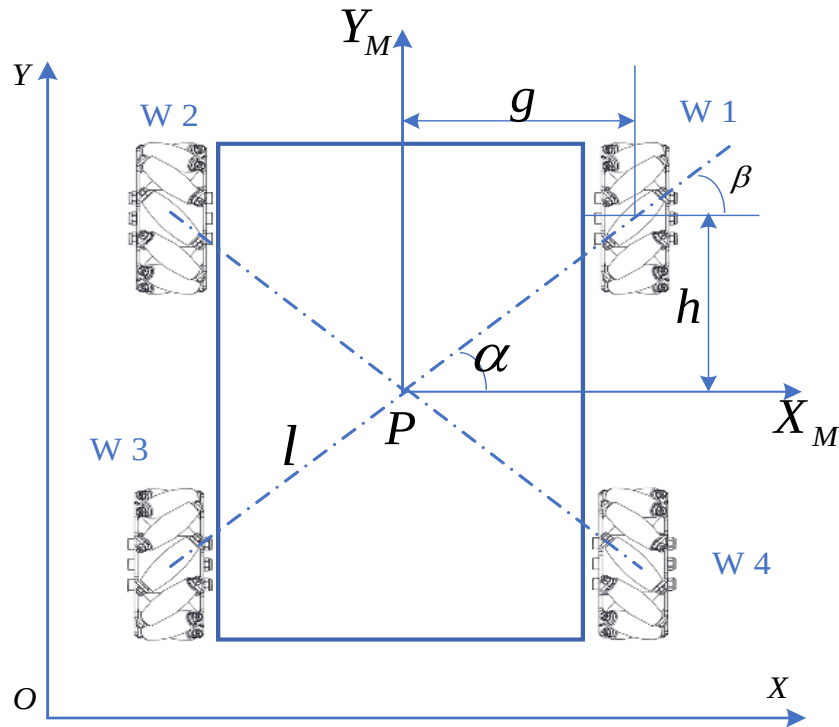
Từ đó vận tốc của bánh xe $\mathbf{v}_\psi = [\dot{x}_M \quad \dot{y}_M \quad \dot{\phi}]^T$ được tính là: $\mathbf{v}_\psi = \mathbf{H}(\phi)\dot{\mathbf{n}}$

Trong đó: $\dot{\mathbf{n}} = [\dot{x} \quad \dot{y} \quad \dot{\phi}]^T$ là véc tơ vận tốc của robot gắn với hệ quy chiếu quán tính XOY thì phương trình (1.1) có thể được chuyển về dạng sau:

$$[\sin(\alpha + \beta + \gamma) - \cos(\alpha + \beta + \gamma) - l \cos(\beta + \gamma)] \cdot \mathbf{H}(\phi) \dot{\mathbf{q}} = r \dot{\psi} \cos \gamma \quad (1.3)$$

1.3.2. Phương trình động học robot di động bốn bánh xe mecanum

Mô hình robot di động đa hướng mecanum (FMWR) được thiết kế với bốn bánh xe mecanum [54] được sắp xếp đối xứng theo từng cặp như thể hiện trong hình 1.9. Mỗi bánh xe được dẫn động bởi một động cơ riêng biệt, cho phép các bánh xe di chuyển độc lập với nhau nhờ vào thiết kế này mà mỗi bánh xe có thể chuyển động độc lập với nhau để tạo ra lực tịnh tiến và lực trượt ngang khác nhau giúp cho robot có thể di chuyển dọc hay di chuyển ngang mà ko cần thay đổi góc hướng của robot.



Hình 1.9 Mô hình robot bốn bánh xe mecanum

Giả thiết với mô hình FMWR được thiết kế với các bánh xe có cùng kích thước: $r_i = r$ và góc lệch giữa các con lăn so với trục bánh xe $\gamma = 45^\circ$. Khoảng cách giữa các bánh xe $i = 1, 2, 3, 4$ đến trọng tâm P của robot là đối xứng nhau với chiều dài là $l_i = l$ xác định được [54]:

$$\begin{bmatrix} \dot{\psi}_1 \\ \dot{\psi}_2 \\ \dot{\psi}_3 \\ \dot{\psi}_4 \end{bmatrix} = \frac{1}{r} \begin{bmatrix} -1 & -1 & -\sqrt{2}l \sin(\frac{\pi}{4} - \alpha) \\ -1 & 1 & -\sqrt{2}l \sin(\frac{\pi}{4} - \alpha) \\ 1 & 1 & -\sqrt{2}l \sin(\frac{\pi}{4} - \alpha) \\ 1 & -1 & -\sqrt{2}l \sin(\frac{\pi}{4} - \alpha) \end{bmatrix} \begin{bmatrix} \cos \phi & \sin \phi & 0 \\ -\sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} \quad (1.4)$$

Trong đó: $\alpha = \tan^{-1}(h / g)$

Các thông số α_i, β_i và γ_i của từng bánh xe được thể hiện ở trong bảng 1-1.

Bảng 1-1. Thông số của các bánh xe mecanum

Wheels	α_i	β_i	γ_i
1	$\tan^{-1}(h / g)$	$-\tan^{-1}(h / g)$	$(45^\circ + \pi / 2)$
2	$\pi - \tan^{-1}(h / g)$	$\tan^{-1}(h / g)$	$-(45^\circ + \pi / 2)$
3	$\pi + \tan^{-1}(h / g)$	$-\tan^{-1}(h / g)$	$(45^\circ + \pi / 2)$
4	$2\pi - \tan^{-1}(h / g)$	$\tan^{-1}(h / g)$	$-(45^\circ + \pi / 2)$

Từ đó ma trận Jacobian được xác định:

$$\mathbf{J} = \begin{bmatrix} -1 & -1 & -\sqrt{2}l \sin(\frac{\pi}{4} - \alpha) \\ -1 & 1 & -\sqrt{2}l \sin(\frac{\pi}{4} - \alpha) \\ 1 & 1 & -\sqrt{2}l \sin(\frac{\pi}{4} - \alpha) \\ 1 & -1 & -\sqrt{2}l \sin(\frac{\pi}{4} - \alpha) \end{bmatrix} \begin{bmatrix} \cos \phi & \sin \phi & 0 \\ -\sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (1.5)$$

Phương trình động học thuận mô tả sự liên kết giữa vị trí của robot và vận tốc quay của các bánh xe có thể được diễn đạt như sau:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \frac{1}{2} r \mathbf{J}^+ \begin{bmatrix} \dot{\psi}_1 \\ \dot{\psi}_2 \\ \dot{\psi}_3 \\ \dot{\psi}_4 \end{bmatrix} \quad (1.6)$$

Trong đó: $\mathbf{J}^+ = (\mathbf{J}^T \mathbf{J})^{-1} \mathbf{J}^T$ là ma trận giả nghịch đảo của $\mathbf{J}$

Phương trình động học ngược của FMWR trong hệ tọa độ toàn cục được xác định như sau:

$$\begin{bmatrix} \dot{x}_R \\ \dot{y}_R \\ \dot{\phi}_R \end{bmatrix} = \begin{bmatrix} \cos \phi & \sin \phi & 0 \\ -\sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} \quad (1.7)$$

Phương trình động học thuận có thể được viết dưới dạng ma trận như sau:

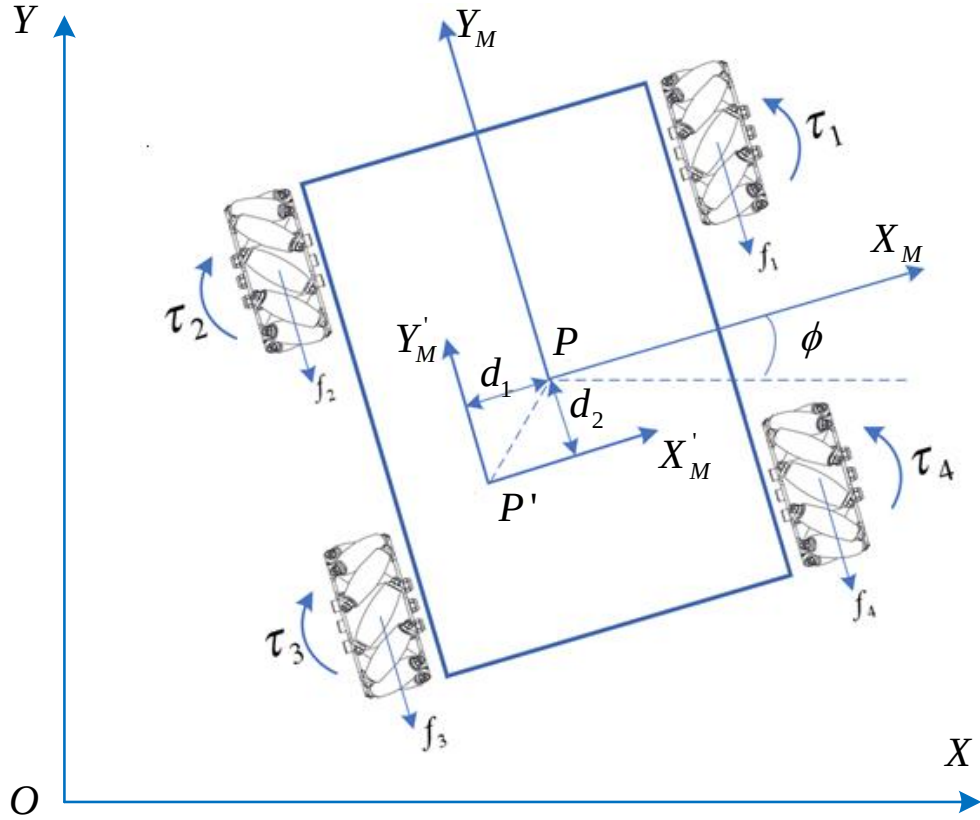
$$\dot{\mathbf{\eta}}_R = \mathbf{H}(\phi) \dot{\mathbf{\eta}} \quad (1.8)$$

Trong đó: $\mathbf{\eta}_R = [\dot{x}_R \quad \dot{y}_R \quad \dot{\phi}_R]^T$ - là vận tốc theo trục x, y và góc hướng của robot trên hệ trục tọa độ gắn với robot. $\mathbf{\eta} = [\dot{x} \quad \dot{y} \quad \dot{\phi}]^T$ - là vận tốc theo trục x, y và góc hướng của robot so với hệ trục tọa độ gốc.

1.3.3. Phương trình động lực học robot di động bốn bánh xe mecanum

Xét mô hình robot sử dụng bốn bánh xe mecanum, trong đó trọng tâm thực tế không trùng với tâm hình học của mô hình robot. Điều này có nghĩa là vị trí trọng tâm của robot có thể lệch khỏi trung tâm hình học của cấu trúc robot, và sự khác biệt này ảnh hưởng đến động lực học và các đặc tính điều khiển của robot, điều đó ảnh hưởng đến chất lượng điều khiển cho robot. Mô hình robot có trọng tâm thay đổi (FMWR-ME) thể hiện trong hình 1.10. Trong đó, vị trí trọng tâm $\mathbf{P} = [x \quad y]^T$ được xét đến là vị trí trọng tâm theo hệ quy chiếu của robot và vị trí $\mathbf{P}'$ là vị trí trọng tâm thay đổi của robot được xét đến có vị trí tương đối thể hiện bởi $\mathbf{P}' = [x - d_1 \quad y - d_2]^T$ theo khung tọa độ của robot. Vị trí trọng tâm $\mathbf{P}'$ có thể là cố định hoặc biến thiên trong trường hợp mang tải trọng hàng hóa có trọng tâm khó xác định (hoặc bất định). Để đảm bảo cho FMWR cân bằng, có thể hoạt động ổn định mà không bị lật khi di chuyển thì trong luận án giới hạn độ lệch trọng tâm của robot không vượt quá:

$$d_1 < \frac{1}{2}g \text{ và } d_2 < \frac{1}{2}h.$$



Hình 1.10 Mô hình FMWR có xét đến độ lệch trọng tâm

Tổng động năng của robot bao gồm cả khối lượng của robot và bốn bánh xe được tính như sau [54]:

$$T = \frac{1}{2} \left[m_b \mathbf{P}^T \mathbf{P}' + J_b \dot{\phi}^2 + \sum_{i=1}^4 m_{wi} (r \dot{\psi}_i)^2 + \sum_{i=1}^4 J_i \dot{\psi}_i^2 \right] \quad (1.9)$$

Trong đó: m_b là khối lượng của robot; m_{wi} là khối lượng của bánh xe thứ i ; J_b là moment quán tính của robot, J_{wi} là moment quán tính của bánh xe thứ i so với trục bánh xe.

Giải thiết robot hoạt động trên mặt phẳng sàn nên thế năng toàn phần $V = 0$, nên phương trình Lagrange trở thành: $L = T - V = T$. Phương trình động lực học của robot có thể được suy ra bằng cách sử dụng phương pháp đạo hàm theo phương trình Lagrange [54], [53]:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\mathbf{q}}_i} \right) - \frac{\partial L}{\partial \mathbf{q}_i} = \mathbf{Q}_i, \quad i = 1, 2, 3 \quad (1.10)$$

Trong đó: $\mathbf{\eta}_i$ - là tọa độ tổng quát thứ i ,

$\mathbf{Q}_i$ - là lực, moment tác động thứ i

Thay phương trình (1.9) vào (1.10) và thực hiện lấy đạo hàm ta được:

$$\begin{aligned} Q_1 = & \left[\tau_1 - r \operatorname{sgn}(\dot{\psi}_1) \zeta_1 \right] \left[-\frac{1}{r} (\cos \phi - \sin \phi) \right] + \left[\tau_2 - r \operatorname{sgn}(\dot{\psi}_2) \zeta_2 \right] \left[-\frac{1}{r} (\cos \phi + \sin \phi) \right] \\ & + \left[\tau_3 - r \operatorname{sgn}(\dot{\psi}_3) \zeta_3 \right] \left[\frac{1}{r} (\cos \phi - \sin \phi) \right] + \left[\tau_4 - r \operatorname{sgn}(\dot{\psi}_4) \zeta_4 \right] \left[\frac{1}{r} (\cos \phi + \sin \phi) \right] \end{aligned} \quad (1.11)$$

$$\begin{aligned} Q_2 = & \left[\tau_1 - r \operatorname{sgn}(\dot{\psi}_1) \zeta_1 \right] \left[-\frac{1}{r} (\sin \phi + \cos \phi) \right] + \left[\tau_2 - r \operatorname{sgn}(\dot{\psi}_2) \zeta_2 \right] \left[-\frac{1}{r} (\sin \phi - \cos \phi) \right] \\ & + \left[\tau_3 - r \operatorname{sgn}(\dot{\psi}_3) \zeta_3 \right] \left[\frac{1}{r} (\sin \phi + \cos \phi) \right] + \left[\tau_4 - r \operatorname{sgn}(\dot{\psi}_4) \zeta_4 \right] \left[\frac{1}{r} (\sin \phi - \cos \phi) \right] \end{aligned} \quad (1.12)$$

$$\begin{aligned} Q_3 = & (\tau_1 + \tau_2 + \tau_3 + \tau_4) \left[-\frac{\sqrt{2}}{r} l \sin(\pi / 4 - \alpha) \right] \\ & + \left[\operatorname{sgn}(\dot{\psi}_1) \zeta_1 + \operatorname{sgn}(\dot{\psi}_2) \zeta_2 + \operatorname{sgn}(\dot{\psi}_3) \zeta_3 + \operatorname{sgn}(\dot{\psi}_4) \zeta_4 \right] [\sqrt{2} l \sin(\pi / 4 - \alpha)] \end{aligned} \quad (1.13)$$

Với: ζ_i - là ma sát của bánh xe thứ i với mặt sàn

Từ (1.11), (1.12) và (1.13) ta đưa ra được phương trình động lực học viết dưới dạng ma trận như sau:

$$\mathbf{M}(\boldsymbol{\eta}) \ddot{\boldsymbol{\eta}} + \mathbf{C}(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}}) \dot{\boldsymbol{\eta}} + \mathbf{D} \boldsymbol{\delta} = \mathbf{D} \boldsymbol{\tau} \quad (1.14)$$

Trong đó: $\boldsymbol{\tau} = [\tau_1 \quad \tau_2 \quad \tau_3 \quad \tau_4]^T$; $\boldsymbol{\zeta} = [\zeta_1 \quad \zeta_2 \quad \zeta_3 \quad \zeta_4]^T$,

$$\boldsymbol{\delta} = r \cdot \mathbf{U} \cdot \boldsymbol{\zeta}; \quad \mathbf{U} = \operatorname{diag} [\operatorname{sgn}(\dot{\psi}_1) \quad \operatorname{sgn}(\dot{\psi}_2) \quad \operatorname{sgn}(\dot{\psi}_3) \quad \operatorname{sgn}(\dot{\psi}_4)]$$

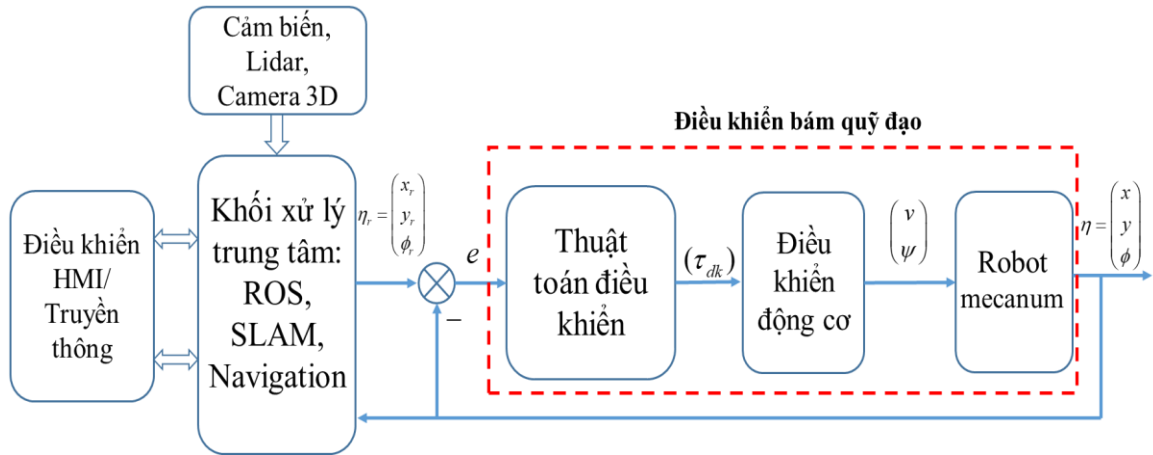
$$\begin{aligned}
\mathbf{M} &= \begin{bmatrix} m_1 & 0 & m_3 \\ 0 & m_1 & m_4 \\ m_3 & m_4 & m_2 \end{bmatrix}; m_1 = m_b + 4 \left(m_w + \frac{J_w}{r^2} \right); \\
m_3 &= m_b (d_1 \sin \phi + d_2 \cos \phi); m_4 = m_b (-d_1 \cos \phi + d_2 \sin \phi); \\
m_2 &= m_b (d_1^2 + d_2^2) + J_b + 8 \left(m_w + \frac{J_w}{r^2} \right) l^2 \sin^2(\pi/4 - \alpha) \\
\mathbf{C} &= \begin{bmatrix} 0 & 0 & c_1 \\ 0 & 0 & c_2 \\ 0 & 0 & 0 \end{bmatrix}; c_1 = m_b \dot{\phi} (d_1 \cos \phi - d_2 \sin \phi); c_2 = m_b \dot{\phi} (d_1 \sin \phi + d_2 \cos \phi); \\
\mathbf{D} &= \frac{1}{r} \begin{bmatrix} -(c-s) & -(s+c) & -\sqrt{2}l \sin(\frac{\pi}{4} - \alpha) \\ -(c+s) & -(s-c) & -\sqrt{2}l \sin(\frac{\pi}{4} - \alpha) \\ c-s & s+c & -\sqrt{2}l \sin(\frac{\pi}{4} - \alpha) \\ c+s & s-c & -\sqrt{2}l \sin(\frac{\pi}{4} - \alpha) \end{bmatrix}^T; s = \sin \phi, c = \cos \phi
\end{aligned}$$

Phương trình động lực học (1.14) của FMWR có các tham số d_1, d_2 thể hiện độ lệch của trọng tâm thực tế so với tâm hình học của robot. Các kết quả này được sử dụng để triển khai các thuật toán điều khiển, nhằm đánh giá ảnh hưởng của độ lệch trọng tâm đến chất lượng điều khiển bám quỹ đạo. Từ đó đề xuất các thuật toán điều khiển thích nghi để thích ứng với các thay đổi của tham số mô hình và nâng cao chất lượng điều khiển.

1.4. Cấu trúc hệ thống điều khiển robot di động

Cấu trúc chung của hệ thống điều khiển robot di động được mô tả trong hình 1.11 chia ra làm 4 khối chính: (1) Khối giao tiếp: Gồm các bảng khiển nút bấm, màn hình điều khiển người máy HMI, giao tiếp truyền thông, (2) Khối cảm biến: Gồm các tín hiệu cảm biến tiệm cận phát hiện va chạm, camera 3D thu thập hình ảnh môi trường làm việc, Lidar quét, thu thập dữ liệu bản đồ đưa tín hiệu về khối xử lý trung tâm. (3) Khối xử lý trung tâm là một máy tính nhúng có chức năng chạy hệ điều hành ROS và thực hiện các nhiệm vụ bản đồ hóa môi trường, lập quỹ đạo, điều hướng, xử lý các nhiệm vụ truyền thông. (4) Khối điều khiển bám quỹ đạo: thực hiện các thuật

toán điều khiển bám theo quỹ đạo đặt được gửi từ khối xử lý trung tâm, các thuật toán phải đảm bảo robot hoạt động chính xác theo quỹ đạo đặt với các sai lệch nhỏ và gửi các phản hồi về vị trí về cho bộ điều khiển trung tâm.



Hình 1.11 Sơ đồ tổng quan hệ thống điều khiển robot di động

Để robot hoạt động chính xác thì quá trình điều khiển chuyển động của robot được chia làm 3 giai đoạn chính như sau: Lập phương án chuyển động (Motion planning), thiết kế quỹ đạo di chuyển (Trajectory Generation), điều khiển bám quỹ đạo (Trajectory Tracking)

- *Giai đoạn lập phương án chuyển động*: Dựa vào môi trường làm việc của robot và các yêu cầu nhiệm vụ hoạt động của robot để thiết kế ra đường di chuyển của robot gồm điểm đầu, đích đến mà chưa quan tâm đến vận tốc, gia tốc khi di chuyển. Các phương án chuyển động được tính toán để tối ưu quãng đường di chuyển và tránh vật cản trong quá trình di chuyển của robot đảm bảo robot có thể di chuyển đến điểm đích. Trong giai đoạn này các thuật toán A*, EKF-SLAM, MPC [2, 3] hoặc thuật toán học máy [4, 5] được áp dụng vào để lập các phương án chuyển động cho robot được tối ưu.

- *Giai đoạn thiết kế quỹ đạo di chuyển*: Từ phương án chuyển động được lập ra thì giai đoạn này các đường quỹ đạo được thiết kế nhằm đảm bảo rằng robot có thể di chuyển một cách hiệu quả từ vị trí này đến vị trí khác. Quá trình thiết kế quỹ đạo bao gồm việc xác định các điều kiện ràng buộc về vị trí, vận tốc, gia tốc và thời gian di chuyển của mỗi phần của quỹ đạo sao cho robot có thể thực hiện di chuyển mượt mà và không gặp rủi ro. Tùy thuộc vào ứng dụng và môi trường làm việc của robot

mà các quỹ đạo di chuyển của robot có thể là đường thẳng, đường cong hay quỹ đạo bậc 3 (Cubic Trajectory) đòi hỏi mượt mà cao hơn.

- *Giai đoạn điều khiển bám quỹ đạo*: Đây là giai đoạn quan trọng quyết định đến độ chính xác và hiệu quả làm việc của robot. Từ quỹ đạo được xác định ở giai đoạn trước thì robot được điều khiển để di chuyển bám theo quỹ đạo được đặt ra với các sai lệch nhỏ nhất và đảm bảo thỏa mãn các điều kiện về vận tốc và gia tốc theo thời gian. Để điều khiển bám chính xác quỹ đạo thì các thuật toán điều khiển được áp dụng như bộ điều khiển truyền thống PID, PID thích nghi hay các bộ điều khiển trượt SMC [6-8], hay thời gian gần đây các thuật toán điều khiển dựa trên mạng nơ-ron nhân tạo, học máy được áp dụng [9-12]. Mặt khác, khi robot chuyển động phụ thuộc vào các yếu tố phức tạp như lực kéo, lực ma sát bánh xe, ma sát hệ thống dẫn động, lực cản, thay đổi khối lượng, trọng tâm và các tác động của môi trường làm ảnh hưởng đến chất lượng điều khiển bám quỹ đạo của robot. Từ đó đặt ra các thuật toán điều khiển bám quỹ đạo phải thích nghi được với các ảnh hưởng của các nhiễu tác động đến mô hình. Do đó luận án tập trung vào nghiên cứu, đề xuất thuật toán điều khiển bám quỹ đạo cho FMWR với sai lệch nhỏ và thích nghi được với các tham số của mô hình thay đổi đặc biệt quan tâm đến khối lượng và trọng tâm của robot thay đổi.

1.5. Tổng quan tình hình nghiên cứu trong và ngoài nước

1.5.1. Tình hình nghiên cứu trong nước

Tình hình nghiên cứu và ứng dụng robot di động tại Việt Nam ngày càng phát triển, với sự tham gia nghiên cứu của các trường đại học, viện nghiên cứu và các công ty công nghệ lớn điển hình như: Đại học Bách Khoa Hà Nội (HUST), Trường Đại học Công nghệ TP.HCM (HUTECH), Viện Khoa học và Công nghệ Việt Nam (VAST), Trường Đại học FPT, Trường Đại học Phenikaa, Toshiba Vietnam, Vinamilk... Các nghiên cứu và ứng dụng đang được triển khai ở nhiều lĩnh vực bao gồm công nghiệp, y tế, nông nghiệp, và giao thông. Các hướng nghiên cứu tập trung chủ yếu vào bài toán điều hướng, lập quỹ đạo và điều khiển bám quỹ đạo cho robot di động. Các đối tượng robot được áp dụng vào trong nghiên cứu là robot sử dụng bánh xe truyền thống dạng (non-holonomic), gần đây một số nghiên cứu về robot di động đa hướng (holonomic) được công bố. Một số nghiên cứu về bài toán điều khiển bám quỹ đạo được các tác giả và nhóm tác nghiên cứu công bố như sau:

Các tác giả trong nghiên cứu [7] đã trình bày phương pháp điều khiển robot di động AGV bám theo quỹ đạo với bộ điều khiển PID có tham số biến đổi theo thời gian, các tham số biến đổi được nhóm tác giả đề xuất lựa chọn thông qua sai số trung bình nhỏ nhất, nghiên cứu này tác giả chưa xét đến mô hình động lực học của robot. Trong nghiên cứu [13-17] tác giả đã phát triển một luật điều khiển bám thích nghi sử dụng mạng nơ-ron 2 lớp ẩn, 1 lớp đầu ra dùng để xấp xỉ các phần không biết trong mô hình hệ thống kết hợp với phương pháp điều khiển backstepping dựa trên mạng sóng Gaussian cho bài toán điều khiển bám quỹ đạo robot di động hai bánh xe chủ động, tác giả đã xét đến ảnh hưởng của trượt bánh xe. Nghiên cứu [18] tác giả đã trình bày một thuật toán điều khiển trượt kết hợp với kỹ thuật backstepping để thực hiện điều khiển bám quỹ đạo cho robot di động bánh xe (WMR), trong đó cấu trúc điều khiển sử dụng một mạch vòng và hai bộ điều khiển cho các hệ con. Nghiên cứu [19] đã nghiên cứu về điều khiển logic mờ cho robot bám quỹ đạo cho trước trong không gian làm việc. Nghiên cứu về bộ điều khiển mờ lai cho robot di động bám quỹ đạo cho robot hai bánh chủ động [20] với hai đầu vào và một đầu ra, khi trong hệ thống xuất hiện sai số thì bộ điều khiển mờ sẽ tự động điều chỉnh các thông số của bộ điều khiển PID để cho robot nhanh chóng tìm đúng quỹ đạo với sai lệch nhỏ nhất, nhưng vấn đề tác động của nhiễu lên hệ thống chưa được đề cập trong nghiên cứu.

Gần đây nghiên cứu về robot sử dụng bánh xe đa hướng được quan tâm nhiều hơn. Trong đó nghiên cứu [21] trình bày thuật toán DSC kết hợp mờ và mạng nơ-ron trong công bố [22] được áp dụng để nâng cao chất lượng bám quỹ đạo, công bố [23] điều khiển trượt theo hàm mũ, các nghiên cứu này áp dụng cho đối tượng robot bánh xe omni có mô hình bất động và chịu tác động của nhiễu. Trong nghiên cứu [24] đã sử dụng con quay hồi chuyển kỹ thuật số để khắc các ảnh hưởng của sai số mô hình robot đến khả năng bám quỹ đạo chạy thẳng của robot đa hướng, cùng với đó là tác giả so sánh khả năng di chuyển của bánh xe omni và bánh xe mecanum. Tuy nhiên các nghiên cứu này đang áp dụng cho đối tượng robot có khối lượng và trọng tâm không đổi.

Các cơ sở toán học và phương pháp nghiên cứu về mô hình động học, động lực học cho robot được trình bày trong các tài liệu [25], [26] và [27]. Cơ sở lý thuyết và

phương pháp thiết kế điều khiển các hệ thống phi tuyến cho robot được trình bày ở [28],[29] và điều khiển mờ [30].

Trong các nghiên cứu về bài toán điều khiển bám quỹ đạo cho robot di động tại Việt Nam sử dụng các thuật toán điều khiển truyền thống, gần đây là các thuật toán điều khiển thích nghi hay học máy cũng đang được nghiên cứu. Nhưng đa số các nghiên cứu thường chưa xét đến ảnh hưởng của độ lệch trọng tâm robot. Khi robot làm việc, trọng lượng của robot thay đổi dẫn đến trọng tâm sẽ thay đổi theo điều đó làm cho robot mất đi tính ổn định.

1.5.2. Tình hình nghiên cứu ngoài nước

Hiện nay trên thế giới, robot di động đa hướng được các nhà nghiên cứu ở các quốc gia như: Trung Quốc, Nhật Bản, Đức, Ấn Độ, Mỹ... quan tâm nhiều và có nhiều công bố liên quan. Trong các nghiên cứu về điều khiển bám quỹ đạo cho robot di động đa hướng thì các thuật toán điều khiển truyền thống hay gần đây là các thuật toán học máy, học tăng cường cũng được nghiên cứu áp dụng vào điều khiển bám quỹ đạo cho robot đạt được các kết quả tốt hơn.

Những nghiên cứu về robot đa hướng sử dụng bánh xe omni được thiết kế với ba bánh xe hoặc bốn bánh xe được thể hiện trong các nghiên cứu [31-36] trong đó các thuật toán điều khiển PID, điều khiển trượt SMC, Backstepping, hay kết hợp các luật điều khiển mờ nhằm nâng cao chất lượng điều khiển bám quỹ đạo. Các nghiên cứu về robot di động đa hướng sử dụng bánh xe mecanum [37-51] dựa trên phương trình Euler- Lagrange để mô hình hóa, xây dựng phương trình động học trong tọa độ đề các và phương trình động lực học. Các mô hình động học và động lực học trong nghiên cứu thường tương đồng, dựa trên nguyên lý cân bằng giữa động năng và thế năng của robot, hoặc theo phương trình cân bằng lực theo các định lý Newton. Tuy nhiên, điểm khác biệt nằm ở cách mô hình động lực học liên kết vận tốc của robot với vận tốc của từng bánh xe, điều này xuất phát từ sự khác biệt về cấu trúc phần cứng của robot được ước lượng bằng các tham số khác nhau. Nghiên cứu [52] đã đưa ra cấu tạo mô hình robot mecanum được sắp xếp bánh xe theo các cặp kiểu A hay kiểu B sẽ ảnh hưởng đến mô hình toán học của robot. Trong nghiên cứu [53] và [54] đã xây dựng mô hình toán học đầy đủ cho FMWR, có các tham số mô hình thể hiện trọng tâm thực tế của robot so với vị trí tâm hình học và ma sát bất định.

Những nghiên cứu về điều khiển bám quỹ đạo sử dụng các thuật toán điều khiển truyền thống như Fuzzy- PID, Backstepping hay điều khiển trượt thể hiện trong các công trình công bố [55-61] sử dụng hàm điều khiển thích nghi Lyapunov, thuật toán điều khiển trượt, hệ logic mờ hoặc mạng nơ-ron [62-65]. Thuật toán khiển dự báo mô hình phân tán (DMPC) và tối ưu hóa (PSO) được áp dụng trong nghiên cứu [66]. Các thuật toán điều khiển thông minh cho phép kết hợp luật điều khiển có các tham số thích nghi nhờ sử dụng mạng nơ-ron trong [6, 10, 67-71]. Các nghiên cứu này cho thấy tính hiệu quả của thuật toán góp phần nâng cao chất lượng điều khiển robot.

Hiện nay thuật toán học tăng cường (Reinforcement Learning- RL) là một nhánh quan trọng của học máy [72, 73] đã nhận được nhiều sự quan tâm trong kỹ thuật điều khiển nhờ khả năng giải quyết các bài toán điều khiển tối ưu, các thuật toán chủ yếu được sử dụng để giải phương trình Hamilton- Jacobi- Bellman (HJB). Trong nghiên cứu [74] một giải pháp cho bài toán tối ưu hệ thống phi tuyến bằng cách sử dụng cấu trúc actor- critic dựa trên mạng nơ-ron. Việc áp dụng chúng với mô hình động lực chưa biết rõ có thể được giải quyết thông qua cấu trúc học actor- critic- identifier trong các nghiên cứu [75-77]. Trong nghiên cứu [78] bài toán điều khiển bám quỹ đạo cho robot di động đa hướng bánh xe omni được chuyển thành bài toán điều khiển tối ưu theo thời gian thông qua thuật toán học tăng cường, áp dụng thuật toán học đồng bộ actor-critic online.

Trong nghiên cứu [79] một đối tượng robot di động đa hướng sử dụng 3 bánh xe omni được xét đến có trọng tâm của robot không trùng với tâm hình học, điều này làm cho chất lượng điều khiển bám quỹ đạo ảnh hưởng. Thuật toán điều khiển được thiết kế dựa trên hệ thống vòng kín bằng cách sử dụng hàm Lyapunov với đạo hàm thời gian bán xác định trên cơ sở lý thuyết về phương trình giới hạn. Tác giả trong nghiên cứu [80] đã đề xuất một bộ điều khiển truyền dẫn cho xe đẩy đa hướng lệch trọng tâm. Một khối lượng, quán tính ảo được xác định bằng các tham số động như khối lượng, hệ số giảm chấn và vị trí, kết quả của nghiên cứu được thử nghiệm trên mô hình nguyên mẫu cho kết quả khả quan.

Các nghiên cứu hiện nay trên thế giới đã áp dụng đa dạng các thuật toán điều khiển, từ các phương pháp truyền thống đến các kỹ thuật điều khiển thích nghi và học máy, nhằm giải quyết bài toán điều khiển bám quỹ đạo cho robot di động đa hướng. Trong đó các vấn đề về nhiễu và các thay đổi của môi trường làm việc tác động lên

robot cũng được xét đến. Tuy nhiên, vấn đề về ảnh hưởng của thay đổi khối lượng cũng như độ lệch trọng tâm của robot chưa có nhiều nghiên cứu đề cập đến. Với các phân tích như trên thì nghiên cứu ảnh hưởng của độ lệch trọng tâm đến việc điều khiển bám quỹ đạo của robot di động đa hướng là cần thiết. Từ đó luận án nghiên cứu đối tượng robot di động đa hướng mecanum, áp dụng các thuật toán điều khiển thích nghi được các thay đổi của tham số mô hình robot và ảnh hưởng của môi trường làm việc, để đạt được chất lượng điều khiển tốt hơn.

1.6. Kết luận chương 1

Trong lĩnh vực nghiên cứu về robot di động thì robot di động đa hướng mecanum đang là đối tượng được quan tâm nhiều, nhờ vào khả năng di chuyển linh hoạt đa hướng, đáp ứng một số yêu cầu hoạt động trong không gian hạn chế, điều này các robot truyền thống không thể đáp ứng được. Bên cạnh đó, để FMWR hoạt động ổn định thì yêu cầu các thuật toán điều khiển cần đạt được độ chính xác bám quỹ đạo với sai lệch nhỏ, đáp ứng về mặt thời gian có khả năng thích ứng với các nhiễu và thành phần bất định của mô hình robot. Trong quá trình hoạt động, FMWR có thể mang thêm một khối lượng hàng hóa, cho nên trọng lượng của robot thay đổi dẫn đến trọng tâm của robot thay đổi khác với tâm hình học của robot. Điều đó ảnh hưởng trực tiếp đến độ ổn định và khả năng di chuyển cũng như chất lượng bám quỹ đạo của robot. Bên cạnh đó, bài toán điều khiển bám quỹ đạo cho mobile robot là bài toán rất quan trọng vì nó quyết định đến khả năng di chuyển chính xác và an toàn của robot trong môi trường thực tế. Việc điều khiển robot di chuyển theo một quỹ đạo đã định trước không chỉ liên quan đến độ chính xác trong việc định vị mà còn ảnh hưởng đến khả năng tránh vật cản, tối ưu hóa năng lượng, và thực hiện nhiệm vụ hiệu quả trong các ứng dụng như giao hàng tự động, tuần tra, hoặc dẫn đường trong kho bãi. Bài toán này thường đối mặt với các thách thức như: Tác động của nhiễu từ môi trường (ví dụ: địa hình gồ ghề, cảm biến sai số), đặc tính phi tuyến của hệ thống robot, và sự thay đổi động học theo thời gian thực. Do vậy, luận án sẽ tập trung nghiên cứu đề xuất các thuật toán điều khiển thích nghi và học tăng cường nhằm nâng cao chất lượng điều khiển bám quỹ đạo cho FMWR có trọng tâm thay đổi, giúp cho robot hoạt động chính xác và ổn định hơn.

CHƯƠNG 2. THUẬT TOÁN ĐIỀU KHIỂN BÁM QUỸ ĐẠO CHO ROBOT DI ĐỘNG ĐA HƯỚNG MECANUM

Sau khi xây dựng mô hình động học và động lực học cho FMWR, luận án sẽ áp dụng một số thuật toán điều khiển bám quỹ đạo để đánh giá mô hình robot. Đồng thời, việc phân tích các ưu và nhược điểm của các thuật toán điều khiển từ đó đề xuất thuật toán điều khiển mới giúp nâng cao tính hiệu quả.

2.1. Bộ điều khiển PID điều khiển bám quỹ đạo cho FMWR

Bộ điều khiển PID (Proportional-Integral-Derivative) là một bộ điều khiển phản hồi vòng kín, hoạt động bằng cách tính toán sai số giữa giá trị thực tế và giá trị mục tiêu. Bộ điều khiển này điều chỉnh đầu vào để giảm thiểu sai số một cách hiệu quả. Các nghiên cứu [64, 81-83] đã áp dụng bộ điều khiển PID cho FMWR, dựa trên mô hình động học của FMWR.

Từ phương trình động học của FMWR trong (1.6):

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \frac{1}{2} r \mathbf{J}^+ \begin{bmatrix} \dot{\psi}_1 \\ \dot{\psi}_2 \\ \dot{\psi}_3 \\ \dot{\psi}_4 \end{bmatrix}$$

Xét quỹ đạo đặt cho robot như sau: $\boldsymbol{\eta}_d = [x_d \quad y_d \quad \phi_d]^T$

Sai lệch của hệ thống:

$$\mathbf{e} = \boldsymbol{\eta} - \boldsymbol{\eta}_d \quad (2.1)$$

Đạo hàm sai lệch theo thời gian:

$$\dot{\mathbf{e}} = \dot{\boldsymbol{\eta}} - \dot{\boldsymbol{\eta}}_d \quad (2.2)$$

Thay phương trình (1.6) vào (2.2) ta được:

$$\dot{\mathbf{e}} = \frac{1}{2} r \mathbf{J}^+ \begin{bmatrix} \dot{\psi}_1 \\ \dot{\psi}_2 \\ \dot{\psi}_3 \\ \dot{\psi}_4 \end{bmatrix} - \begin{bmatrix} \dot{x}_d \\ \dot{y}_d \\ \dot{\phi}_d \end{bmatrix} \quad (2.3)$$

Vận tốc góc của từng bánh xe được tính toán để đảm bảo sự ổn định của hệ thống điều khiển vòng kín, giúp duy trì khả năng điều khiển chính xác và ổn định trong suốt quá trình vận hành của robot:

$$\begin{bmatrix} \dot{\psi}_1 \\ \dot{\psi}_2 \\ \dot{\psi}_3 \\ \dot{\psi}_4 \end{bmatrix} = \frac{1}{2} r(\mathbf{J}^+)^{-1} \left(-\mathbf{K}_p \begin{bmatrix} x_e \\ y_e \\ \phi_e \end{bmatrix} - \mathbf{K}_I \begin{bmatrix} \int_0^t x_e dt \\ \int_0^t y_e dt \\ \int_0^t \phi_e dt \end{bmatrix} \right) \quad (2.4)$$

Trong đó: $\mathbf{K}_p, \mathbf{K}_I$ là các ma trận hệ số điều khiển, có đặc tính đường chéo xác định dương.

Thông thường, bộ điều khiển PID được áp dụng dựa trên phương trình động học, điều này chưa tính đến các yếu tố thay đổi như trọng tâm hay khối lượng của robot. Mặc dù mô hình FMWR thực tế là phi tuyến và có tham số bất định, việc xác định chính xác các tham số bộ điều khiển trở nên khó khăn. Để khắc phục điều này, một số nghiên cứu [49], [84] đã tích hợp bộ chỉnh định mờ nhằm tối ưu hóa việc chọn lựa các tham số của bộ điều khiển. Tuy nhiên, bộ điều khiển PID thường hoạt động tốt với các hệ thống tuyến tính, nhưng gặp khó khăn khi đối mặt với các yếu tố phi tuyến bất định của mô hình robot. Hiện nay, các bộ điều khiển phi tuyến đang được nghiên cứu và phát triển mạnh mẽ để vượt qua những hạn chế của các phương pháp điều khiển truyền thống. Trong đó, phương pháp điều khiển trượt có khả năng xử lý các hệ thống phi tuyến và độ không chắc chắn của mô hình, giúp cải thiện hiệu suất điều khiển trong các tình huống thay đổi và phức tạp.

2.2. Bộ điều khiển trượt cơ bản cho FMWR

Thuật toán điều khiển trượt (SMC - Sliding Mode Control) là một thuật toán điều khiển phi tuyến [60], [61] được áp dụng để điều khiển các hệ thống có mô hình động lực học phức tạp và không chắc chắn. Trong thuật toán này, các thành phần điều khiển sử dụng các thông số động lực học của mô hình FMWR, đồng thời xem xét các giới hạn của các yếu tố bất định. Mô hình động lực học của FMWR có xét đến độ

lệch trọng tâm và khối lượng thay đổi, được sử dụng giúp đánh giá độ chính xác và hiệu quả của thuật toán điều khiển trong các tình huống thực tế.

Các bước thiết kế thuật toán điều khiển:

$$\text{Chọn các biến:} \quad \begin{cases} \mathbf{x}_1 = \boldsymbol{\eta} \\ \mathbf{x}_2 = \dot{\mathbf{x}}_1 \\ \dot{\mathbf{x}}_2 = \ddot{\boldsymbol{\eta}} \end{cases} \quad (2.5)$$

Sai lệch của hệ thống được xác định: $\mathbf{e} = \mathbf{x}_1 - \mathbf{x}_{1d}$

Mặt trượt thứ nhất được xác định như sau:

$$\mathbf{S} = \Delta \mathbf{e} + \dot{\mathbf{e}} \quad (2.6)$$

Trong đó hệ số mặt trượt: $\Delta > 0$

Lấy đạo hàm của phương trình (2.6) ta được:

$$\dot{\mathbf{S}} = \Delta \dot{\mathbf{e}} + \ddot{\mathbf{e}} \quad (2.7)$$

Thay phương trình động lực học (1.14) vào (2.7) ta được:

$$\dot{\mathbf{S}} = \Delta \dot{\mathbf{e}} + \mathbf{M}^{-1}[\mathbf{D}\boldsymbol{\tau} - \mathbf{D}\boldsymbol{\delta} - \mathbf{C}\mathbf{x}_1] - \ddot{\mathbf{x}}_{1d} \quad (2.8)$$

Chọn hàm Lyapunov:

$$V = \frac{1}{2} \mathbf{S}^T \mathbf{S} \quad (2.9)$$

Lấy đạo hàm của (2.9) theo thời gian được:

$$\dot{V} = \mathbf{S}^T \{ \Delta \dot{\mathbf{e}} + \mathbf{M}^{-1}[\mathbf{D}\boldsymbol{\tau} - \mathbf{D}\boldsymbol{\delta} - \mathbf{C}\dot{\mathbf{x}}_1] - \ddot{\mathbf{x}}_{1d} \} \quad (2.10)$$

Lựa chọn tín hiệu điều khiển cho hệ thống có dạng:

$$\boldsymbol{\tau} = -\mathbf{D}^T (\mathbf{D}\mathbf{D}^T)^{-1} [\mathbf{M}(\Delta \dot{\mathbf{e}} - \ddot{\mathbf{x}}_{1d} + k_1 \text{sgn}(\mathbf{S})) - \mathbf{C}\dot{\mathbf{x}}_1 - \mathbf{D}\boldsymbol{\delta}] \quad (2.11)$$

Thay tín hiệu điều khiển vào (2.10) ta thu được đạo hàm của hàm Lyapunov:

$$\dot{V} = -k_1 \mathbf{S}^T \text{sgn}(\mathbf{S}) \leq 0$$

với k_1 là hằng số dương thì hệ ổn định theo tiêu chuẩn Lyapunov.

Nhận xét: Thuật toán điều khiển trượt (SMC) có ưu điểm nổi bật là ít nhạy với sự biến đổi của các tham số mô hình cũng như ảnh hưởng của nhiễu bên ngoài. Tuy nhiên, một hạn chế đáng kể của phương pháp này là các tham số điều khiển có dạng hàm dấu, điều này dẫn đến hiện tượng rung "chattering". Hiện tượng này có thể gây ra sự dao động mạnh trong hệ thống, làm giảm hiệu suất điều khiển và thậm chí có thể ảnh hưởng đến sự ổn định của hệ thống nếu không được kiểm soát đúng cách.

2.3. Thuật toán điều khiển Backstepping - SMC

2.3.1. Thiết kế thuật toán

Thuật toán được xây dựng dựa trên sự kết hợp giữa phương pháp Backstepping và điều khiển trượt. Ngoài ra, luật điều khiển Backstepping phi tuyến cũng được triển khai cho các mô hình động học và động lực học, đã được kiểm chứng trong các nghiên cứu trước đây [85], [86]. Tuy nhiên, sự tác động của nhiễu có thể khiến bộ điều khiển Backstepping mất dần tính ổn định, đặc biệt là đối với các hệ thống bậc cao, nơi nhiễu có ảnh hưởng mạnh mẽ hơn. Trong khi đó, thuật toán điều khiển trượt (SMC) có khả năng giúp hệ thống duy trì ổn định và đối phó hiệu quả với các nhiễu cũng như sự thay đổi tham số mô hình. Do vậy, việc kết hợp Backstepping với SMC sẽ giúp đảm bảo sự ổn định của hệ thống và nâng cao độ chính xác trong việc bám sát quỹ đạo. Thuật toán điều khiển này được xây dựng trên cả phương trình động học và động lực học của FMWR, đồng thời xem xét sự thay đổi trọng tâm của hệ thống.

Quy trình thiết kế thuật toán điều khiển bao gồm các bước sau:

Từ phương trình động học (1.7) và phương trình động lực học (1.14) của FMWR, ta thực hiện các phép biến đổi như sau:

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{H}\mathbf{v} \\ \dot{\mathbf{v}} = \mathbf{M}^{-1}(\mathbf{D}\boldsymbol{\tau} - \mathbf{D}\delta - \mathbf{C}\dot{\boldsymbol{\eta}}) \end{cases} \quad (2.12)$$

Với các biến trong đó:

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{x} & \dot{y} & \dot{\phi} \end{bmatrix}^T$$

$$\mathbf{v} = \boldsymbol{\eta}_R = \begin{bmatrix} \dot{x}_R & \dot{y}_R & \dot{\phi}_R \end{bmatrix}^T$$

Xét quỹ đạo $\mathbf{x}$ và quỹ đạo tham chiếu $\mathbf{x}_r$, ta có véc tơ sai lệch:

$$\mathbf{e} = \mathbf{x} - \mathbf{x}_r \quad (2.13)$$

Lựa chọn tín hiệu điều khiển ảo được thực hiện như sau:

$$\mathbf{v}_r = -\mathbf{H}^{-1}(c_1\mathbf{e} - \dot{\mathbf{x}}_r) \quad (2.14)$$

Trong đó: Ma trận $c_1 = \text{diag}(c_{1x}, c_{1y}, c_{1\phi})$ có đường chéo xác định dương

Với tín hiệu điều khiển ảo đã được lựa chọn, ta xác định hàm Lyapunov theo cách sau:

$$V_1 = \frac{1}{2} \mathbf{e}^T \mathbf{e} \quad (2.15)$$

Đạo hàm phương trình (2.15) thu được:

$$\begin{aligned} \dot{V}_1 &= \mathbf{e}^T \dot{\mathbf{e}} = \mathbf{e}^T (\dot{\mathbf{x}} - \dot{\mathbf{x}}_r) \\ &= \mathbf{e}^T (\mathbf{H}\mathbf{v} - \mathbf{H}\mathbf{v}_r - c_1\mathbf{e}) \end{aligned} \quad (2.16)$$

Thay $\mathbf{v}$ bởi $\mathbf{v}_r$ ta được:

$$\dot{V}_1 = -\mathbf{e}^T c_1 \mathbf{e} \leq 0 \quad (2.17)$$

Với đạo hàm của hàm Lyapunov $\dot{V}_1 \leq 0$, tín hiệu điều khiển ảo được lựa chọn đảm bảo tính ổn định của hệ thống.

Thiết kế tín hiệu điều khiển đầu vào cho hệ thống được thực hiện, với sai lệch Backstepping được xác định như sau:

$$\begin{aligned} \mathbf{v}_e &= \mathbf{v} - \mathbf{v}_r = \mathbf{H}^{-1}\dot{\mathbf{x}} + \mathbf{H}^{-1}(c_1\mathbf{e} - \dot{\mathbf{x}}_r) \\ &= \mathbf{H}^{-1}c_1\mathbf{e} + \mathbf{H}^{-1}\dot{\mathbf{e}} \end{aligned} \quad (2.18)$$

Xác định mặt trượt:

$$\mathbf{S} = \Delta\mathbf{e} + \mathbf{H}\mathbf{v}_e = (\Delta + c_1)\mathbf{e} + c_1\dot{\mathbf{e}} \quad (2.19)$$

Trong đó: ma trận hệ số $\Delta = \text{diag}(\Delta_1, \Delta_2, \Delta_3)$ là ma trận có các phần tử trên đường chéo đều xác định dương.

Đạo hàm phương trình (2.19) ta được:

$$\begin{aligned}\dot{\mathbf{S}} &= \Delta\dot{\mathbf{e}} + \dot{\mathbf{H}}\mathbf{v}_e + \mathbf{H}\dot{\mathbf{v}}_e \\ &= \Delta\dot{\mathbf{e}} + \dot{\mathbf{H}}\mathbf{v}_e + \mathbf{H}(\mathbf{M}^{-1}(\mathbf{D}\boldsymbol{\tau} - \mathbf{D}\delta - \mathbf{C}\dot{\boldsymbol{\eta}}) - \dot{\mathbf{v}}_r)\end{aligned}\quad (2.20)$$

Kiểm tra tính ổn định của hệ thống được thực hiện bằng cách lựa chọn hàm Lyapunov thứ hai theo công thức sau:

$$V_2 = \frac{1}{2} \mathbf{S}^T \mathbf{S} \quad (2.21)$$

Đạo hàm (2.21) ta được:

$$\dot{V}_2 = \mathbf{S}^T \dot{\mathbf{S}} \quad (2.22)$$

Giả sử $\dot{\mathbf{S}} = 0$, khi đó tín hiệu điều khiển bao gồm:

$$\boldsymbol{\tau} = \boldsymbol{\tau}_{eq} - \boldsymbol{\tau}_{sw} \quad (2.23)$$

Trong đó: - $\boldsymbol{\tau}_{eq}$ là tín hiệu giữ cho hệ thống có trạng thái ổn định trên mặt trượt

- $\boldsymbol{\tau}_{sw}$ tín hiệu này là tín hiệu điều khiển giúp hệ thống tiến tới và duy trì trạng thái trên mặt trượt.

Để cho sai lệch hệ thống $\mathbf{e} \rightarrow 0$ ta chọn các tín hiệu:

$$\boldsymbol{\tau}_{eq} = -\mathbf{D}^T (\mathbf{D}\mathbf{D}^T)^{-1} (\mathbf{M}(\mathbf{H}^{-1}(\Delta\dot{\mathbf{e}} + \dot{\mathbf{H}}\mathbf{v}_e + \mathbf{H}\dot{\mathbf{v}}_e) - \mathbf{C}\dot{\boldsymbol{\eta}} - \mathbf{D}\delta)) \quad (2.24)$$

$$\boldsymbol{\tau}_{sw} = -\mathbf{D}^T (\mathbf{D}\mathbf{D}^T)^{-1} \mathbf{M}\mathbf{H}^{-1} (c_2 \operatorname{sgn}(\mathbf{S}) + c_3 \mathbf{S}) \quad (2.25)$$

Với $c_2 = \operatorname{diag}(c_{2x}, c_{2y}, c_{2\theta})$ và $c_3 = \operatorname{diag}(c_{3x}, c_{3y}, c_{3\theta})$ các ma trận hệ số này có các phần tử trên đường chéo được xác định dương.

Đạo hàm của hàm Lyapunov ta được:

$$\dot{V}_2 = -\mathbf{S}^T (c_2 \operatorname{sgn}(\mathbf{S}) + c_3 \mathbf{S}) \leq 0 \quad (2.26)$$

Đạo hàm của hàm Lyapunov âm thỏa mãn điều kiện ổn định, điều đó dẫn đến mặt trượt hội tụ về 0, sai lệch quỹ đạo $\mathbf{e}$ và sai lệch Backstepping $\mathbf{v}_e$ cũng tiến về 0 hệ thống ổn định.

2.3.2. Mô phỏng thuật toán

Áp dụng phần mềm Matlab/ Simulink để mô phỏng thuật toán, trong đó thêm nhiễu ngoại vào tín hiệu điều khiển nhằm kiểm tra tính ổn định của hệ thống. Đồng thời, so sánh hiệu quả của bộ điều khiển đề xuất với bộ điều khiển PID thông thường để đánh giá chất lượng và hiệu suất hoạt động của thuật toán mới.

Các tham số của mô hình robot được lựa chọn như sau:

$$\begin{aligned} m &= 30(kg); m_w = 0.9(kg); \\ J_b &= 5(kg.m^2); J_w = 0.1(kg.m^2); g = 0.2(m), h = 0.3(m); r = 0.075(m); \\ g &= 9.8(m/s^2); f = [0.05 \quad 0.05 \quad 0.05 \quad 0.05]^T (N) \end{aligned} \quad (2.27)$$

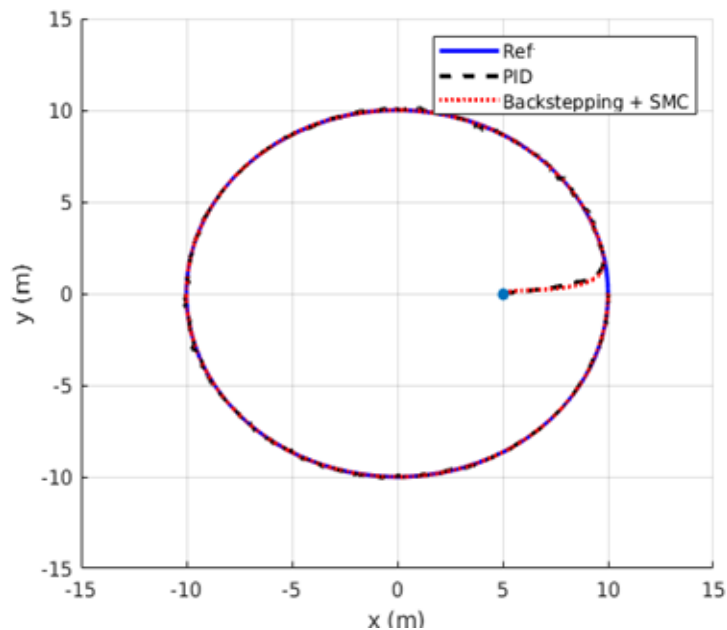
Các tham số của bộ điều khiển được lựa chọn:

$$\Delta = \text{diag}(10, 10, 10); c_1 = c_2 = c_3 = \text{diag}(20, 20, 20); K_p = 12; K_i = 15; K_d = 1$$

Phương trình quỹ đạo đặt được xác định:
$$\begin{cases} x_r = 10 \cos \omega t \\ y_r = 10 \sin \omega t \\ \phi_r = \beta \end{cases}$$

Vị trí đặt ban đầu của robot: $x = 5m, y = 0, \phi = -\frac{\pi}{4} \text{ rad}$

Kết quả mô phỏng:

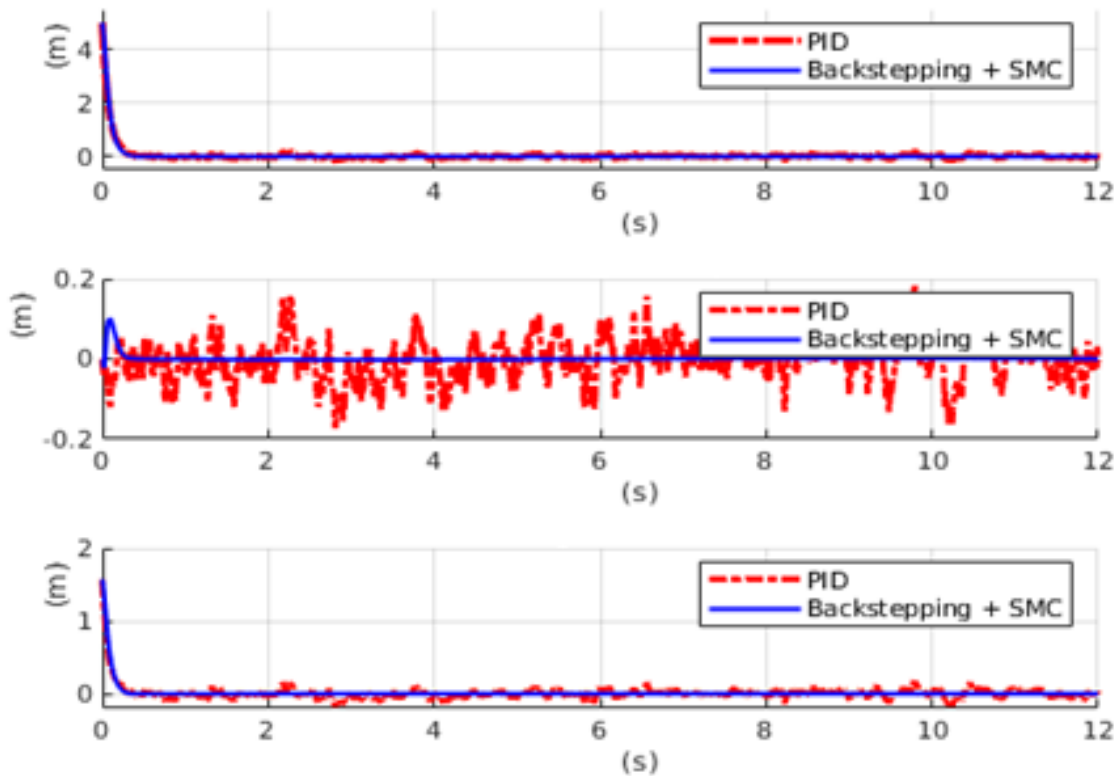


Hình 2.1 Quỹ đạo di chuyển của FMWR

Tiến hành mô phỏng đồng thời thuật toán điều khiển Backstepping kết hợp SMC với bộ điều khiển PID truyền thống nhằm đánh giá hiệu quả và chất lượng của bộ điều khiển được đề xuất. Thời gian mô phỏng $T=12s$.

Kết quả mô phỏng trong hình 2.1 cho thấy khi bắt đầu mô phỏng, robot di chuyển tiến đến đường quỹ đạo đặt và bắt đầu bám theo quỹ đạo đường tròn trong thời gian ngắn, sau đó robot tiếp tục bám theo quỹ đạo đặt và hoạt động ổn định.

Các kết quả trong hình 2.2 cho thấy robot sử dụng bộ điều khiển PID có tốc độ bám quỹ đạo chậm hơn so với bộ điều khiển Backstepping kết hợp SMC được đề xuất. Trong đó sai lệch theo các trục x, y đang dần hội tụ về 0 với sai lệch tĩnh nhỏ.



Hình 2.2 Sai lệch quỹ đạo của FMWR

Khi nhiễu được thêm vào hệ thống, bộ điều khiển PID gặp phải hiện tượng dao động và dần mất ổn định trên quỹ đạo, điều này được thể hiện rõ trong kết quả ở hình 2.2, Trong khi đó bộ điều khiển Backstepping + SMC vẫn giúp robot hoạt động ổn định và giảm khả năng tác động của nhiễu. Từ đó cho thấy bộ điều khiển Backstepping + SMC không chỉ có khả năng điều khiển robot bám quỹ đạo chính xác hơn, mà còn duy trì sự ổn định tốt hơn khi có tác động của nhiễu.

2.4. Thuật toán điều khiển mặt trượt động cho FMWR

Thuật toán điều khiển mặt trượt động (DSC – Dynamic Surface Control) được phát triển dựa trên phương pháp điều khiển đa mặt trượt và kỹ thuật Backstepping, giúp khắc phục các nhược điểm liên quan đến các thành phần bất định ảnh hưởng đến mô hình. Đồng thời, thuật toán này giảm thiểu hiện tượng rung "chattering" đáng kể nhờ vào việc sử dụng bộ lọc thông thấp. Thuật toán điều khiển DSC, như được trình bày trong các nghiên cứu [87], [88], [89] bao gồm hai thành phần chính: khối đa mặt trượt (MSS) và khối lọc thông thấp. Khối đa mặt trượt tiếp nhận các giá trị từ trạng thái hiện tại của hệ thống, cùng với các tín hiệu điều khiển đã được xử lý qua bộ lọc thông thấp. Tín hiệu lọc này giúp giảm nhiễu và làm mịn tín hiệu điều khiển, giúp hệ thống tránh các hiện tượng rung. Mỗi bề mặt trượt tương ứng với một trạng thái điều khiển cụ thể của hệ thống. Bộ lọc bộ lọc thông thấp được áp dụng để xử lý các đầu vào từ hệ thống, giúp tách biệt các tín hiệu tần số cao không mong muốn. Các giá trị đầu ra từ bộ lọc được sử dụng để tạo ra các tín hiệu điều khiển phù hợp. Những giá trị này không chỉ làm đầu vào cho quá trình điều khiển hệ thống mà còn được đưa trở lại khối đa mặt trượt, giúp cập nhật và điều chỉnh liên tục tín hiệu điều khiển. Quá trình này giúp hệ thống tự động điều chỉnh và thích nghi với các biến động trong thời gian thực, đảm bảo hiệu suất điều khiển ổn định và chính xác.

Các bước thiết kế thuật toán điều khiển DSC được thực hiện như sau [88]:

$$\text{Đặt biến trạng thái: } \begin{cases} \mathbf{x}_1 = \boldsymbol{\eta} \\ \mathbf{x}_2 = \dot{\mathbf{x}}_1 \\ \dot{\mathbf{x}}_2 = \ddot{\boldsymbol{\eta}} \end{cases}$$

Trong đó: $\mathbf{x}_1$ - biến trạng thái của vị trí robot

$\mathbf{x}_{1d}$ - trạng thái vị trí của quỹ đạo đặt

Xác định mặt trượt thứ nhất:

$$\mathbf{S}_1 = \mathbf{x}_1 - \mathbf{x}_{1d} \quad (2.28)$$

Lấy đạo hàm mặt trượt $\mathbf{S}_1$ ta được:

$$\dot{\mathbf{S}}_1 = \dot{\mathbf{x}}_1 - \dot{\mathbf{x}}_{1d} = \mathbf{x}_2 - \dot{\mathbf{x}}_{1d} \quad (2.29)$$

Chọn biến trạng thái $\bar{\mathbf{x}}_2$ được xác định:

$$\bar{\mathbf{x}}_2 = -K_1 \mathbf{S}_1 + \dot{\mathbf{x}}_{1d} \quad (2.30)$$

Xác định bộ lọc là một khâu quán tính bậc nhất, trong đó tín hiệu điều khiển ảo $\mathbf{x}_{2d}$ sẽ bám theo sát tín hiệu điều khiển thực tế $\bar{\mathbf{x}}_2(0) = \mathbf{x}_{2d}(0)$

$$\bar{\mathbf{x}}_2 = \tau \dot{\mathbf{x}}_{2d} + \mathbf{x}_{2d} \quad (2.31)$$

Xác định mặt trượt thứ hai:

$$\mathbf{S}_2 = \mathbf{x}_2 - \mathbf{x}_{2d} \quad (2.32)$$

Đạo hàm mặt trượt S_2 ta được:

$$\dot{\mathbf{S}}_2 = -\dot{\mathbf{x}}_{2d} + \mathbf{u} + f(\mathbf{x}) \quad (2.33)$$

$$\text{Đặt: } \begin{cases} \mathbf{u} = \mathbf{M}^{-1} \mathbf{D} \tau \\ f(\mathbf{x}) = -\mathbf{M}^{-1} (\mathbf{C} \dot{\mathbf{q}} - \mathbf{D} \delta) \end{cases} \quad (2.34)$$

Kết hợp với phương trình (2.31) ta xác định hàm điều khiển như sau:

$$\mathbf{u} = \frac{\bar{\mathbf{x}}_2 - \mathbf{x}_{2d}}{\tau} - f(\mathbf{x}) - K_2 \mathbf{S}_2 \quad (2.35)$$

Dựa trên đó, tín hiệu điều khiển với cấu trúc của thuật toán DSC đã được thiết kế sẽ được xác định:

$$\tau = -\mathbf{D}^T (\mathbf{D} \mathbf{D}^T)^{-1} [\mathbf{M} (-\frac{\bar{\mathbf{x}}_2 - \mathbf{x}_{2d}}{\tau} + K_2 \mathbf{S}_2) - \mathbf{C} \dot{\mathbf{q}} - \mathbf{D} \delta] \quad (2.36)$$

Bộ điều khiển với mặt trượt động, trong đó trạng thái $\mathbf{x}_1$ và vận tốc $\mathbf{x}_2$ được phản hồi. Kết hợp mặt trượt $\mathbf{S}_1$ và mặt trượt $\mathbf{S}_2$, ta xác định:

$$\dot{\mathbf{S}}_1 = -K_1 \mathbf{S}_1 + \mathbf{S}_2 \quad (2.37)$$

$$\dot{\mathbf{S}}_2 = -K_2 \mathbf{S}_2 \quad (2.38)$$

Chọn một hàm Lyapunov phù hợp để phân tích tính ổn định của hệ thống

$$V = \frac{\mathbf{S}_1^T \mathbf{S}_1 + \mathbf{S}_2^T \mathbf{S}_2}{2} \geq 0 \quad (2.39)$$

Đạo hàm của hàm Lyapunov (2.39) ta được:

$$\dot{V} = \mathbf{S}_1^T \dot{\mathbf{S}}_1 + \mathbf{S}_2^T \dot{\mathbf{S}}_2 \quad (2.40)$$

Thay công thức (2.37) và công thức (2.38) vào trong công thức (2.40) ta biến đổi:

$$\dot{V} = -K_1 \mathbf{S}_1^T \mathbf{S}_1 - K_2 \mathbf{S}_2^T \mathbf{S}_2 + \mathbf{S}_1^T \mathbf{S}_2 \quad (2.41)$$

Sử dụng bất đẳng thức sau đây để phân tích và chứng minh tính ổn định của hệ thống:

$$\frac{\mathbf{S}_1^T \mathbf{S}_1 + \mathbf{S}_2^T \mathbf{S}_2}{2} \geq \mathbf{S}_1^T \mathbf{S}_2 \quad (2.42)$$

Thay vào công thức (2.41) ta thu được kết quả như sau:

$$\dot{V} \leq -\left(K_1 - \frac{1}{2}I\right) \mathbf{S}_1^T \mathbf{S}_1 - \left(K_2 - \frac{1}{2}I\right) \mathbf{S}_2^T \mathbf{S}_2 \quad (2.43)$$

Trong đó K_1 và K_2 là các tham số điều khiển được thiết kế sao cho hàm $\dot{V}$ xác định âm. Do đó, nếu $K_1 > \frac{1}{2}I$ và $K_2 > \frac{1}{2}I$ thì, $\dot{V} < 0$ đối với mọi $\mathbf{S}_1$ và $\mathbf{S}_2$. Do đó, $\mathbf{S}_1 \rightarrow 0$ và $\mathbf{S}_2 \rightarrow 0$ khi thời gian $t \rightarrow \infty$, ổn định tiệm cận của hệ thống theo tiêu chuẩn ổn định Lyapunov. Đặc biệt, nếu các sai lệch khác không, phương pháp điều khiển DSC sẽ làm giảm năng lượng của hệ thống ($\dot{V} < 0$), do đó điều chỉnh các sai lệch. Hệ thống nếu $\mathbf{x}_1$ và $\mathbf{x}_2$ bám theo quỹ đạo mong muốn $\mathbf{x}_{1d}$ và $\mathbf{x}_{2d}$, tương ứng khi thời gian $t \rightarrow \infty$.

2.5. Mô phỏng thuật toán PID, SMC và DSC trên Matlab/Simulink

Mô phỏng các thuật toán đã thiết kế với cùng bộ tham số của mô hình robot để so sánh đánh giá kết quả. Các tham số mô phỏng được chọn như sau:

Tham số của mô hình FMWR được lựa chọn như trong (2.27):

Các tham số của bộ điều khiển:

- Thuật toán PID: $K_I = [15, 15, 15]$; $K_P = [20, 20, 20]$; $K_D = 1$

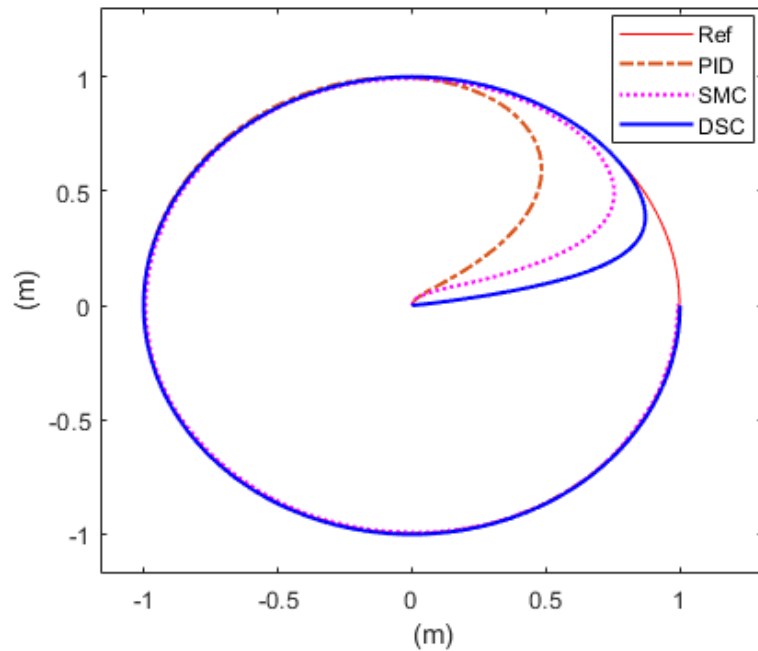
- Thuật toán SMC: $\Delta=[5,5,5]; K_1=[10,10,10]$

- Thuật toán DSC: $K_1=[5,5,5]; K_2=[10,10,10];$

$$\text{Quỹ đạo đặt đường tròn: } \begin{cases} x_r = \cos \omega t \\ y_r = \sin \omega t \text{ với } \omega = \frac{2\pi}{15} \\ \phi_r = \frac{\pi}{4} \end{cases}$$

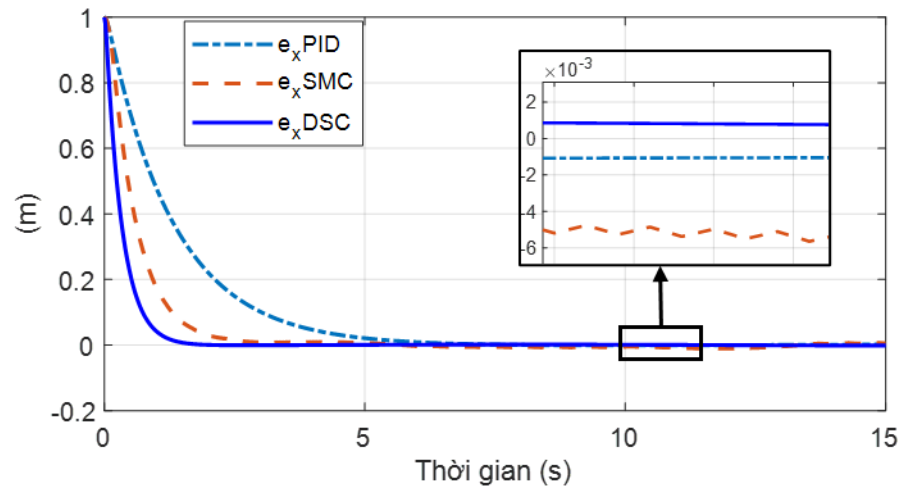
- **Kịch bản 1:** Mô phỏng với mô hình FMWR có trọng tâm không đổi

Thực hiện mô phỏng trong thời gian $T=15s$. Kết quả mô phỏng của các bộ điều khiển thu được như sau:

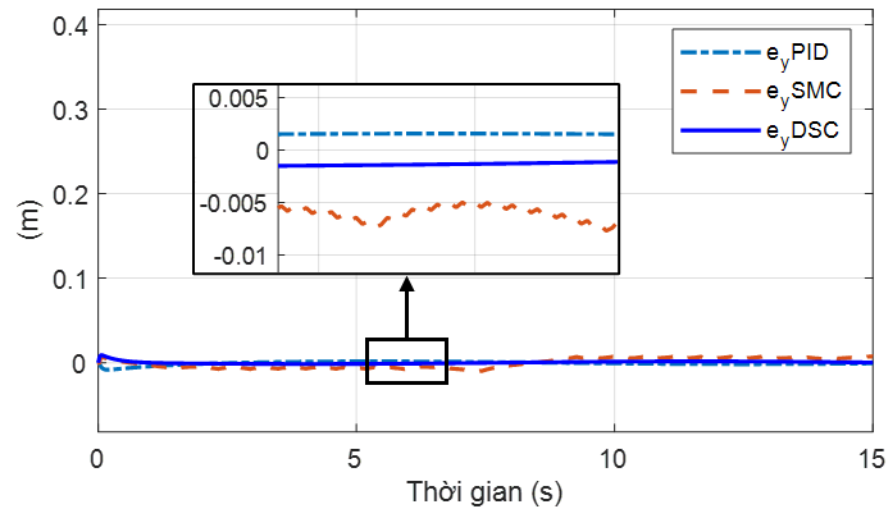


Hình 2.3 Quỹ đạo bám của các thuật toán với FMWR có trọng tâm không đổi

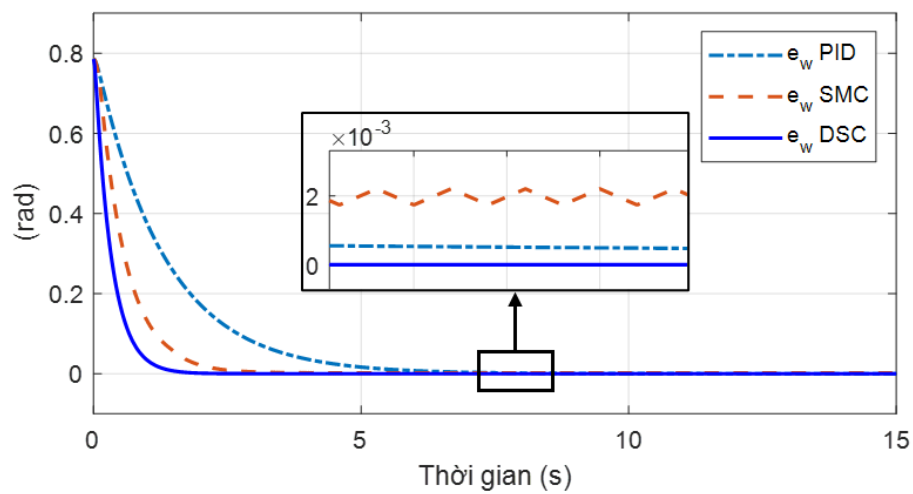
Quỹ đạo di chuyển của robot, được trình bày trong hình 2.3, cho thấy rằng các thuật toán điều khiển PID, SMC và DSC đều giúp robot bám sát quỹ đạo đặt. Trong đó, thuật toán PID giúp robot đạt được quỹ đạo mục tiêu trong khoảng thời gian 4 giây, tuy nhiên vẫn còn sai lệch lớn. Thuật toán SMC đạt được quỹ đạo mục tiêu trong thời gian ngắn, và sai lệch tĩnh cũng nhỏ hơn so với PID. Thuật toán DSC mang lại kết quả bám quỹ đạo tốt hơn với thời gian hoàn thành bám quỹ đạo ngắn hơn.



Hình 2.4 Sai số của quỹ đạo theo trục x



Hình 2.5 Sai số của quỹ đạo theo trục y

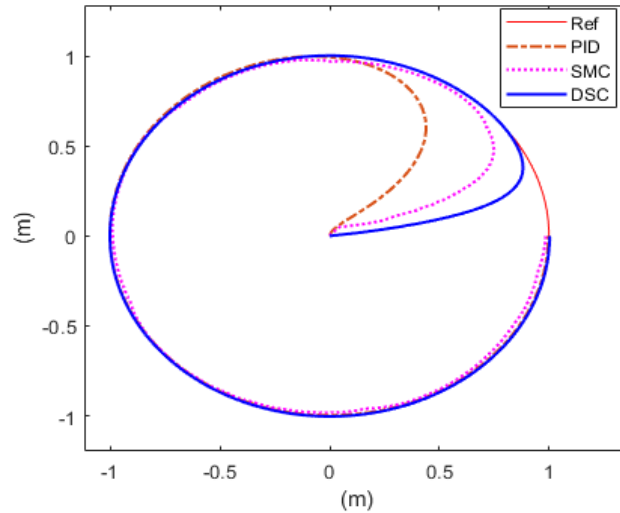


Hình 2.6 Sai số của quỹ đạo theo góc hướng.

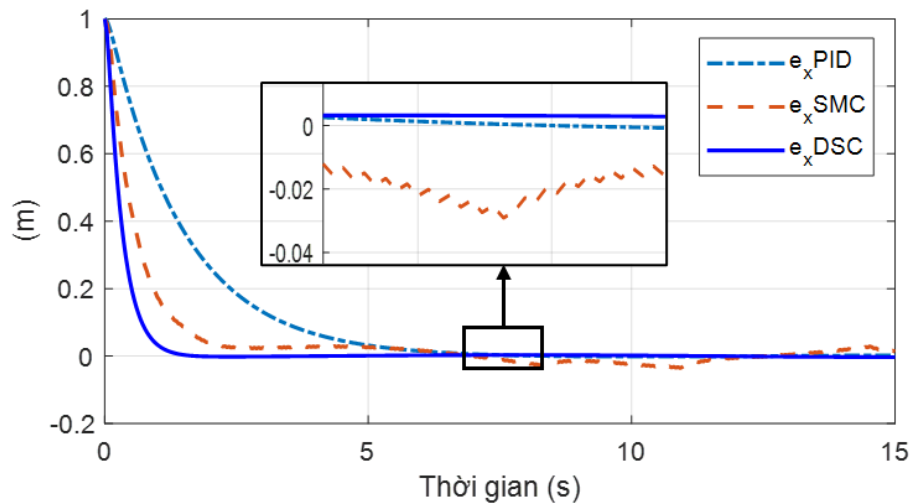
Kết quả sai lệch theo các trục x , y và góc hướng trong hình 2.4 và hình 2.6 cho thấy thuật toán PID còn tồn tại sai lệch lớn. Khi hệ thống ở trạng thái xác lập thì thuật toán SMC vẫn còn tồn tại sai lệch và có hiện tượng rung ‘chattering’. Trong khi đó thuật toán điều khiển DSC cho chất lượng điều khiển bám quỹ đạo tốt hơn với thời gian đáp ứng nhanh hơn, trong đó sai lệch tĩnh nhỏ và hiện tượng "chattering" giảm mạnh so với thuật toán SMC.

- **Kịch bản 2:** Mô phỏng với mô hình FMWR-ME có độ lệch trọng tâm $d_1 = d_2 = 0.1m$, khối lượng thay đổi $\Delta m = 5kg$

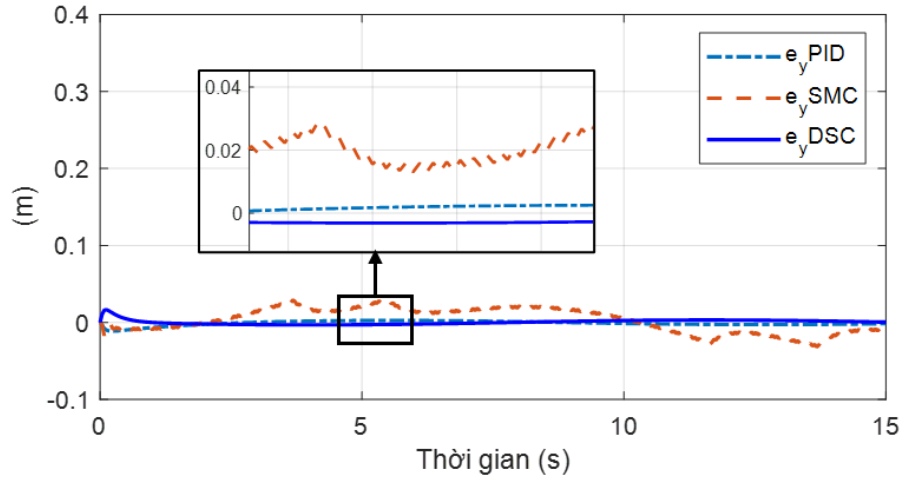
Tiến hành mô phỏng trong thời gian $T=15s$. Các bộ điều khiển được mô phỏng cùng nhau để so sánh đánh giá kết quả:



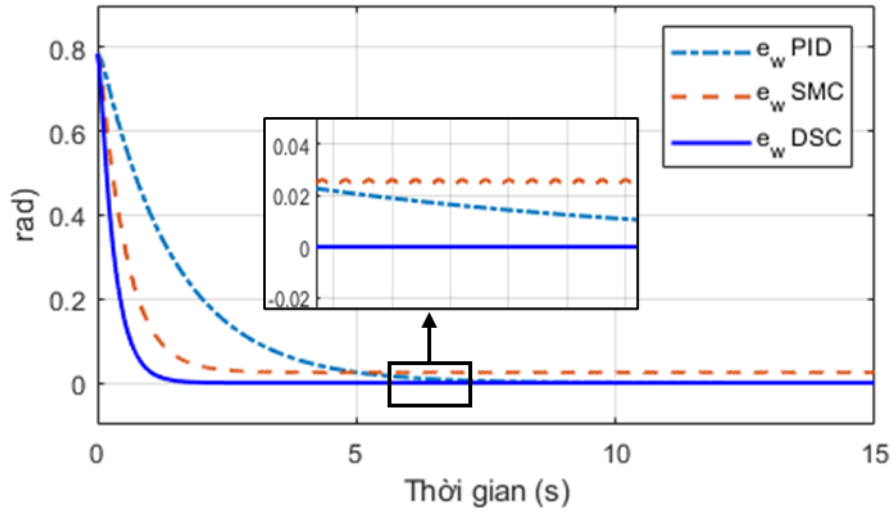
Hình 2.7 Quỹ đạo bám của FMWR-ME khi trọng tâm thay đổi



Hình 2.8 Sai lệch quỹ đạo trục x của FMWR-ME



Hình 2.9 Sai lệch quỹ đạo trực y của FMWR-ME

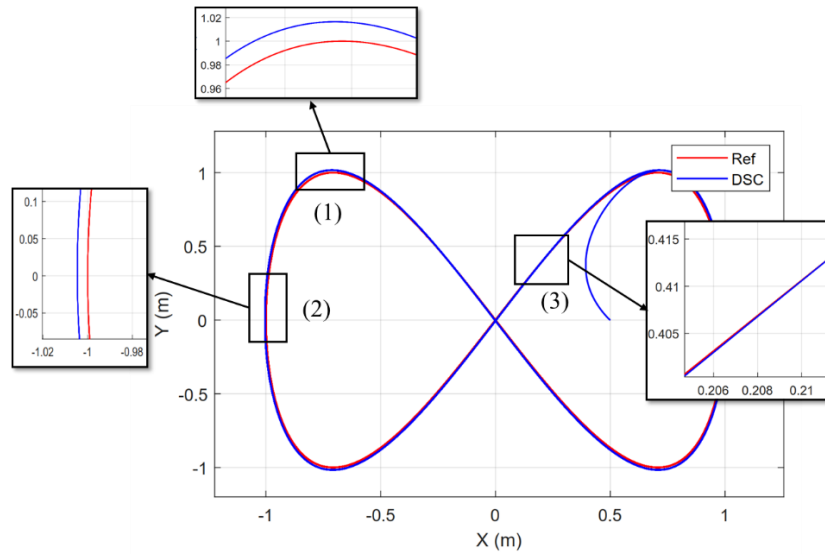


Hình 2.10 Sai lệch quỹ đạo theo góc hướng của FMWR-ME

Dựa trên kết quả mô phỏng trong hình 2.7 – 2.10, có thể thấy rằng khi trọng tâm của robot thay đổi, bộ điều khiển PID gặp phải hiện tượng mất ổn định và sai lệch lớn hơn. Trong khi đó, các bộ điều khiển SMC và DSC vẫn duy trì khả năng hoạt động và bám quỹ đạo, tuy nhiên sai lệch vẫn lớn hơn so với trường hợp trọng tâm của robot không thay đổi, hiện tượng rung ‘chattering’ của thuật toán SMC vẫn còn tồn tại. Tuy sai lệch bám quỹ đạo của thuật toán DSC nhỏ hơn của PID và SMC nhưng bị ảnh hưởng của độ lệch trọng tâm gây nên sai lệch đáng kể.

- **Kịch bản 3:** Thay đổi quỹ đạo mô phỏng hình số 8, cho bởi phương trình sau:

$$x_r = \sin(t); y_r = \sin(2t); \theta_r = \frac{\pi}{4}$$



Hình 2.11 Quỹ đạo bám của thuật toán DSC với hình số 8

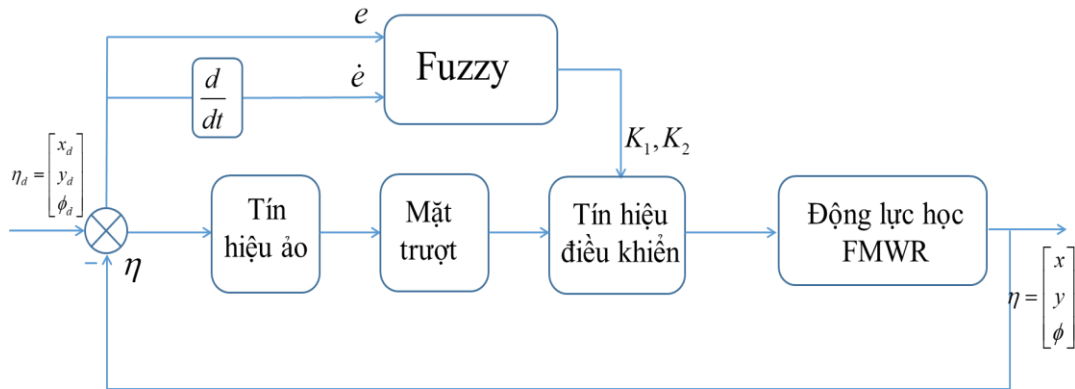
Kết quả mô phỏng trong hình 2.11 cho thấy khi thay đổi quỹ đạo thì thuật toán DSC điều khiển FMWR di chuyển tiến tới bám theo quỹ đạo và hoạt động ổn định trên quỹ đạo đặt. Tuy nhiên, khi robot di chuyển đến các vị trí quỹ đạo có góc cua khác nhau tồn tại các sai lệch khác nhau. Cụ thể khi robot di chuyển đến vị trí số (1) có góc cua hẹp, chịu ảnh hưởng của lực ly tâm lớn dẫn đến tồn tại sai lệch lớn nhất, bên cạnh đó tại vị trí số (3) quỹ đạo di chuyển ổn định hơn thì sai lệch nhỏ hơn so với vị trí số (2) có đường vòng cung lớn. Điều này cho thấy các tham số điều khiển của thuật toán DSC hiện tại có thể chưa tối ưu, dẫn đến hiệu suất bám quỹ đạo chưa được ổn định.

Nhận xét: Từ các kịch bản mô phỏng cho thấy rằng thuật toán điều khiển DSC có tham số điều khiển K_1 và K_2 ảnh hưởng đến chất lượng bám quỹ đạo của robot. Cụ thể, tham số K_1 quyết định tốc độ tiến đến mặt trượt, trong khi K_2 ảnh hưởng đến sự ổn định của hệ thống trên mặt trượt. Tuy nhiên, việc lựa chọn các tham số phù hợp để đạt được hiệu suất tối ưu thường phức tạp và phụ thuộc vào nhiều yếu tố. Để cải thiện khả năng thích nghi và nâng cao hiệu suất điều khiển, cần đề xuất sử dụng một bộ chỉnh định mờ nhằm tối ưu hóa các tham số điều khiển. Bộ chỉnh định mờ có thể liên tục điều chỉnh K_1 và K_2 theo trạng thái hiện tại của robot và các ảnh hưởng của môi trường, từ đó đảm bảo chất lượng bám quỹ đạo tốt hơn.

2.6. Đề xuất thuật toán điều khiển mặt trượt động chỉnh định thích nghi mờ cho robot di động đa hướng mecanum (Fuzzy-DSC)

2.6.1. Thiết kế thuật toán Fuzzy-DSC

Bộ điều khiển logic mờ được tích hợp vào hệ thống để tự động điều chỉnh các tham số K_1 và K_2 của thuật toán điều khiển DSC. Trong quá trình di chuyển trên quỹ đạo, các tín hiệu đầu vào của bộ chỉnh định bao gồm sai lệch vị trí $\mathbf{e}$ và sai lệch vận tốc $\dot{\mathbf{e}}$. Dựa trên những tín hiệu này, bộ điều khiển logic mờ sẽ điều chỉnh các tham số sao cho phù hợp với từng giai đoạn của quá trình điều khiển. Thuật toán điều khiển mặt trượt động chỉnh định thích nghi mờ (Fuzzy-DSC) đã được thiết kế để tối ưu hóa quá trình điều khiển bám quỹ đạo. Sơ đồ cấu trúc của thuật toán này được minh họa trong hình 2.12.



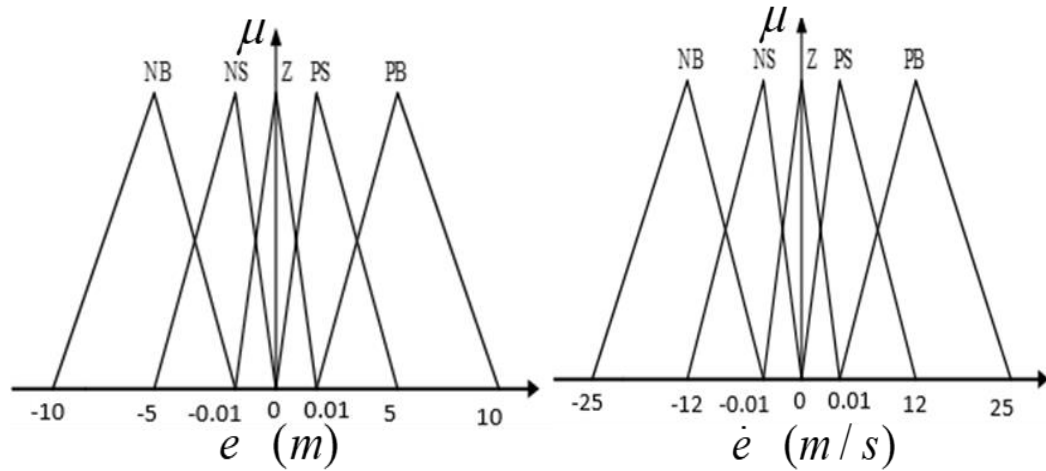
Hình 2.12 Sơ đồ khối thuật toán điều khiển Fuzzy-DSC

Tín hiệu đầu vào của bộ chỉnh định mờ được thiết kế dựa trên sai lệch vị trí $\mathbf{e}$ và sai lệch của vận tốc theo thời gian $\dot{\mathbf{e}}$. Các tập mờ biểu diễn cho các biến ngôn ngữ đầu vào được định nghĩa và ký hiệu như trong bảng 2-1.

Bảng 2-1 : Xác định ngôn ngữ tập mờ của biến đầu vào

Biến đầu vào $\mathbf{e}$	Biến đầu vào $\dot{\mathbf{e}}$	Mô tả
NB	NB	Mức độ âm lớn
NS	NS	Mức độ âm nhỏ
Z	Z	Mức độ 0
PS	PS	Mức độ dương nhỏ
PB	PB	Mức độ dương lớn

Trong đó, các tập mờ được thiết kế theo mô hình mờ Sugeno [90], với các hàm tam giác được biểu diễn trong hình 2.13.



Hình 2.13 Tập mờ đối với biến đầu vào e_1 , e_2

Các giá trị của tham số đầu ra K_1, K_2 được định nghĩa và ký hiệu thể hiện như trong bảng 2-2.

Bảng 2-2: Xác định tập mờ của giá trị đầu ra

Đầu ra	Mô tả	Đầu ra tín hiệu K_1	Đầu ra tín hiệu K_2
VS	Rất bé	2	5
S	Bé	4	10
M	Trung bình	10	15
B	To	13	20
VB	Rất to	20	30

Hệ thống quy tắc suy diễn cơ bản của bộ chỉnh định mờ cho đầu ra (K_1, K_2) trình bày trong bảng 2-3 và bảng 2-4.

Bảng 2-3: Hệ luật cơ sở suy diễn cho tham số K_1

e_2	e_1				
	NB	NS	Z	PS	PB
NB	<i>M</i>	<i>S</i>	<i>VS</i>	<i>S</i>	<i>M</i>
NS	<i>B</i>	<i>M</i>	<i>S</i>	<i>M</i>	<i>B</i>
Z	<i>VS</i>	<i>B</i>	<i>M</i>	<i>B</i>	<i>VS</i>

PS	B	M	S	M	B
PB	M	S	VS	S	M

Bảng 2-4: Hệ luật cơ sở suy diễn cho tham số K_2

$\dot{\mathbf{e}}_1$	$\mathbf{e}_1$				
	NB	NS	Z	PS	PB
NB	M	B	VB	B	M
NS	S	M	B	M	S
Z	VB	S	M	S	VB
PS	S	M	B	M	S
PB	M	B	VB	B	M

2.6.2. Mô phỏng và đánh giá kết quả

Để kiểm chứng hiệu quả của thuật toán đề xuất, quá trình mô phỏng được thực hiện trên Matlab/Simulink. Thuật toán được mô phỏng với các cấu trúc mô phỏng đã được thiết kế, như trong **PL1**.

Tham số của mô hình FMWR được lựa chọn như (2.27):

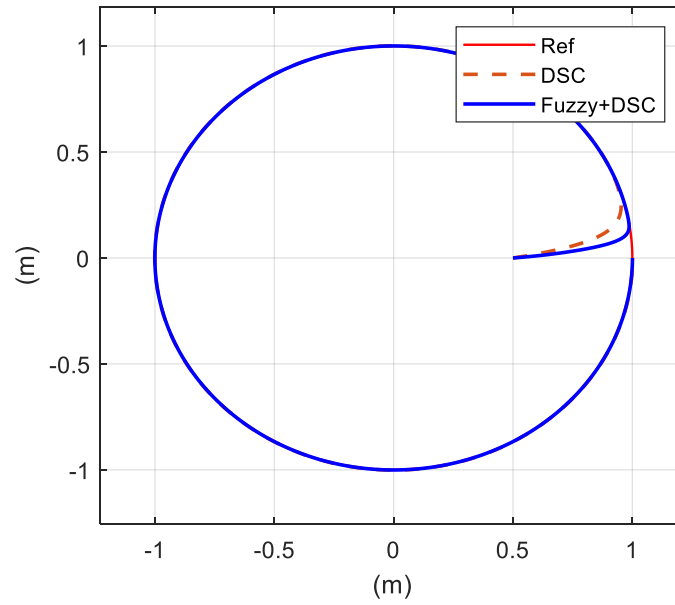
Tham số của bộ điều khiển: $K_1 = \text{diag}(5, 5, 5); K_2 = \text{diag}(10, 10, 10)$

➤ Mô phỏng thuật toán Fuzzy-DSC với quỹ đạo đường tròn:

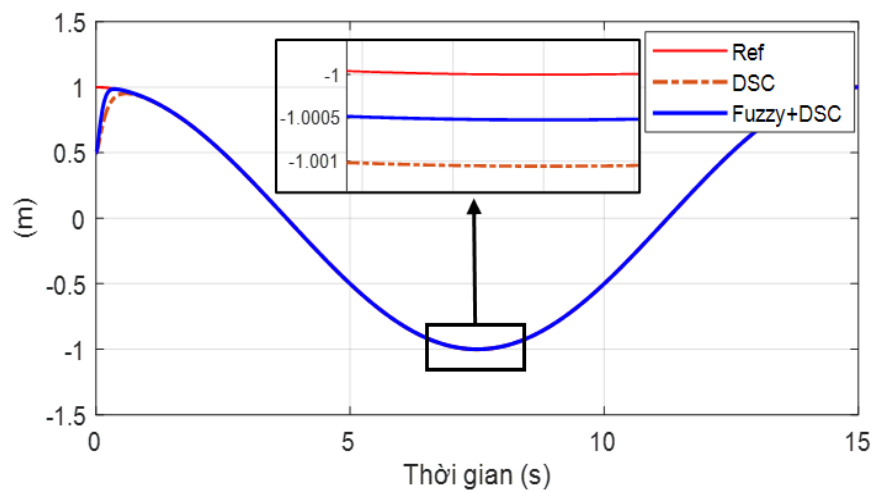
Phương trình mô tả quỹ đạo đường tròn:

$$x_r = \cos\left(\frac{2\pi}{15}t\right); y_r = \sin\left(\frac{2\pi}{15}t\right); \phi_r = \frac{\pi}{4}$$

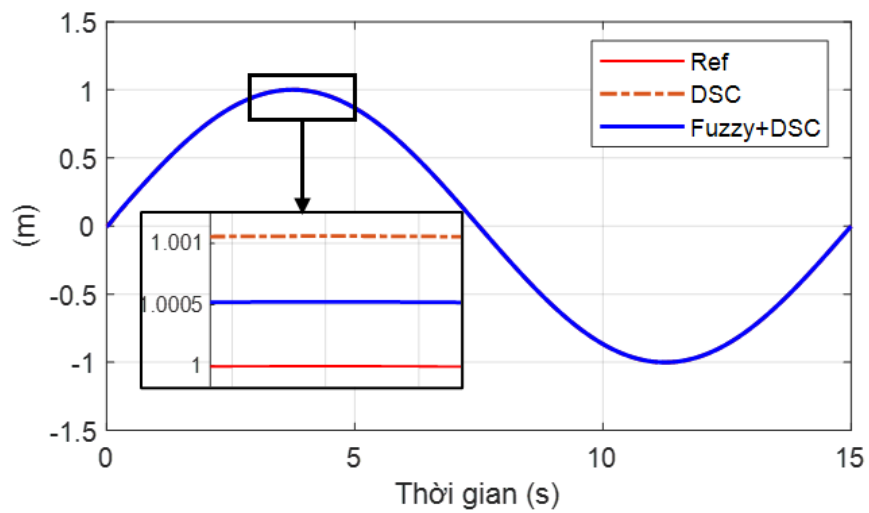
Kết quả mô phỏng trong hình 2.14 cho thấy robot đã bắt đầu từ vị trí có tọa độ (0.5,0) và di chuyển theo quỹ đạo tròn đã định, duy trì việc bám sát quỹ đạo với sai lệch nhỏ và đảm bảo hoạt động ổn định.



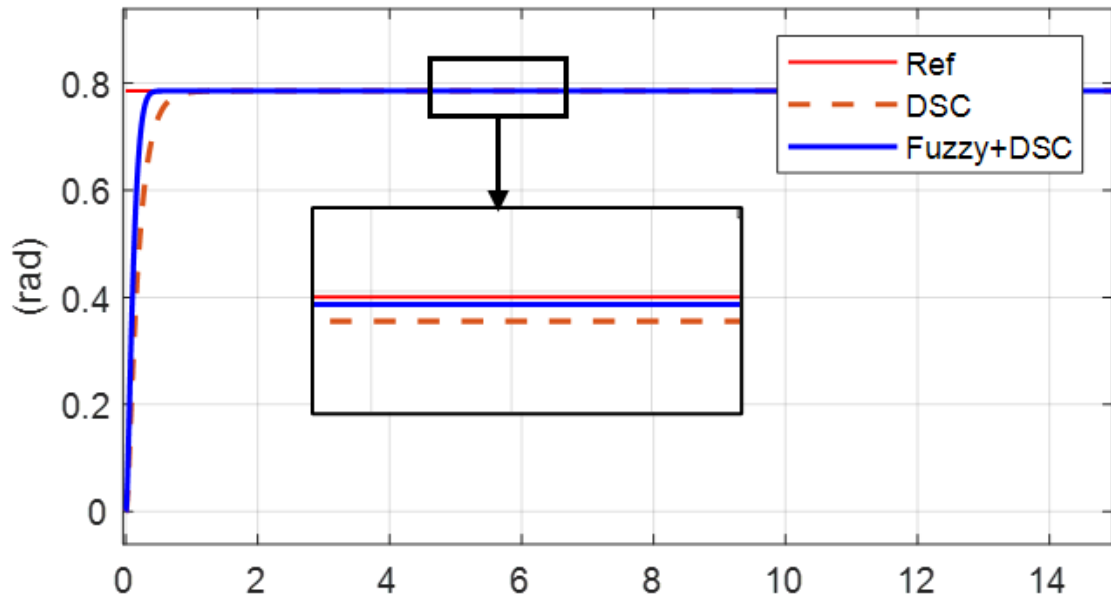
Hình 2.14 Kết quả mô phỏng thuật toán Fuzzy- DSC với quỹ đạo tròn



Hình 2.15 Quỹ đạo mô phỏng của robot theo trục x

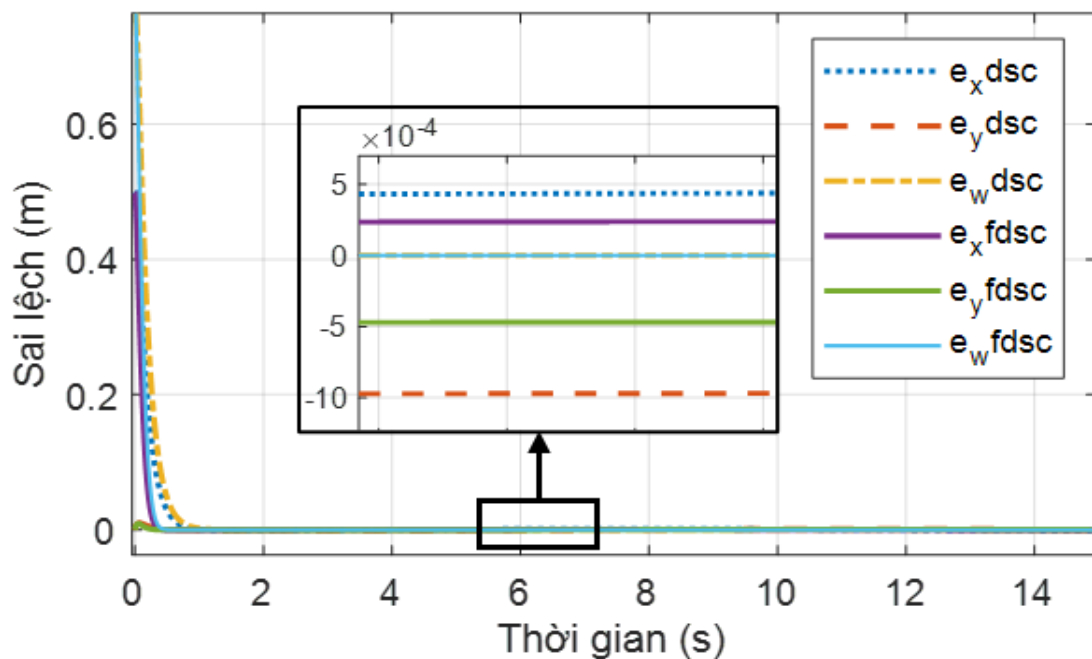


Hình 2.16 Quỹ đạo mô phỏng của robot theo trục y

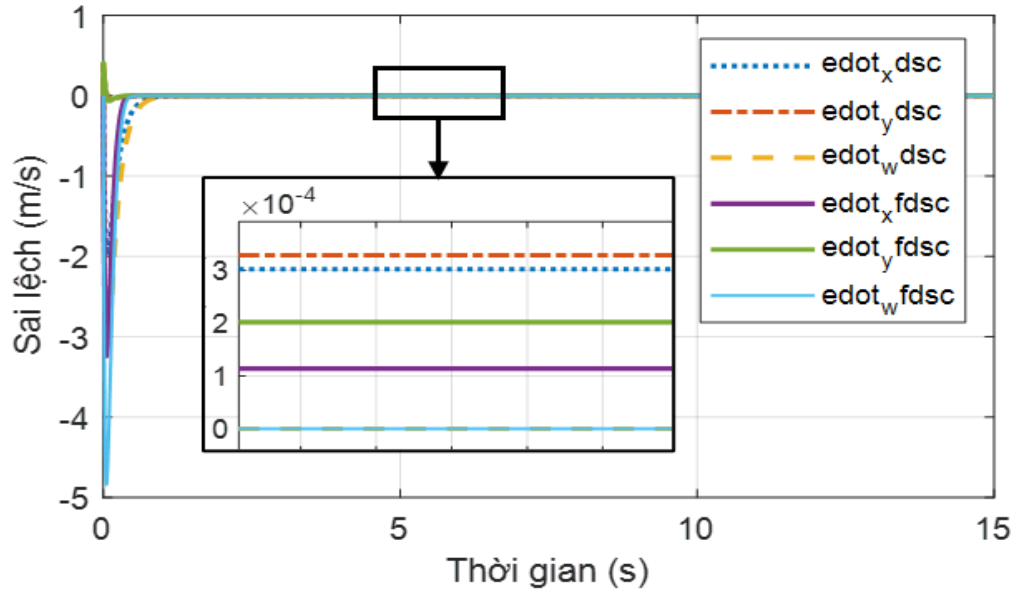


Hình 2.17 Quỹ đạo mô phỏng của robot theo góc hướng

Kết quả mô phỏng quỹ đạo di chuyển của robot FMWR theo các trục x , y và góc hướng trong hình 2.15, hình 2.16 và hình 2.17. Theo đó, robot bắt đầu chuyển động từ vị trí tọa độ $(0.5, 0)$ và di chuyển theo quỹ đạo đặt với thời gian ngắn. Góc hướng của robot cũng duy trì sự chính xác, bám sát theo quỹ đạo được đặt ra. Kết quả mô phỏng còn cho thấy hiện tượng rung đã được giảm đáng kể, góp phần nâng cao hiệu quả và chất lượng điều khiển.

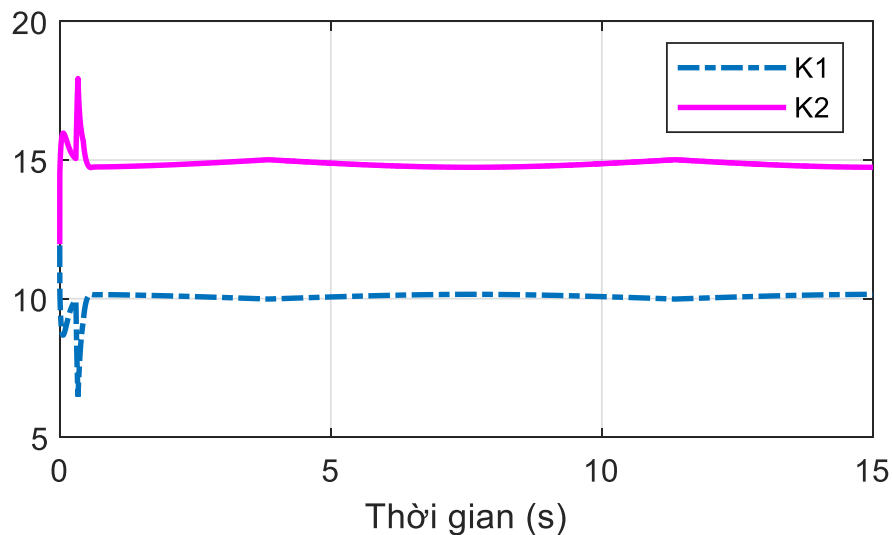


Hình 2.18 Sai lệch quỹ đạo bám của robot



Hình 2.19 Sai lệch của vận tốc theo giờ gian

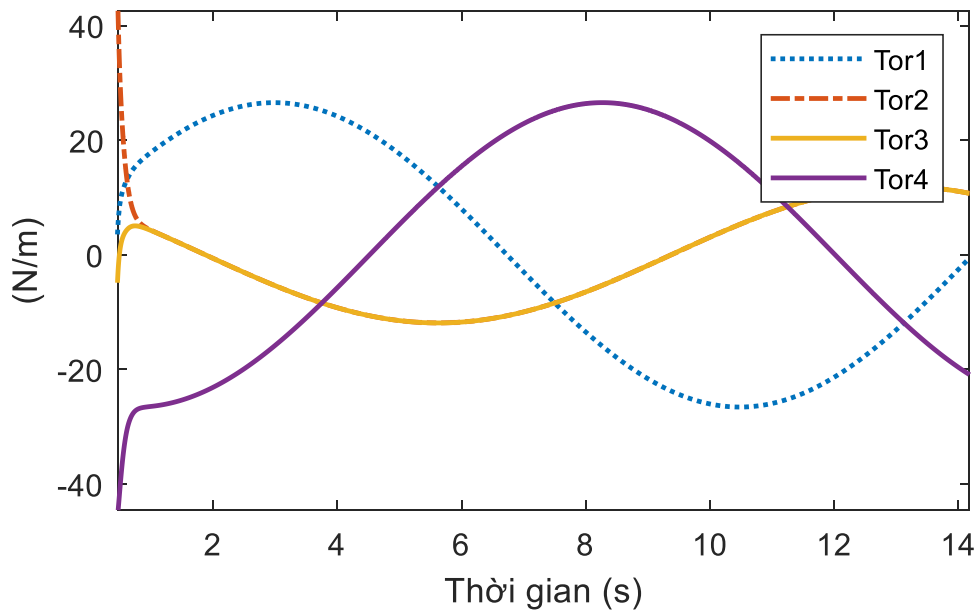
Các sai lệch bám của FMWR được thể hiện trong hình 2.18, sai lệch tiến về 0 trong thời gian ngắn và các sai lệch duy trì và ổn định. Sai lệch tốc độ thể hiện trong hình 2.19 cho thấy thuật toán đề xuất có chất lượng bám tốt hơn thuật toán DSC thông thường. Sai lệch bám được đưa vào trong hệ thống để làm đầu vào của các tập mờ, từ đó chỉnh định các bộ tham số K_1 , K_2 giúp cho robot có thể bám sát được quỹ đạo đặt với sai số nhỏ nhất. Các tham số được chỉnh định thể hiện trong kết quả mô phỏng hình 2.20.



Hình 2.20 Các tham số K_1 , K_2 chỉnh định theo thời gian

Giai đoạn đầu tiên tham số các tham số K_1 , K_2 đã tự chỉnh định dựa vào sai lệch vị trí và sai lệch vận tốc so với quỹ đạo đặt, từ đó đưa ra được tham số điều khiển phù

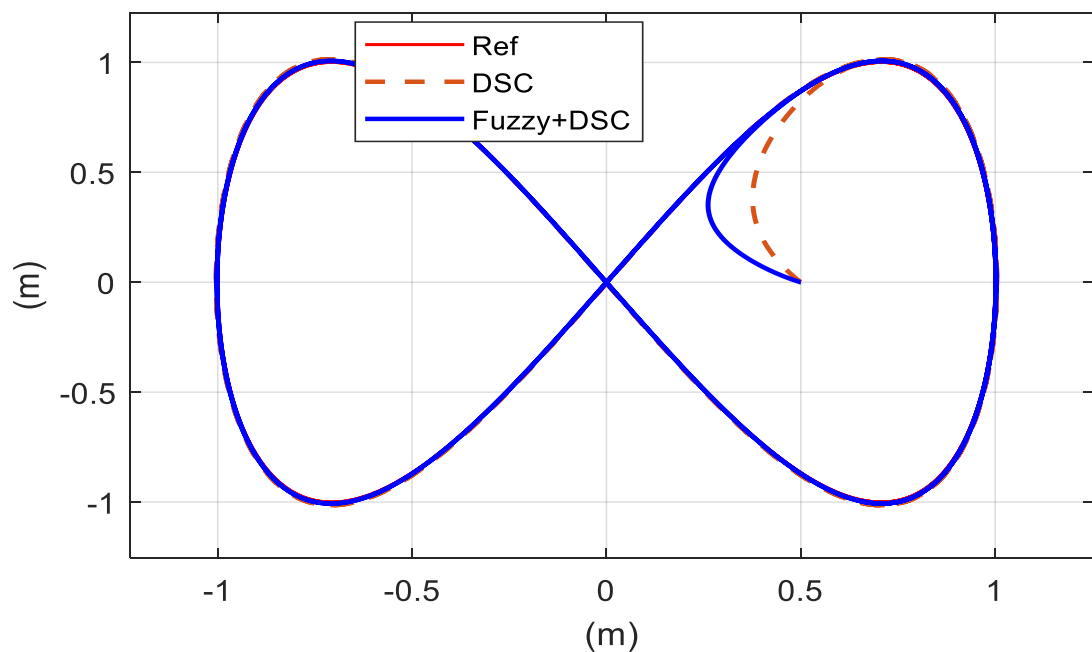
hợp giúp robot tiến đến bám quỹ đạo trong thời gian ngắn, và duy trì ổn định trên đường quỹ đạo đặt.



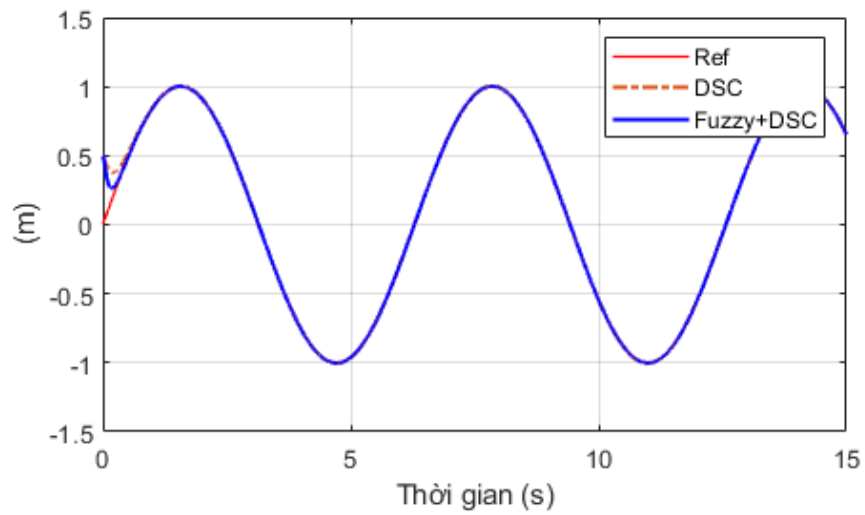
Hình 2.21 Tín hiệu điều khiển Tor với quỹ đạo đường tròn

➤ **Thay đổi quỹ đạo mô phỏng hình số 8:** Quỹ đạo đặt hình số 8 có dạng phương

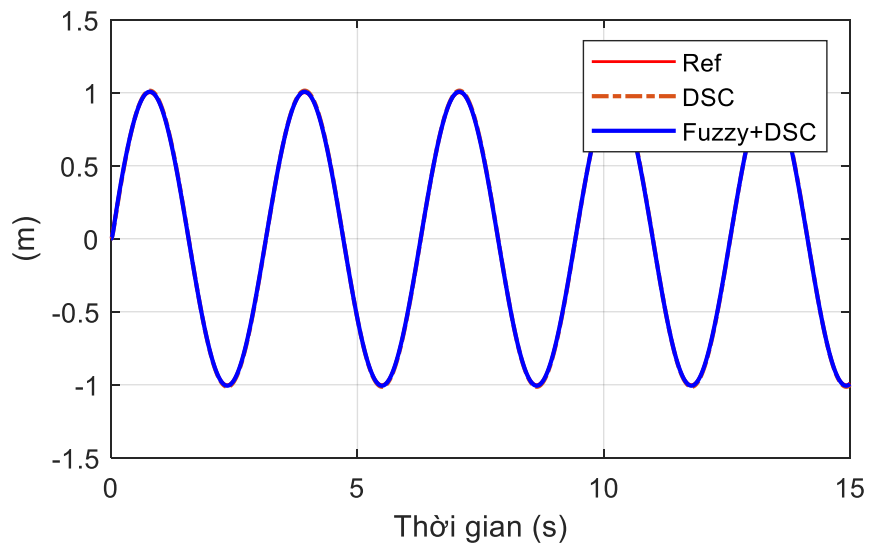
trình: $x_r = \sin(t); y_r = \sin(2t); \phi_r = \frac{\pi}{4}$ mô phỏng được kết quả như sau:



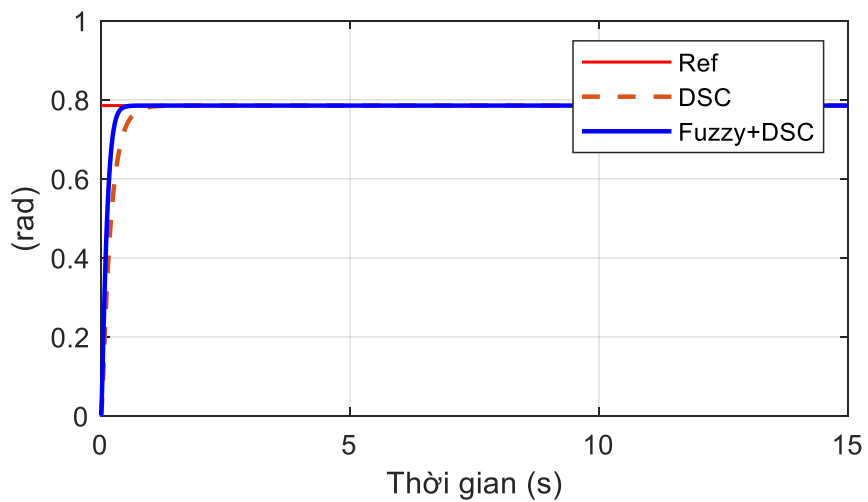
Hình 2.22 Chuyển động của FMWR với quỹ đạo hình số 8



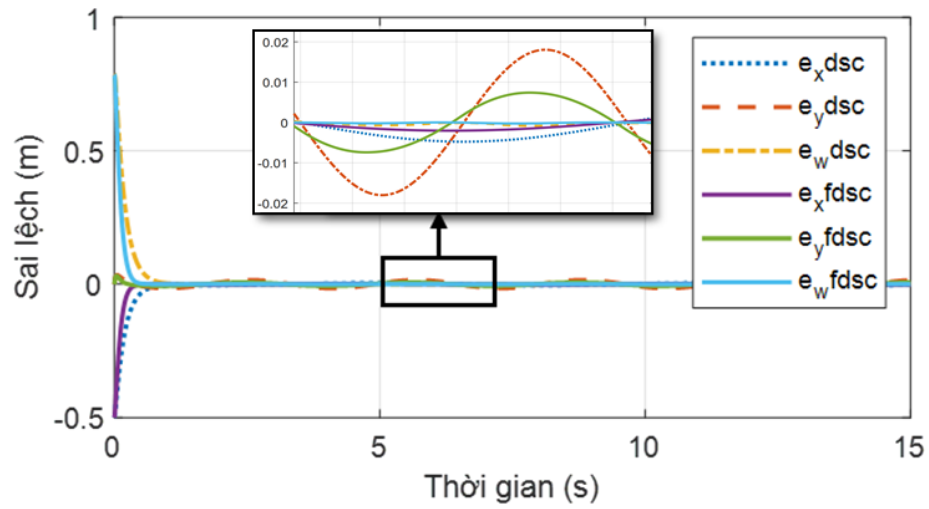
Hình 2.23 Quỹ đạo di chuyển của FMWR theo trục x



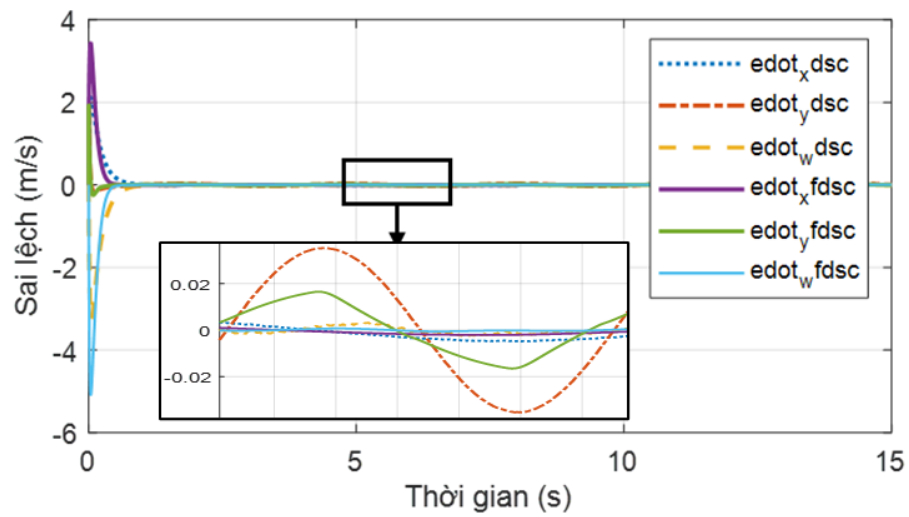
Hình 2.24 Quỹ đạo di chuyển của FMWR theo trục y



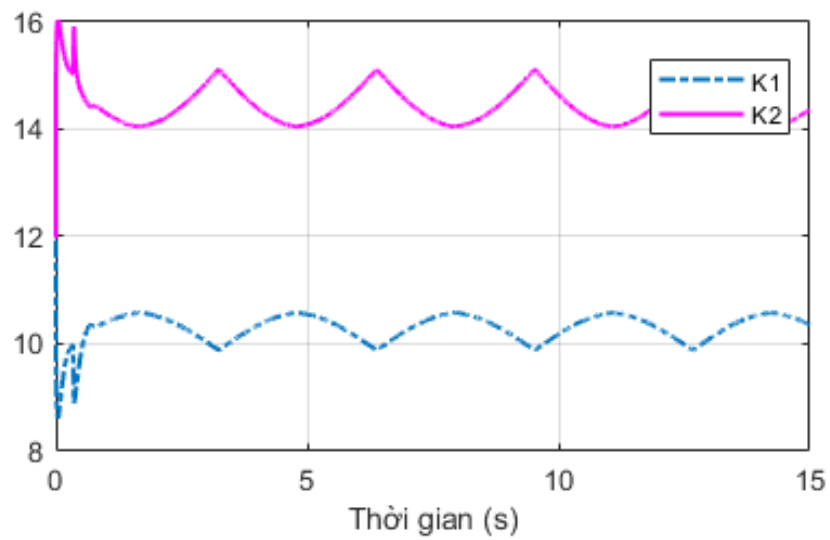
Hình 2.25 Quỹ đạo di chuyển của FMWR theo góc hướng



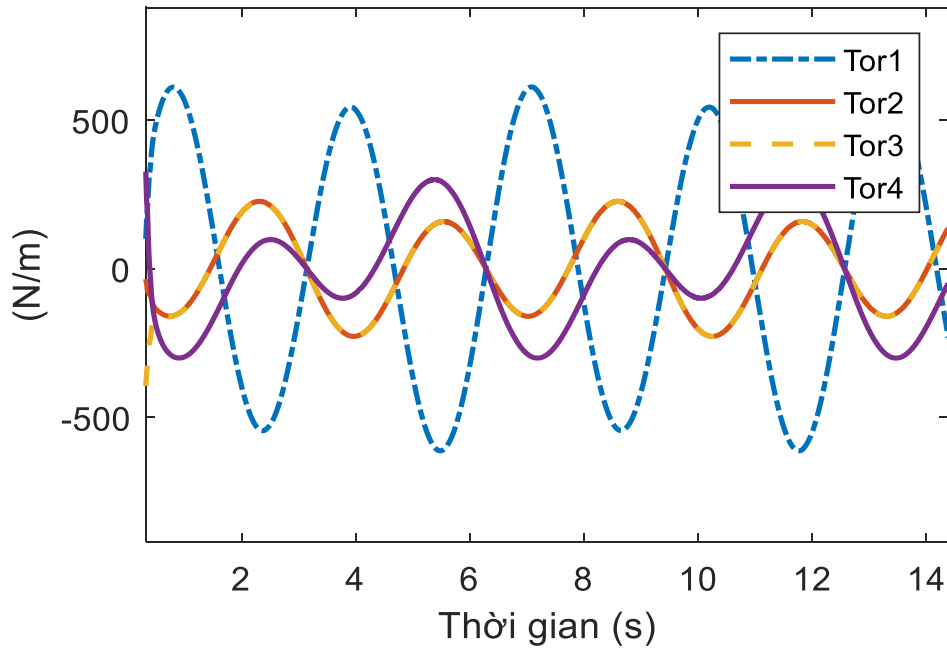
Hình 2.26 Sai lệch quỹ đạo với trọng tâm robot không đổi



Hình 2.27 Sai lệch vận tốc với trọng tâm robot không đổi



Hình 2.28 Chính định các tham số K_1 , K_2



Hình 2.29 Tín hiệu điều khiển Tor với quỹ đạo hình số 8

Bảng 2-5: Bảng so sánh sai lệch quỹ đạo thuật toán DSC và Fuzzy-DSC

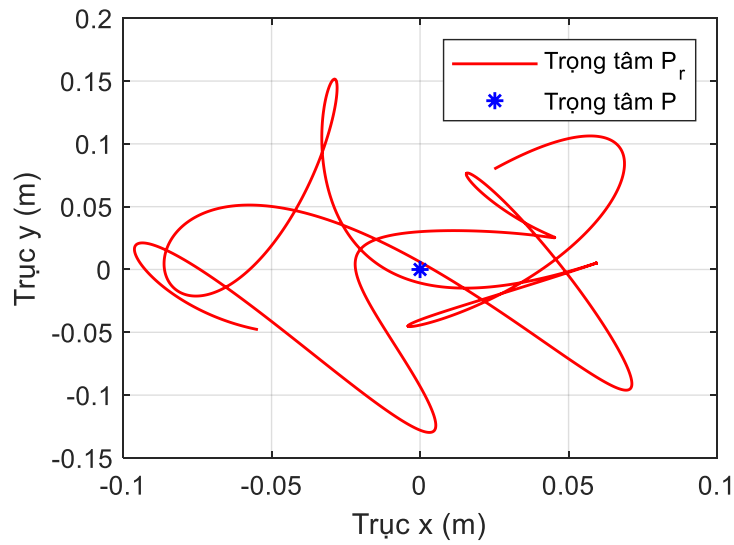
Thời gian (s)	DSC Sai lệch (%)		Fuzzy-DSC Sai lệch (%)	
	x-axis	y-axis	x-axis	y-axis
1	0.97%	3.03%	0.27%	1.34%
2	0.78%	1.40%	0.32%	0.99%
2.5	0.60%	3.05%	0.23%	1.37%
3	0.27%	1.90%	0.09%	0.73%
4	0.49%	2.98%	0.24%	1.39%
5	0.80%	0.60%	0.33%	0.59%

Nhận xét: Khi quỹ đạo mô phỏng được thay đổi, robot vẫn di chuyển theo quỹ đạo đã định trong thời gian ngắn. Tuy nhiên, sự thay đổi quỹ đạo dẫn đến các sai lệch và dao động tại những thời điểm khác nhau trong quá trình di chuyển của robot. Tại thời điểm 1s robot di chuyển ở đỉnh đường cong của quỹ đạo có góc cua hẹp dẫn đến lực ly tâm lớn nên sai lệch tại đây là lớn nhất, đối với thuật toán DSC thông thường là 3,03% còn thuật toán Fuzzy-DSC có sai lệch nhỏ hơn là 1,34%. Tương tự tại các thời điểm khác nhau thuật toán được đề xuất cũng giúp cho robot bám quỹ đạo với sai lệch nhỏ hơn, cho thấy tính hiệu quả của thuật toán được đề xuất.

➤ **Mô phỏng thuật toán Fuzzy-DSC và DSC thông thường với mô hình robot có trọng tâm và khối lượng thay đổi (FMWR-ME)**

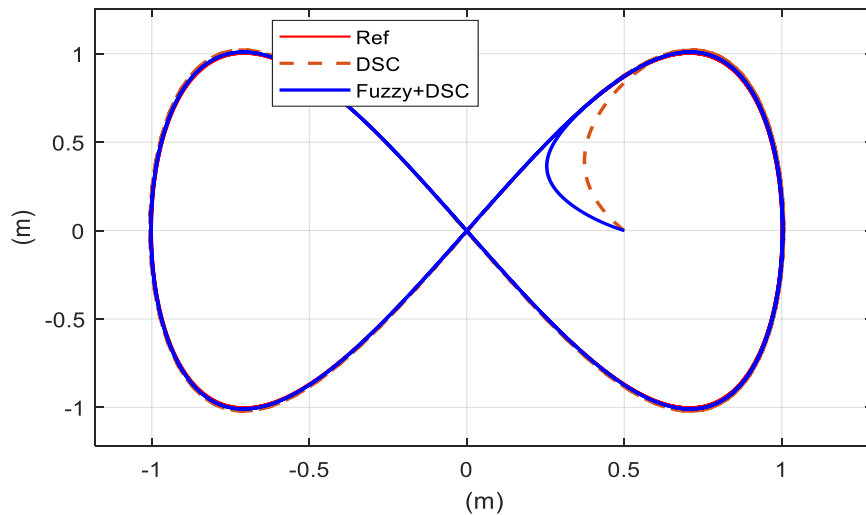
- Giả thiết vị trí trọng tâm robot thay đổi biến thiên được mô tả bởi phương trình như sau:

$$\begin{aligned} d_1 &= 0.05\sin(0.8t) + 0.025\cos(1.5t) + 0.03\sin(2.5t) \\ d_2 &= 0.03\sin(0.3t) + 0.06\cos(1.2t) + 0.05\sin(2.8t) + 0.02\cos(3.5t) \quad (2.44) \\ \Delta m &= 5kg \end{aligned}$$



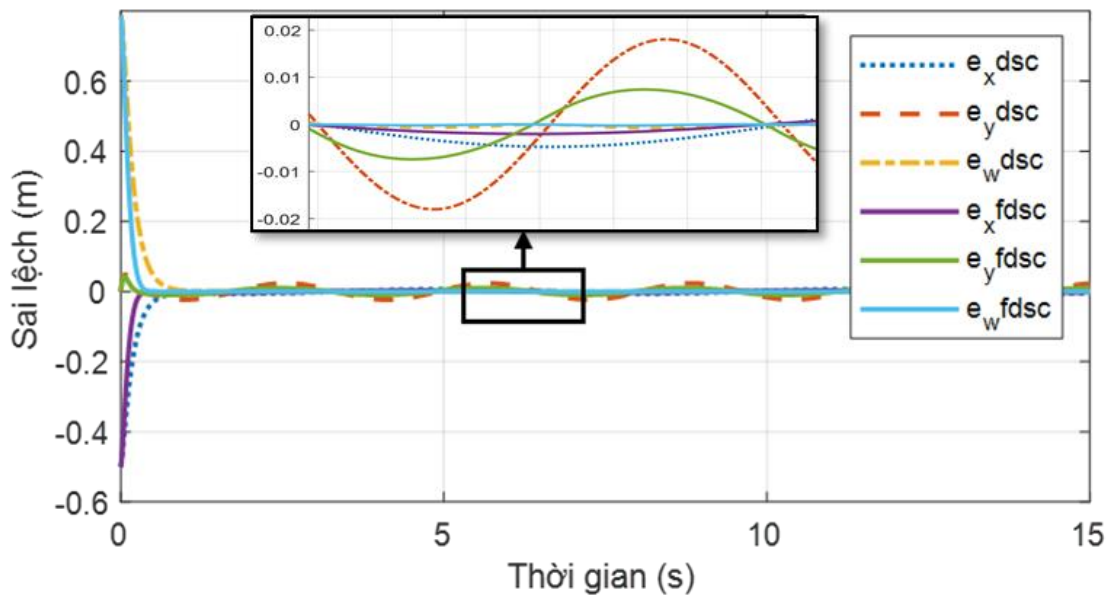
Hình 2.30 Mô tả vị trí trọng tâm robot thay đổi biến thiên

Trọng tâm của robot biến thiên như mô tả trong hình 2.30 ứng với trường hợp thực tế robot có găng cánh tay cộng tác vừa di chuyển vừa thao tác chuyển động cánh tay làm thay đổi trọng tâm thực tế (P_r) của robot so với vị trí trọng tâm hình học (P).

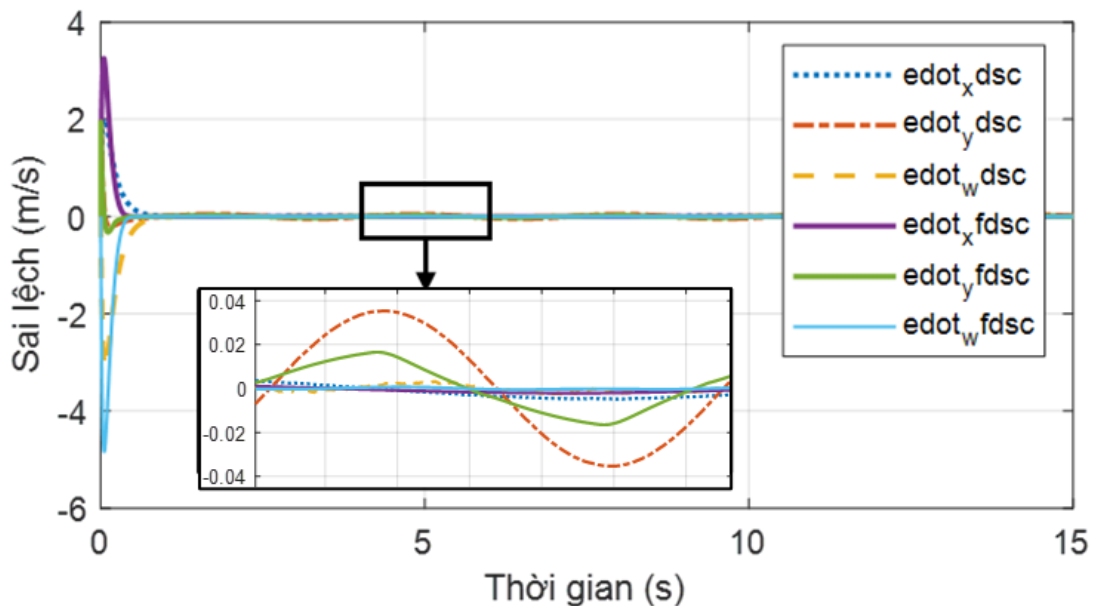


Hình 2.31 Quỹ đạo di chuyển của FMWR-ME

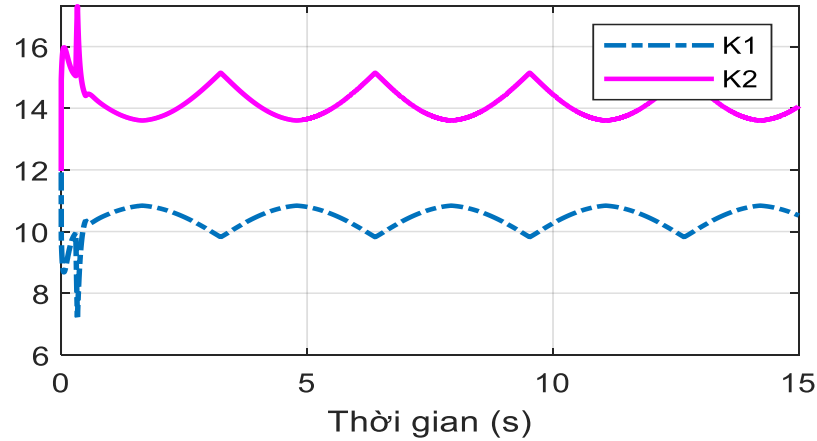
Kết quả trong hình 2.31 cho thấy khi khối lượng và trọng tâm của robot thay đổi cả hai thuật toán vẫn có thể điều khiển robot di chuyển theo quỹ đạo tham chiếu và hoạt động ổn định. Tuy nhiên, có một độ lệch đáng chú ý, như trong hình 2.32, trong đó thuật toán DSC thể hiện lỗi theo dõi quỹ đạo lớn hơn là $e_{ydsc}=0,023m$. Hình 2.33 cho thấy rằng thuật toán Fuzzy-DSC đạt được hiệu suất bám vận tốc tốt hơn. Ngoài ra, thuật toán Fuzzy-DSC điều chỉnh các tham số bộ điều khiển K_1 , K_2 , như thể hiện trong hình 2.34, để thích ứng với các thay đổi của mô hình, dẫn đến chất lượng điều khiển được cải thiện với lỗi theo dõi nhỏ hơn là $e_{yfdsc}=0,011m$.



Hình 2.32 Sai lệch của quỹ đạo FMWR-ME



Hình 2.33 Sai lệch vận tốc của FMWR-ME



Hình 2.34 Chỉnh định các tham số khi lệch trọng tâm

**Bảng 2-6: Bảng đánh giá sai lệch của quỹ đạo robot
với trọng tâm thay đổi**

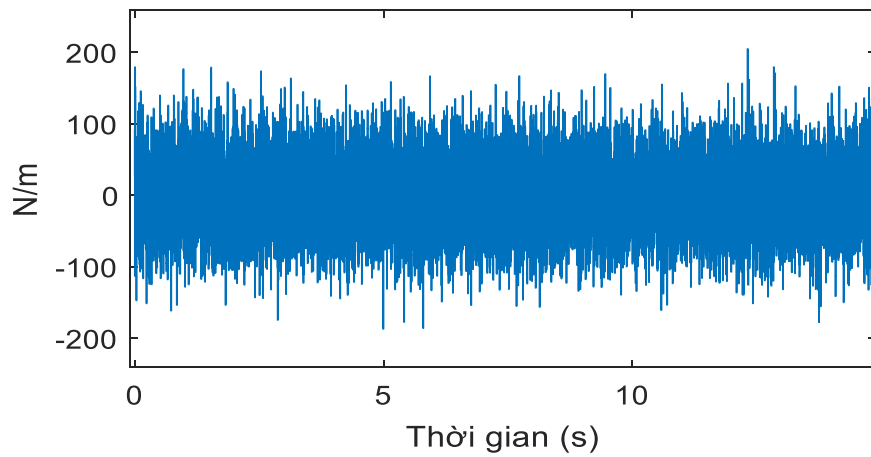
Thời gian (s)	DSC Sai lệch (%)		Fuzzy-DSC Sai lệch (%)	
	x-axis	y-axis	x-axis	y-axis
1	1.35%	3.30%	0.28%	1.42%
2	0.87%	1.41%	0.33%	1.00%
2.5	0.67%	3.33%	0.25%	1.39%
3	0.31%	2.18%	0.11%	0.74%
4	0.50%	3.23%	0.27%	1.44%
5	0.89%	0.62%	0.36%	0.61%

Trong mô phỏng với robot có khối lượng và trọng tâm bị thay đổi thì quỹ đạo di chuyển của robot vẫn bám theo quỹ đạo đặt nhưng tồn tại sai lệch lớn hơn. Cụ thể tại các thời điểm 1s và 2,5s robot đi chuyển đến quỹ đạo có vòng cua nhỏ nên chịu tác động của độ lệch trọng tâm nhiều hơn gây nên sai lệch lớn hơn đối với thuật toán DSC thông thường như trong bảng 2-6. Bên cạnh đó với bộ chỉnh định fuzzy được thêm vào đã giúp cho thuật toán Fuzzy-DSC giảm được sự ảnh hưởng của lệch trọng tâm và khối lượng thay đổi giúp cho robot vẫn bám sát với quỹ đạo đặt với sai lệch nhỏ hơn. Các sai lệch về vận tốc thể hiện trong hình 2.33 thấy chất lượng bám trong thuật toán được đề xuất là nhỏ hơn và không còn hiện tượng rung. Điều đó cho thấy thuật toán Fuzzy-DSC đề xuất có hiệu quả hơn so với thuật toán DSC thông thường

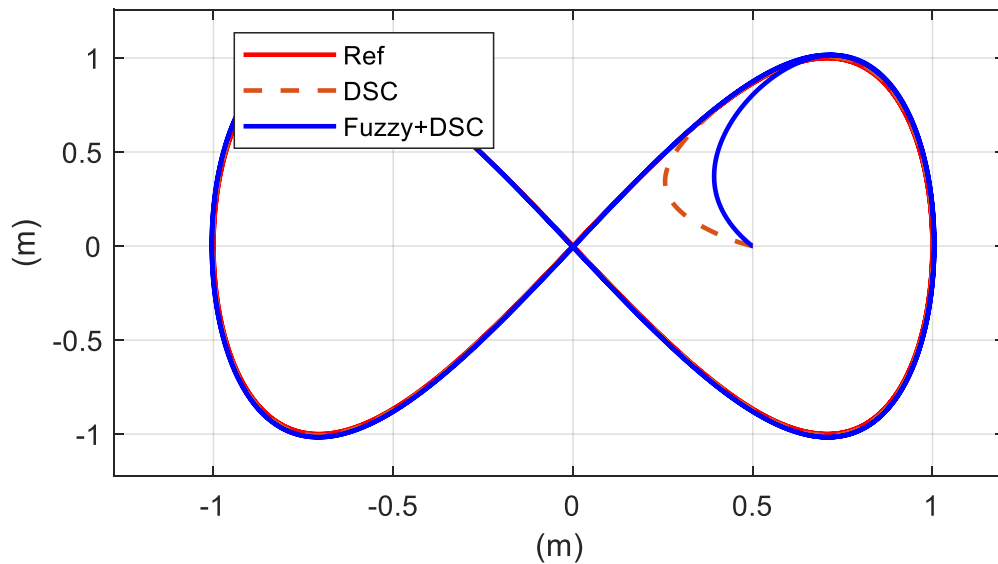
➤ **Mô phỏng trường hợp có nhiễu ngoại tác động:**

Nhiều Gaussian được thêm vào mô hình của FMWR để đánh giá khả năng thích nghi với nhiễu của bộ điều khiển đề xuất. Nhiễu ngẫu nhiên này không nằm trong thuật toán điều khiển và khó xác định chính xác, nhiễu có dạng như hình 2.35 được mô tả như sau:

$$\epsilon_{Gaus} = \text{normrnd}(0, 50, \text{size}(t)) \quad (2.45)$$

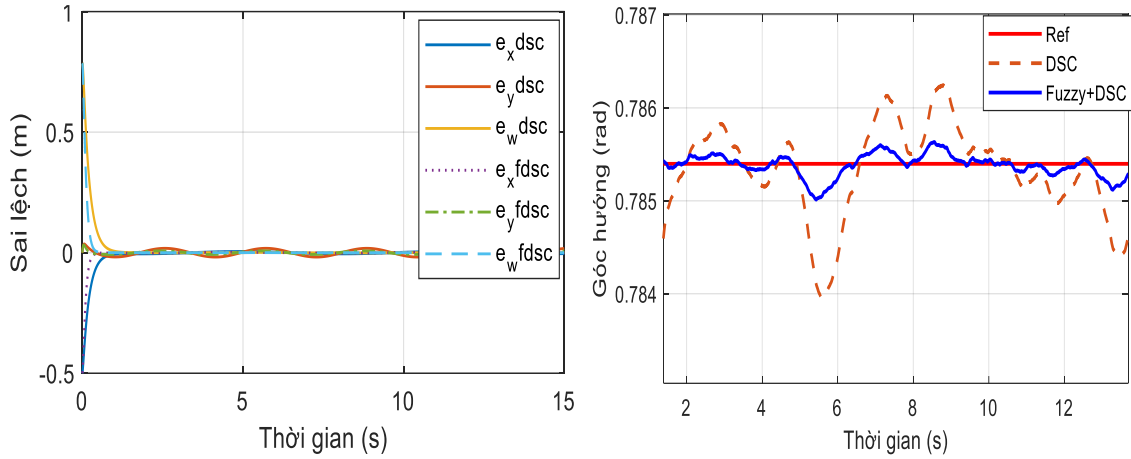


Hình 2.35 Nhiễu ngoại lực Gaussian



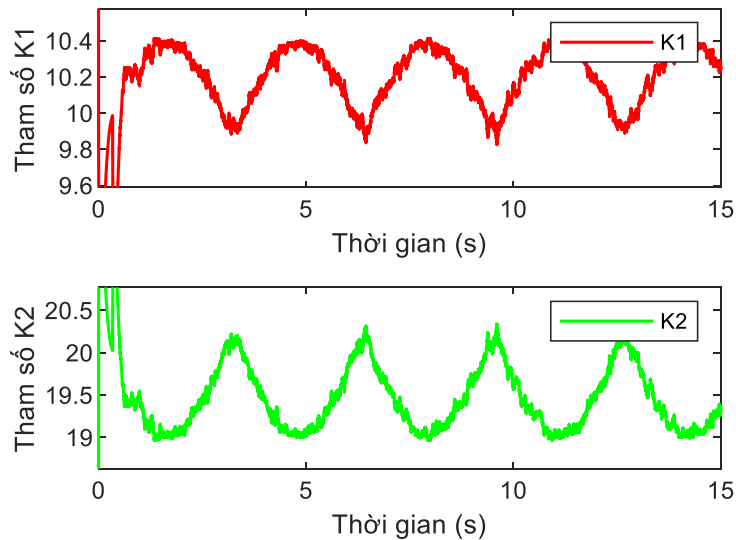
Hình 2.36 Kết quả mô phỏng có nhiễu ngoại

Kết quả mô phỏng hình 2.36 cho thấy khi nhiễu được đưa vào mô hình thì cả hai thuật toán đều điều khiển robot bám theo quỹ đạo đặt. Tuy nhiên, còn gặp phải sự ảnh hưởng từ nhiễu, dẫn đến hiện tượng dao động.



Hình 2.37 Sai lệch quỹ đạo khi có nhiễu ngoại

Trong đó đối với thuật toán Fuzzy-DSC khi có nhiễu xuất hiện, bộ chỉnh định mờ nhận diện và thích nghi với sự tác động của nhiễu bằng cách chỉnh định các tham số điều khiển K_1 , K_2 sao cho phù hợp với tác động của nhiễu như trong hình 2.38. Do vậy, sai lệch quỹ đạo của thuật toán Fuzzy-DSC nhỏ hơn so với thuật toán DSC thông thường thể hiện trong hình 2.37.



Hình 2.38 Chỉnh định tham số K_1 , K_2 khi có nhiễu

Nhận xét: Thuật toán Fuzzy-DSC tỏ ra vượt trội so với DSC thông thường trong việc xử lý nhiễu và độ lệch trọng tâm nhờ vào khả năng tự điều chỉnh tham số và thích nghi linh hoạt với các tín hiệu nhiễu. Cụ thể, bộ điều khiển mờ Fuzzy có thể thay đổi nhanh chóng các tham số điều khiển như thông qua các luật mờ được thiết lập, giúp hệ thống điều khiển thích nghi với các thay đổi bất ngờ của môi trường hoặc

hiệu. Điều này giúp tối ưu hóa hiệu suất hệ thống điều khiển và duy trì tính ổn định trong suốt quá trình di chuyển của robot. Nhờ vậy, FMWR có thể bám sát quỹ đạo đặt một cách chính xác hơn, đồng thời giảm thiểu các dao động do nhiễu gây ra, mang lại kết quả điều khiển tốt hơn so với bộ điều khiển DSC thông thường.

2.7. Kết luận chương 2

Trong chương 2, một số thuật toán điều khiển đã được áp dụng để điều khiển bám quỹ đạo cho FMWR và thực hiện mô phỏng thuật toán trên Matlab/Simulink để kiểm chứng và đánh giá thuật toán. Từ đó đề xuất thuật toán mới nhằm nâng cao chất lượng điều khiển FMWR:

- Áp dụng một số thuật toán điều khiển truyền thống vào điều khiển bám quỹ đạo cho robot FMWR:

- + Bộ điều khiển PID
- + Điều khiển trượt SMC
- + Điều khiển Backstepping + SMC
- + Điều khiển mặt trượt động DSC

- **Đề xuất thuật toán điều khiển mặt trượt động chỉnh định thích nghi mờ để nâng cao chất lượng điều khiển bám quỹ đạo cho robot tự hành đa hướng bốn bánh xe mecanum có trọng tâm thay đổi.** Thuật toán này không chỉ giúp cải thiện khả năng bám quỹ đạo trong các điều kiện thay đổi về vị trí trọng tâm mà còn nâng cao tính ổn định và độ chính xác trong suốt quá trình di chuyển. Việc sử dụng bộ chỉnh định mờ cho phép thuật toán thích nghi chỉnh định các tham số điều khiển trong thời gian thực, phù hợp với sự thay đổi về khối lượng và vị trí trọng tâm, từ đó đạt được hiệu quả điều khiển tốt hơn.

Kết quả nghiên cứu trong chương 2 đã được công bố trong các công trình sau:

[CT01] **Dong Nguyen Minh**, Tien Ngo Manh, Hiep Do Quang, Nam Duy Buy, Linh Bui Van, Cuong Nguyen Manh, “*Backstepping Aggregated with Sliding Mode Control Algorithm for Mecanum Wheeled Mobile Robot*” Proceedings the 6th Vietnam International Conference and Exhibition on Control and Automation VCCA-2021, ISBN 978-604-73-5569-3.

[CT02] **Nguyễn Minh Đông**, Ngô Mạnh Tiến, Đỗ Quang Hiệp, Bùi Văn Bắc, Chu Văn Vương & Nguyễn Đức Thắng (2022). “*Xây dựng bộ điều khiển mặt trượt động điều khiển bám quỹ đạo cho robot di động đa hướng bốn bánh Mecanum*” Journal of Military Science and Technology, (FEE), 41–49.

[CT03] **Nguyễn Minh Đông**, Đỗ Quang Hiệp, Ngô Mạnh Tiến, Bùi Văn Bắc & Chu Văn Vương, “*Xây dựng thuật toán điều khiển thích nghi mờ mặt trượt động điều khiển bám quỹ đạo cho mecanum robot*” Hội thảo quốc gia lần thứ XXV: Một số vấn đề chọn lọc của Công nghệ thông tin và truyền thông – Hà nội 8-9/12/2022.

[CT04] **Minh Dong Nguyen**, Hiep, D. Q., Nam, D. P., & Tien, N. M. (2025). “*Trajectory tracking control for four-mecanum-wheeled robots considering mass eccentricity: A fuzzy adaptive and dynamic surface control approach*”. Journal of Computer Science and Cybernetics, (1). <https://doi.org/10.15625/1813-9663/21537>

CHƯƠNG 3. THUẬT TOÁN HỌC TĂNG CƯỜNG ĐIỀU KHIỂN BÁM QUỠ ĐẠO CHO ROBOT DI ĐỘNG ĐA HƯỚNG MECANUM

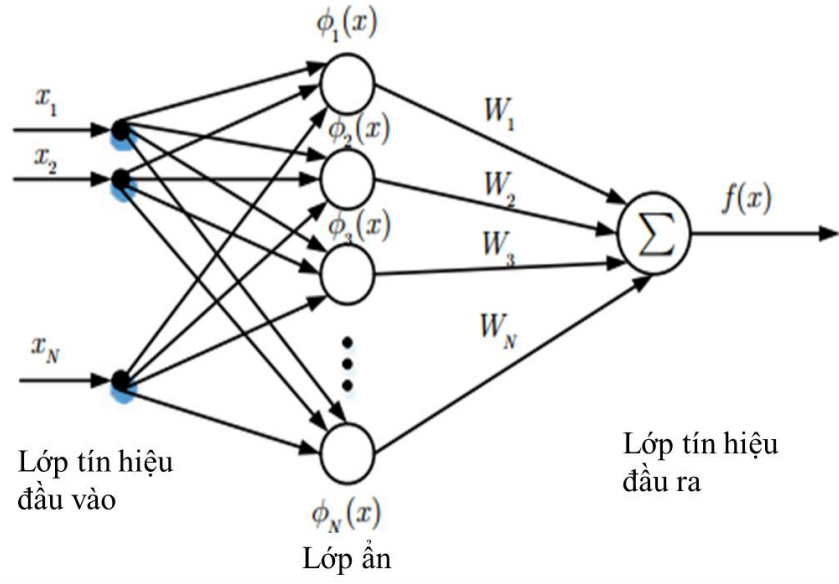
Trong chương 2, thuật toán Fuzzy-DSC được đề xuất là một giải pháp mạnh mẽ trong điều khiển bám quỹ đạo cho robot di động đa hướng, đặc biệt khi hệ thống là phi tuyến và có các yếu tố bất định. Thuật toán có tốc độ hội tụ nhanh và ổn định với các nhiễu nhỏ khi các tham số và luật mờ được thiết kế phù hợp. Tuy nhiên, thuật toán có khả năng thích nghi hạn chế trong môi trường thay đổi hoặc có nhiễu bất định không dự đoán trước, vì các luật mờ thường được thiết kế trước và khó tự động điều chỉnh trong thời gian thực. Việc thiết kế luật mờ và lựa chọn các tham số điều khiển là một nhiệm vụ khó khăn, phụ thuộc vào kinh nghiệm và kỹ năng của người thiết kế. Bên cạnh đó việc đưa các ràng buộc giới hạn của mô hình robot vào thuật toán để chứng minh ổn định thì rất phức tạp. Do vậy thuật toán học tăng cường được đề xuất áp dụng điều khiển bám quỹ đạo cho FMWR, để cải thiện khả năng thích nghi với các thay đổi của môi trường thay đổi hoặc chịu các nhiễu bất định của mô hình robot. Thuật toán học tăng cường sử dụng cấu trúc Actor-Critic giúp robot học từ các trải nghiệm thực tế và tối ưu hóa hành động để đạt được mục tiêu bám quỹ đạo mà không cần biết chính xác về các yếu tố bất định của mô hình. Khi môi trường thay đổi hoặc có các bất định không xác định trước thì robot có thể tự học cách xử lý các yếu tố này mà không yêu cầu mô hình chính xác của các yếu tố bất định.

Trong chương này, luận án sẽ trình bày cơ sở phương pháp nghiên cứu của thuật toán học tăng cường với cấu trúc actor-critic, từ đó đề xuất áp dụng vào điều khiển bám quỹ đạo cho FMWR và mô phỏng với các kịch bản khác nhau đánh giá chất lượng thuật toán.

3.1. Cơ sở phương pháp luận nghiên cứu

3.1.1. Mạng nơ-ron xấp xỉ hàm

Trong thuật toán học tăng cường, mạng nơ-ron dùng để xấp xỉ hàm có cấu trúc đơn giản sử dụng mạng MLP (Multi-layer Perceptron) [91] truyền thẳng một lớp ẩn, có một đầu ra duy nhất được biểu diễn trong hình 3.1.



Hình 3.1 Mạng MLP một lớp ẩn [91]

Lựa chọn hàm đầu ra của mạng nơ-ron được xác định theo [92],[93] như sau:

$$\hat{f}(\mathbf{x}) = \sum_{i=1}^N \mathbf{w}_i \phi_i(\mathbf{x}) = \hat{\mathbf{W}}^T \phi(\mathbf{x}) \quad (3.1)$$

trong đó vector đầu vào: $\mathbf{x} \in \Omega_x \subset \mathbb{R}^n$, Vector trọng số $\hat{\mathbf{W}} = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_N]^T \in \mathbb{R}^N$

Theo kết quả nghiên cứu của các công trình [94] đối với một hàm liên tục bất kỳ $f(\mathbf{x}) : \Omega_x \rightarrow \mathbb{R}$, trong đó $\Omega_x \subset \mathbb{R}^n$ là một tập compact, luôn tồn tại một NN sao cho hàm $f(\mathbf{x})$ được xấp xỉ:

$$f(\mathbf{x}) = \mathbf{W}^T \phi(\mathbf{x}) + \varepsilon(\mathbf{x}), \forall \mathbf{x} \in \Omega_x \quad (3.2)$$

Trong đó: $\mathbf{W}$ là trọng số lý tưởng của mạng nơ-ron, N số lượng nút ở lớp ẩn. $\phi(\mathbf{x}) = [\phi_1(\mathbf{x}), \phi_2(\mathbf{x}), \dots, \phi_N(\mathbf{x})]^T \in \mathbb{R}^N$ véc-tơ hàm kích hoạt và $\varepsilon(\mathbf{x})$ là sai số xấp xỉ NN. Sai số xấp xỉ ε sẽ hội tụ $\varepsilon \rightarrow 0$ khi $N \rightarrow \infty$. Ngoài ra khi N hữu hạn ε và $\frac{\partial \varepsilon}{\partial \mathbf{x}}$ sẽ bị chặn bởi các hằng số dương trong tập đóng.

Kết hợp phương trình (3.1) và (3.2) ta được:

$$\Delta f = f(\mathbf{x}) - \hat{f}(\mathbf{x}) = \tilde{\mathbf{W}}^T \phi(\mathbf{x}) + \varepsilon, \quad \forall \mathbf{x} \in \Omega_x \quad (3.3)$$

Trong đó: $\hat{\mathbf{W}} = \mathbf{W} - \hat{\mathbf{W}}$, $\hat{\mathbf{W}}$ là ước lượng của $\mathbf{W}$. Để cho hệ thống hội tụ và ổn định thì điều kiện PE là điều kiện rất quan trọng. Các trọng số mạng NN và nhận dạng chính xác động học hệ thống [95]. Tuy nhiên, về thực tế điều kiện PE rất khó kiểm tra, hơn nữa điều kiện PE chỉ là điều kiện đủ.

3.1.2. Ổn định bị chặn tới hạn đều (UUB- Uniformly Ultimate Bounded)

Định nghĩa 1.1 Xét hệ phi tuyến sau [96]:

$$\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}) \quad (3.4)$$

Nghiệm của phương trình (3.4) là:

- UB nếu tồn tại hằng số dương c độc lập với $t_0 \geq 0$ và $\forall a \in (0, c)$ thì tồn tại $\beta = \beta(a) > 0$ độc lập với t_0 sao cho:

$$\|\mathbf{x}(t_0)\| \leq a \Rightarrow \|\mathbf{x}(t)\| \leq \beta, \quad \forall t \geq t_0 \quad (3.5)$$

- GUB nếu (3.5) thỏa mãn $\forall a$ lớn tùy ý.
- UUB với biên chặn b nếu tồn tại các số dương b và c , độc lập với $t_0 \geq 0$ và $\forall a \in (0, c)$, thì tồn tại $T = T(a, b) \geq 0$, độc lập với t_0 , sao cho:

$$\|\mathbf{x}(t_0)\| \leq a \Rightarrow \|\mathbf{x}(t)\| \leq b, \quad \forall t \geq t_0 + T \quad (3.6)$$

- GUUB nếu (3.6) cho $\forall a$ lớn tùy ý.

Trong trường hợp của hệ kín là hệ dừng, nghiệm của hệ chỉ phụ thuộc vào $t - t_0$, do đó có thể bỏ thuật ngữ “Uniformly” nên kết quả chỉ cần là UB (Ultimate Bounded).

Định lý 1.1 [96] Gọi $D \subset \mathbb{R}^n$ là miền chứa gốc và $V : [0, \infty) \times D \rightarrow \mathbb{R}$ là một hàm có đạo hàm liên tục sao cho:

$$\alpha_1(\|\mathbf{x}\|) \leq V(t, \mathbf{x}) \leq \alpha_2(\|\mathbf{x}\|) \quad (3.7)$$

$$\frac{\partial V}{\partial t} + \frac{\partial V}{\partial \mathbf{x}} \mathbf{f}(t, \mathbf{x}) \leq -W_3(\mathbf{x}), \quad \forall \|\mathbf{x}\| \geq \mu > 0 \quad (3.8)$$

$\forall t \geq 0$ và $\forall \mathbf{x} \in D$, trong đó α_1 và α_2 là những hàm thuộc lớp $\mathcal{K}$ và $W_3(\mathbf{x})$ là một hàm liên tục xác định dương. Gọi $r > 0$ sao cho $B_r \subset D$ và giả thiết rằng:

$$\mu < \alpha_2^{-1}(\alpha_1(r)) \quad (3.9)$$

Tồn tại β là hàm thuộc lớp $\mathcal{KL}$ và với mọi trạng thái đầu $\mathbf{x}(t_0)$, thỏa mãn $\|\mathbf{x}(t_0)\| \leq \alpha_2^{-1}(\alpha_1(r))$, có $T \geq 0$ (không phụ thuộc vào $\mathbf{x}(t_0)$ và μ) sao cho nghiệm của (3.4) thỏa mãn:

$$\|\mathbf{x}(t)\| \leq \beta(\|\mathbf{x}(t_0)\|, t - t_0), \quad \forall t_0 \leq t \leq t_0 + T \quad (3.10)$$

$$\|\mathbf{x}(t)\| \leq \alpha_1^{-1}(\alpha_2(\mu)), \quad \forall t \geq t_0 + T \quad (3.11)$$

Các bất phương trình (3.10) và (3.11) cho biết rằng $\mathbf{x}(t)$ bị chặn đều với mọi $t \geq t_0$ và UUB (Uniformly Ultimate Bounded) với biên chặn $\alpha_1^{-1}(\alpha_2(\mu))$. Biên chặn là hàm lớp $\mathcal{K}$ của μ , do vậy giá trị của μ càng nhỏ thì giá trị của biên chặn càng nhỏ. Khi $\mu \rightarrow 0$ thì biên chặn tiến tới 0.

Nhận xét:

Tiêu chuẩn ổn định ISS (Input-to-State Stability) và tiêu chuẩn ổn định UUB có điểm giống nhau các nghiệm của hệ (3.4) đều tiến về lân cận (tập bất biến) của gốc, lân cận này lớn hay nhỏ phụ thuộc vào hệ và bộ điều khiển được thiết kế cho hệ. Trong nội dung của Định lý 1.1 thể hiện rằng mọi hệ ổn định ISS thì đều ổn định UUB nhưng có điểm khác biệt tiêu chuẩn ISS không đề cập tới thời gian đi về lân cận của gốc, trong khi đó tiêu chuẩn UUB đề cập tới thời gian hữu hạn khi tiến về lân cận của gốc, điều này có ý nghĩa quan trọng đảm bảo sau các bước cập nhật NN hữu hạn hệ đi vào ổn định, các trọng số của mạng NN hội tụ.

3.2. Thuật toán quy hoạch động (ADP- Adaptive Dynamic Programming)

Bài toán điều khiển tối ưu không chỉ đòi hỏi phải đảm bảo mục đích tối ưu mà còn đảm bảo tính ổn định của hệ thống. Phương trình HJB là phương trình vi phân đạo hàm riêng phi tuyến, trong trường hợp đặc biệt hệ là tuyến tính thì phương trình HJB trở thành phương trình đại số Riccati. Thuật toán quy hoạch động (ADP) được sử dụng để tìm nghiệm xấp xỉ của phương trình HJB [94] thông qua các giải thuật vòng lặp PI để giải phương trình Lyapunov (LE) [97]. Thuật toán ADP sử dụng hai mạng nơ-ron để xấp xỉ nghiệm phương trình HJB dựa trên cấu trúc mạng nơ-ron nhân tạo Actor và Critic. Trong đó mạng Actor NN dùng để xấp xỉ luật điều khiển tối ưu,

Critic NN dùng để xấp xỉ hàm Bellman. Cấu trúc AC-NNs được sử dụng trong các công trình nghiên cứu [94], [97] dựa trên luật cập nhật tuần tự tham số của mạng NN. Hay các nghiên cứu [74, 75, 98, 99] tiến hành cập nhật các tham số song song hay cập nhật trực tuyến.

Xét hệ thống phi tuyến sau:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u} \quad (3.12)$$

Trong đó $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{f}(\mathbf{x}) \in \mathbb{R}^n$, $\mathbf{g}(\mathbf{x}) \in \mathbb{R}^{n \times m}$ là những véc-tơ hàm phi tuyến, $\mathbf{u} \in \mathbb{R}^m$ là đầu vào điều khiển.

Giả thiết 1.1 Hệ (3.12) có nghiệm và có điểm cân bằng là gốc.

Hàm chi phí của hệ được định nghĩa:

$$J(\mathbf{x}, \mathbf{u}) = \int_0^{\infty} r(\mathbf{x}(t), \mathbf{u}(t)) dt \quad (3.13)$$

Giả sử rằng tồn tại một luật điều khiển để tối thiểu hóa hàm chi phí và hàm Bellman $V^* : \mathbb{R}^n \rightarrow \mathbb{R}_{\geq 0}$ được định nghĩa [74]:

$$V^*(\mathbf{x}(t)) = \min_{\mathbf{u} \in \Psi(\Omega)} \int_t^{\infty} r(\mathbf{x}, \mathbf{u}) dt \quad (3.14)$$

Trong đó $\Psi(\Omega)$ là một tập của các luật chấp nhận được, $r(\mathbf{x}, \mathbf{u}) \in \mathbb{R}$ được định nghĩa:

$$r(\mathbf{x}, \mathbf{u}) = q(\mathbf{x}) + \mathbf{u}^T \mathbf{R} \mathbf{u} = \mathbf{x}^T \mathbf{Q}_T \mathbf{x} + \mathbf{u}^T \mathbf{R} \mathbf{u} \quad (3.15)$$

Với $q(\mathbf{x}) \in \mathbb{R}^+$; $q(\mathbf{x}) = \mathbf{x}^T \mathbf{Q}_T \mathbf{x}$; xác định dương, $\mathbf{R} \in \mathbb{R}^{m \times m}$ là ma trận đối xứng xác định dương. Sử dụng kết quả trong nghiên cứu [74] thu được:

$$\min_{\mathbf{u} \in \Psi(\Omega)} [r(\mathbf{x}, \mathbf{u}) + \frac{\partial V^*(\mathbf{x})}{\partial \mathbf{x}} (\mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u})] = 0 \quad (3.16)$$

Định nghĩa hàm Hamilton tối ưu:

$$H(\mathbf{x}, \mathbf{u}, \frac{\partial V^*}{\partial \mathbf{x}}) = r(\mathbf{x}, \mathbf{u}) + \frac{\partial V^*(\mathbf{x})}{\partial \mathbf{x}} (\mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u}) \quad (3.17)$$

Từ phương trình (3.16) và (3.17) ta có phương trình HJB sau:

$$\min_{\mathbf{u} \in \Psi(\Omega)} \left[H(\mathbf{x}, \mathbf{u}^*, \frac{\partial V^*}{\partial \mathbf{x}}) \right] = 0 \quad (3.18)$$

Luật điều khiển tối ưu được xác định theo điều kiện: $\partial H(\mathbf{x}, \mathbf{u}, \frac{\partial V^*}{\partial \mathbf{x}}) / \partial \mathbf{u} = 0$

$$\mathbf{u}^*(\mathbf{x}) = -\frac{1}{2} \mathbf{R}^{-1} \mathbf{g}^T(\mathbf{x}) \left(\frac{\partial V^*(\mathbf{x})}{\partial \mathbf{x}} \right)^T \quad (3.19)$$

Thay luật điều khiển tối ưu (3.19) vào phương trình (3.16) thu được phương trình HJB có dạng như sau:

$$\begin{cases} q(\mathbf{x}) + \left(\frac{\partial V^*(\mathbf{x})}{\partial \mathbf{x}} \right) \mathbf{f}(\mathbf{x}) - \frac{1}{4} \left(\frac{\partial V^*(\mathbf{x})}{\partial \mathbf{x}} \right) \mathbf{g}(\mathbf{x}) \mathbf{R}^{-1} \mathbf{g}^T(\mathbf{x}) \left(\frac{\partial V^*(\mathbf{x})}{\partial \mathbf{x}} \right)^T = 0 \\ V^*(\mathbf{0}) = 0 \end{cases} \quad (3.20)$$

Phương trình vi phân đạo hàm riêng phi tuyến HJB (3.20) rất khó giải để tìm được nghiệm $V^*(\mathbf{x})$ bằng các phương pháp giải tích thông thường. Từ đó đề xuất giải thuật ADP với cấu trúc điều khiển AC-NNs được phân tích sau đây là phương pháp xấp xỉ $V^*(\mathbf{x})$ và luật điều khiển tối ưu $\mathbf{u}^*(\mathbf{x})$.

- *Điều khiển tối ưu với yếu tố chiết khấu và phương trình HJB hoàn thiện*

Việc sử dụng RISE, các tham số bất định hoặc nhiễu có thể được xử lý bằng cách ước lượng ảnh hưởng lên hệ thống thông qua phương trình (3.22) khi đó sai lệch sẽ hội tụ tiến về 0. Phương trình tổng quát hệ thống được mô tả như sau:

$$\frac{d}{dt} \mathbf{x} = \mathbf{F}(\mathbf{x}) + \mathbf{G}(\mathbf{x}) \hat{\mathbf{u}}_r \quad (3.21)$$

Phương trình ước lượng nhiễu đầu vào [100]:

$$\chi(t) = (k_s + 1)(e_2(t) - e_2(0)) + \int_0^t ((k_s + 1)\lambda_2 e_2(\tau) + \beta_1 \text{sgn}(e_2(\tau))) d\tau \quad (3.22)$$

Mục tiêu của bài toán điều khiển tối ưu vô hạn là tìm một chiến lược điều khiển tối ưu $\mathbf{u}^*(\mathbf{x})$ sao cho vừa đạt được mục tiêu bám quỹ đạo vừa tối thiểu hóa hàm chi phí sau với một yếu tố chiết khấu dương $\lambda > 0$:

$$V(\mathbf{x}(t), \hat{\mathbf{u}}_{rl}(t)) = \int_t^\infty e^{-\lambda(s-t)} U(\mathbf{x}(s), \hat{\mathbf{u}}_{rl}(s)) ds \quad (3.23)$$

Trong đó: $U(\mathbf{x}, \hat{\mathbf{u}}_{rl}) \triangleq \mathbf{x}^T \bar{\mathbf{Q}} \mathbf{x} + \hat{\mathbf{u}}_{rl}^T \mathbf{R} \hat{\mathbf{u}}_{rl}$, $\mathbf{R} \in \mathbb{R}^{n \times n}$ là ma trận hằng số đối xứng xác định

đương, và $\bar{\mathbf{Q}}$ được định nghĩa như sau: $\bar{\mathbf{Q}} = \begin{bmatrix} \mathbf{Q} & \mathbf{0}_{2n \times 2n} \\ \mathbf{0}_{2n \times 2n} & \mathbf{0}_{2n \times 2n} \end{bmatrix}$

Trong đó: $\bar{\mathbf{Q}} \in \mathbb{R}^{2n \times 2n}$ là một ma trận xác định dương, có vai trò điều chỉnh trọng số của trạng thái $\mathbf{x}$. Việc thêm hệ số chiết khấu λ vào chỉ số hiệu suất (3.23) là để giữ tín hiệu điều khiển $\hat{\mathbf{u}}_{rl}$ ổn định khi thời gian tiến đến vô cùng. Bằng cách chèn quỹ đạo đặt vào véc tơ trạng thái $\mathbf{x} = [\mathbf{z}^T \quad \boldsymbol{\eta}_d^T \quad \dot{\boldsymbol{\eta}}_d^T]^T \in \mathbb{R}^{4n}$, thách thức của việc phát triển thuật toán RL cho các hệ thống thay đổi theo thời gian được thực hiện. Do đó, hàm Bellman đối với thời gian t tùy ý được biểu diễn bằng hàm tĩnh $V^*(\mathbf{x}(t))$ như sau:

$$V^*(\mathbf{x}(t)) = \min_{\hat{\mathbf{u}}_{rl}(\mathbf{x}(t)) \in Y(U)} V(\mathbf{x}(t), \hat{\mathbf{u}}_{rl}(\mathbf{x}(t))) \quad (3.24)$$

Hàm Hamiltonian hoàn thiện với hệ số chiết khấu λ được xây dựng bằng cách lấy đạo hàm thời gian của hàm Bellman tĩnh (3.24) theo hai phương pháp khác nhau. Đầu tiên là cách suy ra trực tiếp như sau:

$$\frac{d}{dt} V^*(\mathbf{x}(t)) = \frac{\partial V^*}{\partial \mathbf{x}} \frac{d\mathbf{x}}{dt} = \frac{\partial V^*}{\partial \mathbf{x}} (F(\mathbf{x}) + G(\mathbf{x}) \mathbf{u}^*) \quad (3.25)$$

Cách xác định thứ hai là tính toán bằng phương pháp lập trình động (DP) cho hàm Bellman (3.24) tại thời điểm tức thời để đạt được phương trình sau:

$$V^*(\mathbf{x}(t)) = \int_t^{t+\Delta} e^{-\lambda(s-t)} U(\mathbf{x}(s), \mathbf{u}^*(\mathbf{x}(s))) ds + e^{-\lambda\Delta} V^*(\mathbf{x}(t+\Delta)) \quad (3.26)$$

Do đó, xác định được:

$$\frac{V^*(\mathbf{x}(t)) - V^*(\mathbf{x}(t+\Delta))}{\Delta} = \frac{1}{\Delta} \int_t^{t+\Delta} e^{-\lambda(s-t)} U(\mathbf{x}(s), \mathbf{u}^*(\mathbf{x}(s))) ds + \frac{(e^{-\lambda\Delta} - 1)}{\Delta} V^*(\mathbf{x}(t+\Delta)) \quad (3.27)$$

Khi $\Delta \rightarrow 0$ (khoảng thời gian cực nhỏ), bài toán tối ưu được giải lặp đi lặp lại cho mỗi bước thời gian, đảm bảo tín hiệu điều khiển được điều chỉnh thu được $V^*(\mathbf{x}(t))$ từ $\mathbf{u}^*(t)$:

$$U(\mathbf{x}(t), \mathbf{u}^*(t)) - \lambda V^*(\mathbf{x}(t)) + \frac{\partial V^*}{\partial \mathbf{x}} (F(\mathbf{x}) + \mathbf{G}(\mathbf{x})\mathbf{u}^*) = 0 \quad (3.28)$$

Bài toán tối ưu hóa được đưa ra để có được hàm nghịch đảo $V^*(\mathbf{x}(t)) \rightarrow \mathbf{u}^*(t)$ bằng cách sử dụng hàm giá trị tối ưu $V^*(\mathbf{x}(t))$ và phương pháp (DP) như sau:

$$V^*(\mathbf{x}(t)) = \min_{\hat{\mathbf{u}}_{rl}(\mathbf{x}) \in Y(\mathbf{u})} \left(\int_t^{t+\Delta} U(\mathbf{x}, \hat{\mathbf{u}}_{rl}) ds + e^{-\lambda\Delta} V^*(\mathbf{x}(t+\Delta)) \right) \quad (3.29)$$

Với $\Delta \rightarrow 0^+$, bài toán tối ưu sau được đưa ra để giải quyết hàm nghịch đảo $V^*(\mathbf{x}(t)) \rightarrow \mathbf{u}^*(t)$:

$$\min_{\hat{\mathbf{u}}_{rl}(\mathbf{x}) \in Y(U)} \left(U(\mathbf{x}, \hat{\mathbf{u}}_{rl}) - \lambda V^*(\mathbf{x}) + \frac{\partial V^*}{\partial \mathbf{x}} (F(\mathbf{x}) + \mathbf{G}(\mathbf{x})\hat{\mathbf{u}}_{rl}) \right) = 0 \quad (3.30)$$

Bằng cách sử dụng hàm Hamiltonian đã sửa đổi với việc bổ sung hệ số chiết khấu $\lambda > 0$ ta được:

$$H(\mathbf{x}, \hat{\mathbf{u}}_{rl}, \nabla V^*, V^*) = \mathbf{x}^T \bar{Q} \mathbf{x} + \hat{\mathbf{u}}_{rl}^T(\mathbf{x}) \mathbf{R} \hat{\mathbf{u}}_{rl}(\mathbf{x}) - \lambda V^*(\mathbf{x}) + \nabla V^{*T}(\mathbf{x}) (F(\mathbf{x}) + \mathbf{G}(\mathbf{x})\hat{\mathbf{u}}_{rl}(\mathbf{x})) \quad (3.31)$$

Với $\nabla V^*(\mathbf{x}) \triangleq \frac{\partial V^*(\mathbf{x})}{\partial \mathbf{x}}$. Mặt khác, theo (3.30) và (3.31) thì tín hiệu điều khiển tối

ưu $\mathbf{u}^*(\mathbf{x})$ sau đó thu được như sau:

$$\mathbf{u}^*(\mathbf{x}) = \underset{\hat{\mathbf{u}}_{rl}(\mathbf{x}) \in Y(\Omega)}{\operatorname{argmin}} \left[H(\mathbf{x}, \hat{\mathbf{u}}_{rl}, \nabla V^*, V^*) \right] = -\frac{1}{2} \mathbf{R}^{-1} \mathbf{G}^T(\mathbf{x}) \nabla V^*(\mathbf{x}) \quad (3.32)$$

Ngoài ra, theo quan điểm của điều khiển tối ưu $\mathbf{u}^*(\mathbf{x})$ (25) và bài toán tối ưu hóa (23), phương trình đạo hàm riêng (PDE) sau khi được thiết lập như sau:

$$\begin{aligned} H^*(\mathbf{x}, \hat{\mathbf{u}}_{rl}, \nabla V^*, V^*) &= \\ &= \mathbf{x}^T \bar{Q} \mathbf{x} - \lambda V^*(\mathbf{x}) - \frac{1}{4} \nabla V^{*T}(\mathbf{x}) \mathbf{G}(\mathbf{x}) \mathbf{R}^{-1} \mathbf{G}^T(\mathbf{x}) \nabla V^*(\mathbf{x}) + \nabla V^{*T}(\mathbf{x}) F(\mathbf{x}) = 0 \end{aligned} \quad (3.33)$$

Tuy nhiên, không thể giải bài toán tích phân đạo hàm riêng của phương trình (3.33) bằng giải tích thông thường. Do đó, áp dụng thuật toán actor- critic để giải và tìm tín hiệu điều khiển tối ưu.

Nhận xét: Việc sử dụng hệ số chiết khấu trong hàm chi phí (3.23) là cần thiết. Thực tế quỹ đạo tham chiếu $\mathbf{\eta}_d$ trong phương trình $\mathbf{x} = [\mathbf{z}^T \quad \mathbf{\eta}_d^T \quad \dot{\mathbf{\eta}}_d^T]^T \in \mathbb{R}^{4n}$ có thể không hội tụ về 0 trong các hệ thống thực tế. Do đó, hàm chi phí (3.23) sẽ trở thành vô cùng sau khi loại bỏ hệ số chiết khấu λ , trong khi giá trị hàm tối ưu $V^*(\mathbf{x}(t))$ cần phải có giá trị hữu hạn. Ngoài ra, khác với chỉ số hiệu suất bậc hai [75, 101], xem xét một hệ số chiết khấu bổ sung λ trong hàm chi phí (3.23) để duy trì tín hiệu điều khiển $\hat{\mathbf{u}}_{rl}$ khi thời gian tiến đến vô cùng. Vì vậy, hàm Hamiltonian hoàn thiện (3.31) được xây dựng bằng cách đạo hàm theo thời gian của hàm Bellman với hai phương pháp tiếp cận khác nhau như đã nêu trên.

Hàm chi phí tối ưu và luật điều khiển tối ưu sử dụng NN để xấp xỉ, được định nghĩa:

$$V^*(\mathbf{x}) = \mathbf{W}^T \phi(\mathbf{x}) + \varepsilon(\mathbf{x}) \quad (3.34)$$

$$\mathbf{u}^*(\mathbf{x}) = -\frac{1}{2} \mathbf{R}^{-1} \mathbf{g}^T(\mathbf{x}) \left(\left(\frac{\partial \phi}{\partial \mathbf{x}} \right)^T \mathbf{W} + \left(\frac{\partial \varepsilon}{\partial \mathbf{x}} \right)^T \right) \quad (3.35)$$

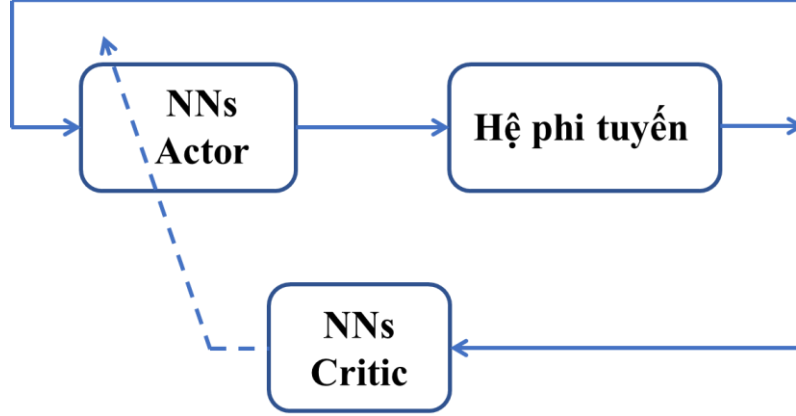
Trong đó $\mathbf{W} \in \mathbb{R}^N$ là các trọng số lý tưởng của NN, với N là số nơ-ron, $\phi(\mathbf{x}) = [\phi_1(\mathbf{x}), \phi_2(\mathbf{x}), \dots, \phi_N(\mathbf{x})]^T \in \mathbb{R}^N$ là hàm kích hoạt của NN và $\varepsilon \in \mathbb{R}$ là sai số xấp xỉ NN.

Critic NN $\hat{V}(\mathbf{x})$ và actor NN $\hat{\mathbf{u}}(\mathbf{x})$ xấp xỉ hàm Bellman (3.34) và luật điều khiển tối ưu (3.35) được định nghĩa:

$$\hat{V}(\mathbf{x}) = \hat{\mathbf{W}}_c^T \phi(\mathbf{x}) \quad (3.36)$$

$$\hat{\mathbf{u}}(\mathbf{x}) = \frac{1}{2} \mathbf{R}^{-1} \mathbf{g}^T(\mathbf{x}) \left(\frac{\partial \phi(\mathbf{x})}{\partial \mathbf{x}} \right)^T \hat{\mathbf{W}}_a \quad (3.37)$$

Trong đó $\hat{\mathbf{W}}_c \in \mathbb{R}^N$ và $\hat{\mathbf{W}}_a \in \mathbb{R}^N$ là các xấp xỉ trọng số lý tưởng của các NN tương ứng. Sai lệch trọng số của các actor NN và critic NN được định nghĩa $\tilde{\mathbf{W}}_c \triangleq \mathbf{W} - \hat{\mathbf{W}}_c \in \mathbb{R}^N$ và $\tilde{\mathbf{W}}_a \triangleq \mathbf{W} - \hat{\mathbf{W}}_a \in \mathbb{R}^N$ tương ứng. Sử dụng gradient của xấp xỉ hàm Bellman (3.34) [102] để xác định luật điều khiển tối ưu được xác định như sau:



Hình 3.2 Cấu trúc AC-NNs xấp xỉ nghiệm phương trình HJB

Sơ đồ cấu trúc mạng nơ-ron xấp xỉ nghiệm của phương trình HJB được trình bày trong hình 3.2. Tín hiệu điều khiển từ Actor sẽ liên tục được cập nhật và đưa vào kích thích hệ thống cho đến khi tín hiệu điều khiển là tối ưu. Các luật cập nhật của trọng số critic NN $\hat{\mathbf{W}}_c \in \mathbb{R}^N$ và actor NN $\hat{\mathbf{W}}_a \in \mathbb{R}^N$ được thực hiện đồng thời trong một bước lặp dựa theo tối thiểu hóa sai số hàm Bellman tương ứng:

$$\delta_{hjb} = H(\mathbf{x}, \hat{\mathbf{u}}, \frac{\partial \hat{V}}{\partial \mathbf{x}}) - H(\mathbf{x}, \mathbf{u}^*, \frac{\partial V^*}{\partial \mathbf{x}}) = H(\mathbf{x}, \hat{\mathbf{u}}, \frac{\partial \hat{V}}{\partial \mathbf{x}}) \quad (3.38)$$

Trong đó $H(\mathbf{x}, \mathbf{u}^*, \frac{\partial V^*}{\partial \mathbf{x}}) = 0$, sai số hàm Bellman có thể viết như sau:

$$\delta_{hjb} = \hat{\mathbf{W}}_c^T \mathbf{w}(\mathbf{x}, \hat{\mathbf{u}}) + r(\mathbf{x}, \hat{\mathbf{u}}) \quad (3.39)$$

Với $\mathbf{w}(\mathbf{x}, \hat{\mathbf{u}}) = \frac{\partial \phi}{\partial \mathbf{x}}(\mathbf{F}(\mathbf{x}) + \mathbf{G}(\mathbf{x})\hat{\mathbf{u}}) \in \mathbb{R}^N$ là véc-tơ hồi quy critic NN

a, Luật cập nhật cho Critic NN theo phương pháp bình phương tối thiểu

Gọi $E_c(\delta_{hjb}) \in \mathbb{R}^+$ biểu thị sai số tích phân bình phương sai lệch Bellman được tính như sau:

$$E_c = \int_0^t \delta_{hjb}^2(\tau) d\tau \quad (3.40)$$

Luật cập nhật cho trọng số critic NN được xác định như [75]:

$$\dot{\mathbf{W}}_c = -k_c \mathbf{\Gamma} \frac{\mathbf{w}}{1 + \gamma \mathbf{w}^T \mathbf{\Gamma} \mathbf{w}} \delta_{hjb} \quad (3.41)$$

Trong đó $k_c, \gamma \in \mathbb{R}$ là các số dương, $\mathbf{\Gamma} \triangleq \left(\int_0^t \mathbf{w}(\tau) \mathbf{w}^T(\tau) d\tau \right)^{-1} \in \mathbb{R}^{N \times N}$ là ma trận hệ số ước lượng đối xứng được tính và có thể bị chặn như sau:

$$\frac{d}{dt} \mathbf{\Gamma} = -k_c \mathbf{\Gamma} \frac{\mathbf{w}}{1 + \gamma \mathbf{w}^T \mathbf{\Gamma} \mathbf{w}} \mathbf{\Gamma}; \mathbf{\Gamma}(t_r^+) = \mathbf{\Gamma}(0) = \varphi_0 \mathbf{I} \quad (3.42)$$

Trong đó t_r^+ là thời gian khởi động lại tại thời điểm $\lambda_{\min}(\mathbf{\Gamma}(t)) \leq \varphi_0$, $\varphi_1 > \varphi_0 > 0$ Ma trận hiệp phương sai $\mathbf{\Gamma}(t)$ có thể bị chặn như sau:

$$\varphi_0 \mathbf{I} \leq \mathbf{\Gamma}(t) \leq \varphi_1 \mathbf{I} \quad (3.43)$$

b. Luật cập nhật cho Actor NN theo gradient

Cũng giống như việc cập nhật trọng số của critic NN dựa trên việc thu nhỏ sai số hàm Bellman $\delta_{hjb}(\cdot)$. Tuy nhiên, không giống như các trọng số của critic NN, các trọng số của actor NN biểu thị tính chất phi tuyến trong $\delta_{hjb}(\cdot)$, gây khó khăn khi phát triển một luật cập nhật theo cực tiểu hóa bình phương sai lệch (LS). Do đó, một luật cập nhật theo độ dốc (gradient) được phát triển cho actor NN để tối thiểu hóa bình phương sai lệch Bellman $E_a \triangleq \delta_{hjb}^2$, được định nghĩa như sau:

$$\frac{\partial E_a}{\partial \hat{\mathbf{W}}_a} = 2 \frac{\partial \delta_{hjb}}{\partial \hat{\mathbf{W}}_a} \delta_{hjb} = 2 \left(\hat{\mathbf{W}}_c^T \frac{\partial \phi}{\partial \mathbf{x}} \frac{\partial (\mathbf{f} + \mathbf{g}\hat{\mathbf{u}})}{\partial \hat{\mathbf{u}}} \frac{\partial \hat{\mathbf{u}}}{\partial \hat{\mathbf{W}}_a} + 2 \hat{\mathbf{u}}^T \mathbf{R} \frac{\partial \hat{\mathbf{u}}}{\partial \hat{\mathbf{W}}_a} \right) \delta_{hjb} \quad (3.44)$$

Luật cập nhật trọng số của actor NN được mô tả theo công thức [75]:

$$\dot{\hat{\mathbf{W}}}_a = -k_{a1} \frac{1}{\sqrt{1 + \mathbf{w}^T \mathbf{w}}} \left(\frac{\partial \phi}{\partial \mathbf{x}} \right) \mathbf{g} \mathbf{R}^{-1} \mathbf{g}^T \left(\frac{\partial \phi}{\partial \mathbf{x}} \right)^T (\hat{\mathbf{W}}_a - \hat{\mathbf{W}}_c) \delta_{hjb} - k_{a2} (\hat{\mathbf{W}}_a - \hat{\mathbf{W}}_c) \quad (3.45)$$

Với $k_{a1}, k_{a2} \in \mathbb{R}^+$ là các hằng số thích nghi dương.

Thuật toán 1.1: ADP cấu trúc AC-NNs cập nhật đồng bộ

Bước 1: Thực hiện khởi tạo luật điều khiển $\mathbf{u}^{(0)}(\mathbf{x})$ với giá trị $V^{(0)}(\mathbf{x}) = 0$ trong đó $\forall \mathbf{x} \in \Omega$.

✓ Gán $i = 0$.

Bước 2: Cập nhật đồng thời trọng số AC-NNs:

✓ Cập nhật trọng số critic NN theo (3.41)

✓ Cập nhật trọng số actor NN theo (3.45)

✓ Xác định hàm $\hat{V}^{(i+1)}(\mathbf{x})$ dựa vào critic NN (3.36) và $\hat{\mathbf{u}}^{(i+1)}(\mathbf{x})$ dựa vào actor NN (3.37)

Bước 3: Kiểm tra điều kiện kết thúc giải thuật:

- Nếu thỏa mãn tiêu chuẩn hội tụ sao cho $|\hat{V}^{(i+1)}(\mathbf{x}) - \hat{V}^{(i)}(\mathbf{x})| \leq \varepsilon$ trong đó ε là số dương đủ nhỏ thì tiến hành gán $\mathbf{u}^*(\mathbf{x}) = \hat{\mathbf{u}}^{(i+1)}(\mathbf{x})$ và $V^*(\mathbf{x}) = \hat{V}^{(i+1)}(\mathbf{x}) \rightarrow$ kết thúc giải thuật.

- Nếu không thỏa mãn, gán $i = i + 1$ và quay lại bước 2.

Nhận xét: Cấu trúc của giải thuật ADP sử dụng một mạng Critic NN để xấp xỉ hàm Bellman $\hat{V}(\mathbf{x})$ sau đó việc tính toán luật điều khiển $\hat{\mathbf{u}}(\mathbf{x})$ sử dụng đạo hàm Jacobi của hàm Bellman $\frac{\partial \hat{V}(\mathbf{x})}{\partial \mathbf{x}}$ đã xấp xỉ mà không cần tới mạng Actor NN để xấp xỉ luật điều khiển trong các công trình [103, 104]. Thuật toán có ưu điểm không phức tạp, quá trình tính toán đơn giản, nhưng bên cạnh đó thì việc đạo hàm của đại lượng được xấp xỉ nên quá trình xấp xỉ diễn ra nhanh thì dẫn tới đạo hàm của đại lượng sẽ sai số lớn, ảnh hưởng tới quá trình hội tụ của thuật toán. Sai số lớn làm cho sai lệch bám sẽ lớn ảnh hưởng tới mục tiêu giải quyết bài toán bám tối ưu. Vì vậy giải thuật ADP được nghiên cứu đề xuất sử dụng cấu trúc gồm critic NN dùng xấp xỉ hàm Bellman, actor NN dùng xấp xỉ luật điều khiển.

Thuật toán học tăng cường với giải thuật ADP, sử dụng cấu trúc On-Policy tín hiệu điều khiển từ Actor sẽ liên tục được cập nhật và đưa vào kích thích hệ thống cho đến khi tín hiệu điều khiển, thuật toán có ưu điểm ổn định khi học tập bởi vì các dữ liệu huấn luyện được cập nhật liên tục từ môi trường, thuật toán đơn giản hiệu quả

cao. Do vậy luận án sẽ tập trung vào nghiên cứu ứng dụng giải thuật ADP cấu trúc mạng AC on-policy để giải quyết bài toán điều khiển bám quỹ đạo cho FMWR có trọng tâm thay đổi.

3.3. Thiết kế thuật toán học tăng cường điều khiển bám quỹ đạo cho FMWR

3.3.1. Biến đổi mô hình động lực học FMWR

Từ mối liên hệ giữa vận tốc của robot trong hệ quy chiếu gắn với tâm và hệ quy chiếu quán tính được tính ở trong chương 2 ta có được:

$$\mathbf{v}_q = \mathbf{H}(\phi)\dot{\boldsymbol{\eta}} \quad (3.46)$$

Trong đó: $\mathbf{v}_q = \dot{\boldsymbol{\eta}}_R$ là vận tốc của robot, $\mathbf{H}(\phi)$ là ma trận chuyển từ hệ quy chiếu gắn với tâm robot sang hệ quy chiếu quán tính:

$$\mathbf{H}(\phi) = \begin{bmatrix} \cos \phi & \sin \phi & 0 \\ -\sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.47)$$

Lấy đạo hàm của phương trình (3.46) theo thời gian ta được:

$$\dot{\mathbf{v}}_q = \dot{\phi}\dot{\mathbf{H}}(\phi)\dot{\boldsymbol{\eta}} + \mathbf{H}(\phi)\ddot{\boldsymbol{\eta}} \quad (3.48)$$

Biến đổi phương trình động lực học (1.14) :

$$\ddot{\boldsymbol{\eta}} = \mathbf{M}^{-1}\mathbf{D}\boldsymbol{\tau} - \mathbf{M}^{-1}\mathbf{D}\boldsymbol{\delta} - \mathbf{M}^{-1}\mathbf{C}\dot{\boldsymbol{\eta}} \quad (3.49)$$

Kết hợp phương trình (3.49) vào (3.48) ta được:

$$\dot{\mathbf{v}}_q = \mathbf{H}(\phi)\mathbf{M}^{-1}\mathbf{D}\boldsymbol{\tau} - \mathbf{H}(\phi)\mathbf{M}^{-1}\mathbf{D}\boldsymbol{\delta} + [\dot{\phi}\dot{\mathbf{H}}(\phi)\mathbf{H}(\phi)^{-1} - \mathbf{M}^{-1}\mathbf{C}]\mathbf{v}_q \quad (3.50)$$

Đưa ra được phương trình động lực học cho FMWR có dạng:

$$\dot{\mathbf{v}}_q = \bar{\mathbf{f}}(\boldsymbol{\eta})\mathbf{v}_q + \bar{\mathbf{g}}(\boldsymbol{\eta})\boldsymbol{\tau} + \bar{\mathbf{g}}(\boldsymbol{\eta})d(\boldsymbol{\eta}, \mathbf{v}_q) \quad (3.51)$$

$$\bar{\mathbf{f}}(\boldsymbol{\eta}) = [\dot{\phi}\dot{\mathbf{H}}(\phi)\mathbf{H}(\phi)^{-1} - \mathbf{M}^{-1}\mathbf{C}]$$

Trong đó: $\bar{\mathbf{g}}(\boldsymbol{\eta}) = \mathbf{H}(\phi)\mathbf{M}^{-1}\mathbf{D}$

$$fd(\boldsymbol{\eta}, \mathbf{v}_q) = -\mathbf{H}(\phi)\mathbf{M}^{-1}\mathbf{D}\boldsymbol{\delta}$$

Để giải quyết bài toán cho robot di chuyển với quỹ đạo $\boldsymbol{\eta}(t)$ bám theo quỹ đạo đặt $\boldsymbol{\eta}_r(t)$ ta thực hiện:

$$\mathbf{x} = (\boldsymbol{\eta}^T, \mathbf{v}_q^T)^T,$$

Đặt biến: $f(\mathbf{x}) = (\mathbf{v}_q^T, (f(\boldsymbol{\eta})\mathbf{v}_q)^T)^T$, từ đó (3.51) được viết lại như sau:

$$g(\mathbf{x}) = (\mathbf{0}_{3 \times 3}, g(\boldsymbol{\eta})^T)^T$$

$$\dot{\mathbf{x}} = f(\mathbf{x}) + g(\mathbf{x})\boldsymbol{\tau} + g(\mathbf{x})d(\boldsymbol{\eta}, \mathbf{v}_q) \quad (3.52)$$

Để đạt được mục tiêu bám quỹ đạo, ta xét quỹ đạo đặt của hệ thống như sau:

$$\dot{\mathbf{x}}_r = h_r(\mathbf{x}_r) \quad (3.53)$$

Trong đó, $\mathbf{x}_r = (\boldsymbol{\eta}_r^T, \mathbf{v}_{qr}^T)^T$ và $\dot{\boldsymbol{\eta}}_r = \dot{\mathbf{v}}_r, h_r(\mathbf{x}_r)$ là một hàm liên tục chuyển ma trận Lipschitz, xác định $\mathbf{e} \triangleq \mathbf{x} - \mathbf{x}_r$ là sai lệch bám, tốc độ bám quỹ đạo được mô tả như sau:

$$\dot{\mathbf{e}} = f(\mathbf{x}) + g(\mathbf{x})\boldsymbol{\tau} - h_r(\mathbf{x}_r) = f(\mathbf{x}) - h_r(\mathbf{x}_r) + g(\mathbf{x})\boldsymbol{\tau} + g(\mathbf{x})\mathbf{u} \quad (3.54)$$

Trong đó: $\mathbf{u} = \boldsymbol{\tau} - \boldsymbol{\tau}_r$, $\boldsymbol{\tau}_r(\mathbf{x}_r) = g^+(\mathbf{x}_r)(h_r(\mathbf{x}_r) - f(\mathbf{x}_r))$ (3.55)

Đặt $\mathbf{z} = (\mathbf{e}^T, \mathbf{x}_r^T)^T$ là biến trạng thái mới, từ đó ta có mô hình của hệ thống được mô tả theo $\mathbf{z}$ như sau:

$$\dot{\mathbf{z}} = \mathbf{F}(\mathbf{z}) + \mathbf{G}(\mathbf{z})\mathbf{u} \quad (3.56)$$

$$\text{Với: } \mathbf{F}(\mathbf{z}) = \begin{pmatrix} f(\mathbf{x}) - h_r(\mathbf{x}_r) + g(\mathbf{x})\boldsymbol{\tau}_r' \\ h_r(\mathbf{x}_r) \end{pmatrix}, \quad \mathbf{G}(\mathbf{z}) = \begin{pmatrix} g(\mathbf{x}) \\ \mathbf{0}_{6 \times 4} \end{pmatrix}$$

Thực hiện biến đổi toán học từ bài toán điều khiển bám quỹ đạo cho robot có dạng (3.51) có thể chuyển thành bài toán tối ưu điều khiển $\mathbf{u}^*(\mathbf{z})$ như phương trình (3.56) để tạo ra các tín hiệu vòng kín của hệ thống nằm ở trạng thái ổn định. Bài toán được giải quyết bằng cách sử dụng phương trình HJB.

3.3.2. Bộ điều khiển tối ưu và phương trình HJB

Để điều khiển tối ưu cho hệ thống (3.56) ta xét hàm chi phí sau:

$$V(\mathbf{z}(t)) = \int_t^\infty \mathbf{e}^{-\gamma(s-t)} \mathbf{U}(\mathbf{z}(s), \mathbf{u}(s)) ds \quad (3.57)$$

Trong đó, $\gamma > 0$, $\mathbf{U}(\mathbf{z}, \mathbf{u}) = \lambda_M^2(\mathbf{z}) + \mathbf{z}^T \bar{\mathbf{Q}}\mathbf{z} + \mathbf{u}^T \mathbf{R}\mathbf{u}$, $\mathbf{R} \in \mathbf{R}^{3 \times 3}$ là một ma trận hằng số xác định dương, và $\bar{\mathbf{Q}}$ là ma trận được xác định như sau:

$$\bar{\mathbf{Q}} = \begin{pmatrix} \mathbf{Q} & \mathbf{0}_{6 \times 6} \\ \mathbf{0}_{6 \times 6} & \mathbf{0}_{6 \times 6} \end{pmatrix} \quad (3.58)$$

Trong đó, $\mathbf{Q} \in \mathbf{R}^{(6 \times 6)}$ là một ma trận xác định dương

Ta có phương trình vi phân $V(\mathbf{z}(t))$ như sau:

$$\lambda_M^2(\mathbf{z}) + \mathbf{z}^T \bar{\mathbf{Q}}\mathbf{z} + \mathbf{u}^T \mathbf{R}\mathbf{u} - \gamma V(\mathbf{z}) + \nabla V^T(\mathbf{z})(\mathbf{F}(\mathbf{z}) + \mathbf{G}(\mathbf{z})\mathbf{u}) = 0 \quad (3.59)$$

Trong đó, $\nabla V(\mathbf{z}) = ((\partial V(\mathbf{z})) / \partial \mathbf{z})$

Phương trình Hamiltonian được cho bởi:

$$\mathbf{H}(\mathbf{z}, \mathbf{u}, \nabla V) = \lambda_M^2(\mathbf{z}) + \mathbf{z}^T \bar{\mathbf{Q}}\mathbf{z} + \mathbf{u}^T(\mathbf{z})\mathbf{R}\mathbf{u}(\mathbf{z}) - \gamma V(\mathbf{z}) + \nabla V^T(\mathbf{z})(\mathbf{F}(\mathbf{z}) + \mathbf{G}(\mathbf{z})\mathbf{u}(\mathbf{z})) \quad (3.60)$$

Từ đó có thể xác định hàm chi phí tối ưu như sau:

$$\mathbf{V}^*(\mathbf{z}) = \min_{\mathbf{u} \in \pi(\Omega)} \int_t^\infty \mathbf{e}^{-\gamma(s-t)} (\lambda_M^2(\mathbf{z}) + \mathbf{z}^T \bar{\mathbf{Q}}\mathbf{z} + \mathbf{u}^T \mathbf{R}\mathbf{u}) \quad (3.61)$$

Trong đó, $\pi(\Omega)$ biểu thị một tập luật điều khiển chấp nhận được trên tập $\Omega \subset \mathbf{R}^{12}$.

Khi đó hàm chi phí tối ưu $\mathbf{V}^*(\mathbf{z})$ thỏa mãn điều kiện của phương trình HJB sau:

$$\begin{aligned} \mathbf{H}^*(\mathbf{z}, \mathbf{u}^*, \nabla V^*) &= \lambda_M^2(\mathbf{z}) + \mathbf{z}^T \bar{\mathbf{Q}}\mathbf{z} + \mathbf{u}^{*T}(\mathbf{z})\mathbf{R}\mathbf{u}^*(\mathbf{z}) - \gamma V^*(\mathbf{z}) \\ &+ \nabla V^{*T}(\mathbf{z})(\mathbf{F}(\mathbf{z}) + \mathbf{G}(\mathbf{z})\mathbf{u}^*(\mathbf{z})) = 0 \end{aligned} \quad (3.62)$$

Từ đó, tín hiệu điều khiển tối ưu được xác định như sau:

$$\mathbf{u}^*(\mathbf{z}) = \arg \min_{\mathbf{u} \in \pi(\Omega)} \left[\mathbf{H}(\mathbf{z}, \mathbf{u}, \nabla V^*(\mathbf{z})) \right] = -\frac{1}{2} \mathbf{R}^{-1} \mathbf{G}^T(\mathbf{z}) \nabla V^*(\mathbf{z}) \quad (3.63)$$

Thay tín hiệu điều khiển tối ưu vào (3.62) ta được phương trình HJB trở thành:

$$\begin{aligned} \mathbf{H}^*(\mathbf{z}, \nabla V^*) &= \lambda_M^2(\mathbf{z}) + \mathbf{z}^T \bar{\mathbf{Q}}\mathbf{z} - \gamma V^*(\mathbf{z}) - \frac{1}{4} \nabla V^{*T}(\mathbf{z}) \mathbf{G}(\mathbf{z}) \mathbf{R}^{-1} \mathbf{G}^T(\mathbf{z}) \nabla V^*(\mathbf{z}) \\ &+ \nabla V^{*T}(\mathbf{z}) \mathbf{F}(\mathbf{z}) = 0 \end{aligned} \quad (3.64)$$

Để có được tín hiệu điều khiển tối ưu (3.63) ta cần giải bài toán cực trị trong phương trình HJB, việc giải bài toán này là cực kỳ khó giải vậy nên ta áp dụng thuật toán actor- critic vào để giải phương trình này.

3.3.3. Thuật toán học tăng cường cấu trúc Actor- Critic

Trong phần này ta sử dụng thuật toán học tăng cường với cấu trúc Actor-Critic (RL-AC) kết hợp với mạng nơ-ron trong thời gian thực để thực hiện việc tìm nghiệm gần đúng của phương trình HJB.

Dựa trên định lý xấp xỉ bậc cao Weierstrass, hàm chi phí $\mathbf{V}^*(\mathbf{z})$ có thể được xấp xỉ bằng cách sử dụng mạng nơ-ron một lớp như sau:

$$\mathbf{V}^*(\mathbf{z}) = \mathbf{W}^T \phi(\mathbf{z}) + \varepsilon_v(\mathbf{z}) \quad (3.65)$$

Trong đó, $\mathbf{W} \in \mathbf{R}^N$ là một vector trọng số lý tưởng không đổi được giới hạn bởi một vector hằng dương đã biết $\bar{\mathbf{W}}$ sao cho $\|\mathbf{W}\| \leq \bar{\mathbf{W}}, \phi(\mathbf{z}) \in \mathbf{R}^N$ là một hàm kích hoạt phù hợp. N là số nơ-ron, $\varepsilon_v(\mathbf{z})$ là sai số xấp xỉ. Từ đó chúng ta xác định được gradient của $\mathbf{V}^*(\mathbf{z})$:

$$\nabla \mathbf{V}^*(\mathbf{z}) = \nabla \phi^T(\mathbf{z}) \mathbf{W} + \nabla \varepsilon_v(\mathbf{z}) \quad (3.66)$$

Đối với hàm kích hoạt $\phi(\mathbf{z})$ chúng ta có giả thiết: Hàm kích hoạt và gradient của nó bị giới hạn $\|\phi(\mathbf{z})\| \leq b_\phi, \|\nabla \phi(\mathbf{z})\| \leq b_{\phi_z}$ phương trình Hamiltonian tối ưu sẽ trở thành

$$\mathbf{H}^*(\mathbf{z}, \mathbf{W}) = \lambda_M^2(\mathbf{z}) + \mathbf{z}^T \bar{\mathbf{Q}} \mathbf{z} + \mathbf{W}^T \nabla \phi \mathbf{F} - \gamma \mathbf{W}^T \phi - \frac{1}{4} \mathbf{W}^T \mathbf{D}_1 \mathbf{W} + \varepsilon_H = 0 \quad (3.67)$$

Trong đó: $\mathbf{D}_1 = \nabla \phi \mathbf{G} \mathbf{R}^{-1} \mathbf{G}^T \nabla \phi^T, \mathbf{D}_2 = \mathbf{G} \mathbf{R}^{-1} \mathbf{G}^T$ và

$$\varepsilon_H = \nabla \varepsilon_v^T \mathbf{F} - \frac{1}{2} \nabla \varepsilon_v^T \mathbf{G} \mathbf{R}^{-1} \mathbf{G}^T \nabla \phi^T \mathbf{W} - \frac{1}{4} \nabla \varepsilon_v^T \mathbf{D}_2 \nabla \varepsilon_v - \gamma \varepsilon_v(\mathbf{z})$$

Tồn tại một giới hạn không đổi ε_h sao cho $\|\varepsilon_H\| \leq \varepsilon_h$ là số đơn vị lớp ẩn N tăng lên. Tín hiệu điều khiển tối ưu trở thành:

$$\mathbf{u}^*(\mathbf{z}) = -\frac{1}{2}\mathbf{R}^{-1}\mathbf{G}^T(\mathbf{z})(\nabla\phi^T(\mathbf{z})\mathbf{W} + \nabla\mathbf{\varepsilon}_v(\mathbf{z})) \quad (3.68)$$

Dựa trên công thức (3.65) và (3.68) critic NN xấp xỉ cho hàm chi phí tối ưu và actor NN xấp xỉ cho chính sách tối ưu được đưa ra như sau:

$$\hat{\mathbf{V}}(\mathbf{z}, \hat{\mathbf{W}}_c) = \hat{\mathbf{W}}_c^T \phi(\mathbf{z}) \quad (3.69)$$

$$\hat{\mathbf{u}}(\mathbf{z}, \hat{\mathbf{W}}_a) = -\frac{1}{2}\mathbf{R}^{-1}\mathbf{G}^T(\mathbf{z})\nabla\phi^T(\mathbf{z})\hat{\mathbf{W}}_a \quad (3.70)$$

Trong đó, $\hat{\mathbf{W}}_a, \hat{\mathbf{W}}_c$ là ước lượng các trọng số lý tưởng $\mathbf{W}$. Từ đó bộ điều khiển bám quỹ đạo cho FMWR có thể thu được như sau:

$$\boldsymbol{\tau} = -\frac{1}{2}\mathbf{R}^{-1}\mathbf{G}^T(\mathbf{z})\nabla\phi^T(\mathbf{z})\hat{\mathbf{W}}_a + g^+(\mathbf{x}_r)(h_r(\mathbf{x}_r) - f(\mathbf{x}_r)) \quad (3.71)$$

Lấy các giá trị xấp xỉ $\hat{\mathbf{u}}$ và $\hat{\mathbf{V}}$ thay vào (3.60) có thể suy ra được giá trị gần đúng của hàm Hamiltonian như sau:

$$\begin{aligned} \hat{\mathbf{H}}(\mathbf{z}, \hat{\mathbf{W}}_c, \hat{\mathbf{W}}_a) &= \lambda_M^2(\mathbf{z}) + \mathbf{z}^T \bar{\mathbf{Q}} \mathbf{z} + \frac{1}{4} \hat{\mathbf{W}}_a^T \mathbf{D}_1 \hat{\mathbf{W}}_a - \gamma \hat{\mathbf{W}}_c^T \phi \\ &\quad + \hat{\mathbf{W}}_c^T \nabla \phi \left(\mathbf{F} - \frac{1}{2} \mathbf{G} \mathbf{R}^{-1} \mathbf{G}^T \nabla \phi^T \hat{\mathbf{W}}_a \right) \end{aligned} \quad (3.72)$$

Xác định hàm Bellman là $\delta = \hat{\mathbf{H}} - \mathbf{H}^*$ ta được:

$$\delta(\mathbf{z}, \hat{\mathbf{W}}_c, \hat{\mathbf{W}}_a) = \lambda_M^2(\mathbf{z}) + \mathbf{z}^T \bar{\mathbf{Q}} \mathbf{z} + \frac{1}{4} \hat{\mathbf{W}}_a^T \mathbf{D}_1 \hat{\mathbf{W}}_a + \hat{\mathbf{W}}_c^T (\nabla \phi(\mathbf{F} + \mathbf{G}\hat{\mathbf{u}}) - \gamma \phi) \quad (3.73)$$

Để giải quyết bài toán tối ưu điều khiển ta sử dụng phương pháp bình phương tối thiểu được thiết kế để đào tạo mạng nơ-ron critic sao cho giảm thiểu sai lệch tích phân $E_c = \int_0^t \delta^2(s) ds$ như sau:

$$\dot{\hat{\mathbf{W}}}_c = -\eta_c \Gamma(t) \frac{\sigma(t)}{m_\sigma^2(t)} \delta \quad (3.74)$$

$$\dot{\Gamma}(t) = -\eta_c \left(-\beta \Gamma(t) + \Gamma(t) \frac{\sigma(t) \sigma^T(t)}{m_\sigma^2(t)} \Gamma(t) \right) \quad (3.75)$$

Trong đó, $\sigma = \nabla \phi(\mathbf{F} + \mathbf{G}\mathbf{u}) - \gamma \phi$, $m_\sigma \triangleq (1 + v\sigma^T \Gamma \sigma)^{1/2}$, η_c, v là các hằng số dương, $\beta \in (0,1)$, $\Gamma(t)$ là ma trận khuếch đại ước lượng.

Xác định sai lệch ước lượng của trọng số critic là $\tilde{\mathbf{W}}_c = \mathbf{W} - \hat{\mathbf{W}}_c$ và sai lệch ước lượng của trọng số actor là $\tilde{\mathbf{W}}_a = \mathbf{W} - \hat{\mathbf{W}}_a$ từ đó sai lệch Bellman có thể được xác định:

$$\delta = -\tilde{\mathbf{W}}_c^T \sigma + \frac{1}{4} \tilde{\mathbf{W}}_a^T \mathbf{D}_1 \tilde{\mathbf{W}}_a - \epsilon_H \quad (3.76)$$

Đặt $\bar{\sigma} = (\sigma / m_\sigma)$ áp dụng (3.75) và (3.76) sai lệch động của trọng số critic có thể được tính như sau:

$$\dot{\tilde{\mathbf{W}}}_c = \dot{\mathbf{W}} - \dot{\hat{\mathbf{W}}}_c = -\eta_c \Gamma \bar{\sigma} \bar{\sigma}^T \tilde{\mathbf{W}}_c + \eta_c \Gamma \frac{\bar{\sigma}}{m_\sigma} \left(\frac{1}{4} \tilde{\mathbf{W}}_a^T \mathbf{D}_1 \tilde{\mathbf{W}}_a - \epsilon_H \right) \quad (3.77)$$

Giả thiết điều kiện PE được áp dụng cho $\bar{\sigma}(t)$ để đảm bảo rằng $\hat{\mathbf{W}}_c$ sẽ tiến tới $\mathbf{W}$

Tồn tại $T > 0, \beta_1 > 0, \beta_2 > 0$ sao cho với mọi t

$$\beta_1 \mathbf{I} \leq \int_t^{t+T} \bar{\sigma}(s) \bar{\sigma}^T(s) ds \leq \beta_2 \mathbf{I} \quad (3.78)$$

Giả thiết tồn tại $\alpha_2 > \alpha_1 > 0$ sao cho

$$\alpha_1 \mathbf{I} \leq \Gamma(t) \leq \alpha_2 \mathbf{I} \quad \forall t \in [0, \infty] \quad (3.79)$$

Từ đó kết luận rằng:

$$\|\bar{\sigma}(t)\| \leq \frac{1}{\sqrt{v\alpha_1}} \quad \forall t \in [0, \infty] \quad (3.80)$$

Luật điều chỉnh cho trọng số nơ-ron của actor được phát triển như sau:

$$\dot{\hat{\mathbf{W}}}_a = -\eta_{a1} (\hat{\mathbf{W}}_a - \hat{\mathbf{W}}_c) - \eta_{a2} \hat{\mathbf{W}}_a + \frac{\eta_a}{4} \mathbf{D}_1 \hat{\mathbf{W}}_a \frac{\bar{\sigma}^T}{m_\sigma} \hat{\mathbf{W}}_c \quad (3.81)$$

Trong đó, $\eta_{a1} > 0$, $\eta_{a2} > 0$ và $\eta_a > 0$ là các tham số điều chỉnh. Vì sai lệch Bellman là phi tuyến đối với các ước tính trọng số, phương pháp bình phương nhỏ nhất không thể được sử dụng để cập nhật trọng số [105].

Xác định sai lệch

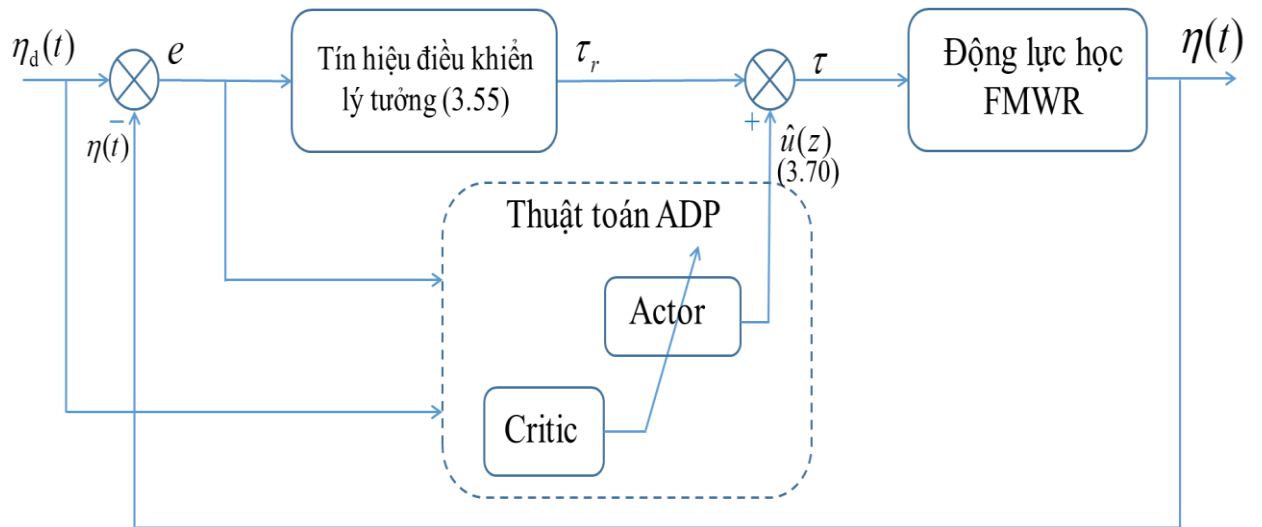
$$\tilde{\mathbf{u}} = \hat{\mathbf{u}}(\mathbf{z}, \hat{\mathbf{W}}_a) - \tilde{\mathbf{u}}_c(\mathbf{z}, \hat{\mathbf{W}}_c) = -\frac{1}{2} \mathbf{R}^{-1} \mathbf{G}^T(\mathbf{z}) \nabla \phi^T(\mathbf{z}) (\hat{\mathbf{W}}_a - \hat{\mathbf{W}}_c) \quad (3.82)$$

và hàm mục tiêu $\mathbf{E}_u = \frac{1}{2} \tilde{\mathbf{u}}^T \mathbf{R} \tilde{\mathbf{u}}$ để giảm thiểu hàm mục tiêu ta có thể sử dụng thuật toán Gradient- descent để điều chỉnh trọng số nơ-ron của actor:

$$\hat{\mathbf{W}}_a = -\eta_{a1} (\mathbf{E}_u / \hat{\mathbf{W}}_a) = -\eta_{a1} \mathbf{D}_1 (\hat{\mathbf{W}}_a - \hat{\mathbf{W}}_c) \quad (3.83)$$

Vì vậy (3.81) có thể xem là một luật sửa đổi điều chỉnh cho trọng số của actor và số hạng thứ hai của (3.81) là để cải thiện độ bền vững cho hệ thống, còn số hạng thứ ba của (3.81) được thiết kế để đảm bảo sự ổn định của vòng kín hệ thống theo hàm Lyapunov

Sơ đồ cấu trúc điều khiển được trình bày trong hình 3.3, trong đó tín hiệu điều khiển gồm hai thành phần: Tín hiệu điều khiển lý tưởng được xác định trong công thức (3.55) và tín hiệu điều khiển tối ưu $\hat{\mathbf{u}}(\mathbf{z}, \hat{\mathbf{W}}_a)$ được xác định như công thức (3.70) đưa vào động lực học của FMWR.



Hình 3.3 Sơ đồ điều khiển FMWR sử dụng cấu trúc actor-critic

3.3.4. Xét tính ổn định cho hệ thống

Xét hệ thống (3.56), trọng số NN của critic và tín hiệu điều khiển được cho bởi (3.69) và (3.70). Luật điều chỉnh trọng số cho critic và actor NNs được cho bởi (3.75) và (3.81). Trạng thái của hệ thống vòng kín, sai lệch trọng số NN critic $\tilde{\mathbf{W}}_c$ và sai lệch trọng số NN actor $\tilde{\mathbf{W}}_a$ đều bị chặn theo điều kiện UB [106]. Hơn thế sai số ước lượng trọng số $\tilde{\mathbf{W}}_c$ hội tụ theo cấp số nhân

Xét hàm Lyapunov như sau:

$$\begin{aligned}\mathbf{V}(t) &= \mathbf{V}^*(t) + \mathbf{V}_1(t) + \mathbf{V}_2(t) \\ &= \mathbf{V}^*(t) + \frac{1}{2\eta_c} \tilde{\mathbf{W}}_c^T(t) \Gamma^{-1}(t) \tilde{\mathbf{W}}_c(t) + \frac{1}{2\eta_a} \tilde{\mathbf{W}}_a^T(t) \tilde{\mathbf{W}}_a(t)\end{aligned}\quad (3.84)$$

Trong đó, $\mathbf{V}^*(t)$ là hàm chi phí tối ưu và có đạo hàm theo thời gian như sau:

$$\begin{aligned}\dot{\mathbf{V}}^*(t) &= \nabla \mathbf{V}^{*T}(\mathbf{F} + \mathbf{G}\hat{\mathbf{u}}) = (\nabla \phi^T \mathbf{W} + \nabla \epsilon_v)^T \left(\mathbf{F} - \frac{1}{2} \mathbf{G} \mathbf{R}^{-1} \mathbf{G}^T \nabla \phi^T \hat{\mathbf{W}}_a \right) \\ &= \mathbf{W}^T \nabla - \frac{1}{2} \mathbf{W}^T \mathbf{D}_1 \hat{\mathbf{W}}_a + \nabla \epsilon_v^T \left(\mathbf{F} - \frac{1}{2} \mathbf{G} \mathbf{R}^{-1} \mathbf{G}^T \nabla \phi^T \hat{\mathbf{W}}_a \right)\end{aligned}\quad (3.85)$$

Từ phương trình HJB (3.67), ta có:

$$\dot{\mathbf{V}}^*(t) = -\lambda_M^2(\mathbf{z}) - \mathbf{z}^T \bar{\mathbf{Q}} \mathbf{z} + \gamma \mathbf{W}^T \phi + \frac{1}{4} \mathbf{W}^T \mathbf{D}_1 \mathbf{W} - \frac{1}{2} \mathbf{W}^T \mathbf{D}_1 \hat{\mathbf{W}}_a - \epsilon_H + \epsilon_1 \quad (3.86)$$

Trong đó $\epsilon_1 = \dot{\epsilon}_v = \nabla \epsilon_v^T \left(\mathbf{F} - (1/2) \mathbf{G} \mathbf{R}^{-1} \mathbf{G}^T \nabla \phi^T \hat{\mathbf{W}}_a \right)$,

$$\epsilon_1 - \epsilon_H = \frac{1}{2} \tilde{\mathbf{W}}_a^T \nabla \phi \mathbf{D}_2 \nabla \epsilon_v + \frac{1}{4} \nabla \epsilon_v^T \mathbf{D}_2 \nabla \epsilon_v + \gamma \epsilon_v(\mathbf{z})$$

Từ đó phương trình (3.86) được viết lại như sau:

$$\begin{aligned}\dot{\mathbf{V}}^*(t) &= -\lambda_M^2(\mathbf{z}) - \mathbf{z}^T \bar{\mathbf{Q}} \mathbf{z} + \gamma \mathbf{W}^T \phi + \frac{1}{4} \mathbf{W}^T \mathbf{D}_1 \mathbf{W} + \frac{1}{2} \tilde{\mathbf{W}}_a^T \mathbf{D}_1 \mathbf{W} \\ &\quad + \frac{1}{2} \tilde{\mathbf{W}}_a^T \nabla \phi \mathbf{D}_2 \nabla \epsilon_v + \frac{1}{4} \nabla \epsilon_v^T \mathbf{D}_2 \nabla \epsilon_v + \gamma \epsilon_v(\mathbf{z})\end{aligned}\quad (3.87)$$

Thay (3.85) vào (3.87) và đạo hàm theo thời gian của $\mathbf{V}_1^*(t)$ ta có:

$$\dot{\mathbf{V}}_1(t) = -\frac{1}{2} \tilde{\mathbf{W}}_c^T \bar{\sigma} \bar{\sigma}^T \tilde{\mathbf{W}}_c - \frac{\beta}{2} \tilde{\mathbf{W}}_c^T \Gamma^{-1} \tilde{\mathbf{W}}_c + \tilde{\mathbf{W}}_c^T \frac{\bar{\sigma}}{m} \left(\frac{1}{4} \tilde{\mathbf{W}}_a^T \mathbf{D}_1 \tilde{\mathbf{W}}_a - \boldsymbol{\varepsilon}_H \right) \quad (3.88)$$

Với $\mathbf{V}_2^*(t)$, sử dụng (3.81) và $\hat{\mathbf{W}}_c = \mathbf{W} - \tilde{\mathbf{W}}_c$; $\hat{\mathbf{W}}_a = \mathbf{W} - \tilde{\mathbf{W}}_a$, ta có:

$$\begin{aligned} \dot{\mathbf{V}}_2(t) &= -\frac{1}{\eta_a} \tilde{\mathbf{W}}_a^T \dot{\hat{\mathbf{W}}}_a = \\ &= -\frac{(\eta_{a1} + \eta_{a2})}{\eta_a} \tilde{\mathbf{W}}_a^T \tilde{\mathbf{W}}_a + \frac{\eta_{a1}}{\eta_a} \tilde{\mathbf{W}}_a^T \tilde{\mathbf{W}}_c + \frac{\eta_{a2}}{\eta_a} \tilde{\mathbf{W}}_a^T \mathbf{W} - \frac{1}{4} \tilde{\mathbf{W}}_a^T \mathbf{D}_1 \hat{\mathbf{W}}_a \frac{\bar{\sigma}^T}{m_\sigma} \hat{\mathbf{W}} \end{aligned} \quad (3.89)$$

Thay vào đạo hàm theo thời gian của $V(t)$ ta được:

$$\begin{aligned} \dot{\mathbf{V}}(t) &= -\lambda_M^2(\mathbf{z}) - \mathbf{e}^T \mathbf{Q}_e - \frac{1}{4} \mathbf{W}^T \mathbf{D}_1 \mathbf{W} - \frac{\tilde{\mathbf{W}}_c^T \bar{\sigma}}{m_\sigma} \boldsymbol{\varepsilon}_H - \tilde{\mathbf{W}}_c^T \mathbf{A}_1 \tilde{\mathbf{W}}_c \\ &\quad - \tilde{\mathbf{W}}_a^T \mathbf{A}_2 \tilde{\mathbf{W}}_a + \tilde{\mathbf{W}}_a^T \mathbf{A}_3 \tilde{\mathbf{W}}_c + \tilde{\mathbf{W}}_a^T \mathbf{B}_1 + \mathbf{B}_2 \end{aligned} \quad (3.90)$$

Trong đó:

$$\mathbf{A}_1 = \frac{1}{2} \bar{\sigma} \bar{\sigma}^T + \frac{\beta}{2} \Gamma^{-1}, \quad \mathbf{A}_2 = \frac{(\eta_{a1} + \eta_{a2})}{\eta_a} \mathbf{I} - \frac{1}{4} \mathbf{D}_1 \frac{\bar{\sigma}^T}{m_\sigma} \mathbf{W}, \quad \mathbf{A}_3 = \frac{1}{4} \mathbf{D}_1 \mathbf{W} \frac{\bar{\sigma}^T}{m_\sigma} + \frac{\eta_{a1}}{\eta_a} \mathbf{I}$$

$$\mathbf{B}_1 = \frac{1}{2} \mathbf{D}_1 \mathbf{W} + \frac{1}{2} \nabla \phi \mathbf{D}_2 \boldsymbol{\varepsilon}_v - \frac{1}{4} \mathbf{D}_1 \mathbf{W} \frac{\bar{\sigma}^T}{m_\sigma} \mathbf{W} + \frac{\eta_{a2}}{\eta_a} \mathbf{W},$$

$$\mathbf{B}_2 = \gamma \mathbf{W}^T \phi + \frac{1}{4} \nabla \boldsymbol{\varepsilon}_v^T \mathbf{D}_2 \nabla \boldsymbol{\varepsilon}_v + \gamma \boldsymbol{\varepsilon}_v(\mathbf{z})$$

Tồn tại các hằng số dương k_1, k_2, k_3 sao cho $\|\mathbf{B}_1\| \leq k_1$, $\|\mathbf{B}_2\| \leq k_2$,

$\|(1/4)\mathbf{D}_1(\bar{\sigma}^T / m_\sigma)\mathbf{W}\| \leq k_3$. Áp dụng bất đẳng thức Young thì (3.90) trở thành:

$$\begin{aligned} \dot{\mathbf{V}}(t) &\leq -\underline{q} \|\mathbf{e}\|^2 - \frac{\beta}{4\alpha_2} \|\tilde{\mathbf{W}}_c\|^2 - \frac{\eta_{a1} + \eta_{a2}}{2\eta_a} \|\tilde{\mathbf{W}}_a\|^2 - \left(\frac{\beta}{4\alpha_2} - \frac{1}{2\zeta} \left(\frac{\eta_{a1}}{\eta_a} + k_3 \right) \right) \|\tilde{\mathbf{W}}_c\|^2 \\ &\quad - \left(\frac{\eta_{a1} + \eta_{a2}}{2\eta_a} - k_3 - \frac{\zeta}{2} \left(\frac{\eta_{a1}}{\eta_a} + k_3 \right) \right) \|\tilde{\mathbf{W}}_a\|^2 + \frac{\varepsilon_h}{\sqrt{\nu\alpha_1}} \|\tilde{\mathbf{W}}_c\| + k_1 \|\tilde{\mathbf{W}}_a\| + k_2 \end{aligned} \quad (3.91)$$

Trong đó, $\underline{q} = \lambda_{\min}(Q)$. Chọn các thông số sao cho $(\beta / 4\alpha_2) \geq (1 / 2\varsigma)((\eta_{a1} / \eta_a) + k_3)$ và $(\eta_{a1} + (\eta_{a2} / 2\eta_a)) \geq k_3 + (\varsigma / 2)((\eta_{a1} / \eta_a) + k_3)$, đặt $\xi = (e^T, \tilde{W}_c^T, \tilde{W}_a^T)^T$, $\bar{\omega}_1 = \min(\underline{q}, (\beta / 4\alpha_2))$ và $\bar{\omega}_2 = \max((\varepsilon_h / \sqrt{v\alpha_1}), k_1)$ thì (3.91) trở thành

$$\dot{V}(t) \leq -\bar{\omega}_1 \|\xi\|^2 + \bar{\omega}_2 \|\xi\| + k_2 \quad (3.92)$$

Do đó, nếu bất đẳng thức $\|\xi\| > (\bar{\omega}_2 / 2\bar{\omega}_1 + ((\bar{\omega}_2^2 / 4\bar{\omega}_1^2) + (k_2 / \bar{\omega}_1)))^{1/2}$ giữ nguyên thì ta có thể suy ra rằng đạo hàm của hàm Lyapunov $\dot{V}(t)$ âm. Sử dụng lý thuyết ổn định Lyapunov ta có thể thấy rằng trạng thái của hệ thống vòng kín và sai lệch ước lượng trọng số $\tilde{W}_a, \tilde{W}_c$ bị chặn

$$\|\mathbf{u}^* - \hat{\mathbf{u}}\| \leq \frac{1}{2\lambda_{\min}(\mathbf{R})} b_g (b_{\phi z} b_{\varsigma} + b_{\varepsilon z}) = b_u \quad (3.93)$$

Trong đó b_u là một hằng số dương.

3.3.5. Mô phỏng thuật toán

Trong phần này thuật toán học tăng cường áp dụng điều khiển bám quỹ đạo cho robot di động bốn bánh xe mecanum được mô phỏng với các kịch bản khác nhau trên Matlab để đánh giá hiệu quả của thuật toán.

Tham số của mô hình FMWR được lựa chọn như (2.27):

Để đảm bảo hệ thống điều khiển hoạt động hiệu quả và ổn định trong quá trình bám quỹ đạo, các tham số ban đầu của thuật toán được lựa chọn bao gồm: các ma trận trọng số, hệ số điều chỉnh như sau:

$$\begin{aligned} \hat{\mathbf{W}}_c(0) &= 600 \times \mathbf{1}_{51}, \hat{\mathbf{W}}_a(0) = 6 \times \mathbf{1}_{51}, \mathbf{\Gamma}(0) = 10 \times \mathbf{I}_{51}, \mathbf{Q} = \mathbf{I}_6, \mathbf{R} = \mathbf{I}_4, \\ \eta_c &= 3, \eta_{a1} = 3, \eta_{a2} = 0.001, \eta_a = 0.001, \beta = 0.001, \gamma = 0.0001, v = 0.002 \end{aligned}$$

Hàm kích hoạt được chọn như sau:

$$\begin{aligned} \phi(z) &= \frac{1}{2} [z_1^2 \quad z_2^2 \quad z_3^2 \quad z_1 z_4 \quad z_1 z_5 \quad z_1 z_6 \quad z_2 z_4 \quad z_2 z_5 \quad z_2 z_6 \quad z_3 z_4 \quad z_3 z_5 \quad z_3 z_6 \quad z_1^2 z_2^2 \quad z_1^2 z_3^2 \quad z_2^2 z_3^2 \\ &\quad z_1^2 z_7^2 \quad z_1^2 z_8^2 \quad z_1^2 z_9^2 \quad z_1^2 z_{10}^2 \quad z_1^2 z_{11}^2 \quad z_1^2 z_{12}^2 \quad z_2^2 z_7^2 \quad z_2^2 z_8^2 \quad z_2^2 z_9^2 \quad z_2^2 z_{10}^2 \quad z_2^2 z_{11}^2 \quad z_2^2 z_{12}^2 \quad z_3^2 z_7^2 \quad z_3^2 z_8^2 \\ &\quad z_3^2 z_9^2 \quad z_3^2 z_{10}^2 \quad z_3^2 z_{11}^2 \quad z_3^2 z_{12}^2 \quad z_4^2 z_7^2 \quad z_4^2 z_8^2 \quad z_4^2 z_9^2 \quad z_4^2 z_{10}^2 \quad z_4^2 z_{11}^2 \quad z_4^2 z_{12}^2 \quad z_5^2 z_7^2 \quad z_5^2 z_8^2 \quad z_5^2 z_9^2 \quad z_5^2 z_{10}^2 \quad z_5^2 z_{11}^2 \\ &\quad z_5^2 z_{12}^2 \quad z_6^2 z_7^2 \quad z_6^2 z_8^2 \quad z_6^2 z_9^2 \quad z_6^2 z_{10}^2 \quad z_6^2 z_{11}^2 \quad z_6^2 z_{12}^2]^T \end{aligned}$$

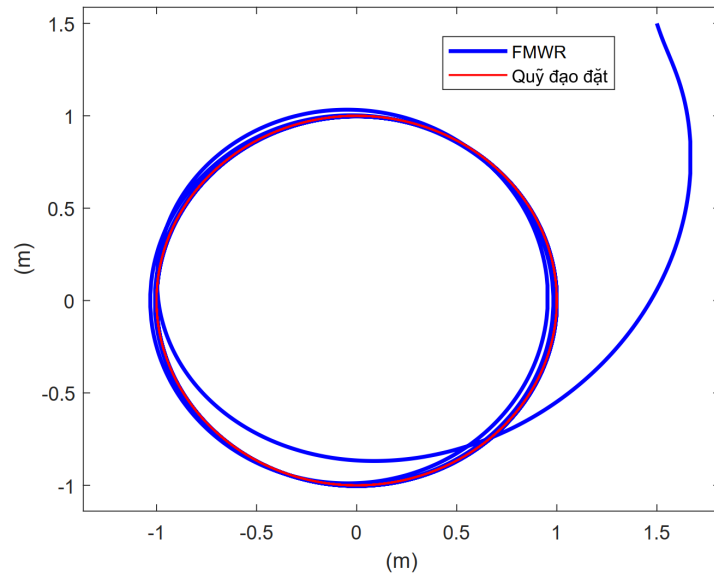
Quỹ đạo đặt đường tròn được cho bởi phương trình:

$$\mathbf{x}_r(t) = [\sin(t) \quad \cos(t) \quad \cos(0.5t)]^T$$

Vị trí và vận tốc ban đầu được chọn: $\mathbf{x}(0) = [1.5 \quad 1.5 \quad 1.3 \quad 0 \quad 0 \quad 0]^T$

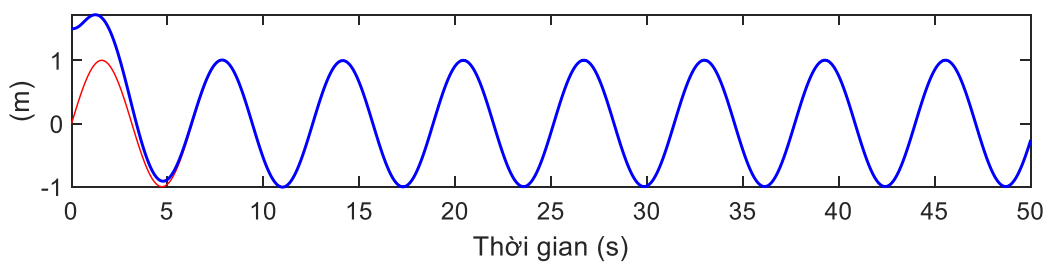
Thực hiện mô phỏng trong thời gian $T=50s$, với các kịch bản khác nhau:

➤ **Kịch bản mô phỏng 1:** Mô hình robot có khối lượng và trọng tâm không đổi

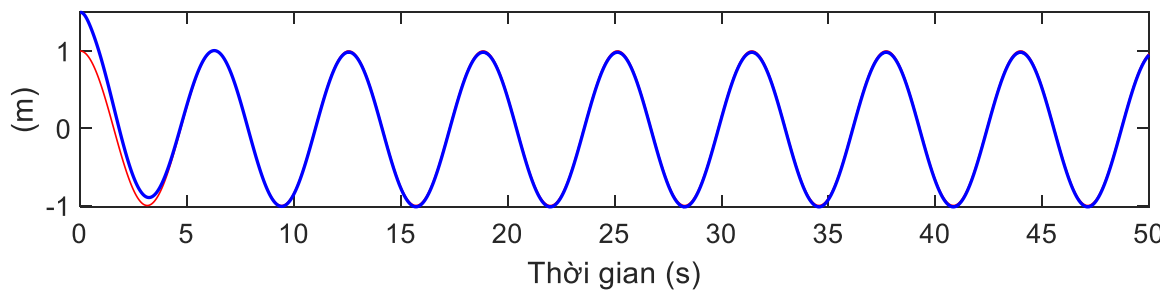


Hình 3.4 Kết quả mô phỏng của FMWR với quỹ đạo đường tròn

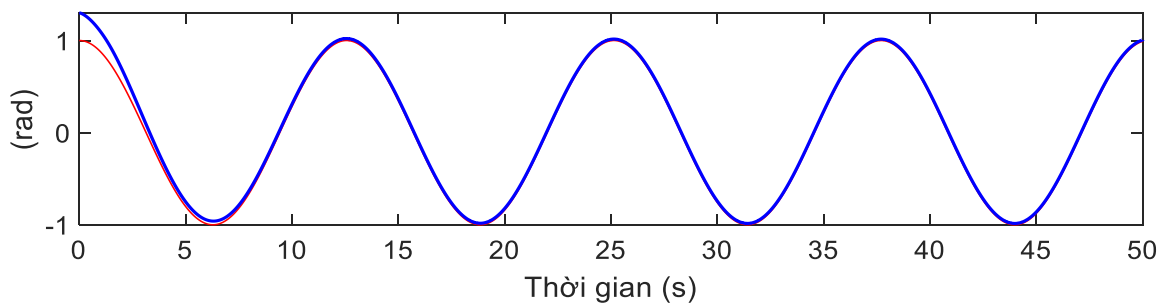
Trong 5 giây đầu tiên, robot di chuyển tiến gần đến quỹ đạo đặt và tự động học để điều chỉnh hành động của mình, giúp bám theo quỹ đạo một cách chính xác hơn. Trong suốt quá trình này, thuật toán học tăng cường cho phép robot điều chỉnh các trọng số điều khiển sao cho phù hợp, dần dần cải thiện khả năng bám sát quỹ đạo mong muốn. Sau khoảng thời gian ngắn, robot đã ổn định và tiếp tục duy trì quỹ đạo đặt.



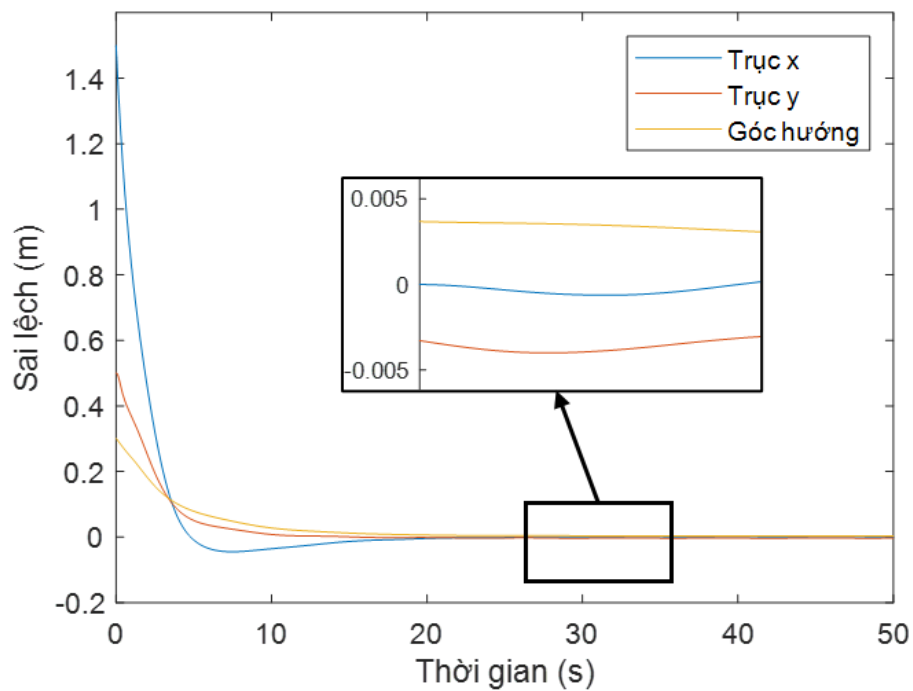
Hình 3.5 Quỹ đạo di chuyển của FMWR theo trục x



Hình 3.6 Quỹ đạo di chuyển của FMWR theo trục y

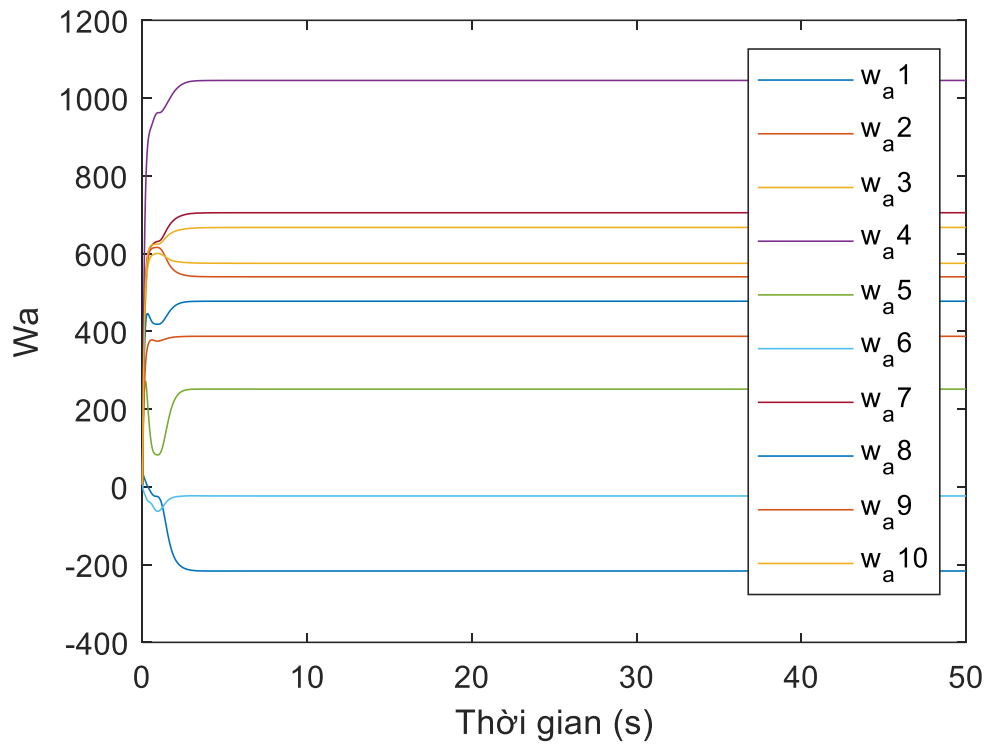


Hình 3.7 Quỹ đạo di chuyển của FMWR theo góc hướng

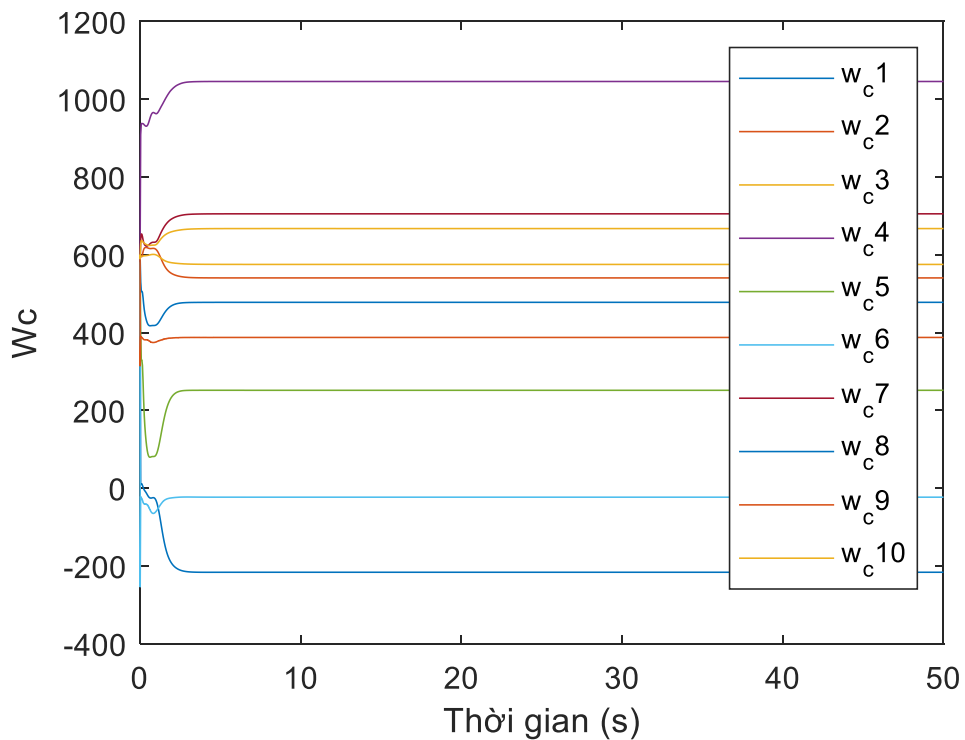


Hình 3.8 Sai lệch quỹ đạo của FMWR theo các trục x, y và góc hướng

Sai lệch quỹ đạo thực tế so với quỹ đạo đặt thể hiện trong hình 3.8 cho thấy robot di chuyển rất gần với quỹ đạo đặt, dù có một vài sai lệch nhỏ cụ thể là $[0.00062, -0.00374, 0.00318]$ m theo các trục x, y và góc quay.



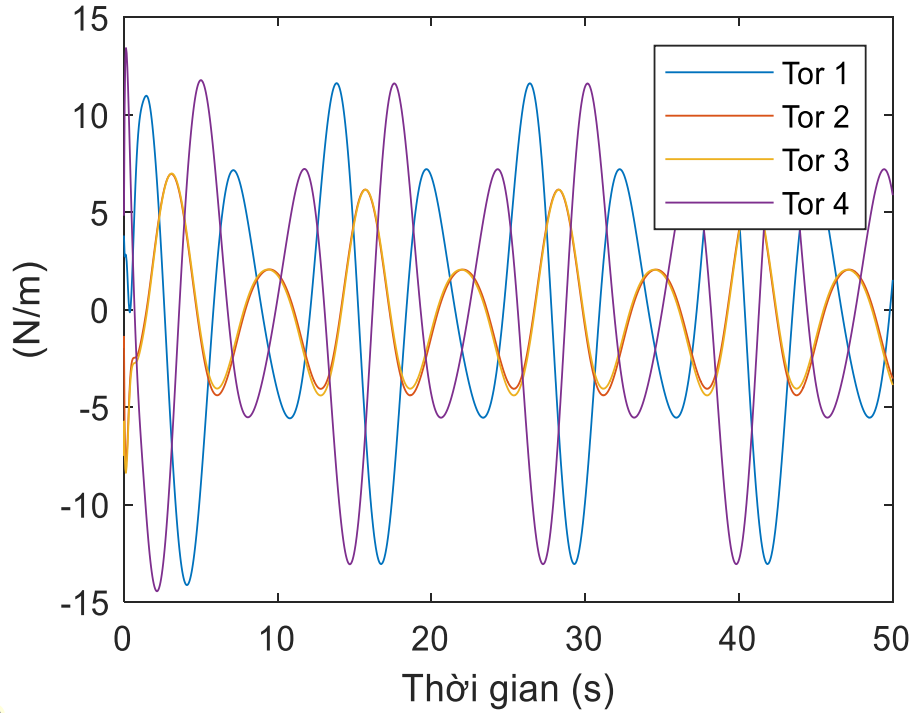
Hình 3.9 Trọng số của mạng nơ-ron Actor



Hình 3.10 Trọng số của mạng nơ-ron Critic

Sau khoảng thời gian 5 giây, các trọng số mạng nơ-ron của Actor và Critic trong thuật toán đã hoàn thành quá trình tự học và điều chỉnh, hội tụ về các giá trị tối ưu với các giá trị trọng số $W_a = [-226.6 \ 560.3 \ 578.7 \ 973.9 \ 328.5 \ 81.9 \ 666 \ 501.1$

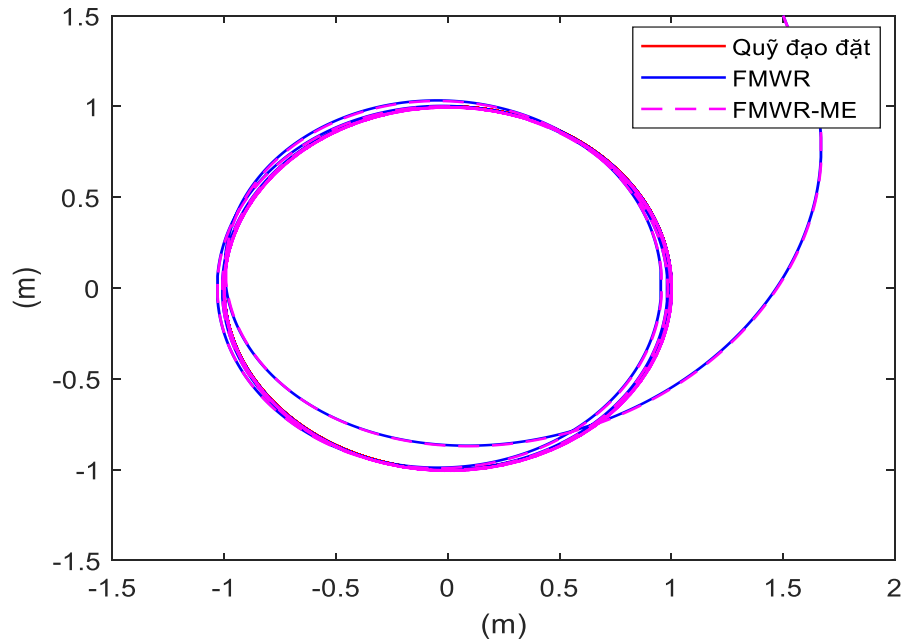
422.9 647.3] và $W_c = [-226.5 \ 560.1 \ 578.5 \ 973.6 \ 328.4 \ 81.8 \ 665.8 \ 501 \ 422.8 \ 647]$ thể hiện trong hình 3.9 và hình 3.10. Quá trình hội tụ này chứng tỏ rằng mạng nơ-ron Actor đã học được cách lựa chọn hành động tối ưu dựa trên trạng thái hiện tại của hệ thống, trong khi mạng nơ-ron Critic đánh giá các giá trị của các hành động này để cải thiện quá trình học. Sau khi các trọng số hội tụ, robot tiếp tục duy trì quỹ đạo đặt ổn định và giảm thiểu sai lệch theo thời gian.



Hình 3.11 Tín hiệu mô men điều khiển của bánh xe

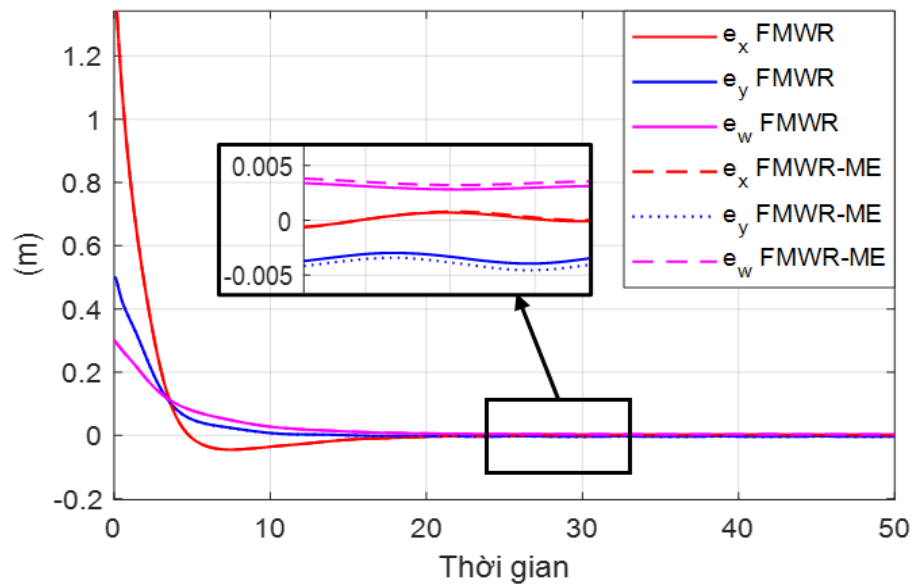
Tín hiệu điều khiển của bánh xe được thể hiện trong hình 3.11 với các giá trị mô tả lực mô-men xoắn tác động lên các bánh xe của robot. Tín hiệu điều khiển này phản ánh các giá trị lực mô-men xoắn cần thiết để robot có thể bám theo quỹ đạo một cách chính xác. Các giá trị lực mô-men xoắn Tor lớn nhất cho các bánh xe trong mô phỏng được ghi nhận là [12.06, 6.45, 6.45, 12.06] N/m, tương ứng với các bánh xe của robot.

➤ **Kịch bản mô phỏng 2:** Mô hình robot có khối lượng và trọng tâm thay đổi (FMWR-ME) như trong công thức (2.44):



Hình 3.12 So sánh quỹ đạo FMWR và FMWR-ME

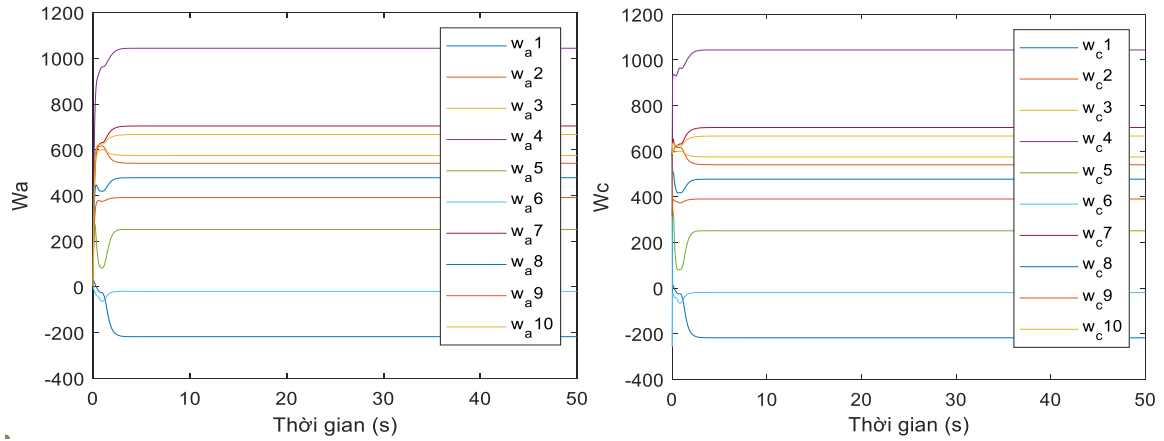
Dựa vào kết quả mô phỏng hình 3.10 nhận thấy khi trọng tâm của robot thay đổi thì bộ điều khiển học tăng cường vẫn giúp cho robot bám theo quỹ đạo đặt và tự học trong khoảng thời gian là 6s, sau đó tiếp tục di chuyển và duy trì ổn định trên quỹ đạo đặt.



Hình 3.13 Sai lệch quỹ đạo khi trọng tâm thay đổi

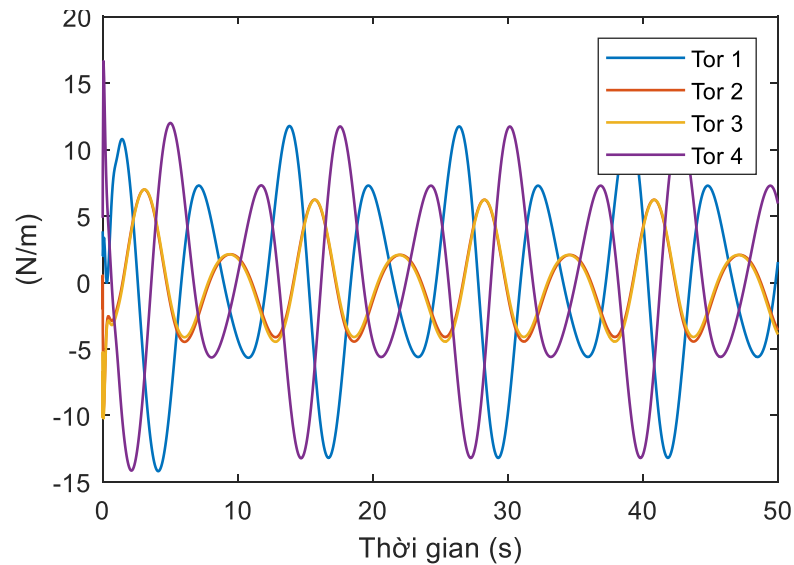
Sai lệch thể hiện trong hình 3.13 cho thấy robot có trọng tâm bị lệch thì tồn tại sai lệch theo các trục x, y và góc hướng lần lượt là $[0.0066 \ -0.0038 \ 0.00032]$ m. Việc thay đổi trọng tâm ảnh hưởng đến động học và động lực học của robot, dẫn đến việc

robot có thể khó khăn hơn trong việc duy trì quỹ đạo. Khi bị lệch trọng tâm thì một khối lượng lớn hơn ở một phía của robot có thể khiến bánh xe chuyển chậm hơn hoặc phản ứng không kịp thời với các tín hiệu điều khiển, trong khi bên còn lại khối lượng nhỏ hơn có thể khiến bánh xe dễ bị lệch khỏi quỹ đạo.



Hình 3.14 Sự hội tụ của trọng số nơ-ron Actor- Critic

Khi bắt đầu mô phỏng thì trọng số nơ-ron Actor và Critic cần khoảng thời gian 6s để tự học và hội tụ, thể hiện trong hình 3.14 với các giá trị $W_a = [-223.1 \ 558.6 \ 577.8 \ 967.1 \ 343.4 \ 81.4 \ 665.7 \ 506.7 \ 423.1 \ 646.6]$ và $W_c = [-223.2 \ 558.8 \ 578 \ 967.4 \ 343.5 \ 81.45 \ 665.98 \ 506.85 \ 423.3 \ 646.8]$.



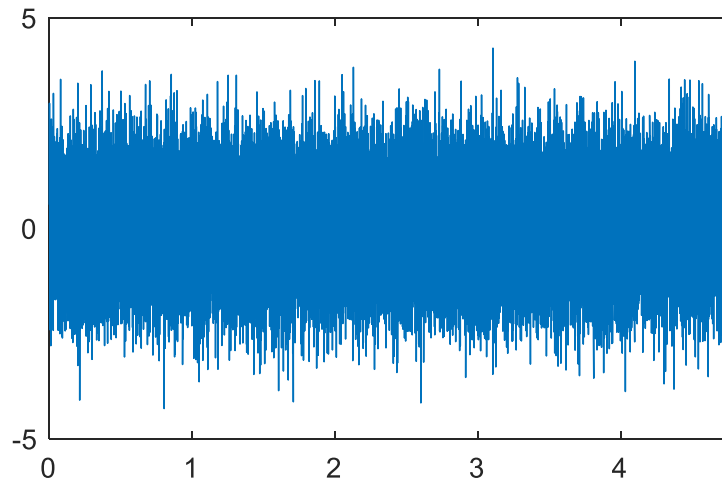
Hình 3.15 Tín hiệu điều khiển Tor của FMWR-ME

Tín hiệu điều khiển Tor trong hình 3.15 cho thấy khi bắt đầu mô phỏng, tín hiệu điều khiển đã được tính toán để giúp robot tiến đến bám quỹ đạo đặt. Sau khoảng thời

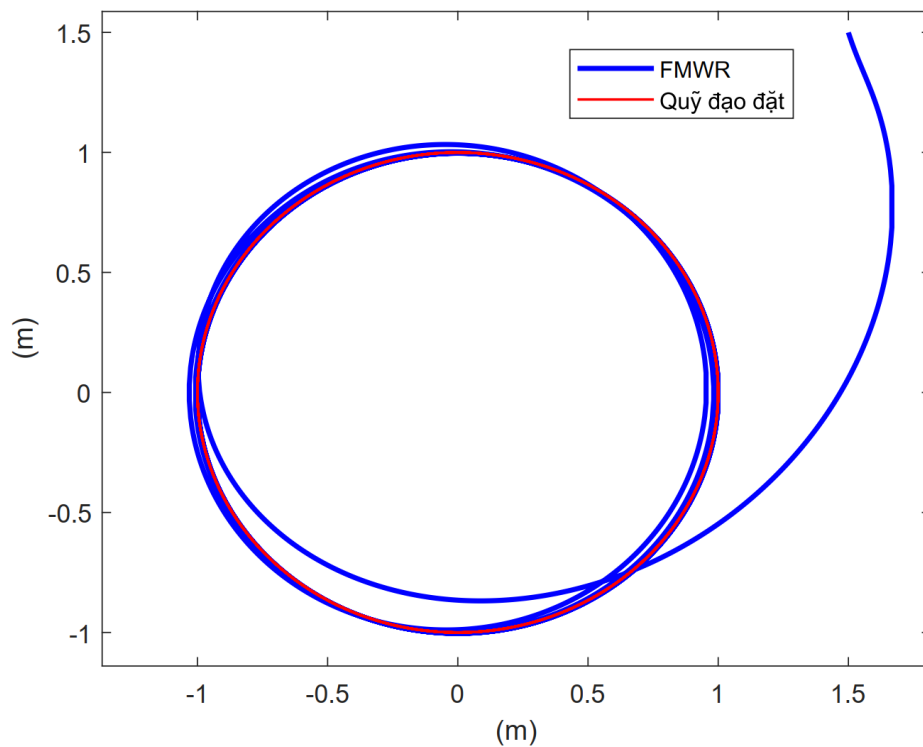
gian 10s thì robot đã ổn định di chuyển trên quỹ đạo với các tín hiệu lớn nhất là: $Tor=[12.06 \ 6.45 \ 6.45 \ 12.06]^T$ N/m đảm bảo khả năng đáp ứng thực tế của thuật toán.

➤ **Kịch bản mô phỏng 3:** thử nghiệm với nhiễu ngoại lực tác động vào mô hình FMWR. Tín hiệu nhiễu Gaussian có dạng như sau:

$$\epsilon_{Gaus} = \text{normrnd}(0, 1.5, \text{size}(t))$$

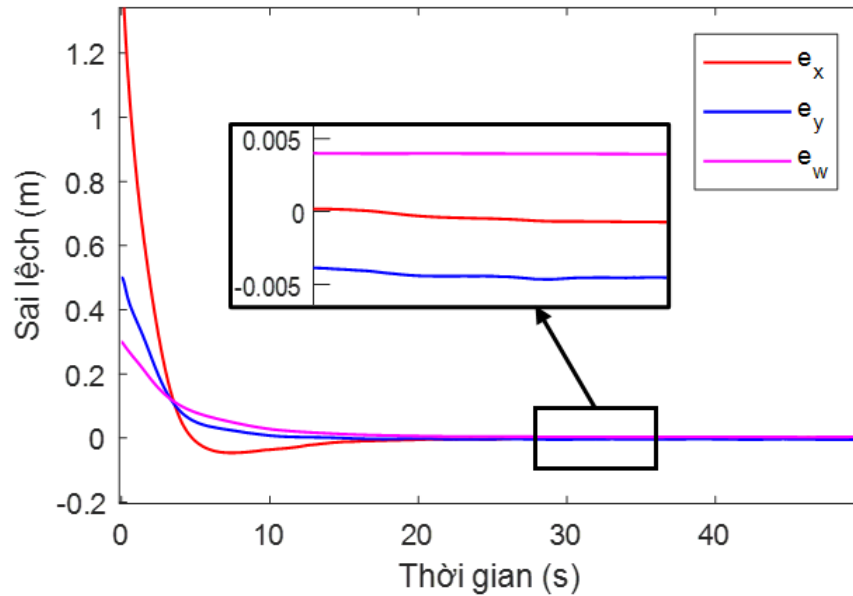


Hình 3.16 Nhiễu ngoại lực Gaussian

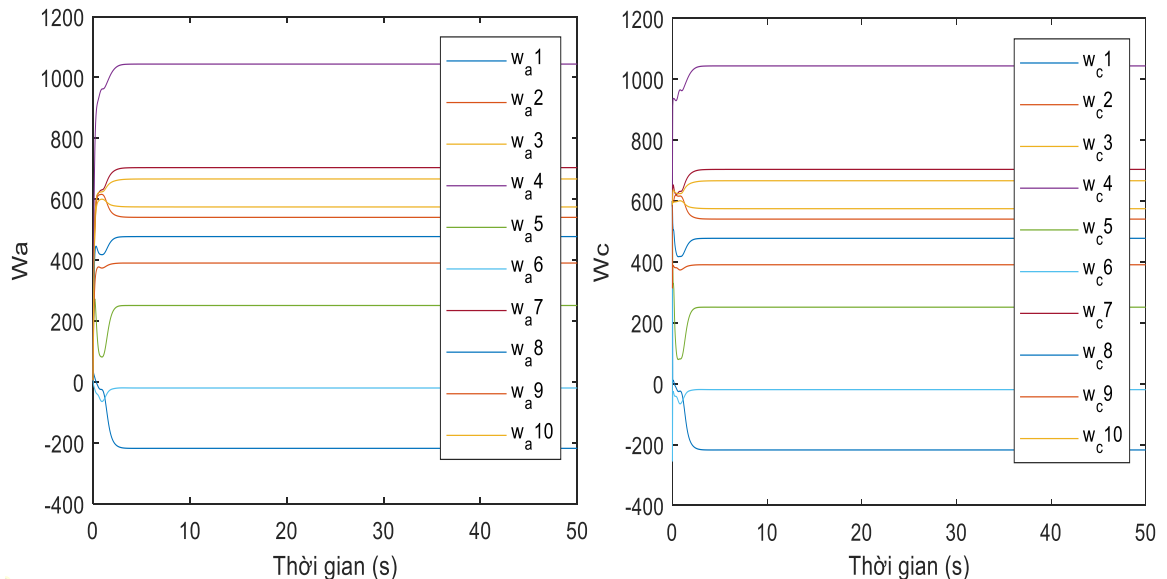


Hình 3.17 Quỹ đạo di chuyển khi có nhiễu tác động

Khi có tín hiệu nhiễu tác động vào mô hình robot thì thuật toán học tăng cường vẫn giúp robot di chuyển bám theo quỹ đạo đặt, mặc dù còn tồn tại sai lệch nhỏ thể hiện như trong hình 3.18. Điều này đạt được nhờ khả năng học hỏi và điều chỉnh hành động của robot thông qua phản hồi từ môi trường, từ đó giúp robot thích ứng và giảm thiểu ảnh hưởng của nhiễu.



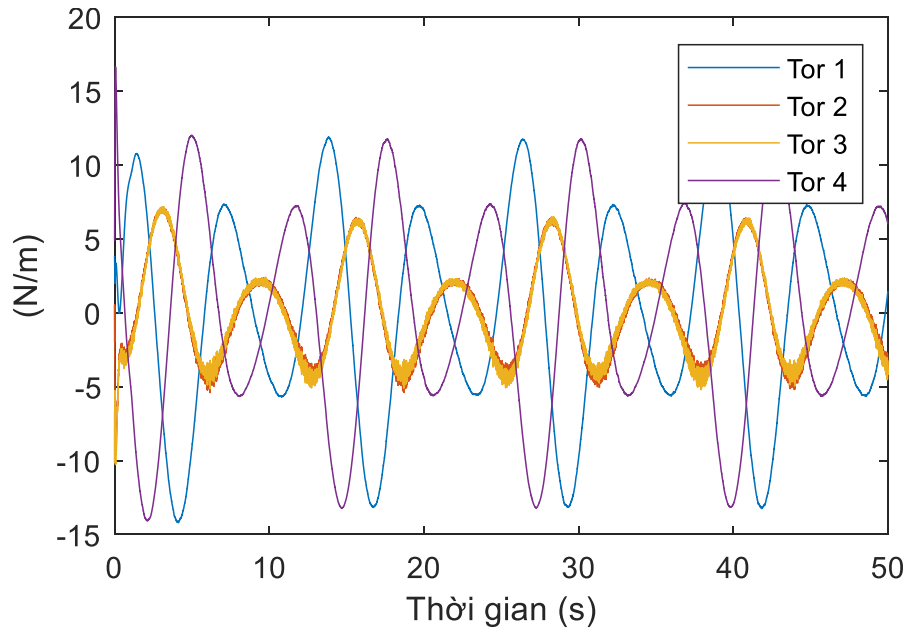
Hình 3.18 Các sai lệch bám khi có nhiễu tác động vào mô hình



Hình 3.19 Các trọng số W_a và W_c khi có nhiễu tác động

Sau khoảng thời gian 6 giây, quá trình tự học và thích nghi với tín hiệu nhiễu hoàn thành khi đó các trọng số trong mạng NNs đã hội tụ hình 3.19 và robot đã bắt đầu di chuyển ổn định để bám sát quỹ đạo đặt. Điều này cho thấy rằng quá trình huấn

luyện và điều chỉnh đã đạt được sự hội tụ tốt và robot có thể thực hiện được nhiệm vụ di chuyển theo quỹ đạo mục tiêu một cách chính xác.



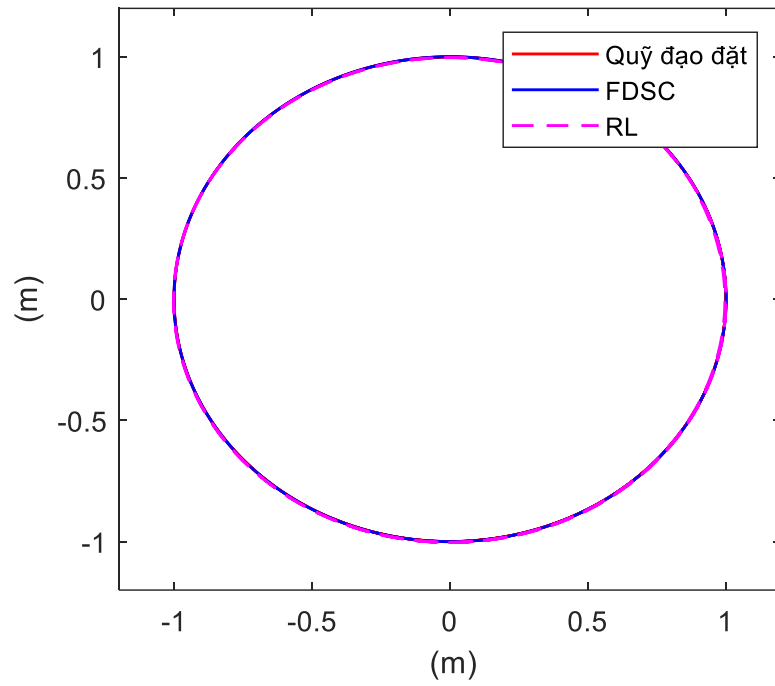
Hình 3.20 Tín hiệu điều khiển Tor khi có nhiều tác động vào mô hình

Sau khi tín hiệu nhiễu được đưa vào mô hình, robot đã bắt đầu chịu tác động từ nhiễu. Tuy nhiên, điều đáng chú ý là các tín hiệu điều khiển Tor đã điều chỉnh thích nghi với sự thay đổi này trong hình 3.20. Nhờ vậy, robot vẫn duy trì ổn định và tiếp tục bám theo quỹ đạo mục tiêu. Điều này cho thấy tính thích nghi của hệ thống với nhiễu và sự thay đổi của môi trường.

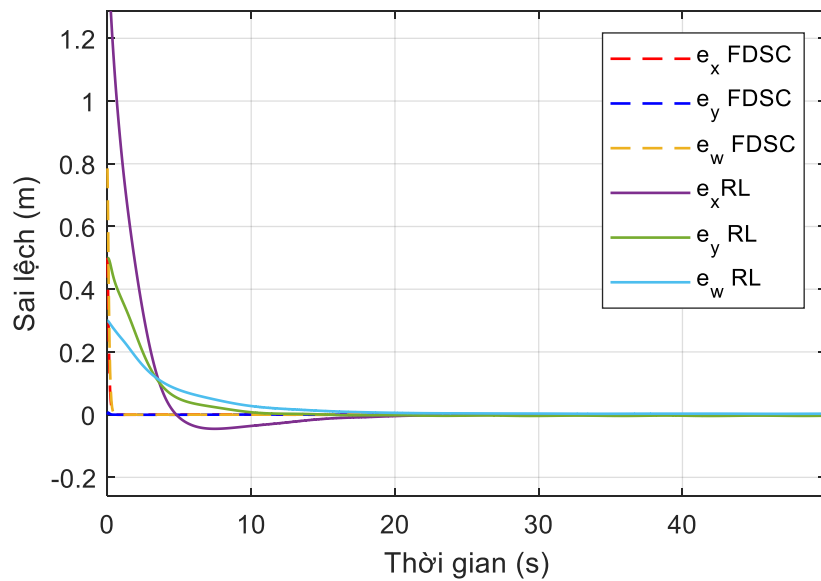
➤ **Kịch bản mô phỏng 4:** So sánh với thuật toán Fuzzy-DSC được đề xuất trong chương 2.

Thực hiện mô phỏng đồng thời thuật toán Fuzzy-DSC và thuật toán học tăng cường cấu trúc Actor-Critic với mô hình FMWR có cùng tham số và nhiễu ngoại tác động như nhau, kết quả mô phỏng hình 3.21.

Kết quả mô phỏng cho thấy thuật toán Fuzzy-DSC ngay lập tức ổn định và bám sát quỹ đạo với độ chính xác cao với thời gian ngắn chỉ khoảng 1 giây, thể hiện trong hình 3.22. Bộ chỉnh định mờ giúp thuật toán điều khiển có thể thích nghi được với các tác động của nhiễu ngoại và ảnh hưởng của độ lệch trọng tâm. Tuy nhiên, thuật toán chưa tối ưu hóa chính sách điều khiển theo thời gian, thể hiện như trong hình 3.23 cho thấy mô men điều khiển chưa được tối ưu so với thuật toán học tăng cường.



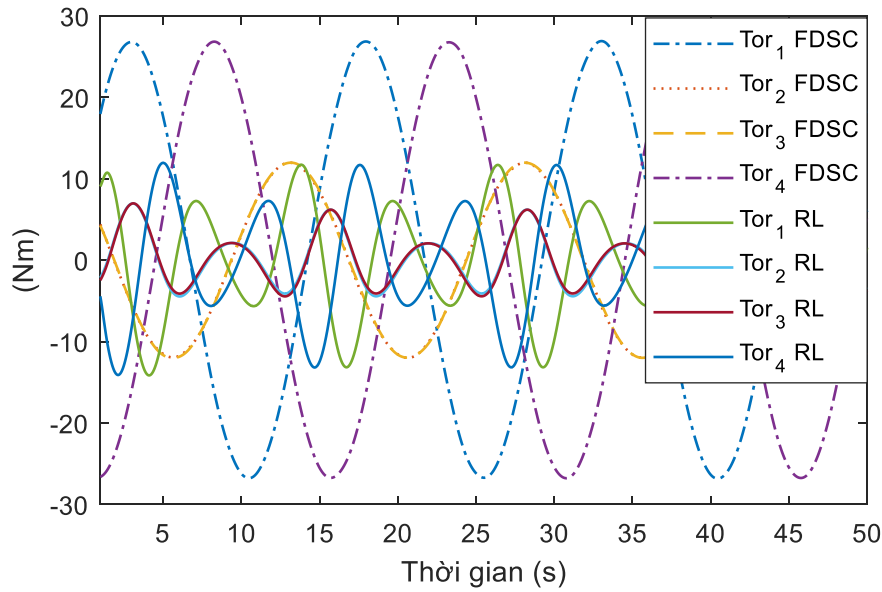
Hình 3.21 Quỹ đạo di chuyển thuật toán Fuzzy-DSC và RL-AC



Hình 3.22 So sánh sai lệch bám thuật toán Fuzzy-DSC và RL-AC

Bên cạnh đó, thuật toán học tăng cường có khả năng tự học và tối ưu hóa các hành động điều khiển theo thời gian, thuật toán cần thời gian để cho mạng actor và critic tự học sau đó đưa ra hành động giúp robot bám quỹ đạo đặt trong khoảng 18 giây. Sau khi học xong thuật toán có thể tối ưu trong việc bám quỹ đạo và tối ưu năng lượng thể hiện trong hình 3.23 cho thấy tín hiệu điều khiển mô men nhỏ hơn. Tuy

nhiên, thuật toán đòi hỏi thời gian học ban đầu nên sai lệch trong giai đoạn học còn lớn, thời gian hội tụ lâu.



Hình 3.23 So sánh tín hiệu điều khiển Fuzzy-DSC và RL-AC

Nhận xét: Cả hai thuật toán được đề xuất đều có những ưu nhược điểm riêng, và phù hợp với từng ứng dụng nhau. Nếu yêu cầu robot có khả năng tối ưu hóa dài hạn, khả năng tự học từ môi trường thì thuật toán RL-AC là lựa chọn phù hợp, mặc dù quá trình học có thể kéo dài và yêu cầu cấu hình máy tính tính toán lớn. Mặt khác, nếu yêu cầu ổn định ngay lập tức và khả năng thích nghi với nhiễu mạnh mẽ thì thuật toán Fuzzy-DSC là lựa chọn tốt, đặc biệt trong các ứng dụng yêu cầu tính ổn định cao và môi trường ít thay đổi.

3.4. Kết luận chương 3

Trong chương này, thuật toán học tăng cường dựa trên giải thuật ADP sử dụng cấu trúc Actor – Critic đã được nghiên cứu. Thuật toán học tăng cường này đã được đánh giá, phân tích về ưu điểm và nhược điểm, và từ đó đề xuất thuật toán nhằm khắc phục các khó khăn kỹ thuật trong việc điều khiển robot, đặc biệt là với tính chất biến đổi theo thời gian của bài toán điều khiển tối ưu.

Đề xuất thuật toán học tăng cường cấu trúc Actor-Critic điều khiển tối ưu bám quỹ đạo cho robot tự hành đa hướng bốn bánh xe mecanum có trọng tâm thay đổi. Thuật toán được áp dụng sử dụng giải thuật ADP với cấu trúc mạng nơ-ron

Actor- Critic trong thuật toán này được cập nhật đồng bộ và liên tục trong từng bước lặp. Các sai số ước lượng của mạng nơ-ron và sai số bám quỹ đạo đã được chứng minh là có giới hạn chặt chẽ thông qua phân tích dựa trên hàm Lyapunov, đối với hệ biến đổi không phụ thuộc vào thời gian. Kết quả cho thấy thuật toán không chỉ đảm bảo hiệu quả bám quỹ đạo mà còn có tính ổn định cao, kể cả khi có sự tác động của nhiễu bất định từ mô hình. Hơn nữa, thuật toán này có khả năng thích nghi với các thay đổi về khối lượng và trọng tâm của robot.

Các nghiên cứu trong chương 3 được công bố trong công trình sau:

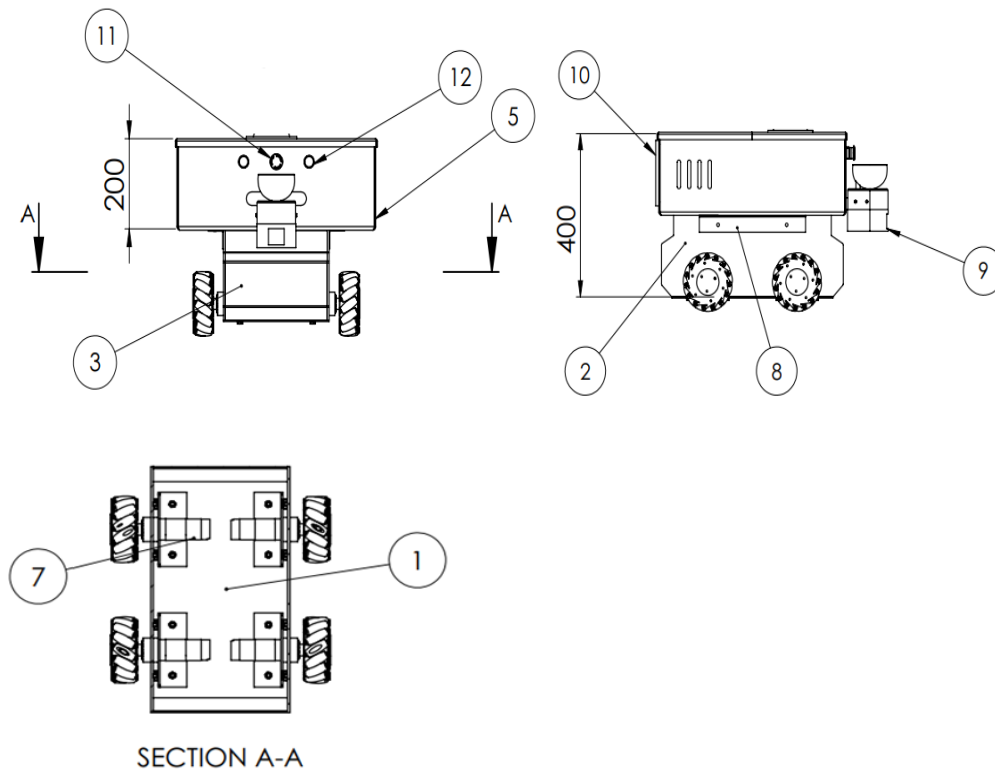
[CT05] **Dong Nguyen Minh**, Tien Ngo Manh, Hiep Do Quang, Nam Dao Phuong, "*Reinforcement Learning-Based Trajectory Control for Mecanum Robot with Mass Eccentricity Considerations*" Journal of Robotics and Control (JRC), Volume 3, Issue 2, March 2022, ISSN: 2715-5072, **Scopus Q2**.

CHƯƠNG 4. THỰC NGHIỆM VỚI MÔ HÌNH ROBOT DI ĐỘNG ĐA HƯỚNG MECANUM

Mô hình robot di động đa hướng mecanum được thiết kế và chế tạo sử dụng trong phòng thí nghiệm. Các thuật toán được đề xuất áp dụng vào mô hình robot để chạy thực nghiệm, từ đó đánh giá khả năng áp dụng thực tế của thuật toán.

4.1. Thiết kế cơ khí cho FMWR

Mô hình robot di động bốn bánh xe mecanum được thiết kế cấu tạo sử dụng bốn bánh xe mecanum được dẫn động độc lập bởi bốn động cơ Planet riêng biệt với kích thước và cách sắp xếp được thể hiện trong hình 4.1



Hình 4.1 Bản vẽ thiết kế cơ khí cho mô hình FMWR

Trong đó các thành phần: 1,2,3- là bộ khung gầm của robot được làm bằng chất liệu thép, 5- hộp thân của robot, 6- bánh xe mecanum, 7- động cơ planet, 8- mặt bích kết nối giữa thân và đế robot, 9- Lidar 306⁰, 10- màn hình điều khiển HMI, 11- nút nhấn dừng khẩn cấp, 12- bộ nút nhấn điều khiển, đèn báo.

Hệ thống dẫn động của mô hình FMWR sử dụng cơ cấu dẫn động trực tiếp trên bánh xe, mỗi bánh xe gắn với một cơ planet có bộ giảm tốc bánh răng hành tinh, hoạt

động ổn định. Mô hình sử dụng bánh xe mecanum chế tạo bằng hợp kim có các con lăn vệ tinh bằng cao su, đường kính mỗi bánh xe là 150 mm. Động cơ planet có công suất 60 W, điện áp 24V DC và sử dụng bộ giảm tốc có tốc độ quay 468 vòng/ phút. Tổng trọng lượng của robot là 30kg đã bao gồm pin, khả năng chịu được tải trọng của robot có thể chịu được là 20kg.

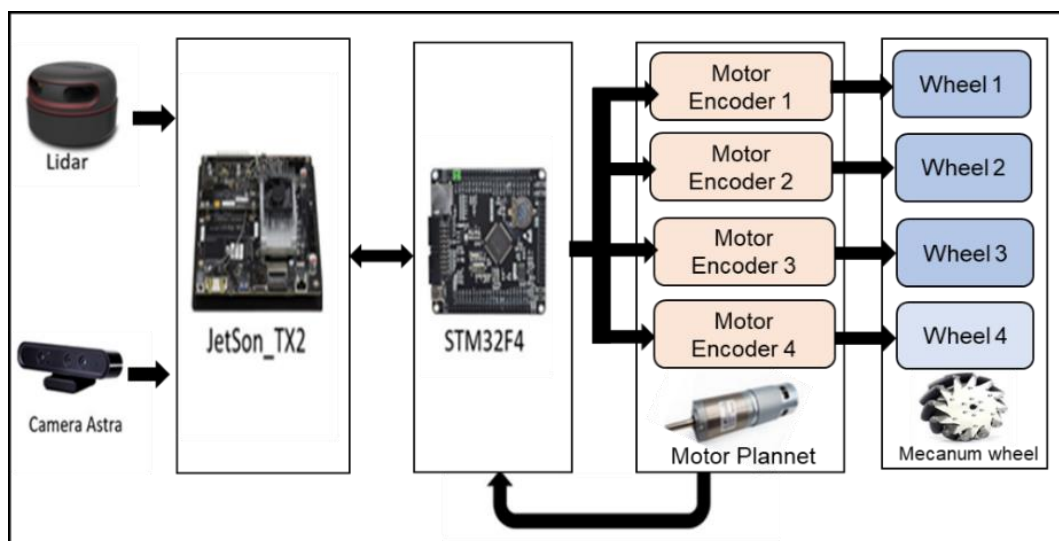


Hình 4.2 Mô hình cơ khí FMWR hoàn thiện

Mô hình cơ khí của FMWR được thiết kế với hệ thống cầu cân bằng đảm bảo bốn bánh xe luôn tiếp xúc với mặt đất kể cả với địa hình bề mặt không bằng phẳng, làm giảm độ trượt của bánh xe trong quá trình điều khiển giúp robot di chuyển được tốt hơn.

4.2. Thiết kế mạch điều khiển cho FMWR

Hệ thống mạch điều khiển của robot được thiết kế với ba khối chính: Khối xử lý trung tâm, khối cảm biến và khối mạch động lực, được thiết kế theo sơ đồ hình 4.3



Hình 4.3 Cấu trúc hệ thống điều khiển của FMWR

Cấu trúc mạch điều khiển được sử dụng bao gồm:

- Khối xử lý trung tâm: Khối này đóng vai trò xử lý trung tâm, được sử dụng bộ chip xử lý hiệu năng cao Jetson-TX2 có nhiệm vụ tính toán, thu thập các tín hiệu từ khối cảm biến và đưa ra các mệnh lệnh đến khối cơ cấu chấp hành. Chạy chương trình ROS, thu thập các dữ liệu từ cảm biến, encoder, camera, LIDAR, IMU và xử lý sau đó gửi tín hiệu đến cho mạch điều khiển.

- Khối cảm biến: Sử dụng Camera, cảm biến tiệm cận và Rplidar quét laser sẽ đưa tín hiệu về khối xử lý trung tâm để xây dựng bản đồ 2D, 3D. Rplidar có tác dụng quét laser 360° giúp cho robot xác định được vị trí bản đồ và nhận diện vật cản với khoảng cách xa.

- Khối mạch động lực: Có nhiệm vụ để điều khiển cơ cấu chấp hành, trong đó sử dụng mạch cầu H (Driver điều khiển động cơ) sử dụng các MOSFET(BJT) là mạch công suất điều khiển các động cơ bánh xe, kết hợp các Encoder để phản hồi tốc độ chính xác các vòng quay của bánh xe. Ngoài ra còn có thêm các cổng mở rộng tín hiệu I/O để có thể điều khiển các cánh tay robot hay cơ cấu chấp hành khác.

- Khối mạch truyền thông RS-232, RF nhận tín hiệu điều khiển từ tay cầm điều khiển hoặc điện thoại di động khi muốn điều khiển trực tiếp.

- Khối điều khiển của robot được giao tiếp thông qua chuẩn truyền thông Ethernet: truyền dữ liệu từ Jetson-TX2 đến khối điều khiển một cách thuận tiện và giảm độ trễ khi truyền, Màn hình HMI được sử dụng điều khiển và giám sát các thông số của robot.



Hình 4.4 Mô hình robot di động bốn bánh xe mecanum hoàn thiện

Hệ thống nguồn điện cung cấp năng lượng cho robot được sử dụng pin Lippo, cung cấp thời gian hoạt động liên tục của robot là 30 phút.

4.3. Hệ điều hành ROS

ROS (Robot Operating System) [107, 108] là phần mềm mã nguồn mở được thiết kế để hỗ trợ phát triển các ứng dụng robot. ROS cung cấp các công cụ, thư viện và các phần mềm hỗ trợ giúp các nhà phát triển dễ dàng xây dựng và điều khiển các hệ thống robot.

ROS chia ứng dụng robot thành các node (nút), mỗi node thực hiện một tác vụ cụ thể, ví dụ như cảm biến, điều khiển động cơ, hoặc xử lý hình ảnh. ROS cung cấp một phương thức giao tiếp giữa các node thông qua các topic, services và actions, của ROS chia thành các package. Mỗi package có thể chứa mã nguồn, cấu hình, thư viện, tài liệu, và các tài nguyên khác. Một số công cụ quan trọng trong ROS:

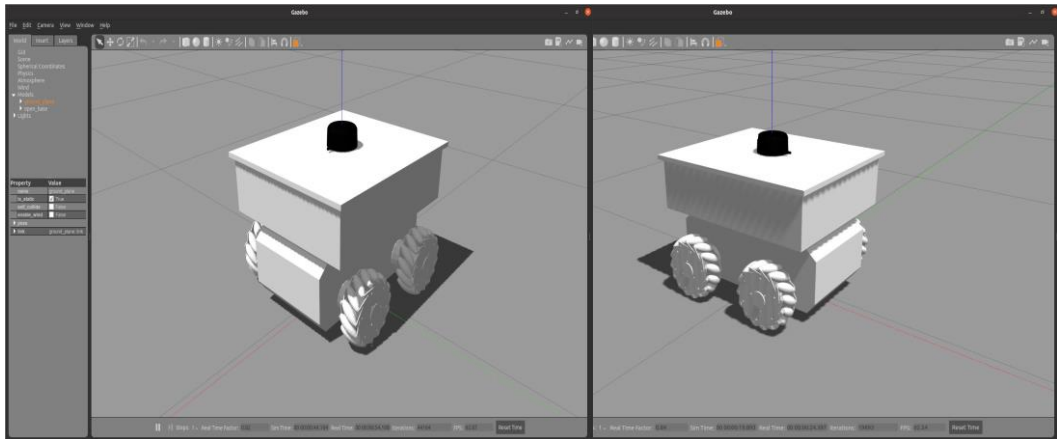
- **RViz:** Một công cụ để trực quan hóa dữ liệu và trạng thái của robot, bao gồm bản đồ, cảm biến, và các mô hình mô phỏng.
- **Gazebo:** Là một phần mềm mô phỏng robot mạnh mẽ, giúp thử nghiệm và mô phỏng các môi trường thực tế mà không cần phần cứng robot thật.
- **Roslaunch:** Một công cụ giúp khởi động các node và cấu hình hệ thống robot một cách tự động.

Ứng dụng của ROS trong lĩnh vực robot di động rất quan trọng. ROS cung cấp các thư viện và công cụ hỗ trợ các phương pháp SLAM (Simultaneous Localization and Mapping), giúp robot xây dựng bản đồ của môi trường xung quanh trong khi vẫn định vị chính xác vị trí của mình. Các thuật toán SLAM phổ biến như GMapping, Hector SLAM, và Cartographer có thể được sử dụng trong ROS để cho phép robot di chuyển tự động trong môi trường bất định. Điều khiển chuyển động (Motion Planning): ROS hỗ trợ các thuật toán lên kế hoạch di chuyển (motion planning) như A*, RRT, và Dijkstra giúp robot tính toán quỹ đạo toàn cục và cục bộ khi gặp vật cản. Thư viện MoveIt! trong ROS là một công cụ rất mạnh mẽ cho lập kế hoạch chuyển động cho robot di động, bao gồm việc tính toán các quỹ đạo và tránh vật cản.

4.4. Chạy thử nghiệm mô hình trên Gazebo

Gazebo [107, 109, 110] là phần mềm mô phỏng mạnh mẽ được tích hợp sử dụng trong môi trường ROS. Nó cho phép mô phỏng robot trong môi trường giống với môi trường thực tế. Hơn nữa, nó cho phép chạy thử nghiệm các thuật toán mà không cần đến phần cứng vật lý. Trong mô phỏng một mô hình robot mecanum được xây dựng với các thông số giống với các thông số của mô hình robot thực tế.

- Kích thước robot: 0,4 x 0,6 (m)
- Bán kính bánh xe: 0.075 (m)
- Vận tốc di chuyển theo phương x và y: 1.5 (m/s)
- Vận tốc góc: 0.5 rad/s
- Cảm biến Lidar: Khoảng cách quét: 0.2 ÷ 10 m, Độ phân giải: 1°, Góc quét: 360°

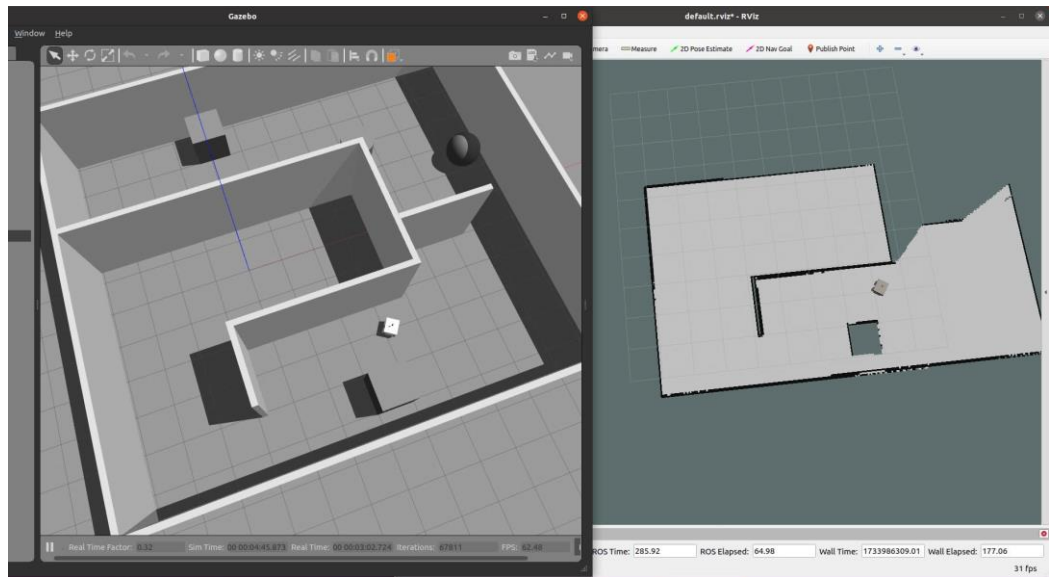


Hình 4.5 Thiết kế mô hình FMWR trong Gazebo

Trên mô hình robot Rplidar được thiết kế đặt ở trên cùng của robot, có thể quét được toàn cảnh 360°, có thể xác định được các vật cản và bản đồ của môi trường xung quanh. Môi trường mô phỏng của robot được thiết kế trong không gian 3D với các vật cản, các phòng được phân chia khác nhau giúp tạo các tình huống mô phỏng giống môi trường thực tế để có thể mô phỏng hệ thống điều hướng cho robot như trong (hình 4.5).

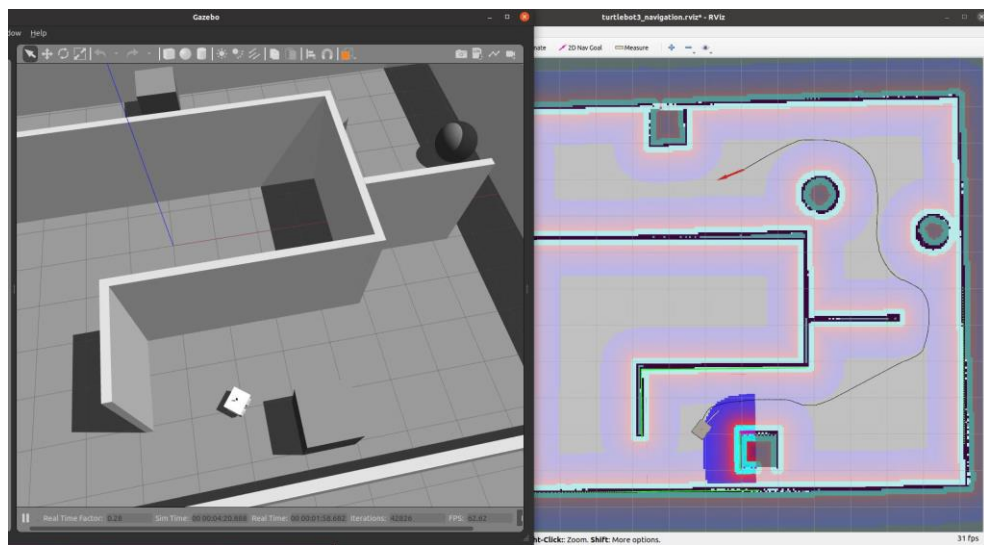
- **Các quá trình mô phỏng hoạt động:**

- *Bản đồ hóa môi trường:* Robot thực hiện quá trình SLAM, sử dụng gói Gmapping để thu thập các tín hiệu từ Lidar quét được, từ đó xác định được môi trường xung quanh và bản đồ hóa môi trường, quá trình SLAM được thể hiện trong hình 4.6



Hình 4.6 Quá trình SLAM bản đồ hóa môi trường

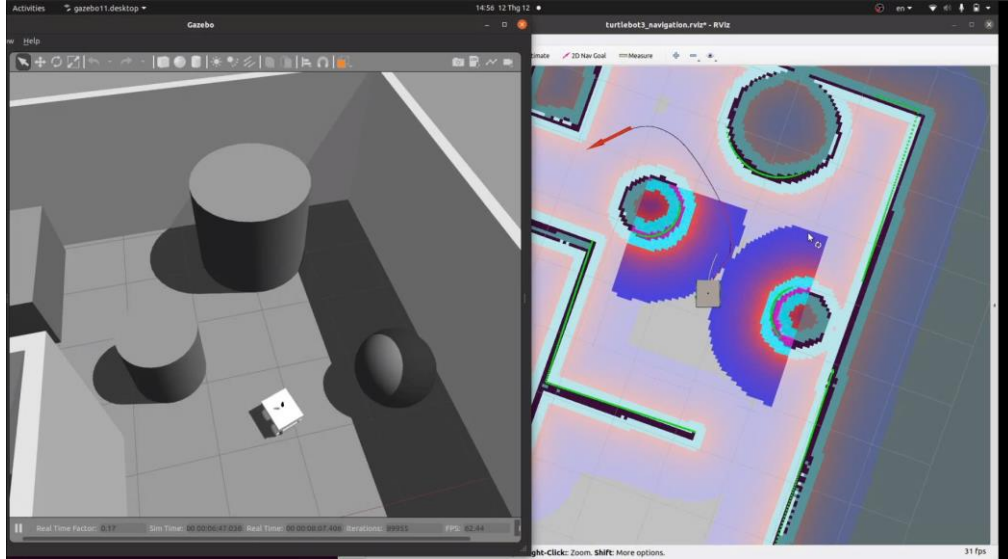
➤ *Điều hướng chuyển động*: Quỹ đạo di chuyển của robot được tính toán sao cho robot di chuyển đến vị trí đích bất kỳ được đặt trước trên bản đồ và đồng thời tránh được vật cản trong quá trình di chuyển, kết quả của quá trình điều hướng được thể hiện trong hình 4.7



Hình 4.7 Quỹ đạo di chuyển của FMWR

Quỹ đạo chuyển động của robot được chia ra làm hai loại: Đường màu đỏ là quỹ đạo toàn cục là quỹ đạo được tính toán sao cho nó là quỹ đạo ngắn nhất từ vị trí hiện tại của robot đến điểm đích. Đường màu xanh là quỹ đạo cục bộ: quỹ đạo này khi gặp vật cản trên đường di chuyển sẽ điều chỉnh quỹ đạo của robot sao cho robot có thể di chuyển an toàn, tránh vật cản mà vẫn đảm bảo robot tiến đến vị trí đích.

Khi robot di chuyển tới một vị trí có vật cản và nhận thấy có khả năng robot va chạm, thì quỹ đạo cục bộ ngay lập tức điều chỉnh lại quỹ đạo cho robot di chuyển vòng tránh xa vật cản hơn nhưng vẫn đảm bảo cho robot di chuyển hướng đến đích, kết quả mô phỏng được thể hiện trong hình 4.8



Hình 4.8 Quá trình điều hướng robot tránh vật cản

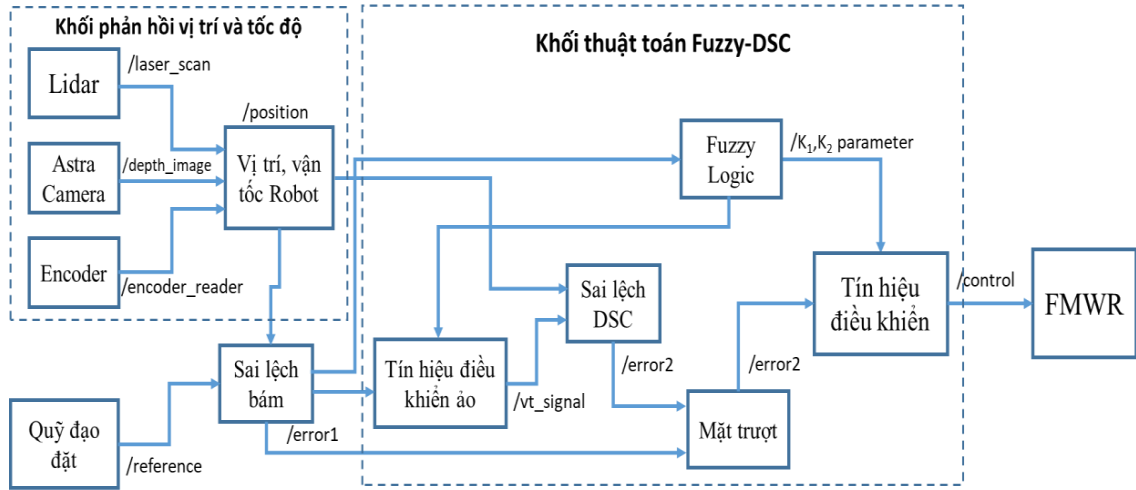
Trong quá trình lập quỹ đạo cục bộ thuật toán TEB [111] được sử dụng sẽ cho ra kế hoạch di chuyển với khoảng cách có thể thay đổi khi gặp vật cản động. Đối với vật cản thấp hơn vùng quét của lidar thì lúc này tín hiệu laser scan không thể quét được vật cản. Khi đó, tín hiệu thu được từ camera sẽ thay thế tín hiệu laser scan để phát hiện vật cản này, tín hiệu đó được đưa vào phần phát hiện vật cản của hệ thống nhận thức. Tiếp theo, quỹ đạo cục bộ sẽ được tính toán và đưa ra quỹ đạo phù hợp để cho robot để có thể tránh được các loại vật cản này, thể hiện ở đường màu xanh. Sau khi robot di chuyển qua vùng vật cản đường quỹ đạo cục bộ sẽ được tự động tính toán sau đó di chuyển tới với đường quỹ đạo toàn cục đã xác định.

4.5. Thực nghiệm mô hình robot

➤ Cài đặt thực nghiệm thuật toán Fuzzy-DSC

Thuật toán điều khiển được lập trình trên máy tính nhúng Jetson-TX2 làm bộ xử lý trung tâm. Các dữ liệu từ cảm biến, lidar và các thiết bị ngoại vi được đưa vào hệ thống thông qua các node, mỗi node là một chương trình thực hiện các tác vụ khác nhau. Sơ đồ khối để cài đặt thuật toán thể hiện trong hình 4.9, thuật toán Fuzzy-DSC

được cài đặt nhúng trên bộ xử lý trung tâm. Các tín hiệu điều khiển cuối cùng Tor được đưa ra mạch điều khiển cơ cấu chấp hành STM32 băm xung PWM điều khiển động cơ từng bánh xe.

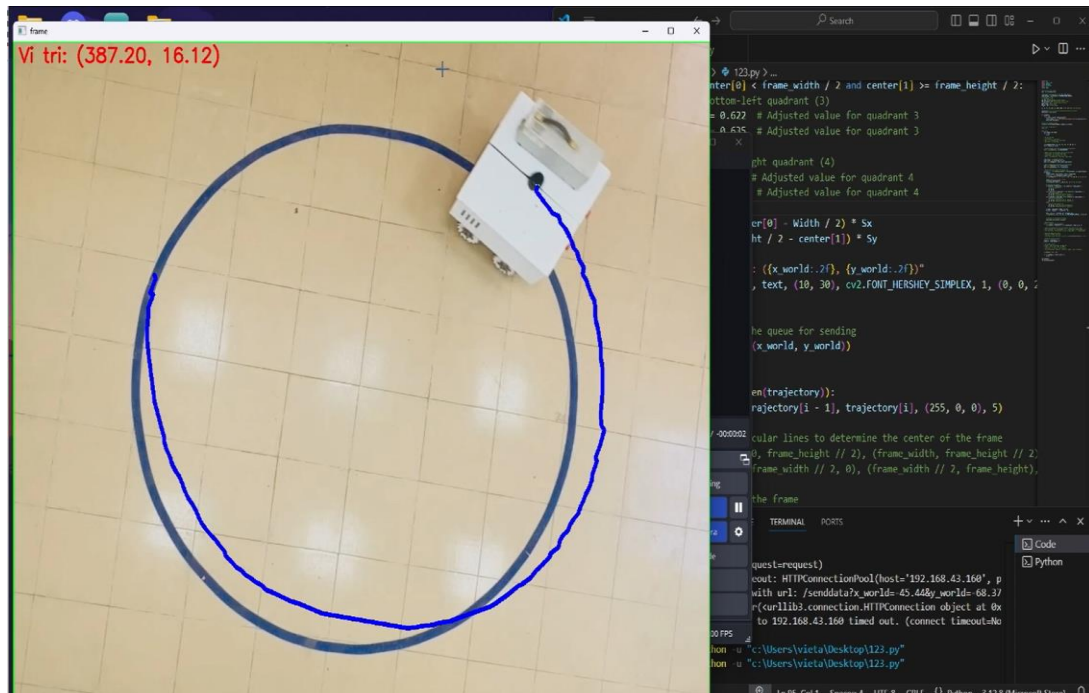


Hình 4.9 Sơ đồ khối lập trình thực nghiệm thuật toán Fuzzy-DSC

Tham số của mô hình robot được đưa vào bộ điều khiển như (2.27), các tham số của bộ điều khiển DSC được thiết kế trong mục lục 2.5 và luật mờ của bộ chỉnh định Fuzzy được lựa chọn như trong mục 2.6 được cài đặt trên Jetson-TX2.

Mô hình robot được thử nghiệm tại phòng nghiên cứu Tự động hóa, Viện Vật lý, Viện Hàn lâm khoa học và Công nghệ Việt Nam. Quá trình chạy thử nghiệm sẽ được lập trình, điều khiển trên mô hình robot đã được chế tạo, mặt sàn thực nghiệm robot là nền nhà phẳng với các điều kiện môi trường của phòng thí nghiệm. Các thiết bị thực nghiệm bao gồm:

- Mô hình FMWR như hình 4.2
- Máy tính có phần mềm Rviz giao tiếp với máy tính nhúng Jetson-TX2 qua kết nối wi-fi có thể điều khiển, quan sát trạng thái, vị trí của robot.
- Vị trí gốc của robot được đo thông qua tính toán từ encoder kết hợp với cảm biến Lidar.
- Các quỹ đạo di chuyển trong quá trình thực nghiệm được dán dưới mặt sàn với các vạch rộng 3cm để dễ dàng quan sát đánh giá.
- Hệ thống camera phía trên phòng thí nghiệm theo dõi quá trình chạy bám quỹ đạo của robot thông qua xử lý ảnh, chọn điểm màu đen (Lidar) ở tâm của robot để xác định vị trí và vẽ đường quỹ đạo thực.



Hình 4.10 Camera theo dõi quỹ đạo di chuyển robot

➤ **Kịch bản thực nghiệm FMWR như sau:**

Các quỹ đạo di chuyển khác nhau được đưa vào để đánh giá khả năng di chuyển bám quỹ đạo của robot. Mô hình robot được thực nghiệm với các vận tốc cố định là 0.3m/s, các quỹ đạo thực nghiệm gồm quỹ đạo đường tròn, quỹ đạo hình số 8 và quỹ đạo đường zích zắc, thực nghiệm trong 2 trường hợp: thứ nhất là robot không mang tải trọng, có độ lệch tâm không đổi. Trường hợp 2 là robot mang tải trọng và có trọng tâm thay đổi.

➤ **Kịch bản 1: thực nghiệm với quỹ đạo đường tròn:**

- Quỹ đạo đường tròn có bán kính 1,5m, vận tốc di chuyển của robot là 0.3 m/s, vị trí đặt ban đầu của robot là ở tâm của đường tròn quỹ đạo.
- Thời gian thực nghiệm là 50s, điều kiện thực nghiệm phòng thí nghiệm

Trường hợp 1: Robot không mang tải trọng, trọng tâm của robot không đổi:

- Kết quả thực nghiệm: Giai đoạn đầu sau khoảng thời gian 4 giây, robot đã ổn định và bám sát vào quỹ đạo đường tròn, với sai số giữa quỹ đạo thực tế và quỹ đạo mục tiêu nhỏ. Điều này cho thấy thuật toán điều khiển bám quỹ đạo đã giúp robot nhanh chóng đạt được trạng thái ổn định.

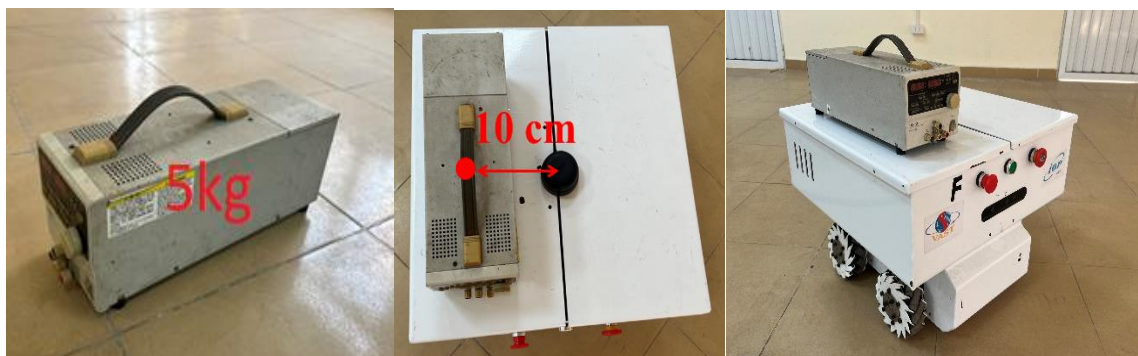


Hình 4.11 Thực nghiệm với quỹ đạo tròn trọng tâm không đổi

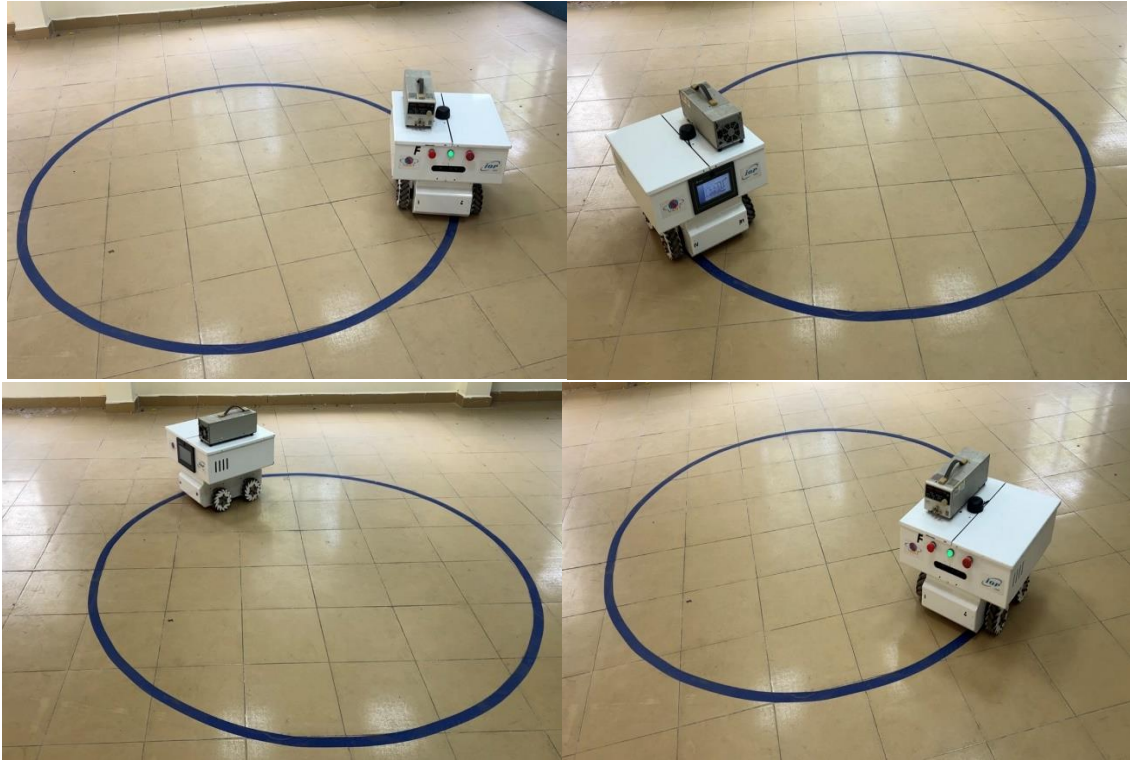
Quá trình di chuyển: Trong suốt quá trình di chuyển, robot duy trì tốc độ ổn định 0.3 m/s và không gặp phải các dao động lớn trong quá trình bám quỹ đạo. Sai số bám quỹ đạo luôn được duy trì dưới 5 cm, cho thấy thuật toán điều khiển hoạt động hiệu quả và chính xác. Robot không bị lệch khỏi quỹ đạo và duy trì hướng di chuyển ổn định trong suốt thời gian thử nghiệm. Không có sự thay đổi đáng kể về vận tốc hay gia tốc, chứng minh rằng hệ thống điều khiển đảm bảo ổn định.

Trường hợp 2: Robot mang tải trọng $\Delta m = 5kg$, độ lệch trọng tâm $d = 0.1m$.

Thêm vật nặng có khối lượng 5kg vào robot, gây nên sự thay đổi trọng tâm so với trạng thái không mang tải trọng.



Hình 4.12 Thêm tải trọng 5kg và đặt lệch trọng tâm trên robot



Hình 4.13 Thực nghiệm với tải trọng 5kg và đặt lệch trọng tâm trên robot

Kết quả thực nghiệm: Khi robot mang tải trọng 5kg, trọng tâm của robot thay đổi, điều này có thể ảnh hưởng đến khả năng bám quỹ đạo. Tuy nhiên, sau khoảng 6 giây từ khi bắt đầu di chuyển, robot đã ổn định và bắt đầu bám sát vào quỹ đạo đường tròn. So với trường hợp không mang tải, robot mất thêm một khoảng thời gian để ổn định do ảnh hưởng của trọng tâm thay đổi.

Trong suốt quá trình di chuyển, robot có xu hướng dao động nhẹ xung quanh quỹ đạo mục tiêu. Sai số giữa quỹ đạo thực tế và quỹ đạo mục tiêu dao động trong phạm vi từ 5 cm đến 10 cm. Điều này cho thấy tải trọng thêm vào và sự thay đổi trọng tâm có ảnh hưởng đến khả năng bám quỹ đạo của robot, tuy nhiên vẫn trong giới hạn chấp nhận được.

Ảnh hưởng của tải trọng: Mặc dù robot vẫn bám được quỹ đạo, nhưng do trọng tâm thay đổi, robot có xu hướng di chuyển kém mượt mà hơn so với trường hợp không mang tải. Tuy nhiên, robot vẫn duy trì tốc độ ổn định và không gặp phải sự bất ổn lớn. Thời gian ổn định tăng thêm một chút so với trường hợp không mang tải trọng.

- Thời gian ổn định: Trường hợp robot không mang tải trọng đạt trạng thái ổn định sau 4 giây, trong khi trường hợp robot mang tải trọng 5kg mất khoảng 6 giây để ổn định.

- Sai số bám quỹ đạo: Trong trường hợp không mang tải trọng, sai số bám quỹ đạo luôn dưới 5 cm. Trong khi đó, khi mang tải trọng, sai số dao động từ 5 cm đến 10 cm, cho thấy sự ảnh hưởng của tải trọng và thay đổi trọng tâm đối với khả năng bám quỹ đạo. Mặc dù robot mang tải trọng vẫn có thể bám theo quỹ đạo, nhưng sự thay đổi trọng tâm làm cho quá trình di chuyển có chút dao động.

➤ Kịch bản 2: thực nghiệm với quỹ đạo hình số 8:

Trường hợp 1: Robot không mang tải trọng, trọng tâm không đổi

- Quỹ đạo: Đường hình số 8 (gồm hai vòng tròn đối xứng nối với nhau).
- Bán kính quỹ đạo: 1.5 m cho mỗi vòng tròn.
- Vận tốc di chuyển của robot: 0.3 m/s.
- Vị trí ban đầu: Robot bắt đầu tại điểm giao nhau của hai vòng tròn trong quỹ đạo hình số 8.
- Thời gian thực nghiệm: 50 giây



Hình 4.14 Thực nghiệm với quỹ đạo hình số 8 trọng tâm không đổi

Sau khoảng 5 giây, robot đã hoàn thành bước đầu tiên của quỹ đạo hình số 8, bám theo phần quỹ đạo đường tròn đầu tiên mà không gặp sự cố. Trong suốt quá trình di chuyển, robot giữ vững sai số bám quỹ đạo dưới 5 cm, điều này chứng tỏ thuật toán điều khiển bám quỹ đạo có hiệu quả tốt ngay từ giai đoạn đầu.

Quá trình di chuyển: Robot tiếp tục di chuyển sang vòng thứ hai của hình số 8 mà không bị lệch ra ngoài quỹ đạo. Sai số giữa quỹ đạo thực tế và quỹ đạo lý thuyết vẫn luôn trong khoảng chấp nhận được (dưới 5 cm), với độ ổn định cao. Tốc độ di chuyển của robot được duy trì ổn định và không có hiện tượng trượt hay lệch hướng. robot không gặp phải dao động bất thường trong quá trình bám theo quỹ đạo. Trong suốt thời gian thử nghiệm, robot tiếp tục giữ vững được trạng thái ổn định và bám theo quỹ đạo hình số 8 mà không gặp sự cố đáng kể.

Trường hợp 2: Robot mang tải trọng 5kg, trọng tâm thay đổi

Trọng lượng robot tăng thêm 5kg, độ lệch trọng tâm của robot với $d=0.1m$



Hình 4.15 Thực nghiệm quỹ đạo hình số 8 trọng tâm thay đổi

Kết quả thực nghiệm: Khi robot mang thêm tải trọng, trọng tâm thay đổi, khiến thời gian ổn định ban đầu của robot kéo dài thêm một chút. Sau khoảng 7 giây, robot

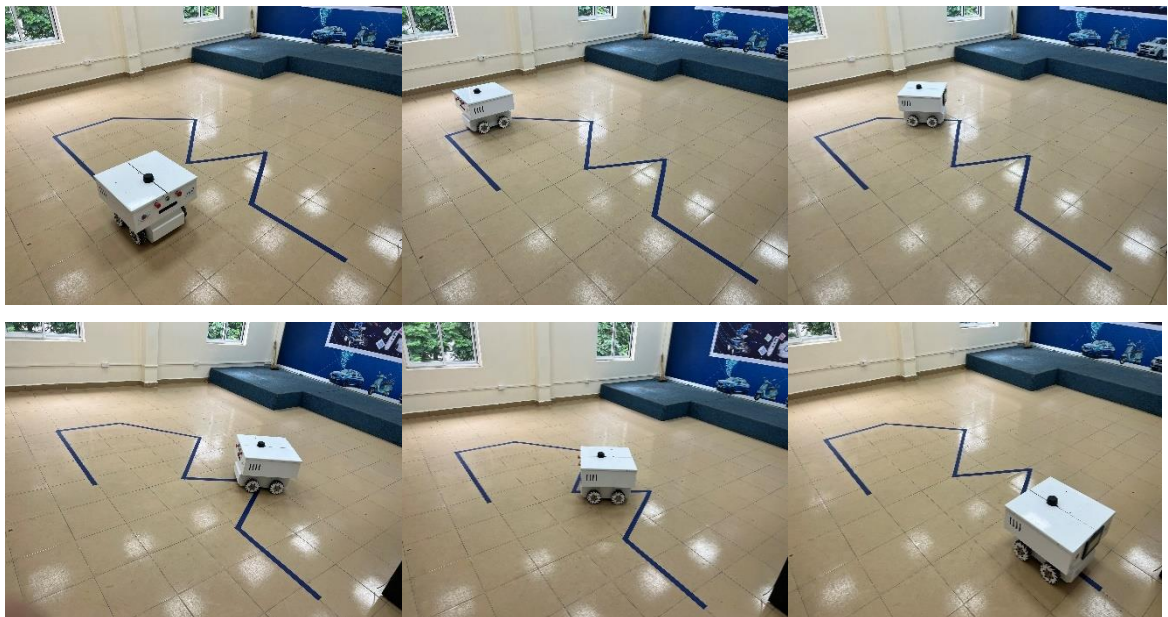
bắt đầu bám vào quỹ đạo hình số 8 và tiếp tục di chuyển với sự ổn định dần. So với trường hợp không mang tải, thời gian ổn định của robot lâu hơn, do sự thay đổi trọng tâm gây ảnh hưởng đến các tham số động lực học.

Quá trình di chuyển: Robot tiếp tục di chuyển qua vòng đầu tiên của quỹ đạo, nhưng sai số bám quỹ đạo dao động từ 5 cm đến 10 cm. Đây là sự ảnh hưởng từ thay đổi trọng tâm khi robot mang thêm tải trọng. Trong vòng thứ hai của quỹ đạo hình số 8, robot đã có những dao động nhẹ nhưng vẫn duy trì được khả năng bám theo quỹ đạo. Mặc dù có sự dao động, robot vẫn hoàn thành quỹ đạo hình số 8 một cách khá ổn định.

Ảnh hưởng của tải trọng: Mặc dù robot vẫn có thể bám theo quỹ đạo, nhưng với sự thay đổi trọng tâm, robot có xu hướng di chuyển ít mượt mà hơn và gặp một số dao động nhẹ, đặc biệt khi chuyển từ vòng tròn này sang vòng tròn kia trong hình số 8. Tuy nhiên, thuật toán điều khiển vẫn đảm bảo rằng robot không lệch quá xa khỏi quỹ đạo và vẫn hoàn thành quỹ đạo hình số 8 trong thời gian quy định.

➤ **Kịch bản 3: thực nghiệm với quỹ đạo hình zích zắc:**

Chạy thử nghiệm FMWR với quỹ đạo hình zích zắc gồm các đoạn đường thẳng thay đổi hướng liên tiếp, góc của các đoạn đường thẳng: 45°



Hình 4.16 Thực nghiệm mô hình FMWR với quỹ đạo đường zích zắc

Khi robot bắt đầu di chuyển đã dần ổn định để bám theo quỹ đạo hình zích zắc. Trong suốt quá trình thử nghiệm, sai số bám quỹ đạo của robot dao động trong khoảng từ 2 cm đến 4 cm.

Quá trình di chuyển: Robot chuyển từ một đoạn đường thẳng sang đoạn tiếp theo một cách mượt mà. Mặc dù có một số dao động nhỏ ở các khúc quanh do mô hình động lực học và của động cơ, robot duy trì quỹ đạo một cách chính xác.

Hiệu suất bám quỹ đạo: Sai số bám quỹ đạo trong suốt quá trình di chuyển không vượt quá 4 cm, và robot không gặp phải hiện tượng lệch hướng hoặc trượt bánh. Quá trình chuyển hướng từ 45° sang 135° được thực hiện mượt mà, và robot tiếp tục duy trì vị trí đúng trên quỹ đạo

Nhận xét: Trong tất cả các quỹ đạo thử nghiệm (đường tròn, hình số 8, hình zích zắc), robot đều có khả năng bám sát quỹ đạo một cách chính xác, với sai số bám quỹ đạo ở mức chấp nhận được. Ảnh hưởng của tải trọng: việc mang tải trọng (5kg) đã làm tăng nhẹ sai số bám quỹ đạo, nhưng robot vẫn có thể di chuyển theo quỹ đạo yêu cầu mà không gặp phải sự gián đoạn lớn. Tải trọng làm thay đổi đặc tính động học của robot và có ảnh hưởng đến tốc độ và phản ứng khi thay đổi hướng. Tuy nhiên, thuật toán điều khiển đã thể hiện khả năng thích nghi tốt với sự thay đổi này, đảm bảo robot vẫn di chuyển ổn định.

Quá trình thực nghiệm đã chứng minh rằng robot di động đa hướng sử dụng bánh xe mecanum, kết hợp với thuật toán điều khiển đã đề xuất, có khả năng bám quỹ đạo một cách chính xác, ổn định và hiệu quả, cả trong điều kiện không mang tải trọng và khi mang thêm tải trọng. Các kết quả thu được cho thấy tính khả thi và tiềm năng của hệ thống robot này trong các ứng dụng thực tế, đồng thời khẳng định được sự hiệu quả của thuật toán điều khiển trong việc duy trì ổn định và tối ưu hóa quá trình di chuyển của robot.

4.6. Kết luận chương 4

Chương 4 tập trung vào việc thực hiện các thử nghiệm thực tế để đánh giá hiệu quả của các thuật toán điều khiển được nghiên cứu trong luận án. Các nội dung nghiên cứu và kết quả đạt được trong chương này bao gồm:

- **Thiết kế, chế tạo mô hình robot di động bốn bánh xe mecanum:** Mô hình cơ khí của robot đã được thiết kế và chế tạo hoàn chỉnh, đảm bảo tính ổn định và khả năng di chuyển đa hướng với các bánh xe mecanum.
- **Chạy thử nghiệm trên phần mềm Gazebo:** Các thuật toán điều khiển đã được kiểm tra và chạy thử nghiệm trên phần mềm Gazebo. Giúp đánh giá trước hiệu suất của các thuật toán trong điều kiện mô phỏng, từ đó đưa ra những nhận xét về khả năng hoạt động của robot trước khi thực hiện trên thực tế.
- **Thực nghiệm trên mô hình robot thực với thuật toán Fuzzy-DSC:** Các thử nghiệm thực tế đã được tiến hành với robot thực, sử dụng các quỹ đạo đường tròn, hình số 8 và hình zích zắc để đánh giá độ chính xác và hiệu quả của các thuật toán điều khiển. Các kết quả thu được cho thấy robot có thể bám sát quỹ đạo một cách chính xác, ổn định, kể cả trong các tình huống thay đổi trọng tâm và tải trọng.

Các kết quả thử nghiệm đã cho thấy khả năng hoạt động ổn định và chính xác của hệ thống, đồng thời cung cấp cơ sở vững chắc để áp dụng thuật toán vào các ứng dụng thực tế trong tương lai

Nghiên cứu trong chương 4 được công bố trong các công trình sau:

[CT06] **Nguyễn Minh Đông**, Đỗ Quang Hiệp, Ngô Mạnh Tiến, Nguyễn Đức Thắng, Nguyễn Minh Dương, Nguyễn Tiến Bắc, “*Xây dựng hệ robot di động có gắn tay máy cộng tác Cobot 6 bậc tự do trên hệ điều hành ROS*” REV-ECIT 2022, NXB Thông tin và TT; 12/2022; pp 388-393, ISBN 978-604-80-7468-5.

[CT07] **Nguyễn Minh Đông**, Ngô Mạnh Tiến, Nguyễn Mạnh Cường, Đỗ Quang Hiệp, Vũ Việt Anh, Bùi Đình Duy, Nguyễn Tuấn Anh, “*Xây dựng Hệ thống Điều hướng Thông minh cho robot di động dạng Mecanum dựa trên nền tảng ROS*” Proceedings the 7th Vietnam International Conference and Exhibition on Control and Automation VCCA-2024, ISBN 978-604-937-357-2.

KẾT LUẬN VÀ KIẾN NGHỊ

Bài toán điều khiển bám quỹ đạo cho robot di động đa hướng là một vấn đề quan trọng trong lĩnh vực điều khiển tự động và robot học, đặc biệt đối với các loại robot có khả năng di chuyển linh hoạt đa hướng mà không cần phải thay đổi góc hướng, từ đó nâng cao khả năng hoạt động và ứng dụng trong thực tế. Thách thức đặt ra là làm sao để robot có thể bám theo đúng quỹ đạo mong muốn một cách chính xác và hiệu quả, đồng thời đảm bảo tính ổn định và an toàn trong suốt quá trình di chuyển. Tuy nhiên trong thực tế FMWR làm việc phải mang thêm tải trọng, hàng hóa dẫn đến trọng tâm của robot thay đổi không trùng với tâm hình học của robot, điều đó gây ảnh hưởng đến chất lượng bám quỹ đạo của robot. Luận án đã tập trung nghiên cứu và đề xuất thuật toán giải quyết vấn đề đặt ra và đạt được các kết quả quan trọng

1. Các kết quả đạt được của Luận án.

Trong luận án đã trình bày các nghiên cứu và đề xuất các thuật toán nhằm nâng cao chất lượng điều khiển bám quỹ đạo cho robot di động đa hướng bốn bánh xe mecanum có trọng tâm thay đổi. Luận án đề xuất những đóng góp chính như sau:

➤ **Đề xuất thuật toán điều khiển mặt trượt động chỉnh định thích nghi mờ để nâng cao chất lượng điều khiển bám quỹ đạo cho robot tự hành đa hướng bốn bánh xe mecanum có trọng tâm thay đổi.** Phương pháp điều khiển được phát triển trên kết hợp thuật toán điều khiển mặt trượt động với điều khiển logic mờ để hiệu chỉnh tham số của bộ điều khiển. Thuật toán được thiết kế với đầu vào của bộ chỉnh định mờ là sai lệch vị trí và vận tốc thông qua mô hình mờ sugeno, để chỉnh định trực tuyến các tham số điều khiển của thuật toán điều khiển mặt trượt động. Phương pháp điều khiển được đề xuất áp dụng cho FMWR có trọng tâm thay đổi, đảm bảo cho robot bám quỹ đạo tốt hơn ngay cả khi khối lượng và trọng tâm thay đổi, thời gian robot tiến đến quỹ đạo nhanh và sai lệch bám nhỏ.

➤ **Đề xuất thuật toán học tăng cường cấu trúc Actor-Critic điều khiển tối ưu bám quỹ đạo cho robot tự hành đa hướng bốn bánh xe mecanum có trọng tâm thay đổi.** Thuật toán học tăng cường được phát triển sử dụng quy hoạch động của Bellman, nghiệm của phương trình HJB được xấp xỉ thông qua giải thuật lặp PI để giải phương trình Lyapunov. Thuật toán sử dụng mạng nơ-ron với cấu trúc Actor – Critic để giải xấp xỉ nghiệm phương trình HJB. Trong đó, mạng Actor NNs dùng để

xấp xỉ luật điều khiển tối ưu, Critic NNs dùng để xấp xỉ hàm Bellman. Tín hiệu điều khiển từ Actor sẽ liên tục được cập nhật và đưa vào kích thích hệ thống cho đến khi tín hiệu điều khiển là tối ưu, các luật cập nhật của critic NNs được thực hiện đồng thời trong một bước lặp dựa theo tối thiểu hóa sai số hàm Bellman tương ứng. Thuật toán được chứng minh tính ổn định theo tiêu chuẩn hàm Lyapunov. Các kết quả mô phỏng của thuật toán được áp dụng điều khiển bám quỹ đạo cho FMWR cho thấy khả năng bám quỹ đạo của robot và ổn định ngay cả với trọng tâm của robot thay đổi, chịu được ảnh hưởng của các nhiễu ngoại lực tác động và hoạt động ổn định với sai lệch nhỏ.

2. Hướng phát triển của Luận án

Hiện nay các nghiên cứu về các thuật toán học máy, học tăng cường hay sử dụng trí tuệ nhân tạo AI vào điều khiển bám quỹ đạo cho robot đa hướng đang ngày càng được tập trung nghiên cứu. Trong tương lai việc nghiên cứu các thuật toán có thể được phát triển như sau:

- Phát triển thuật toán với giải thuật ADP sử dụng mạng nơ-ron nhân tạo nhiều lớp để tăng khả năng thích ứng của thuật toán với các môi trường hoạt động của robot.
- Kết hợp giải thuật ADP – AC với mạng nơ-ron nhiều lớp để khắc phục nhược điểm còn tồn tại của AC-NNs.
- Nghiên cứu áp dụng thuật toán học tăng cường vào mô hình thực tế và đánh giá kết quả để có thể đưa vào ứng dụng thực tiễn.

DANH MỤC CÔNG TRÌNH KHOA HỌC ĐÃ CÔNG BỐ CỦA TÁC GIẢ

[CT01] **Dong Nguyen Minh**, Tien Ngo Manh, Hiep Do Quang, Nam Duy Buy, Linh Bui Van, Cuong Nguyen Manh, “*Backstepping Aggregated with Sliding Mode Control Algorithm for Mecanum Wheeled Mobile Robot*” Proceedings the 6th Vietnam International Conference and Exhibition on Control and Automation VCCA-2021, ISBN 978-604-73-5569-3.

[CT02] **Nguyễn Minh Đông**, Đỗ Quang Hiệp, Ngô Mạnh Tiến, Bùi Văn Bắc & Chu Văn Vương, “*Xây dựng thuật toán điều khiển thích nghi mờ mặt trượt động điều khiển bám quỹ đạo cho mecanum robot*” Hội thảo quốc gia lần thứ XXV: Một số vấn đề chọn lọc của Công nghệ thông tin và truyền thông – Hà nội 8-9/12/2022.

[CT03] **Nguyễn Minh Đông**, Ngô Mạnh Tiến, Đỗ Quang Hiệp, Bùi Văn Bắc, Chu Văn Vương & Nguyễn Đức Thắng. “*Xây dựng bộ điều khiển mặt trượt động điều khiển bám quỹ đạo cho robot di động đa hướng bốn bánh Mecanum*” Journal of Military Science and Technology, (FEE) 2022, 41–49.

[CT04] **Minh Dong Nguyen**, Hiep, D. Q., Nam, D. P., & Tien, N. M. (2025). “*Trajectory tracking control for four-mecanum-wheeled robots considering mass eccentricity: A fuzzy adaptive and dynamic surface control approach*”. Journal of Computer Science and Cybernetics, (1). <https://doi.org/10.15625/1813-9663/21537>

[CT05] **Dong Nguyen Minh**, Tien Ngo Manh, Hiep Do Quang, Nam Dao Phuong, “*Reinforcement Learning-Based Trajectory Control for Mecanum Robot with Mass Eccentricity Considerations*” Journal of Robotics and Control (JRC) 5, no. 5 (2024): 1436-1443, **Scopus Q2**.

[CT06] **Nguyễn Minh Đông**, Đỗ Quang Hiệp, Ngô Mạnh Tiến, Nguyễn Đức Thắng, Nguyễn Minh Dương, Nguyễn Tiến Bắc, “*Xây dựng hệ robot di động có gắn tay máy cộng tác Cobot 6 bậc tự do trên hệ điều hành ROS*” REV-ECIT 2022, NXB Thông tin và TT; 12/2022; pp 388-393, ISBN 978-604-80-7468-5.

[CT07] **Nguyễn Minh Đông**, Ngô Mạnh Tiến, Nguyễn Mạnh Cường, Đỗ Quang Hiệp, Vũ Việt Anh, Bùi Đình Duy, Nguyễn Tuấn Anh, “*Xây dựng Hệ thống Điều hướng Thông minh cho robot di động dạng Mecanum dựa trên nền tảng ROS*” Proceedings the 7th Vietnam International Conference and Exhibition on Control and Automation VCCA-2024, ISBN 978-604-937-357-2.

TÀI LIỆU THAM KHẢO

1. Doroftei, I., V. Grosu, and V. Spinu, 2007, *Omnidirectional mobile robot-design and implementation*, INTECH Open Access Publisher London, UK.
2. Quang, H.D., et al., 2022, Design a nonlinear MPC controller for autonomous mobile robot navigation system based on ROS, *International Journal of Mechanical Engineering and Robotics Research*, 11(6), pp. 379-388.
3. Teslic, L., I. Skrjanc, and G. Klancar, 2011, EKF-Based Localization of a Wheeled Mobile Robot in Structured Environments, *Journal of Intelligent and Robotic Systems*, 62(2), pp. 187-203.
4. Dong, L., et al., 2023, A review of mobile robot motion planning methods: from classical motion planning workflows to reinforcement learning-based architectures, *Journal of Systems Engineering and Electronics*, 34(2), pp. 439-459.
5. Azizi, M.R., A. Rastegarpanah, and R. Stolkin, 2021, Motion planning and control of an omnidirectional mobile robot in dynamic environments, *Robotics*, 10(1), pp. 48.
6. Cao, G., et al., 2022, Fuzzy adaptive PID control method for multi-mecanum-wheeled mobile robot, *Journal of Mechanical Science and Technology*, 36(4), pp. 2019-2029.
7. Thái, N.H., et al., 2020, Điều khiển bám quỹ đạo cho rôbốt tự hành AGV bằng bộ điều khiển PID với tham số điều khiển biến đổi theo thời gian, *Journal of Military Science and Technology*, no. FEE, pp. 21-28.
8. Yadav, P.S., et al., 2022, A robust sliding mode control of mecanum wheelchair for trajectory tracking, *Materials Today: Proceedings*, 56(2), pp. 623-630.
9. L. Yu, S.F. and G. Yang, 2015, A Neural Network Approach for Tracking Control of Uncertain Switched Nonlinear Systems with Unknown Dead-Zone Input, 34(8), pp. 2695–2710.
10. Jiang, M., et al., 2022, Adaptive backstepping control for mecanum-wheeled omnidirectional vehicle using neural networks, *IEEE Transactions on Electrical and Electronic Engineering*, 17(3), pp. 378-386.
11. Liu, Y.-J., et al., 2014, Reinforcement learning design-based adaptive tracking control with less learning parameters for nonlinear discrete-time MIMO systems, *IEEE Transactions on Neural Networks and Learning Systems*, 26(1), pp. 165-176.
12. Ding, L., et al., 2018, Adaptive partial reinforcement learning neural network-based tracking control for wheeled mobile robotic systems, *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 50(7), pp. 2512-2523.
13. Nguyễn, V.T., T.C. Phạm, and M.T. Phạm, 2016, Mô hình hóa và điều khiển robot di động non-holonomic có trượt ngang, *Bản B của Tạp chí Khoa học và Công nghệ Việt Nam*, 58(1).
14. Tinh, N.V., et al., 2016, Modeling and feedback linearization control of a nonholonomic wheeled mobile robot with longitudinal, lateral slips, *2016 IEEE International Conference on Automation Science and Engineering (CASE)*, pp. 996-1001.

15. Van Tinh, N., 2017, Neural network-based adaptive tracking control for a nonholonomic wheeled mobile robot subject to unknown wheel slips, *Journal of Computer Science and Cybernetics*, 33(1), pp. 70-85.
16. Nguyen, T., K. Nguyentien, and P.T. Do T, 2018, Neural network-based adaptive sliding mode control method for tracking of a nonholonomic wheeled mobile robot with unknown wheel slips, model uncertainties, and unknown bounded external disturbances, *Acta Polytechnica Hungarica*, 15(2), pp. 103-23.
17. Nguyena, T., et al., 2019, A Gaussian wavelet network-based robust adaptive tracking controller for a wheeled mobile robot with unknown wheel slips, *International Journal of Control*, 92(11), pp. 2681-2692.
18. Sen, P.T.H., et al., 2019, Thiết kế thuật toán điều khiển cho xe tự hành dựa trên kỹ thuật Backstepping và điều khiển trượt, *Hội nghị khoa học toàn quốc lần thứ nhất về Động lực học và Điều khiển*, pp. 117-120.
19. Thoa, M.T., V.V. Thích, and T.Đ. Trung, 2015, Ứng dụng điều khiển mờ trong bài toán robot bám quỹ đạo, *Vietnam Journal of Science and Technology*, 53(2), pp. 244-244.
20. TIẾP, Đ.K., 2021, Thiết kế bộ điều khiển mờ lai cho robot di động bám quỹ đạo với hai bánh chủ động điều khiển độc lập, *Khoa học công nghệ*, Số 7.
21. Ha, D.K.T., et al., 2021, Fuzzy Adaptive Dynamic Surface Control for Omnidirectional Robot, *Soft Computing for Biomedical Applications and Related Topics*, pp. 257-268.
22. Kim, D.H.T., et al., 2021, Adaptive control for uncertain model of omnidirectional mobile robot based on radial basis function neural network, *International Journal of Control, Automation and Systems*, 19, pp. 1715-1727.
23. Duyên, H.T.K., et al., 2017, Điều khiển bám quỹ đạo đối tượng Robot di động bằng thuật toán điều khiển trượt theo hàm mũ, *Tạp chí Nghiên cứu KH&CN quân sự*, Số đặc san ACMEC.
24. Tuấn, H.C., et al., 2014, Điều khiển robot sử dụng bánh xe omni chạy thẳng hướng, *Hội nghị toàn quốc lần thứ 7 về Cơ điện tử - VCM*.
25. Quốc, L.H. and C.T. Lâm, 2002, *Nhập môn Robot công nghiệp*, Nhà xuất bản Khoa Học và Kỹ thuật.
26. Hiệp, Đ.V., 2003, *Kỹ thuật Robot*, Nhà xuất bản khoa học và Kỹ thuật.
27. Phúc, N.T., 2004, *Robot công nghiệp*, Nhà xuất bản Khoa học và Kỹ thuật.
28. Phước, N.D., 2002, *Lý Thuyết điều khiển nâng cao*, Nhà xuất bản Khoa Học và Kỹ thuật.
29. Phước, N.D., P.X. Minh, and H.T. Trung, 2003, *Lý thuyết điều khiển phi tuyến*, Nhà xuất bản Khoa Học và Kỹ thuật.
30. Minh, P.X. and N.D. Phước, 2006, *Lý thuyết điều khiển Mờ*, Nhà xuất bản Khoa học và Kỹ thuật.
31. Wu, X., et al., 2018, Research on the Design of Educational Robot with Four-wheel Omni-direction Chassis, *Journal of Computers (Taiwan)*, 29(4), pp. 284-294.
32. Myung-Jin, J., et al., 2000, Omnidirectional mobile base OK-II, *International Conference on Robotics and Automation, Symposia Proceedings (Cat. No.00CH37065)*, 4, pp. 3449-3454.

33. Olsen, M.M. and H.G. Petersen, 2001, A new method for estimating parameters of a dynamic robot model, *IEEE Transactions on Robotics and Automation*, 17(1), pp. 95-100.
34. Williams, R.L., et al., 2002, Dynamic model with slip for wheeled omnidirectional robots, *IEEE Transactions on Robotics and Automation*, 18(3), pp. 285-293.
35. Loh, W.K., K.H. Low, and Y.P. Leow, 2003, Mechatronics design and kinematic modelling of a singularityless omni-directional wheeled mobile robot, *Robot ics and Automation, Proceedings. ICRA 03. IEEE International Conference 3*, pp. 3237-3242.
36. Leow, Y.P., K. Low, and W. Loh, 2002, Kinematic Modelling and Analysis of Mobile Robot s with Omni-Directional Wheels, *Seventh International Conference on Control, Automation, Robot ics And Vision (ICARCV'O2)*, 2, pp. 820-825.
37. Qian, J., et al., 2017, The design and development of an omni-directional mobile robot oriented to an intelligent manufacturing system, *Sensors*, 17(9), pp. 2073.
38. Taheri, H., B. Qiao, and N. Ghaeminezhad, 2015, Kinematic model of a four mecanum wheeled mobile robot, *International journal of computer applications*, 113(3), pp. 6-9.
39. Hendzel, Z., 2020, A description of the motion of a mobile robot with Mecanum wheels–kinematics, *Automation, Robotics and Measurement Techniques*, 920, pp. 346-355.
40. Hendzel, Z., 2020, A description of the motion of a mobile robot with mecanum wheels–dynamics, *Automation, Robotics and Measurement Techniques*, 920, pp. 337-345.
41. Tlale, N. and M. de Villiers, 2008, Kinematics and dynamics modelling of a mecanum wheeled mobile platform, *international conference on mechatronics and machine vision in practice*, pp. 657-662.
42. Salih, J., et al., 2006, Designing Omni-Directional Mobile Robot with Mecanum Wheel, *American Journal of Applied Sciences*, vol. 3(5), pp. 1831-1835.
43. Jia, Y., X. Song, and S.S.-D. Xu, 2013, Modeling and motion analysis of four-Mecanum wheel omni-directional mobile platform, *International Automatic Control Conference (CACS)*, pp. 328-333.
44. Qi-Ye, Z., et al., 2015, A TSK fuzzy model and adaptive sliding-mode controller design for four-Mecanum-wheel omni-directional mobile free-bases, *Chinese Automation Congress (CAC)*, pp. 1862-1867.
45. Mišković, D., et al., 2022, Implementation of Robots Integration in Scaled Laboratory Environment for Factory Automation, *Applied Sciences*, 12(3), pp. 1228.
46. Li, Y., et al., 2018, Modeling and kinematics simulation of a Mecanum wheel platform in RecurDyn, *Journal of Robotics*, 2018.
47. Hendzel, Z., 2017, Modelling of dynamics of a wheeled mobile robot with mecanum wheels with the use of Lagrange equations of the second kind, *International Journal of Applied Mechanics and Engineering*, 22(1), pp. 81-99.

48. Abdelrahman, M., et al., 2014, A description of the dynamics of a four wheel mecanum mobile system as a basis for a platform concept for special purpose vehicles for disabled persons, *Ilmenau Scientific Colloquium*, pp. 1-10.
49. Han, K.-L., et al., 2009, Design and control of mobile robot with mecanum wheel, *ICCAS-SICE*, pp. 2932-2937.
50. Hou, L., et al., 2021, Practical model for energy consumption analysis of omnidirectional mobile robot, *Sensors*, 21(5), pp. 1800.
51. Zimmermann, K., I. Zeidis, and M. Abdelrahman, 2014, Dynamics of mechanical systems with mecanum wheels, *Applied Non-Linear Dynamical Systems*, pp. 269-279.
52. Abd Mutalib, M.A. and N.Z. Azlan, 2020, Prototype development of mecanum wheels mobile robot: A review, *Applied Research and Smart Technology (ARSTech)*, 1(2), pp. 71-82.
53. Becker, F., et al., 2014, An approach to the kinematics and dynamics of a four-wheel Mecanum vehicle, *Sci. J. IFToMM Probl. Mech*, 2, pp. 27-37.
54. Lin, L.-C. and H.-Y. Shih, 2013, Modeling and Adaptive Control of an Omni-Mecanum-Wheeled Robot, *Intelligent Control and Automation*, 04(02), pp. 166-179.
55. Sun, Z., et al., 2021, Trajectory-tracking control of Mecanum-wheeled omnidirectional mobile robots using adaptive integral terminal sliding mode, *Computers & Electrical Engineering*, 96, pp. 107500.
56. Sun, Z., et al., 2021, Path-following control of Mecanum-wheels omnidirectional mobile robots using nonsingular terminal sliding mode, *Mechanical Systems and Signal Processing*, 147, pp. 107128.
57. Pizá, R., et al., 2022, Nonuniform dual-rate extended kalman-filter-based sensor fusion for path-following control of a holonomic mobile robot with four mecanum wheels, *Applied Sciences*, 12(7), pp. 3560.
58. Alakshendra, V. and S.S. Chiddarwar, 2017, Adaptive robust control of Mecanum-wheeled mobile robot with uncertainties, *Nonlinear Dynamics*, 87, pp. 2147-2169.
59. Hendzel, Z. and M. Kołodziej, 2021, Robust Tracking Control of Omni-Mecanum Wheeled Robot, *Conference on Automation*, pp. 219-229.
60. Solea, R., A. Filipescu, and U. Nunes, 2009, Sliding-mode control for trajectory-tracking of a wheeled mobile robot in presence of uncertainties, *7th Asian control conference*, pp. 1701-1706.
61. Alakshendra, V. and S.S. Chiddarwar, 2016, A robust adaptive control of mecanum wheel mobile robot: simulation and experimental validation, *International Conference on Intelligent Robots and Systems (IROS)*, pp. 5606-5611.
62. Ovalle, L., et al., 2019, Omnidirectional mobile robot robust tracking: Sliding-mode output-based control approaches, *Control Engineering Practice*, 85, pp. 50-58.
63. Wang, D. and C.B. Low, 2008, Modeling and analysis of skidding and slipping in wheeled mobile robots: Control design perspective, *IEEE Transactions on Robotics*, 24(3), pp. 676-687.
64. Wu, P., et al., 2017, Optimal design for pid controller based on de algorithm in omnidirectional mobile robot, *MATEC Web of Conferences*, 95, pp. 08014.

65. Szeremeta, M. and M. Szuster, 2022, Neural tracking control of a four-wheeled mobile robot with mecanum wheels, *Applied Sciences*, 12(11), pp. 5322.
66. Zhang, T. and X. Zhang, 2023, Distributed Model Predictive Control with Particle Swarm Optimizer for Collision-Free Trajectory Tracking of MWMR Formation, *Actuators*, 12(3), pp. 127.
67. Zhao, T., X. Zou, and S. Dian, 2022, Fixed-time observer-based adaptive fuzzy tracking control for Mecanum-wheel mobile robots with guaranteed transient performance, *Nonlinear Dynamics*, pp. 1-17.
68. Qin, Y., et al., 2021, Trajectory Tracking Control for Mecanum-Wheel Cambered Mobile Robots Based on Online Adaptive Critic Optimal Controller, *40th Chinese Control Conference (CCC)*, pp. 314-319.
69. Wang, D., et al., 2020, A robust model predictive control strategy for trajectory tracking of omni-directional mobile robots, *Journal of Intelligent & Robotic Systems*, 98, pp. 439-453.
70. Malayjerdi, E., H. Kalani, and M. Malayjerdi, 2018, Self-tuning fuzzy pid control of a four-mecanum wheel omni-directional mobile platform, *Electrical Engineering (ICEE), Iranian Conference on*, pp. 816-820.
71. Zijie, N., et al., 2019, Fuzzy control strategy for course correction of omnidirectional mobile robot, *International Journal of Control, Automation and Systems*, 17, pp. 2354-2364.
72. Anzai, Y., 2012, *Pattern Recognition and Machine Learning*, Elsevier.
73. Jordan, M.I. and T.M. Mitchell, 2015, Machine learning: Trends, perspectives, and prospects, *Science*, 349(6245), pp. 255-260.
74. Vamvoudakis, K.G. and F.L. Lewis, 2010, Online actor-critic algorithm to solve the continuous-time infinite horizon optimal control problem, *Automatica*, 46(5), pp. 878-888.
75. Bhasin, S., et al., 2013, A novel actor-critic-identifier architecture for approximate optimal control of uncertain nonlinear systems, *Automatica*, 49(1), pp. 82-92.
76. Modares, H., F.L. Lewis, and M.-B. Naghibi-Sistani, 2014, Integral reinforcement learning and experience replay for adaptive optimal control of partially-unknown constrained-input continuous-time systems, *Automatica*, 50(1), pp. 193-202.
77. Kamalapurkar, R., P. Walters, and W.E. Dixon, 2016, Model-based reinforcement learning for approximate optimal regulation, *Automatica*, 64, pp. 94-104.
78. Zhang, D., G. Wang, and Z. Wu, 2022, Reinforcement learning-based tracking control for a three mecanum wheeled mobile robot, *IEEE Transactions on Neural Networks and Learning Systems*, 35(1), pp. 1445-1452.
79. Andreev, A.S. and O.A. Peregudova, 2020, On global trajectory tracking control for an omnidirectional mobile robot with a displaced center of mass, *Russian Journal of Nonlinear Dynamics*, 16(1), pp. 115-131.
80. Terui, N. and M. Wada, 2023, Variable Center of Mass Control for Omni-Directional Cart with Power-Assist Function, *21st International Conference on Advanced Robotics (ICAR)*, pp. 551-557.

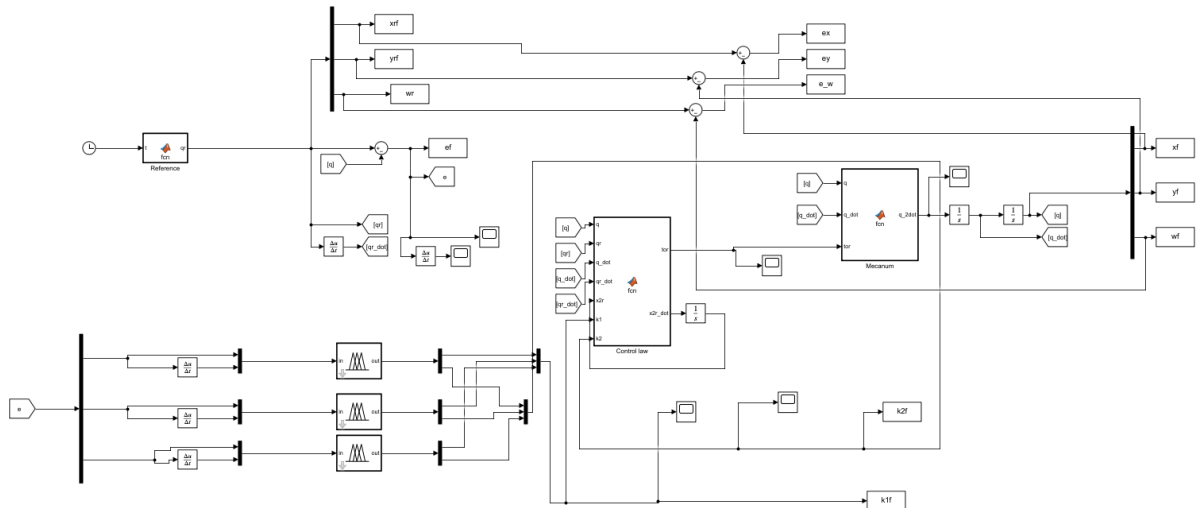
81. Shahin, S., et al., 2017, Simulation, control and construction of a four mecanum-wheeled robot, *4th International Conference on Knowledge-Based Engineering and Innovation (KBEI)*, pp. 0315-0319.
82. Shimada, A., et al., 2005, Mecanum-wheel vehicle systems based on position corrective control, *31st Annual Conference of IEEE Industrial Electronics Society*, pp. 6 pp.
83. Shijin, C. and K. Udayakumar, 2017, Speed control of wheeled mobile robots using PID with dynamic and kinematic modelling, *International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, pp. 1-7.
84. Wen, R. and M. Tong, 2017, Mecanum wheels with Astar algorithm and fuzzy PID algorithm based on genetic algorithm, *International Conference on Robotics and Automation Sciences (ICRAS)*, pp. 114-118.
85. Katpatal, A., A. Parwekar, and A.K. Jha, 2021, Model-Based synchronized control of a robotic dual-arm manipulator, *Advances in Mechanical Engineering: Select Proceedings of ICAME 2020*, pp. 645-654.
86. Dumitrascu, B., A. Filipescu, and V. Minzu, 2011, Backstepping control of wheeled mobile robots, *15th International Conference on System Theory, Control and Computing*, pp. 1-6.
87. Swaroop, D., et al., 2000, Dynamic surface control for a class of nonlinear systems, *IEEE transactions on automatic control*, 45(10), pp. 1893-1899.
88. Tobergte, D.R. and S. Curtis, 2013, *Dynamic Surface Control of Uncertain Nonlinear Systems*, Vol. 53, Springer.
89. Song, B., A. Howell, and K. Hedrick, 2001, Dynamic surface control design for a class of nonlinear systems, *Proceedings of the 40th IEEE Conference on Decision and Control*, 3, pp. 2797-2802.
90. Nguyen, H.T. and M. Sugeno, 2012, *Fuzzy systems: modeling and control*, Vol. 2, Springer Science & Business Media.
91. Kruse, R., et al., 2022, *Multi-layer perceptrons*, in *Computational intelligence: a methodological introduction*, Springer, pp. 53-124.
92. Hornik, K., M. Stinchcombe, and H. White, 1990, Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks, *Neural networks*, 3(5), pp. 551-560.
93. Finlayson, B.A., 2013, *The method of weighted residuals and variational principles*, SIAM.
94. Abu-Khalaf, M. and F.L. Lewis, 2005, Nearly optimal control laws for nonlinear systems with saturating actuators using a neural network HJB approach, *Automatica*, 41(5), pp. 779-791.
95. Ioannou, P. and B. Fidan, 2006, *Adaptive control tutorial*, SIAM.
96. Khalil, H.K., 2002, *Nonlinear Systems*, 3rd Edition, Prentice Hall.
97. Vamvoudakis, K.G., 2011, *Online learning algorithms for differential dynamic games and optimal control*, The University of Texas at Arlington.
98. Modares, H., F.L. Lewis, and M.-B. Naghibi-Sistani, 2013, Adaptive optimal control of unknown constrained-input systems using policy iteration and neural networks, *IEEE Transactions on neural networks and learning systems*, 24(10), pp. 1513-1525.

99. Vamvoudakis, K.G., D. Vrabie, and F.L. Lewis, 2011, Online adaptive learning of optimal control solutions using integral reinforcement learning, *Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*, pp. 250-257.
100. Dupree, K., et al., 2011, Asymptotic optimal control of uncertain nonlinear Euler–Lagrange systems, *Automatica*, 47(1), pp. 99-107.
101. Pham, T.L. and P.N. Dao, 2022, Disturbance observer-based adaptive reinforcement learning for perturbed uncertain surface vessels, *ISA transactions*, 130, pp. 277-292.
102. Kiumarsi, B., et al., 2017, Optimal and autonomous control using reinforcement learning: A survey, *IEEE transactions on neural networks and learning systems*, 29(6), pp. 2042-2062.
103. Wu, H.-N. and B. Luo, 2012, Neural Network Based Online Simultaneous Policy Update Algorithm for Solving the HJI Equation in Nonlinear H_{∞} Control, *IEEE Transactions on Neural Networks and Learning Systems*, 23(12), pp. 1884-1895.
104. Başar, T. and P. Bernhard, 2008, *H-infinity optimal control and related minimax design problems: a dynamic game approach*, Springer Science & Business Media.
105. Kamalapurkar, R., et al., 2015, Approximate optimal trajectory tracking for continuous-time nonlinear systems, *Automatica*, 51, pp. 40-48.
106. Huang, H.-C. and C.-C. Tsai, 2008, Adaptive trajectory tracking and stabilization for omnidirectional mobile robot with dynamic effect and uncertainties, *IFAC Proceedings Volumes*, 41(2), pp. 5383-5388.
107. Pyo, Y., et al., 2017, *ROS Robot Programming (English)*, ROBOTIS Co.,Ltd.
108. Koubaa, A. and J. Kacprzyk, 2017, *Robot Operating System (ROS)*, Springer ed, Vol. 707.
109. Morgan, Q., B. Gerkey, and W.D. Smart, 2015, *Programming Robots with ROS: a practical introduction to the Robot Operating System*, O'Reilly Media, Inc.
110. Joseph, L. and J. Cacace, 2018, *Mastering ROS for Robotics Programming: Design, build, and simulate complex robots using the Robot Operating System*, Packt Publishing Ltd.
111. Do Quang, H., 2022, *Nghiên cứu hệ thống xây dựng bản đồ, lập quỹ đạo và điều khiển bám quỹ đạo cho robot di động bốn bánh đa hướng*, Luận án tiến sĩ, Viện Khoa học và Công nghệ quân sự.

PHỤ LỤC

1. Code mô phỏng thuật toán Fuzzy-DSC

Sơ đồ mô phỏng của thuật toán trên Simulink



Hình PL.1 Sơ đồ mô phỏng thuật toán Fuzzy-DSC

- Code khối Reference:

```
function qr = fcn(t)
```

```
R0=1;
```

```
xr=R0*sin(t);
```

```
yr=R0*sin(2*t);
```

```
thetar=pi/4;
```

```
qr=[xr;yr;thetar];
```

- Code khối Control law:

```
function [tor,x2r_dot]= fcn(q,qr,q_dot,qr_dot,m, Jw, Jb, L1, L2, r,u,  
g,x2r,T,k1,k2)
```

```
k11=k1(1);
```

```
k22=k1(2);
```

```
k33=k1(3);
```

```
b11=k2(1);
```

```
b22=k2(2);
```

```
b33=k2(3);
```



```

theta=q(3);
x_dot=q_dot(1);
y_dot=q_dot(2);
theta_dot=q_dot(3);

%% dynamic matrix for control
M=[m+4*Jb/(r^2) 0 0;0 m+4*Jb/(r^2) 0;0 0 I+(4*Jb*(L1+L2)^2)/(r^2)];
Q=[(m+4*Jb/(r^2))*y_dot*theta_dot;-(m+4*Jb/(r^2))*x_dot*theta_dot;0];
B=(1/r)*[cos(theta) -sin(theta) 0;sin(theta) cos(theta) 0;0 0 L1+L2]*[1 1 1
1;-1 1 -1 1;-1 1 1 -1];
shi=(r*u*m*g/4)*sign(transpose(B)*q_dot);

%% Control
k1=[k11 0 0;0 k22 0;0 0 k33];
k2=[b11 0 0;0 b22 0;0 0 b33];
s1=q-qr;
x2ngang=qr_dot-k1*s1;
x2r_dot=(x2ngang-x2r)/T;
s2=q_dot-x2r;
tor=-(B.')*inv(B*(B.'))*(M*(-x2r_dot+k2*s2)-Q-B*shi);

```

- Code khối Model_FMWR:

```

function q_2dot = fcn(q,q_dot,tor,mb,mw, J, Jw, k, h, r ,g ,d1 , d2, mu1)
theta=q(3);
x_dot=q_dot(1);
y_dot=q_dot(2);
theta_dot=q_dot(3);
l=sqrt(k^2+h^2);
m1=mb+4*(mw+J/r^2);
alpha=(tan(h/k))^(-1);
m2=mb*(d1^2+d2^2)+Jw+8*(mw+J/r^2)*l^2*sin(pi/4-alpha);
m3=mb*(d1*sin(theta)+d2*cos(theta));
m4=mb*(-d1*cos(theta)+d2*sin(theta));

```

$$M=[m1 \ 0 \ m3; 0 \ m1 \ m4; m3 \ m4 \ m2];$$

$$c1=mb*\theta_{dot}*(d1*\cos(\theta)-d2*\sin(\theta));$$

$$c2=mb*\theta_{dot}*(d1*\sin(\theta)+d2*\cos(\theta));$$

$$Q=[0 \ 0 \ c1; 0 \ 0 \ c2; 0 \ 0 \ 0];$$

$$B = 1/r*[-\cos(\theta)+\sin(\theta) \ -\cos(\theta)-\sin(\theta) \ \cos(\theta)-\sin(\theta) \\ \cos(\theta)+\sin(\theta);$$

$$\begin{aligned} &-\cos(\theta)-\sin(\theta) \ \cos(\theta)-\sin(\theta) \ \cos(\theta)+\sin(\theta) - \\ &\cos(\theta)+\sin(\theta); \end{aligned}$$

$$\begin{aligned} &-\sqrt{2}*l*\sin(\pi/4-\pi/4) \ -\sqrt{2}*l*\sin(\pi/4+\pi/4) \ -\sqrt{2}*l*\sin(\pi/4- \\ &3*\pi/4) \ -\sqrt{2}*l*\sin(\pi/4+3*\pi/4)]; \end{aligned}$$

$$\delta=(r*\mu1*g/4)*\text{sign}(\text{transpose}(B)*q_{dot});$$

$$q_{2dot}=M\backslash(B*tor-B*\delta-Q*q_{dot});$$

- **Tham số của mô hình robot và bộ điều khiển:**

%% Dynamic parameters

mb=30;

mw=0.9;

Jb=5;

Jw=0.1;

k=0.2;

h=0.3;

l=0.25;

r=0.075;

mu1=0.01;

mu2=0.05;

g=9.8;

d1=0.0;

d2=0.0;

%% Control parameters

u=0.01;

T=0.02;

```

c1=5;
c2=15;
%% Simulation parameters
T_sample=0.01;
T_end=15;

```

2. Code mô phỏng thuật toán học tăng cường trên Matlab:

```

%% Simulation RL for Mecanum Wheeled Mobile Robot
clc;clear;close all;
%% Time interval and simulation time
Step = 0.001;T_end = 50;
t = 0:Step:T_end;

```

- Tham số của FMWR và tham số bộ điều khiển:

```

mb = 30+delta_m; % Mass (kg)
mw = 0.9;
a = 0.2; % Length of MWMR (m)
b = 0.3;
l=sqrt(a^2+b^2);
r = 0.075; % Wheeled radius (m)
I = 5; Iw = 0.1; % Rotational inertia (kg.m^2)
Ib = I + 4*(Iw+mw*(a^2+b^2));
g = 9.8;
Q = eye(6); % Positive-definite matrix
R = eye(4); % Positive-definite symmetric constant matrix
n_NN = 51; % Number of NN nodes
% The control gain with
eta_c = 6;
eta_a1 = 3;
eta_a2 = 0.001;
eta_a = 0.001;
beta = 0.001;

```

```

gamma = 0.0001;
v = 0.002;
%% Variables
q_r = cell(1,size(t,2));
q_r_dot = cell(1,size(t,2));
q = cell(1,size(t,2));
q_dot = cell(1,size(t,2));
    phi = cell(1,size(t,3));
phi_dot = cell(1,size(t,3));
upsilon = cell(1,size(t,2)); % upsilon = v_q
upsilon_dot = cell(1,size(t,2));
upsilon_r = cell(1,size(t,2));
upsilon_r_dot = cell(1,size(t,2));
x = cell(1,size(t,2));
x_dot = cell(1,size(t,2));
x_r = cell(1,size(t,2));
x_r_dot = cell(1,size(t,2));
e = cell(1,size(t,2));
z = cell(1,size(t,2));
W_c = cell(1,size(t,2));
W_a = cell(1,size(t,2));
gamma_upper = cell(1,size(t,2));
V_z = cell(1,size(t,2));
grad_V_z = cell(1,size(t,2));
tau = cell(1,size(t,2));
%% Initial Conditions
tau{1} = [0;0;0;0];
q{1} = [1;1;1.3];
upsilon{1} = [0;0;0]; %v_q
W_c{1} = 100*ones(n_NN,1);
W_a{1} = 6*ones(n_NN,1);
gamma_upper{1} = 10*eye(51);

```

- Mô hình toán học của FMWR:

%% Inertia matrix

$$mb = 12 + \delta_m;$$

$$\alpha = 1/\tan(b/a);$$

$$m_{11} = mb + 4*(mw + I/r^2);$$

$$m_{33} = mb*(d1^2 + d2^2) + I_w + 8*(mw + I/r^2)*l^2*\sin(\pi/4 - \alpha)^2;$$

$$m_{13} = mb*(d1*\sin(q_i(3)) + d2*\cos(q_i(3)));$$

$$m_{31} = m_{13};$$

$$m_{23} = mb*(-d1*\cos(q_i(3)) + d2*\sin(q_i(3)));$$

$$m_{32} = m_{23};$$

$$M = [m_{11} \ 0 \ m_{13}; 0 \ m_{11} \ m_{23}; m_{31} \ m_{32} \ m_{33}];$$

$$m_{qr13} = mb*(d1*\sin(\cos(0.5*t(i))) + d2*\cos(\cos(0.5*t(i))));$$

$$m_{qr23} = mb*(-d1*\cos(\cos(0.5*t(i))) + d2*\sin(\cos(0.5*t(i))));$$

$$M_{qr} = [m_{11} \ 0 \ m_{qr13}; 0 \ m_{11} \ m_{qr23}; m_{qr13} \ m_{qr23} \ m_{33}];$$

$$c_{13} = mb*\epsilon_i(3)*(d1*\cos(q_i(3)) - d2*\sin(q_i(3)));$$

$$c_{23} = mb*\epsilon_i(3)*(d1*\sin(q_i(3)) + d2*\cos(q_i(3)));$$

$$C = [0 \ 0 \ c_{13}; 0 \ 0 \ c_{23}; 0 \ 0 \ 0];$$

$$c_{qr13} = mb*-0.5*\sin(0.5*t(i))*(d1*\cos(\cos(0.5*t(i))) - d2*\sin(\cos(0.5*t(i))));$$

$$c_{qr23} = mb*-$$

$$0.5*\sin(0.5*t(i))*(d1*\sin(\cos(0.5*t(i))) + d2*\cos(\cos(0.5*t(i))));$$

$$C_{qr} = [0 \ 0 \ c_{qr13}; 0 \ 0 \ c_{qr23}; 0 \ 0 \ 0];$$

%% H matrix

$$R_m = [\cos(q_i(3)) \ \sin(q_i(3)) \ 0; -\sin(q_i(3)) \ \cos(q_i(3)) \ 0; 0 \ 0 \ 1];$$

$$dR_m = [-\sin(q_i(3)) \ \cos(q_i(3)) \ 0; -\cos(q_i(3)) \ -\sin(q_i(3)) \ 0; 0 \ 0 \ 0];$$

$$\theta_{dot} = \epsilon_i(3);$$

$$B = 1/r*[-\cos(q_i(3)) + \sin(q_i(3)) \ -\cos(q_i(3)) - \sin(q_i(3)) \ \cos(q_i(3)) - \sin(q_i(3)) \ \cos(q_i(3)) + \sin(q_i(3))];$$

$$-\cos(q_i(3)) - \sin(q_i(3)) \ \cos(q_i(3)) - \sin(q_i(3))$$

$$\cos(q_i(3)) + \sin(q_i(3)) \ -\cos(q_i(3)) + \sin(q_i(3));$$

$$-\sqrt{2}*l*\sin(\pi/4 - \pi/4) \ -\sqrt{2}*l*\sin(\pi/4 + \pi/4) \ -\sqrt{2}*l*\sin(\pi/4 - 3*\pi/4) \ -\sqrt{2}*l*\sin(\pi/4 + 3*\pi/4)];$$

```

B_qr = 1/r*[-cos(cos(0.5*t(i)))+sin(cos(0.5*t(i))) -cos(cos(0.5*t(i)))-
sin(cos(0.5*t(i))) cos(cos(0.5*t(i)))-sin(cos(0.5*t(i)))
cos(cos(0.5*t(i)))+sin(cos(0.5*t(i)))];
-cos(cos(0.5*t(i)))-sin(cos(0.5*t(i))) cos(cos(0.5*t(i)))-sin(cos(0.5*t(i)))
cos(cos(0.5*t(i)))+sin(cos(0.5*t(i))) -cos(cos(0.5*t(i)))+sin(cos(0.5*t(i)));
-sqrt(2)*l*sin(pi/4-pi/4) -sqrt(2)*l*sin(pi/4+pi/4) -sqrt(2)*l*sin(pi/4-
3*pi/4) -sqrt(2)*l*sin(pi/4+3*pi/4)];
H = -(1/r)*[1 1 -1 -1;1 -1 -1 1;sqrt(2)*l*sin(pi/4-pi/4)
sqrt(2)*l*sin(pi/4+pi/4) sqrt(2)*l*sin(pi/4-3*pi/4)
sqrt(2)*l*sin(pi/4+3*pi/4)]';
J = 1/sqrt(2)*[1 1 sqrt(2)*l*sin(pi/4-pi/4);1 -1 sqrt(2)*l*sin(pi/4+pi/4);-1
-1 sqrt(2)*l*sin(pi/4-3*pi/4);-1 1 sqrt(2)*l*sin(pi/4+3*pi/4)]*Rm;
J_cong = pinv(J'*J)*J';
Sf = sign((-sqrt(2)/2)*1/r*pinv(J_cong)*upsilon{i})*0.05*r;
% Sf=sign(H*Rm*upsilon{i})*0.05;
%% q Model(2.12)
% f_q
f_q = theta_dot*dRm*pinv(Rm)-pinv(M)*C;
% g_q
g_q = Rm*pinv(M)*B;
% fd(q,vq)
f_d_q_vq = -Rm*pinv(M)*B*Sf;
% d(q,v_q)
% d_q_vq = g_q\f_d_q_vq;
%% x Model(2.2.3)
% upsilon = v_q
x{i} = [q{i}' upsilon{i}']';
f_x = [upsilon{i}' (f_q*upsilon{i})']';
g_x = [zeros(4,3) (g_q)']';
%% z Model
%G_z = [g_x' zeros(6,3)']';

```

```

G_z = [g_x;zeros(6,4)];
%% Reference model
Rm_qr = [cos(cos(0.5*t(i))) sin(cos(0.5*t(i))) 0;-sin(cos(0.5*t(i)))
cos(cos(0.5*t(i))) 0;0 0 1];
dRm_qr = [-sin(cos(0.5*t(i))) cos(cos(0.5*t(i))) 0;-cos(cos(0.5*t(i))) -
sin(cos(0.5*t(i))) 0;0 0 0];
theta_qr_dot = -0.5*sin(0.5*t(i));
g_q_r = Rm_qr*pinv(M)*B_qr;
g_x_r = [zeros(4,3) (g_q_r)']';
%g_cong = pinv(g_x_r);
g_cong = (pinv(g_x_r'*g_x_r))*(g_x_r');

```

- **Quỹ đạo mô phỏng:**

```

% Quy dao duong tron
q_r{i} = [sin(t(i)) cos(t(i)) cos(0.5*t(i))];
q_r_dot{i} = [cos(t(i)) -sin(t(i)) -0.5*sin(0.5*t(i))];
x_r{i} = [q_r{i}' q_r_dot{i}']';
x_r_dot{i} = [cos(t(i)) -sin(t(i)) -0.5*sin(0.5*t(i)) -sin(t(i)) -cos(t(i)) -
0.25*cos(0.5*t(i))];
f_q_r = theta_qr_dot*dRm_qr*pinv(Rm_qr)-pinv(M_qr)*C_qr;
f_x_r = [q_r_dot{i}' (f_q_r*q_r_dot{i})']';

```

- **Thuật toán học tăng cường:**

```

%% Caculate F_z
% tau_r(2.15)
tau_r = g_cong*(x_r_dot{i}-f_x_r);
% F_z(2.16)
F_z = [f_x-x_r_dot{i}+g_x*tau_r; x_r_dot{i}];
%% Tracking error
e{i} = x{i} - x_r{i};
z{i} = [e{i}' x_r{i}']';
%% Activation function
% phi_z = zeros(51,1);
phi_z = (1/2)*[z{i}(1)^2 z{i}(2)^2 z{i}(3)^2 ...

```

$z_{i\{1\}}*z_{i\{4\}} z_{i\{1\}}*z_{i\{5\}} z_{i\{1\}}*z_{i\{6\}} z_{i\{2\}}*z_{i\{4\}} z_{i\{2\}}*z_{i\{5\}}$
 $z_{i\{2\}}*z_{i\{6\}} \dots$

$z_{i\{3\}}*z_{i\{4\}} z_{i\{3\}}*z_{i\{5\}} z_{i\{3\}}*z_{i\{6\}} (z_{i\{1\}}^2)*(z_{i\{2\}}^2)$
 $(z_{i\{1\}}^2)*(z_{i\{3\}}^2) \dots$

$(z_{i\{2\}}^2)*(z_{i\{3\}}^2) (z_{i\{1\}}^2)*(z_{i\{7\}}^2) (z_{i\{1\}}^2)*(z_{i\{8\}}^2)$
 $(z_{i\{1\}}^2)*(z_{i\{9\}}^2) \dots$

$(z_{i\{1\}}^2)*(z_{i\{10\}}^2) (z_{i\{1\}}^2)*(z_{i\{11\}}^2) (z_{i\{1\}}^2)*(z_{i\{12\}}^2)$
 $(z_{i\{2\}}^2)*(z_{i\{7\}}^2) \dots$

$(z_{i\{2\}}^2)*(z_{i\{8\}}^2) (z_{i\{2\}}^2)*(z_{i\{9\}}^2) (z_{i\{2\}}^2)*(z_{i\{10\}}^2)$
 $(z_{i\{2\}}^2)*(z_{i\{11\}}^2) \dots$

$(z_{i\{2\}}^2)*(z_{i\{12\}}^2) (z_{i\{3\}}^2)*(z_{i\{7\}}^2) (z_{i\{3\}}^2)*(z_{i\{8\}}^2)$
 $(z_{i\{3\}}^2)*(z_{i\{9\}}^2) \dots$

$(z_{i\{3\}}^2)*(z_{i\{10\}}^2) (z_{i\{3\}}^2)*(z_{i\{11\}}^2) (z_{i\{3\}}^2)*(z_{i\{12\}}^2)$
 $(z_{i\{4\}}^2)*(z_{i\{7\}}^2) \dots$

$(z_{i\{4\}}^2)*(z_{i\{8\}}^2) (z_{i\{4\}}^2)*(z_{i\{9\}}^2) (z_{i\{4\}}^2)*(z_{i\{10\}}^2)$
 $(z_{i\{4\}}^2)*(z_{i\{11\}}^2) \dots$

$(z_{i\{4\}}^2)*(z_{i\{12\}}^2) (z_{i\{5\}}^2)*(z_{i\{7\}}^2) (z_{i\{5\}}^2)*(z_{i\{8\}}^2)$
 $(z_{i\{5\}}^2)*(z_{i\{9\}}^2) \dots$

$(z_{i\{5\}}^2)*(z_{i\{10\}}^2) (z_{i\{5\}}^2)*(z_{i\{11\}}^2) (z_{i\{5\}}^2)*(z_{i\{12\}}^2)$
 $(z_{i\{6\}}^2)*(z_{i\{7\}}^2) \dots$

$(z_{i\{6\}}^2)*(z_{i\{8\}}^2) (z_{i\{6\}}^2)*(z_{i\{9\}}^2) (z_{i\{6\}}^2)*(z_{i\{10\}}^2)$
 $(z_{i\{6\}}^2)*(z_{i\{11\}}^2) \dots$

$(z_{i\{6\}}^2)*(z_{i\{12\}}^2)]'$;

`% grad_phi_z = zeros(51,12);`

`grad_phi_z = (1/2)*[2*z_{i\{1\}} 0 0 0 0 0 0 0 0 0 0; ...`

`0 2*z_{i\{2\}} 0 0 0 0 0 0 0 0 0; ...`

`0 0 2*z_{i\{3\}} 0 0 0 0 0 0 0 0; ...`

`z_{i\{4\}} 0 0 z_{i\{1\}} 0 0 0 0 0 0 0; ...`

`z_{i\{5\}} 0 0 0 z_{i\{1\}} 0 0 0 0 0 0; ...`

`z_{i\{6\}} 0 0 0 0 z_{i\{1\}} 0 0 0 0 0; ...`

`0 z_{i\{4\}} 0 z_{i\{2\}} 0 0 0 0 0 0 0; ...`

`0 z_{i\{5\}} 0 0 z_{i\{2\}} 0 0 0 0 0 0; ...`


```

0 z{i}(6) 0 0 0 z{i}(2) 0 0 0 0 0 0; ...
0 0 z{i}(4) z{i}(3) 0 0 0 0 0 0 0; ...
0 0 z{i}(5) 0 z{i}(3) 0 0 0 0 0 0; ...
0 0 z{i}(6) 0 0 z{i}(3) 0 0 0 0 0 0; ...
2*z{i}(1)*(z{i}(2)^2) 2*(z{i}(1)^2)*z{i}(2) 0 0 0 0 0 0 0 0; ...
2*z{i}(1)*(z{i}(3)^2) 0 2*(z{i}(1)^2)*z{i}(3) 0 0 0 0 0 0 0 0; ...
0 2*z{i}(2)*(z{i}(3)^2) 2*(z{i}(2)^2)*z{i}(3) 0 0 0 0 0 0 0 0; ...
2*z{i}(1)*(z{i}(7)^2) 0 0 0 0 0 2*(z{i}(1)^2)*z{i}(7) 0 0 0 0; ...
2*z{i}(1)*(z{i}(8)^2) 0 0 0 0 0 0 2*(z{i}(1)^2)*z{i}(8) 0 0 0 0; ...
2*z{i}(1)*(z{i}(9)^2) 0 0 0 0 0 0 2*(z{i}(1)^2)*z{i}(9) 0 0 0; ...
0 0 0 0 0 2*z{i}(6)*(z{i}(10)^2) 0 0 0 2*(z{i}(6)^2)*z{i}(10) 0 0; ...
0 0 0 0 0 2*z{i}(6)*(z{i}(11)^2) 0 0 0 0 2*(z{i}(6)^2)*z{i}(11) 0; ...
0 0 0 0 0 2*z{i}(6)*(z{i}(12)^2) 0 0 0 0 0 2*(z{i}(6)^2)*z{i}(12)];
%% Estimated controlling performance(3.11)
V_z{i} = W_c{i}'*phi_z;
grad_V_z{i} = grad_phi_z'*W_c{i};
%% Controller(3.10)
u = -1/2*(pinv(R))*G_z'*grad_phi_z'*W_a{i} + noise;
tau{i+1} = u + g_cong*(x_r_dot{i} - f_x_r);
%% Update stage(2.12)
F_q = epsilon{i};
F_epsilon = f_q*epsilon{i} + g_q*tau{i} + f_d_q_vq;
if i==size(t,2)
    break;
end
epsilon{i+1} = epsilon{i} + Step*F_epsilon;
q{i+1} = q{i} + Step*F_q;
• Cập nhật trọng số:
% W_c
sigma = grad_phi_z*(F_z+G_z*u)-gamma*phi_z;
W_c{i+1} = W_c{i} + Step*(-
eta_c*gamma_upper{i}*sigma/(1+v*sigma'*gamma_upper{i}*sigma)*delt);

```

```

gamma_upper{i+1} = gamma_upper{i} - Step*eta_c*(-
beta*gamma_upper{i}+gamma_upper{i}*(sigma*sigma')/(1+v*sigma'*gamma
_upper{i}*sigma)*gamma_upper{i});
W_a{i+1} = W_a{i} + Step*(-eta_a1*(W_a{i} - W_c{i})-
eta_a2*W_a{i}+eta_a/4*D_1*W_a{i}*sigma_ngang'/((1+v*sigma'*gamma
_upper{i}*sigma)^(1/2))*W_c{i});
end

```

- **Kết quả mô phỏng:**

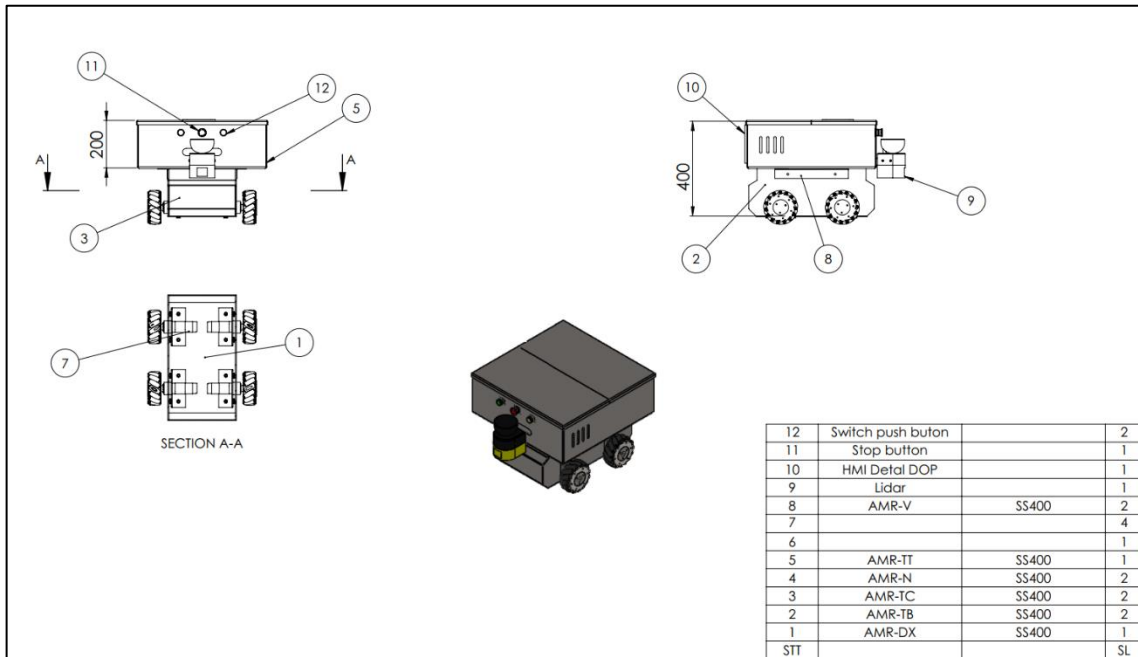
```

q = cell2mat(q);
q_r = cell2mat(q_r);
W_a = cell2mat(W_a);
W_c = cell2mat(W_c);
%% Plot Real Trajectory vs Reference Trajectory
figure;
plot(q_r(1,:),q_r(2,:), 'r', 'linewidth', 0.5);
hold on
plot(q(1,:),q(2,:), 'b', 'linewidth', 0.5);
title('Trajectory');
legend('Real Trajectory', 'Reference Trajectory');
ylabel('y-axis (m)')
xlabel('x-axis (m)')
hold off
%% Plot error trajectory
figure;
plot(t, q(1,:) - q_r(1,:));
hold on
plot(t, q(2,:) - q_r(2,:));
hold on
plot(t, q(3,:) - q_r(3,:));
title('Error signal');
legend('X error', 'Y error', 'Head angle error');
ylabel('Tracking Error (m)')

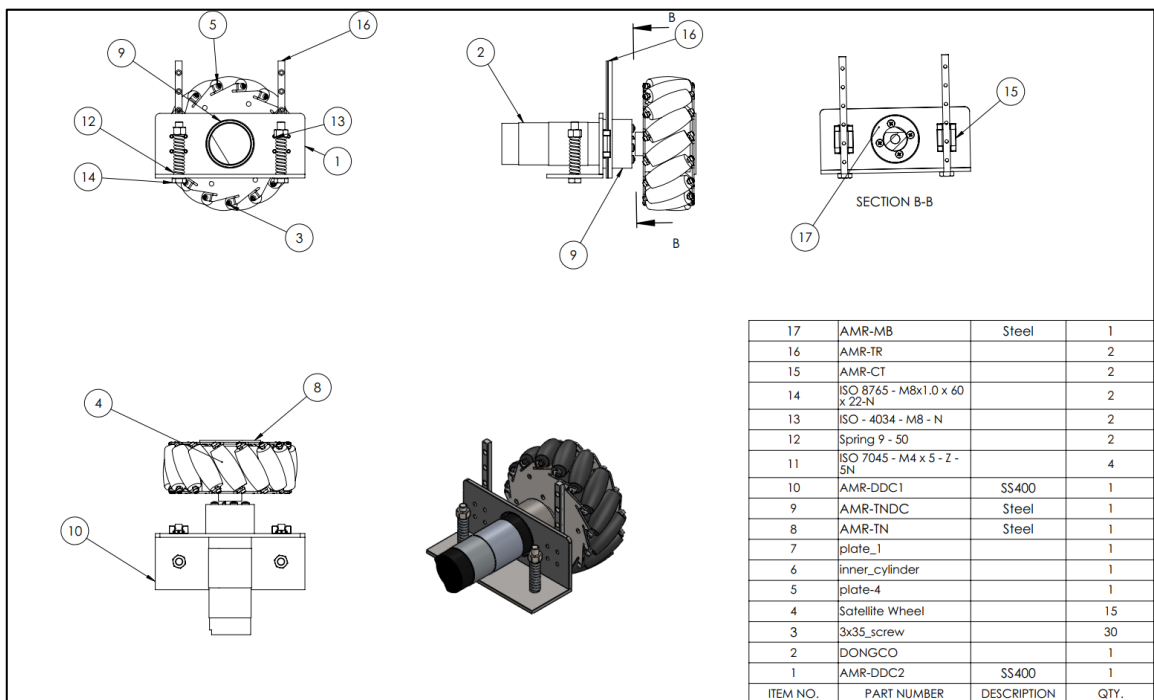
```

3. Thiết kế, chế tạo mô hình FMWR

a, Bản vẽ thiết kế cơ khí



Hình PL.2 Bản vẽ thiết kế mô hình FMWR



Hình PL.3 Bản vẽ chi tiết cơ cấu truyền động bánh xe

b, Thiết kế mạch điều khiển:

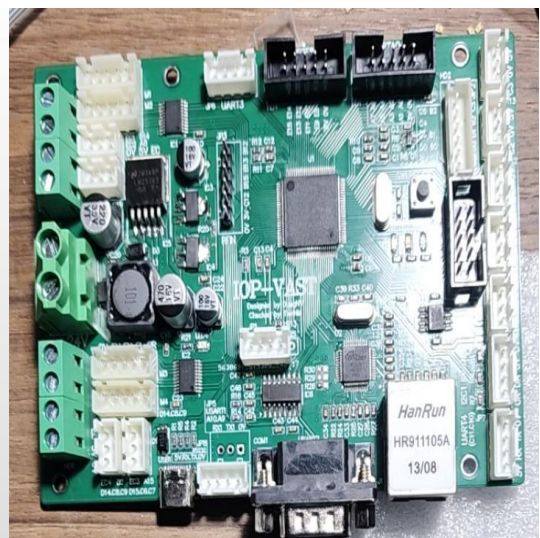
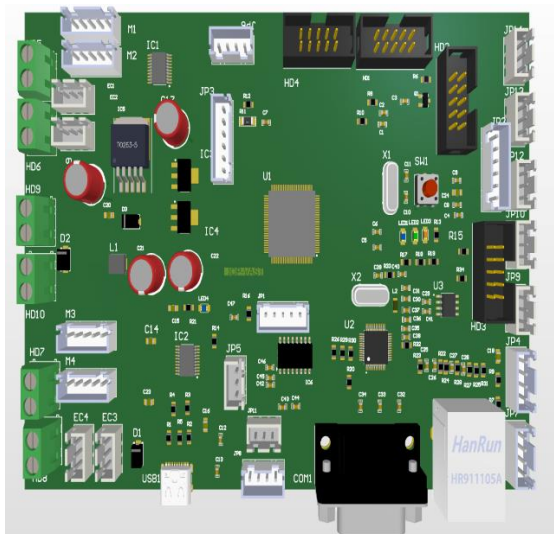
- Bộ xử lý trung tâm Jetson_TX2



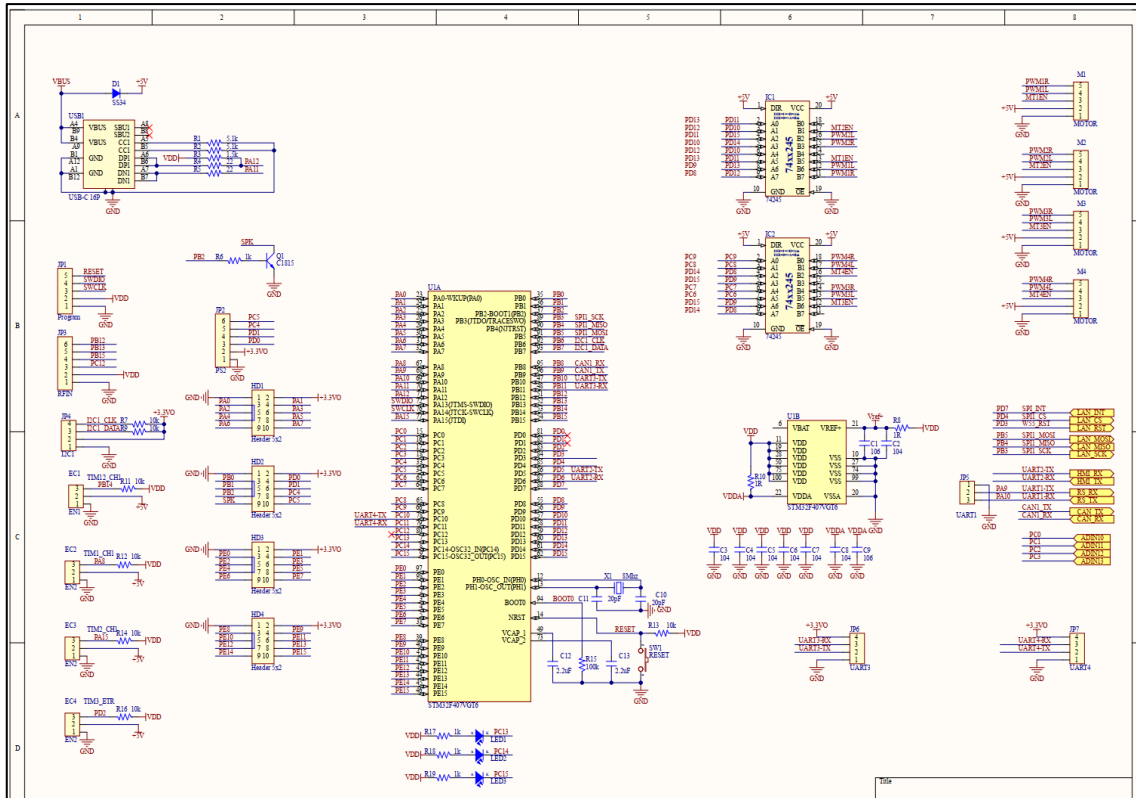
Hình PL.4 Jetson TX2 developer Kit

Cấu hình của Jetson_TX2:

- +GPU: 256-core Pascal™ GPU
 - + CPU: Dual-core Denver 2 64-bit CPU
 - + Quad-core ARM Cortex-A57 MPCore
 - + RAM: 8GB
 - + Lưu trữ: 32GB
- Sơ đồ mạch điều khiển STM32



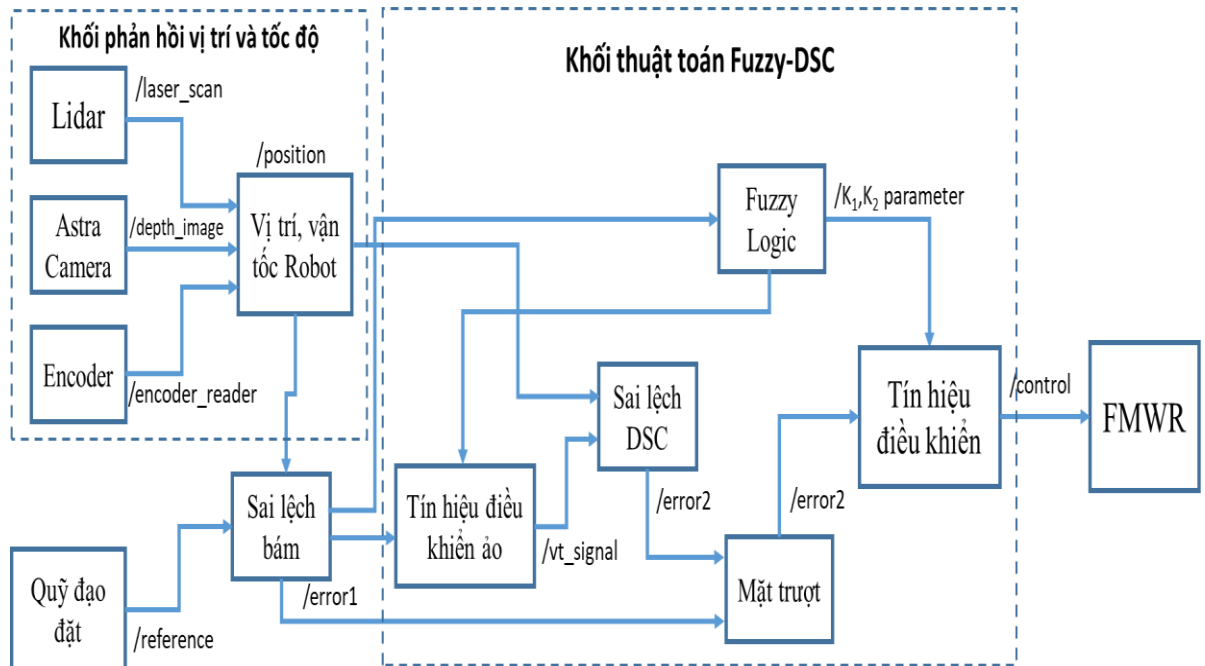
Hình PL.5 Sơ đồ mạch in của mạch STM32



Hình PL.6 Sơ đồ nguyên lý mạch điều khiển STM32

4. Thực nghiệm thuật toán Fuzzy-DSC

a. Sơ đồ khối lập trình phần mềm



Hình PL.7 Sơ đồ khối lập trình thực nghiệm thuật toán Fuzzy-DSC

b. Code lập trình điều khiển trên ROS

- Node lấy tín hiệu từ Lidar

```

<launch>PL23

<arg name="serial_port_lidar" default = "/dev/ttyUSB1"/>

<node name="rplidarNode" pkg="rplidar_ros" type="rplidarNode"
output="screen">

<param name="serial_port" type="string" value="$(arg
serial_port_lidar)"/>

<param name="serial_baudrate" type="int" value="115200"/><!--
A1/A2 -->

<!--param name="serial_baudrate" type="int" value="256000"--><!--
A3 -->

<param name="frame_id" type="string" value="laser"/>

<param name="inverted" type="bool" value="false"/>

<param name="angle_compensate" type="bool" value="true"/>

</node>
</launch>

```

- Node tính toán vị trí robot

```

<launch>

<node pkg="rf2o_laser_odometry" type="rf2o_laser_odometry_node"
name="rf2o_laser_odometry" output="screen">

<param name="laser_scan_topic" value="/scan"/> # topic where the
lidar scans are being published

<param name="odom_topic" value="/odom" /> # topic where tu
publish the odometry estimations

<param name="publish_tf" value="true" /> # wheter or not to
publish the tf::transform (base->odom)

<param name="base_frame_id" value="/base_footprint"/> #
frame_id (tf) of the mobile robot base. A tf transform from the laser_frame to
thebase_frame is mandatory

```

```

<param name="odom_frame_id" value="/odom" /> # frame_id
(tf) to publish the odometry estimations
<param name="init_pose_from_topic" value="" /> # (Odom topic) Leave
empty to start at point (0,0)
<param name="freq" value="5.0"/> # Execution frequency.
<param name="verbose" value="false" /> # verbose
</node>
</launch>

```

- Node tính toán bộ điều khiển

```

<?xml version="1.0" encoding="UTF-8"?>
<launch>
  <rosparam command="load"
    file="$(findros_robotics)/config/vast_controller.yaml" />
  <!-- load the controllers -->
  <node name="controller_spawner" pkg="controller_manager"
    type="spawner" respawn="false"
    output="screen" ns="FMWR_robot" args="joint1_position_controller
    joint2_position_controller joint3_position_controller
    joint4_position_controller" />
  <!-- convert joint states to TF transforms for rviz, etc -->
  <node name="robot_state_publisher" pkg="robot_state_publisher"
    type="robot_state_publisher" respawn="false" output="screen">
  <remap from="/joint_states" to="/vast_robot/joint_states" />
  </node>
</launch>

<?xml version="1.0"?>
<launch>
  <!-- Declare the arguments -->
  <!--arg name="node_start_delay" default="2.0" /-->
  <arg name="serial_UART" default = "/dev/ttyUSB0"/>

```

```

<arg name="serial_lidar" default = "/dev/ttyUSB1"/>

<include
  file="$(findros_robotics)/launch/practical/practical_robot.launch"> </inclu>

<include file="$(find my_serial_node)/launch/serial_port.launch">

<arg name="serial_port_UART" value="$(arg serial_UART)"/>

</include>

<include file="$(find rplidar_ros)/launch/rplidar.launch">

<arg name="serial_port_lidar" value="$(arg serial_lidar)"/>

</include>

<include file="$(find timed_roslaunch)/launch/timed_roslaunch.launch">

<arg name="time" value="2" />

<arg name="pkg" value="rf2o_laser_odometry" />

<arg name="file" value="rd2o_laser_odometry.launch" />

<arg name="node_name" value="timed_roslaunch" />

<!-- This is an optional argument -->

</include>

<!--include
  file="$(find rf2o_laser_odometry)/launch/rf2o_laser_odometry.launch"> </inc
  lude>

<arg name="node_start_delay" default="2.0" /-->

<include file="$(find gmapping)/launch/slam.launch"> </include>

<node pkg="move_base" type="move_base"
  respawn="false" name="move_base" output="screen">

<roscparam
  file="$(findros_robotics)/practical_config/costmap_common_params.yaml"
command="load" ns="global_costmap" />

<roscparam
  file="$(findros_robotics)/practical_config/costmap_common_params.yaml"
command="load" ns="local_costmap" />

<roscparam file="$(findros_robotics)/practical_config/local_costmap_params.y
aml" command="load" />

```



```

<roscpp
  file="$(findros_robotics)/practical_config/global_costmap_params.yaml"
  command="load" />

<roscpp
  file="$(findros_robotics)/practical_config/teb_local_planner_omni_params.y
  aml"

  command="load" />

  <param name="base_global_planner"
  value="global_planner/GlobalPlanner" />
  <param name="planner_frequency" value="1.0" />
  <param name="planner_patience" value="5.0" />
  <param name="base_local_planner"
  value="teb_local_planner/TebLocalPlannerROS" />
  <param name="controller_frequency" value="5.0" />
  <param name="controller_patience" value="15.0" />
</node>

<!-- Show in Rviz
<node name="rviz" pkg="rviz" type="rviz" args="-d $(find
ros_robotics)/rviz/vast01.rviz" required="true" output="screen"/> -->
</launch>

```

- **Code lập trình trên mạch STM32:**

```

/* Includes -----*/

#include "main.h"

/* Private includes -----*/

/* USER CODE BEGIN Includes */

#include "stdio.h"

#include "stdlib.h"

#include "string.h"

```

```

#include "math.h"

/* USER CODE END Includes */

/* USER CODE BEGIN PV */

// UART variable

uint8_t rx_data;

char rx_index;

char rx_buffer[20];

char temp[100];

int check_var = 0;

//encoder variable

int pulse_counter1 = 0, pulse_counter2 = 0, pulse_counter3 = 0,
pulse_counter4 = 0;

//velocity variable

double k = sqrt(2)/2;

float vx = 0, vy = 0, w_angle = 0;

float w_reference1_temp = 0;

float w_reference2_temp = 0;

float w_reference3_temp = 0;

float w_reference4_temp = 0;

float w_reference1 = 0, w_reference2 = 0, w_reference3 = 0, w_reference4 =
0;

float w1 = 0, w2 = 0, w3 = 0, w4 = 0;

float e1 = 0, e2 = 0, e3 = 0, e4 = 0;

float pre_e1 = 0, pre_e2 = 0, pre_e3 = 0, pre_e4 = 0;

```

```

float const max_pwm = 300;

//robot variable

float d_robot = 0.25; //robot radius

float r_wheel = 0.07;

/* Private function -----*/

void SystemClock_Config(void);

static void MX_GPIO_Init(void);

static void MX_TIM3_Init(void);

static void MX_TIM2_Init(void);

static void MX_USART3_UART_Init(void);

static void MX_TIM9_Init(void);

static void MX_TIM12_Init(void);

static void MX_TIM5_Init(void);

/* USER CODE BEGIN PFP */

void check_w_ref()

{

    sprintf(temp, "vx_ref = %.2f\n", vx);

    HAL_UART_Transmit(&huart3, temp, strlen(temp), 10);

    HAL_Delay(100);

    sprintf(temp, "vy_ref = %.2f\n", vy);

    HAL_UART_Transmit(&huart3, temp, strlen(temp), 10);

    HAL_Delay(100);

```

```

    sprintf(temp, "w_ref = %.2f\n", w_angle);

    HAL_UART_Transmit(&huart3, temp, strlen(temp), 10);

    HAL_Delay(100);

    sprintf(temp, "w1_temp = %.2f, w1_ref = %.2f\n", w_reference1_temp
,w_reference1);

    HAL_UART_Transmit(&huart3, temp, strlen(temp), 10);

    HAL_Delay(100);

    sprintf(temp, "w2_temp = %.2f, w2_ref = %.2f\n", w_reference2_temp
,w_reference2);

    HAL_UART_Transmit(&huart3, temp, strlen(temp), 10);

    HAL_Delay(100);

    sprintf(temp, "w3_temp = %.2f, w3_ref = %.2f\n", w_reference3_temp
,w_reference3);

    HAL_UART_Transmit(&huart3, temp, strlen(temp), 10);

    HAL_Delay(100);

    sprintf(temp, "w4_temp = %.2f, w4_ref = %.2f\n", w_reference4_temp
,w_reference4);

    HAL_UART_Transmit(&huart3, temp, strlen(temp), 10);

    HAL_Delay(100);

    HAL_Delay(2000);
}

void signal_feedback_w_wheel()
{

    sprintf(temp, "w1 = %.2f rpm r\n", w1);

    HAL_UART_Transmit(&huart3, temp, strlen(temp), 10);

```

```

    HAL_Delay(100);

    sprintf(temp, "w2 = %.2f rpm r\n", w2);

    HAL_UART_Transmit(&huart3, temp, strlen(temp), 10);

    HAL_Delay(100);

    sprintf(temp, "w3 = %.2f rpm r\n", w3);

    HAL_UART_Transmit(&huart3, temp, strlen(temp), 10);

    HAL_Delay(100);

    sprintf(temp, "w4 = %.2f rpm r\n", w4);

    HAL_UART_Transmit(&huart3, temp, strlen(temp), 10);

    HAL_Delay(2000);

}

int main(void)

{

    HAL_Init();

    SystemClock_Config();

    MX_GPIO_Init();

    MX_TIM3_Init();

    MX_TIM2_Init();

    MX_USART3_UART_Init();

    MX_TIM9_Init();

    MX_TIM12_Init();

    MX_TIM5_Init();

    HAL_TIM_Base_Start_IT(&htim2);

```

```

HAL_UART_Receive_IT(&huart3, &rx_data, 1);

Init_Motor1();

Init_Motor2();

Init_Motor3();

Init_Motor4();

/* USER CODE END 3 */

}

HAL_GPIO_WritePin(GPIOE,
M01_EN_R_Pin|M01_EN_L_Pin|M02_EN_L_Pin|M02_EN_R_Pin
|M02_3_3V_Pin, GPIO_PIN_SET);

HAL_GPIO_WritePin(M01_3_3V_GPIO_Port, M01_3_3V_Pin,
GPIO_PIN_SET);

HAL_GPIO_WritePin(M01_GND_GPIO_Port, M01_GND_Pin,
GPIO_PIN_RESET);

HAL_GPIO_WritePin(M02_GND_GPIO_Port, M02_GND_Pin,
GPIO_PIN_RESET);

HAL_GPIO_WritePin(GPIOD,
M03_EN_R_Pin|M03_EN_L_Pin|M03_3_3V_Pin, GPIO_PIN_SET);

HAL_GPIO_WritePin(M03_GND_GPIO_Port, M03_GND_Pin,
GPIO_PIN_RESET);

HAL_GPIO_WritePin(M04_GND_GPIO_Port, M04_GND_Pin,
GPIO_PIN_RESET);

HAL_GPIO_WritePin(GPIOG, M04_3_3V_Pin|M04_EN_L_Pin,
GPIO_PIN_SET);

HAL_GPIO_WritePin(M04_EN_R_GPIO_Port, M04_EN_R_Pin,
GPIO_PIN_SET);

/*****

```