N° d'ordre : .....

République Algérienne Démocratique et populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université YAHIA FARES DE MEDEA



Faculté de Technologie
Département de Génie Electrique

THESE

Présentée en vue d'obtenir le grade de

DOCTEUR

EN AUTOMATIQUE

Spécialité : AUTOMATIQUE

Présentée par

EULDJI Rafik

Master En AUTOMATIQUE de l'Université Du Médéa.

Thème

Techniques de contrôle supervisé pour véhicules autonomes

Supervisory control techniques for unmanned ground vehicle

Soutenue publiquement le : 27/02/2023

Devant le jury composé de :

Mr : TLEMÇANI AbdElhalim	Professeur à l'université de MEDEA	Président
Mr : BATEL Noureddine	Professeur à l'université de MEDEA	Directeur de thèse
Mr : HOUASSINE Hamza	Professeur à l'université de BOUIRA	Examinateur
Mr : KARA Réda	Maître de conférences à l'université de MEDEA	Examinateur

2022/2023

Thèse préparée au sein du Laboratoire des Systèmes Électroniques Avancés (LSEA)

de l'université de Médéa, ALGÉRIE

Acknowledgments

First and foremost, I thank Allah (SWT) for having given me the strength and the courage to accomplish this modest work. And for letting me live to see this thesis.

I'm deeply grateful to my supervisor**, Pr. BATEL Noureddine** for his coaching, availability, and sense of listening and exchange. And for providing guidance and feedback throughout this project.

Great thanks should be address to **Dr. SKENDER Mohamed redha and Dr REBHI redha** for their encouragement, and the influence that they give.

I would also like to extend my thanks to the members of the jury for having done us the honor and the pleasure of accepting and judging our work.

Also extend my deepest gratitude to all of our teachers for the training they gave us during our university studies.

In the end, I thank all the people who contributed, directly or indirectly, to the production of this thesis.

May the Almighty Allah richly bless all of you.

Dedication

I dedicate my dissertation work to my parents, whose words of encouragement and push for tenacity ring in my ears. And to whom I wish them a long and happy life, may Allah protect them.

To my Brother and sister whose encourage me to continue.

To All my family, friends, and all whose wish me only the best.

Table of Content

الملخص

تقنيات التحكم الخاضعة للإشراف للمركبات ذاتية القيادة

الهدف من هذه الأطروحة هو تصميم تقنيات تحكم قوية للتعامل مع مشكلة تتبع المسار للروبوت المتحرك ذي عجلات ،من أجل تحقيق هذا الهدف ، تم اقتراح ثلاثة حلول جديدة ، يتم فحص هذه الحلول في بيئة الماتلاب والتحقق من صحتها. بشكل تجريبي ، استنادًا إلى مسارات محددة مثل مسارات التدرج المستمر (مسار رباعي ، مسار دائري و على شكل مالانهاية) ، أو مسار متدرج غير مستمر مثل المسار المستطيل.

تتمثل الطريقة الأولى في تصميم تقنية هجينة جديدة تدمج بين وحدة التحكم التقليدية المتمثلة في المصحح التناسبي التكاملى التفاضلي (PID) والاستدلال العصبي الضبابي التكيفي (ANFIS) ، يطلق عليها باختصار باسم ANFIS-PID ، قد تم تحسين هذه التقنية باستخدام العديد من الخوارزميات المستوحاة من الطبيعة ، حيث تم تصميم وحدتي تحكم ANFIS-PID من أجل التعامل مع السرعة وزاوية الدوران للروبوت المتحرك ، هذه التقنية كانت للتعامل مع ديناميكيات الروبوت المتحرك فقط ،لذلك تم إدخال وحدة تحكم تدعى خطوة إلى الوراء (backstepping) في الحلقة الخارجية كوحدة تحكم حركية.

استندت الطريقة الثانية إلى دمج تقنية المصحح التناسبي التكاملى التفاضلي الجزئي (FOPID)، مع تقنية الخطوة الى الوراء ، حيث نجد وحدتي تحكم FOPID في الحلقة الداخلية للتعامل مع المشكلة الديناميكية ، باستخدام تقنيات مستوحاة من الطبيعة هجينة (meta-heuristic) ، يتم إنشاء FOPID الأول من أجل التحكم في السرعة .بينما وحدة التحكم FOPID الثانية للتعامل مع التحكم في زاوية الاتجاه للسيارة الكهربائية. التحسين الذي قدمه النهج المقترح هو أنه تم تحقيق تقدم ملحوظ من حيث تقليل أخطاء تتبع المسار وتقليل إجراءات التحكم ، وتحديداً في مسارات التدرج المستمر.

تتمثل منهجية التحكم الثالثة المقترحة في دمج وحدة المصحح التناسبي التكاملى التفاضلي الجزئي (FOPID) مع وحدة تحكم الاستدلال العصبي الضبابي التكيفي ANFIS ، يطلق عليها وحدة التحكم ANFIS-FOPID ، الغرض من هذه التقنية هو التقليل من خطأ تتبع المسار للمسارات غير المستمرة. تم تحسين وحدة التحكم بناءً على نفس الخوارزميات المستخدمة في الحل السابق ، وهنا أيضًا نجد وحدتي تحكم ANFIS-FOPID على المستوى الديناميكي وذلك للتقليل في نسبة الخطأ الخاصة بكل من الاتجاه و السرعة . مثل المنهجيات السابقة يتم هنا أيضا إضافة التقنية المسماة بالخطوة إلى الوراء للتحكم في الحالة الحركية للربوت. أظهرت النتائج التي تم الحصول عليها باستخدام ANFIS-FOPID تطورًا ملحوظا مقارنة بالنتائج التي تم الوصول إليها بواسطة FOPIDو ANFIS-PID.

أخيرًا ، تم إنشاء منصة روبوت متنقلة لفحص الأداء العام للمنهجية المقترحة في الوقت الفعلي. تتكون بنية الروبوت من محركين يعملان بالتيار المستمر مع حساس يعمل بطريقة التشفير التربيعي بالإضافة إلى عجلتين من اجل الحفاظ على التوازن ، نجد أيضا بطاقة الأردوينو ميغا لقراءة البيانات الآتية من محيط الروبوت. يعمل الأردوينو كمتحكم تابع لجهاز كمبيوتر ، حيث يتواصل مع بعضهم البعض بناءً على بروتوكول اتصال تسلسلي. يتم تقديم أربع دراسات تطبيقية لتأكيد صحة وحدات التحكم المقترحة.

الكلمات المفتاحية: تقنية الخطوة إلى الوراء، المصحح التناسبي التكاملي التفاضلي الجزئي، الاستدلال العصبي الضبابي ألتكيفي،خوارزميات مستوحاة من الطبيعة،تقنيات هجينة، الروبوت المتنقل ذو العجلات،تتبع المسار

Abstract

Supervisory control techniques for unmanned ground vehicle

The aim of this thesis is to design new control techniques to deal with the trajectory tracking problem of the wheeled mobile robot (WMR).in order to achieve the goal, Three new solutions are proposed, these approached are examined in MATLAB environment and validate experimentally, based on specific paths such as the continuous gradient trajectories (quadrifolium path, circular path and the lemniscates), or a non-continuous gradient path like the rectangular trajectory.

The first method is to design a new hybrid technique between the standard PID controller and the adaptive neuro-fuzzy inference (ANFIS), abbreviated as ANFIS-PID, this technique was optimized using numerous nature inspired algorithms ,here two ANFIS-PID controllers were designed in order to deal with speed and the orientation control for the mobile robot, but this technique was for dealing with dynamics of mobile robot only, there for a backstepping controller has been introduced in the external loop as kinematic controller.

The second methods was based on a novel combination using the fractional order proportional integral derivative (FOPID) control strategy, with the nonlinear backstepping approach ,here. Two optimized FOPID controllers are needed in the internal loop to handle the dynamic problem, using new hybrid meta-heuristic techniques, the first FOPID is created in order to control the velocity of WMR. were The second FOPID controller is to deal with the orientation angle control of the electrical vehicle. The influence of the suggested approach is that a noticeable progress has been achieved in terms of minimizing trajectory-tracking errors and reducing control actions, specifically in continuous gradient trajectories.

Third proposed control methodology is the merge of fractional order PID controller with ANFIS controller abbreviated as ANFIS-FOPID controller. The purpose is to minimize the trajectory tracking error of non-continuous trajectories. The controller has been optimized based on the same algorithms used in previous solution, here also we have two blocks of ANFIS-FOPID controllers in the dynamic level to reduce the orientation and the speed error, and like the other methodology here we have added the backstepping controller to take care the kinematic control action. The obtained

results by using ANFIS-FOPID have shown a serious development in reducing errors of path tracking and increasing the control efforts over the results by FOPID, and ANFIS-PID.
Finally, a mobile robot platform has been constructed to examine the overall performance of the proposed methodology in real-time. The architecture of the WMR based on the deferential drive mechanism consists of two DC motor with quadratic encoders and two passive wheels .An Arduino mega to read the sensory data. The Arduino acts as a slave microcontroller, connected with a master PC, communicate with each other based on a serial communication protocol. Four case studies are offered to confirm the validity of the proposed controllers
keywords: Backstepping, FOPID, ANFIS, meta-heuristics, hybrid techniques, UGV, trajectory tracking.WMR.

Résumé

Techniques de contrôle supervisé pour véhicules autonomes

L'objectif de cette thèse est de concevoir nouvelles techniques de commande pour traiter le problème de suivi de trajectoire du robot mobile à roues (WMR). Afin d'atteindre l'objectif, trois nouvelles solutions sont proposées. Elles sont élaborées dans l'environnement MATLAB puis validées expérimentalement en utilisant des trajectoires spécifiques, telles que les trajectoires à gradient continu comme la trajectoire quadrifolium ou trajectoire circulaire ou lemniscate, ainsi que les trajectoires à gradient non continu comme la trajectoire rectangulaire.
La première méthode consiste à concevoir une nouvelle technique hybride, entre le contrôleur PID standard et l'inférence neuro-floue adaptative (ANFIS), en abrégé ANFIS-PID. Cette technique est optimisée à l'aide de nombreux algorithmes inspirés de la nature. Ainsi, deux contrôleurs ANFIS-PID sont conçus pour gérer de façon dynamique, la vitesse et l'orientation du robot mobile. Un contrôleur de backstepping est introduit dans la boucle externe, en tant que contrôleur cinématique.
La deuxième méthode est basée sur une nouvelle combinaison de la stratégie de contrôle par la dérivée intégrale proportionnelle d'ordre fractionnaire (FOPID) avec l'approche non linéaire du backstepping. Afin de gérer le problème dynamique, deux contrôleurs FOPID sont nécessaires dans la boucle interne et sont optimisés par de nouvelles techniques méta-heuristiques hybrides. Le premier FOPID contrôle la vitesse du WMR, tandis que le deuxième FOPID contrôle l'angle d'orientation du véhicule mobile. Un progrès notable est réalisé avec l'approche suggérée, en termes

de minimisation des erreurs de suivi de trajectoire et réduction des actions de contrôle, en particulier dans les trajectoires à gradient continu.

La troisième méthodologie de contrôle suggérée est l'association du contrôleur PID d'ordre fractionnaire avec le contrôleur ANFIS, abrégé en contrôleur ANFIS-FOPID. Le but est de minimiser l'erreur de suivi des trajectoires à gradient non continu. Ce type de contrôleur est optimisé par les mêmes algorithmes utilisés dans la solution précédente. De même, deux blocs de contrôleurs ANFIS-FOPID sont utilisés au niveau dynamique afin de réduire l'orientation et l'erreur de vitesse, et sont associés au contrôleur backstepping qui prend soin de l'action cinématique. Les résultats obtenus ont montré une réduction poussée des erreurs de suivi de chemin avec augmentation des possibilités de contrôle, comparativement aux résultats obtenus par FOPID ou ANFIS-PID.

Une plate-forme robotique mobile est construite afin de mesurer en temps réel, les performances globales de la méthodologie proposée. L'architecture du WMR basée sur le mécanisme d'entraînement différentiel, se compose de deux moteurs à courant continu avec encodeurs quadratiques, de deux roues motrices, et d'un Arduino méga pour lire les données captées.

Mots-clés: Backstepping, FOPID, ANFIS, Meta-heuristiques, techniques hydride, robot mobile autonome, suivi de trajectoire.

Table of Content

Table of content.....i

List of Abbreviations.....xiv

List of Nomenclature and symbols.....xv

List of Figures.....xvi

List of Tables.....xxii

List of Publication.....xxiii

General Introduction

1. Background & Motivation.....1

2. Objectives and contributions of the Research.....2

3. Achievement.....3

5. Thesis organisation.....3

Chapter I: Background and Literature Review

I.1 Introduction.....4

I.2. Background.....4

I.2.1. Autonomous mobile robot classification.....5

I.2.2. Early history of mobile robot.....6

I.2.3. Current day mobile robots.....9

I.3. Holomic and nonholomic proprieties.....12

I.3.1. Holomic wheeled mobile robot.....12

I.3.2. Nonholomic wheeled mobile robot.....14

I.4.wheeled mobile robot control architecture.....17

I.5.Control Problems of the Mobile Robot.....17

I.6.Tracking Error of the Mobile Robot.....18

I.7.Previous Works……..……..……..……19

I.8. Proposed solutions.....21

I.9.conclusion.....21

Chapter II: Mathematical model and classical control approach

II.1. Introduction.....22

Table of Content

II.2. Modeling the Wheeled Mobile Robot.....22
II.2.1. Kinematic Model of the WMR.....24
II.2.2. Dynamical model.....26
II.2.3. Actuator modeling.....30
II.3. The Backstepping controller design using the nonlinear feedback method.....32
II.3.1. Kinematic Controller Design.....33
II.3.2. Dynamic Controller.....33
II.4. Simulation Results.....35
II.4.1. Square Path.....35
II.4.2. circular Path.....36
II.5. Conclusion.....38

Chapter III: Optimal PID and ANFIS-PID controller design

III.1.Introduction.....39
III.2.The general control Strategy Design.....39
III.2.1. Kinematic Controller Design..... .40
III.2.2.The PID Dynamic Controller.....42
III.2.3. Tuning methodology for PID..... 43
III.3. Meta-heuristics..... 44
III.3.1.Particle Swarm Optimizer (PSO).....44
III.3.2.Dragonfly algorithm (DA).....45
III.3.3.Ant Lion Optimizer (ALO).....46
III.3.4.Artificial Bee Colony (ABC).....49
III.3.5.Grey Wolf Optimizer (GWO).....50
III.3.6.Grasshoppers Optimization Algorithm (GOA).....50
III.3.7.Moth flame optimization (MFO) algorithm.....52
III.4.The ANFIS Controller description.....53
III.5. Designing the ANFIS-PID Dynamic Controller.....55
III.6. Results and discussion.....58
III.6.1. Convergence Curve of all Algorithms.....59
III.6.2. Step Response Performance of Speed Controller.....60
III.6.3. Step Response Performance. of Angle Controller.....61
III.6.4. Circuler path (case study 01).....63
III.6.5. Diamond chape trajectory.....64
III.6.6. Zig zag trajectory.....65

III.6.7. Adding the backsteping controller in the external loop.....66
III.6.7.1 Case of lemniscates.....66
III.6.7.2 Case of square trajectory.....67
III.7. Conclusion.....68

Chapter IV: Optimal Back stepping-fractional order PID controller design

IV.1.Introduction.....70
IV.2.Review on Fractional Calculus.....70
IV.2.1.Riemann-Liouville Définition (RL).....71
IV.2.3.Grunwald-Letnikov Definition (GL).....72
IV.2.4.A Modified Approximate Realization Method.....73
IV.2.5.Laplace Transform of Fractional Order transfer function 74
IV.3. Controller Design and Strategy.....79
IV.3.1 Kinematic Controller Design.....79
IV.3.2.Dynamic Controller Description 80
IV.3.3. tuning methodology.....81
IV.4. Meta-Heuristic Algorithms.....82
IV.4.1.Hybrid PSO-GWO Algorithm.....82
IV.4.2.Whale Optimization Algorithm.....83
IV.4.3.Hybrid grey wolf whale optimizer (HWGO).....83
IV.5.Results and Discussion 84
IV.5.1.Convergence Curve of the Algorithms.....84
IV.5.2.Step Response Performance of Speed Controller.....85
IV.5.3 Step Response Performance of Angle Controller.....87
IV.5.4. Path Tracking.....88
IV.6. Conclusions.....91

Chapter V: Optimal ANFIS-FOPID controller design

V.1.Introduction.....92
V.2. Controller Design and Strategy.....92
V.2.1 Kinematic Controller Design.....93
V.2.2. Designing the ANFIS-FOPID Dynamic Controller.....93
V.3. Results and Discussion.....96
V.3.1. Convergence Curve of the Algorithms.....96
V.3.2. Step Response Performance of Speed Controller 97
V.3.3. Step Response Performance of Angle Controller 98

V.3.4. The trajectory coresponding to Step Response in XY plane.....100
V.3.5. Circuler path (case study 01).....101
V.3.6. Five (5) chape trajetory (case study 02) 102
V.3.7. Zig zag trajectory (case study 03).....104
V.3.8. Adding the backsteping controller in the external loop.....105
V.3.8.1.Case of quadrifolium.....105
V.3.8.2. Case of square trajectory.....107
V.4. Conclusion.....108

Chapter VI: Experimental Work

VI.1.Introduction.....110
VI.2.The Global Control Architecture Description.....110
VI.3 Hardware Design.....111
VI.3.1 The self designed robot platform.....111
VI.3.2 The electronics circuit design.....113
VI.4 Software Design..... 114
VI.4.1 The complete Simulink model.....115
VI.4.2 FOPID controller implementation.....116
VI.4.3 Speed Control.....116
VI.4.4 Motor speed calculation from encoder 117
VI.4.5 Pose estimation.....119
VI.5 Experiments Results.....119
VI.5.1 Tracking Circular Trajectory Case Study 1 120
VI.5.2 Tracking Lemniscates Trajectory Case Study 2.....123
VI.5.3 Tracking quadrifolium Trajectory Case Study 3.....126
VI.5.4 tracking rectangular Trajectory Case Study 4 129
VI.6. Conclusion.....132

General Conclusion

1.Conclusions 133
2. Directions for Future Work…… ……..…..134
Bibliography.....135
Annex.....

List of Abbreviations

Item No.	Abbreviation	Description
1	UGV	Unmanned Ground Vehicle(s)
2	WMR	Wheeled mobile robot
3	DDMR	Différentiel drive mobile robot
4	DoF	Degree(s) of Freedom
5	PID	Proportional- Integral-Derivative Controller.
6	FOPID	Fractional Order Proportional- Integral-Derivative Controller.
7	PSO	Particle Swarm Optimization
8	GA	Genetic Algorithm
9	ISE	Integral Square Error
10	GWO	Grey wolf optimizer
11	WOA	Whale Optimization Algorithm
12	GOA	grasshopper algorithm
13	MFO	moth flame optimizer
14	DA	dragonfly optimizer
15	ABC	Artificial bee colony
16	ALO	Ant Lion optimizer
17	HPSOGWO	Hybrid Particle Swarm Grey wolf optimizer
18	HWGO	Hybrid Whale Grey wolf optimizer
19	ITAE	Integral Time Absolute Error
20	RL	Riemann-Liouville
21	GL	Grunwald-Letnikov
22	SISO	Single-Input Single-Output
23	NN	Neural Network(s)
24	ICC	Instantaneous Centre of Curvature
25	2D	Two dimensions
26	DC	Direct Current
27	RPM	Revolution per minutes
28	MIMO	Multi-Input Multi-Output
29	MSE	Mean Square Error
30	FIS	Fuzzy Inference System(s)
31	ANFIS	Adaptive Neuro-Fuzzy Inference System
32	RMSE	Root Mean Square Error
33	USB	Universal Serial Bus
34	PWM	Pulse Width Modulation
35	GPIO	General Purpose Input/output

Nomenclature and Symbol

Symbol	Designation	Unity
L	*Distance between right and left wheels*	*M*
vl,	*Linear velocity of left wheel*	*m/s*
vr	*Linear velocity of right wheel*	*m/s*
φR	right spinning angles of wheels	*RPM*
φL	left spinning angles of wheels	*RPM*
τ_L	The torque of left wheel of the mobile robot.	*N.m*
τ_R	The torque of right wheel of the mobile robot.	*N.m*
(xo,yo)	*Starting point coordinates*	*M*
(xg,yg)	*Target point coordinates*	*M*
(xa,ya)	*Actual coordinates of vehicle*	*M*
A(q)	The matrix associated with the constraints.	-
D	Distance of point Pc from point Po,	*M*
R	Driving wheels radius	*M*
m_c	Mass of the mobile robot without the driving wheels and DC motors,	$Kg.m^2$
m_w	Mass of each driving wheel (with actuator)	*Kg*
I_c	Moment of inertia of the mobile robot about the vertical axis through the center of mass,	*Kg.m2*
I_w	Moment of inertia of each driving wheel with a motor about the Wheel axis,	*Kg.m2*
I_m	Moment of inertia of each driving wheel with a motor about the Wheel diameter,	*Kg.m2*
R_a	Armature winding resistance,	Ω
L_a	Armature winding inductance,	H
K_t	Torque constant.	N·m/A
K_b	Back emf constant.	N·m/A
N	Gear ratio.	-
q	Vector of generalized position coordinates	-
$\dot{q}$	Vector of longitudinal and angular velocities of generalized coordinates.	-

$M(q)$	Symmetric positive definite inertia matrix of the system,	-
C	Centripetal and Coriolis forces matrix,	-
$F(\dot{q})$	Surface friction matrix,	-
$G(q)$	Gravitational vector,	-
$B(q)$	Input transformation matrix,	-
T	Input transformation matrix and input vector.	-
Iter	Current number of iterations	-
Itermax	Maximum number of iterations	-
Iter	Current number of iterations	-
Itermax	Maximum number of iterations	-
Iter	Current number of iterations	-
Itermax	Maximum number of iterations	-
ωm	Angular speed of motor	rad/s
ia	Motor current	A
Ea	Applied voltage to motor	V
vb	Back electromotive force (e.m.f) voltage	V
θ	*Orientation of UGV*	*rad*
ωr	*Angular velocity of right wheel*	*rad/s*
ωl	*Angular velocity of left wheel*	*rad/s*
ω	*Angular velocity of vehicle*	*rad/s*
v	*Velocity of the vehicle*	*m/s*
a	*Distance between the centre of mass and driving wheels' axis in x-direction*	*M*
r	*Radius of vehicle's wheel*	*m*
m	*Mass of vehicle with driving wheels and DC motors*	*kg*
q	*Vector of generalized position coordinates*	-
t_r	*rise time*	*s*
Mp%	*peak percentage, overshoot*	-
Peak	*Maximum overshoot value*	-
t_p	*peak, time*	*s*
t_s	*settling time*	*s*

λ	*The fractional order of integral*	-
μ	*The fractional order of derivative*	-
Kp	*Proportional gain*	-
Ki	*Integral gain*	-
Kd	*Derivative gain*	-
T	*The sampling time*	*ms*
F(s)	*The Laplace transform of f (t)*	-
Γ	*Euler's gamma function*	-

List of Figures

Figure	Figures Name	Page
	Chapter I	
Fig. I.1	Picture of auto assembly plant-spot welding robot of KUKA and a parallel robot Delta of SIG Demaurex SA (invented at EPFL) during packaging of chocolates.	4
Fig. I.2	Different kinds of mobile robot locomotion.	6
Fig. I.3	Analog robots. (a) Tesla's robot (b) flying bomb. Currently hanging in the Imperial War Museum, London (c) Walter's robot	7
Fig. I.4	The first autonomous ground vehicles (a) SRI's Shakey (b) CMU "Rover" (c) Stanford "Cart"	8
Fig. I.5	(a): Khepera (b) Roomba	10
Fig. I.6	Wheeled mobile robots,(a) the Pioneer Robot (b) the Turtlebot (c) Segway	11
Fig. I.7	Walking robots, (a) AIBO (b) Honda ASIMO (c) NAO	11
Fig. I.8	autonomous driving cars,(a) The Google self-driving car (b) The VisLab autonomous vehicle (c) Boss, the autonomous vehicle that won the 2007 DARPA Urban Challenge	12
Fig. I.9	holomic mobile robots (a) URANUS (b) 3WD Compact Omni-Directional	13
Fig. I.10	mecanum wheel with rollers mounted around the wheel rim, each roller axis is at 45 degree from the main wheel plane and at 45 degree to the line parallel with the wheel's axis.	14
Fig. I.11	Differential drive kinematics. Most differential drive robots use castor wheels for stability. (b) the khepera Appears with the kind permission with the kind permission of A. Herzog	15
Fig. I.12	Synchronous drive mechanism. (a) the kinematic of the robot (b). The ARK-1 robot is built around the Cybermotion Navmaster platform. Which utilizes a synchronous drive.	15
Fig. I.13	Tricycle mobile robot. Copyright Mecos Robotics AG. Used with permission.	16
Fig. I.14	Ackerman mechanism (a) robot kinematics . (b) The Oto Melara. Appears with kind permission of OTOBREDA	16
Fig. I.15	General control scheme for UGV.	17

Chapter II

Fig. II.1	Schematic design of the WMR.	22
Fig. II.2	Pure Rolling Motion Constraint.	25
Fig. II.3	Equivalent electrical schema of the dc motor.	31
Fig. II.4	General Control scheme of the WMR.	33
Fig. II.5	Square path in XY plane	35
Fig. II.6	Error corresponding to (a) x,(b) y; (c) the theta angle ,(d) the trajectory tracking	35
Fig. II.7	**C**ircular path in XY plane	36
Fig. II.8	. Error corresponding to (a) x,(b) y; (c) the theta angle ,(d) the trajectory tracking	36

Chapter III

Fig. III.1	General Control scheme of the WMR.	39
Fig. III.2	Trajectory tracking errors	40
Fig. III.3	Block diagram of the PID controller.	42
Fig. III.4	Bloc diagram for tuning methodology.	44
Fig. III.5	The ANFIS architecture with two input parameters x and y.	54
Fig. III.6	The ANFIS in training phase	56
Fig. III.7	a) data configuration for first ANFIS-PID controller,(b) structure of the ANFIS-PID controller (c) data configuration for second ANFIS-PID controller ,(d) error minimization corresponding to the second controller.	56
Fig. III.8	Adjusting the parameters for the designed ANFIS-PID controller using GWO.	57
Fig. III.9	Bloc diagram of the final ANFIS-PID dynamic controller	58
Fig. III.10	Convergence curve of the seven algorithms compared to GWO-ANFIS-PID.	59
Fig. III.11	Step response Comparison for all velocity controllers	60
Fig. III.12	Comparison of step Response for the orientation.	62
Fig. III.13	(a) Circle trajectory in XY plane, (b) error of the distance, (c) linear velocity response for all controllers (d). Angle response of all controllers.	63
Fig. III.14	(a) Diamond shape trajectory in XY plane,(b) linear velocity corresponding to the to controllers, (c) angle response corresponding to the path for the two controllers,(d) error of the distance.	64
Fig. III.15	(a) zigzag path in XY plane ,(b) the tracking error of the controllers (c) the theta angle response corresponding to the controllers ,(d) linear velocity	65

	corresponding to the controllers.	
Fig. III.16	lemniscate path.	66
Fig. III.17	Error corresponding to (a) x,(b) y; (c) the theta angle ,(d) the trajectory tracking.	66
Fig. III.18	**.**Square trajectory	67
Fig. III.19	Error corresponding to (a) x,(b) y; (c) the theta angle ,(d) the trajectory tracking.	68
	Chapter IV	
Fig. IV.1	Control scheme of WMR.	79
Fig. IV.2	Block diagram of fractional order.	80
Fig. IV.3	Generalized FOPID controller.	81
Fig. IV.4	Bloc diagram for parameters tuning.	81
Fig. IV.5	**C**onvergence curve of five algorithms	85
Fig. IV.6	Comparison of velocity step response for all algorithms.	86
Fig. IV.7	Comparison of step Response for the orientation.	87
Fig. IV.8	Star trajectory.	89
Fig. IV.9	Response of the (a) X; (b) Y; (c) θ variable.	89
Fig. IV.10	Response of (a) linear; and (b) angular Velocities (for star shape).	90
Fig. IV.11	**.** Error corresponding to (a) x, (b) y and (c) theta variables; (d) the trajectory tracking.	90
	Chapter V	
Fig. V.1	Control scheme of WMR.	92
Fig. V.2	The ANFIS in training phase.	93
Fig. V.3	data configuration for first ANFIS-FOPID controller,(b) structure of the ANFIS-FOPID controller (c) data configuration for second ANFIS-FOPID controller ,(d) error minimization corresponding to the second controller.	94
Fig. V.4	Bloc diagram for parameters tuning.	96
Fig. V.5	Bloc diagram for Hybrid ANFIS-FOPID dynamic controller.	96
Fig. V.6	Convergence curve of the algorithms.	98

Fig. V.7	Step response Comparison for all velocity controllers.	99
Fig. V.8	**.** Comparison of step Response for the orientation.	100
Fig. V.9	(a) strait trajectory in XY plane,(b) error of the distance, (c) the theta angle error.	101
Fig. V.10	(a) circle trajectory in XY plane,(b) linear velocity response for all controllers ,(c) angle response of all controllers	102
Fig. V.11	Error of the distance, .	103
Fig. V.12	(a) Five shape trajectory in XY plane,(b) angle response corresponding to the path for all controllers, (c) linear velocity corresponding to the path for all controllers, ,(d) error of the distance, (e) the theta angle error.	104
Fig. V.13	(a) zigzag path in XY plane ,(b) the integral square error (ISE) for all controllers	105
Fig. V.14	(a) the theta angle response corresponding to all controllers ,(b) the linear velocity corresponding to all controllers.	106
Fig. V.15	Quadrifolium path.	107
Fig. V.16	Error corresponding to (a) x,(b) y; (c) the theta angle ,(d) the trajectory tracking.	107
Fig. V.17	**.**Square trajectory.	108
Fig. V.18	Error corresponding to (a) x,(b) y; (c) the theta angle ,(d) the trajectory tracking.	108

Chapter VI

Fig. VI.1	self designed robot plat form	110
Fig. VI.2	The global control architecture	111
Fig. VI.3	The platform mechanical design: (a) Top view of the base,(b) the electronic circuit location in the base,(c) Bottom view,(d) the assembled platform.	112
Fig. VI.4	wiring diagram of the entire hardware	113
Fig. VI.5	Arduino blocs library	115
Fig. VI.6	the complete designed simulink bloc diagram	116
Fig. VI.7	The designed simulink model for the PWM signals and orientation.	117
Fig. VI.8	**8** the designed simulink model for encoder reading: (a) the structure of the RPM reading ,(b) the S function bloc and the rollover bloc .	119
Fig. VI.9	the designed simulink model for pose estimation.	119

Fig. VI.10	Circular path; (a) the actual circle trajectory (b) estimated values from encoders	121
Fig. VI.11	circular path control efforts: (a) the tracked speed (b) the tracked orientation (c) error of X axis	122
Fig. VI.12	the tracking errors of the circular path: (a) error of Y axis (b) error of the orientation (c) error of the distance	123
Fig. VI.13	(a) The 2D print of the infinity trajectory, (b): estimated XY plane for infinity path	124
Fig. VI.14	lemniscate control efforts: (a) the tracked speed (b) the tracked orientation (b) error of X axis	125
Fig. VI.15	tracking errors of lemniscates :(a) error of Y axis (b) error of the orientation (c) error of the distance	126
Fig. VI.16	(a) The 2D print of the quadrifolium trajectory, (b): estimated XY plane for quadrifolium path	127
Fig. VI.17	**.** quadrifolium path control efforts: (a) the tracked speed (b) the tracked orientation (c) error of X axis	128
Fig. VI.18	tracking errors for quadrifolium path :(a) error of Y axis (b) error of the orientation (c) error of the distance	129
Fig. VI.19	**(a)** The 2D print of the rectangular path, (b): estimated XY plane for rectangular path	130
Fig. VI.20	control efforts of the rectangular trajectory **:**(a) the tracked speed (b) the tracked orientation (c) error of X axis	131
Fig. VI.21	tracking errors of the rectangular trajectory**:** (a) error of Y axis (b) error of the orientation (c) error of the distance	132

List of Tables

Tables	Tables Name	Page
	Chapter II	
Tab. II.1	Physical parameter of the robot	32
	Chapter III	
Tab. III.1	ISE values of the seven algorithms	59
Tab. III.2	The tuned gain details of linear velocity for all Controllers .	60
Tab. III.3	Performance characteristic for velocity Controller	61
Tab. III.4	The tuned main parameters of all controllers for angle control.	62
Tab. III.5	. Performance characteristic of all controllers for angle control.	62
	Chapter IV	
Tab. IV.1	ISE values of the five algorithms	85
Tab. IV.2	The tuned gain details of FOPID1 Controller for all algorithms.	86
Tab. IV.3	Performance characteristic for FOPID1.	86
Tab. IV.4	. The tuned main parameters of FOPID2 controller for all algorithms.	87
Tab. IV.5	Performance characteristic for FOPID2.	88
	Chapter V	
Tab. V.1	ISE values of all controllers.	98
Tab. V.2	The tuned gain details of linear velocity for all Controllers	99
Tab. V.3	Performance characteristic for velocity Controller.	99
Tab. V.4	The tuned main parameters of the three controllers for angle control.	100
Tab. V.5	Performance characteristic of the three controllers for angle control.	100
	Chapter VI	
Tab. VI.1	hardware component of the platform	102

List of publication

[1] Euldji, Rafik, Noureddine Batel, Redha Rebhi, and Mohamed Redha Skender. "Optimal ANFIS-FOPID with back-stepping Controller design for wheeled Mobile Robot control." Journal of Control Engineering and Applied Informatics 24, no. 2 (2022): 57-68.

[2] Euldji, Rafik, et al. "Optimal Design and Performance Comparison of a Combined ANFIS-PID with Back Stepping Technique, Using Various Meta-Heuristic Algorithms to Solve Wheeled Mobile Robot Trajectory Tracking Problem." Journal Européen des Systèmes Automatisés 55.3 (2022): 281-298.

[3] Euldji, R., Batel, N., Rebhi, R., Kaid, N., Tearnbucha, C., Sudsutad, W., Lorenzini, G., Ahmad, H., Ameur, H., Menni, Y. (2022). Optimal backstepping-FOPID controller design for wheeled mobile robot.Journal Européen des Systèmes Automatisés, Vol.55, No.1, pp.97 107. https://doi.org/10.18280/jesa.550110

General Introduction

General Introduction

1. Background & Motivation

Autonomous systems can be introduced as a programmable and multi-purpose machine, able to learn and to extract data from its environment using sensor technology and control units to plan and execute its mission within its surrounding without human interventions [1]. The autonomous vehicles are known as being used to improve the quality of life, with pollution reduction (greenhouse gas) and accident prevention (for instance, 93% of road traffic accidents are caused by human error) [2]. The autonomous system can to operate numerous kinds of applications including cleanings, house helpers, deliveries, safety, educations, supermarket helpers, receiving officers since it can be used to control the processing time, and to provide precise and reliable outputs. The autonomous processes do not need to be paused. They can work all day without pause or rest.

The mobile robot is one subset of the autonomous system, at the beginning of the robot history most of the researches were focused on robotic arms and manipulators, due to the rapid growth of the automobile industry in the 1980s led to the domination of commercialized manipulators. Since then, the manipulators have become an integral part of the automobile manufacturing industry. In comparison with the manipulators, mobile robotics is still a relatively young area of research. The great deal of interest in mobile robots is because of their mobility [2], basically mobile robots are grouped into numerous categories, including wheeled and legged; aerial, ground, and underwater; holonomic (omnidirectional) and nonholonomic robots; homogeneous multi-robot systems and heterogeneous; and large scale, macro, and micro robots; see reference [3] for more details.

The scope of this thesis is focused on the nonholonomic wheeled mobile robots. The reason behind this attention from many researchers is because of their controllability issue, caused by the limited number of control inputs, while the vehicle may have three degrees of freedom, in another word this type of robots are under-actuated, which means a pure feedback control cannot handle the control problem in complex environment. Nowadays, wheeled mobile robots (WMRs) are become in a wide range of civilian and military applications (such as reconnaissance, agriculture, rescue, space exploration, surveillance, combat, materials transportation, etc.) [3,4]. This led to the appearance of new indoor and outdoor tasks that work in complicate environment, where a plenty of problems are waiting to be addressed in order to improve the industrial environments. Until now, the control problem of the tracking trajectory is still challenging and attracting many researchers.

To achieve the path tracking, a WMR is obligated to follow a specific trajectory such as continuous gradient trajectories and non-continuous gradient trajectories. In order to comply with new tasks requirements, a new powerful control method calls for to be achieved. This requires knowledge from many disciplines of engineering, including mechanical, electrical, and computer engineering.

A WMR consists mainly of components such as sensors, intelligent controllers and actuation systems. Those components are the key element of achieving the autonomy and make a WMR capable of working and performing its tasks without human assistance. Hence, a UGV operates autonomously based on its perception; all the previous challenge motivates us to make these contributions.

2. Objectives and contributions of the Research

The fundamental objective of this thesis is to design an intelligent robust controllers, which is capable to track pre-defined trajectories in a complex environment with less tracking error , using smooth control efforts ,in order to deal with continuous gradient trajectories such as lemniscate, circular, and linear trajectories and non-continuous gradient trajectories such as square trajectories. To achieve the aims . The author objectives of research work can be divided into categories based on the reported research topics as follows:

1) To model the UGV based on non-holonomic kinematic characteristic, dynamic and actuators units.

2) To modify and improve the performance of traditional PID controller by hybridize it with the adaptive neuro fuzzy inference system (ANFIS), abbreviated as ANFIS-PID controller, optimized using various met heuristics algorithms. This is done by finding the optimal torque control action that will minimise the tracking error (the travel time and travelling distance) of the mobile robot by Introducing seven new meta-heuristics techniques to optimise the new controller , based on the objective function under the name integral square error (ISE).where this techniques are: particle swarm optimiser (PSO), grey wolf optimiser algorithms (GWO), dragonfly optimizer (DA), ant lion optimiser (ALO), grasshopper algorithm (GOA) , moth flame optimizer (MFO), and the artificial bee colony algorithm (ABC),

3) To introduce a novel optimal backstepping fractional order proportional integral derivative (FOPID) controllers to tackle the problem of trajectory tracking. The parameters of the FOPID controllers are optimally tuned, using various intelligent techniques. The obtained results have demonstrated an observable development for minimizing the trajectory tracking error and reducing control efforts. Thus, this in turn leads to improve the operational response of the system in continuous and non-continuous gradient trajectories. However, the improvement has not been satisfied with the non-continuous gradient trajectory at this stage.

4) To propose new controllers based on the FOPID controllers and an optimal hybrid adaptive fractional neuro-fuzzy inference controller, abbreviated as ANFIS-FOPID controller, a robust controller that merge the advantages of the FOPID controller with the ANFIS controller and

benefiting from the powerful HWGO optimiser for adjusting the parameters of the proposed controller, based on the cost function namely integral square error (ISE).

5) To practically implement the architecture of the UGV to validate the overall performance of the proposed intelligent controllers.

3. Achievement

The proposed controller plans the desired trajectory of optimal smoothness for the mobile robot and executes the trajectory tracking by generating optimal torque control actions. Simulation results confirm the over robustness of the proposed hybrid ANFIS-FOPID controller in handling both smooth and sharp chapped trajectory, compared to the FOPID and the ANFIS controller, several tests has been made, based on transient parameter, such as rise time (tr), settling time (ts), overshoot (Mp), peak, and peak time (tp). Many cases study were applied to demonstrate the superiority of the controller, like circular path, zigzag path, five shape path, and a square path.

The experimental work show the effectiveness of the proposed control methodologie, which achieved excellent tracking for Lemniscates and Circular as a continuous and rectangle as a non-continuous gradients trajectories actual mobile robot and reduced the tracking error to less than 1cm. The actions of the proposed controller were small smooth values of the torques for right and left wheels without sharp spikes.

4. Thesis organisation

This thesis is presented as follows:

Chapter one brief history to mobile robot and overviews of previous studies related to this field.

Chapter two describes the kinematics and dynamics mathematical model of the nonholonomic wheeled mobile robot.

Chapter three describes the thirst proposed method which is the combination of ANFIS with PID controller

Chapter four is devoted to the proposed novel approach to solve a trajectory-tracking problem based on the optimal fractional order PID controllers, combined with the backstepping controller ,optimized using various controller.

Chapter five the suggestion of an combination between the adaptive neuro fuzzy inference system (ANFIS) controller and the FOPID controller .

Chapter six presents the hardware and software design of the robot platform and the experimental results of the proposed controller.

Finally, a conclusion of the entire work and suggestions for future work.

Chapter I

Background and Literature Review

Chapter I Background and Literature Review

I.1 Introduction

In this chapter, we first present a background on the systems considered in the thesis, Then, we highlight the main issues related to the control problems considered and how they are tackled by the proposed solutions.

I.2 Background

The classic definition of a robot is a machine with a certain level of intelligence, a compilation of algorithms for sensing, reasoning, and moving about space. Basically, robots can be divided into two categories, fixed and mobile robots [2]. Fixed robots are mounted on a fixed surface and materials are brought to the workspace near the robot. A fixed robot is normally used in mass production, as in car factories, for welding or stamping. Mobile robots have the capability to move around in their environment and are not fixed to one physical location; therefore, the mobile robot can be defined as a mechanical device that performs automated tasks, whether according to direct human supervision, a pre-defined program, or a set of general guidelines, using artificial intelligence (AI) techniques [5].

At the beginning of the robot history, most of the research where focusing on developing robotic arms and manipulators. Robot arms comprise a 2 billion dollar industry. Bolted at its shoulder to a specific position in the assembly line, the robot arm can move with great speed and accuracy to perform repetitive tasks such as spot welding and painting (Fig I.1) [5].

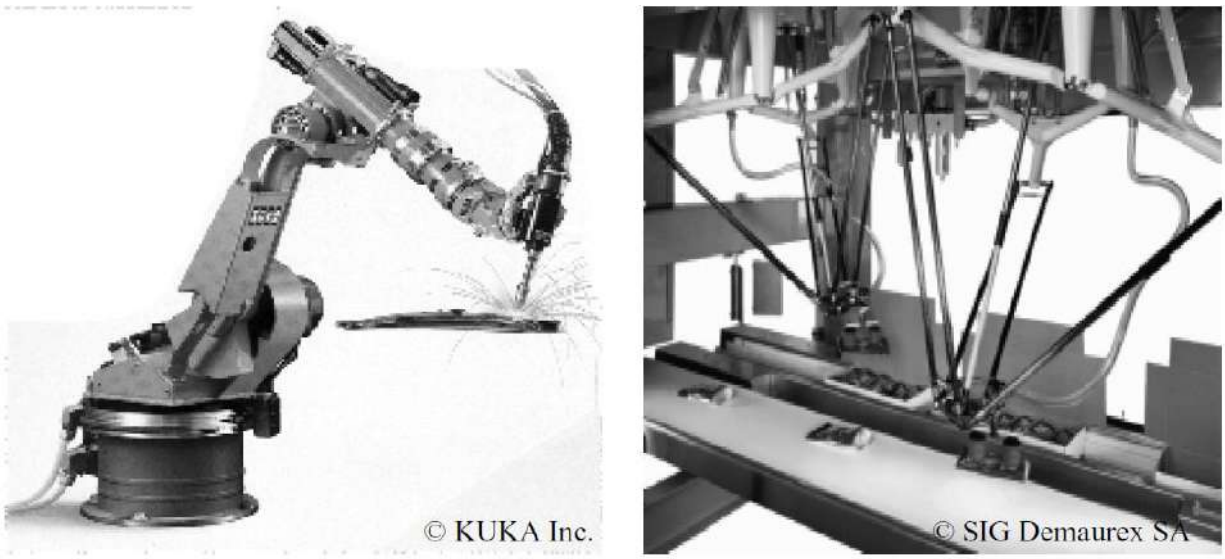


Figure I.1 Picture of auto assembly plant-spot welding robot of KUKA and a parallel robot Delta of SIG Demaurex SA (invented at EPFL) during packaging of chocolates [5].

In the electronics industry, manipulators place surface-mounted components with superhuman precision, making the portable telephone and laptop computer possible. Yet, for all of their successes, these commercial robots suffer from a fundamental disadvantage: lack of mobility.

A fixed manipulator has a limited range of motion that depends on where it is bolted down. In contrast, a mobile robot would be able to travel throughout the manufacturing plant, flexibly applying its talents wherever it is most effective.

Mobile robots can be categorized into wheeled, tracked or legged robots, and they are more useful than fixed robots. Mobile robots are increasingly used in industry, in service robotics, for factories (e.g. in delivering components between assembly stations) and in difficult to access or dangerous areas such as space, military environments, nuclear-waste cleaning and for personal use in the forms of domestic vacuum cleaners and lawn mowers. In comparison with the manipulators, mobile robotics is still a relatively young area of research.

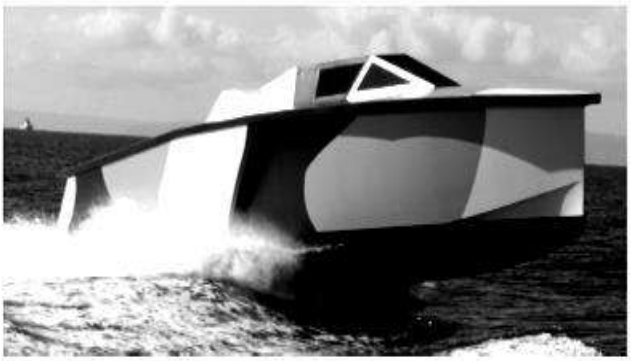
1.2.1 Autonomous mobile robot classification

Several kinds of robot mobilities exist and they can be categories according to their work space/environment. A brief description of the most common types is given below [3-5] :

A. Underwater robots which are widely called Autonomous Underwater Vehicles (AUVs) (Fig I.2a).
B. Boat robots, called Unmanned Surface Vehicles (USVs) (Fig I.2b).
C. Aerial robots which are commonly called Unmanned Aerial Vehicles (UAVs) or drones (Fig I.2c).
D. Ground robots which are commonly called Unmanned Ground Vehicles (UGVs). Furthermore, the UGV can use different devices to move such as legs (Fig I.2d), wheels (Fig I.2e) or specialized wheels [indoor crc press] such as tracks wheels or the one used by the robot Curiosity (Fig I.2f). The different works developed in this manuscript are dedicated to UGVs with wheels. The term mobile robot will always designate, in what follows, a UGV and if not, the mobile robot structure will be explicitly defined in the text. More specifically, the focus will be on the control of UGVs to perform autonomous navigation even in highly cluttered and dynamic environments. In the adopted methodology, it is taken as principle that if the proposed control architecture can be reliable in that kind of complex and constrained environment, it could be obviously even more reliable in a less constrained one.



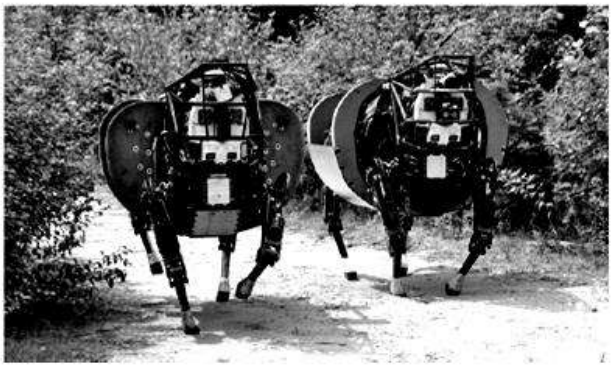
(a)



(b)



(c)



(d)



(e)



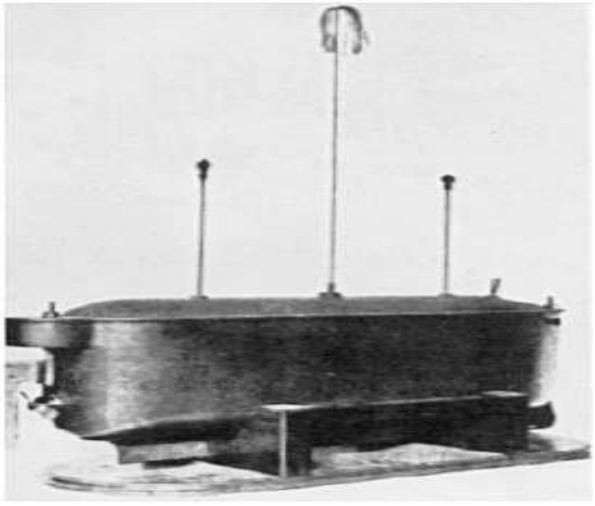
(f)

Figure I.2 Different kinds of mobile robot locomotion [3].

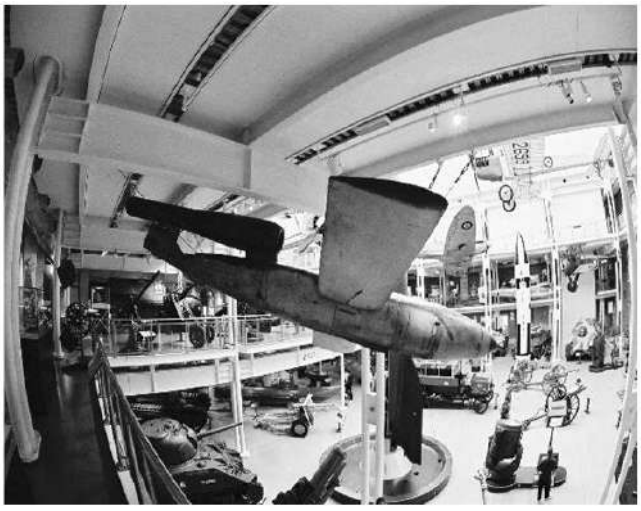
1.2.2 Early history of mobile robot

Perhaps, the first electrical mobile robot was the autonomous vehicles built by Nikola Tesla in the 1890s. Tesla developed wireless, radio-controlled vehicles [6]. Fig I.3a shows one of his remote-controlled aquatic vehicles. During the early to the mid-1940, the first steps in the modern

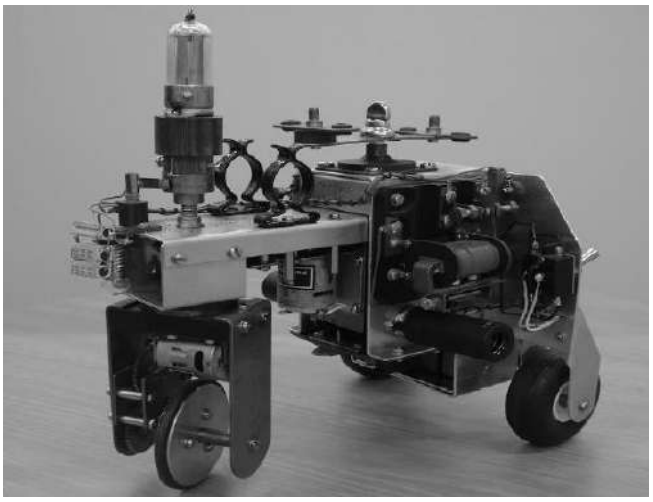
electronic robotic systems were made. The mathematician Norbert Wiener is recognized as the inventor of cybernetics and modern robotics. His studies were focusing on the systems regulation and their application to control. During World War II, he was engaged in a project to develop an anti-aircraft gun (see Fig I.3b).



(a)



(b)

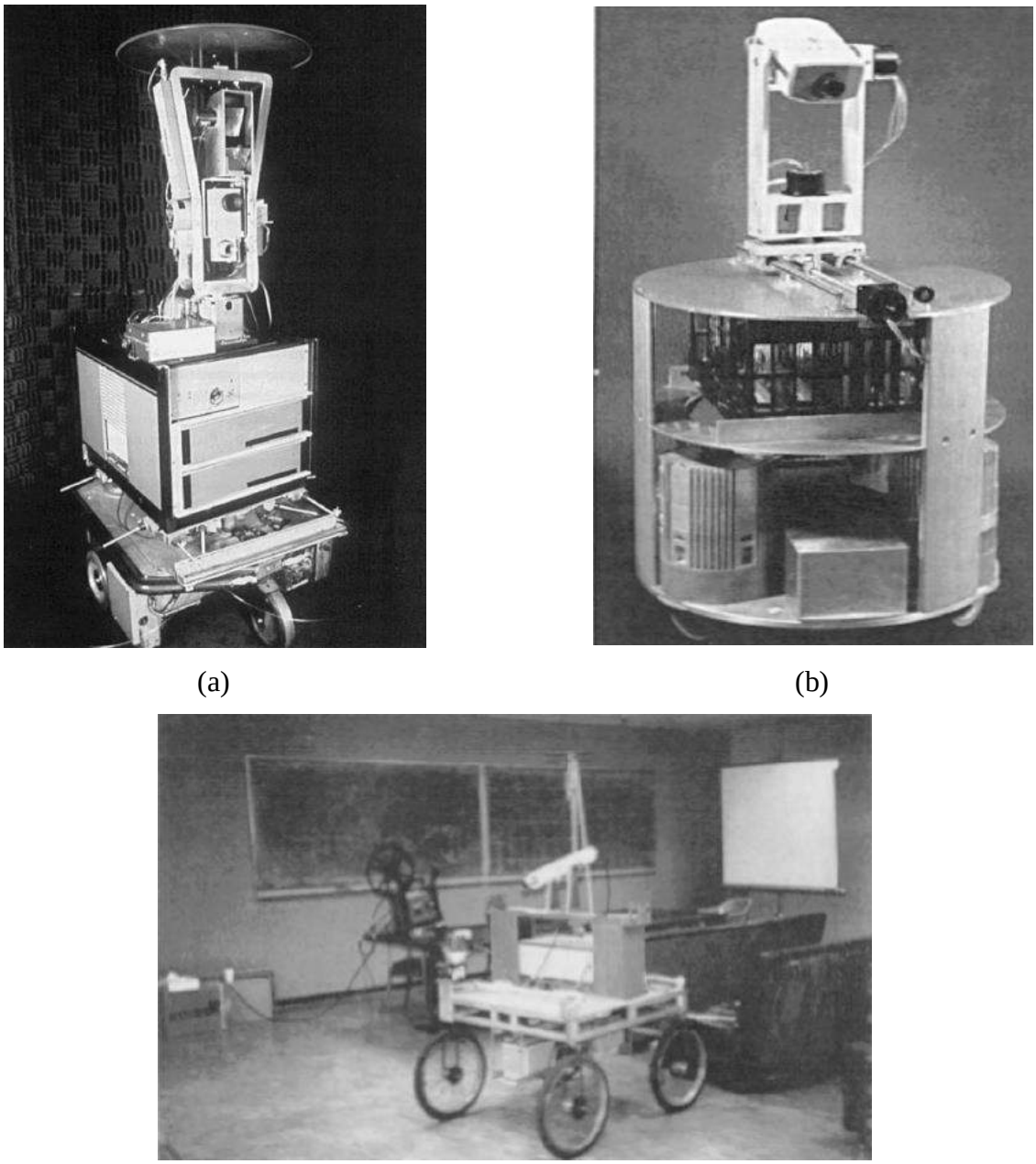


(c)

Figure I.3 Analog robots. (a) Tesla's robot (b) flying bomb. Currently hanging in the Imperial War Museum, London (c) Walter's robot[6]

One of earliest fully autonomous vehicles is the Walter's electronic turtle (see Fig I.3c)[6].named after W. Grey Walter between 1950 and 1951 ,the machine i had phototube eyes, microphone ears, contact-switch feelers, and capacitors used as memory devices to perform associations. Walter named the robot Tortoise after the creature in *Alice in Wonderland*. The Tortoise performed tasks such as heading towards well-lit regions, locating the recharging hutch, and wandering without mishap.

The rapid development of digital computers leads to the appearance of new more complex mobile robots at the Stanford Research Institute Between 1966 and 1972, Nils Nilsen, Charles Rosen, and other researchers developed the first mobile robot to be operated using artificial intelligence techniques [6]. Under the name Shakey, The 5-foot-tall robot used two stepper motors using a differential drive mechanism ,equipped with sensors. An optical rangefinder and vidicon television camera with controllable focus and iris were mounted on a tilt platform for sensing. Two radio channels, one for video and the other for providing the command. Shakey is shown in Fig I.4a.



(a) (b)

(c)

Figure I.4 The first autonomous ground vehicles (a) SRI's Shakey (b) CMU "Rover" (c) Stanford "Cart"[6]

The Later designed robots like the Stanford Cart and the CMU Rover were considered as the extended versions of Shakey and investigate to plugging up its shortcomings. The Stanford Cart was built as a remotely controlled mobile robot with an onboard TV by Adams in 1960 for his research project; later improvements were supervised by Earnest.

The Stanford Cart (see Fig I.4c) was developed between 1973 and 1979 at the Stanford Artificial Intelligence Laboratory and transferred to Carnegie Mellon University (CMU) in 1980. Major modifications and new solutions to a number of classic robot problems were developed. In order to locate objects and planned paths to avoid obstacles, a single camera is used by taken nine images taken at different positions from environment. A comparison of the images over time was used to determine the motion of the cart. The robot was controlled by an off-board computer program that drove the cart through cluttered spaces. The cart moved roughly 1 m every 10 to 15 min. However, the Stanford Cart introduced a number of limitations in order to recognize these limitations, the CMU Rover project (Fig I.4b) was developed (started in 1980), and a synchronous drive-like mechanism was used. Infrared and sonar proximity sensors were added to the robot and modified the camera mount for the video sensor so that it could pan and tilt as well as slide, which were not possible with the Stanford Cart.

I.2.3 Current day mobile robots

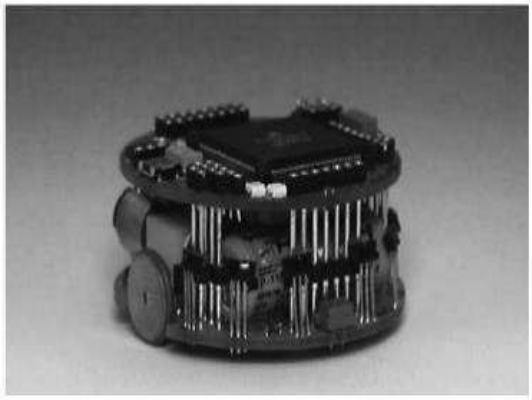
Nowadays, the Walter's and Nilsson's robots have been confined to the museum, now robots are designed with portable computers and easy hardware to be implemented. The robots such as the ASIMO or even AIBO are yet to become a part of human society, partially because of the price and also since these robots are works across various domains, as a coffee delivery robot in office environments, actors of sorts in stage plays, soccer players, assistants to the elderly with their daily chores and various others such arenas. We mention, for example, some modern robots [7]:

1) Khepera

In the late 1990s, the EPFL developed a robot named Khepera and commercial it with K-Team Corporation. the robot was mostly used for education and research and often credited with being a vital aspect for the development of evolutionary robotics. Fig I.5a shows the original version of Khepera [ref].

2) Roomba

In 2002 the iRobot company made a vacuum cleaning robot known as Rooba. This robot can change directions when encountering an obstacle when it vacuuming and it can detect an edge, thus preventing it from tumbling down stairs. To aid there is a pair of brushes, rotating in opposite directions which helps to pick up dirt from the floor. Fig I.5b Shows the Roomba 780.



(a)



(b)

Figure I.5 (a): Khepera (b) Roomba [7]

3) Pioneer Robot

The Pioneer was developed by ActivMedia Robotics, Georgia. The robot is provided with a ring of 8 forward-facing sonars and 8 optional rear-facing sonars. A maximum speed of 1.6 meters per second can be attained by robot as well the robot can carry a payload of up to 23 kg. Fig I.6a shows the Pioneer base is mounted with an arm manipulator [7].

4) The Turtlebot

Turtlebot was established by Melonee Wise and Tully Foote at Willow Garage in late 2010. Fig I.6b represents the Turtlebot 2, which has a Yujin Kobuki base, a Kinect sensor and a laptop with a dual core processor and is available as open source hardware with a BSD license. Priced at about $1000, the robot has found support from the ROS community and is used in numerous universities in their robotics laboratory courses.

5) Segway

The University of Plymouth, in collaboration with BAE Systems and Sumitomo Precision Products, developed a battery-powered, self-balancing two-wheeled scooter, namely Segway. This robot uses gyroscopic sensors and an accelerometer for maintaining balance. The Segway is a recent ancestor of autonomous vehicles of the immediate present and the promise of the future. At a maximum speed of about 12 miles per hour, it is excellent for the golf course and other short tours and it has also been designed into a companion robot (see fig I.6c) [7].



(a) (b) (c)

Figure I.6 Wheeled mobile robots,(a) the Pioneer Robot (b) the Turtlebot (c) Segway

6) AIBO

The AIBO robot was produced by SONY, while Toshitada Doi and Masahiro Fujita are the pioneers whose work led to its development in 1998 . The iconic series of robots of this company were inspired by dogs and lion cubs and their name means `Artificial Intelligence RoBOt, and also suggests the Japanese, aibo, meaning pal, friend, partner (see Fig I.7a).



(a) (b) (c)

Figure I.7 Walking robots, (a) AIBO (b) Honda ASIMO (c) NAO

7) Honda ASIMO

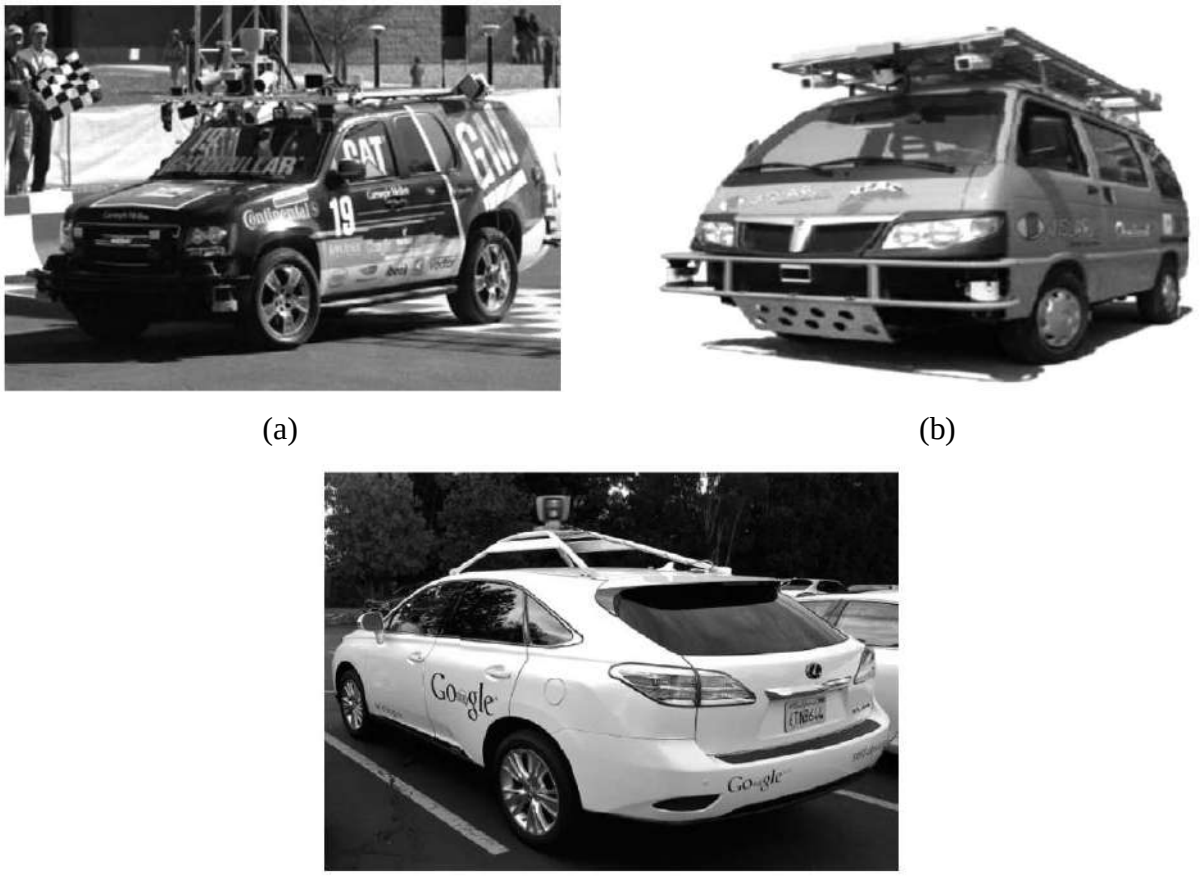
The ASIMO robot of Honda's celebrated humanoid is said to be insprired by Isaac Asimov but the acronym reads otherwise, `Advanced Step in Innovative Mobility'. the robot is designed as a walking assistant (Fig I.7b) [7], stands 130 cm, weighs about 50 kgs, can attain maximum speeds of 9 km per hour in running mode, can be controlled by PC, wireless controller, or voice commands.

8) Nao

This robot was made in 2006 as an open source software by the Franch company, Aldebaran Robotics (see Fig I.7c). In 2013, Nao was acquired by Japan's SoftBank Robotics [7]. Standing at 58 cm and weighing 4.3 kg, it is powered by the Intel Atom processor and has an assortment of sensors, HD cameras, microphones, sonar rangefinder, infrared, tactile and pressure sensors. The robot has quite a reputation: it has acted in a ballet, been the most prolific robot in robo soccer, has been employed as a therapeutic robot for children suffering from autism, has been programmed to be the first robot with deontic ethics and also, to a degree it is the first robot to attain sentience, albeit for a very short time.

9) The autonomous driving car

The autonomous driving car is another common field of robotics engineering that has fascinated and attracts more and more investigators in the last decade [2], from academic, industry, and the military.



(a) (b)

(c)

Figure I.8 autonomous driving cars,(a) The Google self-driving car (b) The VisLab autonomous vehicle (c) Boss, the autonomous vehicle that won the 2007 DARPA Urban Challenge

The DARPA Urban Challenge, initiated in 2007, is widely known as the competition of autonomous driving cars. Boss, developed by Carnegie Mellon University (see Figure 1.8.a), was the first winner of this competition with a speed of 22.53 km/h , Two driverless vans, developed by VisLab (see Figure 1.8.b), at University of Parma in Italy, spent three months traveling to the World Expo of Shanghai from Parma, Italy, an overall traveling distance of around 13,000 km ,Google self-driving cars have traveled over 2,400,000 km as of March, 2016, Figure 1.8.(c)[2].

I.3 Holomic and nonholomic proprieties

In the last decade, researchers and scientists have designed numerous locomotion structures for autonomous mobile robots. We have to indicate that the use of wheels as a locomotion mechanism is the most efficient way for unmanned ground vehicles. Because the control of the electric actuators is easy compared to legged vehicles, just changing the voltage input makes the robot changing its velocity and direction. The scope of this thesis is focusing on the nonholonomic wheeled mobile robots. To better understand the difference between the holomic and nonholomic robots, a brief explanation is required[8-10].

I.3.1 Holomic wheeled mobile robot

The Holonomic mobile robots as it shown in Fig1.9 are robots that are able to move in any direction without changing their orientation. Fig I.9b shows a holomic mobile robot with three onmi wheels and FigI.9a shows the Uranus robot with four mecanum wheels (often called Swedish wheels). An omniwheel is an ordinary wheel reinforced with rolls on its outer circumference. These rollers spin liberally about axes in the XY plane of the wheel and tangential to the wheel's outer circumference (see fig) ,they provide sideways sliding while the wheel drives forward or backward without slipping in that direction [8-10]



(a) (b)

Figure I.9 holomic mobile robots (a) URANUS (b) 3WD Compact Omni-Directional[10]

Mecanum wheels are similar, except that the spin axes of the circumferential rollers are not in the plane of the wheel. The sideways sliding allowed by omniwheels and mecanum wheels ensures that there are no velocity constraints on the robot's chassis[9].

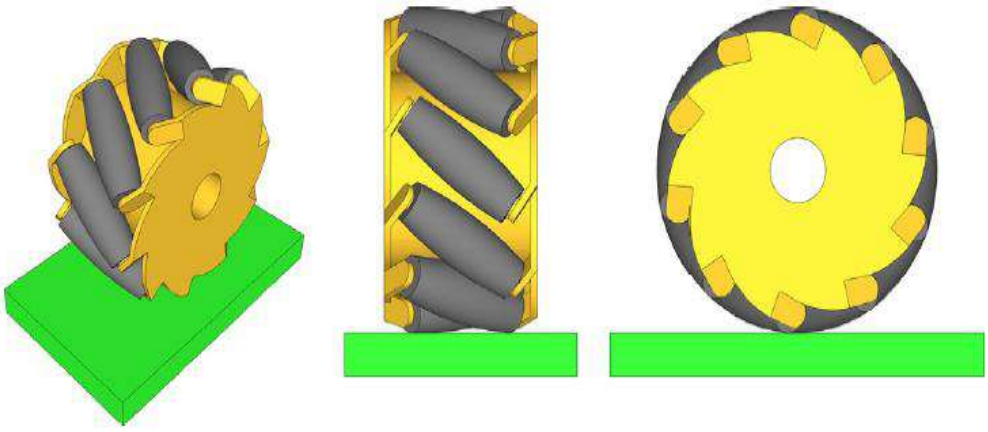


Figure I.10 mecanum wheel with rollers mounted around the wheel rim, each roller axis is at 45 degree from the main wheel plane and at 45 degree to the line parallel with the wheel's axis [9].

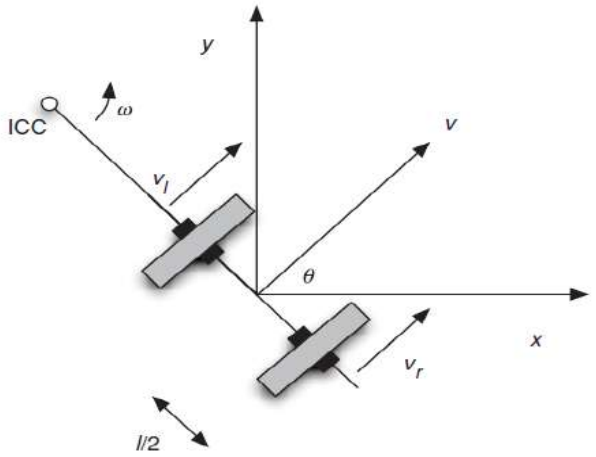
I.3.2 Nonholomic wheeled mobile robot

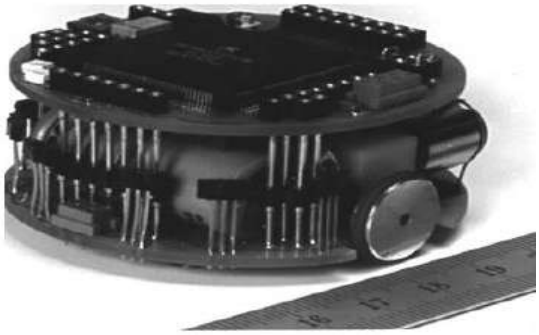
On the other hand, the nonholonomic mobile robots must have to change their orientation in order to move at some directions, using only two inputs to control three degrees-of-freedom outputs. They can be expressed as a class of mobile robots that carry differential constraints (also called nonholonomic or velocity constraints). Nonholonomic wheeled mobile robots work using classic wheels that do not support sideways slipping, they can only move in the direction perpendicular to the axis of the driving wheels. There are four different types of nonholonomic wheeled mobile robots [6,8-10] :

1) Differential Drive

The differential drive mechanism is well known in many indoor applications, because of the simplicity structure and low-cost hardware design .this type of mobile robot consists of two independent driving actuators with wheels mounted on a common axis, are usually balanced by a one or two passive caster wheel and being driven either forward or backward. Whilst the velocity of each side can be varied, the vehicle must rotate about a point that lies along their common left and right wheels' axis. The point that the UGV rotates about is known as the instantaneous centre of curvature (ICC) as shown in Fig. I.10a. By varying the velocities of two wheels, the motion can be varied to track given trajectories.

.



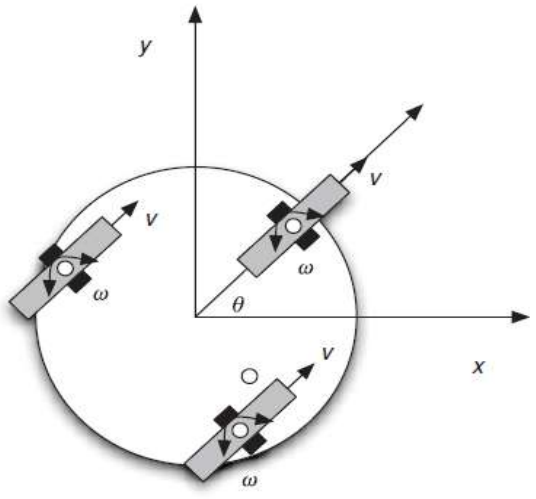


(a) Differential drive (b) Khepera robot

Figure I.11 Differential drive kinematics. Most differential drive robots use castor wheels for stability. (b) the khepera Appears with the kind permission of A. Herzog [6].

2) Synchronous Drive

For the synchronous drive structure, each wheel is able to drive separately. And all the wheels can work in unison based on a typical configuration of wheels , this leads to a holonomic behaviour. The orientation of a steered wheel of the rotation axis can be controlled, as shown in Fig. I.12. In addition,The synchronous drive has the ability to control the orientation of their pose directly . Because it ability to control the wheels in parallel, the synchronous drive always rotates about the centre of the UGV [6].



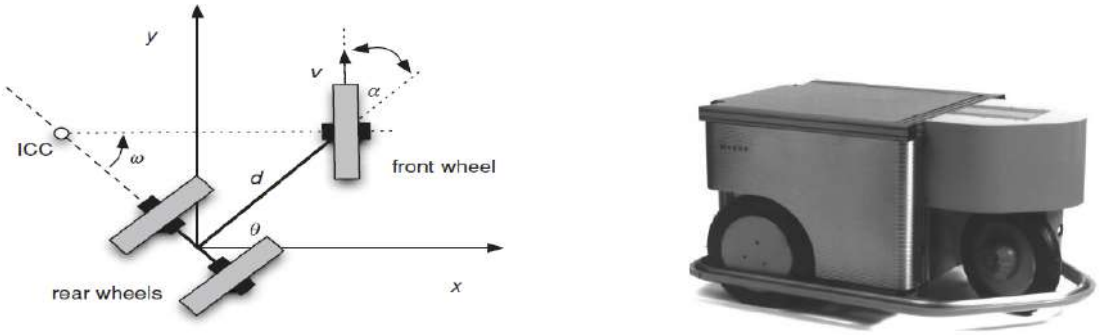


(a) Synchronous drive kinematics (b) ARK-1 robot

Figure I.12 Synchronous drive mechanism. (a) the kinematic of the robot (b). The ARK-1 robot is built around the Cybermotion Navmaster platform. Which utilizes a synchronous drive.

3) Tricycle, bogey, and bicycle drive

Here the study is focusing on the Tricycle robot, because it has very similar kinematics to the bogey (wagon), and bicycle drive robots. And because the tricycle problem is the most often use in mobile robots. A typical tricycle drive is demonstrated in Fig I.13 b, where Fig I.13a shows the kinematic of the robot. in this structure, three wheels with odometers on the two rear wheels, with steering and power provided through the front wheel. The robot motion is controlled by the steering direction α and velocity v provided through the front wheel.

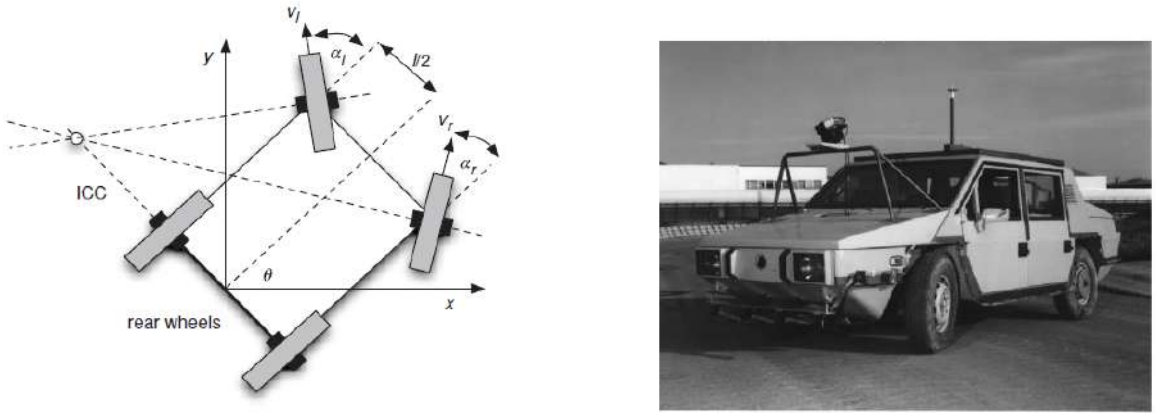


(a) Tricycle shematics (b) Mecos mobile robot

Figure I.13 Tricycle mobile robot. Copyright Mecos Robotics AG. Used with permission.

4) Car Drive (Ackerman Steering)

Commonly, the Ackerman steering drive vehicle method is used for outdoor application. An example of this autonomous vehicle is shown in Fig. I.14 below. This mechanism is an elegant and a simple structure in order to approximate ideal steering used in motor vehicles, the inside front wheel is rotated slightly sharper than the outside wheel to reduce tire slippage. Ackerman steering provides nearly an accurate dead reckoning solution whilst supporting traction and ground clearance.



(a) Ackerman steering (b) Oto Melara

Figure I.14 Ackerman mechanism (a) robot kinematics . (b) The Oto Melara. Appears with kind permission of OTOBREDA

I.4 wheeled mobile robot control architecture

Fig I.15 shows the entire control architecture of a wheeled Mobile robot (WMR). It can be clearly seen that for the trajectory tracking and navigation of the WMR, several elements are necessary, various sensors are utilized for getting data from the environment. Many control approaches can be applied in order to get the localization of WMR in its workspace. processing is necessary in order to make decisions about how to act and plans the next action and it execution in the environment. Finally the robot moves based on a regulation method, for achieving its goals [11].

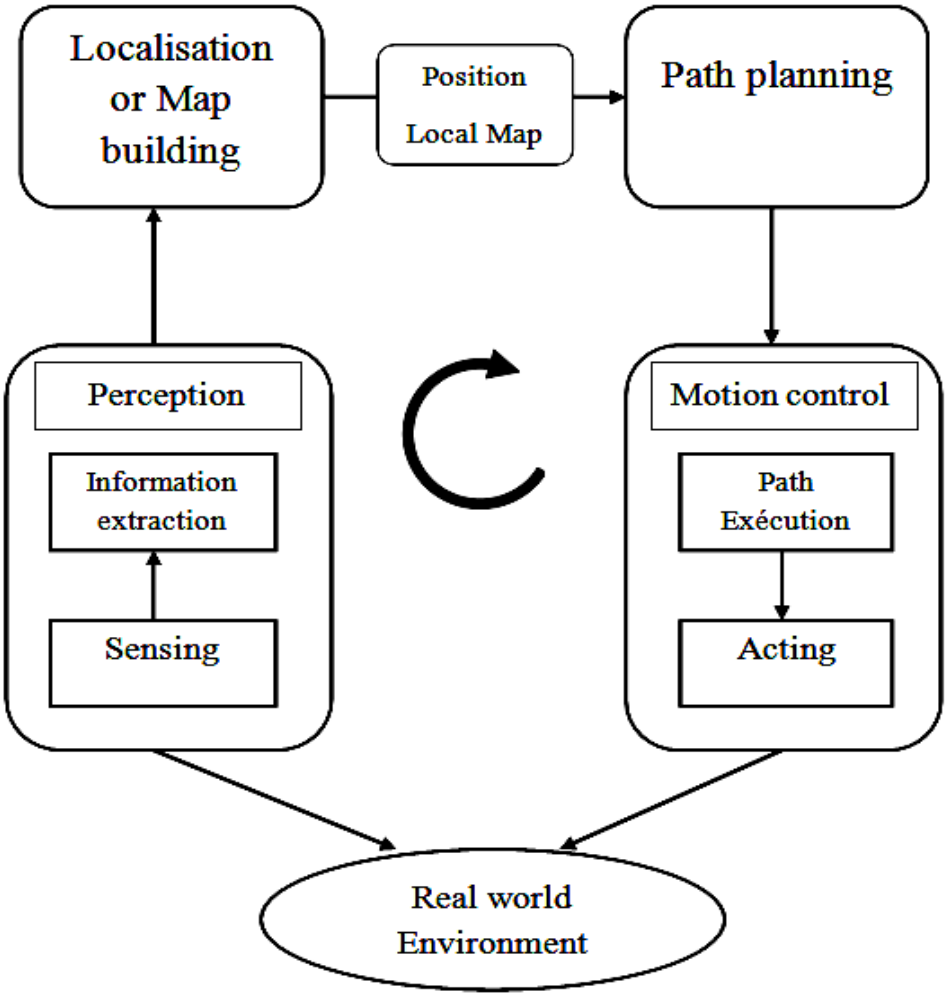


Figure I.15 General Control scheme for UGV.

I.5. Control Problems of the Mobile Robot

Most WMRs can be classified as nonholonomic mechanical systems. The control problems here is that the motion of a WMR in a workspace carrying three degrees of freedom, while it has to be handled using only two control inputs under the nonholonomic constraint. These control issues have freshly interested a great attention of many researchers in the control society .many techniques have been improved to deal with the control problems, this can be divided into three groups [11-13]:

The first group is the sensor-based control approach used for the navigation problems of the WMR . Since the working environment change by the time, the vehicle must use sensory information's to handle the dynamic environmental changes, such as environment configuration prediction and obstacle avoidance.

The second category for navigation problems of the mobile robot is path planning. The path is generated based on a prior map of the environment and used certain optimization algorithms based on a minimal time, minimal distance and minimal energy performance index. Many methods have been developed for avoiding both static and moving obstacles [11-13].

The third category is the motion control of a WMR, where a predefined trajectory must be tracked accurately. In order to achieve a minimum tracking error, the error of the distance and orientation should be minimal. This thesis is focusing on solving this kind of problem. For that purpose, a detailed literature review will be described in the next sections.

I.6. Tracking Error of the Mobile Robot

The path planning of mobile robot calls for providing an optimal trajectory from an initial position to the goal [11-13]. This requires a minimal tracking error at the shortest run time and with the minimum distance. Tracking errors in mobile robots causes deviations from the planned path, which led to accidents with the obstacles. This problem increases the travelling time and distance, finally the robot can't accomplish its mission successfully. There are three principal reasons for increasing tracking error for mobile robots:

The first major reason for tracking error is the instantaneous changing in the direction at the connecting point while executing the path of the WMR. This problem is destined to a sharp, shaped trajectory, like the diamond shaped trajectory and the zigzag trajectory, which contain many points of inflection. At this point, it can be easy for the WMR to diverge from its determined orbit due to the rapid change of orientation. Therefore, in order to deal with this issue, the mobile robot trajectory must be organized so that the rotation radius is kept at a constant value, if possible.

The second reason for the tracking error becoming higher is due to the small rotation radius of some specific trajectories, which impedes the efficiency of the robot in its task role. In another word, the tracking error rises when the rotation radius becomes small. In curved paths, a considerable error is created due to the centrifugal and centripetal forces, which make the robot slide from its trajectory ,this issue can only be solved using a powerful tracking control methodologies.

The third of the main reason for increasing tracking error is that in some applications the robot need to change the rotation radius in order to deal with complex environment. Because of the variable rotation radius the mobile robot need to change its speed and direction every time, this problem is challenging and demand for high control efforts from the actuators, in order to reduce the path tracking error.

I.7. Previous Works

Many modern control methods have been proposed in the last decade. These techniques can be divided into two research criteria. These are described based on the kinematic model or on the dynamic model[14]. The first group of researchers proposes a simple PID controller to deal with the dynamics of the robot mobile and focusing on improving the kinematic controller, an adaptive technique using neural network and gradient decent for self-tuning the gains of the back-stepping controller in [15], using fuzzy logic for the auto-tuning of the kinematic controller in [16], and the radial bases function network in[17].

The aim of the kinematic controller is to perform linear and angular speed outputs to assure that the difference between the physical and desired coordinate of the robot converges to zero [18].

The principal reason behind favouring this type of controller is that the dynamic controller's composition is more complicate than the kinematic one, while the dynamics depend on robot specifications such as the moment of inertia, mass, frictions [18].

The major problem with this approach is that the disturbances and complete knowledge of the dynamics have been neglected. Which means the perfect velocity tracking hypothesis does not hold in experience, Therefore defining the dynamics of robot mobile is necessary, in order to make sure that the velocity of the mobile robot converges to the generated desired velocity, a torque controller is indispensable in many applications, including high-speed motions and heavy transportation [18]. For that purpose, a sliding mode dynamic controller proposed in[19–23], an online physical parameters adaptation of the robot dynamics proposed in[24], an adaptive back-stepping controller in [25], using adaptive neural network controller for dealing with unmodeled dynamics in [26–29], using fuzzy logic controller in[30, 31],and an adaptive fuzzy type 2 controller for the dynamic and kinematic control in[32].

The strategy of PID control has been one of the most practical and most often used techniques in industry because of the simplicity form and the strong robustness in large working condition of this controller. However, the fixed parameters generally degrade the control performance[33–35]. In order to overcome this issue, Several researches papers concentrate on designing new types of modified PID, like fusing the neural network with a conventional PID controller (NN-PID)[36–38]. The approach takes the simplicity of a PID controllers and the powerful capability of learning, adaptability and tackling nonlinearity of neural networks. Many researches papers use an adaptive tuning methods by combining the fuzzy logic controller with PID[39–42],and investigating on a Reinforcement Learning in[43].However, the demand for new control methods becomes higher and higher according to applications complexity and progress.

A fractional-order proportional integral derivative (FOPID) controller has already taken over the place of the traditional PID. This controller gained an important popularity from several researchers

in many areas, including robotics engineering [44–49]. The major reason behind this attention is that the five parameters (Kp, Ki, Kd, λ, μ) of the FOPID controller can be more adapted for designing better controllers, as being faster with less overshoot compared with the integer- order PID controller, which has only three parameters to be adjust[50, 51]. Naturally, a controller with more parameters to be tuned means more its design becomes difficult. This makes it optimisation a challenging task.

According to various studies, nature inspired algorithms have already proved their capability in scaling the FOPID controller factors. We distingue, particle swarm optimiser (PSO)[52, 53], whale optimiser (WOA) [54], grey wolf optimiser (GWO)[55], using sine-cosine algorithm (SCA) in[56], Henry gas solubility optimiser in [57], using salp swarm optimiser in[58], hybrid particle swarm grey wolf optimiser (PSOGWO) in [59], chaotic atom search optimiser in[60].

The major benefit from FOPID controllers is the rapidity in terms of rise time and settling time. On the other hand, some specific trajectory like the sharp chap one makes the robot changes instantaneously its orientation (a case of square trajectory), which leads to a large overshoot. This problem could harm the robot actuators, due to the robot high coupled nonlinearity, which means the FOPID controller with fixed parameter can not handle some special trajectory, a new controller that can deal with all types of path is required[61, 62].

The concept of combining fuzzy logic controller with a neural network has attracted a remarkable attention in various fields especially in control engineering[63–70].The reason behind this popularity is that the adaptive neuro-fuzzy inference system (ANFIS) mixes the efficiency of fuzzy logic to deal with the uncertainties and the ability of neural networks to learn from plants/processes.

According to various research studies [61, 62, 71], the principal advantage of the ANFIS controlled systems is getting practically zero overshoot. However, it takes a large settling time. In addition, very slow response due to an important rise time value, which is the contrary of the FOPID controllers.

To overcome this issue, several hybrid structures has been made in order to benefit from the robustness of the ANFIS controller and the simplicity of the PID controller, we distingua the following combinations: hybrid ANFIS-PID in[72, 73] ,(ANFIS+I) controller optimizing using multidimensional PSO for Quadrotor Position Control in [74], (ANFIS-PD+I) controller for robot manipulator in [75]. Some papers proposed a nonlinear FOPID controller based on neural networks for controlling the mobile robot in [76, 77]. A combination between fuzzy logic and FOPID for controlling two link robot manipulator in[78].All the previously mentioned combinations give a motivating result, which gave us the inspiration for this contribution.

I.8. Proposed solutions

The objective of this work is to design and implementation of an optimal trajectory tracking controller, to generate continuous gradient trajectories such as lemniscate, circular, and linear trajectories and non-continuous gradient trajectories such as square trajectories this can be summarized in four grand line:

The first solution is the design of a new hybrid adaptive neuro-fuzzy inference controller (ANFIS),with proportional integral derivative (PID) controller optimized using grey wolf optimizer (GWO),abbreviated as GWO-ANFIS-PID controller, a robust controller that combine the advantages of the PID controller with the ANFIS controller.

The second solution is to design a optimal backstepping fractional order PID controller. Two blocs of the FOPID controller are used in the internal loop as a dynamic controller to guarantee that the linear and angular speeds of robot mobile follow their reference values. In addition, a backstepping controller located in the external loop for controlling the kinematics, in the context of designing a robust controller, a powerful optimization technique is needed for the parameters tuning, as a consequence, a comparative study of different nature-inspired algorithms has been invested in selecting the best optimum method. In the same context, this paper proposes the HWGO algorithms to solve this kind of problem. The simulation results in MATLAB highlight the over efficiency of the HWGO algorithm compared to those of PSO, GWO, HPSOGWO, and WOA approaches. The performance criteria used for the minimization of the error are obtained using the cost functions, namely integral square error ISE.

The third solution is to design an optimal hybrid adaptive fractional neuro-fuzzy inference controller, abbreviated as ANFIS-FOPID controller, a robust controller that merge the advantages of the FOPID controller with the ANFIS controller and benefiting from the powerful HWGO optimizer for adjusting the parameters of the proposed controller, based on the cost function namely integral square error (ISE).

I.9. conclusion

This chapter introduced briefly the domain of autonomous mobile robotics while focusing on ground robots with its main achievements/developments/levels of autonomy and challenges. It permitted us also to emphasize/clarify the main backgrounds/ paradigms/motivations and definitions useful for the rest of the thesis.

Chapter II:

Mathematical model and classical control approach

Chapter II Mathematical Model and Classical Control Approach

II.1.Introduction

In order to represent and analyzing the behavior of a robotic system, the knowledge of the complete mathematical model is strongly required, to achieve a robust and powerful control strategies. To develop a mathematical model that includes all the parameters and forces that influence the behavior of the robot is very challenging, according to the literature review, many approach has been done based on linearization techniques to reduce the complexity of modeling. However, this could bring to a limited control strategy. The modeling of an unmanned ground vehicle platform should be consists of kinematic, dynamic and electrical actuator modeling .the Kinematic deals with the motion studies without examining the disturbing forces. while, the dynamic modeling studies the motion that takes into consideration these forces. The Actuator modeling is required to get the relationship between the input control voltages and the angular rotation of the wheels. Each part has been handled and explained separately throughout this chapter. Simulation results have been done using MATLAB, were a classical control technique is shown to prove the need for new control methodology based on the literature review.

II.2. Modeling the Wheeled Mobile Robot

The WMR in Fig. II.1 has two pairs of DC motors, where R represents the driving wheels radius, φL and φR are the left and right spinning angles of wheels, respectively. Separated by a distance L, θ is the robot angle. At the distance d from the mind-point A [15].

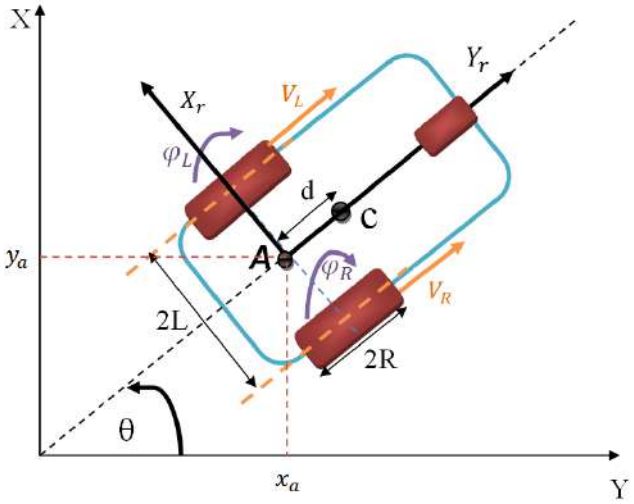


Figure II.1. Schematic design of the WMR.

In order to describe the position of the WMR in his environment, two different coordinate systems (frames) need to be defined [15,42,79].

1. Inertial Coordinate System: This coordinate system is a global frame which is fixed in the environment or plane in which the WMR moves in. Moreover, this frame is considered as the reference frame and is denoted as $\{X_I, Y_I\}$.

2. Robot Coordinate System: This coordinate system is a local frame attached to the WMR, and thus, moving with it. This frame is denoted as $\{X_r, Y_r\}$.

The two defined frames are shown in Fig II.1. The origin of the robot frame is defined to be the mid-point A on the axis between the wheels. As shown in Figure 1, the robot position and orientation in the Inertial Frame can be defined as:

$$q^I = \begin{pmatrix} x_a \\ y_a \\ \theta \end{pmatrix} \tag{II.1}$$

The important issue that needs to be explained at this stage is the mapping between these two frames. The position of any point on the robot can be defined in the robot frame and the inertial frame as follows.

Let $X^r = \begin{bmatrix} x^r \\ y^r \\ \theta^r \end{bmatrix}$ and $X^1 = \begin{bmatrix} x^1 \\ y^1 \\ \theta^1 \end{bmatrix}$ and be the coordinates of the given point in the robot frame and inertial frame, respectively.

Then, the two coordinates are related by the following transformation:

Where R(θ) is the orthogonal rotation matrix

$$R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{II.2}$$

This transformation will enable also the handling of motion between frames.

$$\dot{X}^1 = R(\theta)\dot{X}^r \tag{II.3}$$

It will be seen in the next section that equation (II.3) is very important in deriving the DDMR kinematic and dynamic models as it describes the relationship between the velocities in the Inertial Frame and the Robot Frame.

II.2.1 Kinematic Model of the WMR

Kinematic modeling is the study of the motion of mechanical systems without considering the forces that affect the motion. For the DDMR, the main purpose of kinematic modeling is to represent the robot velocities as a function of the driving wheels velocities along with the geometric parameters of the robot [79]

The motion of a differential-drive mobile robot is characterized by two non-holonomic constraint equations, which are obtained by two main assumptions:

• No lateral slip motion: This constraint simply means that the robot can move only in a curved motion (forward and backward) but not sideward. In the robot frame, this condition means that the velocity of the center-point A is zero along the lateral axis:

$$\dot{y}_a^r = 0 \tag{II.4}$$

Using the orthogonal rotation matrix $R(\theta)$, the velocity in the inertial frame gives

$$-\dot{x}_a \sin\theta + \dot{y}_a \cos\theta = 0 \tag{II.5}$$

• Pure rolling constraint:

The pure rolling constraint represents the fact that each wheel maintains a one contact point P with the ground as shown in Fig II.2. There is no slipping of the wheel in its longitudinal axis (x_r) and no skidding in its orthogonal axis (y_r). The velocities of the contact points in the robot frame are related to the wheel velocities by:

$$\begin{cases} v_{pR} = R\dot{\varphi}_R \\ v_{pL} = R\dot{\varphi}_L \end{cases} \tag{II.6}$$

In the inertial frame, these velocities can be calculated as a function of the velocities of the robot center-point A:

$$\begin{cases} \dot{x}_{pR} = \dot{x}_a + L\theta\cos\theta \\ \dot{y}_{pR} = \dot{y}_a + L\theta\sin\theta \end{cases} \tag{II.7}$$

$$\begin{cases} \dot{x}_{pL} = \dot{x}_a + L\dot{\theta}\cos\theta \\ \dot{y}_{pL} = \dot{y}_a + L\dot{\theta}\sin\theta \end{cases} \tag{II.8}$$

Using the rotation matrix $R(\theta)$, the rolling constraint equations are formulated as follows:

$$\begin{matrix} \dot{x}_{pR}\cos\theta + \dot{y}_{pR}\sin\theta = R\dot{\varphi}_R \\ \dot{x}_{pL}\cos\theta + \dot{y}_{pL}\sin\theta = R\dot{\varphi}_L \end{matrix} \tag{II.9}$$

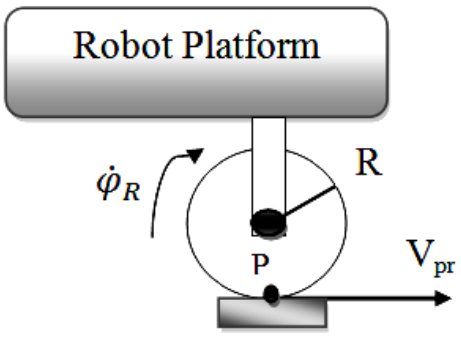


Figure II.2. Pure Rolling Motion Constraint.

Using the contact points velocities from equation (x,y) and substituting in (x ,y), the three constraint equations can be written in the following matrix form:

$$\Lambda(q)\dot{q} = 0 \tag{II.10}$$

Where

$$\Lambda(q) = \begin{bmatrix} -\sin\theta & \cos\theta & 0 & 0 & 0 \\ \cos\theta & \sin\theta & L & -R & 0 \\ \cos\theta & \sin\theta & -L & 0 & -R \end{bmatrix} \tag{II.11}$$

and

$$\dot{q} = \begin{bmatrix} \dot{x}_a & \dot{y}_a & \dot{\theta} & \dot{\varphi}_R & \dot{\varphi}_L \end{bmatrix}^T \tag{II.12}$$

$$\begin{cases} v_R = R\dot{\varphi}_R \\ v_L = R\dot{\varphi}_R \end{cases} \tag{II.13}$$

The above constraints matrix $\Lambda(q)$ will be used in the next section for the DDMR dynamic modeling..The linear velocity of each driving wheel in the Robot Frame is therefore, the linear velocity of the DDMR in the Robot Frame is the average of the linear velocities of the two wheels

$$v = \frac{v_R + v_L}{2} = R\frac{(\dot{\varphi}_R + \dot{\varphi}_L)}{2} \tag{II.14}$$

and the angular velocity of the DDMR is

$$\omega = \frac{v_R + v_L}{2L} = R\frac{(\dot{\varphi}_R + \dot{\varphi}_L)}{2} \tag{II.15}$$

The DDMRs velocities in the robot frame can now be represented in terms of the center-point A velocities in the robot frame as follows:

$$\begin{cases} \dot{x}_a^r = R\frac{(\dot{\varphi}_R + \dot{\varphi}_L)}{2} \\ \dot{y}_a^r = 0 \\ \dot{\theta} = \omega = R\frac{(\dot{\varphi}_R + \dot{\varphi}_L)}{2L} \end{cases} \tag{II.16}$$

Thus

$$\begin{bmatrix} \dot{x}_a^r \\ \dot{y}_a^r \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{R}{2} & \frac{R}{2} \\ 0 & 0 \\ \frac{R}{2L} & -\frac{R}{2L} \end{bmatrix} \begin{bmatrix} \dot{\varphi}_R \\ \dot{\varphi}_L \end{bmatrix} \tag{II.17}$$

The DDMR velocities can be obtained also in the inertial frame as follows:

$$\dot{q}^I = \begin{bmatrix} \dot{x}_a^r \\ \dot{y}_a^r \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{R}{2}\cos\theta & \frac{R}{2}\cos\theta \\ \frac{R}{2}\sin\theta & \frac{R}{2}\sin\theta \\ \frac{R}{2L} & -\frac{R}{2L} \end{bmatrix} \begin{bmatrix} \dot{\varphi}_R \\ \dot{\varphi}_L \end{bmatrix} \tag{II.18}$$

Equation (II.18) represents the forward kinematic model of the DDMR [79]. Another alternative form for the kinematic model can be obtained by representing the DDMR velocities in terms of the linear and angular velocities of DDMR in the Robot frame.

$$\dot{q}^I = \begin{bmatrix} \dot{x}_a^r \\ \dot{y}_a^r \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{II.19}$$

II.2.2 Dynamical model

Dynamics is the study of the motion of a mechanical system taking into consideration the different forces that affect its motion unlike kinematics where the forces are not taken into consideration. The dynamic model of the DDMR is essential for simulation analysis of the DDMR motion and for the design of various motion control algorithms.

A non-holonomic DDMR with n generalized coordinates $q_1, q_2, \ldots, q_n$ and subject to m constraints can be described by the following equations of motion [80]:

$$M(q)\ddot{q} + V(q,\dot{q})\dot{q} + F(\dot{q}) + G(q) + \tau_d = B(q)\tau - \mathrm{A}^{\mathrm{T}}(\mathrm{q})\lambda \tag{II.20}$$

where

$M(q)$ an nxn symmetric positive definite inertia matrix, $V(q,\dot{q})$ is the centripetal and coriolis matrix, $F(\dot{q})$ is the surface friction matrix, $G(q)$ is the gravitational vector, τ_d is the vector of bounded unknown disturbances including unstructured unmodeled dynamics, $B(q)$ is the input matrix, τ is the input vector, $\Lambda^T(q)$ is the matrix associated with the kinematic constraints, and λ is the Lagrange multipliers vector [80].

Lagrange dynamic approach

Lagrange dynamic approach is a very powerful method for formulating the equations of motion of mechanical systems. This method, which was introduced by Lagrange, is used to systematically derive the equations of motion by considering the kinetic and potential energies of the given system. The Lagrange equation can be written in the following form [80]:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) + \frac{\partial L}{\partial q_i} = F - \Lambda^T(q)\lambda \tag{II.21}$$

Where $L=T-V$ is the Lagrangian function, T, is the kinetic energy of the system, V is the potential energy of the system, q_i are the generalized coordinates, F is the generalized force vector, Λ is the constraints matrix, and λ is the vector of Lagrange multipliers associated with the constraints.

The first step in deriving the dynamic model using the Lagrange approach is to find the kinetic and potential energies that govern the motion of the DDMR. Furthermore, since the DDMR is moving in the X_I-Y_I plane, the potential energy of the DDMR is considered to be zero.

For the DDMR, the generalized coordinates are selected as

$$q = [x_a \quad y_a \quad \theta \quad \varphi_R \quad \varphi_L]^T \tag{II.22}$$

The kinetic energies of the DDMR is the sum of the kinetic energy of the robot platform without wheels plus the kinetic energies of the wheels and actuators. The kinetic energy of the robot platform is

$$T_c = \frac{1}{2} m_c v_c^2 + \frac{1}{2} I_c \dot{\theta}^2 \tag{II.23}$$

While the kinetic energy of the right and left wheel is

$$T_{wR} = \frac{1}{2} m_w v_{wR}^2 + \frac{1}{2} I_m \dot{\theta}^2 + \frac{1}{2} I_w \dot{\varphi}_R^2 \tag{II.24}$$

$$T_{wL} = \frac{1}{2} m_w v_{wL}^2 + \frac{1}{2} I_m \dot{\theta}^2 + \frac{1}{2} I_w \dot{\varphi}_L^2 \tag{II.25}$$

where, *mc* is the mass of the DDMR without the driving wheels and actuators (DC motors), *mw* is the mass of each driving wheel (with actuator), I_c is the moment of inertia of the DDMR about the vertical axis through the center of mass, I_w is the moment of inertia of each driving wheel with a motor about the wheel axis, and I_m is the moment of inertia of each driving wheel with a motor about the wheel diameter.

All velocities will be first expressed as a function of the generalized coordinates using the general velocity equation in the inertial frame.

$$v_i^2 = \dot{x}_i^2 + \dot{y}_i^2 \tag{II.26}$$

The X_i and Y_i components of the center of mass and wheels can be obtained in terms of the generalized coordinates as follow

$$\begin{cases} x_c = x_a + d\cos\theta \\ y_c = y_a + d\sin\theta \end{cases} \tag{II.27}$$

$$\begin{cases} x_{wR} = x_a + L\sin\theta \\ y_{wR} = y_a + L\cos\theta \end{cases} \tag{II.28}$$

$$\begin{cases} x_{wL} = x_a - L\sin\theta \\ y_{wL} = y_a + L\cos\theta \end{cases} \tag{II.29}$$

Using equations (II.23-25) along with equations (II.26-29), the total kinetic energy of the DDMR is

$$T = \frac{1}{2}m(\dot{x}_a^2 + \dot{y}_a^2) - m_c d\dot{\theta}(\dot{y}_a\cos\theta - \dot{x}_a\sin\theta) + \frac{1}{2}I_w(\dot{\varphi}_R^2 + \dot{\varphi}_L^2) + \frac{1}{2}I\dot{\theta}^2 \tag{II.30}$$

where the following new parameters are introduced $m = m_c + 2m_w$ is the total mass of the robot, $I = I_c + m_c d^2 + 2m_w L^2 + 2I_m$ and is the total equivalent inertia.

Using equation (II.21) along with the Lagrangian function, L=T the equations of motion of the DDMR are given by

$$m\ddot{x}_a - md\ddot{\theta}\sin\theta - md\ddot{\theta}^2\cos\theta = C_1 \tag{II.31}$$

$$m\ddot{y}_a - md\ddot{\theta}\cos\theta - md\dot{\theta}^2\sin\theta = C_2 \tag{II.32}$$

$$I\ddot{\theta} - md\ddot{x}_a\sin\theta + md\ddot{y}_a\cos\theta = C_3 \tag{II.33}$$

$$I_w\ddot{\varphi}_R = \tau_R + C_4 \tag{II.34}$$

$$I_w\ddot{\varphi}_L = \tau_L + C_5 \tag{II.35}$$

where (C_1, C_2, C_3, C_4, C_5), are coefficients related to the kinematic constraints, which can be written in terms of the Lagrange multipliers vector λ and the kinematic constraints matrix Λ introduced in section 3.

$$\Lambda^T(q) = \begin{bmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \end{bmatrix} \tag{II.36}$$

Now, the obtained equations of motion (II.31-35) can be represented in the general form given by equation (II.20) as

$$M(q)\ddot{q} + V(q,q)\dot{q} = B(q)\dot{\tau} - \Lambda^T(q)^{\lambda} \tag{II.37}$$

Where

$$M(q)=\begin{bmatrix} m & 0 & -md\sin\theta & 0 & 0 \\ 0 & m & md\cos\theta & 0 & 0 \\ -md\sin\theta & md\cos\theta & I & 0 & 0 \\ 0 & 0 & 0 & 0 & \\ 0 & 0 & 0 & 0 & I_w \end{bmatrix}, \qquad V(q,\dot{q})=\begin{bmatrix} 0 & -md\,\dot{\theta}\cos\theta & 0 & 0 & 0 \\ 0 & -md\,\dot{\theta}\sin\theta & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B(q)=\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 1 \\ 0 & 1 \end{bmatrix}, \quad \text{and} \quad \Lambda^T(q)\lambda=\begin{bmatrix} -\sin\theta & \cos\theta & \cos\theta \\ \cos\theta & \sin\theta & \sin\theta \\ 0 & L & -L \\ 0 & -R & 0 \\ 0 & 0 & -R \end{bmatrix}\times\begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \\ \lambda_5 \end{bmatrix}$$

Next, the system described by equation (II.37) is transformed into an alternative form which is more convenient for the purpose of control and simulation [79,80]. The main aim is to eliminate the constraint term $\Lambda^{T}(q)\lambda$ in equation (II.37) since the Lagrange multipliers λ_i are unknown. This is done first by defining the reduced vector

$$\eta=\begin{bmatrix} \dot{\varphi}_R \\ \dot{\varphi}_L \end{bmatrix} \tag{II.38}$$

Next, by expressing the generalized coordinates velocities using the forward kinematic model (19). Then we have

$$\begin{bmatrix} \dot{x}_a \\ \dot{y}_a \\ \dot{\theta} \\ \dot{\varphi}_R \\ \dot{\varphi}_L \end{bmatrix}=\frac{1}{2}\begin{bmatrix} R\cos\theta & R\cos\theta \\ R\sin\theta & R\sin\theta \\ \frac{R}{L} & -\frac{R}{L} \\ 2 & 0 \\ 0 & 2 \end{bmatrix}\begin{bmatrix} \dot{\varphi}_R \\ \dot{\varphi}_L \end{bmatrix} \tag{II.39}$$

This can be written in the form

$$\dot{q}=S(q)\eta \tag{II.40}$$

It can be verified that the transformation matrix $S(q)$ is in the null space of the constraint matrix $\Lambda(q)$. Therefore we have

$$S^T(q)\Lambda^T(q)=0 \tag{II.41}$$

Next, taking the time derivative of equation (II.40) gives

$$\ddot{q}=\dot{S}(q)\eta+S(q)\dot{\eta} \tag{II.42}$$

Substituting equations (II.40) and (II.42) in the main equation (II.37) we obtain

$$M(q)\left[\dot{S}(q)\eta + S(q)\dot{\eta}\right] + V(q,\dot{q})[S(q)\eta] = B(q)\tau - \Lambda^{T}(q)\lambda \quad \text{(II.43)}$$

Next, rearranging the equation and multiplying both sides by leads to

$$S^{T}(q)M(q)S(q)\dot{\eta} + S^{T}(q)[M(q)\dot{S}(q) + V(q,q)S(q)]\eta = S^{T}(q)\dot{B}(q)\tau - S^{T}(q)\Lambda^{T}(q)\lambda \quad \text{(II.44)}$$

where the last term is identically zero. Now defining the new matrices

$$\bar{M}(q) = S^{T}(q)\, M(q)S(q),$$

$$\bar{V} = S^{T}(q)M(q)\dot{S}(q) + S^{T}(q)V(q,\dot{q})S(q),$$

$$\bar{B} = S^{T}(q)B(q)$$

The dynamic equations are reduced to the form

$$\bar{M}(q)\dot{\eta} + \bar{V}(q,\dot{q})\eta = \bar{B}(q)\tau \quad \text{(II.45)}$$

Where

$$\bar{M}(q) = \begin{bmatrix} I_w + \frac{R^2}{4L^2}(mL^2 + I) & \frac{R^2}{4L^2}(mL^2 - I) \\ \frac{R^2}{4L^2}(mL^2 - I) & I_w + \frac{R^2}{4L^2}(mL^2 + I) \end{bmatrix}$$

$$\bar{V}(q,\dot{q}) = \begin{bmatrix} 0 & \frac{R^2}{2L}m_c d\dot{\theta} \\ -\frac{R^2}{2L}m_c d\dot{\theta} & 0 \end{bmatrix}, \quad \bar{B}(q) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Equation (II.45) shows that the DDMR dynamics are expressed only as a function of the right and left wheel angular velocities $(\dot{\varphi}_R, \dot{\varphi}_L)$, the robot angular velocity $\dot{\theta}$ and the driving motor torques (τ_R, τ_L) .The equations of motion (II.45) can be also transformed into an alternative form which is represented by the linear and angular velocities (v,w)of the DDMR. Using the kinematic model equations (II.14) and (II.15), it can be easily shown that the model equations (II.45) can be rearranged in the following compact form

$$\begin{cases} \left(m + \frac{2I_w}{R^2}\right)\dot{v} - m_c d\omega^2 = \frac{1}{R}(\tau_R + \tau_L) \\ \left(I + \frac{2L^2}{R^2}I_w\right)\dot{\omega} + m_c d\omega v = \frac{L}{R}(\tau_R - \tau_L) \end{cases} \quad \text{(II.46)}$$

II.2.3. Actuator modeling

The job of the electrical actuator is to drive the mechanical system of a robot. In this work, two pairs of DC motor are used in order to guide the wheels there a determined motion. Where Fig II.3. represents the equivalent electrical schema of the DC motor.

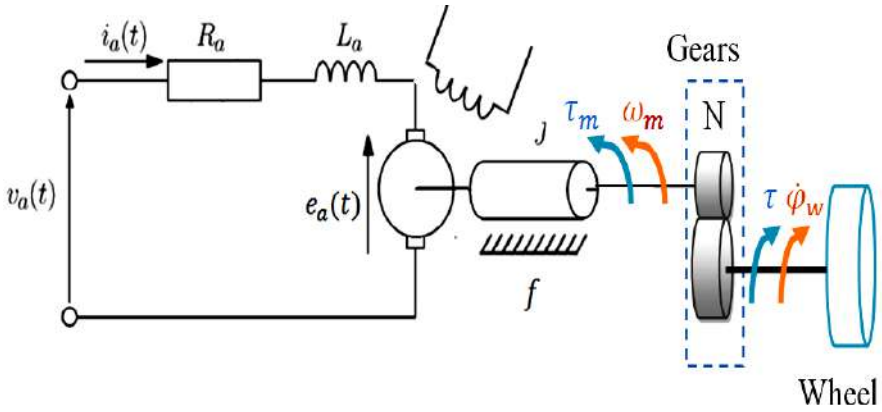


Figure II.3. Equivalent electrical schema of the dc motor.

The dynamic model of the actuators can be represented as [42]:

$$\begin{cases} v_a = R_a i_a(t) + L_a \frac{di_a(t)}{dt} + e_a(t) \\ e_a(t) = K_b \omega_m(t) \\ \tau_m = j \frac{dw_m(t)}{dt} + f w_m(t) + K_t i_a(t) \\ \tau = N \tau_m \end{cases} \quad \text{(II.47)}$$

where, i_a is the armature current, (R_a, L_a) is the resistance and inductance of the armature winding, respectively, e_a is the back emf, ω_m is the rotor angular speed, τ_m is the motor torque, (K_t, K_b) are the torque constant and back emf constant, respectively, N is the gear ratio, and τ is the output torque applied to the wheel Since the robot motors are mechanically coupled to wheels through the gears. Therefore, each dc motor will have:

$$\begin{cases} \omega_{mR} = N\dot{\varphi}_{wR} \quad and \quad \tau_R = N\tau_{mR} \\ \omega_{mL} = N\dot{\varphi}_{wL} \quad and \quad \tau_L = N\tau_{mL} \end{cases} \quad \text{(II.48)}$$

For the two motors, the dynamic model is expressed as:

$$\begin{cases} \frac{1}{(R+L_aP)}(e_{ar} - K_b N\dot{\varphi}_{wR}) = i_R \ \ with \ \ \tau_R = Nk_t i_R \\ \frac{1}{(R+L_aP)}(e_{al} - K_b N\dot{\varphi}_{wL}) = i_L \ \ with \ \ \tau_L = Nk_t i_L \end{cases} \quad \text{(II.49)}$$

The robot physical parameters are provided in Table II.1.[42]:

Table II.1. Physical parameter of the robot

Parameters	Value and Unit
Distance between two wheels, 2 L	0.75 m
Distance of point Pc from point Po, D	0.3 m
Driving wheels radius, R	0.15 m
Mass of the mobile robot without the driving wheels and DC motors, m_c	30 kg
Mass of each driving wheel (with actuator), m_w	1 kg
Moment of inertia of the mobile robot about the vertical axis through the center of mass, I_c	15.625 kg·m^2
Moment of inertia of each driving wheel with a motor about the Wheel axis, I_w	0.005 kg·m^2
Moment of inertia of each driving wheel with a motor about the Wheel diameter, I_m	0.0025 kg·m^2
Armature winding resistance, R_a	1.6 Ω
Armature winding inductance, L_a	0.048 H
Torque constant, K_t	0.2613 N·m/A
Back emf constant, K_b	0.19 rad/s
Gear ratio, N	62.55

II.3. The Backstepping controller design using the nonlinear feedback method

Fig. II.4. reveals the entire control strategy based on the WMR dynamics and kinematics. The role of trajectory generator is to develop desired x, y and θ coordinate, where a back-stepping kinematic controller, placed in the external loop, measures the difference between the desired coordinate and the actual value taken from the robot, after that the kinematic controller produce it own linear and angular speed, this control signals are taken to the internal loop, where nonlinear feed back controller deals with dynamics of the WMR.

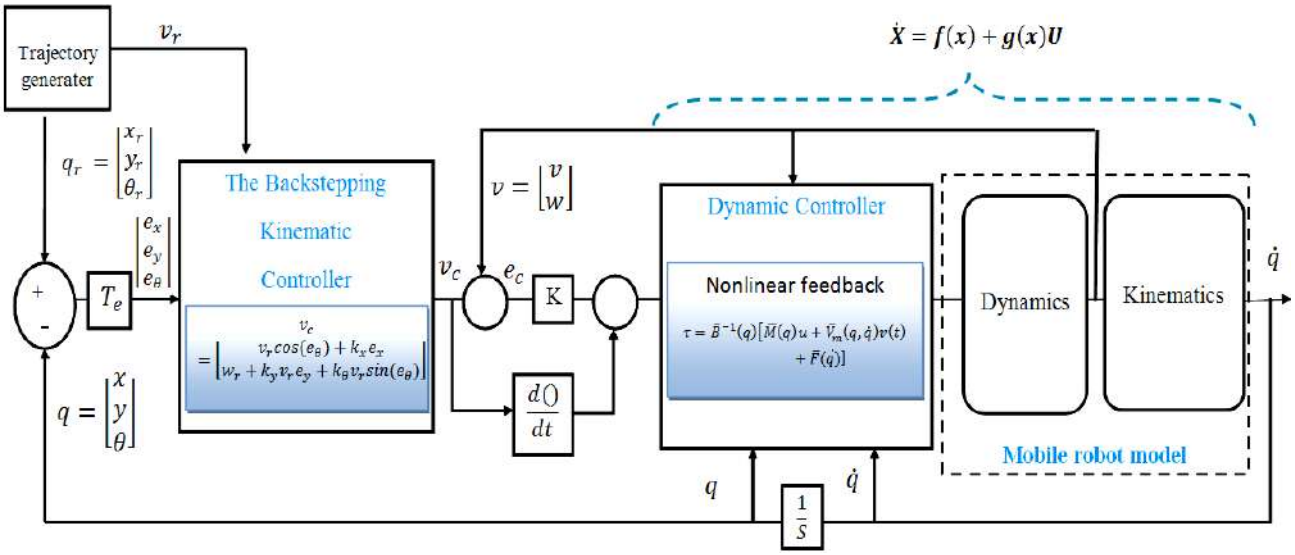


Figure II.4 General Control scheme of the WMR.

II.3.1. Kinematic Controller Design

In order to reach at a nearly zero distance error, a kinematic controller is required. The major advantage of the back-stepping approach is its practical and simplicity structure, since it will depend only on the kinematic model. Moreover, it has been classified as a stable tracking control law, where other techniques, such as nonlinear feedbacks and sliding-mode techniques, demand the knowledge of the dynamic model and their hardware realisation is difficult to implement. For that reason, this controller makes it adopted in our study [40]. The controller structure is introduced in Fig. 3, where the input error and velocity vector (vc) are:

$$\begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix} = \begin{bmatrix} cos\,\theta & sin\,\theta & 0 \\ -sin\,\theta & cos\,\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_r - & X \\ Y_r - & Y \\ \theta_r - & \theta \end{bmatrix} = T_e e_r \tag{II.50}$$

$$v_c = \begin{bmatrix} v_r cos(e_\theta) + k_x e_x \\ w_r + k_y v_r e_y + k_\theta v_r sin(e_\theta) \end{bmatrix} \tag{II.51}$$

Where k_x, k_y and k_θ are tuning parameters.

II.3.2.Dynamic Controller

Feedback linearization is an approach to nonlinear control design which has attracted a great deal of research interest in recent years. The central idea of this approach is to first transform a nonlinear system into a (fully or partially) linear system and then use the well known and powerful linear design techniques to complete the control design. The application of this method to the case of the nonholonomic mobile robot is that by applying an appropriate nonlinear feedback we can linearize the nonlinear dynamics and then apply the kinematic based controller to the linear system. Let u be an auxiliary input, then by applying the nonlinear feedback [81]:

$$\tau = \bar{B}^{-1}(q)\left[\bar{M}(q)u + \bar{V}_m(q,\dot{q})v(t) + \bar{F}(\dot{q})\right] \tag{II.52}$$

We can convert the dynamic equation (II.20) to:

$$\bar{M}(q)\dot{v}(t) + \bar{V}_m(q,\dot{q})v(t) + \bar{F}(\dot{q}) + \bar{G}(q) + \bar{\tau}_d = \bar{B}(q)\tau\left[\bar{B}^{-1}(q)[\bar{M}(q)u + \bar{V}_m(q,\dot{q})v(t) + \bar{F}(\dot{q})]\right]$$
$$\dot{v}(t) = u$$

(II.53)

Therefore the complete equations of system will be:

$$\begin{matrix}\dot{q} = S(q)v(t) \\ \dot{v}(t) = u\end{matrix} \quad \text{(II.54)}$$

In performing the nonlinear feedback (II.52) it is assumed that all the dynamical quantities such as $\bar{M}(q), \bar{V}_m(q,\dot{q})$ and $\bar{F}(\dot{q})$ are exactly known and $\bar{\tau}_d = 0$. Although a nonlinear system can be controllable, a stabilizable smooth state feedback may not exist. This is the case of the system (II.54) where the equilibrium point cannot be made asymptotically stable by any smooth time-invariant state feedback controller. A backstepping controller is used to stabilize system (II.54). Consider a system given by:

$$\begin{matrix}\dot{x} = f(x) + g(x)\xi \\ \dot{\xi} = u\end{matrix} \quad \text{(II.55)}$$

With *f(0)=0* and *f* and *g* are smooth functions. Designing a feedback control to stabilize the system at *(x=0, ξ=0)* is called a backstepping controller. Comparing equations (II.55)and (II.54) one can find out that the linearized nonholonomic mobile robot equations are in the form of (II.55) and a backstepping control algorithm will make it stable. The proposed control input to stabilize system (II.54) is as follows:

$$u = \dot{v}_c + K_4(v_c - v) \quad \text{(II.56)}$$

Where:

$$v_c = \begin{matrix}\begin{bmatrix} v_r\cos e_\theta + K_x e_x \\ \omega_r + K_y v_r e_y + K_\theta v_r \sin e_\theta \end{bmatrix} \\ v_c = f(e_p, v_r, K) \\ K = (K_x, K_y, K_\theta)\end{matrix} \quad \text{(II.57)}$$

$$\dot{v}_c = \begin{bmatrix} \dot{v}_r\cos e_\theta - v_r\dot{e}_\theta \sin e_\theta + K_x\dot{e}_x \\ \dot{\omega}_r + K_y\dot{v}_r\dot{e}_y + K_\theta\dot{v}_r\sin e_\theta + K_\theta v_r\dot{e}_\theta\cos e_\theta \end{bmatrix} \quad \text{(II.58)}$$

$$v_c = \begin{bmatrix} \dot{v}_r\cos e_\theta \\ \dot{\omega}_r + K_y\dot{v}_r e_y + K_\theta\dot{v}_r\sin e_\theta \end{bmatrix} + \begin{bmatrix} K_x & 0 & -v_r\sin e_\theta \\ 0 & K_y v_r & K_\theta v_r\cos e_\theta \end{bmatrix}\dot{e}_p \quad \text{(II.59)}$$

$$e_p = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} x_r - x \\ y_r - y \\ \theta_r - \theta \end{bmatrix} = \begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix} = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} \quad \text{(II.60)}$$

K_4 is a positive definite, diagonal matrix given by:

$$K_4 = K_4 I \quad \text{(II.61)}$$

The stability of the above control algorithm (II.56) has been already proved in references [81,82] .

II.4 Simulation Results

In this part, two desired trajectories are made as inputs for the implemented model and control system to investigate the trajectory tracking error. This examines the capacity of tracking continuous and non-continuous gradients.

II.4.1 Square Path

The reference and actual trajectories of the square path are shown in Fig. II.5, highlighted in blue and red lines, respectively, Fig. II.5 (a)-(d) display the errors of x, y, θ, and the tracking trajectory, respectively. From Fig. II.6 (d), the minimum tracking error is about 0.07 meter. This is an important error that needs to be minimized, for that reason a new tracked control method is necessary to handle the sharp shaped trajectory problem.

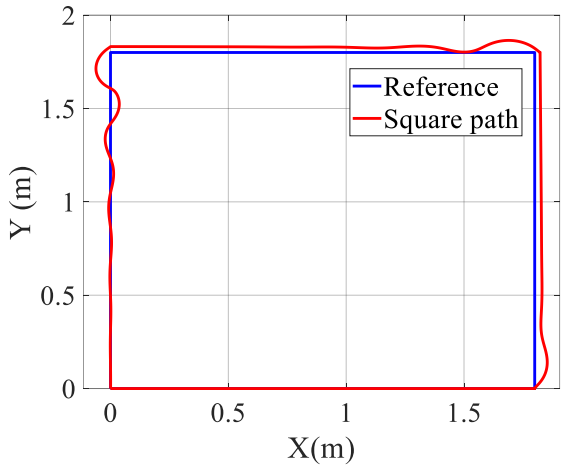


Fig II.5. Square path in XY plane

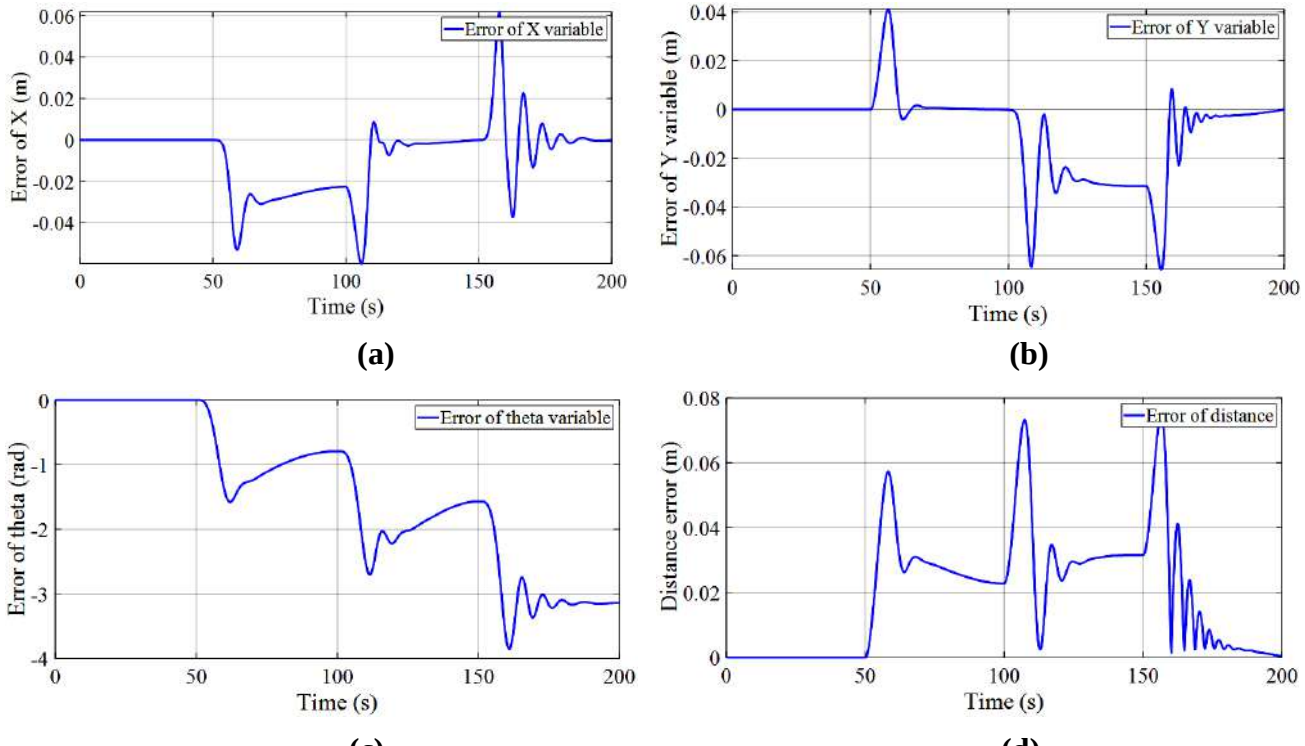


Fig. II.6. Error corresponding to (a) x,(b) y; (c) the theta angle ,(d) the trajectory tracking.

II.4.2 Circular Path

The reference and actual trajectories of the circular trajectory are shown in Fig.II.7, highlighted in blue and red lines, respectively, Fig. II.8 (a)-(d) display the errors of x, y, θ, and the tracking trajectory, respectively. From Fig. II.8 (d), the minimum tracking error is about 0.05 meter. Which is high too, for that reason we can say that anew tracked control method is needed the smooth trajectory too.

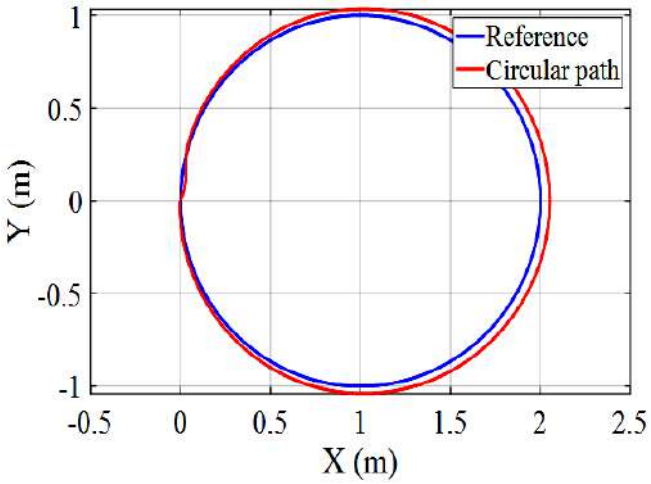


Fig II.7. Circular path in XY plane

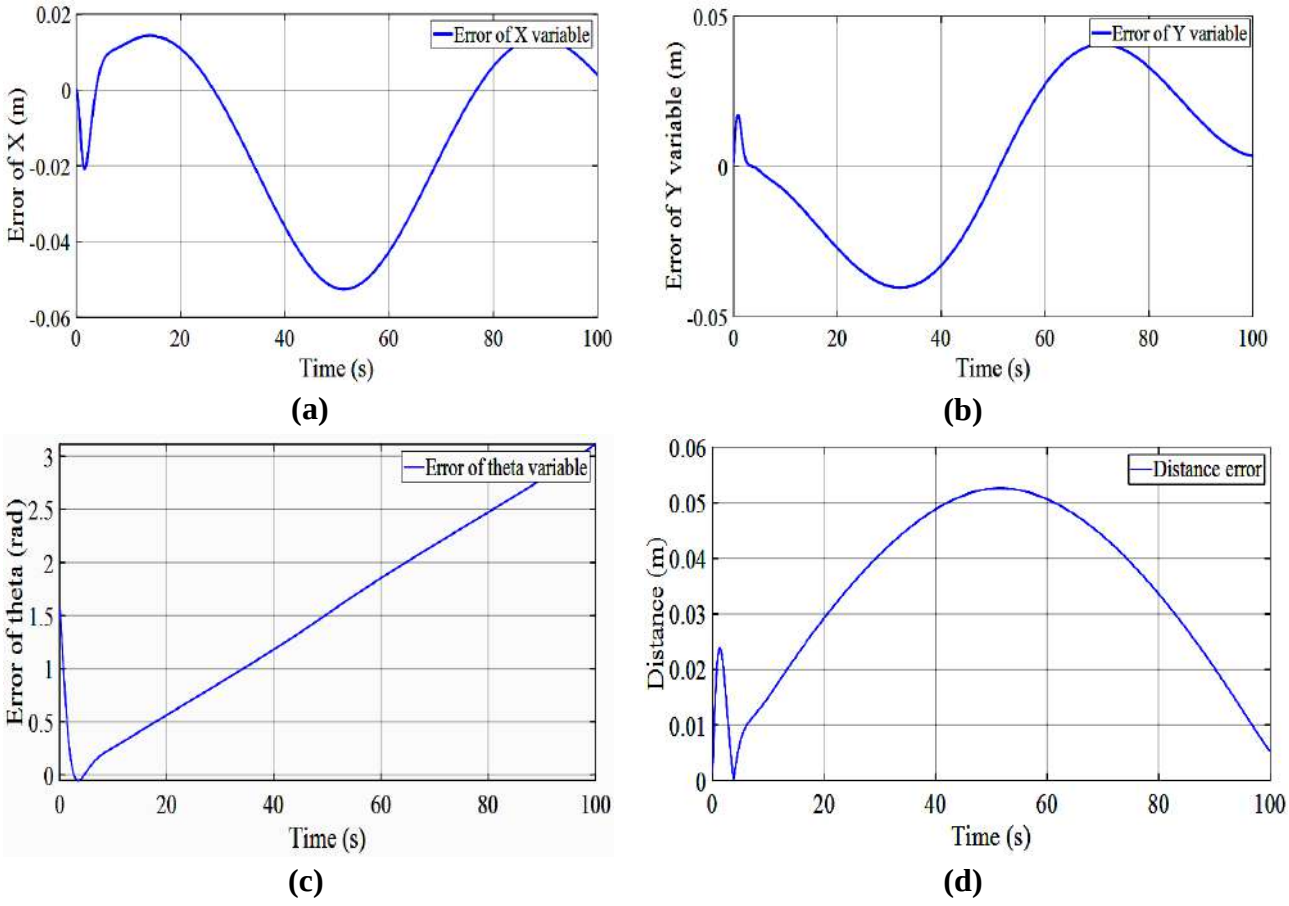


Fig. II.8. Error corresponding to (a) x,(b) y; (c) the theta angle ,(d) the trajectory tracking

II.5 Conclusion

This chapter deals with three essential factors for the modeling of the WMR, kinematic, dynamic and actuators. Each factor is carefully studied and modeled. A traditional control approach based on the combination of a backstepping controller with the nonlinear feedback dynamic controller has been studied. The simulation results are handled based on two specific trajectories that contain continuous and non-continuous gradient directions. Simulation results demonstrate that the trajectory tracking error is still extremely observed in both continuous and non-continuous gradient paths. The results demonstrate that a new control methodology is needed to improve trajectory-tracking error. The new proposed techniques will be discussed in the next chapters.

Chapter III:

Optimal PID and ANFIS-PID controller design

Chapter III Optimal PID and ANFIS-PID controller design

III.1.Introduction

This chapter suggests a novel control technique that merge the advantage of the adaptive Neuro- Fuzzy inference system (ANFIS) with the proportional integral derivative (PID) controller, abbreviated as (ANFIS-PID),for dealing with the dynamics of the of wheeled mobile robot (WMR). Tuning parameters of the proposed hybrid controller is a challenging task, a comparative study of various meta-heuristic optimization algorithms has been made. Results in MATLAB–Simulink environment revealed the highest efficiency of the proposed ANFIS-PID technique compared to seven designed PID controllers, in terms of settling and rise time, overshoot, and the integral square error (ISE) as a cost function. Various cases, study (circular path, diamond path, zigzag path) have highlighted the over performance of mentioned controller. Moreover, this hybrid technique is fused with back-stepping approach for the kinematic control. A lemniscate and a square trajectory were performed to clarify the capability of the mentioned controller

III.2.The general control Strategy Design

Fig.III.1 explains the simplified form of the general control concept based of the WMR dynamics and kinematics. The role of path generator is to supplies the desired variable x, y and θ controller.

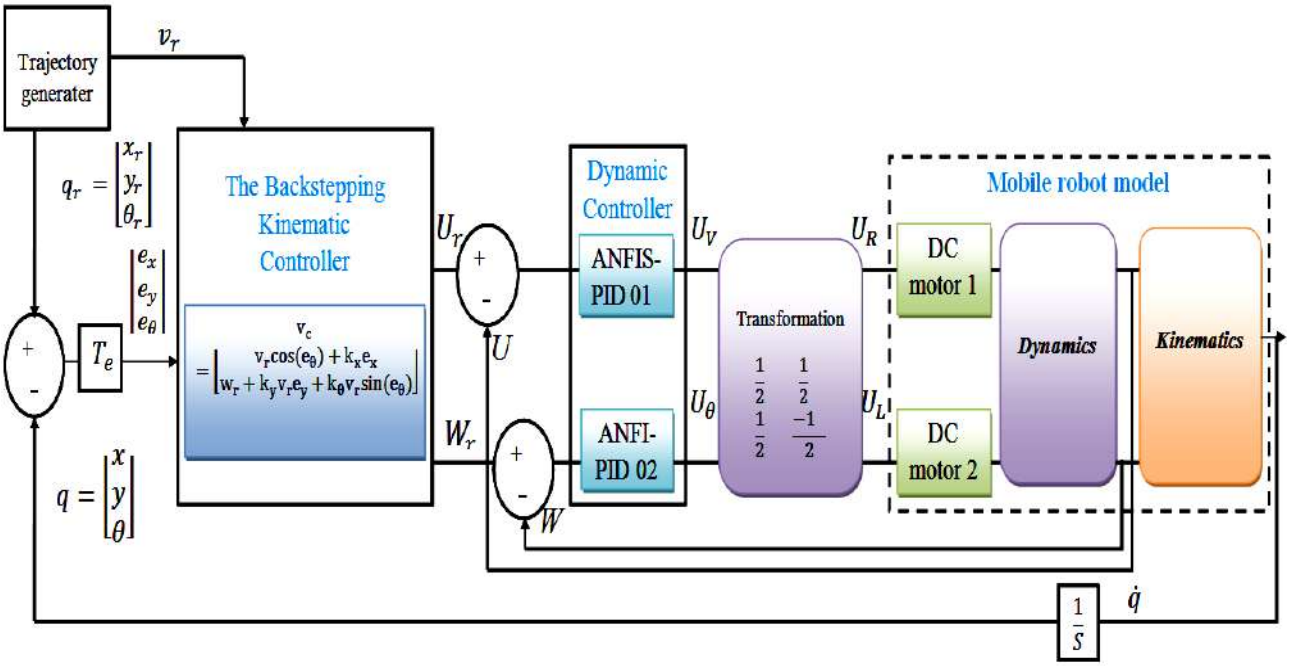


Figure III.1 General Control scheme of the WMR.

A back-stepping act as a kinematic controller, located in the external loop, this controller measures the difference between the recommended signals (taken from the bloc trajectory generator) and real value from the robot, at next realizes it own linear speed and angle which they are passed to the internal loop, where two ANFIS-PID controller deal with the dynamics. Here, the two dynamic controllers give their own angular and linear speeds based on the data taken from the kinematic

III.2.1. Kinematic Controller Design

In order to reach at a nearly zero distance error, a kinematic controller is required. The major advantage of the back-stepping approach is its practical and simplicity structure, since it will depend only on the kinematic model. Moreover, it has been classified as a stable tracking control law, where other techniques, such as nonlinear feedbacks and sliding-mode techniques, demand the knowledge of the dynamic model and their hardware realisation is difficult to implement. For that reason, this controller makes it adopted in our study [40]. The controller structure is introduced in Fig.III.2, where the input error and velocity vector (vc) are:

$$\begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix} = \begin{bmatrix} cos\,\theta & sin\,\theta & 0 \\ -sin\,\theta & cos\,\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_r - & X \\ Y_r - & Y \\ \theta_r - & \theta \end{bmatrix} = T_e e_r \tag{III.1}$$

$$v_c = \begin{bmatrix} v_r cos(e_\theta) + k_x e_x \\ w_r + k_y v_r e_y + k_\theta v_r sin(e_\theta) \end{bmatrix} \tag{III.2}$$

Where k_x, k_y and k_θ are tuning parameters.

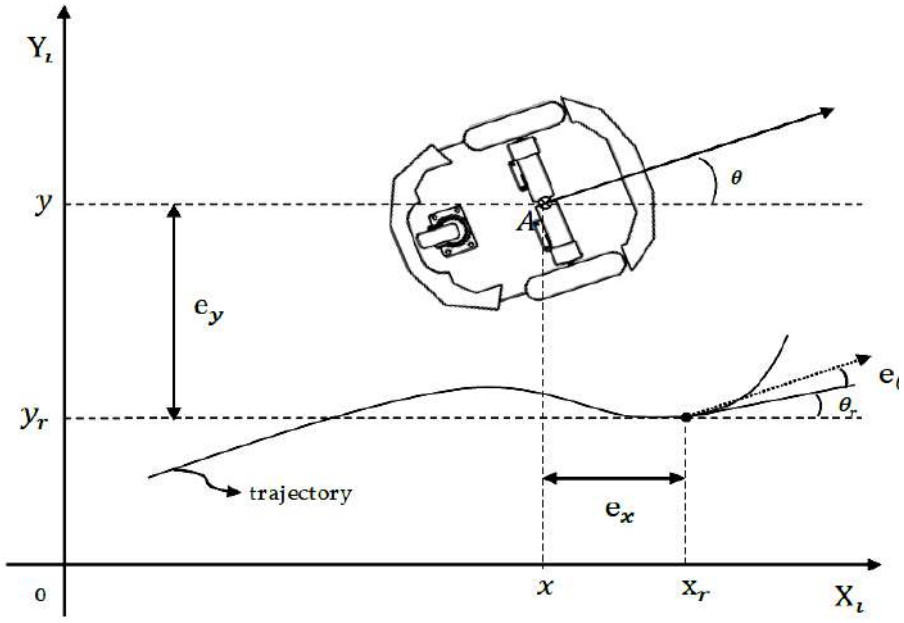


Figure III.2 Trajectory tracking errors

Lyapunov stability analysis

The Lyapunov stability analysis of the control rule in equation is described as follows [15,81,82]:

Lemma 1:

From Equation (III.1):

$$\begin{bmatrix}\dot{e}_x\\ \dot{e}_y\\ \dot{e}_\theta\end{bmatrix} = \dot{e}_p = f(t, e_p) = \begin{bmatrix}\omega e_y - v + v_r \cos e_\theta\\ -\omega e_x + v_r \sin e_\theta\\ \omega_r - \omega\end{bmatrix} \quad \text{(III.3)}$$

This result can be proved by using Equation (III.1) and Equation (III.2):

$$\begin{bmatrix}e_x\\ e_y\\ e_\theta\end{bmatrix} = \begin{bmatrix}\cos\theta & \sin\theta & 0\\ -\sin\theta & \cos\theta & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}x_r - x\\ y_r - y\\ \theta_r - \theta\end{bmatrix} = \begin{bmatrix}(x_r - x)\cos\theta + (y_r - y)\sin\theta\\ -(x_r - x)\sin\theta + (y_r - y)\cos\theta\\ \theta_r - \theta\end{bmatrix} \quad \text{(III.4)}$$

$$\dot{e}_x = (\dot{x}_r - \dot{x})\cos\theta + (\dot{y}_r - \dot{y})\sin\theta - (x_r - x)\dot{\theta}\, sin\,\theta + (y_r - y)\dot{\theta}\cos\theta = e_y\omega - v +$$
$$\dot{x}_r \cos(\theta_r - \theta) + \dot{y}_r \sin(\theta_r - \theta) = e_y\omega - v + v_r \cos e_\theta \quad \text{(III.5)}$$

$$\dot{e}_x = -(\dot{x}_r - \dot{x})\, sin\,\theta + (\dot{y}_r - \dot{y})\, cos\,\theta - (x_r - x)\dot{\theta}\, cos\,\theta + (y_r - y)\dot{\theta}\, sin\,\theta = -e_x\omega -$$
$$\dot{x}_r\, sin(\theta_r - \theta) + \dot{y}_r\, cos(\theta_r - \theta) = -e_x\omega + v_r \sin e_\theta \quad \text{(III.6)}$$

$$\dot{e}_\theta = \dot{\theta}_r - \dot{\theta} = \omega_r - \omega \quad \text{(III.7)}$$

Substituting v and ω by $v(e_p, q_r)$ and $\omega(e_p, q_r)$ Lemma 1 was obtained.

Lemma 2:

$$\begin{bmatrix}\dot{e}_x\\ \dot{e}_y\\ \dot{e}_\theta\end{bmatrix} = \dot{e}_p = f(t, e_p) = \begin{bmatrix}\left(\omega_r - v_r\left(K_y e_y + K_\theta \sin e_\theta\right)\right) e_y + K_x e_x\\ -(\omega_r - v_r\left(\left(K_y e_y + K_\theta\, sin\, e_\theta\right)\right) e_x + v_r \sin e_\theta\\ -v_r(K_y e_y + K_\theta\, sin\, e_\theta)\end{bmatrix} \quad \text{(III.8)}$$

Proposition 1:

By using the in Equation (III.2), it can be said that when the reference velocity $v_r > 0$, $e_p = 0$, is a stable equilibrium.

When a scalar function V is proposed as a Lyapunov function candidate for the system:

$$V = \frac{1}{2}\left(e_x^2 + e_y^2\right) + \frac{(1 - \cos e_\theta)}{K_y} \quad \text{(III.9)}$$

Clearly, $V \geq 0$, $e_p = 0 \Rightarrow V = 0$ and $if\ e_p \neq 0 \Rightarrow V > 0$, thus the Equation (38) above is a positive definite function. By Lemma 2:

$$\dot{V} = \dot{e}_x e_x + \dot{e}_y e_y + \frac{\dot{e}_\theta \sin(e_\theta)}{K_y} = \left[\left(\omega_r - v_r\left(K_y e_y + K_\theta \sin e_\theta\right)\right) e_y + K_x e_x\right] e_x +$$

$$\left[-\left(\omega_r - v_r\left(\left(K_y e_y + K_\theta \sin e_\theta\right)\right) e_x + v_r \sin e_\theta\right] e_y + \frac{\left[-v_r(K_y e_y + K_\theta \sin e_\theta)\right] \sin e_\theta}{K_y} =$$

$$-K_x e_x^2 - \frac{v_r K_\theta \sin^2 e_\theta}{K_y} \leq 0 \qquad \text{(III.10)}$$

Consequently the Lyapunov function V's derivative is a negative definite function. That means uniformly asymptotically stability around $e_p = 0$ under the conditions that $v_r, \omega_r, K_x, K_y, K_\theta$ are bounded and v_r, ω_r are continuous.

III.2.2.The PID Dynamic Controller

The proportional integral derivative (PID) is a feedback controller that is commonly utilized in various applications, especially for robotics and automation industry due to its simplicity structure and reliability that usually leads to satisfying results [83]. Fig. III.3 presents the parallel structure of the PID controller block diagram. Where, E(S) distinguishes for the input and errors(E(S) and U(S)).

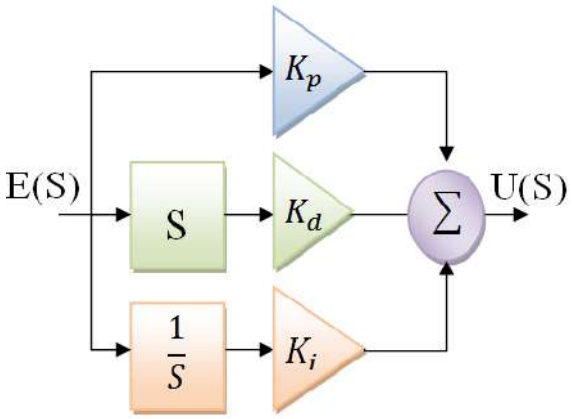


Figure III.3 Block diagram of the PID controller.

The representation of the transfer function for the PID in the time domain (Eq. (III.11)) and the transfer function in the Laplace domain (Eq. (III.12)).

$$u(t) = K_p e(t) + k_i \int_0^t e(t) + K_d \frac{de(t)}{dt} \qquad \text{(III.11)}$$

$$C(s) = \frac{U(s)}{E(s)} = K_p + \frac{K_i}{S} + K_d S \tag{III.12}$$

Where, K_p, K_i and K_d are proportional, integral and derivative gains constants, respectively,

The proportional constant kp role is to regulate the rise time value. where the purpose of the integral action ki is to eliminate steady-state error, the derivative gain kd for reducing the value of overflow and improving a transient response.

The success in designing PID controller rests to the optimal selection of parameters. The Ziegler–Nichols (Z-N) was supported in adjusting the tuning details for the PID. Today, the tuning methodology reaches a new era for soft computing approaches like the meta-heuristics algorithms. For that reason, the next section will focus on describing the tuning strategies

III.2.3. Tuning methodology for PID

The block diagram for WMR tuning strategy is highlighted in Fig.III.4. while The first PID controller obtains the contrast between the desired and real speed from the right DC motor wheel as an input, where the reference velocity is fixed at: Ur =1 m/s. The second PID controller in the left wheel examines the difference between the physical and desired directions angle as input. Here, the reference angle is a constant value: θr=0.785 rad.

The following equations (26) represent the relation between the input voltage of the right and left DC actuators, respectively U_R,U_L with the outputs of the first and second controller U_V,U_θ ,respectively.

$$\begin{cases} U_R = \dfrac{U_V + U_\theta}{2} \\ U_L = \dfrac{U_V - U_\theta}{2} \end{cases} \tag{III.13}$$

The cost function used in the study is the integral square error (ISE) which is demonstrated by equation (III.14):

$$ISE = \left(\int_0^\infty [e_v(t)]^2 dt + \int_0^\infty [e_\theta(t)]^2 dt \right) \tag{III.14}$$

where:

$$\begin{cases} e_v = U_r - U_m \\ e_\theta = \theta_r - \theta_m \end{cases} \tag{III.15}$$

U_r is the desired velocity,U_m the actual velocity, $e_v$the velocity error,θ_r the desired orientation,θ_m the measured orientation, and e_θ is the orientation error,

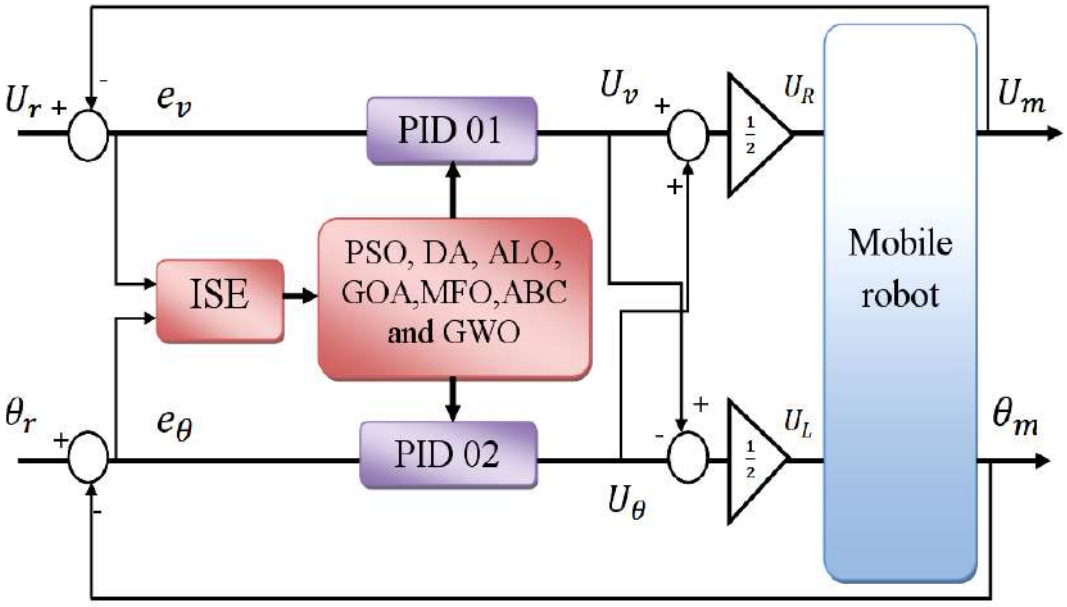


Figure III.4. Bloc diagram for tuning methodology.

The optimization algorithms used to tune the parameters (K_p, K_I, K_d) are described in the next section.

III.3. Meta-heuristics

A comparative study has been made to demonstrate the capability of the grey wolf optimizer (GWO) ,compared to particle swarm optimizer (PSO), dragonfly optimizer (DA), Ant Lion optimizer (ALO), grasshopper algorithm (GOA) , moth flame optimizer (MFO), and the artificial bee colony algorithm (ABC). Therefore, a brief description of all algorithms is given:

III.3.1. Particle Swarm Optimizer (PSO)

Each particles in the swarm update their positions according to the following equations [84,85]:

$$V_i^{k+1} = w\,V_i^k + C_1R_1\left(p_i^k + x_i^k\right) + C_2R_2\left(gbest_i + x_i^k\right) \quad \text{(III.16)}$$

$$X_i^{k+1} = X_i^k + V_i^{k+1} \quad \text{(III.17)}$$

Here, i refer to the particle in the swarm. k the iteration step carried out, w the inertia weight parameter and R_1 and R_2 are random numbers in the range [0, 1]. The coefficients C_1 and C_2 are the optimization parameters, X: position vector, and p_i^k: best position information achieved by the i^{th} particle, $gbest_i$: best position information available in the swarm .

III.3.2 .Dragonfly algorithm (DA)

This approached was first introduced by Mirjalili [86], and was inspirited by the dynamic and static behaviour of dragonflies in nature explained in [86–89]. The life cycle of dragonflies is divided into two phases: the nymph phase and the adult phase [87]. The natural behaviour of each agents in the swarm obligates the attraction to nurturing sources and distract outward enemies.

As demonstrated by Reynolds [89], the behaviour of swarm follows the following principles [86,88,89] :

- Separation

This step aims to avoid individuals' collision with their neighbours that is near to its position, the static swarm defined by the following equation:

$$S_i = \sum_{j=1}^{n} X - X_j \tag{III.18}$$

Here X and X_j are the positions of the current individual and the j^{th} neighbouring individual, respectively. n is the number of neighbouring individuals.

- Alignment

The velocity matching between dragonflies of the same group is given by:

$$A_i = \frac{\sum_{j=1}^{n} V_j}{n} - X \tag{III.19}$$

Where v_j is the velocity of neighbouring individual j.

- Cohesion

The tendency of members towards the centre of the swarm's group is known as the cohesion and it is defined as :

$$C_i = \frac{\sum_{j=1}^{n} X_j}{n} - X \tag{III.20}$$

- Attraction to food

In order to survive, all individuals must move toward the food. As presented by this mathematical formula:

$$F_i = X^+ - X \tag{III.21}$$

Where X^+ shows the position of the food source.

- Distraction from enemy

This equation represent the movement of dragonflies far away from the enemy's sources to survive:

$$E_i = X^- - X \tag{III.22}$$

Where X^- shows the position of the enemy.

The position update of each dragonfly for testing another weight solution and receiving another fitness value is gotten by calculating ΔX and X using Eq. (III.22) and Eq. (III.23) [86,88]:

$$\Delta X_i = (sS_i + aA_i + cC_i + fF_i + eE_i) + w \tag{III.23}$$

$$X_{t+1} = X_t + \Delta X_{t+1} \tag{III.24}$$

$$e = 0.1 - 1 * \left(\frac{0.1}{\frac{I}{2}}\right) \tag{III.25}$$

$$w = 0.9 - i * (\frac{0.9-0.4}{I}) \tag{III.26}$$

Where s, a, c, f, e, and w are the weights of their correspondent element. w is calculated using Eq. (III.26) here i is the current iteration and I is the number of iterations, and e is calculated in Eq. (III.25). s, a, and c are three different random numbers between 0 and 2e, f is a random number between 0 and 2. More details can be found elsewhere [86].

III.3.3. Ant Lion Optimizer (ALO)

This method proposed by Mirjalili [90] is inspired by the hunting mechanism of antlions in nature. The antlions have about three years average lifespan that it is spent as larvae except for 3-5 weeks of that period is spent in adulthood [91]. The mechanism is explained elsewhere [91,92].
The mathematical illustration is given as follows [90–92]:

- Random walks of Ants

$$X(t) = [0, cumsum(2r(t_1 - 1), cumsum(2r(t_2) - 1, \cdots, cumsum(2r(t_n) - 1)] \tag{III.27}$$

Where n is the maximum number of iterations, cumsum calculates the cumulative sum, and $t$is the step of the random walk. thus,

$$r(t) = \begin{cases} 1 & if\ rand > 0.5 \\ 0 & if\ rand < 0.5 \end{cases} \tag{III.28}$$

Here $r(t)$ is a stochastic function and rand is a random number generated with uniform distribution at interval of [0, 1].The positions of ants is saved and utilised during optimisation in the matrix:

$$M_{ant} = \begin{bmatrix} A_{1,1} & A_{1,2} & \cdots & \cdots & A_{1,d} \\ A_{2,1} & A_{2,2} & \cdots & \cdots & A_{2,d} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ A_{n,1} & A_{n,2} & \cdots & \cdots & A_{n,d} \end{bmatrix} \quad \text{(III.29)}$$

Where M_{Ant} is the matrix for saving the position of each ant, $A_{i,j}$ shows the value of the j^{th} variable of i^{th} ant, $n$is the number of ants, and d is the number of variables.Based on the random walk, each ant updates their position with each step of optimisation. The min-max normalisation equation is used to normalise the random walks:

$$X_i^t = \frac{(X_i^t - a_i) \times (d_i - c_i^t)}{(d_i^t - a_i)} + c_i \quad \text{(III.30)}$$

Here a_i is the minimum of random walk of i^{th} variable, d_i is the maximum of random walk of i^{th} variable, c_i^t is the minimum of i^{th} variable at t^{th} iteration, and d_i^t is the maximum of i^{th} variable at t^{th} iteration.

- Trapping in Antlion's pit

The random walks of ants are influenced by the antlion's trap. It is defined as:

$$\begin{aligned} c_i^t &= Antlion_j^t + c^t \\ d_i^t &= Antlion_j^t + d^t \end{aligned} \quad \text{(III.31)}$$

Where c^t represents the minimum of all variables at t^{th} iteration, d^t indicates the vector including the maximum of all variables at t^{th} iteration, c_i^t is the minimum of all variables for i^{th} ant, d_i^t is the maximum of all variables for i^{th} ant, and $Antlion_j^t$ shows the position of the selected j^{th} antlion at t^{th} iteration.

- Building trap

During optimisation the algorithm used a roulette wheel operator to choose antlions according to their fitness. This method increased the chance if catching ants.

- Sliding Ants toward Antlion

The following equations compute the decrease adaptively to the radius of the random walks hyper sphere of ants:

$$c^t = \frac{c^t}{I}$$
$$d^t = \frac{d^t}{I} \quad \text{(III.32)}$$

Where I is a ratio, c^t is the minimum of all variables at t^{th} iteration, and d^t indicates the vector, including the maximum of all variables at t^{th} iteration.

- Catching Pray and Rebuilding the Pit

In the final hunting stage, antlions update their position to improve the possibility of catching new prey according to the position of the latest hunted ant:

$$Antlion_j^t = Ant_i^t \; if \; f(Ant_i^t) > f\left(Antlion_j^t\right) \quad \text{(III.33)}$$

Here t represents the current iteration, $Antlion_j^t$ is the position of the selected j^{th} antlion at t^{th} iteration, and Ant_i^t represents the position of i^{th} ant at t^{th} iteration.

- Elitism

Through the whole iteration, the best-obtained antlion is called an elite. This elite influences the movement of the rest of the ants during iteration. So, it is considered that every prey walks randomly around a chosen individual by the roulette wheel and the elite altogether represented by the following equation:

$$Ant_i^t = \frac{R_A^t + R_E^t}{2} \quad \text{(III.34)}$$

Where R_A^t is the random walk around the antlion selected by the roulette wheel at t^{th} iteration, R_E^t is the random walk, and Ant_i^t represents the position of i^{th} ant at t^{th} iteration.

III.3.4 .Artificial Bee Colony (ABC)

The artificial bee colony algorithm was first introduced by karaboga [93] in 2005, while its behaviour [94] was analysed in 2007. The behaviours of honeybees in finding food sources, splitting the knowledge with the nest, are its main inspiration. The approached classified bees to three types (employed, onlooker, and scout) where each agent plays different roles in the process. More details can be found at [88] while the process of the ABC algorithm is the following [88]:

- Initialization

At first, a ratio of the population is randomly sprayed into the solution area. Then, the fitness value, the nectar amounts, is fixed.

- Move the Onlookers

This equation calculates the probability of selecting a food source:

$$P_i = \frac{F(\theta_i)}{\sum_{k=1}^{S} F(\theta_k)} \tag{III.35}$$

Where θ_i denotes the position of the ith employed bee, S represents the number of employed bees, and P_i is the probability of selecting the ith employed bee. Later, onlooker bees move by the roulette wheel selection. The amount of nectar is then determined:

$$x_{ij}(t+1) = \theta_{ij} + \phi\left(\theta_{ij}(t) - \theta_{kj}(t)\right) \tag{III.36}$$

Here x_i denotes the position of the i_{th} onlooker bee, t denotes the iteration number, θ_k is the randomly chosen employed bee, j represents the dimension of the solution and $\phi(\cdot)$ produces a series of a random variable in the range $[-1, 1]$.

- Move the Scouts

the given equation shows the move of scouts:

$$\theta_{ij} = \theta_{ijmin} + r \cdot \left(\theta_{ijmax} - \theta_{ijmin}\right) \tag{III.37}$$

Where r is a random number and $r \in [0, 1]$.

Next, the best food source obtained so far and the fitness value are updated according to the response, with the respect of termination condition the algorithm outputs the results or repeats the program from step 2.

III.3.5. Grey Wolf Optimizer (GWO)

The grey wolf algorithm inspired from the social leadership and hunting behavior of grey wolves in nature [84,85,95]. This approached consist of social hierarchy, enriching prey, search for prey, attacking prey, and hunting. Social hierarchy: The alpha (α) wolf is the leader of wolves, while Beta (β) and delta (δ) wolves are the second and third levels in the group, respectively. Those three wolves are considers as the best solution to lead the rest wolves known as omega (ω) wolves toward promising areas in order to find the global solution.

$$d = \left|c \cdot x_p(n) - x(n)\right| \tag{III.38}$$

$$x(n+1) = x_p(n) - a \cdot d \tag{III.39}$$

Where, x_p: position vector of the prey, n: current iteration, and x: position vector of a grey wolf.

The coefficient vectors a and c are defined as follows:

$$\vec{a}_{(.)} = 2\vec{l} \cdot \vec{r}_1 - \vec{l} \tag{III.40}$$

$$\vec{c}_{(.)} = 2 \cdot \vec{r}_2 \tag{III.41}$$

Here, r_1 and r_2 are random numbers in [0, 1].

The following equations illustrate the mechanism of hunting:

$$\begin{cases} \vec{d}_\alpha = |\vec{c}_1 \cdot \vec{x}_\alpha - \vec{x}| \\ \vec{d}_\beta = \left|\vec{c}_2 \cdot \vec{x}_\beta - \vec{x}\right| \\ \vec{d}_\delta = |\vec{c}_3 \cdot \vec{x}_\delta - \vec{x}| \end{cases} \tag{III.42}$$

$$\begin{cases} \vec{x}_1 = \vec{x}_\alpha - \vec{a}_1 \cdot \left(\vec{d}_\alpha\right) \\ \vec{x}_2 = \vec{x}_\beta - \vec{a}_2 \cdot \left(\vec{d}_\beta\right) \\ \vec{x}_3 = \vec{x}_\delta - \vec{a}_3 \cdot \left(\vec{d}_\delta\right) \end{cases} \tag{III.43}$$

$$x(n+1) = \frac{\vec{x}_1 + \vec{x}_2 + \vec{x}_3}{3} \tag{III.44}$$

III.3.6 Grasshoppers Optimization Algorithm (GOA)

The Grasshoppers Optimization Algorithm simulated by [96] is inspired from the natural flocking behaviour of grasshoppers. Those insects are regarded as pests because they damage agricultural corps. The swarming behaviour can be mathematically explained by the following equations [96–98]:

$$X_i = S_i + G_i + A_i \quad \text{(III.45)}$$

Where, X_i is the position of the i[th] grasshopper, S_i is the classical interaction, G_i is the gravity force onthe i[th] grasshopper, and A_i is the wind advection. The classical interaction is given by:

$$S_i = \sum_{j=1}^{N} s(d_{ij})\hat{d}_{ij} \quad \text{(III.46)}$$

$$d_{ij} = |x_i - x_j| \quad \text{(III.47)}$$

$$\hat{d}_{ij} = \frac{x_j - x_i}{d_{ij}} \quad \text{(III.48)}$$

Here, d_{ij} is the distance between the i[th] and j[th] grasshoppers, and s is a function for the definition of the strength of social forces defined as follows:

$$s(r) = fe^{-\frac{r}{l}} - e^{-r} \quad \text{(III.49)}$$

Where, f is the intensity of attraction, and l is the attractive length scale. The distance between grasshoppers ranges between 0 and 15. Repulsion is observed in the interval [0 2.079]. The grasshoppers enter the comfort zone if they are far from 2.079 units from other grasshoppers.

G component is defined as follows:

$$G_i = -g\hat{e}_g \quad \text{(III.50)}$$

Here, g is the gravitational constant and $\hat{e}_g$ is a unity vector towards the center of the earth. Parameter A is given by:

$$A_i = u\hat{e}_w \quad \text{(III.51)}$$

Where, u is a constant drift and $\hat{e}_w$ is a unit vector in the direction of the wind. The position is updated to a new position based on the common position of the grasshopper, the food source position, and the position of all other grasshoppers:

$$X_i = \sum_{\substack{j=1 \\ j \neq i}}^{N} s(|x_j - x_i|)\frac{x_j - x_i}{d_{ij}} - g\hat{e}_g + u\hat{e}_w \quad \text{(III.52)}$$

Here, N is the number of grasshoppers .Although, this equation (65) cannot be straight-forward applied for optimization because grasshoppers do not converge to a specified point. Hence, the following is a corrected equation to update the grasshopper's position:

$$X_i^d = c\left|\sum_{\substack{j=1 \\ j\neq 1}}^{N} c\,\frac{ub_d - lb_d}{2}\, s(\left|x_j^d - x_i^d\right|)\,\frac{x_j - x_i}{d_{ij}}\right| + \hat{T}_d \tag{III.53}$$

$$C = (C_{max} - 1)\,\frac{C_{max} - C_{min}}{Max\ Itr} \tag{III.54}$$

Where, ub_d is the upper bound; lb_d is the lower bound;$\hat{T}_d$ is the value of the D_{th} dimension in the target space (optimal solution found so far); and c is a decreasing coefficient to shrink the comfort zone, repulsion zone, and attraction zone.

III.3.7. Moth flame optimization (MFO) algorithm

The Moth-flame optimization algorithm was first introduced by Mirjalili [99]. Those fancy insets are quite similar to the butterfly family. The natural behaviour of them in navigation known as transverse orientation is the main motivation for the algorithm. Moths fly using a fixed angle with respect to the moon, which is a very efficient mechanism for long travelling distances in a straight line. The MFO algorithm follows the following steps:

$$MFO = (R, V, T) \tag{III.55}$$

moths randomly and also the fitness values of them, *V* is a main function that makes the moths move around the search space and *T* is a termination criterion flag.

In *V* function, the position of each moth with regard to a flame is updated as per Equation (III.56):

$$M_i = S(M_i, G_j) \tag{III.56}$$

where *S* is the spiral function, M_i is the i^{th} moth, and G_j is the j^{th} flame, whereas $S(M_i, G_j)$is calculated using the following equations:

$$S\big(M_i, G_j\big) = D_i e^{bl}\cos(2\pi l) + G_j \tag{III.57}$$

Here, *Di* represents distance of i^{th} moth from j^{th} flame, *b* is constant for announcing the shape of logarithmic spiral, and *l* is a random number between [− 1, 1]. *D* is calculated using the following equation.

$$D_i = \left|G_j - M_i\right| \quad \text{(III.58)}$$

Where G_j indicates the j[th] flame and $M_i$is the *i*[th] moth.

The position of moths is updated with respect to n different locations in the search space, which can degrade the best promising solutions exploitation. Consequently, the number of flames adaptively decreases over the course of iterations using the following formula:

$$flame_{no} = round(FN - 1 * \frac{N-1}{IN}) \quad \text{(III.59)}$$

where *I* is the current number of iterations, *I N* is the maximum number of iterations and FN is the maximum number of flames. More details can be found elsewhere [99–101].

III.4.The ANFIS Controller description

Adaptive Network-Based Fuzzy Inference System (ANFIS) is a hybrid soft computing algorithm that combined the neuro-fuzzy technique, namely fuzzy inference system (FIS) with artificial neural network (ANN).ANFIS was firstly introduced by Jang [102] and it is based on the first-order Sugeno fuzzy model [103].The ANFIS approaches integrate the best feature of fuzzy systems and neural networks by exercising learning ability and capability of FIS and ANN, respectively [104].The ANFIS architecture consists of if-then rules and couples of input-output data of fuzzy. For training, ANFIS uses neural network learning algorithms [104,105].As the algorithm used it to fine-tune the membership function (MF) and the associated parameter that methods the desired data sets.

For clarifying, a Sugeno fuzzy model with two inputs, x and y, and one output is considered. Normally, the fuzzy rules are reported as follows [102–105]:

Rule I: if $x=A_1$ *and* B_1, then

$$f_1 = p_1 x + q_1 y + r_1 \quad \text{(III.60)}$$

Rule II: if $x=A_2$ *and* $y=B_2$, then

$$f_2 = p_2 x + q_2 y + r_2 \quad \text{(III.61)}$$

Where, *f* is an output parameter (response), *p*, *q*, & r are linear parameters, and *A*&*B* are nonlinear parameters.Fig III.5. Illustrates the five layers used to create ANFIS structure while the function of each layer is represented as follows:

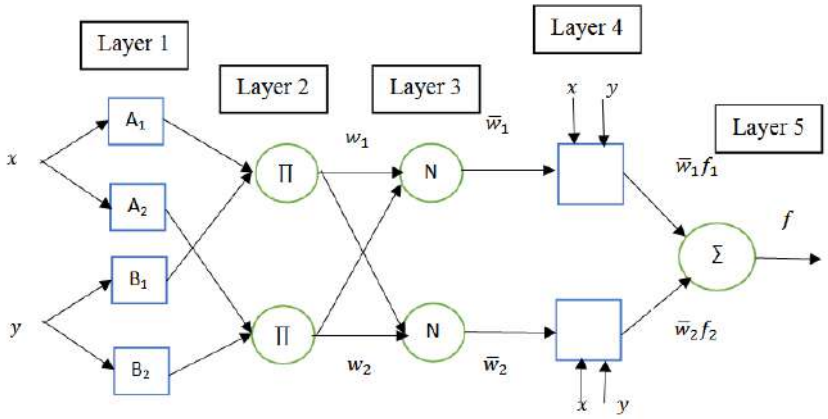


Figure III.5. The ANFIS architecture with two input parameters x and y.

Layer 1.

input nodes: The nodes of this layer generate membership grades to which they belong to each of the relevant fuzzy sets using MFs.

$$\begin{cases} O_i^1 = \mu_{Ai}(x) \\ O_i^2 = \mu_{Bi}(y) \end{cases} \quad \text{(III.62)}$$

Here, i= 1, 2 ,x, y are the crisp inputs to node i, and A_i &B_i. The linguistic label (small, large, etc.) associated with the node function. μ_{Ai} ,μ_{Bi} respectively.

Typically, μ_{Ai} (or μ_{Bi}) is selected as:

$$\mu_{Ai}(x) = \frac{1}{1+\left[\left(x-{}^{c_i}/_{a_i}\right)^2\right]^{b_i}} \quad \text{(III.63)}$$

or

$$\mu_{Ai}(x) = exp\left\{-\left(\frac{x-c_i}{a_i}\right)^2\right\} \quad \text{(III.64)}$$

Where, $\{a_i, b_i, c_i\}$ are the premise parameter set.

Layer 2.

rule nodes: The firing strength of each rule is calculated with mathematical multiplication .For instance,

$$O_i^2 = w_i = \mu_{A_i}(x) \cdot \mu_{B_i}(y) \quad \text{(III.65)}$$

Each node output represents the firing strength of a rule.

Layer 3.

Average nodes: The i^{th} node calculates the ratio of the i^{th} rule firing strength to the summation of the firing strengths of all rules consequently, defined as:

$$O_i^3 = \overline{w}_i = \frac{w_i}{w_1+w_2} \quad \text{(III.66)}$$

w_i is taken as the normalized firing strength.

Layer 4.

Consequent nodes: The node function of the fourth layer calculates the contribution of each i^{th} rules toward the overall output, as given :

$$O_i^4 = \overline{w}_i f_i = \overline{w}_i(p_i x + q_i y + r_i) \quad \text{(III.67)}$$

where, $\overline{w}_i$isthe output of the ***layer 3***, and$\{p_i, q_i, r_i\}$the parameter set.

Layer 5.

*Output nodes:*The single-node computes the overall output by adding all the incoming signals from the 4th layer :

$$O_i^5 = overall\ output = \sum_i \overline{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \quad \text{(III.68)}$$

The final output of ANFIS can be given as:

$$f_{out} = \overline{w}_1 f_1 + \overline{w}_2 f_2 = \frac{w_1}{w_1+w_2} f_1 + \frac{w_2}{w_1+w_2} f_2 = (\overline{w}_1 x)p_1 + (\overline{w}_1 y)q_1 + (\overline{w}_1)r_1 + (\overline{w}_2 x)p_2 + (\overline{w}_2 y)q_2 + (\overline{w}_2)r_2 \quad \text{(III.69)}$$

III.5 .Designing the ANFIS-PID Dynamic Controller

The complete control approach that controls the dynamics of the robot mobile based on the ANFIS-PID is divided into three steps:

- ***First step:***

Fig.III.6, represent the first step, which indicates the training stage of the ANFIS controller. The ANFIS use their adaptive and learning skills to learn and to predict the best recommended control efforts based on the already data. Which they are obtained from the designed GWO-PID controllers in the previous section.

The realized data were saved in a MATLAB/SIMULINK file. From the command ***anfisedit***, these data were taken for training. The complete process was illustrated in Fig**.** III.7.

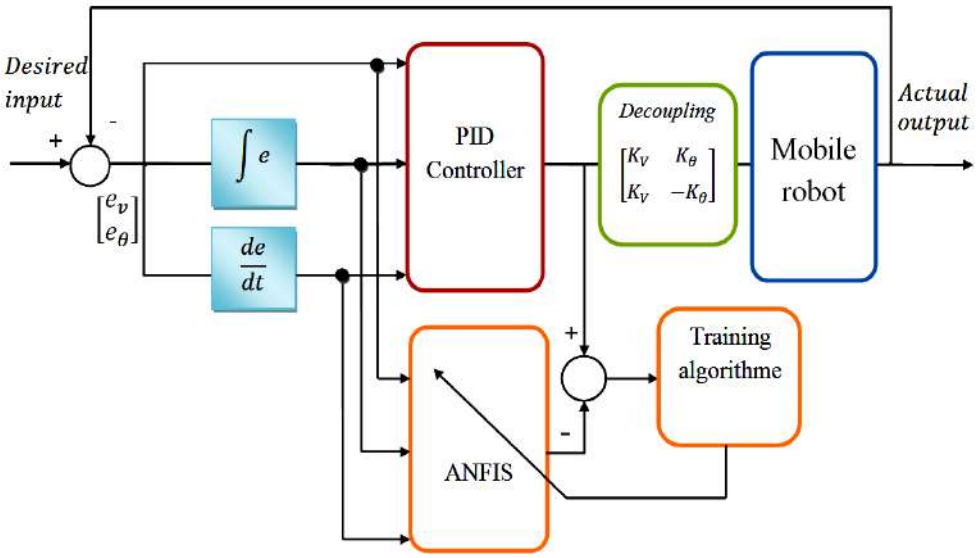
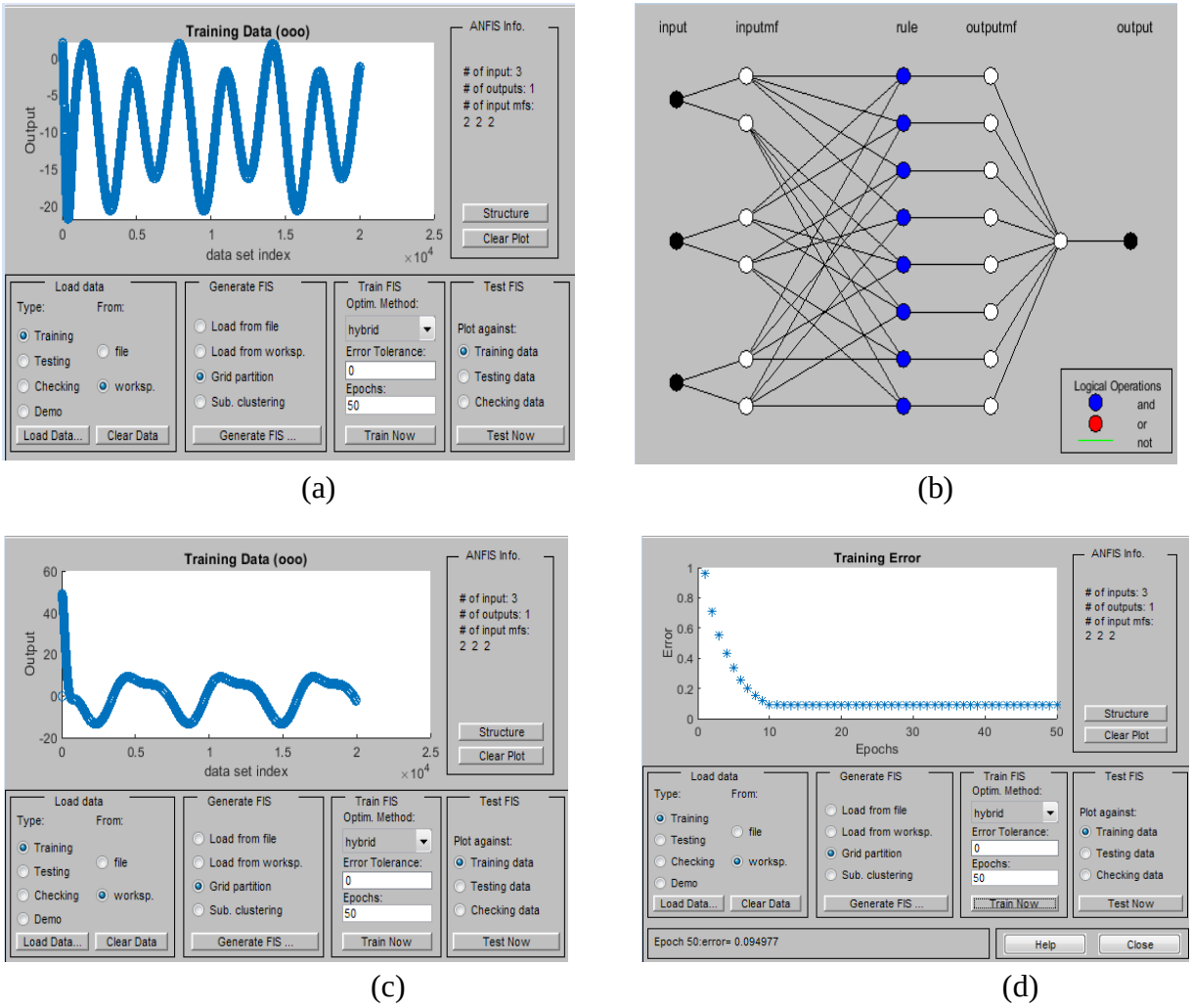


Figure III.6. The ANFIS in training phase



(a) (b)

(c) (d)

Figure III.7 (a) data configuration for first ANFIS-PID controller,(b) structure of the ANFIS-PID controller (c) data configuration for second ANFIS-PID controller ,(d) error minimization corresponding to the second controller.

Fig. III.7 presents the internal composition of the ANFIS controller. The method used for ANFIS training is the hybrid training algorithm, with the input nodes (2, 2, 2), gaussmf membership function for the inputs and linear membership function for the output, each having 8 rules (see Fig. III.7.b), the epoch length was used is 50 iterations for each sample, with, 0.01s as the Simulink sampling time.

At the end of the training the result was saved as a *fis* file with respect to Sugeno-style. The root mean square error (RMSE) was found to be 0.1053 and be 0.094977 06 for ANFIS-PID1 and ANFIS-PID2, respectively.

- ***Second step:*** scaling the new parameters using the GWO optimizer.

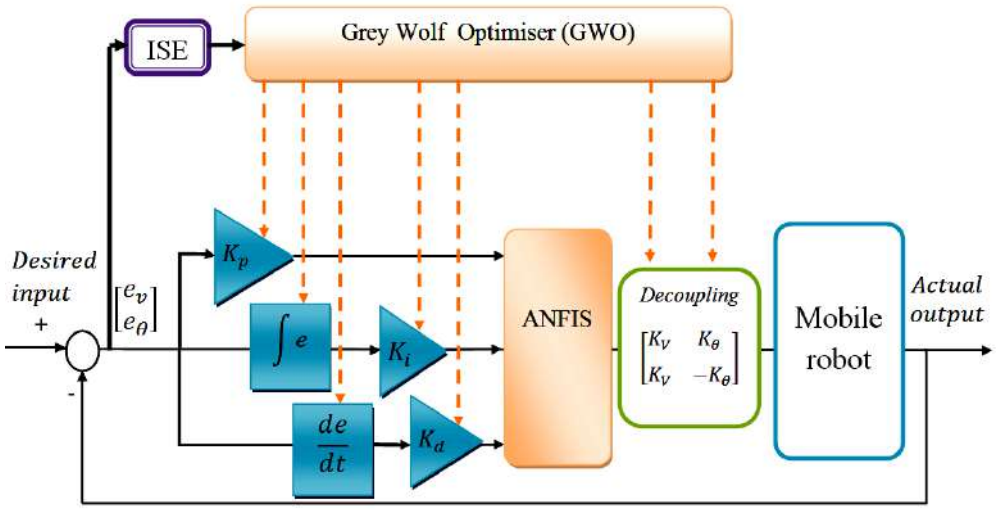


Figure III.8. Adjusting the parameters for the designed ANFIS-PID controller using GWO.

After the training step is complete and the optimal data are received, the designed ANFIS-PID controller must be developed by improving the parameters, i.e.,Kp, KI, Kd, Kv, Kθ as demonstrated in Fig.III.8. To find the optimal values of the scaling gains, grey wolf method is proposed to guarantee a minimum ISE value.

- ***Third step:***

The gained control scheme of the dynamic controller is showed in Fig.III.9. It reveals the hybridize between the ANFIS and PID controller, abridged as ANFIS-PID. Two trained ANFIS-PID controllers are selected for running the right and left DC motor voltage of the WMR, where a decoupling process is applied to separate the two control actions.

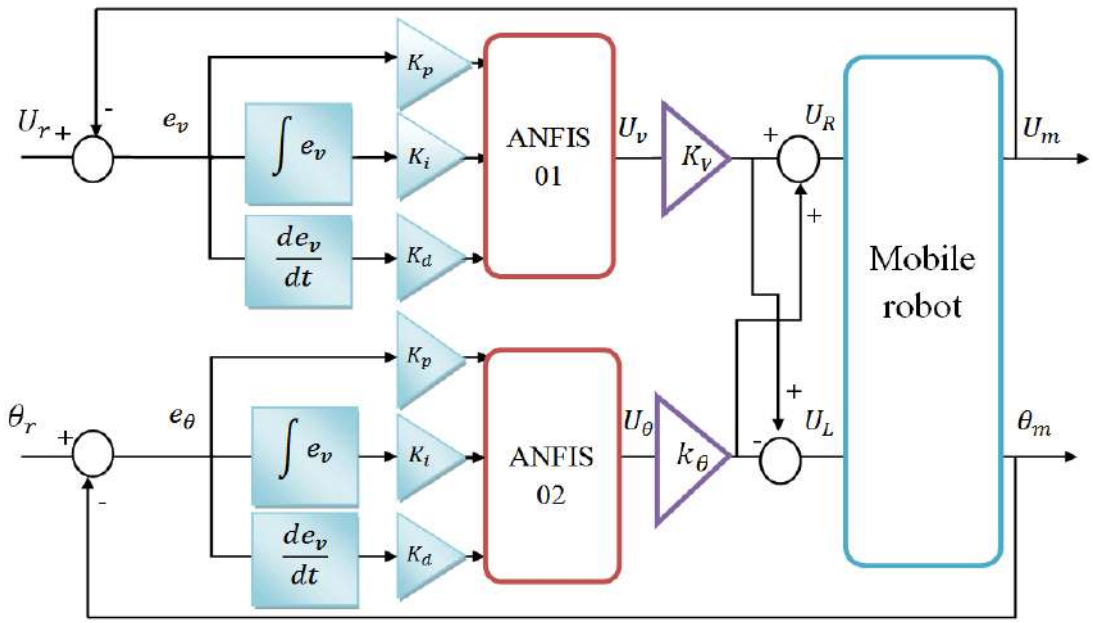


Figure III.9. Bloc diagram of the final ANFIS-PID dynamic controller.

The first controller takes the difference between the actual and reference speed from the right wheel as an input. The angle controller located in the left wheel studies the difference between the actual and desired orientations as an input.

Equations (III.70) describe the relationship between the input voltages of the right and left DC actuators U_R and U_L, respectively, with the outputs of the first and second controller U_V and U_θ, respectively, where the factors of the decoupling system are respectively Kv and Kθ.

$$\begin{cases} U_R = K_V \cdot U_V + K_\theta \cdot U_\theta \\ U_L = K_V \cdot U_V - K_\theta \cdot U_\theta \end{cases} \quad \text{(III.70)}$$

Where

$$\begin{cases} e_v = U_r - U_m \\ e_\theta = \theta_r - \theta_m \end{cases} \quad \text{(III.71)}$$

U_r is the desired velocity, U_m the actual velocity, e_v the velocity error, θ_r the desired angle, θ_m the measured orientation, and e_θ is the orientation error.

III.6. Results and discussion

This study selected the use of various criteria and evaluations, a comparative study of the convergence curve performance between all controllers, the transition parameters of linear speed and of the angle for all techniques, also several cases study for the path tracking has been created, in order to confirm the over performance and high robustness of the suggested GWO-ANFIS-PID controller, In comparison to GWO-PID ,PSO-PID ,ALO-PID, GOA-PID, DA-PID, ABC-PID and

MFO-PID. In order to examine the convergence behavior of the nature inspired algorithms, all simulation experiments were executed more than 30 times for each method on a PC separately, the maximum number iterations was 100, and population size is set to 20.

III.6.1. Convergence Curve of all Algorithms

Fig.III.10 highlights the convergence curves of the seven selected optimizations algorithms for designing the PID controller in comparison to our proposed hybrid controller (GWO-ANFIS-PID). The result proves the over-performance of the GWO-ANFIS-PID controller. The minimum ISE value attained for all approaches used in the study is shown in Table III.1.

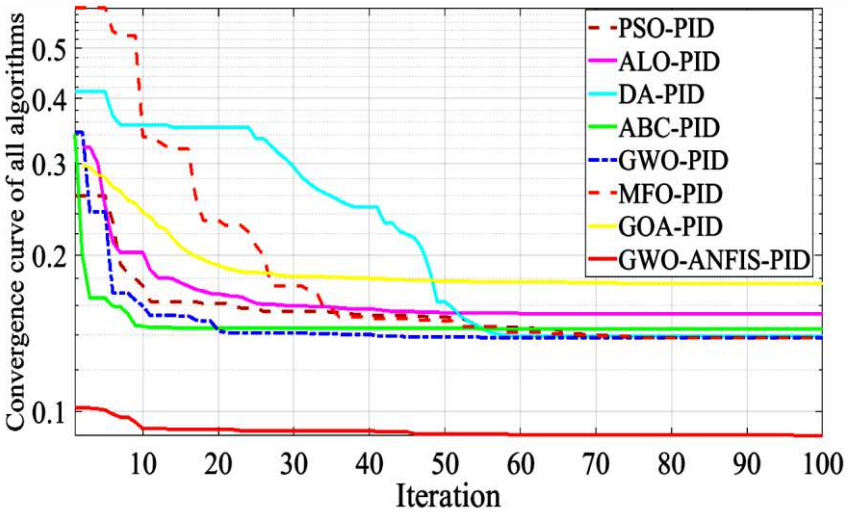


Figure III.10. Convergence curve of the seven algorithms compared to GWO-ANFIS-PID.

Table III.1. ISE values of the seven algorithms.

algorithms	ISE
PSO-PID	0.138826
DA-PID	0.139292
ALO-PID	0.154018
ABC-PID	0.143813
GOA-PID	0.176242
MFO-PID	0.138489
GWO-PID	0.138510
GWO-ANFIS-PID	0.089690

From Fig.III.10 and Table.III.1, it's straightforward that the GWO optimizer gives the lowest ISE value with the faster convergence behavior compared to the other six techniques. Which were made it the best candidate for designing the ANFIS-PID controller.

We know that the mobile robot is high nonlinear MIMO system ,where the conventional PID controller with fixed parameters has deteriorate the control performance. This is because the PID controller has a simple linear structure which is not designed to deal with complex nonlinear

system. Our proposed strategy (ANFIS-PID) is nonlinear control scheme which seems to be more useful, this because the ANFIS take the optimal value from the GWO PID controller during the online training process from various scenarios and try to solve the nonlinearity of the system, this is the reason why we can see the big difference in the ISE value (see table III.1).

III.6.2. Step Response Performance of Speed Controller

Fig.III.11 reveals the comparative analyses of the step responses performance for the linear speed that was designed with the help of the seven different techniques in comparison to the proposed GWO-ANFIS-PID controller.

Table III.2 shows the parameter details of all meta-heuristic approaches, for the linear speed controller, and table III.3 presents the performance achieved in the time domain.

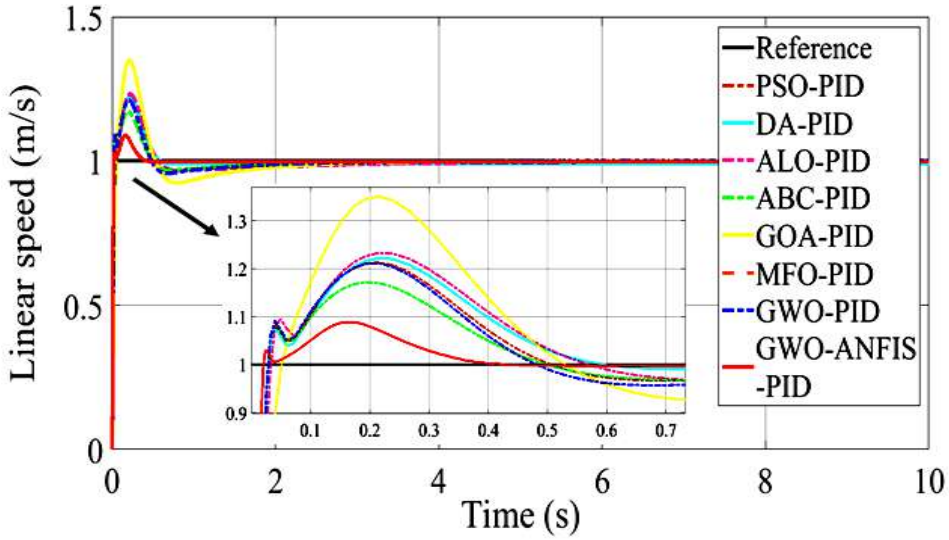


Figure III. 11. Step response Comparison for all velocity controllers.

Table III.2 The tuned gain details of linear velocity for all Controllers .

Methods	**Speed controller parameter tuning**				
	K_p	K_i	K_d	K_V	K_θ
PSO-PID	149.98	97.40	1.013	0.50	0.50
DA-PID	150	1	1	0.50	0.50
ALO-PID	108.46	52.24	0.72	0.50	0.50
ABC-PID	148.78	142.40	1	0.50	0.50
GOA-PID	81.60	89.97	1	0.50	0.50
MFO-PID	149.88	149.97	1	0.50	0.50
GWO-PID	150	148.94	1.001	0.50	0.50
GWO-ANFIS-PID	452.007	13.59	1.25	1.5	0.9

Table III.3. Performance characteristic for velocity Controller

Methods	**Transition parameters for speed control**					
	t_r	Mp%	Undershoot	Peak	t_p	t_s
PSO-PID	0.0190	21.3279	3.3018e-05	1.2133	0.2148	1.5344
DA-PID	0.0191	22.2135	3.3017e-05	1.2221	0.2148	0.5248
ALO-PID	0.0221	23.2889	3.5750e-05	1.2329	0.2148	1.9670
ABC-PID	0.0190	17.1998	3.3097e-05	1.1720	0.1933	1.3128
GOA-PID	0.0332	35.0362	7.8275e-05	1.3504	0.2148	2.0684
MFO-PID	0.0189	21.1015	3.3025e-05	1.2110	0.2148	1.4805
GWO-PID	0.0189	21.1147	3.3017e-05	1.2111	0.2148	1.4849
GWO-ANFIS-PID	0.0125	8.8853	0	1.0889	0.1700	0.2547

The best overshoot percentage (Mp %) was found by the ABC-PID, and best settling time (ts) is reached by DA-PID.where the GWO-PID controller advantage is the fast rise time and accepted settling time (ts) value obtained, but a high overshoot, this result were shared with the PSO-PID and MFO-PID.

While the GWO-ANFIS-PID controller enclose all advantages of the previous seven techniques by giving the best settling time and an excellent overshoot value and fastest rise time (tr) .

III.6.3 Step Response Performance of Angle Controller

Fig.III.12 describes the comparative analyses of the angle step responses performance that was designed with the aid of the seven previous mentioned approaches in comparison to the recommended GWO-ANFIS-PID controller, the comparative results of transient response are highlighted in Table III.4.

From Fig.III.12, Table III.4 and Table III.5, we can say that the GWO-ANFIS-PID gives a zero-overshoot value, with the best rise time (t_r for 10% → 90%) and settling time value (t_s for ±2% tolerance), were the GWO-PID comes at the second place . for that reason Our next examinations are concentrating only on analyzing the two best candidates (GWO-PID and GWO-ANFIS-PID).

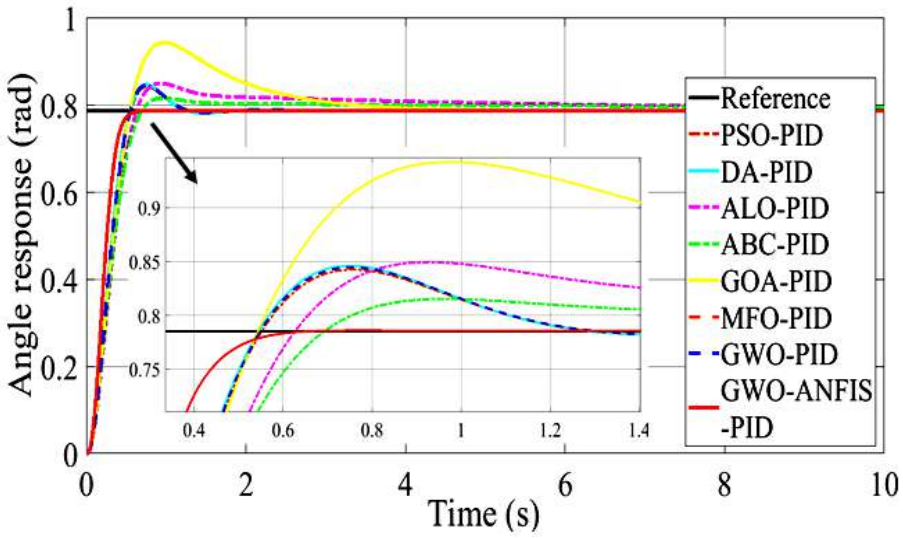


Figure III.12. Comparison of step Response for the orientation.

Table III.4. The tuned main parameters of all controllers for angle control.

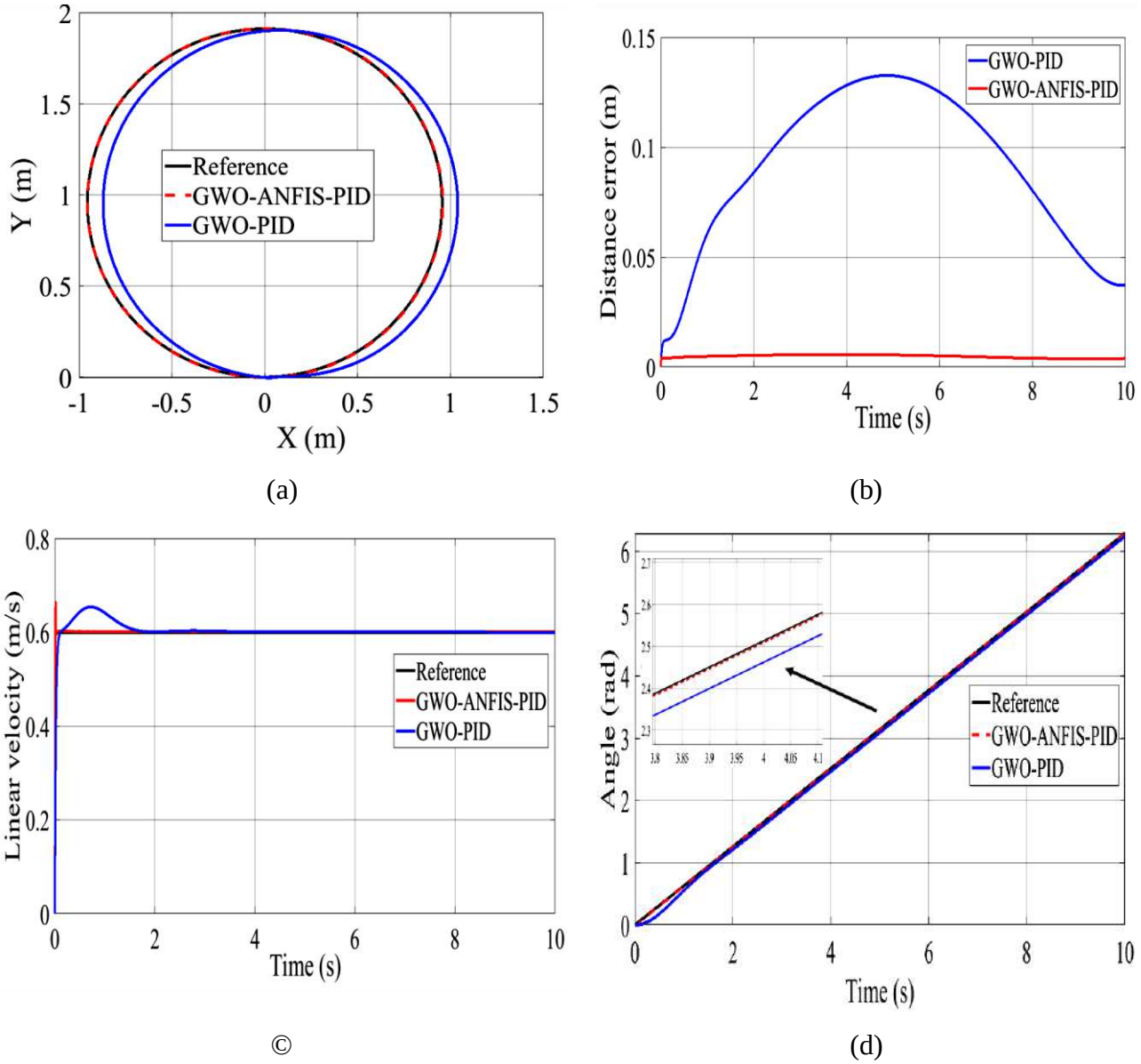
Methods	**angle controller parameter tuning**				
	K_p	K_i	K_d	K_V	K_θ
PSO-PID	149.93	1	31.019	0.50	0.50
DA-PID	150	1	30.53	0.50	0.50
ALO-PID	131.90	22.60	33.21	0.50	0.50
ABC-PID	145.24	11.46	37.86	0.50	0.50
GOA-PID	149.95	112.96	41.14	0.50	0.50
MFO-PID	150	1	30.85	0.50	0.50
GWO-PID	150	1.388	30.83	0.50	0.50
GWO-ANFIS-PID	500	1.01	75.31	1.5	0.9

Table III.5. Performance characteristic of all controllers for angle control.

Methods	**Transition parameters for angle controller**					
	t_r	Mp%	Undershoot	Peak	t_p	t_s
PSO-PID	0.3539	7.2995	0	0.8423	0.7449	1.1016
DA-PID	0.3499	7.7670	0	0.8460	0.7449	1.1006
ALO-PID	0.4034	8.1941	0	0.8493	0.9259	6.0690
ABC-PID	0.4247	3.8441	0	0.8152	0.9539	3.1976
GOA-PID	0.3606	20.0551	0	0.9424	0.9826	3.3034
MFO-PID	0.3522	7.5138	0	0.8440	0.7449	1.1013
GWO-PID	0.3518	7.5961	0	0.8446	0.7449	1.1057
GWO-ANFIS-PID	0.2834	0.1047	0	0.7858	0.7700	0.4302

III.6.4. Circuler path (case study 01)

This case, represent a conventional type of smooth trajectory that have been made to confirm the capacity of the GWO-ANFIS-PID controller in dealing with this kind of paths, in comparison to the GWO-PID.



(a) (b)

© (d)

Figure III.12. (a) Circle trajectory in XY plane, (b) error of the distance, (c) linear velocity response for all controllers (d). Angle response of all controllers.

Here, a progressive variation in both linear and angular speed is applied for that purpose, results proves that the GWO-ANFIS-PID controller better handling with smooth trajectory than the GWO-PID controller, due to a minimum tracking error from the suggested controller than the GWO-PID one. and since the PID controller gives a very high overshoot in the linear velocity response at starting time. Which makes it unsupported for handling the tracking issue.

III.6.5. Diamond chape trajetory

The Diamond shaped trajectory is a clear example of a non-continuous gradient path. The main problem of this type of paths is the sharp and a non-continuous motion, which calls for a serious control technique. Fig III.13 (a)- (d) expresses, respectively, the trajectory in the XY plane for both selected controllers compared to the reference path, the corresponding results of the angular velocity, comparative results of the linear speed, and at last the tracking error according to the distance.

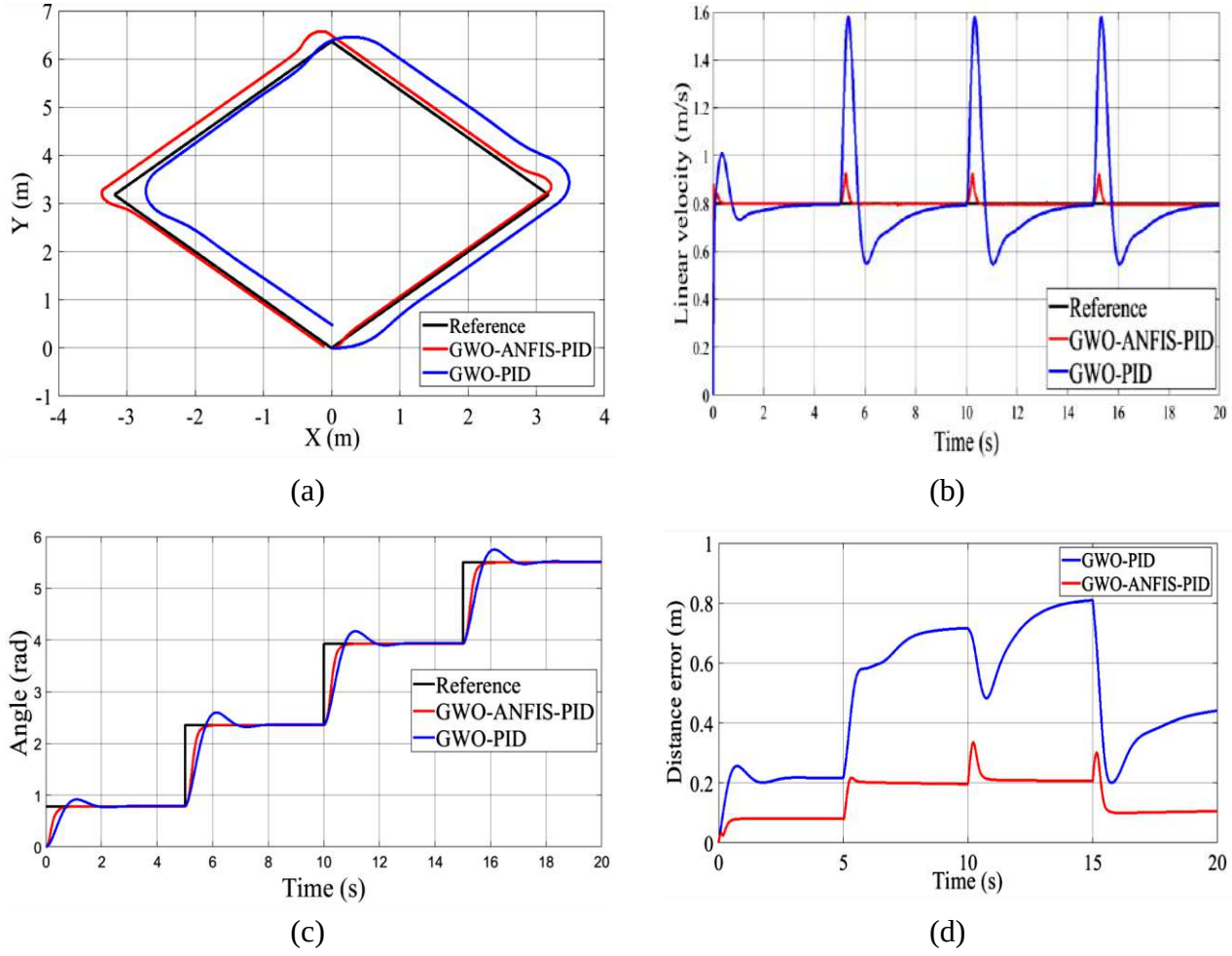


Figure III.13. (a) diamod shape trajectory in XY plane,(b) linear velocity corresponding to the to controllers, (c) angle response corresponding to the path for the two controllers,(d) error of the distance.

The spontaneous changing in velocity and due to the slow rise time and settling time, makes the GWO-PID controller gives slow actions, that causes high distance error and a very high overshoot value too from this controller, that might generate an unstable and a troubled motion, which could by the time damage the DC actuators (see Fig.III.13 (b)), for that reason it's unsuitable for this kind of paths. *The GWO-ANFIS-PID* controller has a tolerable overshoot contrasted to the GWO-PID

and rapidly time response, and a very small distance error too, which makes it supported for this type of trajectory.

III.6.6. Zig zag trajectory

Here a trapezoidal signal was applied, for that purpose progressively changing can be seen for both linear speed and angle (see Fig.III.14 (c), and (d)), the GWO- ANFIS-PID controller gives less overshoot and better time response, contrary to the GWO-PID which gives an important overshoot and a high distance error value.

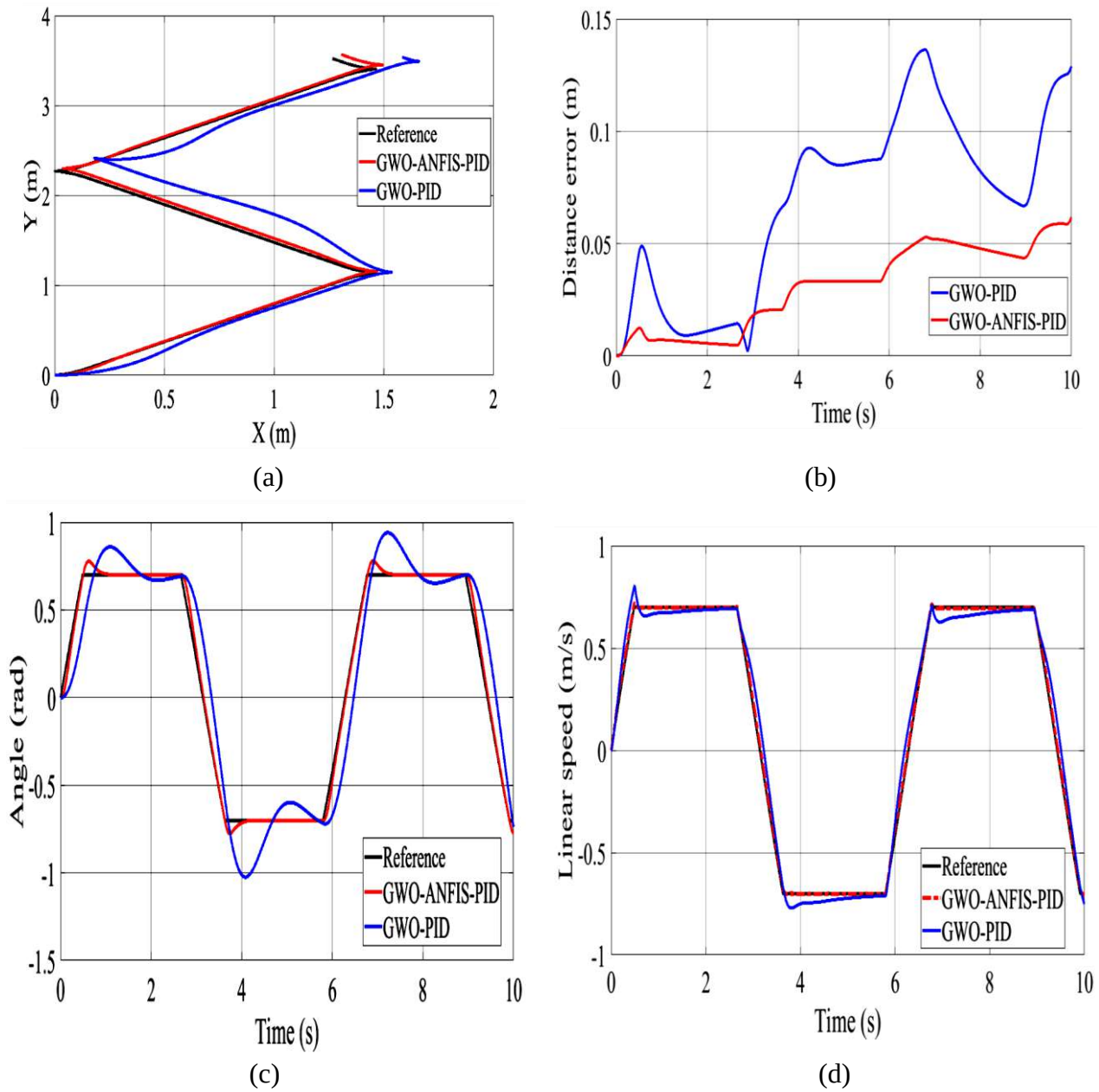


Figure III.14. (a) zigzag path in XY plane ,(b)the tracking error of the controllers (c) the theta angle response corresponding to the controllers ,(d) linear velocity corresponding to the controllers.

III.6.7. Adding the backsteping controller in the external loop

The designed ANFIS-PID controller role is to guarantee a minimum linear and angular velocity error, which calls for a kinematic controller. Accordingly, a back-stepping controller was suggested to maintain a minimal distance error that has been already demonstrated in Fig.III.16 (d). The error is less than 0.002 m. To highlight the robustness of the Back-stepping combined with ANFIS-PID controller, a square and a lemniscates trajectory were preferred.

III.6.7. 1 Case of lemniscates

In order to generate this path, the following equation were applied [106]:

$$x_R(n) = 2.5 * cos\left(2 * \pi * \frac{t}{20}\right) \tag{III.72}$$

$$y_R(n) = -2.5 * sin\left(2 * \pi * \frac{t}{30}\right) \tag{III.73}$$

$$\theta_R(n) = a\tan 2\left[\frac{\left(\frac{y_R(n) - y_R(n-1)}{t+\epsilon}\right)}{\frac{(x_R(n) - x_R(n-1))}{t+\epsilon}}\right] \tag{III.74}$$

Where the back-stepping parameters are selected as: k_x=10, k_y=80 and k_θ=15.

The desired and physical paths of the lemniscates are presented in Fig.III.15, displayed in blue and red lines, respectively, Fig.III.16 (a)- (d) illustrate the errors of x, y, θ, and the tracking trajectory, respectively. The effect of the included kinematic controller in the external loop is indicated in Fig.III.16(d), the distance error is less than 0.002 meter. For that reason, the two trajectories have exactly the same form in Fig.III.15.

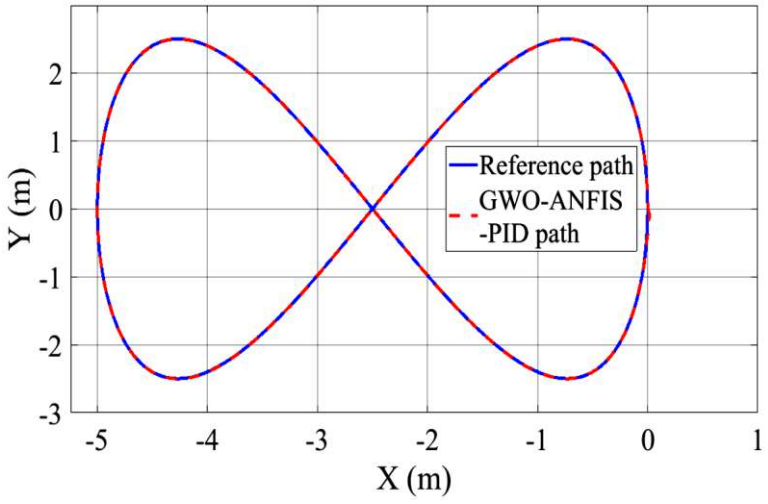


Figure III. 15. lemniscate path.

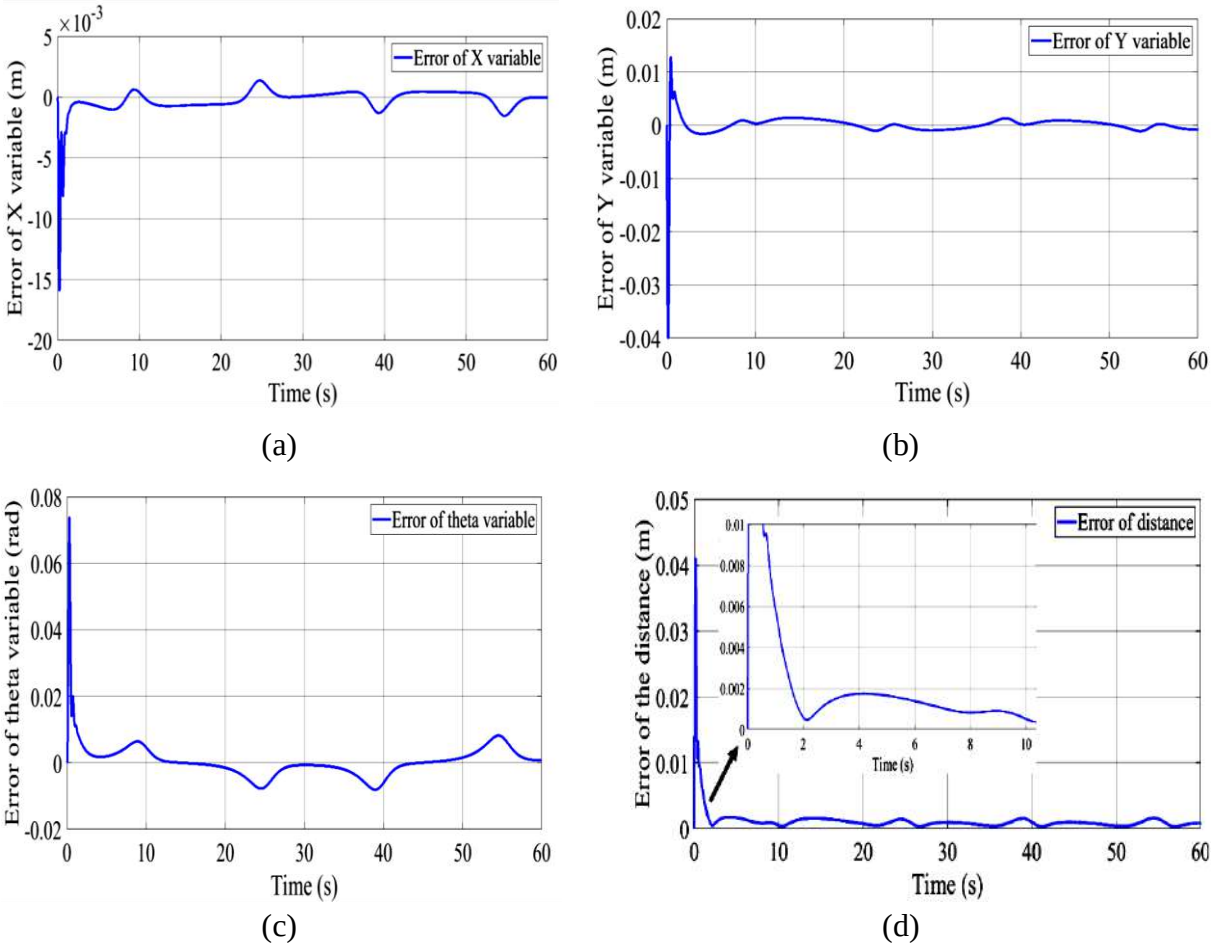


Figure III.16. Error corresponding to (a) x,(b) y; (c) the theta angle ,(d) the trajectory tracking.

III.6.7. 2 .Case of square trajectory

The recommended and actual trajectories of the square path are shown in Fig.III.17, highlighted in blue and red lines, respectively, Fig.III.18 (a)-(c) display the errors of x, y, θ, and the tracking distance error, respectively.

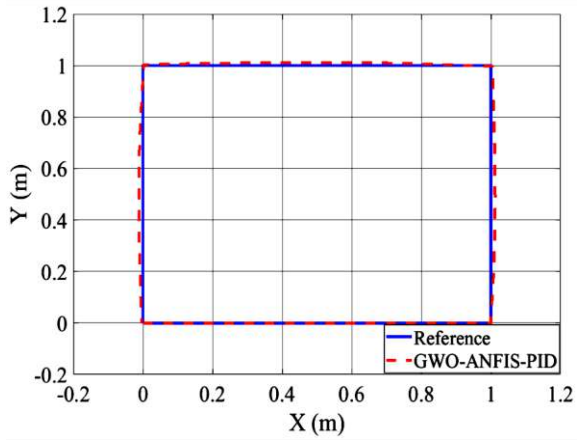


Figure III.17. Square trajectory.

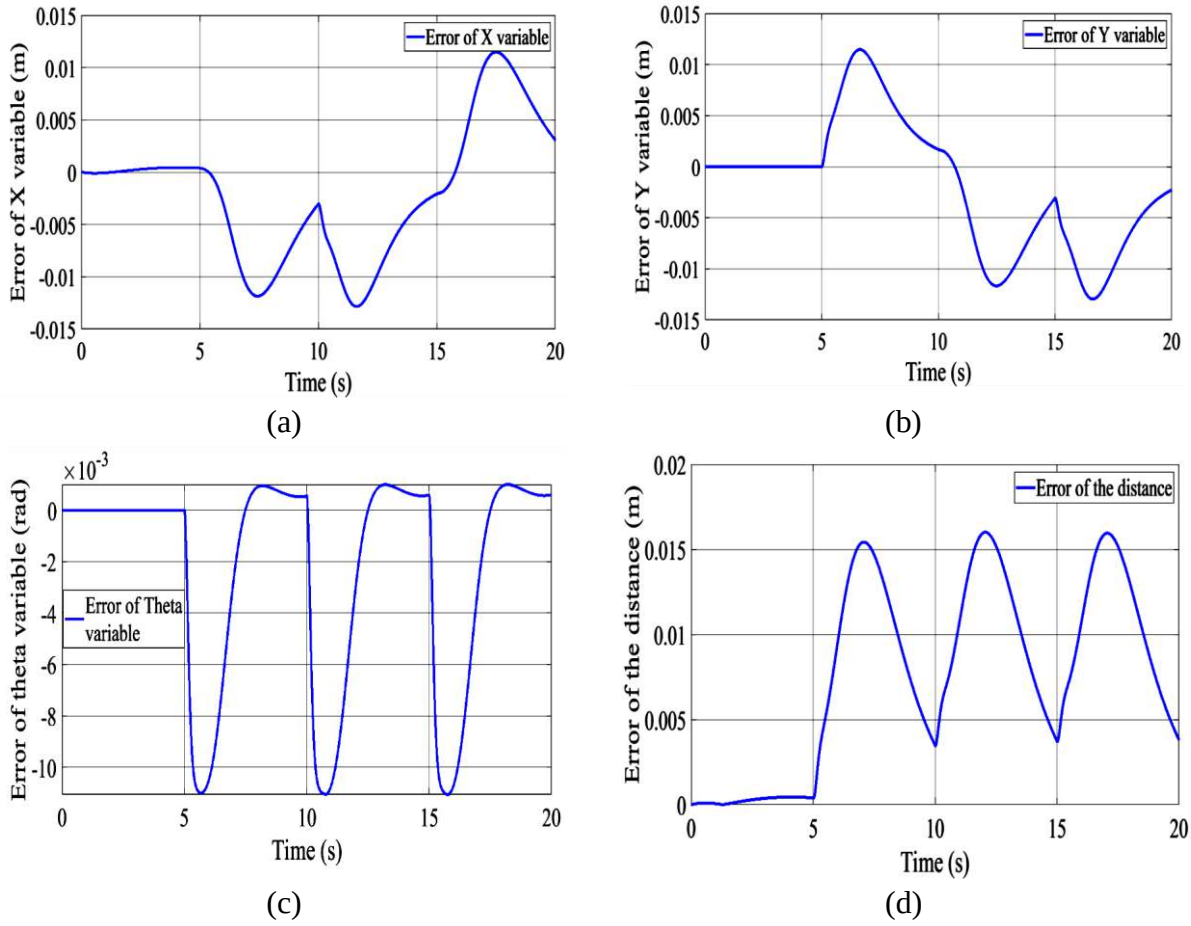


Figure III.18. Error corresponding to (a) x,(b) y; (c) the theta angle ,(d) the trajectory tracking.

The square trajectory is classified as a sharp-shaped path too, therefore it was chosen for this study in order to justify the capability of the proposed hybrid controller, a minimum distance error reached (about 0.015 meter), and an error angle fewer than 0.01 rad, therefore we can see the nearly superposition of the actual on reference path in Fig.III. 17.

III.7. Conclusion

This chapter suggested a new combination technique between the PID controller and an adaptive neuro-fuzzy inference system (ANFIS), shortened to ANFIS-PID controller, to deal with the motion regulation of the wheeled mobile robot (WMR). Tuning parameter of the proposed hybrid controller are challenging tasks, a comparative study of various nature inspired methods are introduced in the study, based on the integral square error (ISE). the selected algorithms namely, particle swarm optimizer (PSO),grey wolf optimizer algorithms (GWO), dragonfly optimizer(DA),ant lion optimizer (ALO),grasshopper algorithm (GOA) ,moth flame optimizer (MFO),and the artificial bee colony algorithm (ABC), simulation in MATLAB environment gives a demonstration and judgment between the ANFIS-PID controller and these seven techniques ,in terms of lowest overshoot value,

minimum ISE value, convergence curve, less peak, peak time, and faster rise time and settling time received. Various study cases have been made, demonstrates the ability and stability of the proposed ANFIS-PID controller tuned by GWO algorithm, which has the capacity to generate smooth and suitable control signals for both linear and angular speeds, great performance over the PID controllers. Moreover, to guarantee a low tracking error, a back-stepping technique was applied at the kinematic control level, where a lemniscates and a square trajectory have already demonstrated the robustness of the mentioned controller in both smooth and sharp, shaped trajectory.

Chapter IV:

Optimal backstepping Fractional order PID Controller design

Chapter IV Optimal Back stepping-fractional order PID controller design

IV.1.Introduction

The PID control technique (traditional proportional integral derivative) is the oldest control strategy. It's still employed in various industrial processes. This is mainly because of the simple structure and eases of design and implementation. When significant robustness and transient efficiency are required, the control effect is limited. To handle this kind of problems, the fractional order proportional integral derivative (FOPID) controller is adopted for this purpose. It is obtained by the combination of fractional calculus and the traditional PID controller, which introduces two new parameters λ and μ (the integral and derivative fractional-order, respectively) to the classic PID control. Those two added parameters have increased the performance and the robustness of the controller; several papers demonstrate the high efficiency and performance of FOPID controller compared to classic PID controller [45-48].

In the last decade, the popularity of FOPID control has widely increased and attracted more attention in many engineering areas, including robotics engineering [45-58]. Tuning the five parameters (Kp, Kd, KI, λ and μ) of the FOPID controller is a challenging task compared to conventional PID control. Also, because WMR is a high coupled nonlinear system, taking into account some particular constraints such as the gain margin and sensitivity conditions.

In this chapter, a combination between the back stepping controller with the fractional order proportional integral derivative (FOPID) controller has been proposed for handling the trajectory tracking problem of wheeled mobile robots (WMR).as mentioned before Tuning the parameters of the FOPID is challenging, to overcome this issue a hybrid meta-heuristic optimization algorithm has been utilized. This evolutionary technique is known as the hybrid whale grey wolf optimizer (HWGO), which benefits from the performances of the two traditional algorithms, the whale optimizer algorithm (WOA) and the grey wolf optimizer (GWO), to obtain the most suitable solution. The efficiency of the HWGO algorithm is compared against those of the original algorithms WOA, GWO, the particle swarm optimizer (PSO), and the hybrid particle swarm grey wolf optimizer (HPSOGWO). The simulation results in MATLAB–Simulink environment revealed the highest efficiency of the suggested HWGO technique compared to the other methods in terms of settling and rise time, overshoot, as well as steady-state error. Finally, a star trajectory is made to illustrate the capability of the mentioned controller.

IV.2. Review on Fractional Calculus

The fractional calculus is a name for the theory of integrals and derivatives of arbitrary order, which unify and generalize the notions of integer-order differentiation and n-fold integration [107]. The first appearance of fractional calculus was before three centuries ago. In 1695, the derivative of the order α=1/2 was described by Leibniz [108], it has been developed up to nowadays.

The Fractional order system is a dynamical system that can be modeled by a fractional differential equation which derivatives of non-integer order. Fractional order systems are useful in studying the anomalous behavior of dynamical systems in electrochemistry, biology, viscoelasticity, and chaotic systems. The fractional calculus is used in different fields with many applications in physics, engineering, mathematical biology, finance, life sciences, and optimal control. Fractional calculus is a successful tool for describing complex quantum field dynamical systems, dissipation, and long-range phenomena that cannot be well illustrated using ordinary differential and integral operators [108]. The generalized differintegrator operator may be as [109]:

$$ {}_aD_x^{\alpha} f(t) = \frac{d^{\alpha} f(t)}{[d(x-a)]^2} \tag{IV.1} $$

Where, a is starting limitation, x is the final limitation, and α is a calculus order. The differential and integral operators can be generalized into one fundamental operator ${}_aD_x^{\alpha}$ where:

$$ {}_aD_x^{\alpha} = \begin{cases} \frac{d^{\alpha}}{dt^{\alpha}} & if\ \alpha > 0 \\ 1 & if\ \alpha = 0 \\ \int_a^x (d\tau)^{-\alpha} & if\ \alpha < 0 \end{cases} \tag{IV.2} $$

Where, a is starting limitation, x is the final limitation, and α is a calculus order.

There are several mathematical definitions used for fractional differintegral. These definitions do not always lead to exact results, but approximate result. The most important of these mathematical definitions are as follows [109]:

IV.2.1.Riemann-Liouville Definition (RL).

The general operator D for a real order of differentiation is integral if the order of differentiation D is less than 0.

The Riemann-Liouville integral definition depends on Cauchy's formula that is based on the multi integral.

Indefinite integrals of **Cauchy's formula** are given by [110]:

$$ {}_aD_x^{\alpha} f(x) = \frac{1}{\Gamma(\alpha)} \int_a^x (x-t)^{\alpha-1} f(t) dt \ , \alpha \epsilon N \quad \text{(IV.3)} $$

Where, a is starting limitation, x is the final limitation, and α is a calculus order.

Proof: Indefinite integrals of Cauchy's formula, see Appendix (A).

Where, $\Gamma(\cdot)$ is the well-known Euler's gamma function.

The Euler's Gamma function Γ: (0, ∞) →R, defined by [110].

$$ \Gamma(x) = \int_0^{\infty} t^{x-1} e^{-t} dt, \quad \text{(IV.4)} $$

For n ∈ N, we have (n−1)! =Γ (n). For more information about Euler's Gamma function, see Appendix (A).

The Riemann-Liouville derivative definition is depending on the law of exponents of an operator D^n of integer calculus.

$$ D^n D^{\alpha} f(t) = D^n [D^{\alpha} f(t)] = D^{n+\alpha} f(t) \quad \text{(IV.5)} $$

$$ D^n I^n f(t) = f(t) \quad \text{(IV.6)} $$

The derivative definition when α>0 will be defined using the definition for α<0 as follows [110]:

$$ {}_aD_x^{\alpha} f(x) = aD_x^{\alpha} \left[aD_x^{-(n-\alpha)} f(x) \right] = \frac{1}{\Gamma(n-\alpha) dx^n} \frac{d^n}{dx^n} \left[\int_a^x \frac{f(t)}{(x-t)^{\alpha-n+1}} dt \right] \quad \text{(IV.7)} $$

For $\alpha \in R^+$

Then, the general Riemann-Liouville definition is:

$$ {}_a^{RL}D_x^{\alpha} f(x) = \begin{cases} \frac{1}{\Gamma(-\alpha)} \int_a^x (x-t)^{-\alpha-1} f(t) dt & if \alpha < 0 \\ \frac{1}{\Gamma(n-\alpha)} \frac{d^n}{dx^n} \left[\int_a^x \frac{f(t)}{(x-t)^{\alpha-n+1}} dt \right], n = min\{k \in N : k > \alpha\}, & if\ \alpha > 0 \\ f(t) & if\ \alpha = 0 \end{cases} \quad \text{(IV.8)} $$

IV.2.2.Caputo Definition (C) [110]

The second fractional calculus definition is the Caputo definition.

- The integral definition of Caputo method is same as the Riemann-Liouville integral definition.

$$ {}_aD_x^{-\alpha} = \frac{1}{\Gamma(\alpha)} \int_a^x (x-t)^{\alpha-1} f(t) dt \quad \alpha \in N \quad \text{(IV.9)} $$

- The Caputo definition derivative is same as the Riemann-Liouville, depending on the law of exponents of operator D^n.

$$ {}_aD_x^{\alpha} f(x) = aD_x^{\alpha-n}[aD_x^n f(x)] = \frac{1}{\Gamma(n-\alpha)} \int_a^x \frac{f^n(\tau)d\tau}{(x-\tau)^{\alpha-n+1}} \quad \text{(IV.10)} $$

The general Caputo definition is

$$ {}_a^cD_x^{\alpha} f(x) = \begin{cases} \frac{1}{\Gamma(-\alpha)} \int_a^x (x-t)^{-\alpha-1} f(t)dt & if \alpha < 0 \\ \frac{1}{\Gamma(n-\alpha)} \int_a^x \frac{f^n(\tau)d\tau}{(x-\tau)^{\alpha-n+1}}, n = min\{k \in N : k > \alpha\}, & if\ \alpha > 0 \\ f(t) & if\ \alpha = 0 \end{cases} \quad \text{(IV.11)} $$

IV.2.3. Grunwald-Letnikov Definition (GL).

In the Grunwald-Letnikov, the derivative and integral have the same definition which is depending on the multi derivative integer calculus, so the general definition of Grunwald-Letnikov is:

$$ {}_a^{GL}D_x^{\alpha} f(t) = lim_{h \to 0} \frac{1}{h^{\alpha}} \sum_{j=0}^{\left[\frac{x-a}{h}\right]} (-1)^j \binom{\alpha}{j} f(t-jh) \quad \text{(IV.12)} $$

Where, $\left[\frac{x-a}{h}\right]$ means the integer part, the first method is derived from the Grunwald-Letnikov definition which is coming from the basic definition of derivative:

$$ Df(t) = lim_{h \to 0} \frac{f(t)-f(t-h)}{h} \quad \text{(IV.13)} $$

Let reckon explicit expressions for higher order derivatives. The first two cases are;

$$ D^2 f(t) = lim_{h \to 0} \frac{Df(t)-Df(t-h)}{h} = lim_{h \to 0} \frac{\frac{f(t)-f(t-h)}{h} - \frac{f(t-h)-f(t-2h)}{h}}{h} = $$

$$ lim_{h \to 0} \frac{f(t)-2f(t-h)+f(t-2h)}{h^2} \quad \text{(IV.14)} $$

$$ D^3 f(t) = lim_{h \to 0} \frac{D^2 f(t)-D^2 f(t-h)}{h} = lim_{h \to 0} \frac{\frac{f(t)-2f(t-h)+f(t-2h)}{h^2} - \frac{f(t-h)-2f(t-2h)+f(t-3h)}{h^2}}{h} = $$

$$ lim_{h \to 0} \frac{f(t)-3f(t-h)+3f(t-2h)-f(t-3h)}{h^3} \quad \text{(IV.15)} $$

$$ f(t) - 3f(t-h) + 3f(t-2h) - f(t-3h) = \sum_{j=0}^{3} (-1)^j \binom{3}{j} f(t-jh) \quad \text{(IV.16)} $$

The numerical calculation formula of fractional derivative can be achieved as [110]:

$$_{t-R}D_x^{\alpha} f(t) \approx h^{-\alpha} \sum_{j=0}^{\frac{R}{T}} b_j f(t-jh) \quad \text{(IV.17)}$$

Where, R is the length of memory. T is the sampling time that always replaces the time increment h during approximation. The weighting coefficients bj can be calculated recursively by [109,110]:

$$b_o = 1, b_j = \left(1 - \frac{1+\alpha}{j}\right) b_{j-1}, (j \geq 1) \quad \text{(IV.18)}$$

The second method $\binom{\alpha}{j}$ is the binomial coefficient .

$$\binom{\alpha}{0} = 1, \binom{\alpha}{j} = \frac{\alpha(\alpha-1)(\alpha-2)\ldots(\alpha-j+1)}{j!} \quad \text{(IV.20)}$$

It can be replaced by Euler's Gamma function [109,110]

$$\binom{\alpha}{j} = \frac{\Gamma(\alpha+1)}{j!(\alpha-j+1)} \quad \text{(IV.21)}$$

Then, the equation (IV.13) becomes:

$$_{a}D_x^{\alpha} f(t) = lim_{h\to 0} \frac{1}{h^{\alpha}} \sum_{j=0}^{\frac{x-a}{h}} (-1)^j \frac{\Gamma(\alpha+1)}{j!\Gamma(\alpha-j+1)} f(t-jh) \quad \text{(IV.22)}$$

Fractional derivatives are often difficult to find analytically. Numerical approximations are consequently needed. The simplest is the using of the Grunwald- Letnikov definition.

All these methods are same as the Runge-Kutta method which is used to plot the output of the transfer function isn the time domain, but in frequency domain and in MATLAB block, Simulink of these methods cannot be used [109].

Transfer functions of fractional order are not easy to implement for computational purposes. Simulations are usually carried out by integer powers of s only. This makes the task of finding integer order approximations of fractional order transfer functions the most important one. Fractional transfer functions are usually replaced by integer transfer functions, with a behavior close enough to the one desired, but much easier to handle. A band-limit implementation of fractional order controller is important in practice, and the finite dimensional approximation of the fractional order controller should be done in a proper range of frequencies of practical interest. This is true

since the fractional order controller in theory has an infinite memory and some sort of approximation using finite memory must be done There are many different methods of finding such approximations one, of these methods is the Analog Rational Approximation.

IV.2.4.A Modified Approximate Realization Method

There are many ways to carry out the approximation of fractional transfer functions, such as Crone approximation, the Carlson approximation and Analog Rational approximation, etc. In chapter five in this thesis, the method of Analog Rational approximation was adopted. This method can be carried out in the following two steps [111]:

- Calculating the frequency range (ω_b, ω_h), where it is used to approximate the fractional order transfer function.
- An approximate realization method for fractional derivative in the frequency range of interest (ωb, ωh). This method here gives a better approximation than Oustaloup's method in both low frequency and high frequency parts. Performing an analog integer-order approximation of the fractional order transfer function. If there is one pole or zero fractional order ,it can represent in transfer function consist of multi zeros and multi poles. [109-111]

Let;

$$s^{\alpha} \approx \left(\frac{1+\frac{s}{\frac{d}{b}\omega_b}}{1+\frac{s}{\frac{b}{d}\omega_h}} \right)^{\alpha} \tag{IV.23}$$

Where, $s = j\omega, 0 < \alpha < 1, b > 0, d > 0$.Then

$$s^{\alpha} \approx \left(\frac{(\frac{d}{b}\omega_b + s)\frac{b}{d}\omega_h}{(\frac{b}{d}\omega_h + s)\frac{d}{b}\omega_b} \right)^{\alpha} \tag{IV.24}$$

$$s^{\alpha} \approx \left(\frac{(\frac{d}{b}\omega_b + s)\frac{b}{d}\omega_h}{(\frac{b}{d}\omega_h s + s^2)} \right)^{\alpha} \left(\frac{bs}{d\omega_b} \right)^{\alpha} \tag{IV.25}$$

$$s^{\alpha} \approx \left(\frac{d\omega_b\omega_h + b\omega_h s}{b\omega_h s + ds^2} \right)^{\alpha} \left(\frac{bs}{d\omega_b} \right)^{\alpha} \tag{IV.26}$$

$$s^{\alpha} \approx \left(\frac{b\omega_h s + ds^2}{b\omega_h s + ds^2} + \frac{d\omega_b\omega_h - ds^2}{b\omega_h s + ds^2} \right)^{\alpha} \left(\frac{bs}{d\omega_b} \right)^{\alpha} \tag{IV.27}$$

$$s^{\alpha} \approx \left(1 + \frac{d\omega_b\omega_h - ds^2}{b\omega_h s + ds^2} \right)^{\alpha} \left(\frac{bs}{d\omega_b} \right)^{\alpha} \tag{IV.28}$$

Where, $\omega_b * \omega_h = 1$

$$s^{\alpha} \approx \left(\frac{bs}{d\omega_b} \right)^{\alpha} \left(1 + \frac{d - ds^2}{ds^2 + bs\omega_h} \right)^{\alpha} \tag{IV.29}$$

By using Taylor series expansion within (ω_b, ω_h).

$$\left(1+\frac{d-ds^2}{ds^2+bs\omega_h}\right)^{\alpha}=1+\alpha\left(1+\frac{d-ds^2}{ds^2+bs\omega_h}\right)^{\alpha}+\frac{\alpha(\alpha-1)}{2}\left(1+\frac{d-ds^2}{ds^2+bs\omega_h}\right)^{\alpha}+\cdots \quad \text{(IV.30)}$$

$$s^{\alpha}\approx\frac{(d\omega_b)^{\alpha}b^{-\alpha}}{\left(1+\alpha p(s)+\frac{\alpha(\alpha+1)}{2}p^2(s)+\cdots\right)}\left(\frac{1+\frac{s}{\frac{d}{b}\omega_b}}{1+\frac{s}{\frac{b}{d}\omega_h}}\right)^{\alpha} \quad \text{(IV.31)}$$

Where;

$$p(s)=\frac{-ds^2+d}{ds^2+bs\omega_h} \quad \text{(IV.32)}$$

Amputating the Taylor series to the first order term;

$$s^{\alpha}\approx\left(\frac{d\omega_b}{b}\right)^{\alpha}\left(\frac{ds^2+bs\omega_h}{d(1-\alpha)s^2+bs\omega_h+d\alpha}\right)\left(\frac{1+\frac{s}{\frac{d}{b}\omega_b}}{1+\frac{s}{\frac{b}{d}\omega_h}}\right)^{\alpha} \quad \text{(IV.33)}$$

Equation (2.28) is stable, if all poles are on the left half s-plane. Note there are three poles; one is $-b\omega_h/d$, this pole is negative because ω_h, b, d > 0. The other two poles are depending on the roots of following equation:

$$d(1-\alpha)s^2+bs\omega_h+d\alpha=0 \quad \text{(IV.34)}$$

Whose real parts are negative as $0<\alpha<1$.

Based on the well known zig-zag line approximation idea in Bode plot, let

$$\left(\frac{1+\frac{s}{\frac{d}{b}\omega_b}}{1+\frac{s}{\frac{b}{d}\omega_h}}\right)^{\alpha}=\lim_{N\to\infty}\prod_{k=-N}^{N}\frac{1+\frac{s}{\omega_k'}}{1+\frac{s}{\omega_k}} \quad \text{(IV.35)}$$

Where $-\omega_k'$ and $-\omega_k$ are zeros and poles of rang *k*.

$$\omega_k'=\left(\frac{d\omega_b}{b}\right)^{\frac{\alpha-2k}{2N+1}} \quad \text{(IV.36)}$$

$$\omega_k=\left(\frac{b\omega_h}{d}\right)^{\frac{\alpha+2k}{2N+1}} \quad \text{(IV.37)}$$

Hence,

$$s^{\alpha}\approx K\left(\frac{ds^2+bs\omega_h}{d(-\alpha)s^{\alpha}+bs\omega_h+d\alpha}\right)\prod_{k=-N}^{N}\frac{s+\omega_k'}{s+\omega_k} \quad \text{(IV.38)}$$

Where,

$$K=\left(\frac{d\omega_b}{b}\right)^{\alpha}\prod_{k=-N}^{N}\frac{\omega_k}{\omega_k'} \quad \text{(IV.39)}$$

The following steps are a brief summarized the procedure for the modified approximation [111]:

- Suppose the frequency range (ω_b, ω_h) and N.
- Based on the fractional order α, calculate ω'_k and ω_k according to (IV.30) and (IV.31)
- Compute *K* from the equation (IV.33)
- Obtain the approximate rational transfer function from (IV.32) to replace s^α.

IV.2.5.Laplace Transform of Fractional Order transfer function [110]

Let recall some basic facts about the Laplace transform.

The function *F* (s) of the complex variable *s* is defined by;

$$F(s) = L\{f(t); s\} = \int_0^\infty e^{-st} f(t)dt \tag{IV.40}$$

The Laplace transform of the convolution

$$(f * g)(t) = \int_0^t f(t-\tau)g(\tau)d\tau = \int_0^t f(t)g(t-\tau)d\tau \tag{IV.41}$$

The Laplace transform of those functions

$$L\{(f * g)(t); s\} = F(s)G(s) \tag{IV.42}$$

The property of this equation can be used for the evaluation of the Laplace transform of fractional order [26].

$$L\{f^n; s\} = s^n F(s) - \sum_{k=0}^{n-1} s^{n-k-1} f^{(k)}(0) = s^n F(s) - \sum_{k=0}^{n-1} s^k f^{(n-k-1)}(0) \tag{IV.43}$$

A) Laplace Transform of the Riemann-Liouville Fractional Derivative [110]

Laplace transform of the Riemann-Liouvillc definition of D^p is

$$L\left[{}_0D_t^p f(t)\right] = \begin{cases} s^p F(s), & p \le 0 \\ s^p F(s) - \sum_{k=0}^{n-1} s^k \left[{}_0D_t^{p-k-1} f(0)\right], & n-1 < p \le n \in N \end{cases} \tag{IV.44}$$

The proof that: The result is obvious for p=0. For negative orders,

$$L\left[{}_0D_t^p f(t)\right] = L\left[\frac{1}{\Gamma(-p)}\int_a^t (t-\tau)^{-p-1} f(\tau)d\tau\right] \quad \text{(IV.45)}$$

By using the Laplace transform of the convolution, this becomes

$$L\left[{}_0D_t^p f(t)\right] = \frac{1}{\Gamma(-\alpha)} L[t^{-p-1}] * L[f(t)] \quad \text{(IV.46)}$$

$$L\left[{}_0D_t^p f(t)\right] = \frac{1}{\Gamma(-p)} s^{-1+p+1}\Gamma(-1-p+1)F(s) = s^p F(s) \quad \text{(IV.47)}$$

Where,

$$L[t^p] = s^{-1-p}\Gamma(p+1) \quad \text{(IV.48)}$$

For positive orders,

$$L\left[{}_0D_t^p f(t)\right] = L\left[D^n\, {}_0D_t^{p-n} f(t)\right] \quad \text{(IV.49)}$$

By the Laplace transform of an integer derivative,

$$L\left[{}_0D_t^p f(t)\right] = s^n s^{p-n}F(s) - \sum_{k=1}^{n} s^{n-k} D^{k-1}\, {}_0D_t^{p-n} f(0) \quad = s^p F(s) - \sum_{k=1}^{n-1} s^{n-k-1}\, {}_0D_t^{p-n+k} f(0) \; = s^p F(s) - \sum_{k=1}^{n-1} s^k\, {}_0D_t^{p-k-1} f(0) \quad \text{(IV.50)}$$

B) Laplace Transform of the Caputo Derivative

The Laplace transform of the Caputo Derivative definition of D is

$$L\left[{}_0D_t^p f(t)\right] = \begin{cases} s^p F(s) & p \le 0 \\ s^p F(s) - \sum_{k=1}^{n-1} s^{p-k-1}\, {}_0D_t^k f(0) & , n-1 < p \le n \in N \end{cases} \quad \text{(IV.51)}$$

The proof that: The result is obvious for *p*=0. For negative orders, the same as for the Laplace transform of the Riemann-Liouvillc definition. The second case is positive order.

The proof:

$$L\left[{}_0D_t^p f(t)\right] \; = L\left[{}_0D_t^{p-n} D^n f(t)\right] \; = s^{p-n}[s^n F(s) - \sum_{k=1}^{n} s^{p-k} D^{k-1} f(0)] \; = s^p F(s) - \sum_{k=1}^{n-1} s^{p-k-1} D^k f(0) \quad \text{(IV.52)}$$

C) Laplace Transform of the Grunwald-Letnikov Fractional Derivative

The Laplace transform of the Grünwald-Letnikov definition of D^p is [110].

$$L\left[{}_0D_t^p f(t)\right] = s^p F(s) \quad \text{(IV.53)}$$

IV.3. Controller Design and Strategy

Fig IV.1.presents the schematic design of the control strategy based on the dynamic and kinematic of the wheeled mobile robot. The trajectory generator produces the reference coordinate x, y and θ, a back-stepping kinematic controller is located in the external loop, here the controller experience the difference between the reference data received from the bloc trajectory generator and actual value from the robot then generate his own angular and linear speeds that are sent into the internal loop where two blocs of FOPID controller are used as a dynamic controller. Here the two FOPID controllers yield their own signals of angular and linear speeds based on the reference values obtained from the kinematic controller.

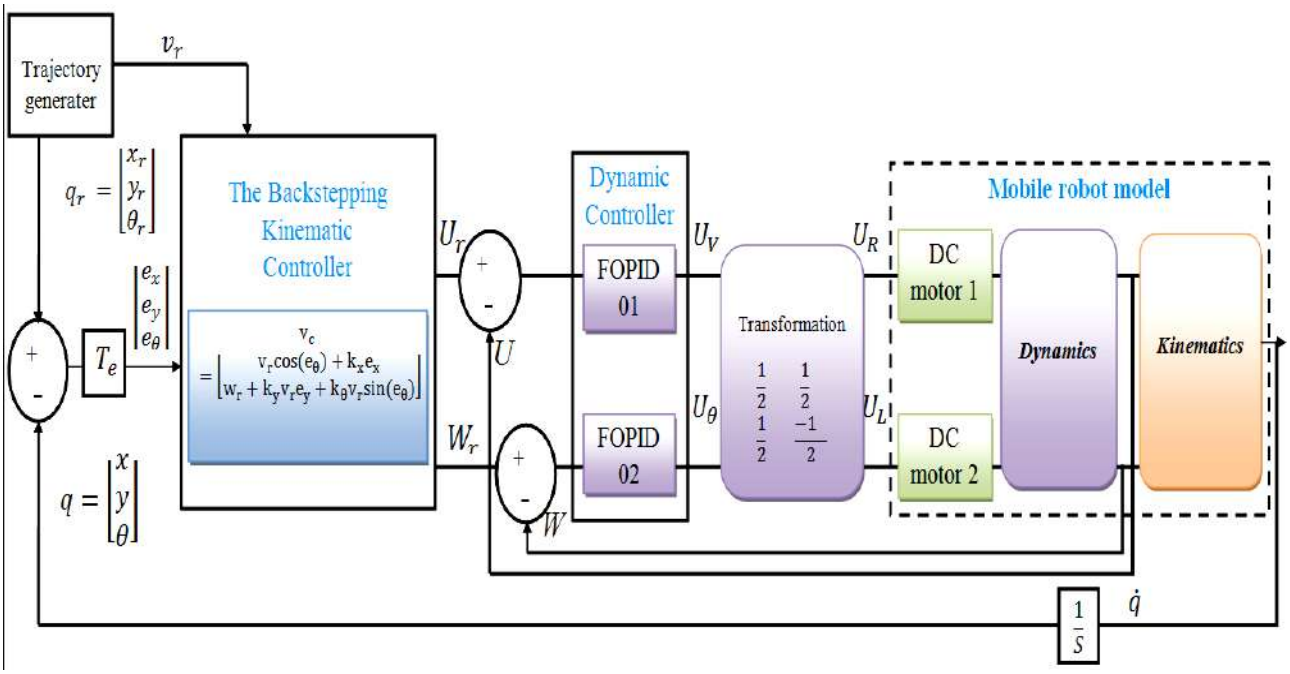


Figure IV.1. Control scheme of WMR.

IV.3.1 Kinematic Controller Design

One of the goals here is to realize path tracking. The back-stepping technique has been widely used and is well known as a stable tracking control rule which makes it adopted for this purpose [16]. The controller structure is highlighted in Fig IV.1., where the input error and velocity vector (v_c) are:

$$\begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix} = \begin{bmatrix} cos\,\theta & sin\,\theta & 0 \\ -sin\,\theta & cos\,\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_r - & X \\ Y_r - & Y \\ \theta_r - & \theta \end{bmatrix} = T_e e_r \tag{IV.54}$$

$$v_c = \begin{bmatrix} v_r cos(e_\theta) + k_x e_x \\ w_r + k_y v_r e_y + k_\theta v_r sin(e_\theta) \end{bmatrix} \tag{IV.55}$$

Where, k_x, k_y and k_θ are the tuning parameters.

IV.3.2.Dynamic Controller Description

The FOPID controller was presented for the first time by Podlubny in 1999 [112]. The FOPID controller is a natural extension of the classical PID controller with an arbitrary order of its integration and differentiation actions [113]. With a more sophisticated controller, new design strategies are possible with the most flexible controller for plant with performance limitations. The block diagram of a single input-single output (SISO) closed loop control system with fractional order ($PI^{\lambda}D^{\delta}$) controller is illustrated in Fig IV.2.

Fig. IV.2. presents the parallel structure of the FOPID controller block diagram, with input E(S) and output U(S).

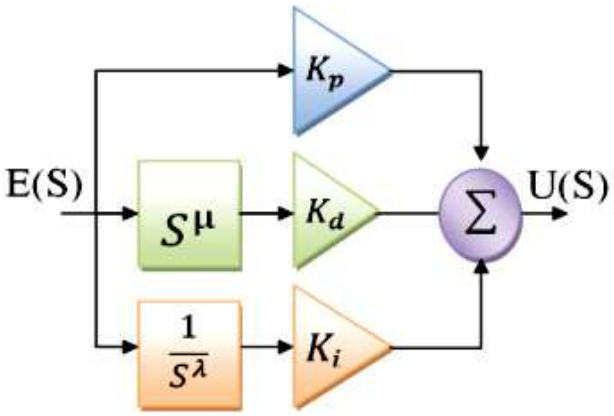


Figure IV.2. Block diagram of fractional order.

The representation of the transfer function for the Fractional Order PID (FOPID) in the time domain is given by Equation (IV.56), and in the Laplace domain by Equation (IV.57).

$$u(t) = K_p e(t) + k_i D_t^{-\lambda} e(t) + K_d D_t^{\mu} e(t) \tag{IV.56}$$

$$C(s) = \frac{U(s)}{E(s)} = K_p + \frac{K_i}{S^{\lambda}} + K_d S^{\mu} \tag{IV.57}$$

Where, K_p, K_i and K_d are proportional, integral and derivative gains constants, respectively, λ and μ are factional order of the integral and derivative term. All type of the classic PID may be obtained by taking (λ, μ) = {(1, 1), (1, 0), (0, 1), (0, 0)}, as shown in Fig IV.3.

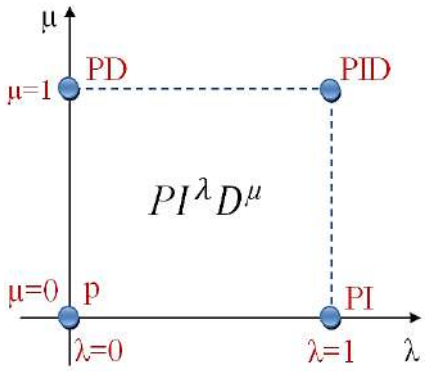


Figure IV.3. Generalized FOPID controller.

IV.3.3. tuning methodology

The WMR block diagram for tuning parameters of the Two FOPID controllers is highlighted in Fig IV.4. The first controller receives the difference between the reference and actual velocity from the right wheel as an input where the desired velocity is constant: Ur =1 m/s.

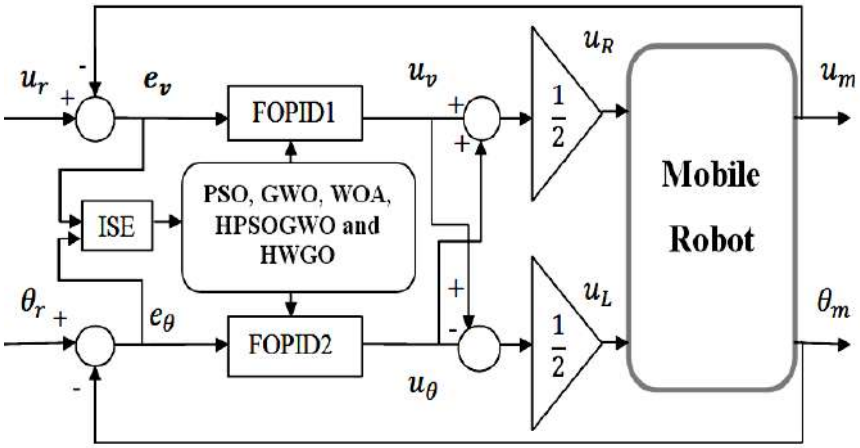


Figure IV.4. Bloc diagram for parameters tuning.

The second controller in the left wheel experiences the difference between the actual and desired orientations as input. Where the reference angle was represented by a constant value: θ_r=0.785 rad

The following equations (IV.58) represent the relation between the input voltage of the right and left DC actuators, respectively U_R, U_L with the outputs of the first and second controller, respectively U_V, U_θ:

$$\begin{cases} U_R = \dfrac{U_V + U_\theta}{2} \\ U_L = \dfrac{U_V - U_\theta}{2} \end{cases} \qquad \text{(IV.58)}$$

The cost function used in the study is the integral square error (ISE) which is demonstrated by equation (IV.59):

$$ISE = \left(\int_0^{\infty} [e_v(t)]^2 dt + \int_0^{\infty} [e_\theta(t)]^2 dt \right) \quad \text{(IV.59)}$$

where:

$$\begin{cases} e_v = U_r - U_m \\ e_\theta = \theta_r - \theta_m \end{cases} \quad \text{(IV.60)}$$

U_r is the desired velocity, U_m the actual velocity, e_v the velocity error, θ_r the desired orientation, θ_m the measured orientation, and e_θ is the orientation error,

The optimization algorithms used to tune the parameters (K_p, K_I, K_d, λ, μ) are described in the next section.

IV.4. Meta-Heuristic Algorithms

A comparative study has been made to demonstrate the capability of the hybrid grey wolf whale optimizer (HWGO), compared to particle swarm optimizer (PSO),The grey wolf optimizer (GWO) whale optimizer algorithm (WOA),the hybrid particle swarm grey wolf optimizer (HPSOGWO), Therefore, a brief description of the algorithms is necessary ,while the PSO and GWO has been already described in previous chapter.

IV.4.1.Hybrid PSO-GWO Algorithm

HPSOGWO is the hybridize of the Particle Swarm Optimization with the Grey Wolf Optimizer algorithm that uses the functionalities of both variants to enhance exploitation in PSO and GWO and to generated both variants' strength [85].

In this algorithm, the position of the first three agents is updated in the search space using Equation (III.41). The control of the grey wolf exploration and exploitation is conducted with the inertia constant. Using the following modified equations:

$$\begin{aligned} \vec{d}_\alpha &= |\vec{c}_1 \cdot \vec{x}_\alpha - \omega * \vec{x}| \\ \vec{d}_\beta &= |\vec{c}_1 \cdot \vec{x}_\beta - \omega * \vec{x}| \\ \vec{d}_\delta &= |\vec{c}_1 \cdot \vec{x}_\delta - \omega * \vec{x}| \end{aligned} \quad \text{(IV.61)}$$

The following equation is for combining the variants of PSO and GWO:

$$\begin{aligned} v_i^{k+1} &= \omega * \left(v_i^k + c_1 r_1 (x_1 - x_1^k) + c_2 r_2 (x_2 - x_i^k) + c_3 r_3 (x_3 - x_i^k) \right) \\ x_i^{k+1} &= x_i^k + v_i^{k+1} \end{aligned} \quad \text{(IV.62)}$$

IV.4.2.Whale Optimization Algorithm

This Algorithm consists of three stages [114,115]:

Bubble-net attacking method:

Using the bubble-net strategy, the humpback whales attack the prey. This is shown in the following two methods [114,115]:

- Shrinking encircling mechanism: It is applicable by reducing the amount of l in Equation (III.40).
- Spiral updating position: The mathematical description of the helix-shape motion of humpback whales is:

$$\vec{X}(n+1) = \vec{D'} \cdot e^{bv} \cdot \cos(2\pi v) + \vec{X}^*(n) \quad \text{(IV.63)}$$

$$\vec{D'} = \left|\vec{X}^*(n) - \overrightarrow{X(n)}\right| \quad \text{(IV.64)}$$

A probability of 50% to choose either a shrinking mechanism or a helical path down below the mathematical formula:

$$\vec{X}(n+1) = \begin{cases} \vec{X}^*(n) - \vec{A} \cdot \vec{D} \text{ if } p < 0.5 \\ \vec{D'} \cdot e^{bv} \cdot \cos(2\pi v) + \vec{X}^*(n) \text{ if } p > 0.5 \end{cases} \quad \text{(IV.65)}$$

where, p is a random number in [0, 1].

Search for prey:

The humpback whales search randomly for prey. Exploitation phase fellow those rules :

$$\vec{D} = \left|\vec{C} \cdot \vec{X}_{rand} - \vec{X}\right| \quad \text{(IV.66)}$$

$$\vec{X}(n+1) = \vec{X}_{rand} - \vec{A} \cdot \vec{D} \quad \text{(IV.67)}$$

IV.4.3.Hybrid grey wolf whale optimizer (HWGO)

The HWGO algorithm aims to enhance the efficiency of the WOA algorithm by applying the leadership hierarchy of GWO [116]. The next step is to apply the outcome to the WOA attacking strategy. The HWGO selected the three best candidate solutions: the 1st, 2nd, and 3rd levels in the group are alpha (a), beta (b), and delta (d).

The mathematical model for updating the position of whales using the leadership hierarchy of GWO during optimization is given as [116]:

The updating of the position of whales using the hierarchy leadership of GWO is expressed by Equation (III.42, III.43 and III.4).

The updating of the position along of a spiral-shape path of humpback whales is as follows:

$$\begin{cases} \vec{D}'_{\alpha} = \left|\vec{X}_{\alpha}(n) - \vec{X}\right| \\ \vec{D}'_{\beta} = \left|\vec{X}_{\beta}(n) - \vec{X}\right| \\ \vec{D}'_{\delta} = \left|\vec{X}_{\delta}(n) - \vec{X}\right| \end{cases} \quad \text{(IV.68)}$$

$$\begin{cases} \vec{X}_1(n) = \vec{X}_{\alpha}(n) + \vec{D}_{\alpha} \cdot e^{bv} \cdot \cos(2\pi v) \\ \vec{X}_2(n) = \vec{X}_{\beta}(n) + \vec{D}_{\beta} \cdot e^{bv} \cdot \cos(2\pi v) \\ \vec{X}_3(n) = \vec{X}_{\delta}(n) + \vec{D}_{\delta} \cdot e^{bv} \cdot \cos(2\pi v) \end{cases} \quad \text{(IV.69)}$$

$$\vec{X}(n+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \quad \text{(IV.70)}$$

IV.5.Results and Discussion

This investigation adopted the use of various analyses such as convergence curve, the step response of linear velocity and orientation, and trajectory tracking in order to validate the performance and robustness of the suggested HWGO algorithm.In comparison to four optimization algorithms under the name of PSO, GWO, WOA, and HPSOGWO.

All simulation experiments of this study were performed on a PC independently, which had the following configuration: operating system, 64-bit Windows 10; CPU, Intel Core i5-4200 2.30 GHz, and simulation software, MATLAB R2019. To measure the robustness of the compared meta-heuristics in solving this problem we run more than 30 simulation experiments for each algorithm. Moreover, in order to study the convergence behavior of the algorithms, the maximum number iterations and the population size are set to 100 and 20, respectively.

IV.5.1.Convergence Curve of the Algorithms

Fig IV.5. presents the convergence curve of the five selected optimizations algorithms. That's proved the over-performance of the HWGO algorithm. The obtained value of ISE for all algorithms used in the study is shown in Table IV.1.

From fig IV.5. and Table 1, it quite obvious that the PSO fells easy in local minimum, with a cost value (0.0504) compare to GWO that has obtained a better object function value (0.0477). The HPSOGWO could solve the problem by avoiding the local minimum related to PSO, and could reach a better precision than the GWO (0.0464) . The WOA optimizer also as the PSO optimizer fells in the local minimum with the worst ISE value obtained (0.0533). However, this algorithm takes the less number of iterations (need only 10 iteration) compare to the previous mentioned

approached. Lastly, the proposed HWGO algorithm that benefited from the rapidity of the WOA algorithm, and the precision of the GWO algorithm, could obtain the best cost value (0.0452).

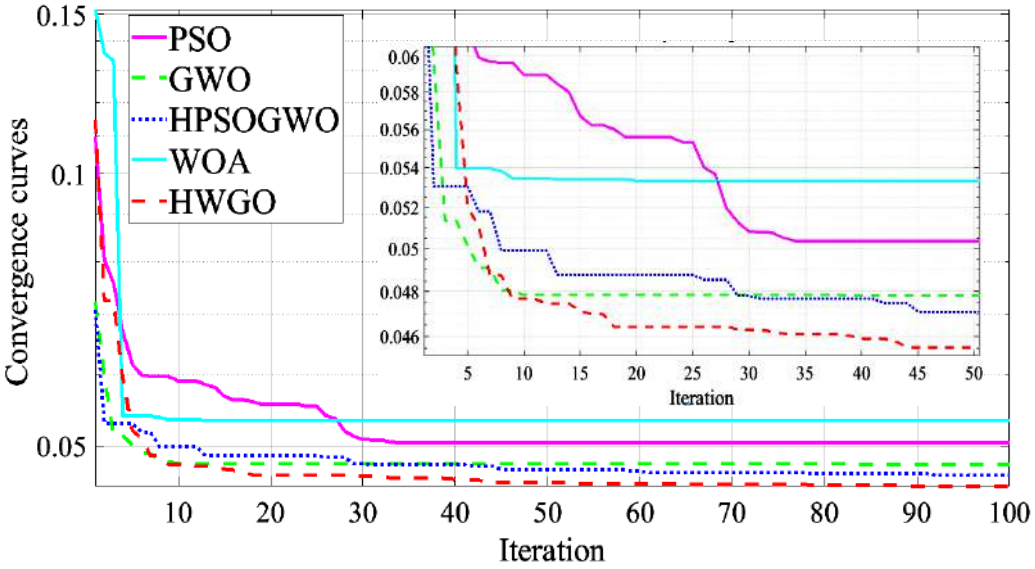


Figure IV.5. Convergence curve of five algorithms

Table IV.1. ISE values of the five algorithms.

algorithms	PSO	GWO	HPSOGWO	WOA	HWGO
ISE	0.0504	0.0477	0.0464	0.0533	0.0452

In summary, the HWGO algorithm has the lowest ISE amount with the highest rapidity in terms of convergence. It is noted that the algorithm only took 50 iterations to obtain the best optimal values

IV.5.2.Step Response Performance of Speed Controller

Fig IV.6. presents the comparative findings for velocity step responses that were designed with various techniques. As illustrated, the HWGO is characterized by a significant time response, as mentioned below.

From Figure IV.6., Table IV.2. and IV.3., we can say that the best overshoot belongs to HWGO followed by HPSOGWO, GWO, PSO, and WOA by an overshoot of 14.514%, 16.211%, 16.689%, 17.366%, and 19.982%, respectively. The algorithms can classify from worst rise time to the best rise time as follows: PSO (0.0034s), GWO (0.0032s), HPSOGWO (0.0031s), WOA (0.0027s), and HWGO (0.0026s). Based on the settling time the algorithm can be classified as follows: PSO (0.4944s), GWO (0.4924s), WOA (0.3994s), HPSOGWO (0.3655s), and HWGO (0.3168s).

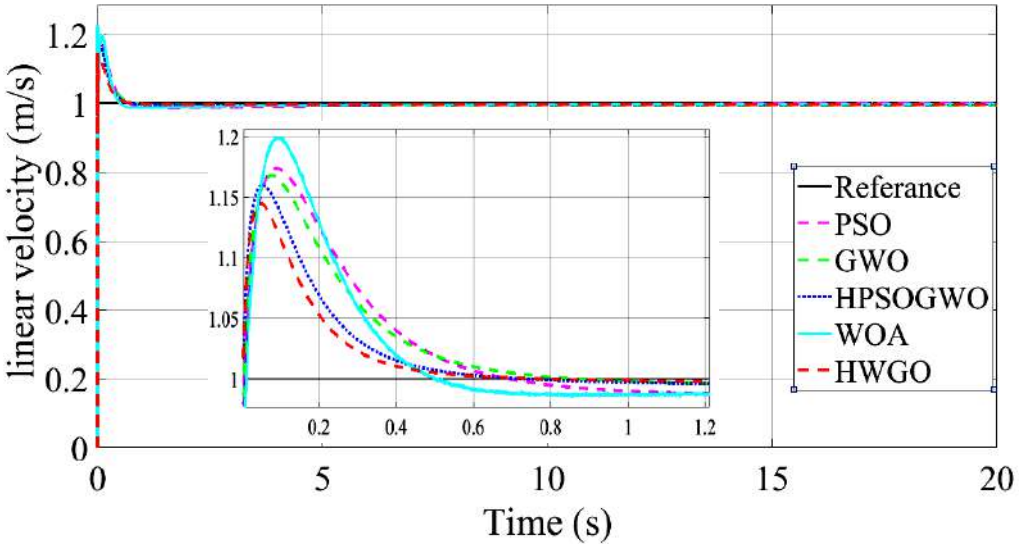


Figure IV.6. Comparison of velocity step response for all algorithms.

Table IV.2. shows the gain details of the FOPID1 controller, for the speed control of the WMR, for all chosen algorithms. While Table IV.3. shows the characteristics obtained in the time domain.

Table IV.2. The tuned gain details of FOPID1 Controller for all algorithms.

Methods	FOPID 1 for speed control				
	K_p	K_i	K_d	λ	μ
PSO	400	100	100	0.96	0.66
GWO	500	10.975	100	0.96	0.66
HPSOGWO	500	99.69	100	0.11	0.65
WOA	500	100	28.15	0.87	0.94
HWGO	500	145	149	0.10	0.56

Table IV.3. Performance characteristic for FOPID1.

Methods	FOPID 1 for speed control				
	t_r	Mp%	Peak	t_p	t_s
PSO	0.0034	17.366	1.1737	0.0890	0.4944
GWO	0.0032	16.689	1.1669	0.0761	0.4924
HPSOGWO	0.0031	16.211	1.1621	0.0509	0.3655
WOA	0.0027	19.982	1.1998	0.0926	0.3994
HWGO	0.0026	14.514	1.1451	0.0479	0.3168

In conclusion, in terms of the lowest amounts of peak percentage; overshoot (Mp%), rise time (t_r for 10% → 90%), and settling time (t_s for ±2 % tolerance), the best result was found by the proposed HWGO-FOPID1 Controller.

IV.5.3 Step Response Performance of Angle Controller

Fig IV.7. presents the comparative findings for angle step responses that were designed by applying the five optimization techniques. As illustrated, the HWGO has a significant time response.

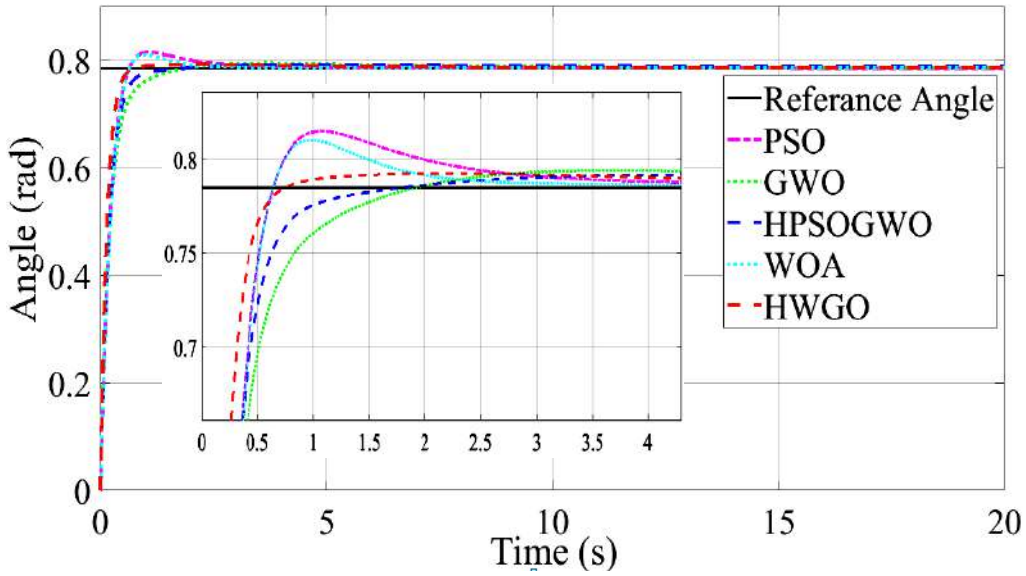


Figure IV.7. Comparison of step Response for the orientation.

Table IV.4. reveals the gain details of the FOPID2 controller for the orientation control of the WMR selected. In contrast, the comparative results of transient response are highlighted in Table IV.5. From Fig IV.7., Table IV.4. and IV.5,

Table IV.4. The tuned main parameters of FOPID2 controller for all algorithms.

Methods	FOPID 2 for Angle control				
	K_p	K_i	K_d	λ	μ
PSO	1	100	69.55	0.1	0.96
GWO	1.21	57.22	74.18	0.15	0.96
HPSOGWO	67.9	6.52	75.17	0.74	0.96
WOA	93.48	3.17	63.46	0.60	0.96
HWGO	63.97	26.4	81.95	0.36	0.95

Table IV.5. Performance characteristic for FOPID2.

Methods	FOPID 2 for speed control				
	t_r	Mp%	Peak	t_p	t_s
PSO	0.3994	3.8347	0.8151	1.0770	1.9409
GWO	0.5235	1.6377	0.7979	2.6191	0.9705
HPSOGWO	0.4017	2.3653	0.8036	3.2517	4.9513
WOA	0.4033	3.2358	0.8104	0.9832	1.4563
HWGO	0.2825	0.8464	0.7916	2.2696	0.4914

In summary, The lowest amounts of peak percentage, overshoot (Mp%), rise time (t_r for 10% → 90%), and settling time (t_s for ±2% tolerance) were also observed for the suggested HWGO-FOPID2 controller.

IV.5.4. Path Tracking

To illustrate the robustness of the Back-stepping - HWGO -FOPID controller, a star trajectory was selected using the following equation [106]:

$$x_R(n) = -2.5 * \sin\left(2 * \pi * \frac{t}{30}\right) \quad \text{(IV.71)}$$

$$y_R(n) = 2.5 * \sin\left(2 * \pi * \frac{t}{20}\right) \quad \text{(IV.72)}$$

$$\theta_R(n) = a\tan 2\left[\frac{\left(\frac{y_R(n) - y_R(n-1)}{t+\epsilon}\right)}{\frac{(x_R(n) - x_R(n-1))}{t+\epsilon}}\right] \quad \text{(IV.73)}$$

Where the gains are chosen as: k_x =10, k_y =80 and k_θ=15.

The reference and actual trajectories of the star path are shown in Figure IV.8, where blue and red lines, respectively, represent them. Figs IV.9a-9c display the desired and actual tracks for x, y, and θ, respectively. The reference and actual amounts of the linear and angular speeds (v and w) are presented in Figs IV.10a. and IV.10b. However, Figures IV.11a-11f. reveal the errors of the v, w, x, y, θ, and the tracking trajectory, respectively.

The HWGO-FOPID controller was designed to handle with the dynamics ,which means a kinematic controller is necessary ,therefore a back-stepping controller was proposed to guarantee a minimum of distance error that already has been proved in fig IV.11f. (error less than 0.004 m),this result demonstrate the reason behind the supposition of actual with the reference trajectories in

figures (IV.9-11). The addition of this controller in the external loop had influenced by decreasing the error of linear and angler velocity (the ISE of the HWGO) of the dynamic controller from 0.0452 to less than 0.002 as shown in figs IV.11a. and IV.11b.

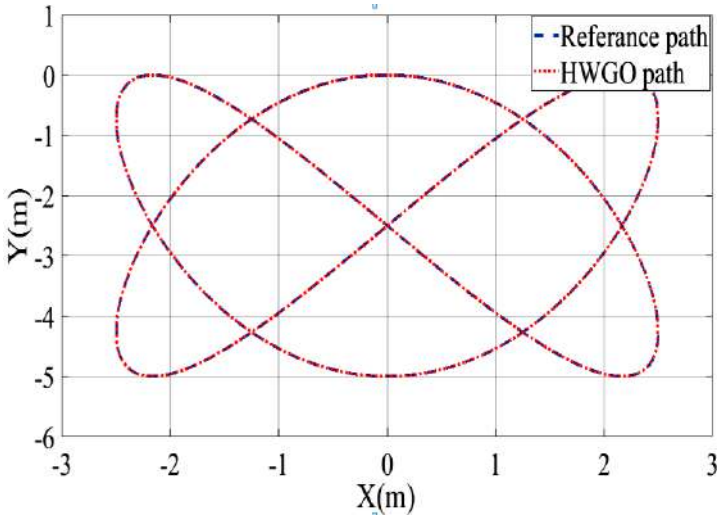
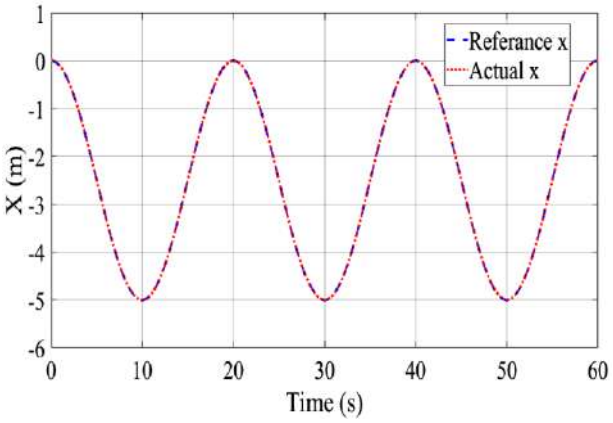
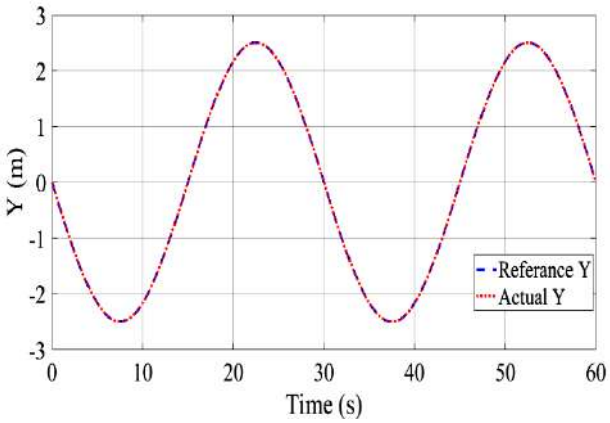


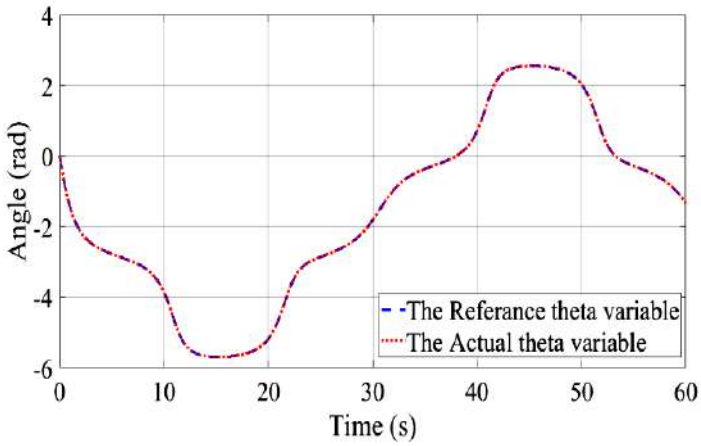
Figure IV.8. Star trajectory.



(a)

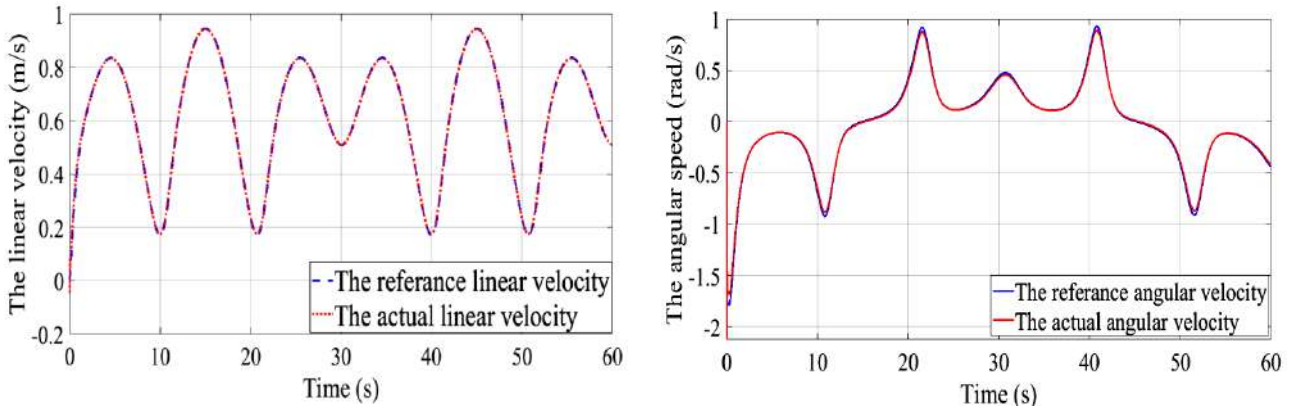


(b)



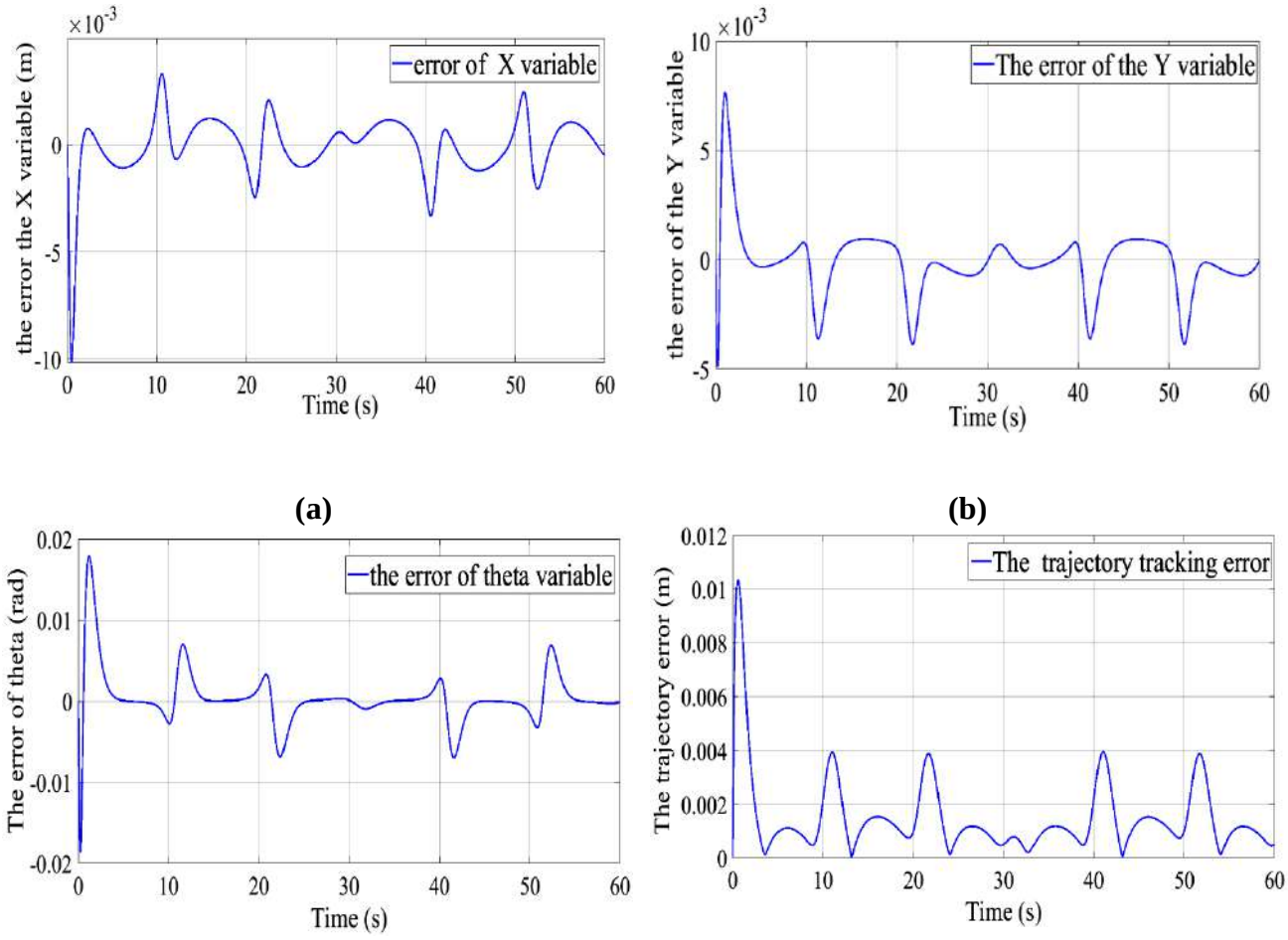
(c)

Figure IV.9. Response of the (a) X; (b) Y; (c) θ variable.



(a) **(b)**

Figure IV.10. Response of (a) linear; and (b) angular Velocities (for star shape).



(a) **(b)**

(c) **(d)**

Figure IV.11. Error corresponding to (a) x, (b) y and (c) theta variables; (d) the trajectory tracking.

IV.6. Conclusions

This chapter presents an advanced optimization method to obtain the turning parameters of fractional order proportional integral derivative controller (FOPID controller). A comparison study was performed for the first time in the robotics field, where several tuning methods have been employed to set the details of two FOPID controllers, which are designed to make sure that the linear and angular speeds of the WMR follow the input reference. Those five meta-heuristics approached namely particle swarm optimizer (PSO), grey wolf optimizer (GWO), hybrid particle swarm grey wolf optimizer (HPSOGWO), whale optimizer (WOA) and hybrid grey wolf whale optimizer (HWGO) were found to be all statistically and graphically approved, by considering integral square error ISE as a cost function .A back-stepping kinematic controller was adopted to guarantee that tracking errors convergence to zero to achieve the trajectory tracking. A transient response in the time domain has been made to examine the performance of selected algorithms. Simulation results have successfully confirmed the over performance of HWGO in comparison to PSO, GWO, HPSOGWO, and WOA in terms of lowest error value obtained, convergence rapidity, with the minimum overshoot, received rise time, peak, peak time, and settling. Finally, a star trajectory demonstrated the robustness of the mentioned controller .

Chapter V:

Optimal ANFIS-FOPID controller design

Chapter V

Optimal ANFIS-FOPID Controller Design

V.1.Introduction

This chapter suggests a novel control technique for solving the trajectory tracking problem of wheeled mobile robot (WMR), a design of an optimal fractional adaptive Neuro- Fuzzy inference system (ANFIS-FOPID) controller for dealing with the dynamics of the WMR. Tuning parameters of the controller are tuned using the (HWGO). Simulation results revealed the highest efficiency of the proposed ANFIS-FOPID technique compared to the (FOPID) controller and the (ANFIS) controller in terms of settling and rise time, overshoot, as well as steady-state error. Many cases study (circular path, zigzag path, five shape path) have highlighted the over performance of mentioned controller. Finally, to accomplish our work a back-stepping control technique is added for the kinematic control, a quadrifolium and a square trajectory were made to illustrate the capability of the mentioned controller.

V.2. Controller Design and Strategy

Fig V. 1 shows the schematic form of the control approach based on the dynamic and kinematic of the wheeled mobile robot. The trajectory generator supplies the desired coordinate x, y and θ,

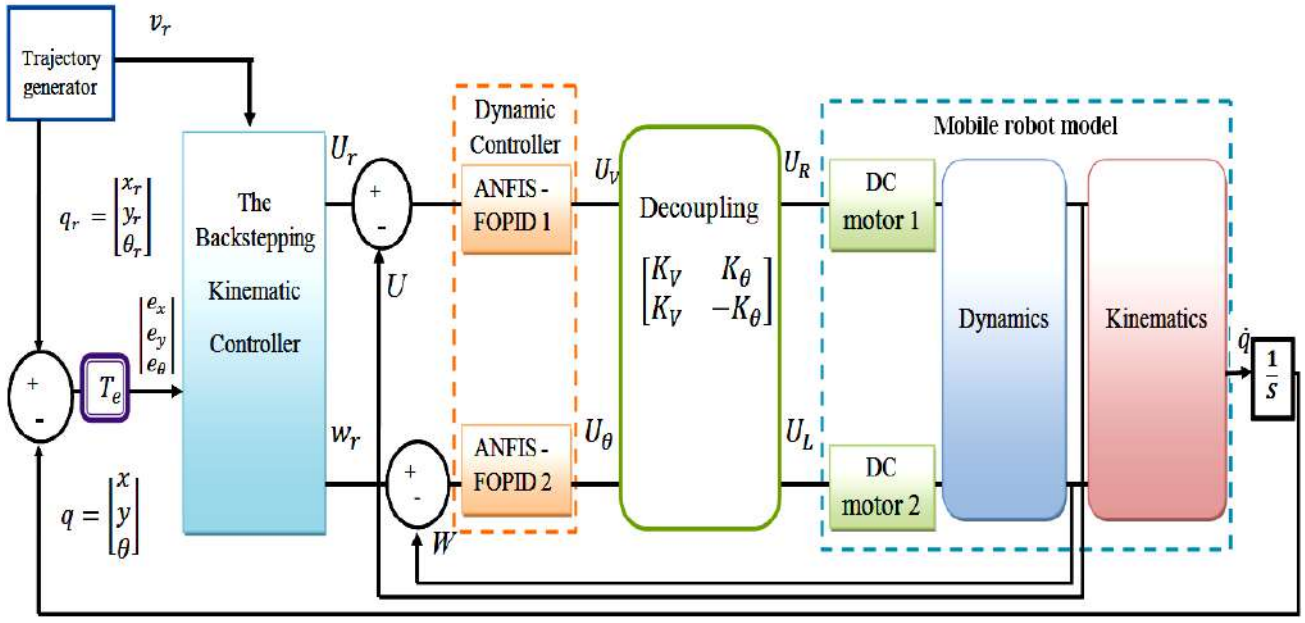


Figure V.1. Control scheme of WMR.

A back-stepping kinematic controller is detected in the external loop, here the controller examines the difference between the reference data (obtained from the bloc trajectory generator) and physical value from the robot then develop his own angular and linear velocities that are delivered into the internal loop, where two blocks of ANFIS-FOPID controller deal with the dynamics. Here, the two dynamic controllers provide their own signals of angular and linear speeds based on the values taken from the kinematic controller

V.2.1 Kinematic Controller Design

Like the previous works , here the controller structure is highlighted in Figure V.1, where the input error and velocity vector (v_c) are:

$$\begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix} = \begin{bmatrix} cos\,\theta & sin\,\theta & 0 \\ -sin\,\theta & cos\,\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_r - & X \\ Y_r - & Y \\ \theta_r - & \theta \end{bmatrix} = T_e e_r \qquad \text{(V.1)}$$

$$v_c = \begin{bmatrix} v_r cos(e_\theta) + k_x e_x \\ w_r + k_y v_r e_y + k_\theta v_r sin(e_\theta) \end{bmatrix} \qquad \text{(V.2)}$$

where k_x, k_y and k_θ are the tuning parameters.

V.2.2. Designing the ANFIS-FOPID Dynamic Controller

The implementation of the full control strategy that handles the dynamics of the WMR based on the ANFIS-FOPID, can be divided into three steps:

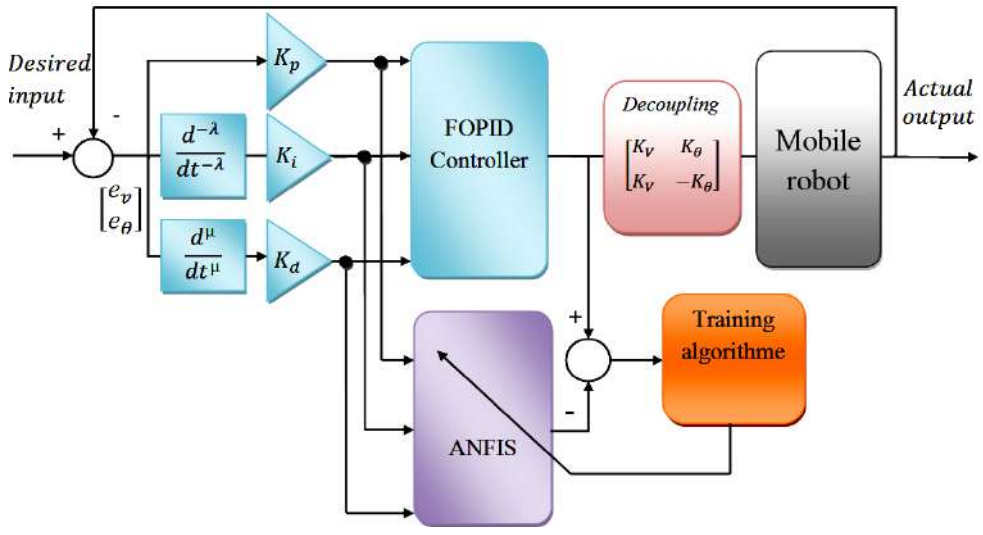


Fig. V.2 The ANFIS in training phase.

- ***First step***

The first stage is represented by Fig. 8, which shows the training phase of the ANFIS controller. The ANFIS utilise their adaptive and learning capacities to learn and to predict the best required

control actions based on the available data. These data are obtained from the already designed FOPID controllers. The gained training data were stored in a file. From MATLAB/SIMULINK using the command anfisedit, these data were taken for training, the complete operation was represented in Fig. V.3

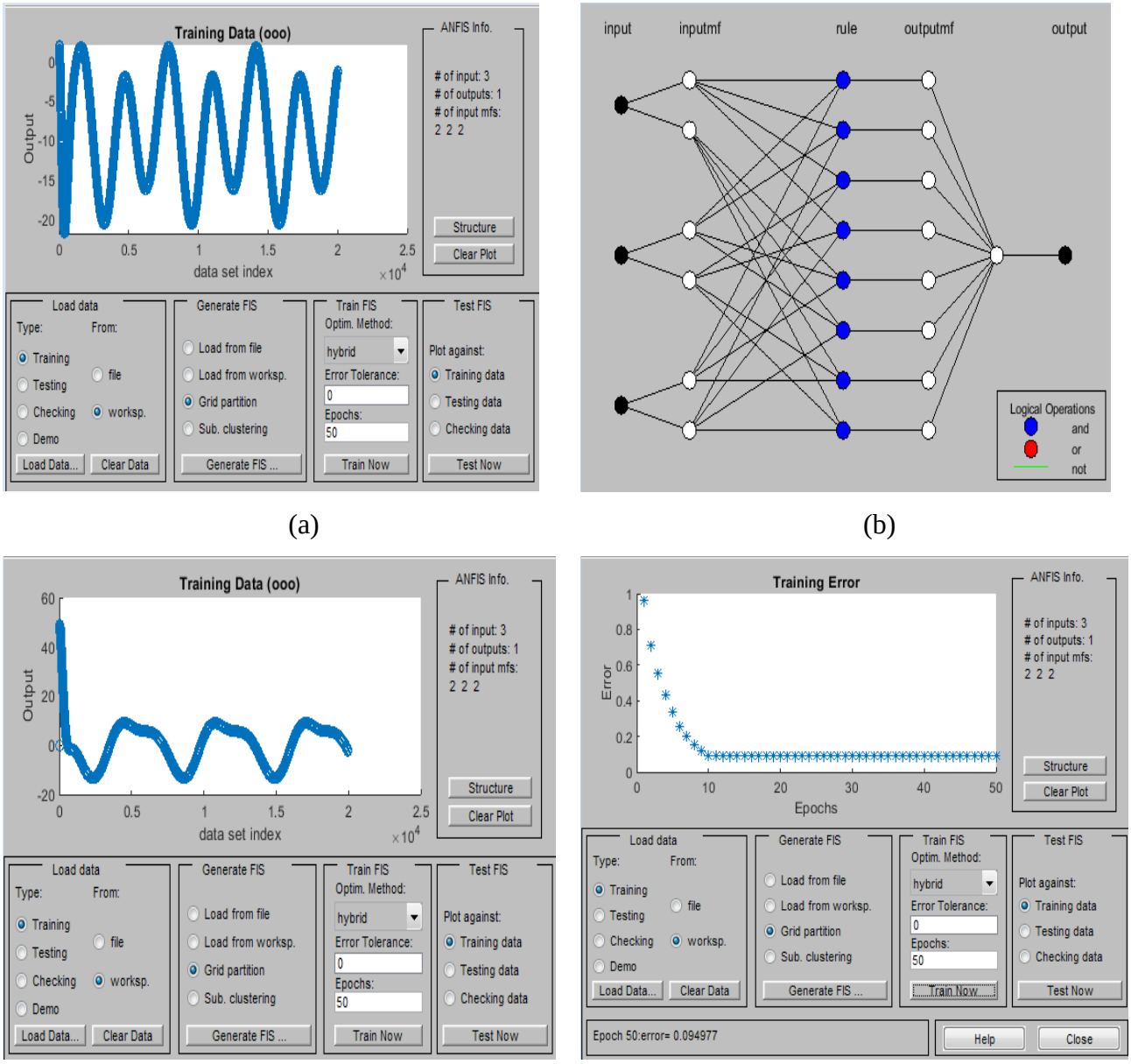


Fig.V.3.(a) data configuration for first ANFIS-FOPID controller,(b) structure of the ANFIS-FOPID controller (c) data configuration for second ANFIS-FOPID controller ,(d) error minimization corresponding to the second controller.

Fig.V.3 shows the Matlab Simulink GUI window of the internal architecture of the ANFIS controller. The ANFIS training used hybrid training algorithm, with the input nodes (2, 2, 2),using the gaussmf membership function for the inputs and linear membership function for the output,

each having 8 rules (see Fig.V3.b), the epoch length was used is 50 iterations for each sample, with, 0.01s as the Simulink sampling time. At the end of the training the result was saved as a *fis* file with respect to Sugeno-style.The root mean square error (RMSE) was found to be 0.1053 and be 0.094977 for ANFIS-FOPID1 and ANFIS-FOPID2, respectively.

- ***Second step:*** adjusting the parameters using the HWGO optimizer.

After the training phase is complete and the optimal value of the data are obtained, the designed ANFIS-FOPID controller should be ameliorated by adjusting the gains, i.e.,K_p, K_I, K_d, λ, μ, Kv, K_θ as shown in Fig.V.4

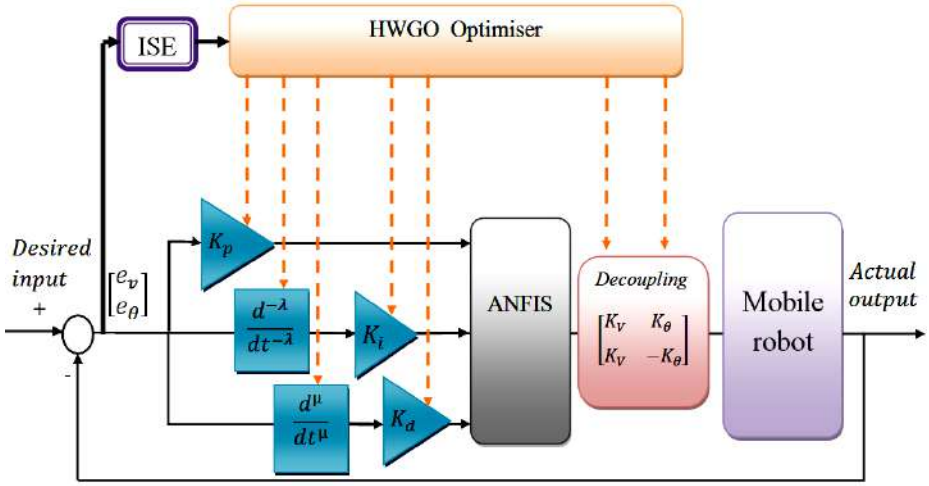


Fig.V.4 Bloc diagram for parameters tuning.

To select the optimal values of the scaling gains, hybrid whale grey wolf method is introduced to reduce the error value. The cost function used in the study is the integral square error (ISE) which is demonstrated by equation (III.14).

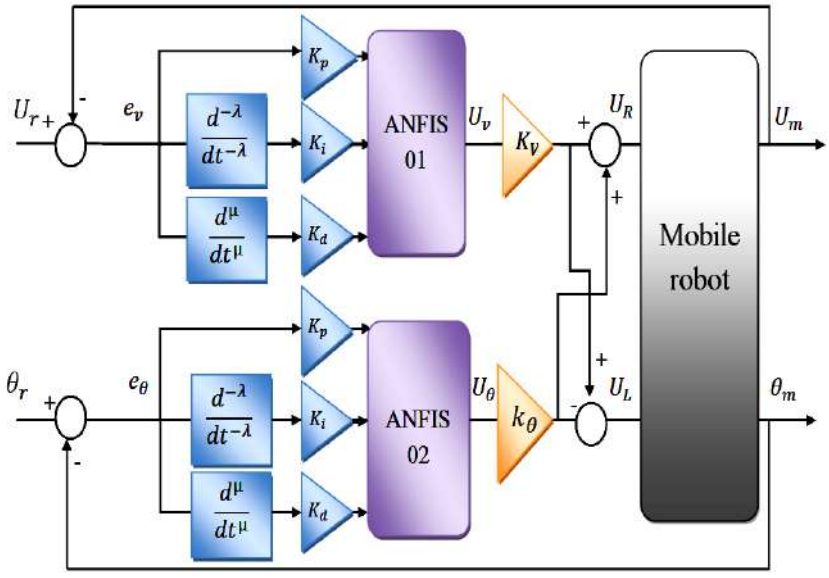


Fig.V.5 Bloc diagram for Hybrid ANFIS-FOPID dynamic controller.

- ***Third step:***

The obtained control strategy of the dynamic controller is represented in Fig.V.5 It illustrates a combination between the ANFIS and the fractional order PID controller, abbreviated as ANFIS-FOPID. Two trained ANFIS-FOPID controllers are adopted for driving the right and left DC motor voltage of the WMR, where a decoupling system is used for separating the two control actions.

The first controller selects the difference between the desired and reference velocity from the right wheel as an input. The second controller in the left wheel examines the difference between the actual and desired orientations as an input.

The following equations (V.3) represent the relation between the input voltage of the right and left DC actuators U_R and U_L, respectively, with the outputs of the first and second controller U_V and U_θ, respectively, and the coefficients of the decoupling process are respectively K_v and K_θ.

$$\begin{cases} U_R = K_V \cdot U_V + K_\theta \cdot U_\theta \\ U_L = K_V \cdot U_V - K_\theta \cdot U_\theta \end{cases} \quad \text{(V.3)}$$

where

$$\begin{cases} e_v = U_r - U_m \\ e_\theta = \theta_r - \theta_m \end{cases} \quad \text{(V.4)}$$

U_r is the desired velocity, U_m the actual velocity, e_v the velocity error, θ_r the desired orientation, θ_m the measured orientation, and e_θ is the orientation error.

V.3. Results and Discussion

This study adopted the use of several evaluations, such as convergence curve, the step response of linear velocity and orientation, also various case study has been made in tracking trajectory, in order to justify the performance and robustness of the proposed HWGO-ANFIS-FOPID controller, In comparison to HWGO-FOPID and the ANFIS.

All simulation experiments of this study were performed on a PC independently, which had the following configuration: operating system, 64-bit Windows 10; CPU, Intel Core i5-4200 2.30 GHz, and simulation software, MATLAB. To measure the robustness of the compared meta-heuristics in solving this problem, we run more than 30 simulation experiments for each algorithm. Moreover, in order to study the convergence behaviour of the algorithms, the maximum number iterations and the population size are set to 100 and 20, respectively.

V.3.1. Convergence Curve of the Algorithms

Fig.V.6 presents the convergence curve of the three selected optimizations algorithms from the designed FOPID controller, compared to our proposed controller. The result proves the over-

performance of the HWGO-ANFIS-FOPID controller. The obtained value of ISE for all algorithms used in the study is shown in Table V.1.

From Fig.V.6 and Table V.1, it quite obvious that the HWGO optimizer gives the lowest ISE value compared to the whale optimizer (WOA) and the grey wolf optimizer (GWO).The ANFIS-FOPID controller has the lowest ISE value (0.0268). For that purpose, the next cases studies are focusing only on comparing the three controllers (FOPID, ANFIS and ANFIS-FOPID) using only the HWGO optimizer.

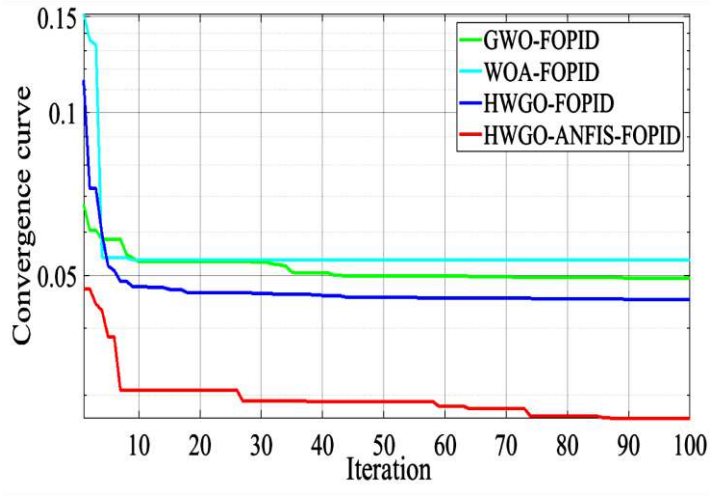


Fig.V.6. Convergence curve of the algorithms.

Table V.1. ISE values of all controllers.

algorithms	**ISE**
GWO-FOPID	0.0477
WOA-FOPID	0.0533
HWGO-FOPID	0.0452
HWGO-ANFIS-FOPID	0.0268

V.3.2. Step Response Performance of Speed Controller

Fig V.7 presents the comparative findings for the linear velocity step responses that were designed with various controllers. Table V.2 shows the gain details of all controllers, for the speed control of the WMR, tuned by HWGO optimizer. Table V.3 shows the characteristics obtained in the time domain. The HWGO-FOPID controller only advantage is the fast rise time obtained, but a high overshoot and slow settling time. The ANFIS controller major advantage is the zero-overshoot obtained value and an acceptable settling time but slow rise time. The HWGO-ANFIS-FOPID controller mixes the two advantages of the previous controllers by giving the best settling time and an acceptable overshoot and rise time.

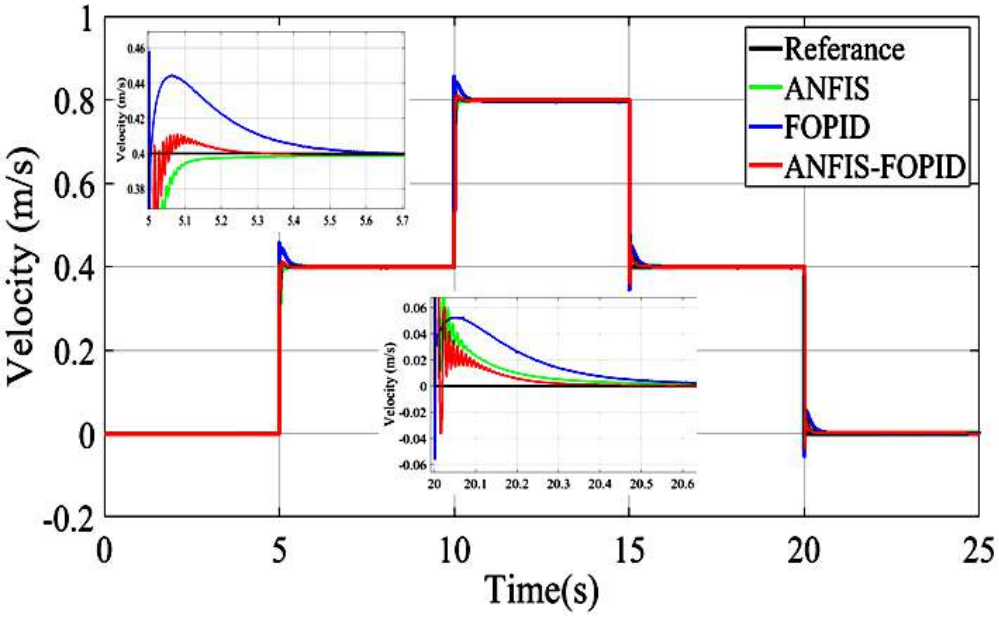


Fig V.7. Step response Comparison for all velocity controllers.

Table V.2. The tuned gain details of linear velocity for all Controllers

Methods	Speed controller						
	K_p	K_i	K_d	λ	μ	K_V	K_θ
HWGO-FOPID	500	145	149	0.10	0.56	0.50	0.50
ANFIS	-	-	-	-	-	-	-
HWGO-ANFIS-FOPID	2	1.01	1	0.2046	0.8674	1.046	0.75

Table V.3. Performance characteristic for velocity Controller.

Methods	Caracteristics of the speed controllers				
	t_r	Mp%	Peak	t_p	t_s
HWGO-FOPID	0.0026	13.1268	0.4525	0.0014	0.2182
ANFIS	0.0451	0	0.3993	10	0.0562
HWGO-ANFIS-FOPID	0.0242	2.4350	0.4097	0.1000	0.0357

V.3.3. Step Response Performance of Angle Controller

Fig. V.8 presents the comparative findings for angle step responses that were designed using the three controllers, by applying the HWGO optimization techniques. Table V.4 reveals the gain details of all controllers selected for the orientation angle control of the WMR. In contrast, the comparative results of transient response are highlighted in Table V.5.

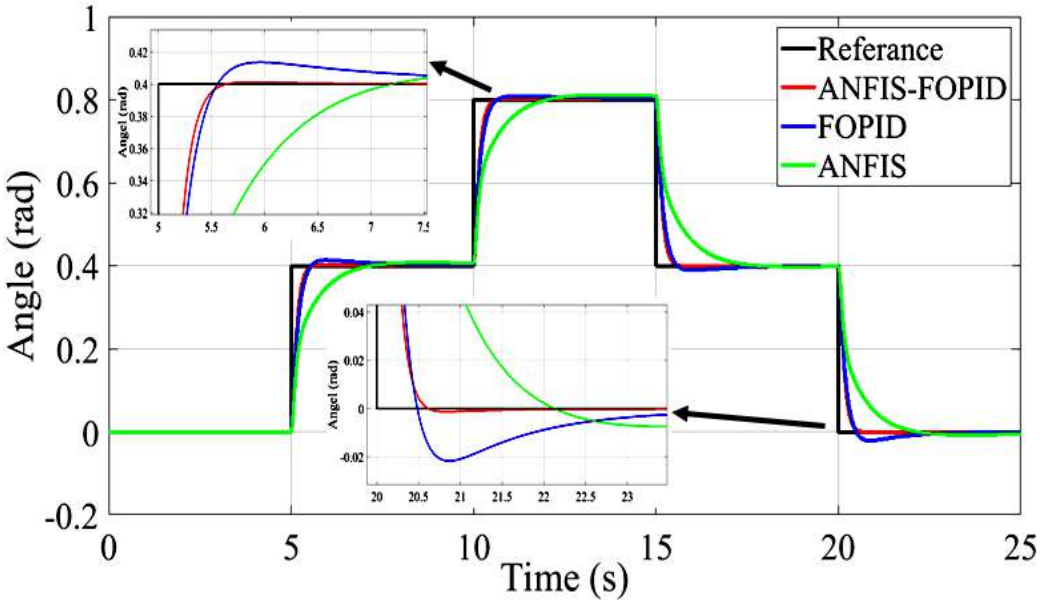


Fig V.8. Comparison of step Response for the orientation.

Table V.4 The tuned main parameters of the three controllers for angle control.

Methods	Angle controller parameter tuning						
	K_p	K_i	K_d	λ	μ	K_V	K_θ
HWGO-FOPID	63.97	26.4	81.95	0.36	0.95	0.50	0.50
ANFIS	-	-	-	-	-	-	-
HWGO-ANFIS-FOPID	6.00	4.92	1.21	0.02	1	1.046	0.73

Table V.5. Performance characteristic of the three controllers for angle control.

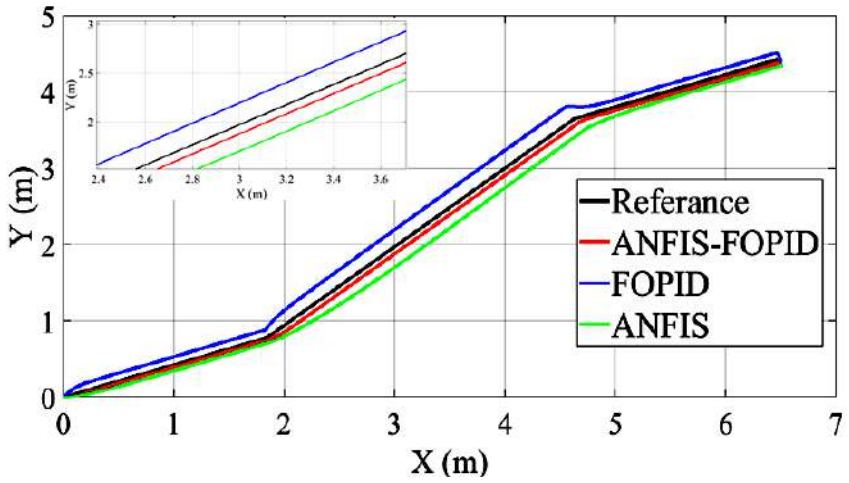
Methods	Transition parameter for angle controller				
	t_r	Mp%	Peak	t_p	t_s
HWGO-FOPID	0.4184	3.4084	0.4136	0.9600	0.4309
ANFIS	1.6633	2.2710	0.4091	4.0512	1.6942
HWGO-ANFIS-FOPID	0.3470	0.1822	0.4007	0.7466	0.3770

From Fig.V.8, Table V.4 and Table V.5, we can say that the lowest amounts of peak percentage, overshoot (Mp%), rise time (t_r for 10% → 90%), and settling time (t_s for ±2% tolerance) were observed for the suggested HWGO-ANFIS-FOPID controller.

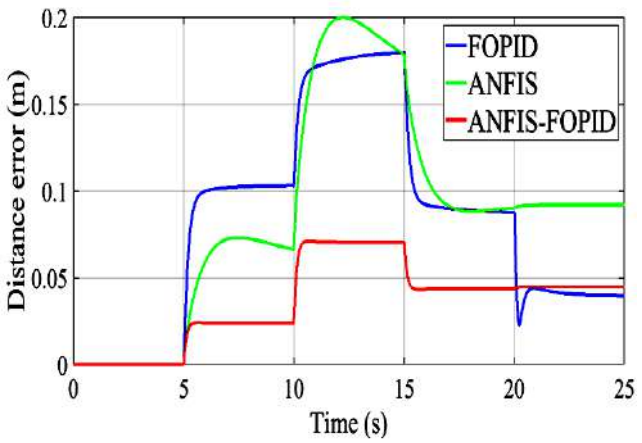
V.3.4. The trajectory coresponding to Step Response in XY plane

In this case, a linear trajectory is aplied acording to the step response. Fig. V.9 (a) shows the relationship between the desire path and the actual path based on each cotroller,where Fig. V.9 (b), and 9(c) shows the error coresponding to the distance and the orrientation angle for each techniques, respectivly.

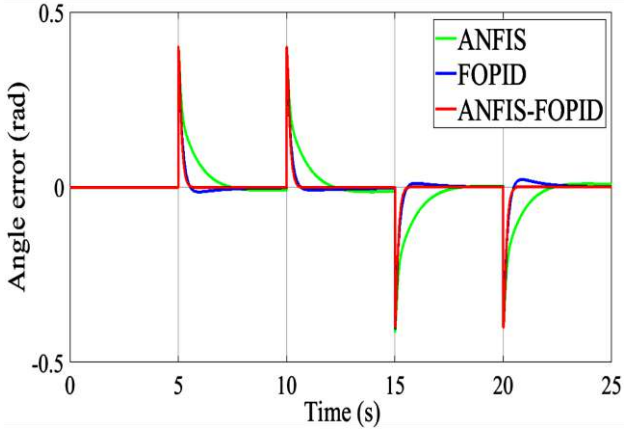
The lowest distance and orientation error was found by the proposed hybrid controller folowing by the HWGO-FOPID controller and the ANFIS controller,the FOPID controller is faster in termes of rise time value which make him sensitive for istantaneus changing of the linear and angular velocity but a high overshoot and an important distance error,contrary to the ANFIS controller, having a slow action reponce to instantaneus changing and also an important distane error butnearly zero overshoot, to handle this problem,the Hybride ANFIS-FOPID gives a very acceptable transition responce and less tracking error and an acceptable overshoot wich make him the ideal controller for strait line trajectory.



(a)



(b)

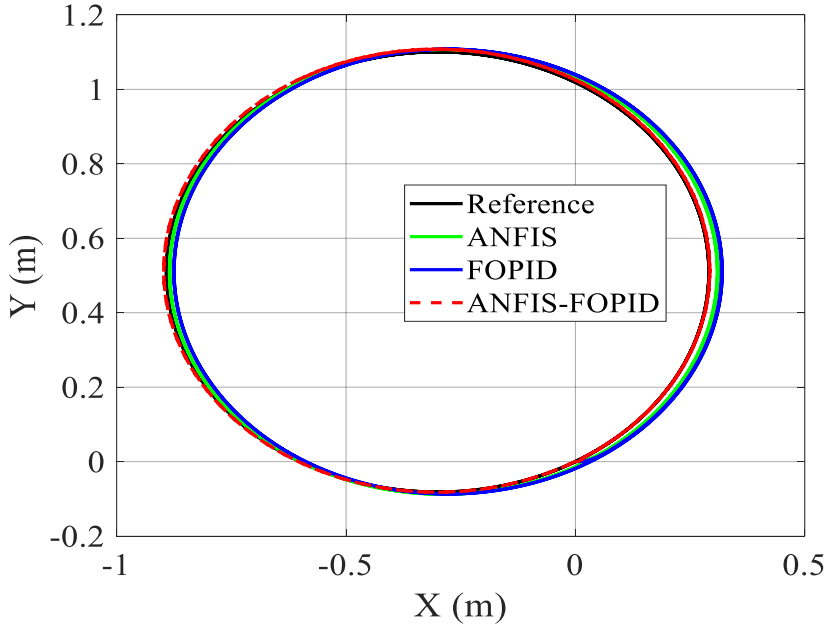


(c)

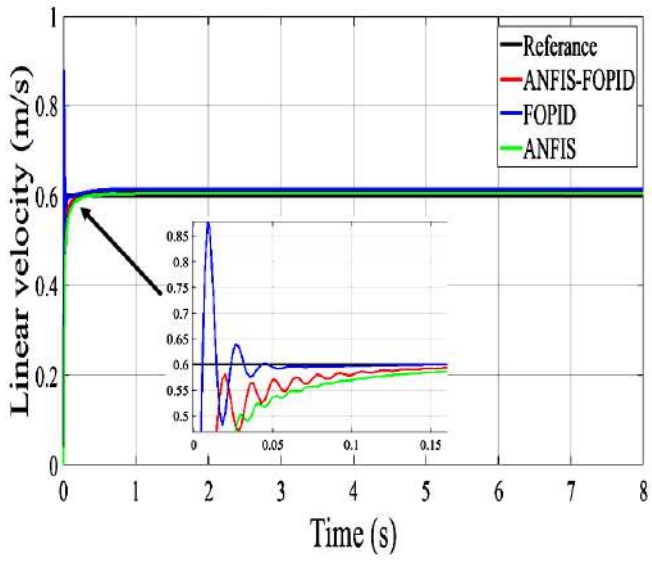
Fig.V.9 (a) strait trajectory in XY plane,(b) error of the distance, (c) the theta angle error.

V.3.5. Circuler path (case study 01)

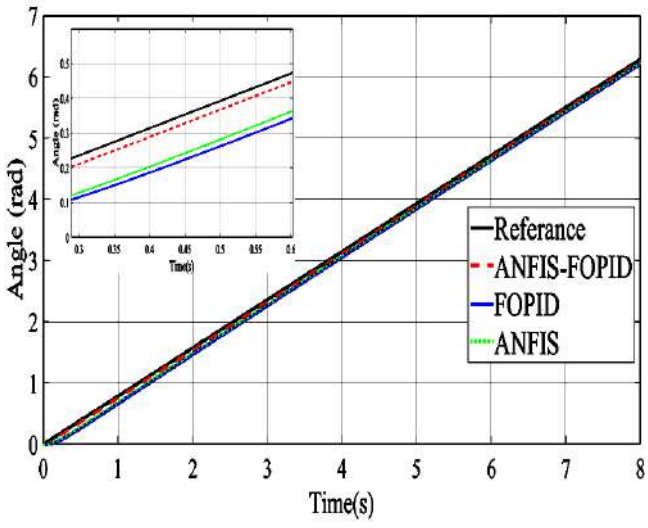
In this case,a circuler trajectory have been made to demonstrate the ability of the ANFIS-FOPID controller in handling smooth trajectory, in comparison to the other controllers. Fig.V.10 (a) to (c) ,and Fig.V.11 shows,respectivly,circuler path of all controllers in relationship to the reference path,the linear velocity response of all controllers,the orientation angle of the three controllers,and finaly the error of distance .



(a)



(b)



(c)

Fig V.10. (a) circle trajectory in XY plane,(b) linear velocity response for all controllers ,(c) angle response of all controllers

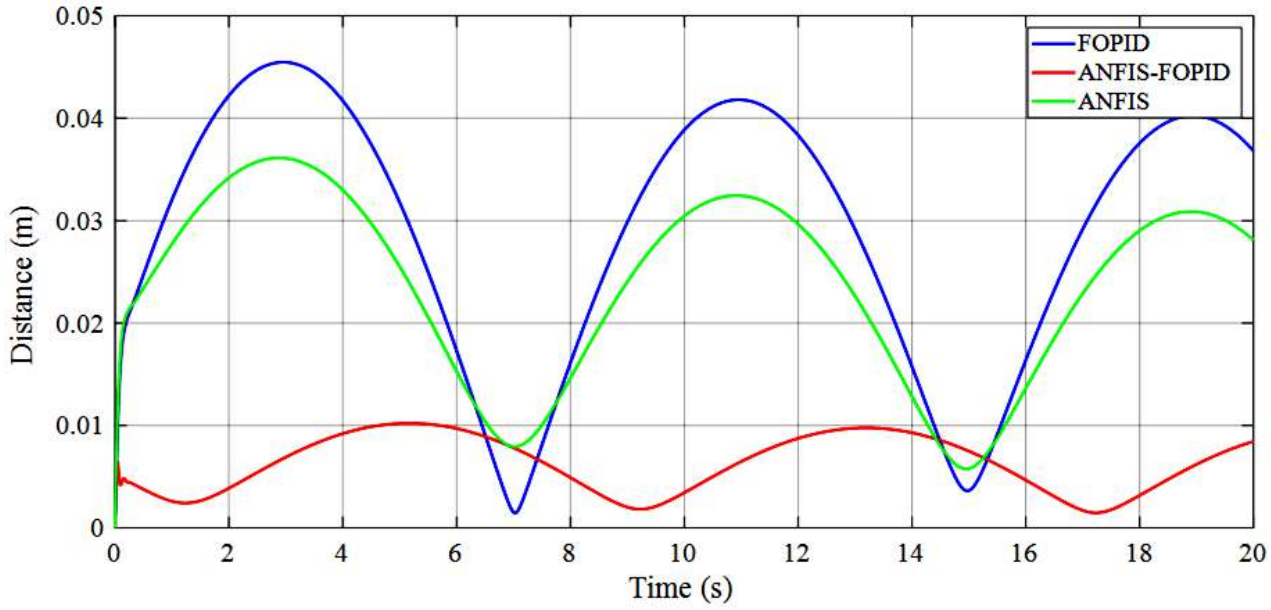


Fig V.11. error of the distance

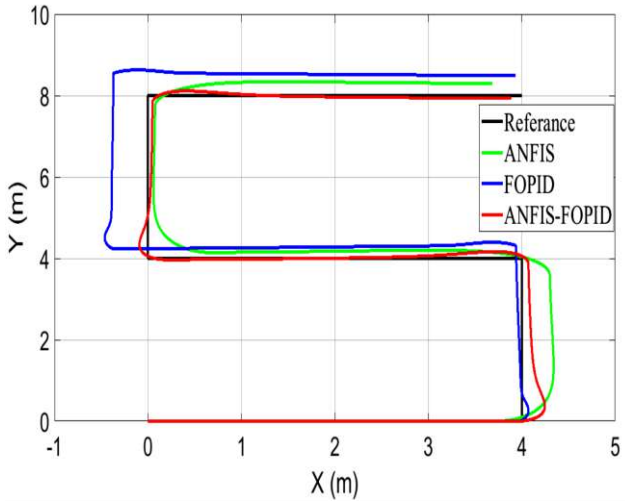
Here, the ANFIS controller can beter dealing with smooth trajectorythan the FOPID controller, due to a progressive changing in both linear and angular velocity. For that purpse, we can see the the error of the tracking trajectory was beter minimized by the ANFIS controller than the FOPID controller.

At the beginning,the FOPID controller gives a very high overshoot in the linear velocity. However, a very good rise time value and beter setling time from this controller than the other.

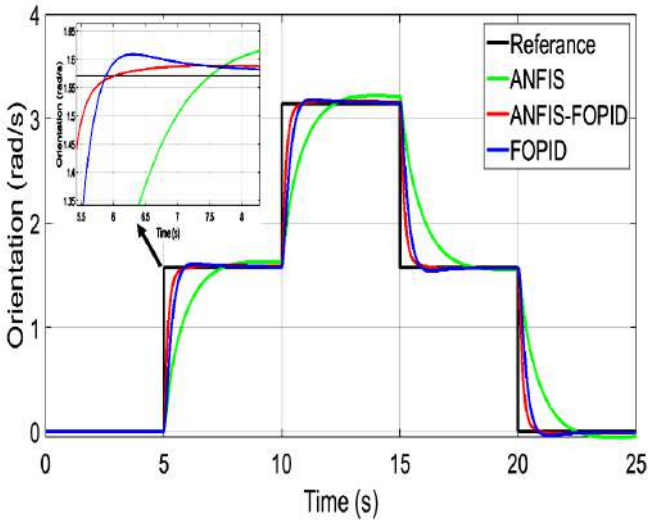
Here the proposed controller also combine the advantage of the two previous controllers and shows the capability of handling the problem related to the trajectory tracking by given the minimum distance error and orientation too.

V.3.6. Five (5) chape trajetory (case study 02)

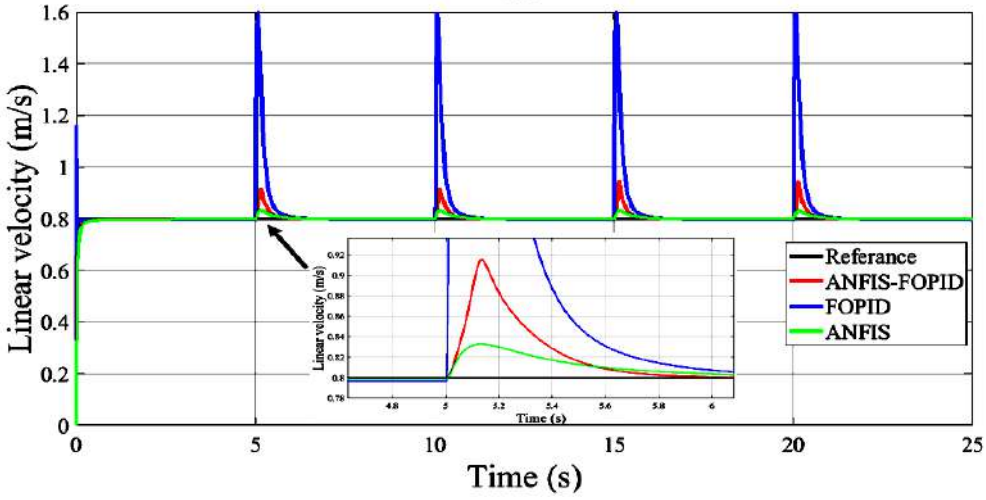
The five (5) shaped trajectory is a straightforward example of a non-continuous gradient trajectory. The major problem of such as paths is the sharp and a non-continuous motion, which requires a high-level control effort. Fig.V.12 (a)- (e) demonstrates the difference between the actual and reference response for all controllers, respectively, the trajectory in the XY plane, comparative results of the angular velocity, comparative findings of the linear speed, and the error according to the distance and orientation.



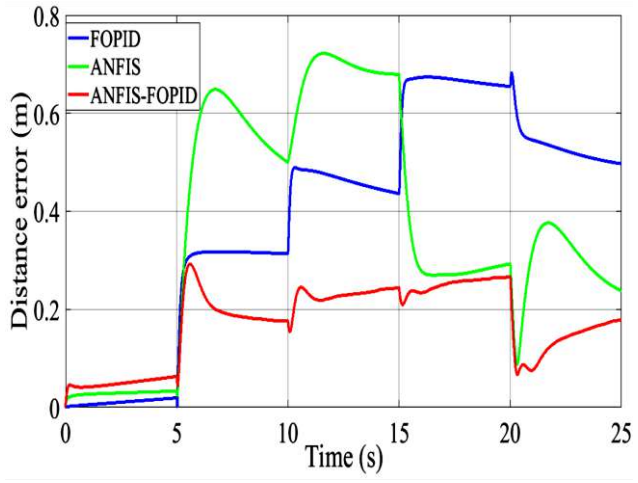
(a)



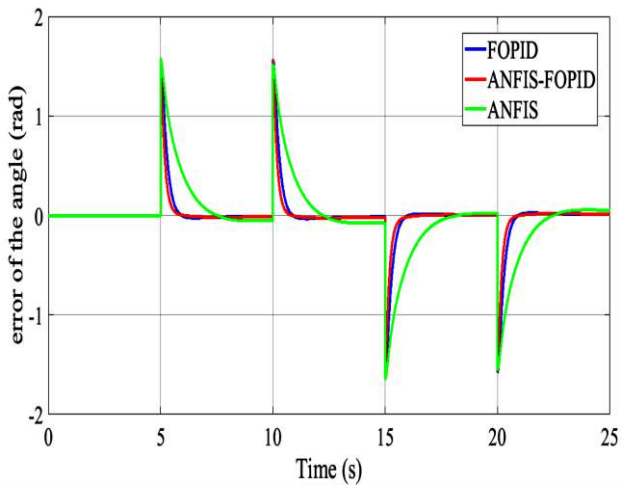
(b)



(c)



(d)



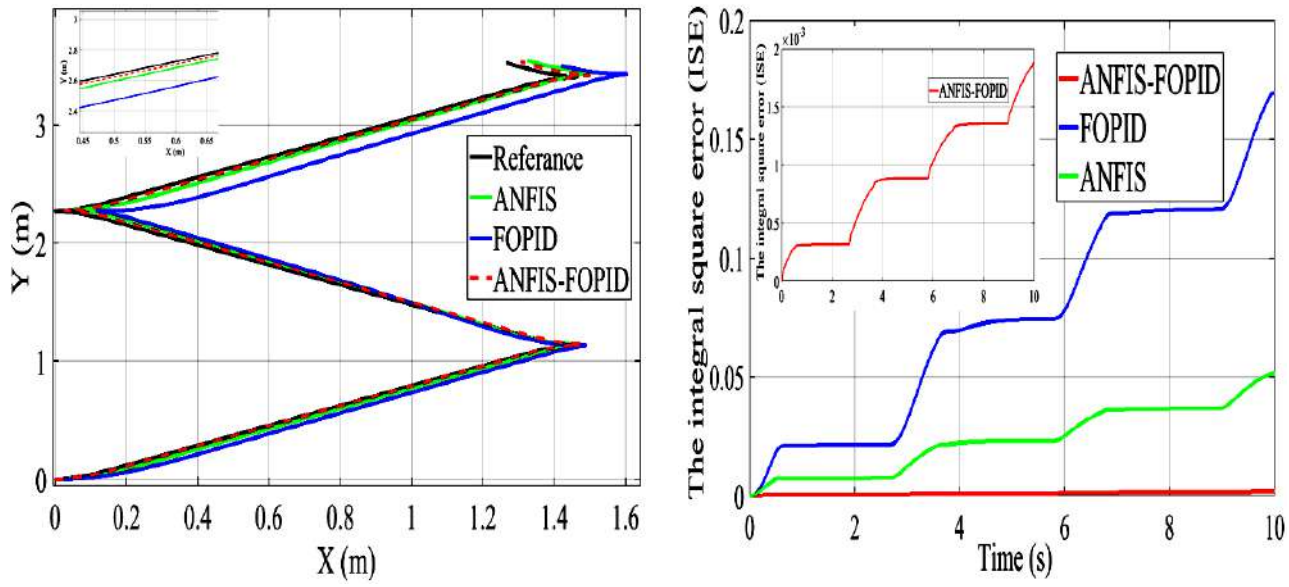
(e)

Fig.V.12. (a) five shape trajectory in XY plane,(b) angle response corresponding to the path for all controllers, (c) linear velocity corresponding to the path for all controllers, ,(d) error of the distance, (e) the theta angle error.

The 5 shaped path is classified as a sharp-shaped path, here the ANFIS controller cannot handle this type trajectory because the spontaneous changing in velocity and due to the slow rise time and settling time, makes this controller gives slow actions, that causes high distance and orientation error. But, we can see a very low overshoot value from this controller. The FOPID controller gives a very high overshoot, that might produce an unstable and a disturbed motion, which could harm the DC actuators (see Fig.V.12 (c)), which makes it unsupported for this kind of paths. The ANFIS-FOPID controller has an acceptable overshoot compared to the FOPID and faster time response than the ANFIS, which makes it suitable for this type of trajectory.

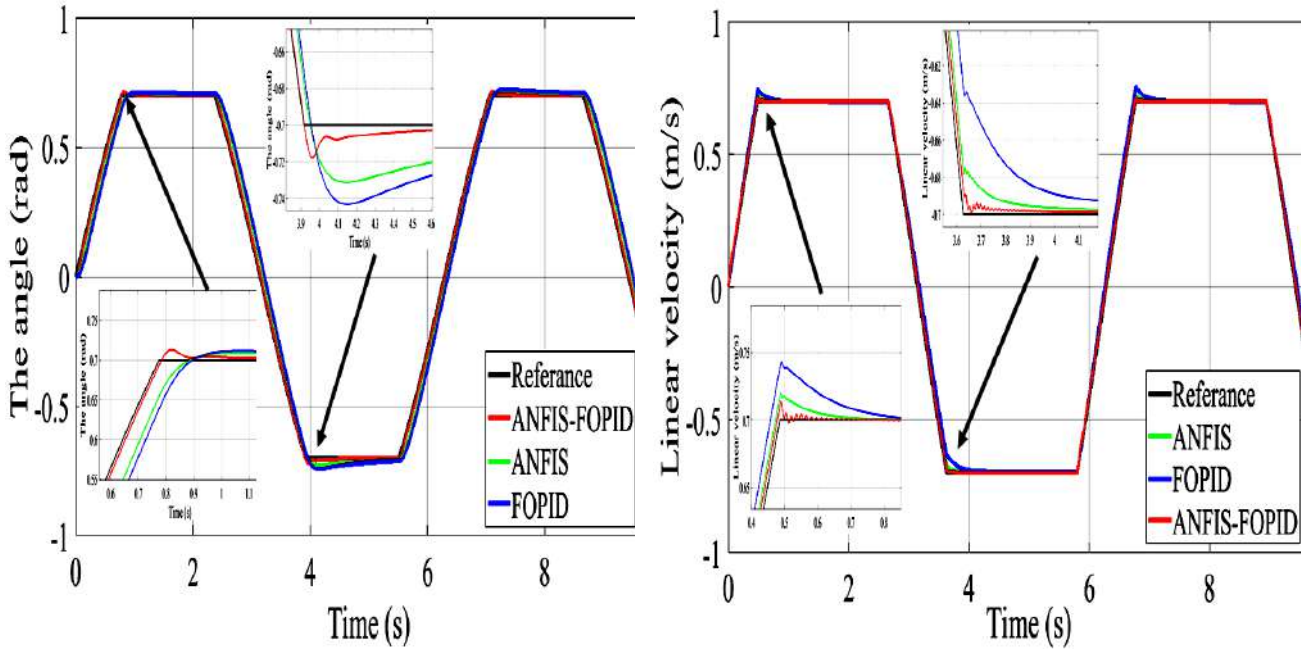
V.3.7. Zig zag trajectory (case study 03)

Here both linear and angular velocity changes progressivly (see Fig.V.14 (c), and (d)), In other words, a fast and high control effort time reponce is not nessessary. For that reason, the ANFIS controller gives a good time response ,contrary to the FOPID which gives an important overshoot and a high ISE value. The ANFIS-FOPID gives the best ISE value (see Fig.V.13 (b)); less than 0.002,compared to FOPID (0.017) and ANFIS (0.005).Moreover, less overshoot and beter time reponse. Here the ANFIS-FOPID controller mix all the advantages of its old versions which makes him ideal for this type of trajectories.



(a) (b)

Fig V.13. (a) zigzag path in XY plane ,(b) the integral square error (ISE) for all controllers



(a) (b)

Fig V.14. (a) the theta angle response corresponding to all controllers ,(b) the linear velocity corresponding to all controllers.

V.3.8. Adding the backsteping controller in the external loop

The ANFIS-FOPID controller was designed to handle with the dynamics, which need a kinematic controller. Therefore, a back-stepping controller was proposed to guarantee a minimum distance error that already has been proved in Fig V.16.(c). The error is less than 0.004 m .To illustrate the robustness of the Back-stepping combined with ANFIS-FOPID controller, a square and a quadrifolium trajectory were selected.

V.3.8.1.Case of quadrifolium

In order to generate this path, the gains are chosen as: k_x=10, k_y=80 and k_θ=15. And using the following equations were applied:

$$x_R(n) = 5 * sin^2\left(2 * \pi * \frac{t}{30}\right) * cos\left(2 * \pi * \frac{t}{30}\right) \quad \text{(V.5)}$$

$$y_R(n) = 5 * cos^2\left(2 * \pi * \frac{t}{30}\right) * sin\left(2 * \pi * \frac{t}{30}\right) \quad \text{(V.6)}$$

$$\theta_R(n) = a\tan 2\left[\frac{\left(\frac{y_R(n) - y_R(n-1)}{t + \epsilon}\right)}{\frac{(x_R(n) - x_R(n-1))}{t + \epsilon}}\right] \quad \text{(V.7)}$$

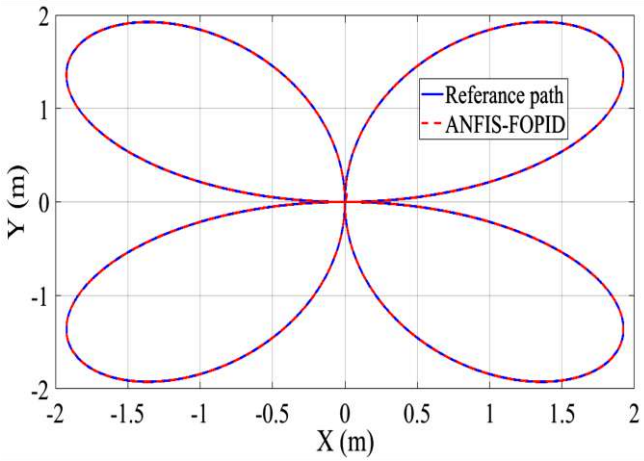


Fig V.15 Quadrifolium path.

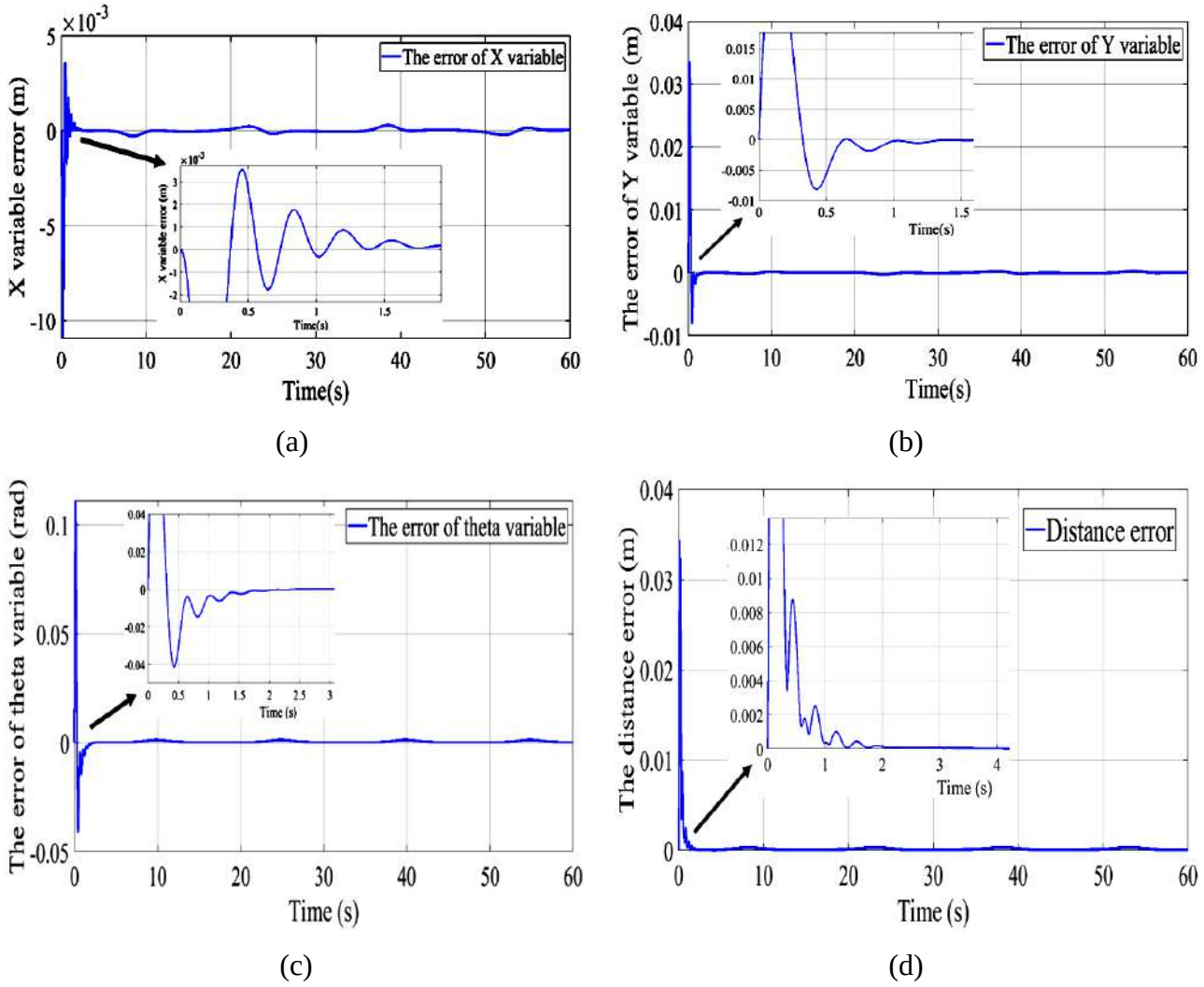


Fig V.16. Error corresponding to (a) x,(b) y; (c) the theta angle ,(d) the trajectory tracking.

The reference and actual trajectories of the quadrifolium path are shown in Fig.V.15, highlighted in blue and red lines, respectively, Fig.V.16 (a)-(d) display the errors of x, y, θ, and the tracking trajectory, respectively. The influence of the added kinematic controller in external loop is

highlighted in Fig.V.16 (d), the minimum tracking error is less than 0.002 meter. for that purpose, we can see that the two paths are identical in Fig. V.15.

V.3.8.2. Case of square trajectory

The reference and actual trajectories of the square path are shown in Fig V.17, highlighted in blue and red lines, respectively, Fig.V.18 (a)-(c) display the errors of x, y, θ, and the tracking trajectory, respectively.

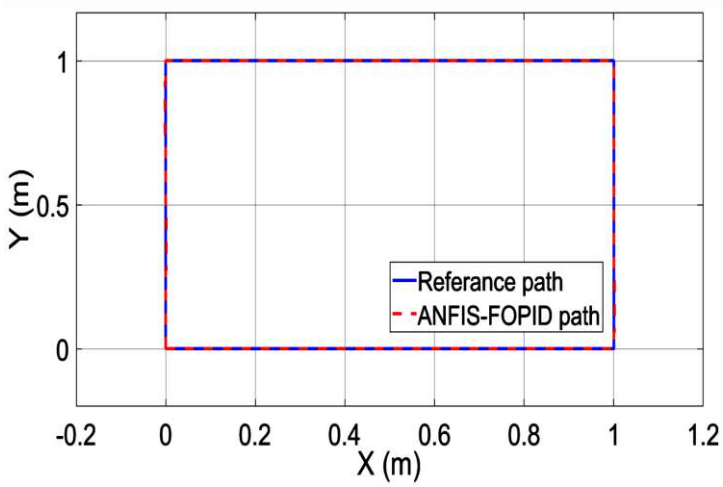
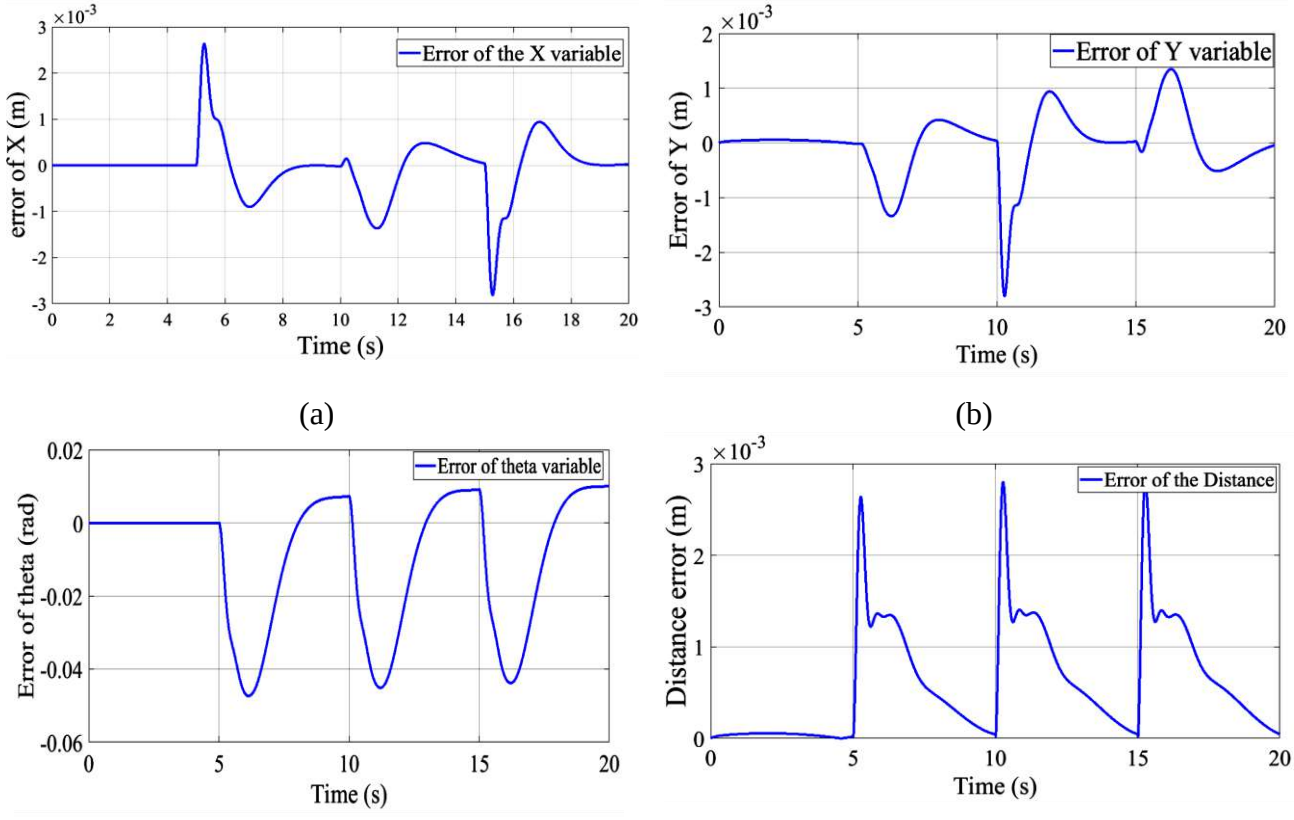


Fig V.17. Square trajectory.



(a) (b)

(c) (d)

Fig.V.18 Error corresponding to (a) x,(b) y; (c) the theta angle ,(d) the trajectory tracking.

The square trajectory is also a sharp-shaped path, here our proposed controller gives a very good response, minimum distance error obtained (less than 0.003 meter),and an error angle less than 0.05 rad, therefore we can see the superposition of the actual on reference path in Fig.V.17.

V.4. Conclusion

This chapter presented the hybridize of the fractional order PID controller with an adaptive neuro-fuzzy inference system (ANFIS), abbreviated as ANFIS-FOPID controller, for the wheeled mobile robot (WMR) motion control. Tuning parameter of the proposed controller is challenging, a powerful nature inspired algorithm is introduced in the study, namely, hybrid whale grey wolf optimizer (HWGO), a hybridization between whale optimizer (WOA) and grey wolf optimizer algorithms (GWO), A comparative study prove the over performance of the proposed controller compared to FOPID and the ANFIS, simulation in MATLAB environment gives a demonstration and judgment between these controllers in terms of lowest error value, convergence curve , minimum overshoot, received rise time, peak, peak time, and settling time obtained. Several paths, as study cases, revealed the efficiency and robustness of the proposed ANFIS-FOPID controller tuned by HWGO algorithm, which has excellent trajectory tracking, and the capacity to produce soft and suitable linear and angular velocities, excellent performance over the ANFIS and FOPID controllers. Moreover, the HWGO technique has better convergence performance than the GWO and WOA algorithms in term of parameter tuning, based on the integral square error (ISE) as a cost function. Finally, to accomplish our study, we applied a back-stepping technique for the kinematic control to guarantee a minimum tracking error, where a quadrrifolium and a square trajectory have already demonstrated the robustness of the mentioned controller in both smooth and sharp shaped trajectory.

Chapter VI:

Experimental Work

Chapter VI Experimental Work

VI.1.Introduction

The objective of this chapter is to develop a real-time trajectory tracking controller based on the hybridization of the fractional order proportional integral derivative (FOPID) controller with the Backstepping technique, using low-cost hardware. Several desired trajectories were executed experimentally, using a self designed wheeled mobile robot platform, in order to validate the applicability of the proposed control methodologies explained in our previous contributions. This work is divided into three parts, consisting of hardware design, software design, and experimental results. The hardware part includes the mechanical and electrical design of the robot platform, where the software part is focusing on developing the hardware code in the Matlab-Simulink environment, using Simulink support package for Arduino. Experimental results show that the system works well and have confirmed the simulation results of our previous proposed technique. Moreover, the prototype is easy to realize, low in cost, spares time, and reduces human effort.

VI.2.The Global Control Architecture Description

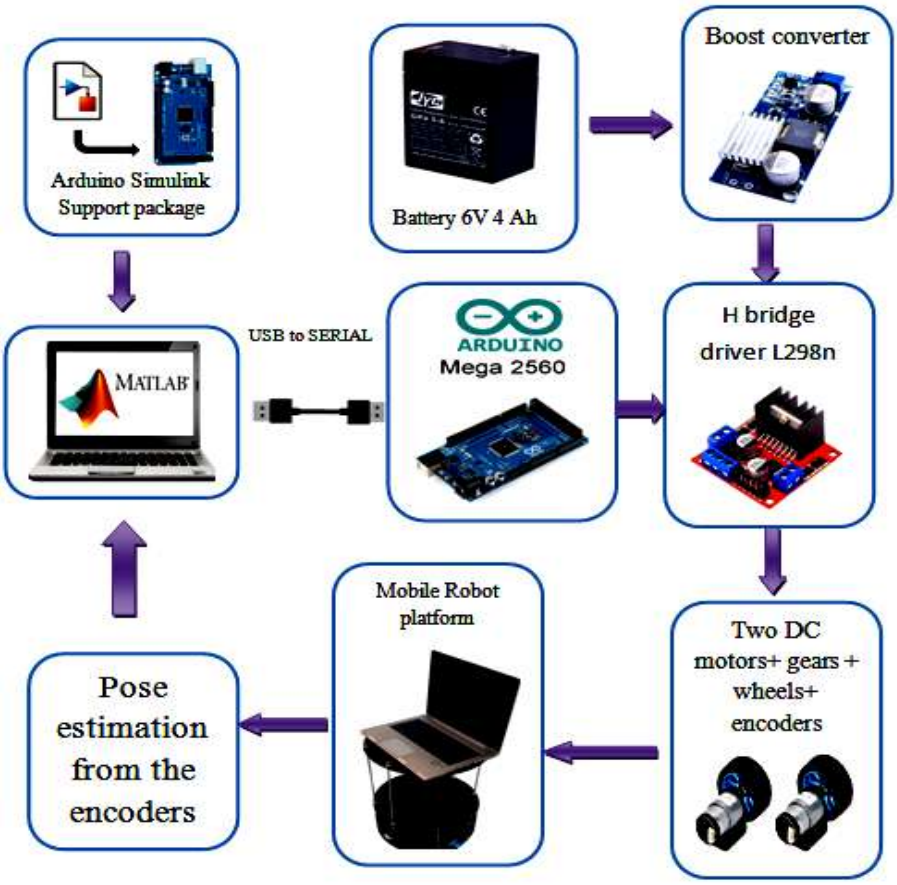
The self designed mobile robot platform is shown in Fig.VI.1. It consists of a base carrying a laptop, power supply, the electronic circuit and a white board marker, in order to act as a 2d printer.



Figure.VI.1 self designed robot plat form

Fig.VI.2 shows a simplified architecture of the complete control strategies of the system. Focusing on the block diagram, at first a Simulink support package need to be installed, to upload directly the Simulink program to the controller board (Arduino mega) without any programming (details are in

software description). Simulink is connected with Arduino mega microcontroller using serial communication with a USB cable (57000 baud rate, 8 bits). Arduino takes decision according to the signal feed by the MATLAB environment and then finally generate PWM signals for controlling the right and left wheels, where a L298n driver amplify the control action to 12v,to control the dc motors ,supplied by a 6v battery where a boost converter amplify the voltage to 12v, the quadrature encoders extract the information from the environment and send back the data to pc, the linear speed and the orientation are estimated using the forward kinematics of the mobile robot .



FigureVI.2. The global control architecture

VI.3 Hardware Design

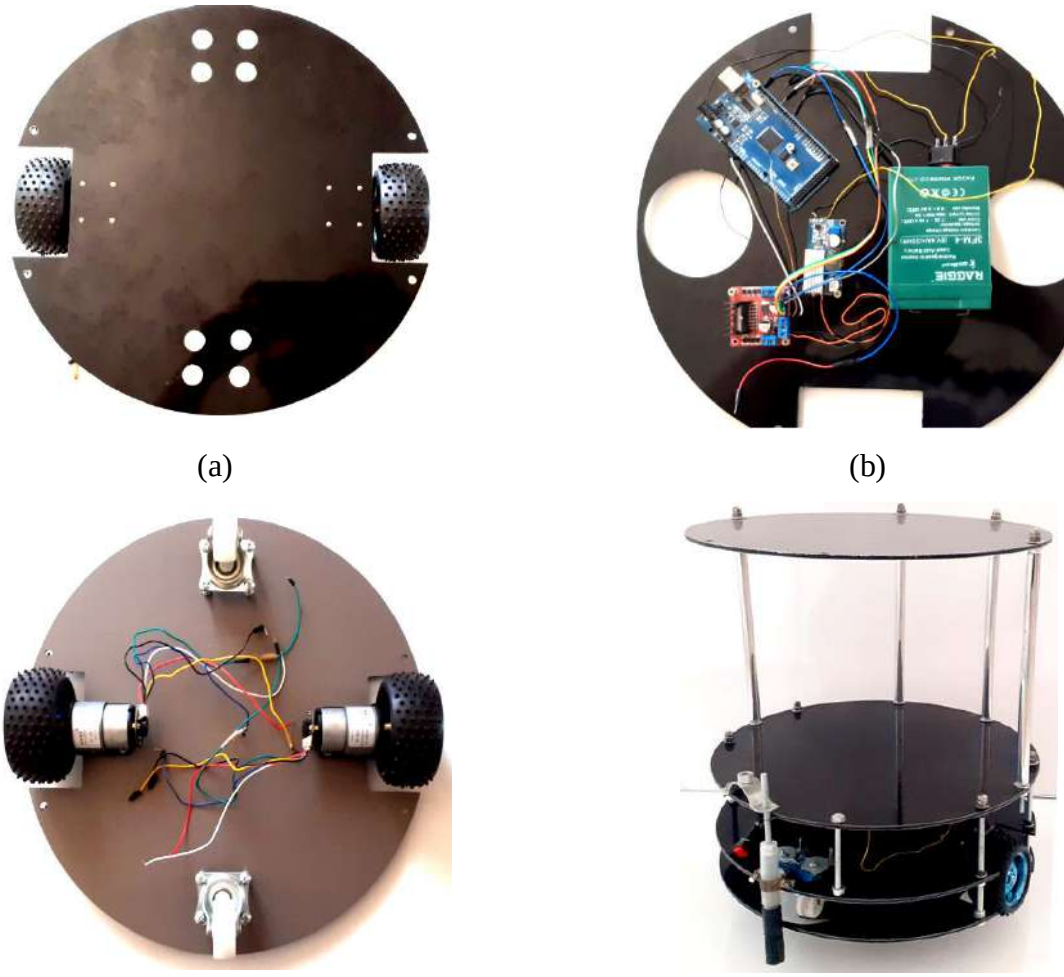
One of the cost-effective solutions for mobile robot navigation is the differential drive system. It's one of the simplest drive mechanisms for a mobile robot and is mainly indented for indoor navigation. This section is divided in two parts :

VI.3.1 The self designed robot platform

The differential drive robot consists of two wheels mounted on a common axis controlled by two separate motors. There are two supporting wheels called caster wheels. This ensures stability and weight distribution of the robot. FigVI.3 shows the self designed platform of the differential drive system an Table VI.1 shows the hardware component of the platform:

Table VI.1 hardware component of the platform

Size platform	The platform has a circular floor plan with a diameter of 34 cm
Wheels	Radius of wheel : R=0.04 m
Drive type	Differential drive with two caster wheels
Maximum speed	0.3 m/s
Motor type	DC motors 12v,107 RPM, with gear ratio 1:90 and stall tork 20kg.cm at maximum current is 3.4 A
Encoder resolution	Hall x Ratio=11x90= 990 PPR (pulse per revolution)
Controller	Arduino Mega2560 board
Power	6v,4Ah battery with boost converter
weight	4 kg
Payload	5 kg



(a) (b)

(c) (d)

Figure VI.3 the platform mechanical design : (a) Top view of the base,(b)the electronic circuit location in the base,(c) Bottom view,(d) the assembled platform.

VI.3.2.The electronics circuit design

The power supply is taken from a (6V) battery with a maximum current up to 4Ah, where a boost converter is used to convert the voltage from 6v to 12v. In order to provide energy to both dc motors and also the power electronic circuits, which is a L298n driver, this H-bridge driver reduces the voltage to 5V through its own regulator to power up the controller board which is the ARDUINO mega. The ground connection of the battery is common to the entire system electronics of the robot. The power control circuit of the motors (L298n) contains two full H-bridges of with maximum supporting current up to 2 A. Although motors may require more current (up to 3 A for a single motor). The added heat sink in the module allows it to work without heating for this application. Moreover, this circuit can be purchased at a very low cost which make it ideal for our application.

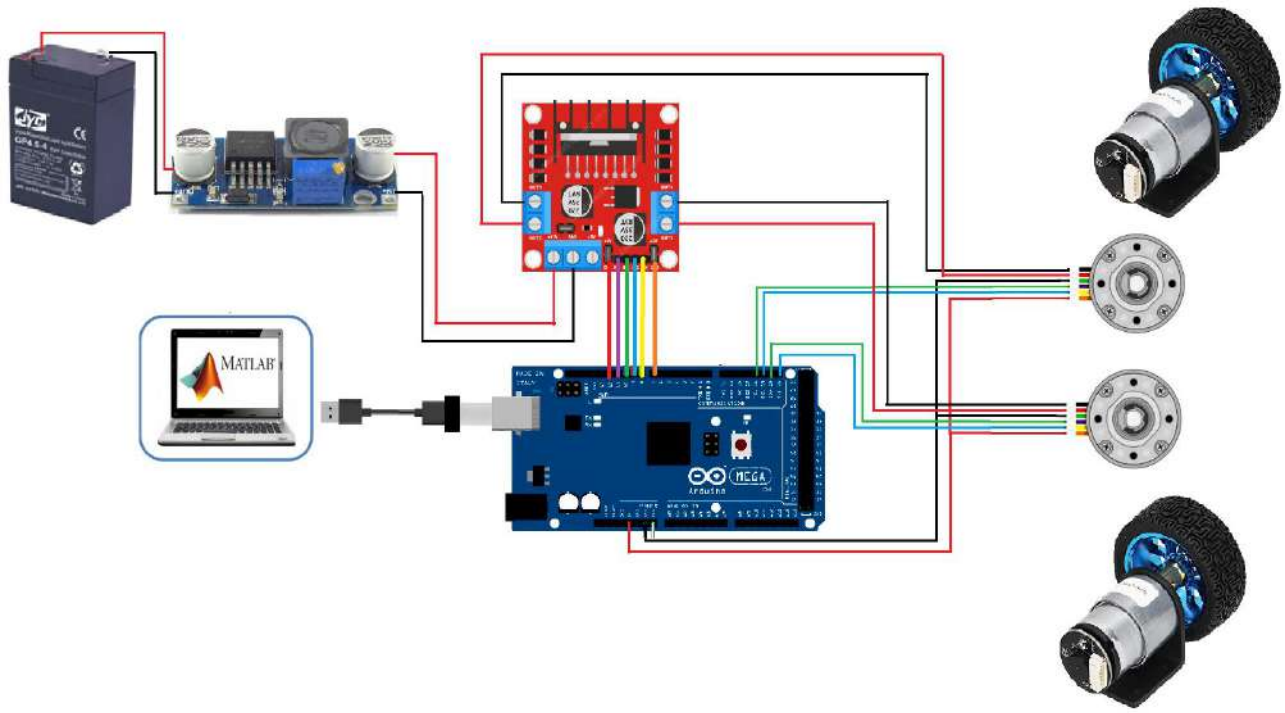


Figure VI.4 wiring diagram of the entire hardware

The 5V pin of the driver is connected to the Vin pin of the Arduino to power the Arduino board. Input 1 and Input 2 pins of the L298n driver are connected to the Arduino pin 07, pin 08, respectively, used to control the direction of Motor 1, while. Input 3 and Input 4 pins of L298n are used to control the direction of Motor 2 and are connected to pin 11, pin 10 of the Arduino, respectively. Enable A and Enable B are connected to the pin 9 and pin 12 of Arduino, respectively, which are used to control the speed of motors using PWM signals.

The encoders of the DC motors are magnetic type, with two digital outputs (A and B) of 5 V pulses that connect to inputs of Arduino interrupt. They allow you to get a resolution of 990 counts per turn. The ARDUINO mega has six external interruption lines. where pin 18, 19,20,21 are used for signals A and B of the encoders of the two actuators.

VI.4 Software Design

Simulink Support Package for Arduino Hardware was selected for combining MATLAB/Simulink environment with Arduino board. The choice was based on the great advantages that this toolbox offers such as creating and running Simulink models in numerals fields, offering a block library for configuring and accessing Arduino sensors, actuators and communication interfaces via a USB cable. This approach allows you to design control algorithms in Simulink and upload it directly to the hardware platform (Arduino board) without a need for programming [117]. These toolboxes are the responsible for enabling the data transference from the computer to the robot and vice-versa. By generating an automatic C and C++ code for embedded systems. This package can be easily found in MATLAB's home tab, on adds-on section. A window containing Arduino blocks library is illustrated in Fig VI.5. The package enables to perform tasks such as [117,118]:

• Acquire analogue and digital sensor data from Arduino board.

• Control devices with digital and PWM outputs.

• Communicate with an Arduino board over a USB cable.

• Access peripheral devices and sensors connected over inter integrated circuit (I2C) or serial peripheral interface (SPI).

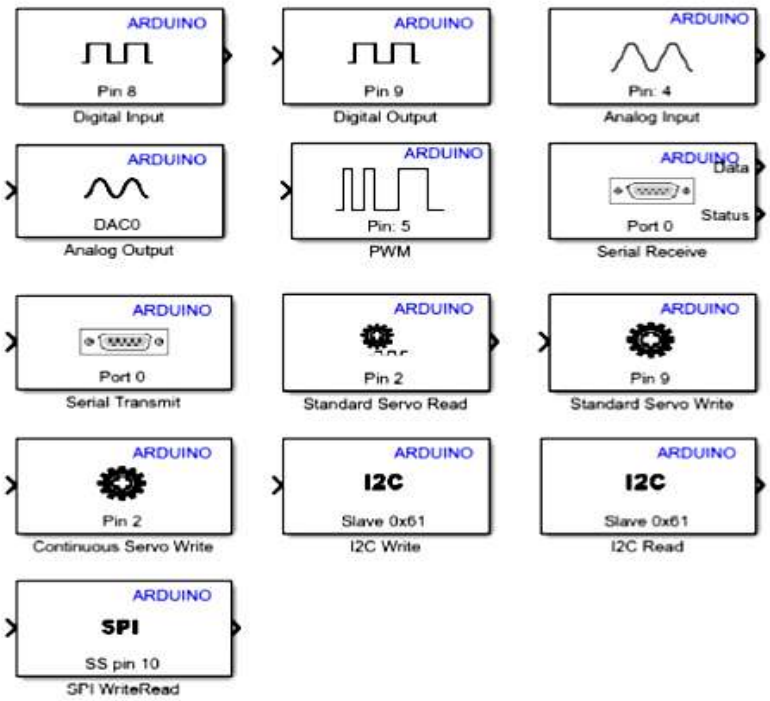


Figure VI.5 Arduino blocs library

There is two main methods for the communication between the board and the PC: Normal Mode and External Mode. Both methods have their advantage and inconvenient , depending on the project, one method may prove to be more beneficial than the other[119].

1) Normal Mode with Simulink IO:

In order to connect to an external hardware such as Arduino and download programs on to it, you can use Simulink I/O. It is a fast mode to execute your code and communicate with the I/O peripherials, but the main disadvantage of this connection mode is that data recording is more difficult than with External Mode.

2) External Mode:

By the External Mode you have the ability to tune parameters and monitor data in real-time, without having to re-download or re-compile the model each time a change is done. Once External mode is started, it uses the serial port to communicate between the Arduino and the computer. Therefore, to run in External mode, the USB should be connected between them. This ability to adjust and monitor changes in real-time is what makes External Mode such an interesting option for running these experiments. You can benefit from being able to see the immediate effects your changes have on the hardware. However, this ability to make changes in real-time has a cost, as in order to communicate back and forth from the serial port, the port is kept opened and has a limited amount of bandwidth. This limitation affects the rate at which the program can run. In other words, this mode is slower than the Normal mode [119].

It is worth to comment that there is another option, called Deploy to Hardware with which we can export all the code to the micro-controller and so there is no need to be connected to a PC. But, in this case, the external mode will be used as we are going to use the Scope blocks of Simulink as oscilloscopes of the most important signals, such as the control signal or the output of the plant.

VI.4.1 The complete Simulink model

Fig VI.6 shows the complete bloc diagram of the designed simulink model which is responsible for controlling the robot platform, it consist of a trajectory generator programmed based on a Matlab function responsible for generating the reference signal, a backstepping controller is located in external loop programmed using a matlab function too, responsible for the kinematic control actions. Two discrete FOPID controller blocs In the internal loop were created by benefiting from FOMCON toolbox (description will be in next sections), one controller was designed for the speed control and the second was for the orientation control, the output signals were send to the two wheels using PWM technique, the quadrature encoder used to determine the robot localization.

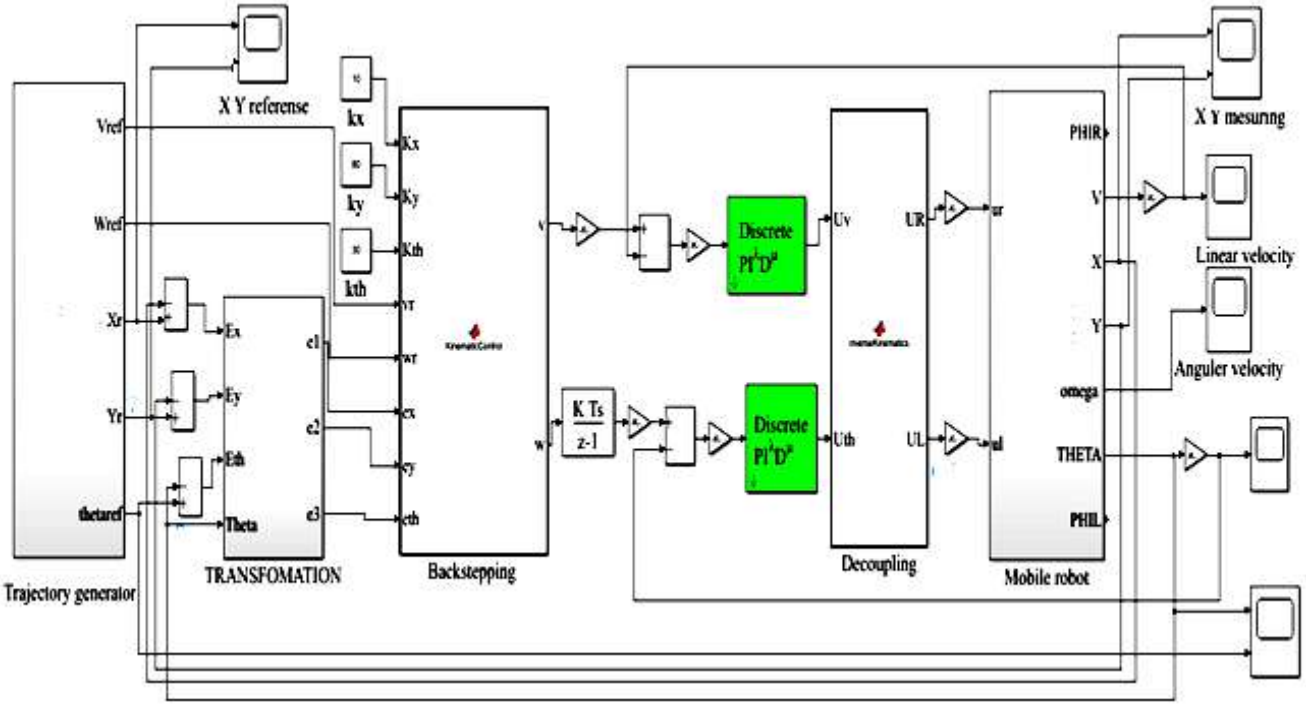


Figure VI.6 the complete designed simulink bloc diagram

VI.4.2 FOPID controller implementation

For the FOPID implementation the FOMCON ("Fractional-order Modeling and Control") toolbox was chosen [120]. The use of the toolbox is facilitated with a set of graphical user interfaces aimed at making. Further motivation for developing FOMCON is the desire to obtain a complete solution for efficient controller design and analog/digital implementation. In particular, FOMCON offers a set of tools for working with fractional-order PID controllers.

VI.4.3 Speed Control

The DC motor Speed is controlled by changing the PWM ratio. PWM is a method of modulating a signal by varying the duty cycle of pulses at constant carrier signal frequency (period) and pulse amplitude. The computer uses digital output pin with the PWM mode as controlling peripherals.
However, the embedded computer cannot connect to DC motor directly, because the limited current and output voltage also is not varying. Therefore, the embedded computer uses the L298N motor driver as an electric power device that drives motor. The motor driver receives on/off signal from the Arduino mega, and it drives the two motors.
A DC motor driver can turn on/off a DC motor, but a DC motor driver does not have a function of a speed control way. Therefore, the embedded computer quickly turns on/off these switches with Pulse Width Modulation (PWM) and controls motor speed and the direction too,

Fig VI.7 shows the designed simulink model that generates the PWM signals. The frequency of the waveform is constant at approximately 490 Hz.

The Output of the first FOPID controllers (see figure VI.6) is responsible for controlling the right wheel velocity by sending the control signal to Arduino digital pin 09 which is the PWM pulse width modulation pin. The duty cycle is controlled from 0 to 255. If logic 0 then duty cycle is zero means the speed of machine is 0 RPM. While logic 255 then speed of motor is 107 RPM that means maximum mobile robot speed, here a Matlab function has been created (see fig VI.7) to deal with the PWM output signal and controlling the direction of the right wheel at the same time ,using PIN 09 , PIN 07 and PIN 08, the Matlab function here is to make sure that the L298n driver will not short circuit. The same technique has been applied for controlling the left wheel using the second FOPID controller, using pin 12 for PWM signal were, pin 10 and pin11 for the direction of left wheel.

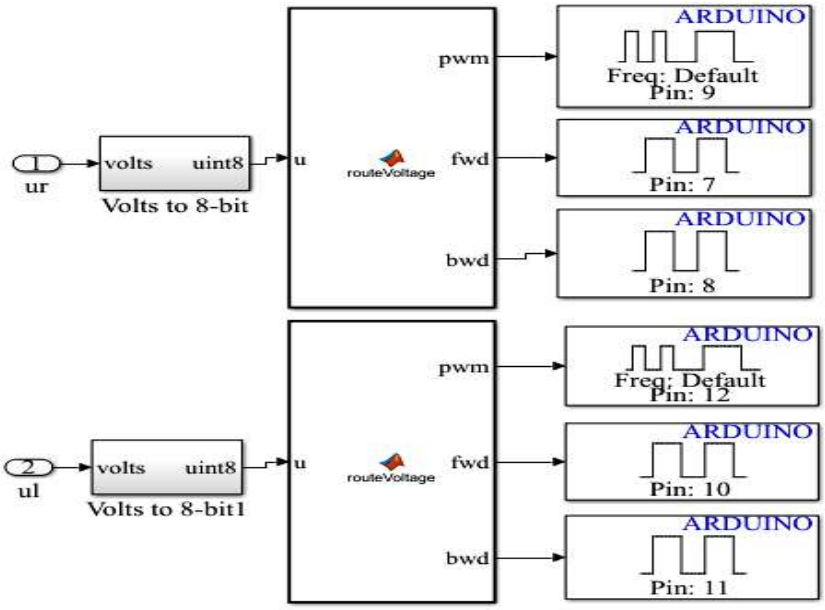


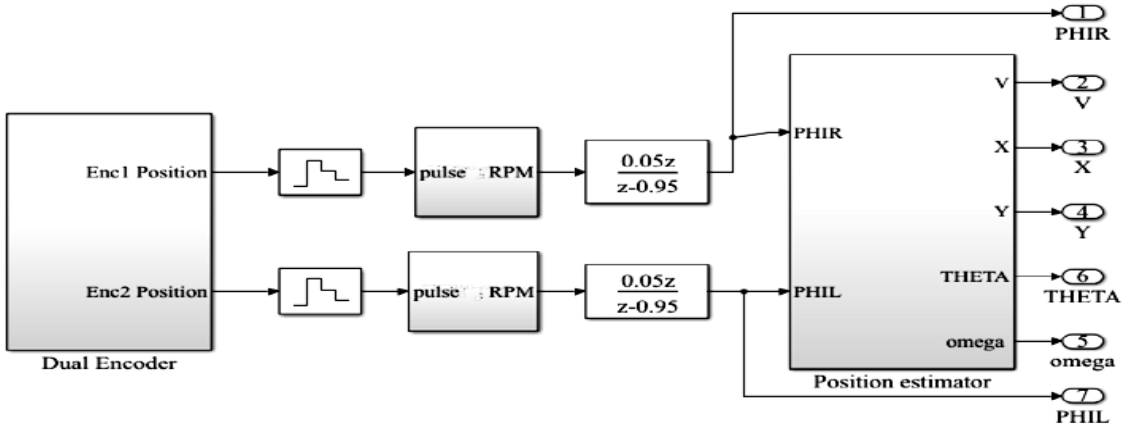
Figure VI.7 the designed simulink model for the PWM signals and orientation.

VI.4.4 Motor speed calculation from encoder

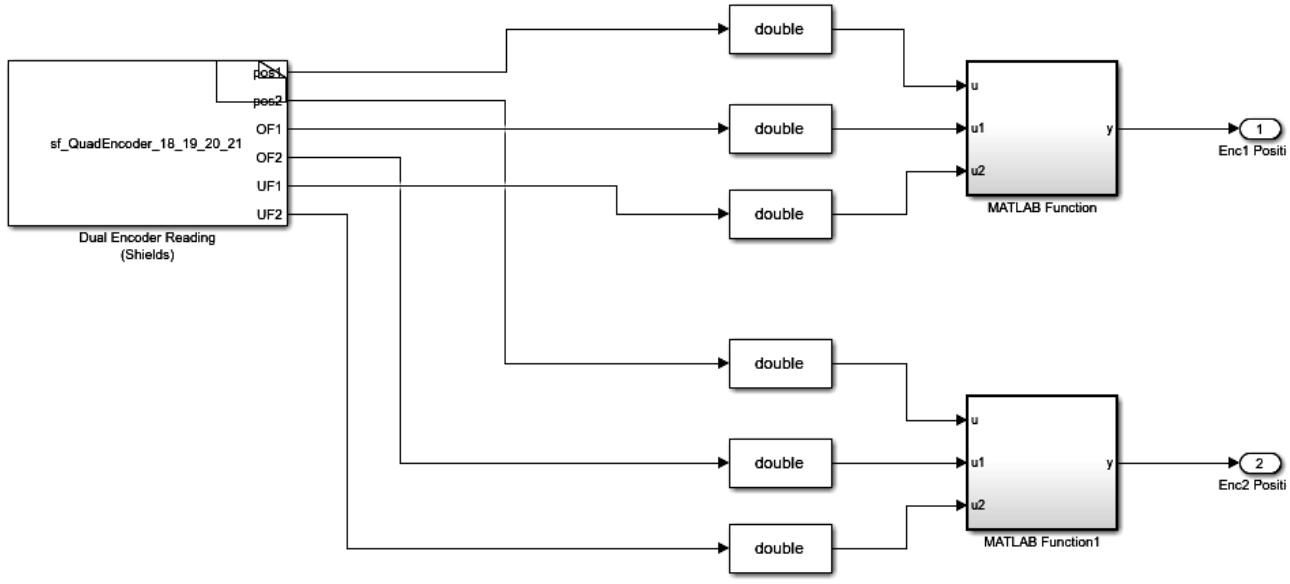
The quadrature encoder allows to control the movement of the vehicle wheels, this sensor produce a signal data in form of a square wave. The number of pulses is related to the motor speed . Fig VI.8, shows the simulink model of the pulse count and the conversion to RPM with the help of a S function using pin 18,19,20,21 for both DC motors, and using a “Rollover block correction”, this bloc was created because the buffer that is keeping track of the number of counts can only represent numbers between -32768 and 32767 (i.e. it uses 16-bits, 15 bits for the number and 1 bit for the sign). This subsystem compares the current number of counts to the number of counts from the

previous sample in order to determine whether a rollover has occurred (at 32767 or at -32768) and then modifies the accumulated number of counts to remove the rollover (see Fig VI.8.b).

The sampling time is: 0.005 seconds. In this study, the encoder of geared DC motor has a reduction ratio of 1:90 (metal gearbox) providing a resolution of 11 pulse per revolution. So, the counts per revolution (PPR) of the motor speed and gear ratio can then be calculated: PPR=11x90=990. The Gain block used for converting pulse to RPM is shown in Fig VI.8.a. Since the output signal has quite a lot of noise. A transfer function filter block was then applied here to reduce some noise to a certain extent.



(a)



(b)

Figure VI.8 the designed simulink model for encoder reading: (a) the structure of the RPM reading ,(b) the S function bloc and the rollover bloc .

VI.4.5 Pose estimation

The Odometric Localization process, also known as Dead Reckoning, is the estimation of a robot's coordinate based on the data taken from the quadrature encoder of the right and left wheels. The forward kinematics model is then applied to estimate the robot location, linear speed and orientation angle. The data is then integrated over time starting from a known initial location, (x(0) and y(0)), and heading to obtain an estimate of the robot pose. Fig.VI.9 shows the simulink model corresponding to the estimation protocol.

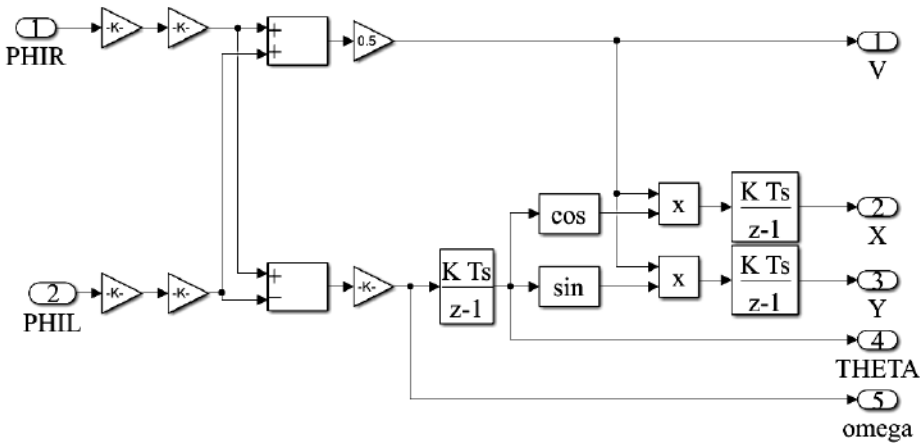


Figure VI.9 the designed simulink model for pose estimation.

This approach is the most basic methodology used in mobile robotics, but is still routinely applied in industrial robotics applications that do not require high-fidelity location estimation, or as a redundant backup system for validation and error detection. It is required to adjust the alignment of the wheels and diameters of wheels are equal. In addition, the floor contact of a workspace should be smooth and flat to avoid slippage as a result of variation in the contact point of the wheels.

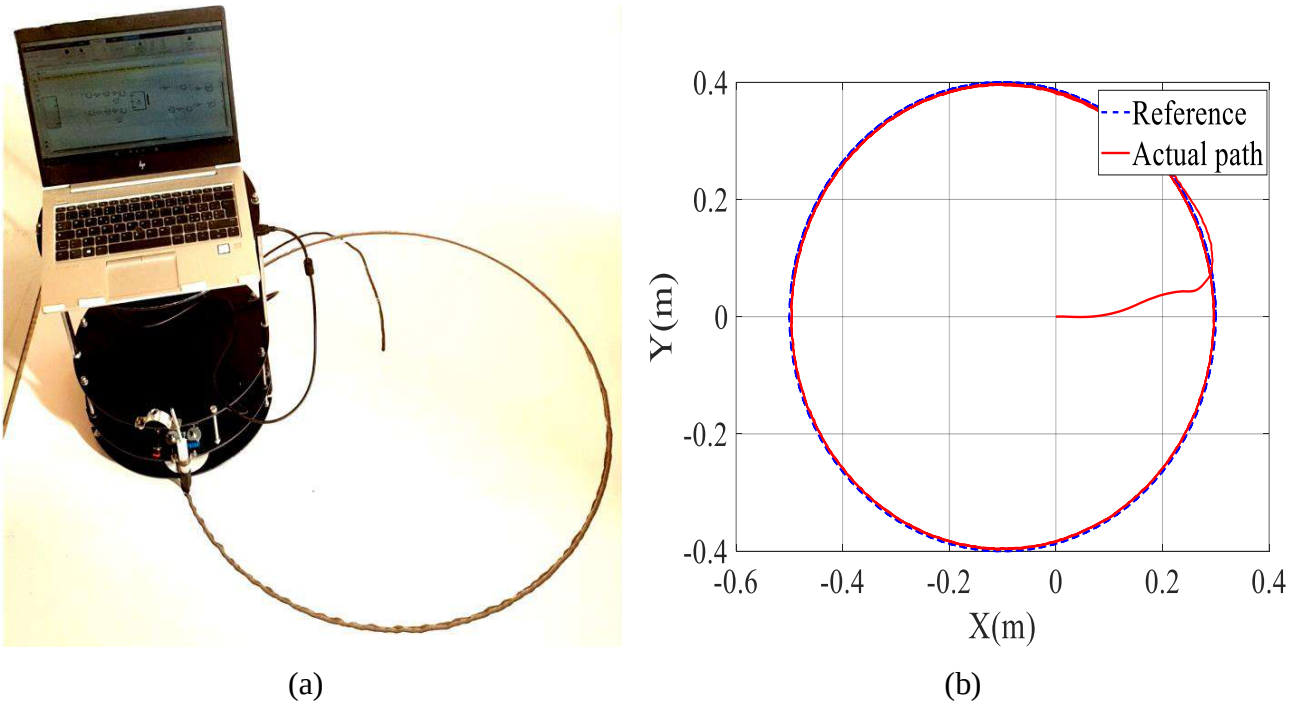
VI.5 Experiments Results

In order to prove the robustness of the proposed control approach, experiments were made using the designed mobile robot platform described in previous section, to track four different desired trajectories.

The implemented technique shows the behavior of the system based on the combination of the FOPID with the backstepping controller in terms of vehicle orientation, velocity control, and tracking errors. In additional, this investigates the capability of tracking any trajectories with continuous and non-continuous gradients. These comparisons are made under identical conditions in order to specify the differences and similarities of using different trajectories.

VI.5.1 Tracking Circular Trajectory Case Study 1

At first, the applicability of the suggested control technique to track the continuous gradient with a constant rotation radius was verified using a desired circular trajectory. Fig VI.10a illustrate the draw of circle trajectory on a white board using the designed platform, while Fig VI.10b is the estimated values in the XY coordinate.



(a) (b)

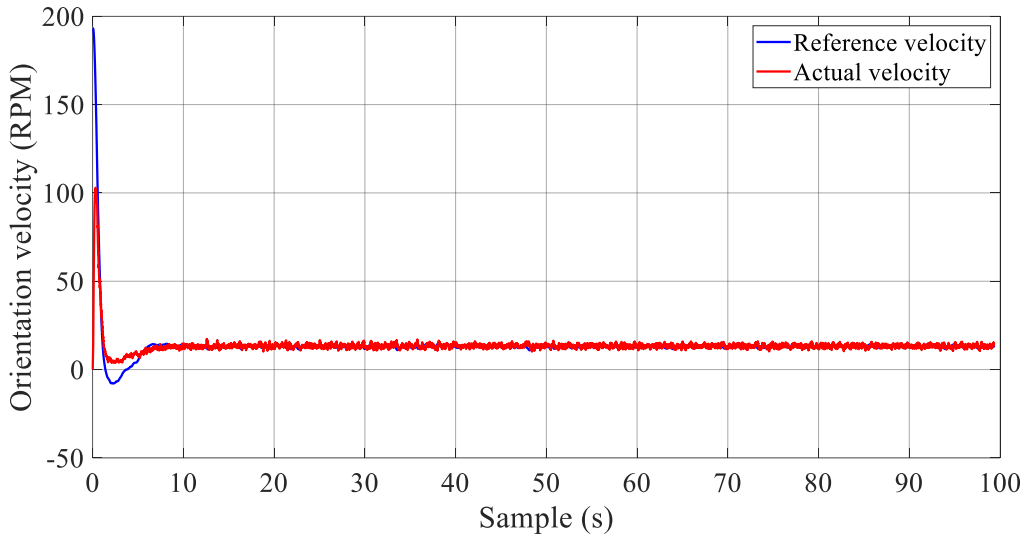
Figure VI.10 Circular path; (a) the actual circle trajectory (b) estimated values from encoders

This trajectory is performed a continuous gradient trajectory motion. Hence, the error between the actual and desired linear velocity is supplied into the first fractional order PID controller.FgVI.11a shows The output of the control action for speed control that is it tracks and converges to the desired reference velocity were. FgVI.11b demonstrates the reference and actual orientation output of the second FOPID controller. It can be observed that the orientation errors are maintained around zero.

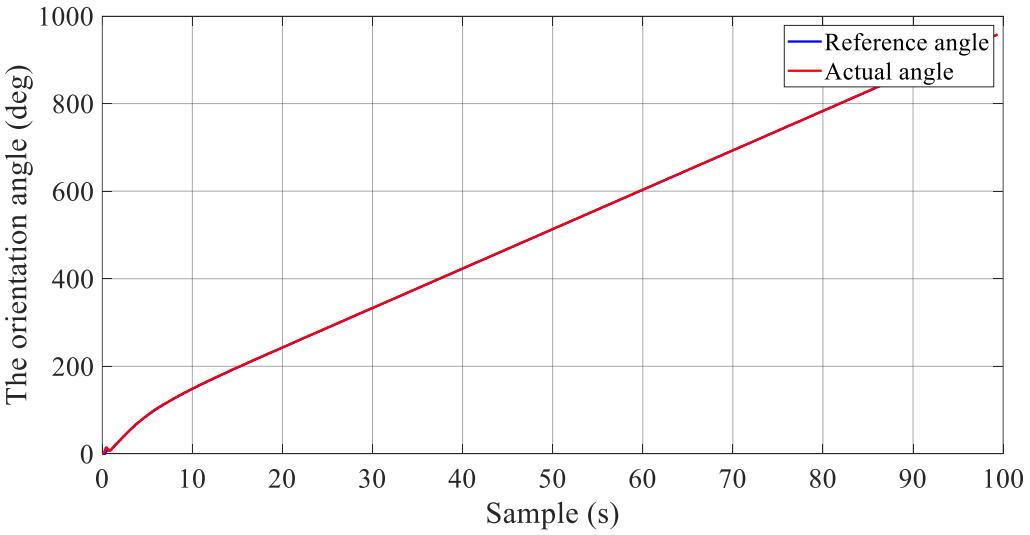
The error of the X coordinate is represented in Fig VI.11c, were FgVI.12: a, b and c shows the errors according to the Y, theta and the distance respectively.

From the previous response estimated from the encoders we can say that the control methodologies have been successfully realized for the circular trajectory.

(a)



(b)



(c)

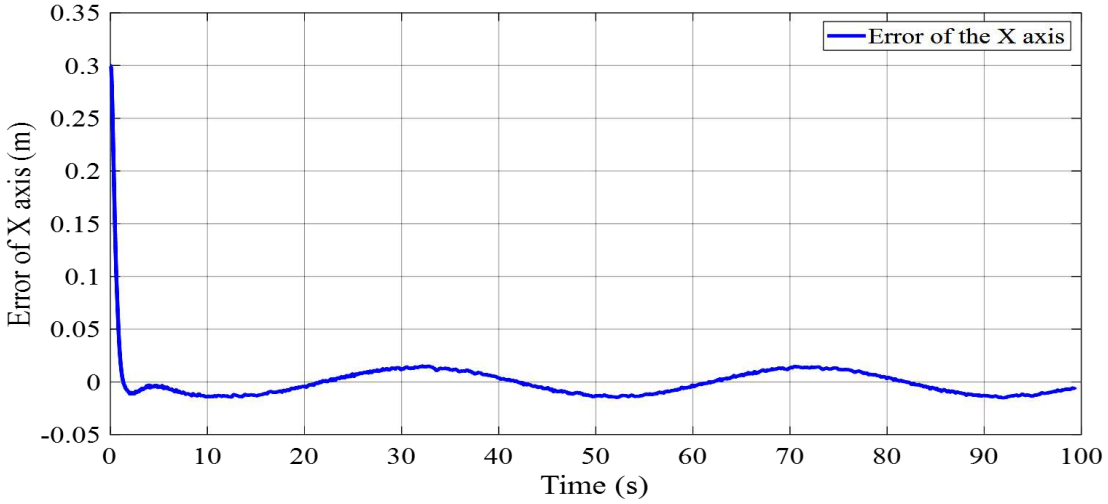
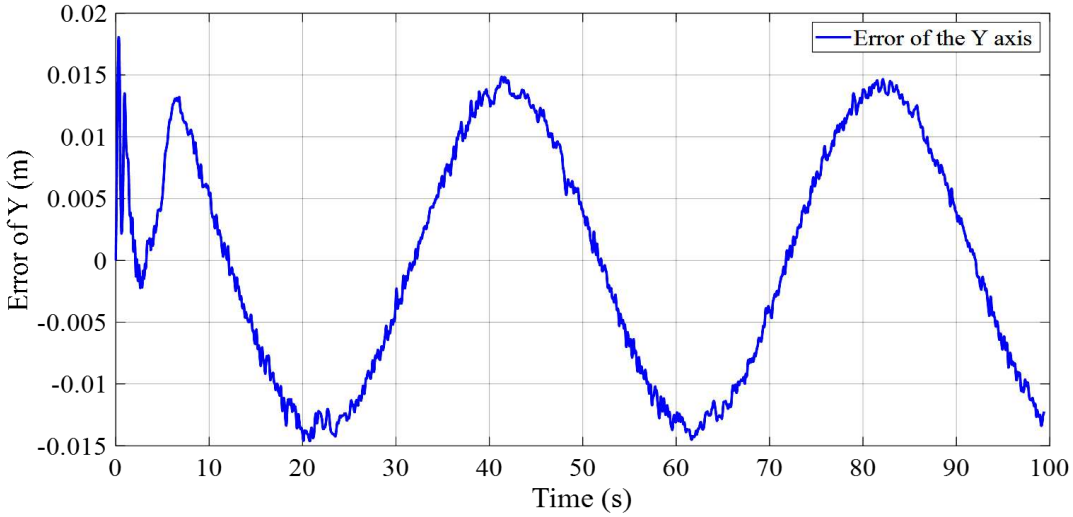
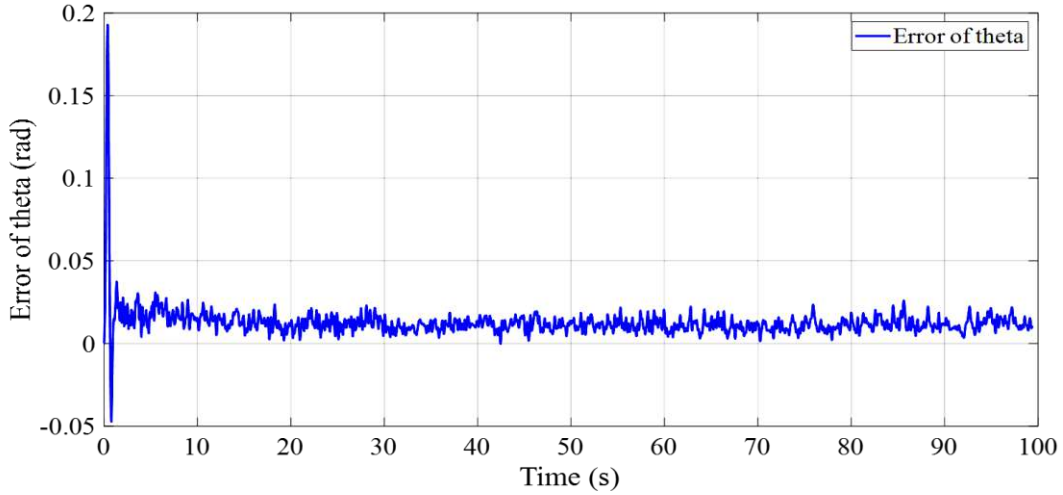


Figure VI.11:circuler path control efforts: (a) the tracked speed (b) the tracked orientation (c) error of X axis

(a)



(b)



(c)

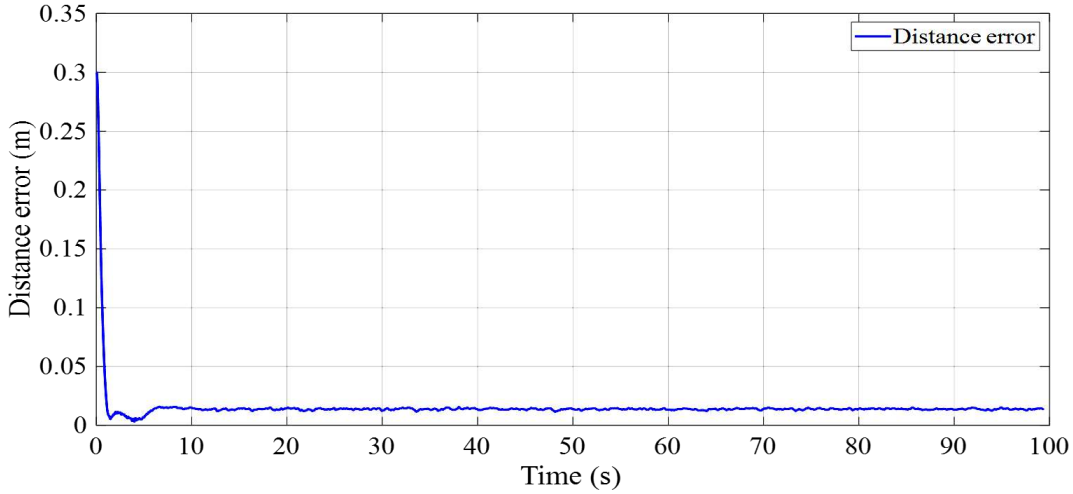
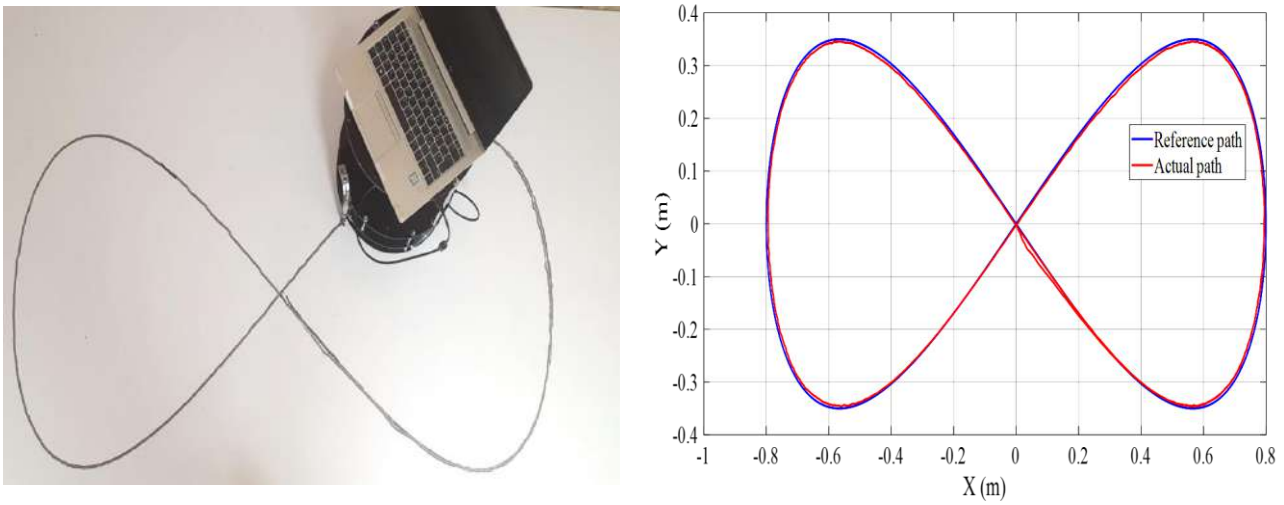


Figure VI.12:the tracking errors of the circular path: (a) error of Y axis (b) error of the orientation (c) error of the distance

VI.5.2 Tracking Lemniscates Trajectory Case Study 2

The lemniscate trajectory is another type of continuous gradient route. Although lemniscate trajectory's orientation is nearly similar to circular trajectory, it is a plane curve with a characteristic shape that it consists of two loops that meet at a central point. It demonstrates a roundabout motion because of changing the directions of coordinates whilst moving. The lemniscate trajectory is generated using the equations (III.72, III.73 and III.74). The 2D print of the infinity trajectory of actual robot platform is presented in Fig VI.13a, were the estimated data taken from quadrature encoders of desired and actual lemniscate trajectories are depicted in Fig VI.13b. It demonstrates a very good response and reasonable performance of trajectory tracking based on the use of the combination between the FOPID controllers and the back-stepping controller.



(a) (b)

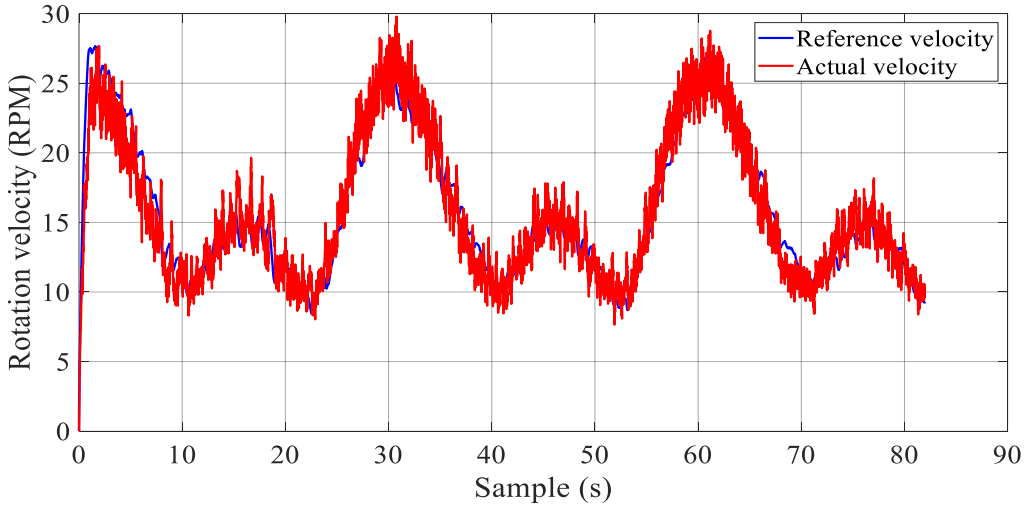
Figure VI.13: (a) The 2D print of the infinity trajectory, (b): estimated XY plane for infinity path

The control actions estimated from the encoders which shows the efforts needed from the robot vehicle to track the speed is given in Fig VI.14a, here The data action are converted from linear velocity (m/sec) to a rotation velocity (revolution per minute : RPM) , the output efforts for the orientation angle control are given in VI.14b. The figures show a reasonable response of system actuators for such a trajectory.

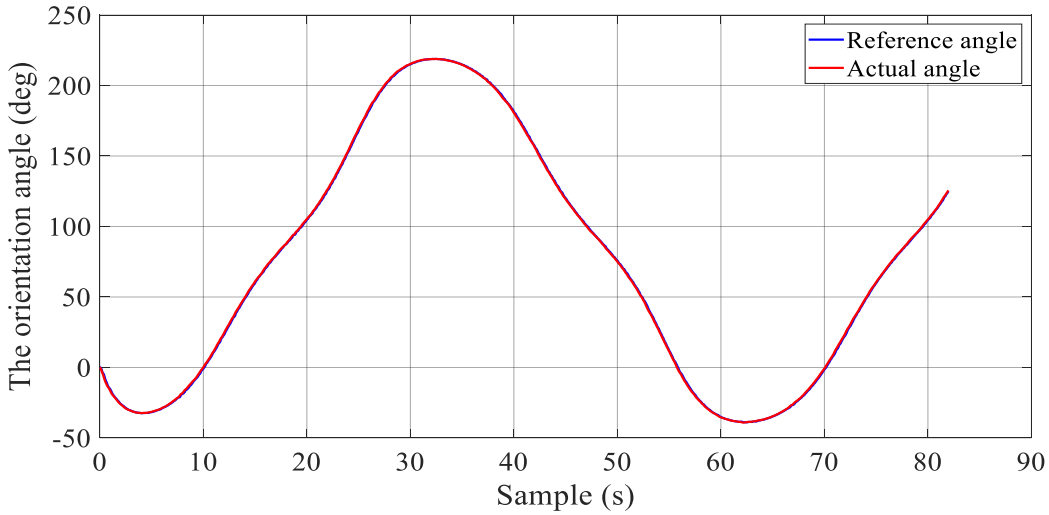
The error of the X coordinate is represented in Fig VI.14c, were FgVI.15: a, b and c shows the errors according to the Y coordinate, the theta orientation and the distance error respectively.

From the previous response estimated from the encoders and the real 2d print of the trajectory, we can say that the control methodologies have been successfully realized.

(a)



(b)



(c)

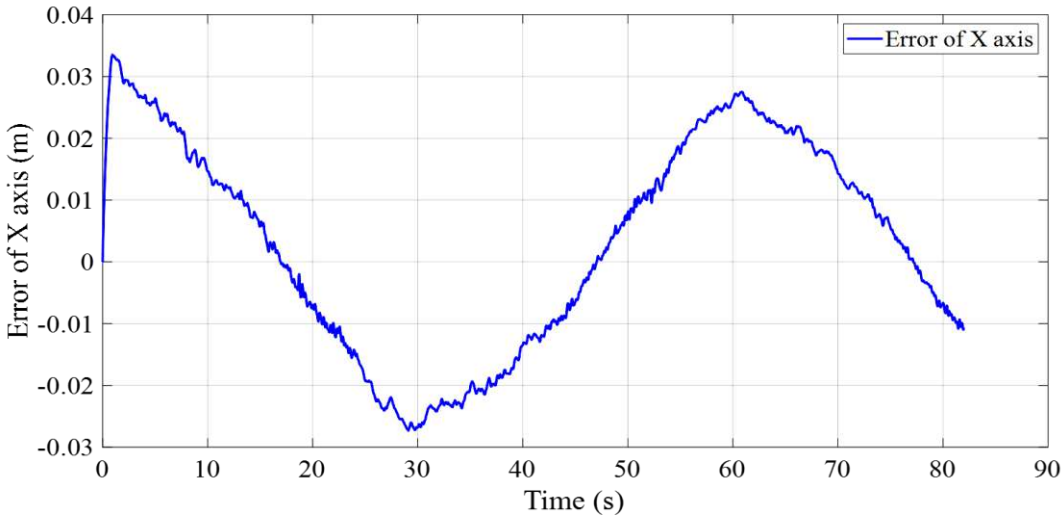
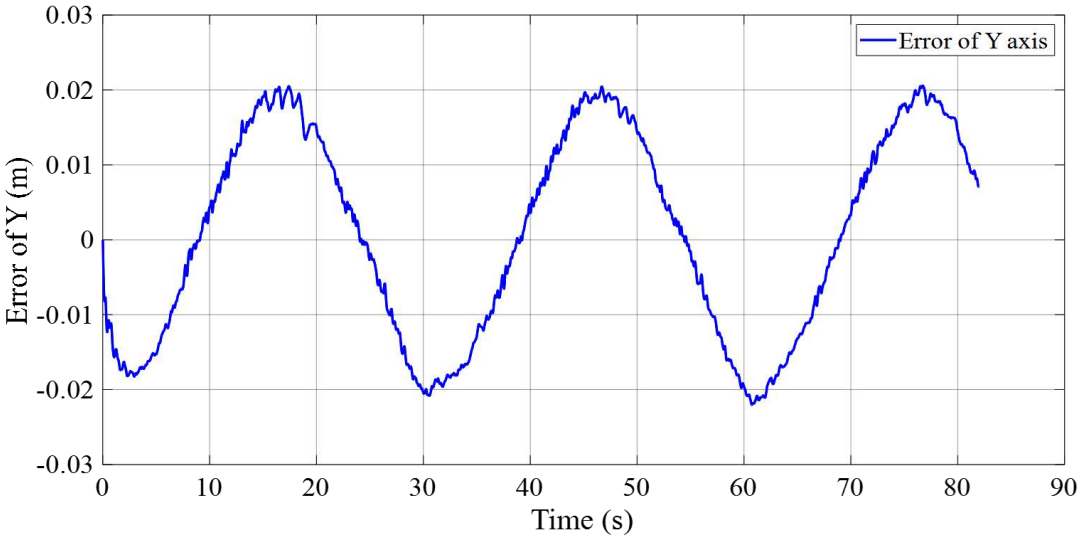
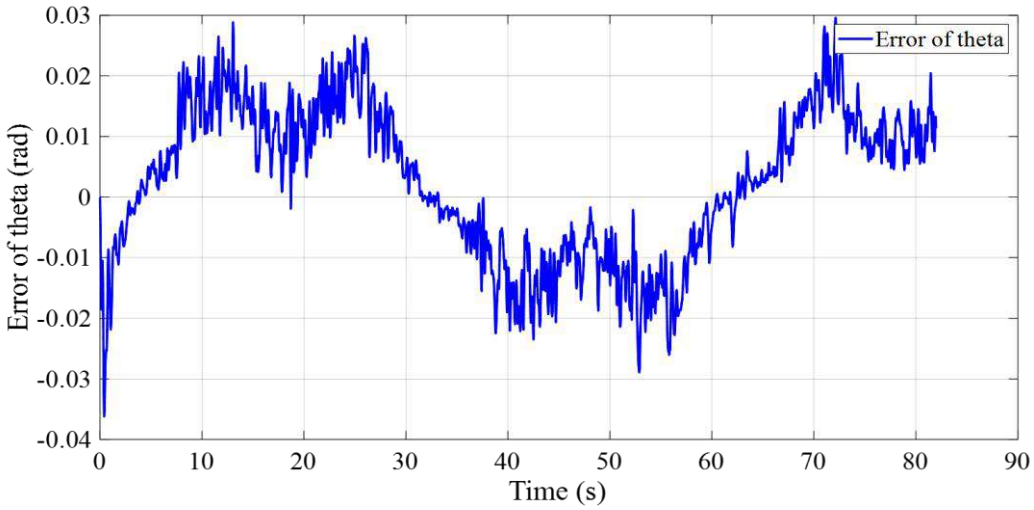


Figure VI.14 lemniscate control efforts: (a) the tracked speed (b) the tracked orientation (b) error of X axis

(a)



(b)



(c)

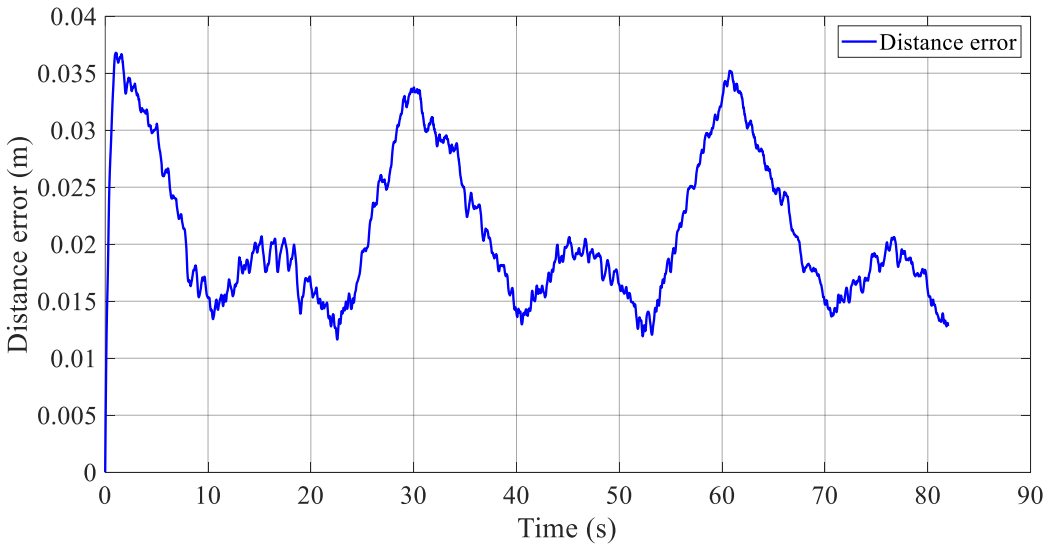
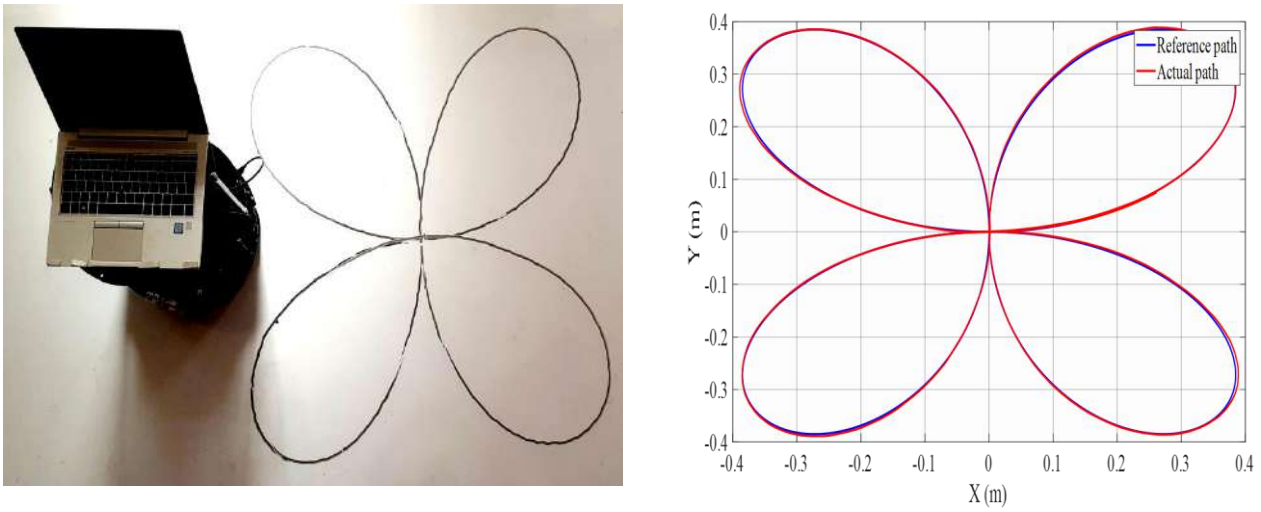


Figure VI.15. tracking errors of lemniscates :(a) error of Y axis (b) error of the orientation (c) error of the distance

VI.5.3 Tracking quadrifolium Trajectory Case Study 3

The quadrifolium trajectory is classified as continuous gradient route. Although a quadrifolium trajectory's orientation is nearly similar to the lemniscates trajectory, it is a plane curve with a characteristic shape that it consists of four loops that meet at a central point. it's much complicate to realize it than the lemniscates,



(a) (b)

Figure VI.16(a) The 2D print of the quadrifolium trajectory, (b): estimated XY plane for quadrifolium path

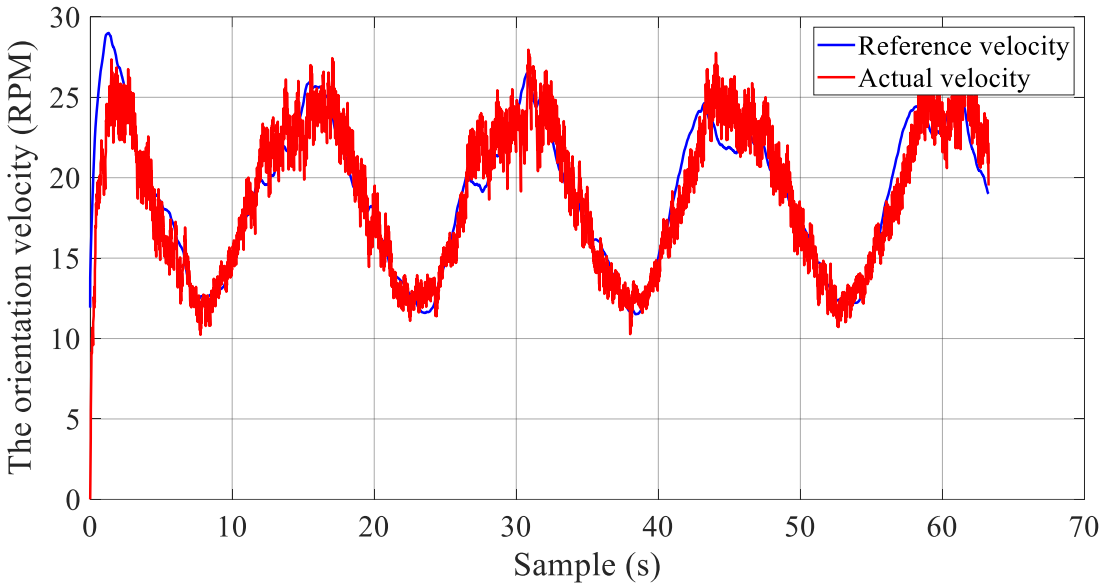
The 2D print of the quadrifolium trajectory using the actual robot platform is presented in Fig VI.16a, were the estimated data taken from quadrature encoders of desired and actual trajectories of quadrifolium are depicted in Fig VI.16b.

The control actions estimated from the encoders which shows the efforts needed from the robot vehicle to track the speed is given in Fig VI.17a, here The data action are converted from linear velocity (m/sec) to a rotation velocity (revolution per minute : RPM) , the output efforts for the orientation angle control are given in VI.17b. The figures show a reasonable response of system actuators for such a trajectory.

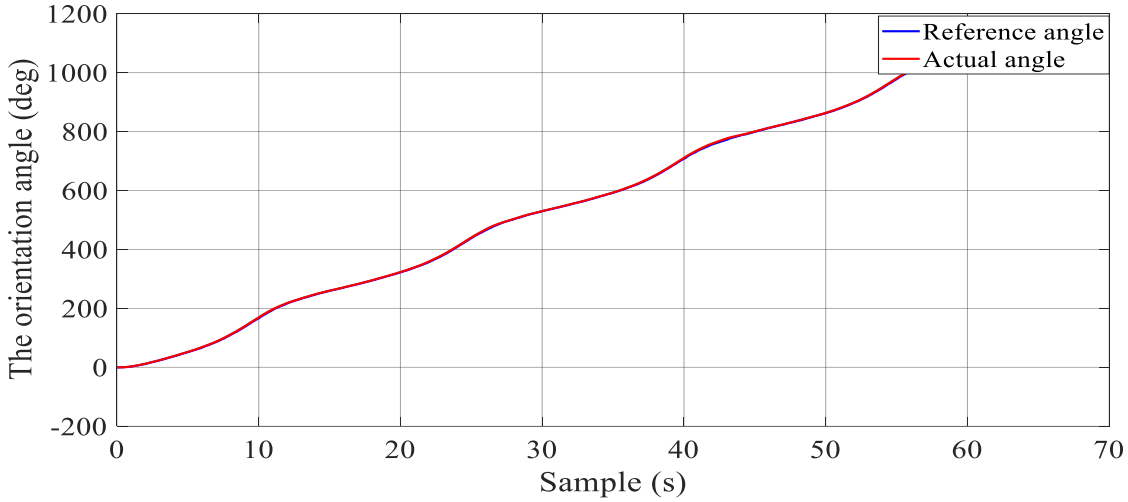
The error of the X coordinate is represented in Fig VI.17c, were FgVI.18: a, b and c shows the errors according to the Y coordinate, the theta orientation and the distance error respectively.

From the previous response estimated from the encoders and the real 2d print of the trajectory, we can say that the control methodologies have been successfully realized.

(a)



(b)



(c)

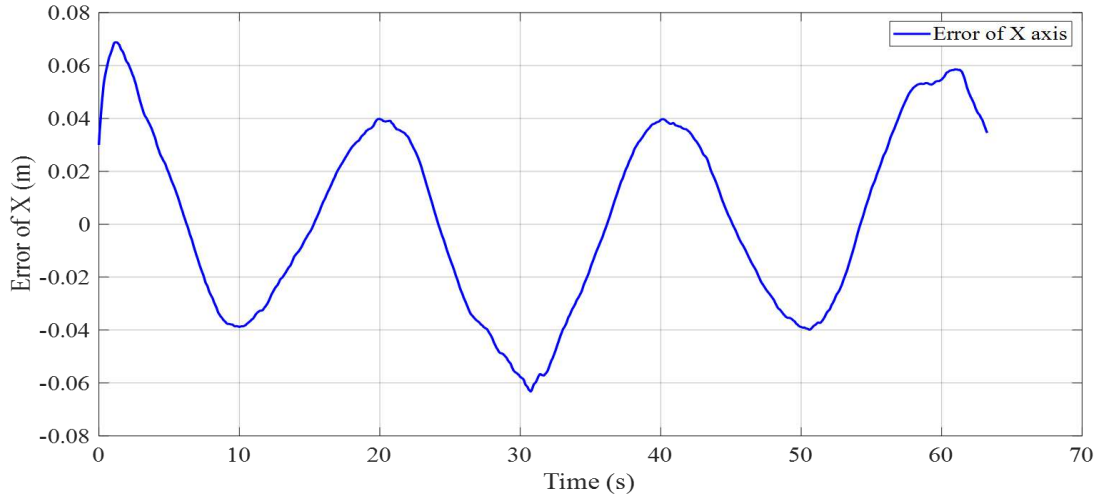
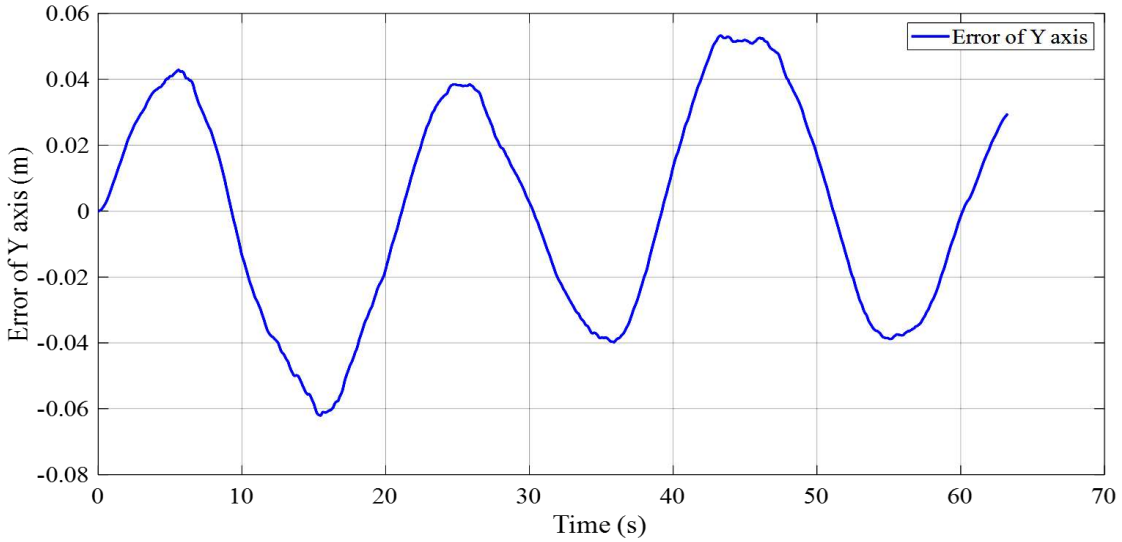
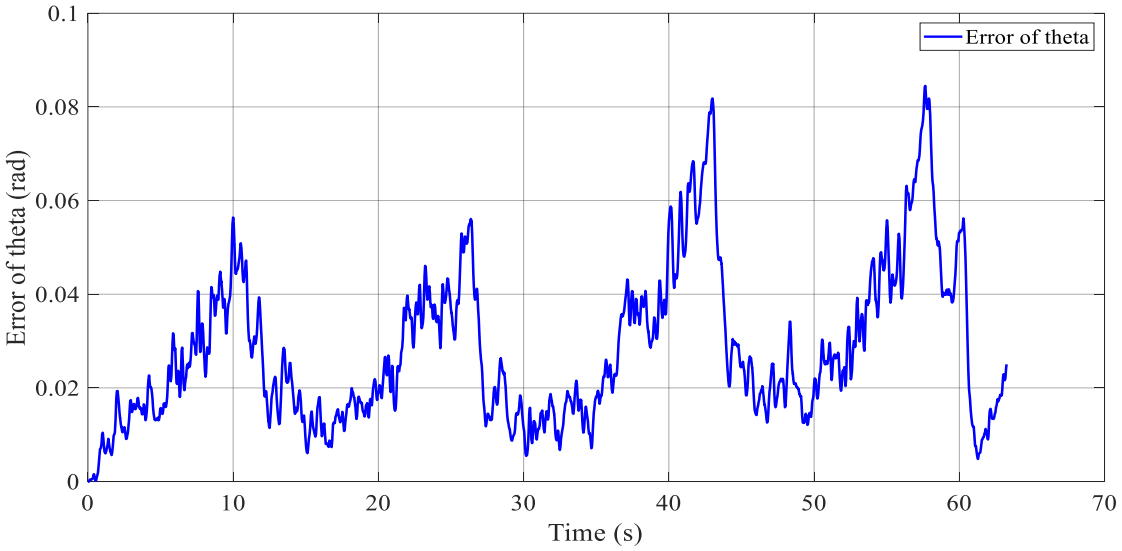


Figure VI.17. quadrifolium path control efforts: (a) the tracked speed (b) the tracked orientation (c) error of X axis

(a)



(b)



(c)

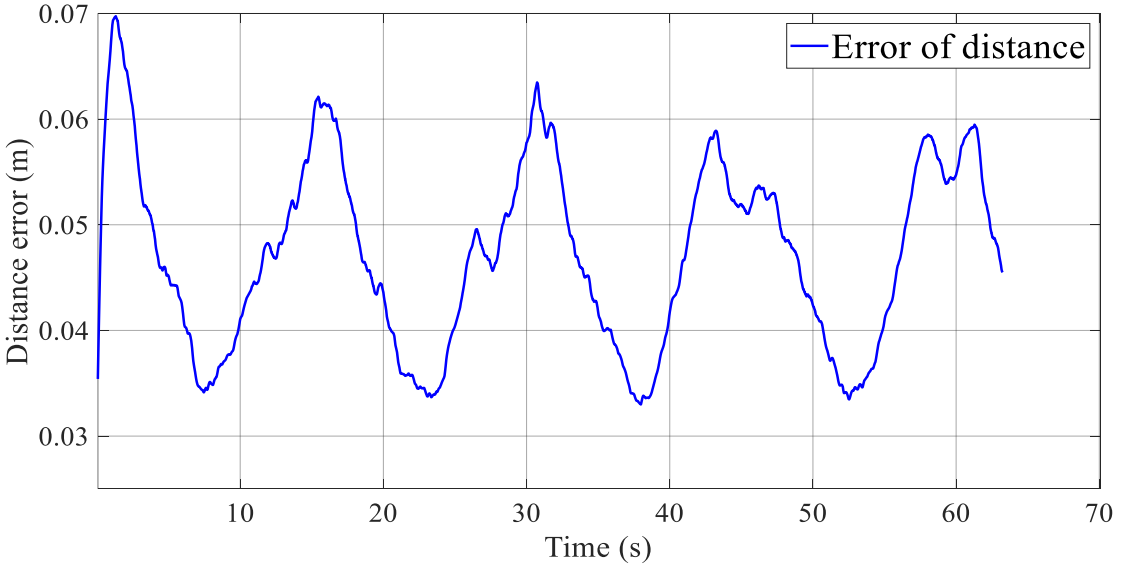
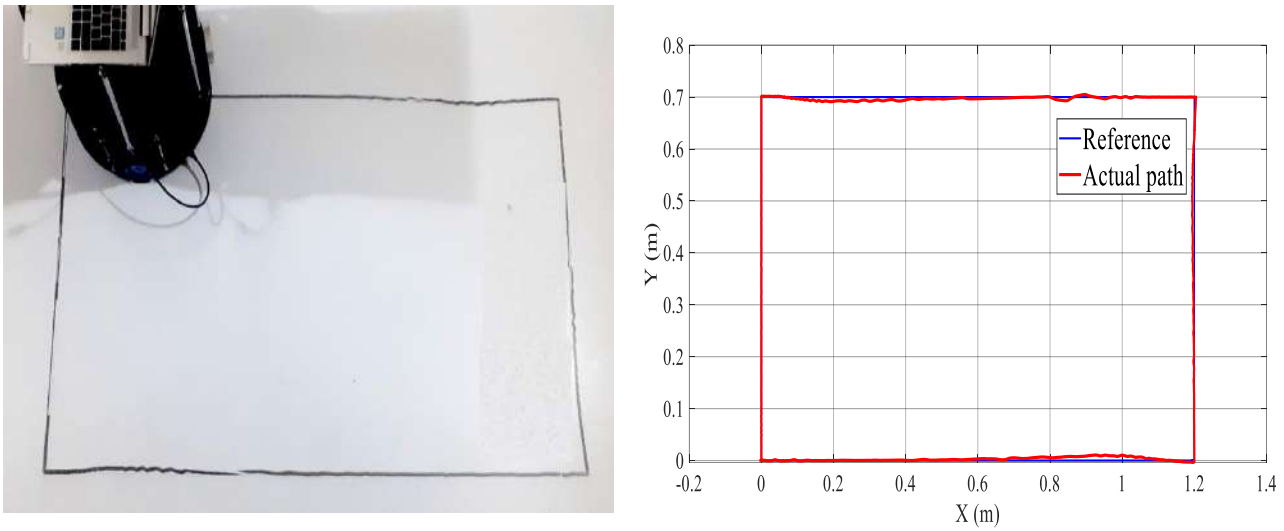


Figure VI.18 tracking errors for quadrifolium path :(a) error of Y axis (b) error of the orientation (c) error of the distance

VI.5.4 tracking rectangular Trajectory Case Study 4

In this study, a non-continuous gradient trajectory is generated, which means a instantaneous changing in the orientation at the connecting point while executing the path of the WMR. The rectangular trajectory is classified as a sharp, shaped trajectory, which contain many points of inflection. This example has been carried out in order to see the capability and the performance of the combined backstepping and FOPID controller, to deal with such as cases.



(a) (b)

Figure VI.19(a) The 2D print of the rectangular path, (b): estimated XY plane for rectangular path

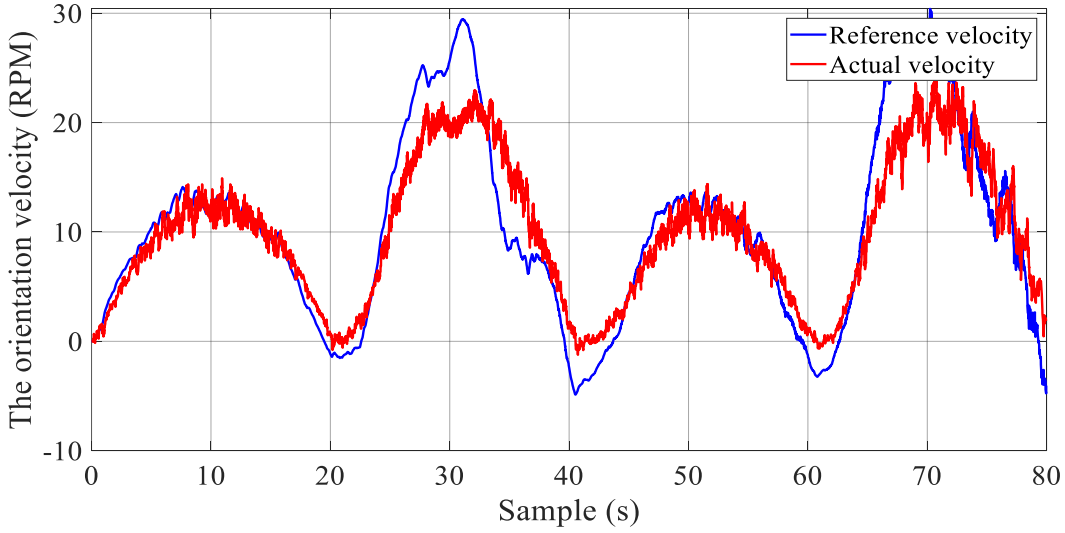
The 2D print of the square trajectory is presented in Fig VI.19a, were the estimated XY plane coordinate that are taken from dc encoders of desired and actual trajectories are depicted in Fig VI.16b.

The control actions estimated from the encoders which shows the efforts needed from the robot vehicle to track the speed is given in Fig VI.20a, here The data action are converted from linear velocity (m/sec) to a rotation velocity (revolution per minute : RPM) , the output efforts for the orientation angle control are given in VI.20b. The figures show a reasonable response of system actuators for such a trajectory.

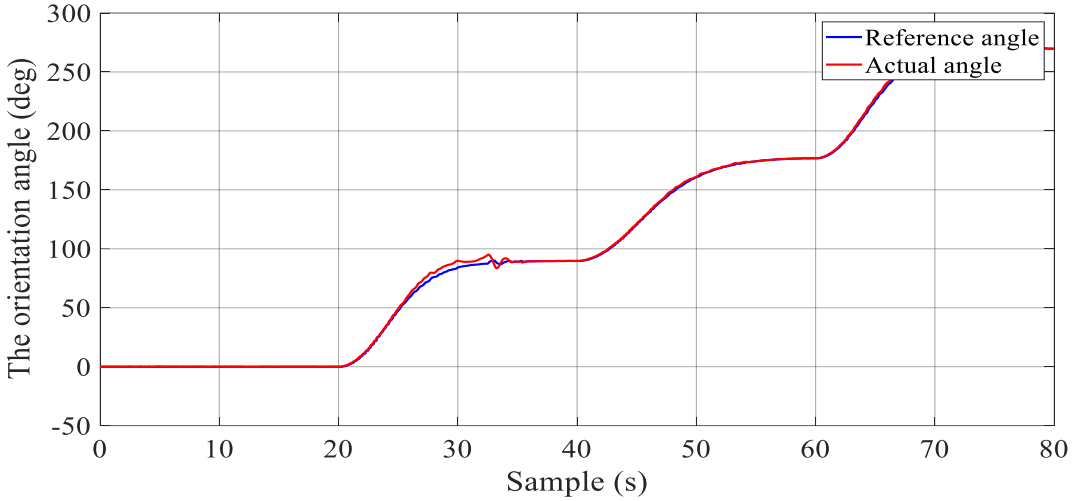
The error of the X coordinate is represented in Fig VI.20c, were FgVI.21: a, b and c shows the errors according to the Y coordinate, the theta orientation and the distance error respectively.

From the previous response estimated from the encoders and the real 2d print of the trajectory, we can say that the control methodologies have been successfully realized with an acceptable tracking error . due to the rapid change of orientation at the rectangle corners.

(a)



(b)



(c)

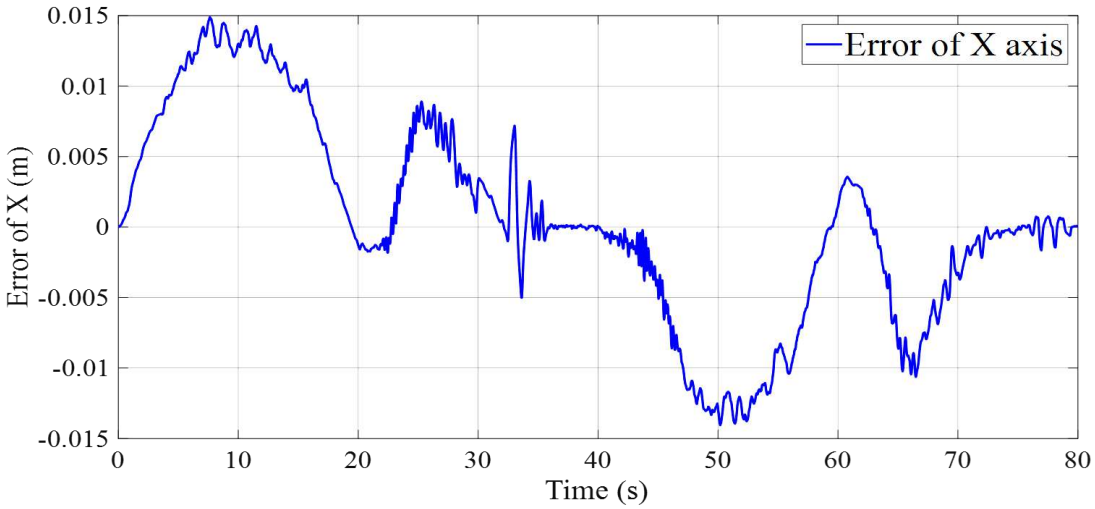
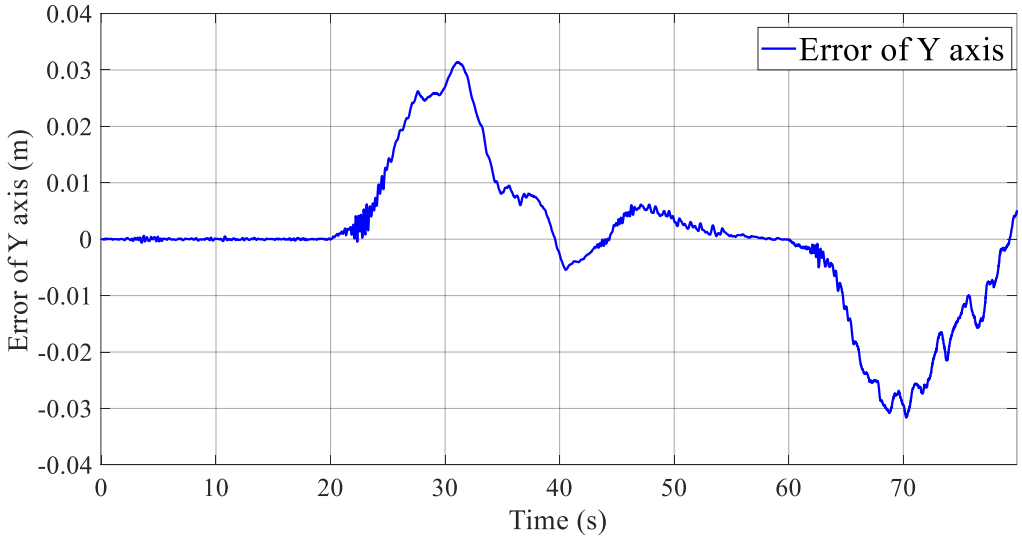
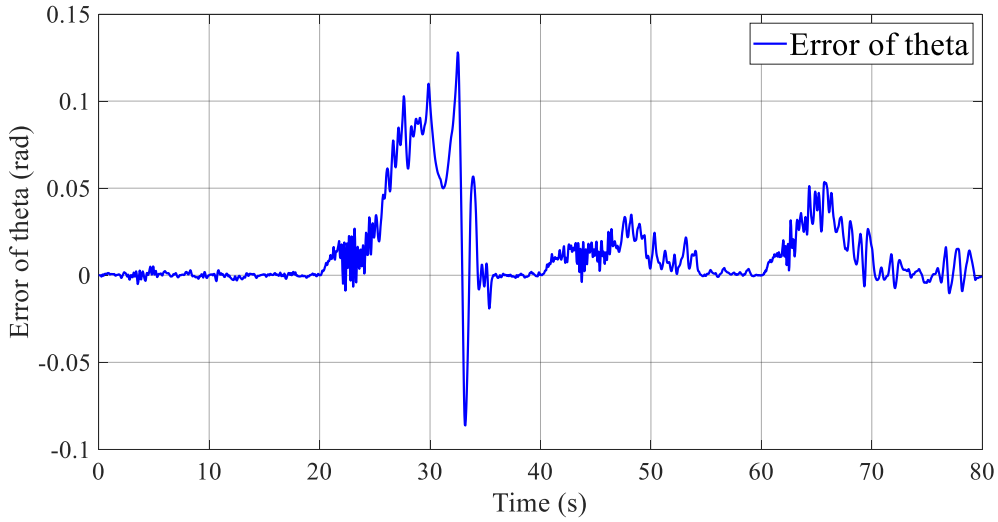


Figure VI.20 control efforts of the rectangular trajectory **:**(a) the tracked speed (b) the tracked orientation (c) error of X axis

(a)



(b)



(c)

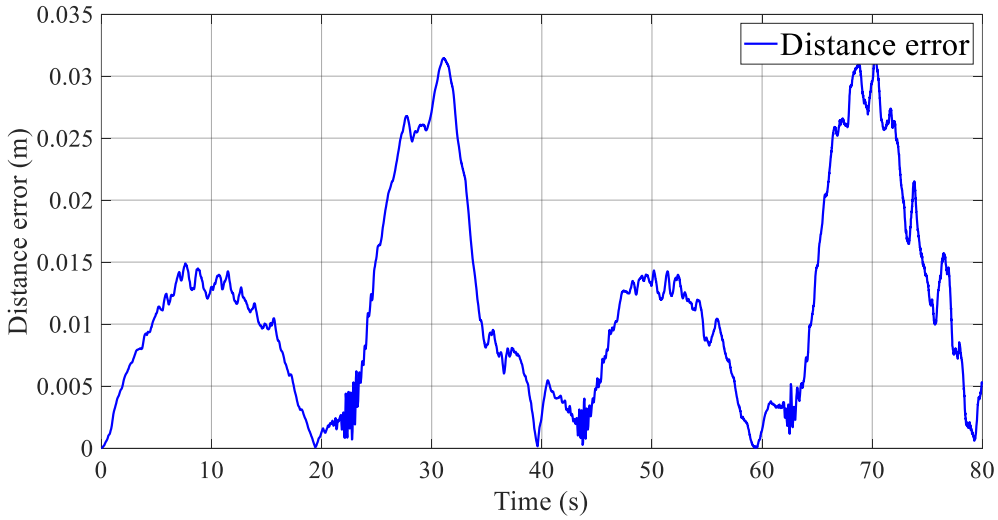


Figure VI.21 tracking errors of the rectangular trajectory**:** (a) error of Y axis (b) error of the orientation (c) error of the distance

VI.6. Conclusions

A real-time implementation of a combined fractional order PID controller with a backstepping technique has been introduced for a self designed robot vehicle. The system uses an encoder sensor to sense its surrounding and localize it self simultaneously. The travelled distances and orientation of the WMR have been estimated based on the quadratic encoders. using the forward kinematics, noise filter has been added to deal with perturbation comes from DC motors. The platform based on the embedded system is investigated using four major case studies to confirm and validate the control technique. The first case study was based on a circular trajectory, which has an explicitly continuous gradient with rotation radius constant. The second case study was based on lemniscates trajectory, which has explicitly continuous gradient with rotation radius changes. The third case study was also smooth shaped trajectory based on a quadrifolium path which is more complicated to realize, finally a rectangular trajectory, which has explicitly non-continuous gradient.The real time experiments have demonstrated that the suggested methods have reached the requirements .

Conclusion

Conclusions and Suggested Future Work

Three new robust control techniques have been proposed to deal with the problem of trajectory tracking of wheeled mobile robot. Those techniques were applied for both continuous and non-continuous gradient trajectories in order to control the movement of the left and right wheels of the autonomous mobile robot using the proposed controllers. The implemented model of the non-holonomic unmanned mobile robot takes into a count both kinematic and dynamic of the vehicle as well the DC actuator characteristics, the three different proposed control methodologies are described as follows:

The first control method aims to combine the traditional PID controller with the ANFIS controller, and applying optimal control theory using meta-heuristic approaches, therefore a comparative study of various nature inspired algorithms was made. Results in simulink show better robustness and efficiency than the old PID controller in terms of the convergence curve, rise time values, settling time and other criteria. However, this technique is only for dealing with dynamics of the vehicle, therefore a new combination technique has been proposed, which is the hybridize of the ANFIS-PID with the backstepping controller. This lately shows smooth control efforts on the right and left wheel velocity

The concept of second control methodology is to design an optimal backstepping-FOPID controller, based on combining the fractional order PID controller with backstepping kinematic controller. In order to deal with dynamics, two blocs of FOPID controller are situated in internal loop, each FOPID controller contains five parameters, need to be regulated. Therefore, the optimal control theory has been applied using a new hybrid nature inspired algorithms namely the hybrid whale grey wolf optimizer (HWGO), therefore a comparative study has been made using several meta-heuristics, will distingue the PSO algorithm, GWO, HPSOGWO, WOA and the HWGO. The criterion of obtained the optimal parameters is based on minimizing the cost function namely integral square error (ISE). The validity of the proposed Backstepping-FOPID controllers has been justified through different trajectories, for both smooth and sharp shaped one.

The third solution proposes a new combination between ANFIS controller and the fractional order PID controller namely ANFIS-FOPID controller. The actual study shows the efficiency of the given ANFIS-FOPID controller. Tuning parameters of the proposed controller are challenging. Therefore, the powerful hybrid whale grey wolf optimizer (HWGO) has been also used in this time, based on the integral square error cost function. A comparative study has been made between the ANFIS-FOPID with respect to the standalone FOPID and the ANFIS, in terms of convergence curve

obtained with lowest error values, and a minimum overshoot observed, rise time, and settling time. As well as several case studies were made for both smooth and sharp-shaped trajectories. Moreover, the combination ANFIS-FOPID with backtepping technique has been demonstrated as a fast learning capability to track the given trajectories. It is observable that the smoothness and faster convergence performance of the tracking error for vehicle velocity and orientation angle have been satisfied. Simulation in MATLAB environment has given the following demonstration and judgment between these three controllers:

The FOPID controller advantage is the fast rise time and settling time value obtained, this controller is ideal for smooth trajectories, but when it comes to sharp shaped trajectories, a high overshoot is observed, which could harm the dc actuators .The ANFIS controller major advantage is the zero-overshoot obtained value, but a slow settling time and rise time value (especially in controlling the angle). The ANFIS-FOPID controller mixes the two advantages of the previous controllers by giving the best settling and rise time value, with an acceptable overshoot, which makes it ideal for both smooth and sharp chapped trajectories. Moreover, a better convergence performance is accomplished when a back-stepping technique for the kinematic control is applied as external loop robot control. This practically guarantees a nearly zero tracking error, where a quadrifolium trajectory used for test.

Furthermore, we have made efforts to validate the proposed algorithms based on real time experiments with the assist of MATLAB package namely simulink support package for arduino. Experimental works were applied on a self designed robot platform, a quadratic encoder sensor is connected to the shaft of the two DC motors to measure the distance, and hence, the localization of the robot can be estimated accordingly. The experimental results are conducted based on four different case studies. They all have demonstrated an effective response in terms of minimum tracking error.

Directions for Future Work

In general, the research objective of this thesis has been strongly reached. The investigation has developed a new control theory and solved the trajectory tracking problem using a new proposed techniques for industrial application. The methods have been verified through a self designed mobile robot platform. A lot of new results as well as problems related to the study have been realized. Consequently, it has been seen that some results can be ameliorated, especially in practice by adding new sensor technology to help the robot for navigation. In addition, some limitations are identified in our work. The error according to the non-continuous gradient trajectories still needs to

be reduced to zero. This reveals that it is possible to extend the work for investigation new constructed problems and developing algorithms that are more efficient. The following points are suggested for the development of the work done in this thesis, such as:

1- Important progresses have been conducted to reduce the tracking error and increase the performance of the process based on the recommended control techniques. Therefore, applying the proposed methodology in another mobile robot types seems to be motivating, such the unmanned Aerial vehicle

2- There still a need of for new nature inspired algorithm that can be used to optimize the parameters of the proposed controllers. In order to get better performance, such as the salp swarm (SSA), equilibrium optimizer (EO), Henry gas solubility optimizer (HGSO). Thus, different optimizing algorithms might provide better results.

3- The fitness functions implemented in this study is based on the integral square error cost function, this fitness functions can be constructed based on different criteria's such as the integral time absolute error ITAE,

4- The navigation problem need to be addressed in the future using vision sensors like the kinect camera and using ultrasonic sensor for the obstacle avoidance, and the Mems sensors technology have a part too in developing the robot platform.

5- The trajectory tracking and navigation problem can be go up to a new level and more precise using new embedded processors like the raspberry pi 4, and the FPGA

6- A multi-agent architecture needs to be done in order to fix the problems that a single mobile robot is can't handle in complicate scenarios.

References

References

References

[1] A. B. E. Al-Mayyahi, "Motion Control of Unmanned Ground Vehicle Using Artificial Intelligence," *Dr. thesis (PhD), Univ. Sussex*, pp. 1–249, 2018, [Online]. Available: http://sro.sussex.ac.uk/76665/1/Al-Mayyahi%2C Auday Basheer Essa.pdf

[2] C. Q. Xiaorui Zhu, Youngshik Kim, Mark Andrew Minor, *Autonomous Mobile Robots in Unknown Outdoor Environments*. CRC Press, 2017.

[3] L. Adouane, *Autonomous Vehicle Navigation*. 2016. doi: 10.1201/b19544.

[4] F. Fahimi, *Autonomous Robots Modeling, Path Planning, and Control*, vol. 53, no. 9. 2013.

[5] Y. Ichikawa and N. Ozaki, *Autonomous Mobile Robot.*, vol. 2, no. 1. 1985. doi: 10.4324/9781003031215-5-5.

[6] G. Dudek, *Computational Principles of Mobile Robotics (Book)*. 2015.

[7] A. Bhaumik, *From AI to Robotics*. 2018. doi: 10.1201/9781315372549.

[8] M. Sani, "Real-time Control of mobile robots using Model Predictive Control and Game-theoretic approaches," 2016, [Online]. Available: https://tel.archives-ouvertes.fr/tel-03685323

[9] G. Klančar, A. Zdešar, S. Blažič, and I. Škrjanc, *Wheeled Mobile Robotics Wheeled Mobile*. 2017. [Online]. Available: https://www.sciencedirect.com/book/9780128042045/wheeled-mobile-robotics

[10] C. C. D. E. Lille, "Planification de trajectoire et commande pour les robots mobiles non-holonomes Path planning and control of non-holonomic mobile robots," 2013.

[11] D. Control, "Optimization Based Solutions for Control and State," vol. 2016, no. December, 2016.

[12] A. Al-Araji, "Design of a cognitive neural predictive controller for mobile robot," no. September, 2012, [Online]. Available: http://bura.brunel.ac.uk/handle/2438/7068

[13] R. C. Solea, : "Department of Chemical Engineering : Faculty of Sciences and Technology : University of Coimbra," *Dr. Philos. Dep. Chem. Eng. Fac. Sci. Technol. ,University Coimbra*, no. March, 2009.

[14] O. W. Abdulwahhab and N. H. Abbas, "Design and Stability Analysis of a Fractional Order State Feedback Controller for Trajectory Tracking of a Differential Drive Robot," *Int. J. Control. Autom. Syst.*, vol. 16, no. 6, pp. 2790–2800, 2018, doi: 10.1007/s12555-017-0234-8.

[15] O. Mohareri, R. Dhaouadi, and A. B. Rad, "Indirect adaptive tracking control of a nonholonomic mobile robot via neural networks," *Neurocomputing*, vol. 88, pp. 54–66, 2012, doi: 10.1016/j.neucom.2011.06.035.

[16] T. Q. Khai, Y. J. Ryoo, W. R. Gill, and D. Y. Im, “Design of Kinematic Controller Based on Parameter Tuning by Fuzzy Inference System for Trajectory Tracking of Differential-Drive Mobile Robot,” *Int. J. Fuzzy Syst.*, vol. 22, no. 6, pp. 1972–1978, 2020, doi: 10.1007/s40815-020-00842-9.

[17] T. Q. Khai and Y. J. Ryoo, “Design of adaptive kinematic controller using radial basis function neural network for trajectory tracking control of differential-drive mobile robot,” *Int. J. Fuzzy Log. Intell. Syst.*, vol. 19, no. 4, pp. 349–359, 2019, doi: 10.5391/IJFIS.2019.19.4.349.

[18] M. Gheisarnejad and M. H. Khooban, “Supervised control strategy in trajectory tracking for a wheeled mobile robot,” *IET Collab. Intell. Manuf.*, vol. 1, no. 1, pp. 3–9, 2019, doi: 10.1049/iet-cim.2018.0007.

[19] X. Wu, P. Jin, T. Zou, Z. Qi, H. Xiao, and P. Lou, “Backstepping Trajectory Tracking Based on Fuzzy Sliding Mode Control for Differential Mobile Robots,” *J. Intell. Robot. Syst. Theory Appl.*, vol. 96, no. 1, pp. 109–121, 2019, doi: 10.1007/s10846-019-00980-9.

[20] G. Wang, C. Zhou, Y. Yu, and X. Liu, “Adaptive sliding mode trajectory tracking control for WMR considering skidding and slipping via extended state observer,” *Energies*, vol. 12, no. 17, 2019, doi: 10.3390/en12173305.

[21] P. T. Phuc, T. P. Tho, N. D. X. Hai, and N. T. Thinh, “Design of Adaptive Fuzzy Sliding Mode Controller for Mobile Robot,” *Int. J. Mech. Eng. Robot. Res.*, vol. 10, no. 2, pp. 54–59, 2021, doi: 10.18178/ijmerr.10.2.54-59.

[22] B. Moudoud, H. Aissaoui, and M. Diany, “Robust trajectory tracking control based on sliding mode of Differential Driving Four-Wheeled Mobile Robot,” *6th Int. Conf. Optim. Appl. ICOA 2020 - Proc.*, 2020, doi: 10.1109/ICOA49421.2020.9094510.

[23] A. Nikranjbar, M. Haidari, and A. A. Atai, “Adaptive Sliding Mode Tracking Control of Mobile Robot in Dynamic Environment Using Artificial Potential Fields,” *J. Comput. Robot.*, vol. 11, no. 1, pp. 1–14, 2018.

[24] F. N. Martins, W. C. Celeste, R. Carelli, M. Sarcinelli-Filho, and T. F. Bastos-Filho, “An adaptive dynamic controller for autonomous mobile robot trajectory tracking,” *Control Eng. Pract.*, vol. 16, no. 11, pp. 1354–1363, 2008, doi: 10.1016/j.conengprac.2008.03.004.

[25] N. T. Binh, N. A. Tung, D. P. Nam, and N. H. Quang, “An Adaptive Backstepping Trajectory Tracking Control of a Tractor Trailer Wheeled Mobile Robot,” *Int. J. Control. Autom. Syst.*, vol. 17, no. 2, pp. 465–473, 2019, doi: 10.1007/s12555-017-0711-0.

[26] G. Wang, X. Liu, Y. Zhao, and S. Han, “Neural Network-Based Adaptive Motion Control for a Mobile Robot with Unknown Longitudinal Slipping,” *Chinese J. Mech. Eng. (English Ed.*, vol. 32, no. 1, 2019, doi: 10.1186/s10033-019-0373-3.

[27] F. G. Rossomando, C. Soria, and R. Carelli, “Autonomous mobile robots navigation using RBF neural compensator,” *Control Eng. Pract.*, vol. 19, no. 3, pp. 215–222, 2011, doi: 10.1016/j.conengprac.2010.11.011.

[28] K. Khnissi, C. Ben Jabeur, and H. Seddik, “A smart mobile robot commands predictor using recursive neural network,” *Rob. Auton. Syst.*, vol. 131, p. 103593, 2020, doi: 10.1016/j.robot.2020.103593.

[29] P. Bozek, Y. L. Karavaev, A. A. Ardentov, and K. S. Yefremov, “Neural network control of a wheeled mobile robot based on optimal trajectories,” *Int. J. Adv. Robot. Syst.*, vol. 17, no. 2, pp. 1–10, 2020, doi: 10.1177/1729881420916077.

[30] T. Das and I. N. Kar, “Design and implementation of an adaptive fuzzy logic-based controller for wheeled mobile robots,” *IEEE Trans. Control Syst. Technol.*, vol. 14, no. 3, pp. 501–510, 2006, doi: 10.1109/TCST.2006.872536.

[31] A. Štefek, V. T. Pham, V. Krivanek, and K. L. Pham, “Optimization of fuzzy logic controller used for a differential drive wheeled mobile robot,” *Appl. Sci.*, vol. 11, no. 13, 2021, doi: 10.3390/app11136023.

[32] V. Q. Ha, S. H. T. Pham, and N. T. T. Vu, “Adaptive Fuzzy Type-II Controller for Wheeled Mobile Robot with Disturbances and Wheelslips,” *J. Robot.*, vol. 2021, 2021, doi: 10.1155/2021/6946210.

[33] S. Slama, A. Errachdi, and M. Benrejeb, “Neural Adaptive PID and Neural Indirect Adaptive Control Switch Controller for Nonlinear MIMO Systems,” *Math. Probl. Eng.*, vol. 2019, 2019, doi: 10.1155/2019/7340392.

[34] P. Kumar, S. Chatterjee, D. Shah, U. K. Saha, and S. Chatterjee, “On comparison of tuning method of FOPID controller for controlling field controlled DC servo motor,” *Cogent Eng.*, vol. 4, no. 1, 2017, doi: 10.1080/23311916.2017.1357875.

[35] Z. Bingul and O. Karahan, “Comparison of PID and FOPID controllers tuned by PSO and ABC algorithms for unstable and integrating systems with time delay,” *Optim. Control Appl. Methods*, vol. 39, no. 4, pp. 1431–1450, 2018, doi: 10.1002/oca.2419.

[36] J. Ye, “Adaptive control of nonlinear PID-based analog neural networks for a nonholonomic mobile robot,” *Neurocomputing*, vol. 71, no. 7–9, pp. 1561–1565, 2008, doi: 10.1016/j.neucom.2007.04.014.

[37] G. Pei *et al.*, “An improved pid controller for the compliant constant-force actuator based on bp neural network and smith predictor,” *Appl. Sci.*, vol. 11, no. 6, 2021, doi: 10.3390/app11062685.

[38] S. S. D. Xu, H. C. Huang, T. C. Chiu, and S. K. Lin, “Biologically-inspired learning and adaptation of self-evolving control for networked mobile robots,” *Appl. Sci.*, vol. 9, no. 5, 2019, doi: 10.3390/app9051034.

[39] D. K. Tiep, K. Lee, D. Y. Im, B. Kwak, and Y. J. Ryoo, “Design of fuzzy-PID controller for path tracking of mobile robot with differential drive,” *Int. J. Fuzzy Log. Intell. Syst.*, vol. 18, no. 3, pp. 220–228, 2018, doi: 10.5391/IJFIS.2018.18.3.220.

[40] T. A. Mai, T. S. Dang, D. T. Duong, V. C. Le, and S. Banerjee, “A combined backstepping and adaptive fuzzy PID approach for trajectory tracking of autonomous mobile robots,” *J. Brazilian Soc. Mech. Sci. Eng.*, vol. 43, no. 3, pp. 1–13, 2021, doi: 10.1007/s40430-020-02767-8.

[41] H. Zhou, R. Chen, S. Zhou, and Z. Liu, “Design and analysis of a drive system for a series manipulator based on orthogonal-fuzzy PID control,” *Electron.*, vol. 8, no. 9, 2019, doi: 10.3390/electronics8091051.

[42] C. Ben Jabeur and H. Seddik, “Design of a PID optimized neural networks and PD fuzzy logic controllers for a two-wheeled mobile robot,” *Asian J. Control*, vol. 23, no. 1, pp. 23–41, 2021, doi: 10.1002/asjc.2356.

[43] S. Wang, X. Yin, P. Li, M. Zhang, and X. Wang, “Trajectory Tracking Control for Mobile Robots Using Reinforcement Learning and PID,” *Iran. J. Sci. Technol. - Trans. Electr. Eng.*, vol. 44, no. 3, pp. 1059–1068, 2019, doi: 10.1007/s40998-019-00286-4.

[44] K. Orman, A. Basci, and A. Derdiyok, “Speed and direction angle control of four wheel drive skid-steered mobile robot by using fractional Order PI controller,” *Elektron. ir Elektrotechnika*, vol. 22, no. 5, pp. 14–19, 2016, doi: 10.5755/j01.eie.22.5.16337.

[45] S. K. Mishra and D. Chandra, “Stabilization and Tracking Control of Inverted Pendulum Using Fractional Order PID Controllers,” *J. Eng. (United Kingdom)*, vol. 2014, 2014, doi: 10.1155/2014/752918.

[46] L. Zhang, L. Liu, and S. Zhang, “Design, Implementation, and Validation of Robust Fractional-Order PD Controller for Wheeled Mobile Robot Trajectory Tracking,” *Complexity*, vol. 2020, 2020, doi: 10.1155/2020/9523549.

[47] H. O. Erkol, “Optimal PIλ Dμ controller design for two wheeled inverted pendulum,” *IEEE Access*, vol. 6, pp. 75709–75717, 2018, doi: 10.1109/ACCESS.2018.2883504.

[48] K. Kankhunthodl, V. Kongratana, A. Numsomran, and V. Tipsuwanporn, “Self-balancing Robot Control Using Fractional-Order PID Controller,” *Lect. Notes Eng. Comput. Sci.*, vol. 2239, no. May, pp. 77–82, 2019.

[49] N. D. Bernardes, F. A. Castro, M. A. S. L. Cuadros, P. F. Salarolli, G. M. Almeida, and C. J. Munaro, “Fuzzy logic in auto-tuning of fractional PID and backstepping tracking control of a differential mobile robot,” *J. Intell. Fuzzy Syst.*, vol. 37, no. 4, pp. 4951–4964, 2019, doi: 10.3233/JIFS-181431.

[50] F. Meng, S. Liu, and K. Liu, “Design of an Optimal Fractional Order PID for Constant Tension Control System,” *IEEE Access*, vol. 8, pp. 58933–58939, 2020, doi: 10.1109/ACCESS.2020.2983059.

[51] R. Cajo, T. T. Mac, D. Plaza, C. Copot, R. De Keyser, and C. Ionescu, “A Survey on Fractional Order Control Techniques for Unmanned Aerial and Ground Vehicles,” *IEEE Access*, vol. 7, pp. 66864–66878, 2019, doi: 10.1109/ACCESS.2019.2918578.

[52] R. Rajesh, “Optimal tuning of FOPID controller based on PSO algorithm with reference model for a single conical tank system,” *SN Appl. Sci.*, vol. 1, no. 7, pp. 1–14, 2019, doi: 10.1007/s42452-019-0754-3.

[53] D. Abdelhamid, T. Bouden, and A. Boulkroune, “Design of fractional-order PID controller (FOPID) for a class of fractional-order MIMO systems using a particle swarm optimization (PSO)

approach," *2013 3rd Int. Conf. Syst. Control. ICSC 2013*, pp. 1055–1060, 2013, doi: 10.1109/ICoSC.2013.6750985.

[54] L. H. Abood and B. K. Oleiwi, "Design of fractional order PID controller for AVR system using whale optimization algorithm," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 23, no. 3, pp. 1410–1418, 2021, doi: 10.11591/ijeecs.v23.i3.pp1410-1418.

[55] S. K. Verma, S. Yadav, and S. K. Nagar, "Optimization of Fractional Order PID Controller Using Grey Wolf Optimizer," *J. Control. Autom. Electr. Syst.*, vol. 28, no. 3, pp. 314–322, 2017, doi: 10.1007/s40313-017-0305-3.

[56] J. Bhookya and R. K. Jatoth, "Optimal FOPID/PID controller parameters tuning for the AVR system based on sine–cosine-algorithm," *Evol. Intell.*, vol. 12, no. 4, pp. 725–733, 2019, doi: 10.1007/s12065-019-00290-x.

[57] S. Ekinci, D. Izci, and B. Hekimoglu, "Henry Gas Solubility Optimization Algorithm Based FOPID Controller Design for Automatic Voltage Regulator," *2nd Int. Conf. Electr. Commun. Comput. Eng. ICECCE 2020*, no. June, pp. 2–7, 2020, doi: 10.1109/ICECCE49384.2020.9179406.

[58] I. A. Khan, A. S. Alghamdi, T. A. Jumani, A. Alamgir, A. B. Awan, and A. Khidrani, "Salp Swarm Optimization Algorithm-Based Fractional Order PID Controller for Dynamic Response and Stability Enhancement of an Automatic Voltage Regulator System," *Electronics*, vol. 8, no. 12, p. 1472, 2019, doi: 10.3390/electronics8121472.

[59] A. K. Mishra, S. R. Das, P. K. Ray, R. K. Mallick, A. Mohanty, and D. K. Mishra, "PSO-GWO Optimized Fractional Order PID Based Hybrid Shunt Active Power Filter for Power Quality Improvements," *IEEE Access*, vol. 8, pp. 74497–74512, 2020, doi: 10.1109/ACCESS.2020.2988611.

[60] B. Hekimoglu, "Optimal Tuning of Fractional Order PID Controller for DC Motor Speed Control via Chaotic Atom Search Optimization Algorithm," *IEEE Access*, vol. 7, no. c, pp. 38100–38114, 2019, doi: 10.1109/ACCESS.2019.2905961.

[61] H. Arpaci and O. F. Özgüven, "ANFIS & PIλ Dμ controller design and comparison for overhead cranes," *Indian J. Eng. Mater. Sci.*, vol. 18, no. 3, pp. 191–203, 2011.

[62] A. Saxena, Y. M. Dubey, M. Kumar, and A. Saxena, "Performance Comparison of ANFIS, FOPID-PSO and FOPID-Fuzzy Tuning Methodology for Optimizing Response of High-Performance Drilling Machine," *IETE J. Res.*, 2021, doi: 10.1080/03772063.2021.1933625.

[63] M. Imen, M. Mansouri, and M. A. Shoorehdeli, "Tracking control of mobile robot using ANFIS," *2011 IEEE Int. Conf. Mechatronics Autom. ICMA 2011*, no. August, pp. 422–427, 2011, doi: 10.1109/ICMA.2011.5985695.

[64] A. Al-Mayyahi, W. Wang, and P. Birch, "Adaptive neuro-fuzzy technique for autonomous ground vehicle navigation," *Robotics*, vol. 3, no. 4, pp. 349–370, 2014, doi: 10.3390/robotics3040349.

[65] K. Premkumar and B. V. Manikandan, “Adaptive Neuro-Fuzzy Inference System based speed controller for brushless DC motor,” *Neurocomputing*, vol. 138, pp. 260–270, 2014, doi: 10.1016/j.neucom.2014.01.038.

[66] B. Selma, S. Chouraqui, and H. Abouaïssa, “Optimization of ANFIS controllers using improved ant colony to control an UAV trajectory tracking task,” *SN Appl. Sci.*, vol. 2, no. 5, pp. 1–18, 2020, doi: 10.1007/s42452-020-2236-z.

[67] M. Elsisi, M. Q. Tran, K. Mahmoud, M. Lehtonen, and M. M. F. Darwish, “Robust design of ANFIS-based blade pitch controller for wind energy conversion systems against wind speed fluctuations,” *IEEE Access*, vol. 9, pp. 37894–37904, 2021, doi: 10.1109/ACCESS.2021.3063053.

[68] F. Pang, M. Luo, X. Xu, and Z. Tan, “Path tracking control of an omni-directional service robot based on model predictive control of adaptive neural-fuzzy inference system,” *Appl. Sci.*, vol. 11, no. 2, pp. 1–18, 2021, doi: 10.3390/app11020838.

[69] I. M. El-Hasnony, S. I. Barakat, and R. R. Mostafa, “Optimized ANFIS Model Using Hybrid Metaheuristic Algorithms for Parkinson’s Disease Prediction in IoT Environment,” *IEEE Access*, vol. 8, pp. 119252–119270, 2020, doi: 10.1109/ACCESS.2020.3005614.

[70] A. F. Bendary, A. Y. Abdelaziz, M. M. Ismail, K. Mahmoud, M. Lehtonen, and M. M. F. Darwish, “Proposed anfis based approach for fault tracking, detection, clearing and rearrangement for photovoltaic system,” *Sensors*, vol. 21, no. 7, 2021, doi: 10.3390/s21072269.

[71] Hidayat, S. Pramonohadi, Sarjiya, and Suharyanto, “A comparative study of PID, ANFIS and hybrid PID-ANFIS controllers for speed control of Brushless DC Motor drive,” *Proceeding - 2013 Int. Conf. Comput. Control. Informatics Its Appl. “Recent Challenges Comput. Control Informatics”, IC3INA 2013*, pp. 117–122, 2013, doi: 10.1109/IC3INA.2013.6819159.

[72] Y. Guo and M. E. A. Mohamed, “Speed Control of Direct Current Motor Using ANFIS Based Hybrid P-I-D Configuration Controller,” *IEEE Access*, vol. 8, pp. 125638–125647, 2020, doi: 10.1109/ACCESS.2020.3007615.

[73] D. Pal and S. K. Bhagat, “Design and Analysis of Optimization based Integrated ANFIS-PID Controller for Networked Controlled Systems (NCSs).,” *Cogent Eng.*, vol. 7, no. 1, 2020, doi: 10.1080/23311916.2020.1772944.

[74] H. Housny, E. A. Chater, and H. El Fadil, “Multi closed-loop adaptive neuro-fuzzy inference system for quadrotor position control,” *Adv. Sci. Technol. Eng. Syst.*, vol. 5, no. 5, pp. 526–535, 2020, doi: 10.25046/AJ050565.

[75] H. Chaudhary, V. Panwar, S. N, and R. Prasad, “ANFIS PD+I Based Hybrid Force/ Position Control of an Industrial Robot Manipulator,” *Int. J. Mater. Mech. Manuf.*, vol. 2, no. 2, pp. 107–112, 2014, doi: 10.7763/ijmmm.2014.v2.110.

[76] G. A. R. Ibraheem, A. T. Azar, I. K. Ibraheem, and A. J. Humaidi, “A Novel Design of a Neural Network-Based Fractional PID Controller for Mobile Robots Using Hybridized Fruit Fly and Particle Swarm Optimization,” *Complexity*, vol. 2020, 2020, doi: 10.1155/2020/3067024.

[77] A. Al-Mayyahi, W. Wang, and P. Birch, “Levenberg-Marquardt optimised neural networks for trajectory tracking of autonomous ground vehicles,” *Int. J. Mechatronics Autom.*, vol. 5, no. 2–3, pp. 140–153, 2015, doi: 10.1504/IJMA.2015.075960.

[78] R. Sharma, K. P. S. Rana, and V. Kumar, “Performance analysis of fractional order fuzzy PID controllers applied to a robotic manipulator,” *Expert Syst. Appl.*, vol. 41, no. 9, pp. 4274–4289, 2014, doi: 10.1016/j.eswa.2013.12.030.

[79] R. D. Ahmad Abu Hatab, “Dynamic Modelling of Differential-Drive Mobile Robots using Lagrange and Newton-Euler Methodologies: A Unified Framework,” *Adv. Robot. Autom.*, vol. 02, no. 02, 2013, doi: 10.4172/2168-9695.1000107.

[80] T. Fukao, H. Nakagawa, and N. Adachi, “Adaptive tracking control of a nonholonomic mobile robot,” *IEEE Trans. Robot. Autom.*, vol. 16, no. 5, pp. 609–615, 2000, doi: 10.1109/70.880812.

[81] R. Fierro and F. L. Lewis, “Control of a nonholonomic mobile robot: Backsteppins kinematics into dynamics,” *J. Robot. Syst.*, vol. 14, no. 3, pp. 149–163, 1997, doi: 10.1002/(sici)1097-4563(199703)14:3<149::aid-rob1>3.3.co;2-n.

[82] OMID MOHARERI, “Mobile Robot Trajectory Tracking Using Neural Networks,” no. December, pp. 1–235, 2009, [Online]. Available: http://hdl.handle.net/11073/133.

[83] B. E. Demir, R. Bayir, and F. Duran, “Real-time trajectory tracking of an unmanned aerial vehicle using a self-tuning fuzzy proportional integral derivative controller,” *Int. J. Micro Air Veh.*, vol. 8, no. 4, pp. 252–268, 2016, doi: 10.1177/1756829316675882.

[84] F. A. Şenel, F. Gökçe, A. S. Yüksel, and T. Yiğit, “A novel hybrid PSO–GWO algorithm for optimization problems,” *Eng. Comput.*, vol. 35, no. 4, pp. 1359–1373, 2019, doi: 10.1007/s00366-018-0668-5.

[85] N. Singh and S. B. Singh, “Hybrid Algorithm of Particle Swarm Optimization and Grey Wolf Optimizer for Improving Convergence Performance,” *J. Appl. Math.*, vol. 2017, 2017, doi: 10.1155/2017/2030489.

[86] S. Mirjalili, “Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems,” *Neural Comput. Appl.*, vol. 27, no. 4, pp. 1053–1073, 2016, doi: 10.1007/s00521-015-1920-1.

[87] M. Yasen, N. Al-Madi, and N. Obeid, “Optimizing Neural Networks using Dragonfly Algorithm for Medical Prediction,” *2018 8th Int. Conf. Comput. Sci. Inf. Technol. CSIT 2018*, pp. 71–76, 2018, doi: 10.1109/CSIT.2018.8486178.

[88] C. W. Reynolds, “Flocks, herds, and schools: A distributed behavioral model,” *Proc. 14th Annu. Conf. Comput. Graph. Interact. Tech. SIGGRAPH 1987*, vol. 21, no. 4, pp. 25–34, 1987, doi: 10.1145/37401.37406.

[89] M. A. Salam, H. M. Zawbaa, E. Emary, K. K. A. Ghany, and B. Parv, “A hybrid dragonfly algorithm with extreme learning machine for prediction,” *Proc. 2016 Int. Symp. Innov. Intell. Syst. Appl. INISTA 2016*, 2016, doi: 10.1109/INISTA.2016.7571839.

[90] S. Mirjalili, “The ant lion optimizer,” *Adv. Eng. Softw.*, vol. 83, pp. 80–98, 2015, doi: 10.1016/j.advengsoft.2015.01.010.

[91] S. Saha and V. Mukherjee, “A novel quasi-oppositional chaotic antlion optimizer for global optimization,” *Appl. Intell.*, vol. 48, no. 9, pp. 2628–2660, 2018, doi: 10.1007/s10489-017-1097-7.

[92] E. Gupta and A. Saxena, “Performance Evaluation of Antlion Optimizer Based Regulator in Automatic Generation Control of Interconnected Power System,” *J. Eng. (United Kingdom)*, vol. 2016, 2016, doi: 10.1155/2016/4570617.

[93] D. Karaboga and B. Basturk, “On the performance of artificial bee colony (ABC) algorithm,” *Appl. Soft Comput. J.*, vol. 8, no. 1, pp. 687–697, 2008, doi: 10.1016/j.asoc.2007.05.007.

[94] X. Yu, W. Chen, and X. Zhang, “An Artificial Bee Colony Algorithm for Solving Constrained Optimization Problems,” *Proc. 2018 2nd IEEE Adv. Inf. Manag. Commun. Electron. Autom. Control Conf. IMCEC 2018*, pp. 2663–2666, 2018, doi: 10.1109/IMCEC.2018.8469371.

[95] S. Mirjalili, S. M. Mirjalili, and A. Lewis, “Grey Wolf Optimizer,” *Adv. Eng. Softw.*, vol. 69, pp. 46–61, 2014, doi: 10.1016/j.advengsoft.2013.12.007.

[96] S. Saremi, S. Mirjalili, and A. Lewis, “Grasshopper Optimisation Algorithm: Theory and application,” *Adv. Eng. Softw.*, vol. 105, pp. 30–47, 2017, doi: 10.1016/j.advengsoft.2017.01.004.

[97] M. Bhuyan, A. K. Barik, and D. C. Das, “GOA optimised frequency control of solar-thermal/sea-wave/biodiesel generator based interconnected hybrid microgrids with DC link,” *Int. J. Sustain. Energy*, vol. 39, no. 7, pp. 615–633, 2020, doi: 10.1080/14786451.2020.1741589.

[98] A. Seifi, M. Ehteram, V. P. Singh, and A. Mosavi, “Modeling and uncertainty analysis of groundwater level using six evolutionary optimization algorithms hybridized with ANFIS, SVM, and ANN,” *Sustain.*, vol. 12, no. 10, 2020, doi: 10.3390/SU12104023.

[99] S. Mirjalili, “Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm,” *Knowledge-Based Syst.*, vol. 89, no. July, pp. 228–249, 2015, doi: 10.1016/j.knosys.2015.07.006.

[100] R. K. Singh, S. Gangwar, D. K. Singh, and V. K. Pathak, “A novel hybridization of artificial neural network and moth-flame optimization (ANN–MFO) for multi-objective optimization in magnetic abrasive finishing of aluminium 6060,” *J. Brazilian Soc. Mech. Sci. Eng.*, vol. 41, no. 6, pp. 1–19, 2019, doi: 10.1007/s40430-019-1778-8.

[101] G. I. Sayed and A. E. Hassanien, “A hybrid SA-MFO algorithm for function optimization and engineering design problems,” *Complex Intell. Syst.*, vol. 4, no. 3, pp. 195–212, 2018, doi: 10.1007/s40747-018-0066-z.

[102] J. S. R. Jang, “ANFIS: Adaptive-Network-Based Fuzzy Inference System,” *IEEE Trans. Syst. Man Cybern.*, vol. 23, no. 3, pp. 665–685, 1993, doi: 10.1109/21.256541.

[103] M. Buragohain and C. Mahanta, “A novel approach for ANFIS modelling based on full factorial design,” *Appl. Soft Comput. J.*, vol. 8, no. 1, pp. 609–625, 2008, doi: 10.1016/j.asoc.2007.03.010.

[104] B. Bektas Ekici and U. T. Aksoy, “Prediction of building energy needs in early stage of design by using ANFIS,” *Expert Syst. Appl.*, vol. 38, no. 5, pp. 5352–5358, 2011, doi: 10.1016/j.eswa.2010.10.021.

[105] M. A. Boyacioglu and D. Avci, “An adaptive network-based fuzzy inference system (ANFIS) for the prediction of stock market return: The case of the Istanbul stock exchange,” *Expert Syst. Appl.*, vol. 37, no. 12, pp. 7908–7912, 2010, doi: 10.1016/j.eswa.2010.04.045.

[106] M. J. Mohamed, “Design a Fuzzy PID Controller for Trajectory Tracking of Mobile Robot,” *Eng. Technol. J.*, vol. 36, no. 1, 2018, doi: 10.30684/etj.36.1a.15.

[107] N. Sadati, M. Zamani, and P. Mohajerin, “Optimum design of fractional order PID for MIMO and SISO systems using particle swarm optimization techniques,” *Proc. 2007 4th IEEE Int. Conf. Mechatronics, ICM 2007*, no. May, pp. 1–6, 2007, doi: 10.1109/ICMECH.2007.4280056.

[108] R. Almeida, A. B. Malinowska, and D. F. M. Torres, “A fractional calculus of variations for multiple integrals with application to vibrating string,” *J. Math. Phys.*, vol. 51, no. 3, 2010, doi: 10.1063/1.3319559.

[109] H. Wu and T. Zhao, “Design of interval type-2 fuzzy fractional order PID controller based on particle swarm optimization,” *J. Phys. Conf. Ser.*, vol. 2258, no. 1, 2022, doi: 10.1088/1742-6596/2258/1/012063.

[110] D. Pedro and M. De Oliveira, “Fractional Robust System Control,” *Changes*, vol. 10, no. October, pp. 1–5, 2005, [Online]. Available: http://web.ist.utl.pt/duarte.valerio/PhDthesis.pdf

[111] D. Xue, C. Zhao, and Y. Q. Chen, “Fractional order PID control of A DC-motor with elastic shaft: A case study,” *Proc. Am. Control Conf.*, vol. 2006, pp. 3182–3187, 2006, doi: 10.1109/acc.2006.1657207.

[112] G. Magdy, G. Shabib, A. A. Elbaset, and Y. Mitani, “Renewable Power Systems Dynamic Security,” *Power Syst.*, vol. 44, no. 1, pp. 1–13, 2020.

[113] D. Sierociuk and M. Macias, “Comparison of variable fractional order PID controller for different types of variable order derivatives,” *Proc. 2013 14th Int. Carpathian Control Conf. ICCC 2013*, pp. 334–339, 2013, doi: 10.1109/CarpathianCC.2013.6560565.

[114] S. Mirjalili and A. Lewis, “The Whale Optimization Algorithm,” *Adv. Eng. Softw.*, vol. 95, pp. 51–67, 2016, doi: 10.1016/j.advengsoft.2016.01.008.

[115] A. M. Hemeida, S. Alkhalaf, A. Mady, E. A. Mahmoud, M. E. Hussein, and A. M. Baha Eldin, “Implementation of nature-inspired optimization algorithms in some data mining tasks,” *Ain Shams Eng. J.*, vol. 11, no. 2, pp. 309–318, 2020, doi: 10.1016/j.asej.2019.10.003.

[116] A. Korashy, S. Kamel, F. Jurado, and A. R. Youssef, “Hybrid Whale Optimization Algorithm and Grey Wolf Optimizer Algorithm for Optimal Coordination of Direction Overcurrent

Relays," *Electr. Power Components Syst.*, vol. 47, no. 6–7, pp. 644–658, 2019, doi: 10.1080/15325008.2019.1602687.

[117] M. Fannakh, M. L. Ehafyani, and S. Zouggar, "Hardware implementation of the fuzzy logic MPPT in an Arduino card using a Simulink support package for PV application," *IET Renew. Power Gener.*, vol. 13, no. 3, pp. 510–518, 2019, doi: 10.1049/iet-rpg.2018.5667.

[118] P. Parikh, R. Vasani, S. Sheth, and J. Gohil, "Actuation of Electro-Pneumatic System using MATLAB Simulink and Arduino Controller- A case of a Mechatronics systems Lab," vol. 137, pp. 59–64, 2017, doi: 10.2991/iccasp-16.2017.10.

[119] D. D. E. La, E. D. E. Ingenieros, and I. D. E. Barcelona, "Escola Tècnica Superior d ' Enginyeria Industrial de Barcelona Universitat Politècnica de Catalunya," 2008.

[120] A. Tepljakov, E. Petlenkov, and J. Belikov, "FOMCON : a MATLAB Toolbox for Fractional-order System Identification and Control," *Int. J. Microelectron. Comput. Sci.*, vol. 2, no. 2, pp. 51–62, 2011.