

PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA

MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH



National Polytechnic School  
Department of Automation  
Process Control Laboratory



Doctoral thesis

Presented by:

Yasmine SAIDI

Magister in Automatic, EMP

In order to obtain the title of

Doctor in Automatic

Option: Automatic Industrial Systems

Entitled:

**Robust Mobile Robot Navigation using Fuzzy Logic Controllers with  
Robustness Analysis**

Thesis defended on March 11<sup>th</sup>, 2021

The Board of Examiners composed of:

F. BOUDJEMA	Professor, ESDAT	President
M. TADJINE	Professor, ENP	Supervisor
A. NEMRA	MC/A, EMP	Supervisor
N. ACHOUR	Professor, USTHB	Examiner
M. HAMERLAIN	Researches Director, CDTA	Examiner
M. HADDAD	Professor, EMP	Examiner
R. ILLOUL	MC/A, ENP	Examiner

*ENP 2021*



PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA

MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH



National Polytechnic School  
Department of Automation  
Process Control Laboratory



Doctoral thesis

Presented by:

Yasmine SAIDI

Magister in Automatic, EMP

In order to obtain the title of

Doctor in Automatic

Option: Automatic Industrial Systems

Entitled:

**Robust Mobile Robot Navigation using Fuzzy Logic Controllers with  
Robustness Analysis**

Thesis defended on March 11<sup>th</sup>, 2021

The Board of Examiners composed of:

F. BOUDJEMA	Professor, ESDAT	President
M. TADJINE	Professor, ENP	Supervisor
A. NEMRA	MC/A, EMP	Supervisor
N. ACHOUR	Professor, USTHB	Examiner
M. HAMERLAIN	Researches Director, CDTA	Examiner
M. HADDAD	Professor, EMP	Examiner
R. ILLOUL	MC/A, ENP	Examiner

*ENP 2021*



REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



Ecole Nationale Polytechnique  
Département d'Automatique  
Laboratoire de Commande des Processus



Thèse de doctorat en sciences

Présentée par :

Yasmine SAIDI

Magistère en Automatique, EMP

En vue de l'obtention du titre de  
Docteur en Automatique

Option : Automatique des Systèmes Industriels

**INTITULÉ**

**Navigation Robuste d'un Robot Mobile basée sur la Logique Floue Type 2 en utilisant le modèle dynamique avec glissement et les incertitudes des paramètres**

Présentée et soutenue publiquement le 11/03/2021

Devant le jury composé de :

F. BOUDJEMA	Professeur, ESDAT	Président de jury
M. TADJINE	Professeur, ENP	Directeur de thèse
A. NEMRA	MC/A, EMP	Co-directeur
N. ACHOUR	Professeur, USTHB	Examinatrice
M. HAMERLAIN	Directeur de recherche, CDTA	Examinateur
M. HADDAD	Professeur, EMP	Examinateur
R. ILLOUL	MC/A, ENP	Examinateur

*ENP 2021*

**ملخص :** في هذه المقالة العلمية، تم تطوير نموذج ديناميكي كامل للملاحة المتينة لروبوت متحرك مع انزلاق العجلات. يتم تطبيق خوارزمية تحكم منطق مبهم من النوع 2 ومقارنتها مع وحدة تحكم مبهم من النوع 1 باستخدام روبوت متحرك بعجلات تفاضلية مع الأخذ بعين الاعتبار انزلاق العجلات. في هذه المقالة نقدم نمودجا ديناميكيا يربط بشكل صريح اضطرابات وانزلاقات الروبوت وكذا ارتيابات مختلف معايير وابعاد الروبوت. يتم التحقيق من صحة ومدى فعالية المراقبين على سلوكيات مختلفة للملاحة من أجل تقييم أدائها ومقارنتها. علاوة على ذلك، تم انشاء العديد من اختيارات متقدمة في المتانة (أخطاء النمذجة، ارتيابات متعلقة بمختلف المعطيات، فقدان كفاءة المحركات، وكذا انزلاق الأرضية) لمقارنة كفاءة كل وحدة تحكم. يتم التحقق من فعالية المراقب المبهم النوع 1 والنوع 2 من وحدات التحكم باستخدام محاكي روباتي V-REP في البيئة MATLAB® وفق عدة تجارب. تؤكد النتائج التي تم الحصول عليها بوضوح فعالية المراقب المبهم نوع IT2FLC مقارنة بالمراقب نوع 1 T1FLC خاصة عند الأخذ بعين الاعتبار انزلاق العجلات.

**الكلمات المفتاحية:** روبوت متحرك بعجلات تفاضلية، انزلاق العجلات، ارتيابات مختلف معايير، المراقب المبهم نوع 2، محاكي روباتي V-REP

**RÉSUMÉ :** Dans ce travail, un modèle dynamique complet pour robot mobile à glissement de roue est développé pour une navigation robuste. L'algorithme du contrôleur de logique floue de type 2 est mis en œuvre et comparé au contrôleur de flou de type I utilisant un robot mobile à roues d'entraînement différentiel (DDWMR) en présence de roues glissant du point de vue de la conception des commandes. Nous présentons un modèle dynamique qui relie explicitement les perturbations au patinage du véhicule et aux incertitudes sur les paramètres. Les deux contrôleurs sont validés sur différents comportements de navigation pour évaluer leurs performances. De plus, de nombreux tests de robustesse avancés (erreurs de modélisation, incertitudes de localisation, perte d'efficacité des moteurs, sol glissant) sont établis pour comparer l'efficacité de chaque contrôleur. Les contrôleurs de logique floue de type 1 et de type 2 à intervalle sont validés à l'aide du simulateur robotique V-REP sur l'environnement MATLAB® et les scénarios réels sont pris en compte. Les résultats obtenus montrent clairement les performances de l'IT2FLC par rapport au contrôleur T1FLC, en particulier lorsque l'on tient compte du patinage important et de l'incertitude des paramètres.

**MOTS CLÉS :** Robot mobile à roues à entraînement différentiel; Glissement de roue dynamique; Paramètres incertitudes; Analyse de robustesse; Contrôleur de logique floue de type 2; Plateforme d'expérimentation de robot virtuel.

**ABSTRACT :** In this work, a full dynamic model for mobile robot with wheel slip is developed for robust navigation. The type 2 fuzzy logic controller algorithm is implemented and compared to the type 1 fuzzy controller using a differential drive wheels mobile robot (DDWMR) in the presence of wheel slipping from the perspective of control design. We present a dynamic model that explicitly relates perturbations to the vehicle slipping and parameters uncertainties. Both controllers are validated on different behaviors of navigation to evaluate their performances. Furthermore, many advanced robustness tests (modeling errors, localisation uncertainties, loss of efficiency of motors, sliding ground) are established to compare the efficiency of each controller. Type 1 and interval type 2 fuzzy logic controllers are validated using V-REP robotic simulator on MATLAB environment and real scenario are considered. The obtained results show clearly the performances of the IT2FLC comparing to the T1FLC controller especially when significant wheel slip and parameters uncertainty are considered.

**KEYWORDS :** Differential Drive Wheeled Mobile Robot; Wheel Slip Dynamic; Parameters uncertainties; Robustness analysis; Type 2 fuzzy logic controller; Virtual Robot Experimentation Platform.

## DEDICATION

*In memory of those who have left us,*

*To my dear parents,*

*To my husband and my children,*

*To all my family,*

*To my friends.*

## **ACKNOWLEDGEMENT**

The research work was carried out within the Process Control Laboratory at the National Polytechnic School of Algiers in collaboration with the Intelligent Autonomous Vehicles Laboratory of the Polytechnic Military School.

First, I will begin by thanking Almighty God for giving me the strength and courage to keep pushing and getting to where I am today.

I would like to express my warm thanks to my thesis director Mr. Mohamed TADJINE Professor at ENP, for allowing me to start this field of research and for his confidence, his availability, his passion for research and the many discussions that we had during the whole period of preparation of this thesis.

I would also like to express my deep gratitude to my thesis co-supervisor, Mr. Abdelkrim NEMRA, for welcoming me to the Laboratory of Intelligent Autonomous Vehicles of the EMP and for everything he taught me. I thank him for all the efforts he made to ensure that this thesis was finalized, I thank him for having trusted me throughout the working period, for following me during the duration of the project, thank you for having known how to listen to me and encourage me.

I would like to thank the jury members for accepting participating in the review of my work; this is an honor for me. I am very sensitive to the interest shown in this work: The president, MF BOUDJEMA professor at the ESDAT, the examiners, Mr. HAMERLAIN director of research at the CDTA, Mrs. ACHOUR professor at the USTHB, Mr. HADDAD professor at EMP, Mr. ILLOUL MC/A at ENP for having accepted to judge this thesis.

My thoughts are also with the people who have participated from far or near to the development of this work, including Mr. HADDAD, Mr. HANK and Mrs. SBARGOUD.

This page would not be complete without thanking my friends: My dear sister Hafsa OSMANI from the Control & Command laboratory for her scientific discussions, her good humor, as well as for her fraternity and her encouragement during difficult times. I was happy to work with my dear Lydia HACHEMI with whom I shared pleasant moments. As well as with my dear Wassila MECHTI for her good humor, diplomacy and non-professional extras, not to mention Fahima Benyounes, whom I had the pleasure of working with during my thesis.

My heartfelt thoughts are with my main supporters, namely my family, especially my parents who have always supported me in what I undertook. Thank you and I will be forever grateful.

Finally, the best for last, I thank my husband who supported me daily through the most difficult times and without whom this work would not have been so successful. His advice, his energy, especially his patience and good humor have been invaluable to me throughout these six years. May this work be for you the testimony of my infinite gratitude for these years of sacrifice.



## TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ACRONYMS

LIST OF ABBREVIATIONS

GENERAL INTRODUCTION .....	22
Chapter I. RELATED WORKS .....	29
I.1. Introduction .....	29
I.2. Mobile Robotics and Navigation Methods .....	29
I.2.1. Current and emerging areas of application.....	29
I.2.2. Types of mobility systems on solid ground .....	33
I.2.3. Robot architectures.....	36
I.2.4. Perception.....	37
I.2.5. Localization.....	40
I.2.6. Mobile Robot Navigation.....	42
I.3. Wheeled mobile robot (WMR).....	47
I.3.1. Definition, current applications, and future potentials.....	47
I.3.2. Research on WMR - modeling, planning and control.....	48
I.4. Differential drive wheeled mobile robot (DDWMR) .....	50
I.5. Robustness analysis for mobile robots .....	51
I.5.1. Modeling error.....	52

I.5.2.	Localization error by odometry.....	53
I.5.3.	Motors efficiency .....	58
I.5.4.	Wheel slip.....	59
I.6.	Conclusion .....	61
Chapter II.	UNCERTAINTIES AND ERRORS MODELLING FOR A MOBILE ROBOT ....	63
II.1.	Introduction .....	63
II.2.	An uncertainty on the radius R of each wheel.....	64
II.2.1.	The inflation .....	64
II.2.2.	Overload and type of driving .....	65
II.2.3.	The condition of a vehicle.....	65
II.3.	Localization error .....	66
II.3.1.	Odometry.....	66
II.3.2.	Robot positioning calculation.....	68
II.3.3.	Modeling motion.....	70
II.4.	Motor efficiency .....	72
II.4.1.	Energy efficiency in mobile robots .....	72
II.4.2.	Energy consumption in mobile robots .....	74
II.5.	Wheel slip.....	76
II.5.1.	General information on pneumatic / pavement contact.....	76
II.5.2.	Longitudinal behavior .....	77

II.5.3.	Lateral behavior.....	78
II.5.4.	Dynamic coupled behavior: lateral - Longitudinal .....	79
II.5.5.	Pneumatic / pavement contact models .....	79
II.6.	Conclusion .....	82
Chapter III.	SYSTEM MODELING .....	84
III.1.	Introduction.....	84
III.2.	Coordinates system .....	84
III.2.1.	Kinematic constraints of the Differential-Drive Robot.....	86
III.2.2.	Pure rolling constraint .....	87
III.3.	DDWMR Kinematic model .....	89
III.4.	Dynamic modeling of the DDWMR without wheel slips.....	90
III.5.	Dynamic modeling of the DDWMR with wheel slips.....	96
III.6.	Driving wheel DC motor modeling .....	99
III.7.	Conclusion .....	101
Chapter IV.	FUZZY TYPE 1 AND INTERVAL TYPE 2 CONTROLLERS .....	103
IV.1.	Introduction.....	103
IV.2.	Overview of fuzzy logic controllers .....	103
IV.2.1.	What is a type-1 fuzzy set? .....	103
IV.2.2.	What is a type 1 fuzzy logic controller?.....	104
IV.2.3.	What is a type-2 fuzzy set? .....	107

IV.2.4.	What is a type 2 fuzzy logic controller?.....	109
IV.3.	Theory of interval type 2 fuzzy logic controller .....	110
IV.3.1.	Type-2 fuzzy sets .....	110
IV.3.2.	Interval Type-2 Fuzzy Sets .....	111
IV.3.3.	Interval Type-2 Fuzzy Logic Controller .....	112
IV.3.4.	Computations in an interval type 2 FLC .....	112
IV.4.	Waypoint navigation.....	115
IV.4.1.	Reactive Unit.....	117
IV.4.2.	Waypoint Knowledge Base .....	117
IV.4.3.	Deliberative Unit.....	117
IV.5.	Navigation using Type 1 Fuzzy controller.....	121
IV.5.1.	Type 1 fuzzy logic controller structure .....	122
IV.5.2.	Type 1 FLC Rule Base .....	123
IV.6.	Waypoint Navigation using interval type 2 Fuzzy logic controller .....	124
IV.7.	Conclusion .....	127
Chapter V.	RESULTS AND DISCUSSION .....	130
V.1.	Introduction .....	130
V.2.	Simulation using DDWMR kinematic model .....	131
V.2.1.	Scenario 1: Go to goal (single waypoint).....	131
V.2.2.	Scenario 2: Multiple waypoints.....	133

V.2.3. Scenario 3 : Robustness analysis.....	135
V.3. Results of dynamic model .....	140
V.3.1. Scenario 1: Go to goal (single waypoint).....	141
V.3.2. Scenario 2 : Multiple waypoints navigation.....	143
V.3.3. Scenario 3 : Robustness analysis.....	145
V.4. V-REP MODEL AND MATLAB/SIMULINK INTERFACING.....	159
V.4.1. Virtual robot experimentation platform (V-REP) .....	159
V.4.2. Matlab/Simulink and V-REP Interfacing.....	164
V.5. V-REP Validation.....	164
V.5.1. Scenario 1: Go to goal (single waypoint).....	165
V.5.2. Scenario 2 : Multiple waypoints navigation.....	167
V.5.3. Scenario 3 : Robustness analysis.....	168
V.6. Conclusion.....	182
GENERAL CONCLUSION .....	184
BIBLIOGRAPHY .....	187

## **LIST OF TABLES**

Table I-1 Summary of characteristics of localization techniques	41
Table III-1: Parameters of robot	101
Table IV-1: Rules of translational velocity (V) and angular velocity (W)	124
Table V-1: Computational time comparison for the two controllers for single waypoint	133
Table V-2: Computational time comparison for the two controllers for multiple waypoints	135
Table V-3: Computational time comparison for the two controllers.	143
Table V-4: Computational time comparison for the two controllers for multiple waypoints.	145
Table V-5: Summary on the results of fuzzy controllers	178
Table V-6: Robustness analysis and controller performances	181

## LIST OF FIGURES

Figure I-1 : Nuclear Robotic .....	30
Figure I-2 : Mars: NASA's OPPORTUNITY robot.....	30
Figure I-3: Sojourner rover .....	31
Figure I-4 : Aquanaut: the autonomous shape shifting submarine robot .....	31
Figure I-5 : Robotic Agriculture.....	32
Figure I-6 : Robotics Competition .....	32
Figure I-7 : Robotics Health.....	33
Figure I-8 : Self Driving and Drone Technologies Services .....	33
Figure I-9: Conventional wheels: (a) Fixed wheel. (b) Centered orientable wheel. (c) Off-centered orientable wheel. (d) Swedish wheel 908. (e) Swedish wheel 458 .....	34
Figure I-10 : Different type of wheeled mobile robots: (a) The KUKA Youbot Omnidirectional robot, (b) Pioneer 2DX: differential wheeled mobile robot, (c) Tricycle robot, (d) Robot type car .....	35
Figure I-11: Crawler mobile robot .....	35
Figure I-12: mobile robot with 6 legs.....	36
Figure I-13: Crawling Quadruped Robot .....	36
Figure I-14 Architecture of mobile robot .....	37
Figure I-15 : Different type of internal sensors : (a) 3-axis gyroscope and accelerometer - MPU-6050, (b) MMA7455L accelerometer, (c) LSM9DS0 9-DOF Accel / Mag / Gyro / Temp Card, (d) 4 axis accelerometer and tilt sensor, (e) magnetic compass, (f) angle inclinometer sensor.....	38
Figure I-16 : A set of external sensors: (a) Pololu IR Beacon Transceiver, (b) LIDAR Lite v3 Laser Telemeter, (c) An USB Optical Telemeter,.....	40
Figure I-17: Localization Methods.....	41
Figure I-18: Navigation strategies.....	43
Figure I-19: In Braitenberg vehicles, the speed of each of the robot's two motors depends on the values from two sensors that detect the light emitted by the goal.....	43
Figure I-20: Reactive navigation: Guidance .....	44
Figure I-21: Action associated with a place [27]. .....	45
Figure I-22: Topological navigation [27].....	46

Figure I-23: Metric navigation [27].	47
Figure I-24: (a) Spirit – NASA Solar System Exploration. (b) Cyclops Mk4D – Miniature Remotely Operated Vehicle (MROV).	48
Figure I-25: The relationship between components in the autonomous control application of WMR.	49
Figure I-26: Main components of differential drive mobile robot	55
Figure I-27: <i>Movement of a differential drive robot</i>	56
Figure I-28: Modeling of the odometry of the robot allowing to estimate the position of the robot from measurements of the incremental wheel encoders (CIR = instantaneous center of rotation)	57
Figure I-29: DC Motor	58
Figure II-1: Different faults and uncertainties applied in this work: (1) an uncertainty on the length L of the robot model, (2) uncertainty on the radius R of the left wheel, (3) localization errors, (4) loss of efficiency of motors, (5) wheel slip	63
Figure II-2: Rotary Incremental Optical Encoder	69
Figure II-3: Examples of the A pulse and B pulse. If the A pulse occurs before the B pulse, the shaft is turning clockwise, and if the B pulse occurs before the A pulse, the shaft is turning counter-clockwise.	69
Figure II-4: Schematic of Typical DC Motor	75
Figure II-5: Forces and moments on a wheel	77
Figure II-6: Lateral deformation of the tire during cornering action	78
Figure II-7 : Possible Categories of Tire Modeling Approaches [85].	81
Figure III-1: Differential Drive Wheeled Mobile Robot (DDWMR) [90].	86
Figure III-2: Pure rolling motion constraint.	87
Figure III-3: Rolling and lateral slip motions of wheel.	96
Figure III-4: Velocity control of DDWMR dynamic model with actuators	101
Figure IV-1: Representing Cold and Hot using (a) crisp sets, and (b) type-1 fuzzy sets [98]	104
Figure IV-2: General structure of a type1 FLC.	105
Figure IV-3: Type-2 fuzzy sets: (a) FOU and a primary membership and (b) a triangle secondary membership function [98].	108



Figure IV-4: (a) FOU with primary membership (dashed) at $x'$ , (b) two possible secondary membership functions (triangle in solid line and interval in dashed line) associated with $x'$ , and, (c) the resulting 3D type-2 fuzzy set. ....	108
Figure IV-5: Three type-2 fuzzy sets that are embedded in the FOU of Low .....	109
Figure IV-6: Overview of the architecture of a Type 2 FLC. ....	109
Figure IV-7: Interval type 2 fuzzy set .....	111
Figure IV-8: Interval type 2 FLC fuzzification and inference .....	113
Figure IV-9: Block diagram of the waypoint navigation system .....	116
Figure IV-10: Waypoints Navigator .....	119
Figure IV-11: Waypoint navigation diagram .....	120
Figure IV-12: Internal structure of Type1 Fuzzy Logic Controller .....	121
Figure IV-13: Input variable "Ro" .....	122
Figure IV-14: Input variable "Etheta" .....	122
Figure IV-15: Parameters for output: The translational velocity .....	123
Figure IV-16: Parameters for output: The angular velocity .....	123
Figure IV-17: Type 2 fuzzy translational velocity control.....	125
Figure IV-18: Type 2 fuzzy angular velocity control .....	126
Figure IV-19: Input variable "Ro" .....	127
Figure IV-20: Input variable "Etheta" .....	127
Figure V-1: Global scheme of control with kinematic model of DDWMR.....	131
Figure V-2: Results obtained for the scenario: Go to goal.....	132
Figure V-3: Results obtained for the scenario: Multiple waypoints navigation. ....	134
Figure V-4: Results obtained for the scenario: Multiple waypoints navigation $\Delta L = 30\%$ . ....	136
Figure V-5: Results obtained for the scenario: Multiple waypoints navigation( $\Delta R = 25\%$ ).....	137
Figure V-6: Results obtained for the scenario: Multiple waypoints (uncertainties of localization $\alpha = 0.003$ ).....	139
Figure V-7: Results obtained for the scenario: Multiple waypoints navigation (uncertainties of localization $\alpha = 0.009$ ).....	140
Figure V-8: Global scheme of control with dynamic model of DDWMR.....	141
Figure V-9: Results obtained for the scenario: Go to goal.....	142
Figure V-10: Results obtained for the scenario: Multiple waypoints navigation. ....	145

Figure V-11: Results obtained for the scenario: Multiple waypoints navigation  $\Delta L = 30\%$ . .... 147

Figure V-12: Results obtained for the scenario: Multiple waypoints navigation( $\Delta R = 25\%$ )... 148

Figure V-13: Results obtained for the scenario: Multiple waypoints navigation( $\Delta R = 50\%$ )... 150

Figure V-14: Results obtained for the scenario: Multiple waypoints navigation( $\Delta RL = 60\%$ ). 152

Figure V-15: Results obtained for the scenario: Multiple waypoints navigation (uncertainties of localization  $\alpha = 0.003$ ). ..... 154

Figure V-16: Results obtained for the scenario: Multiple waypoints navigation (loss of efficiency of motors  $\alpha L = 0.6, \alpha R = 0.1$ ). ..... 155

Figure V-17: Results obtained for the scenario: Multiple waypoints navigation (loss of efficiency of motors  $\alpha L = 0.05, \alpha R = 0.005$ ). ..... 157

Figure V-18: Results obtained with DDWMR navigation with slip dynamics ( $\alpha L = 3, \alpha R = 1$ ). ..... 158

Figure V-19: Results obtained with DDWMR navigation with slip dynamics( $\alpha L = 5, \alpha R = 20$ ). ..... 159

Figure V-20: The built-in V-REP robot models..... 162

Figure V-21: The remote API communication modes: (a) blocking function call, (b) non-blocking function call [22]. ..... 163

Figure V-22: General simulator operation diagram ..... 164

Figure V-23: PIONEER 3DX in V-REP environment with single waypoint. .... 165

Figure V-24: V-REP: Results obtained for the scenario: Go to goal. .... 166

Figure V-25: PIONEER 3DX in V-REP environment with multiple waypoints navigation ..... 167

Figure V-26: V-REP: Results obtained for the scenario: Multiple waypoints navigation ..... 168

Figure V-27: V-REP: Results obtained for the scenario: Multiple waypoints navigation  $\Delta L = 30\%$ ..... 169

Figure V-28: V-REP: Results obtained for the scenario: Multiple waypoints navigation ( $\Delta R = 15\%$ ). ..... 171

Figure V-29: V-REP: Results obtained for the scenario: Multiple waypoints navigation ( $\Delta R = 25\%$ ). ..... 172

Figure V-30: V-REP: Results obtained for the scenario: Multiple waypoints navigation (uncertainties of localization  $\alpha = 0.1$ )..... 174

Figure V-31: V-REP: Results obtained for the scenario: Multiple waypoints navigation  
(uncertainties of localization  $\alpha = 0.3$ ).....175

Figure V-32: V-REP: Results obtained for the scenario: Multiple waypoints navigation (loss of  
efficiency of motors  $\alpha_R = 1, \alpha_L = 0.5$ ). .....177

Figure V-33: V-REP: Results obtained for the scenario: Multiple waypoints navigation (loss of  
efficiency of motors  $\alpha_R = 0.75, \alpha_L = 0.1$ ).....178

## LIST OF ACRONYMS

Symbols	Designation
$R_l$ and $R_r$	Left and right wheel radius
$\Delta R_l$ and $\Delta R_r$	Left and right wheel radius deviation
$(x; y; \theta)$	The robot position and orientation
$(\Delta x; \Delta y; \Delta \theta)$	Path traveled in the last sampling interval
$\Delta s_r$ and $\Delta s_l$	Traveled distances for right and left wheel respectively
$b$	Distance between the two wheels of differential drive robot
$F_x$	The longitudinal force developed between the road and the wheel in the x direction
$F_y$	The lateral force developed between the road and the wheel in the y direction
$F_z$	The normal force on the wheel in z direction
$M_x$	The payment moment around the x-axis
$M_y$	The moment of rolling resistance around the y-axis
$M_z$	The moment of self-alignment around the z-axis
$v_r$	The linear velocity of translation of the contact point between the right wheel and the road
$v_l$	The linear velocity of translation of the contact point between the left wheel and the road
$\omega$	The angular velocity of the wheel rotation
$r$	The rolling radius when a wheel rolls freely
$\xi_r, \xi_l$ and $\delta_r, \delta_l$	The longitudinal and lateral slip displacement for the right and left wheel respectively
$(x, y, z)$	The absolute positions
$(\varphi, \theta, \psi)$	The absolute orientations with the Euler angles
$R$	The radius of each wheel
$d$	The distance between the center of mass (Point $D$ ) and mid-point of the axis center of driving wheels (Point $A$ )
$L$	Each wheel distance to point a and it is the robot length
$\Delta L$	Deviation along the length of the robot
$\dot{\varphi}_R$ and $\dot{\varphi}_L$	The right and left wheel angular velocities respectively
$C$	The distance between point $A$ and instantaneous center of curvature ( $ICC$ )
$m_c$	The DDWMR mass without the driving wheels and actuators (DC Motors)
$m_w$	The mass of each driving wheel (with actuator)
$I_c$	The moment of inertia of the DDWMR following the vertical axis through the center of mass

$I_w$	The moment of inertia of each driving wheel (with actuator) around the wheel axis
$I_m$	The moment of inertia of each driving wheel with a motor around the wheel diameter
$i_a$	The armature current
$(R_a, L_a)$	The resistance and inductance of the armature winding respectively
$m_c$	The DDWMR mass without the driving wheels and actuators (dc motors)
$e_a$	The back fem
$\omega_m$	The rotor angular speed
$\tau_m$	The motor torque,
$(K_t, K_b)$	The torque constant and back fem constant respectively
$N$	The gear ratio
$J$	Moment of inertia of the motor shaft
$\tau$	The output torque applied to the wheel

## **LIST OF ABBREVIATIONS**

**FLC:** Fuzzy Logic Controller

**IT2FLC:** Interval Type 2 Fuzzy Logic Controller

**T1FLC:** Type 1 Fuzzy Logic Controller

**WMRs:** Wheeled Mobile Robots

**DDWMR:** Differential Drive Wheeled Mobile Robot

**V-REP:** Virtual Robot Environment Platform

**FLS:** Fuzzy Logic System

**GPS:** Global Positioning System

# GENERAL INTRODUCTION

## GENERAL INTRODUCTION

Robot or robotics is a fascinating term for human being that attracts lots of attention either in the form of hobby robots or in the form of industrial or military robots. Today, people life is highly attracted by various types of robots. The interest has been boosted by the fiction writers and the moviemakers who showcase innovative and futuristic ideas. Fiction has always fascinated designers/innovators in the field of science and engineering. Due to this fascination, a large community of scientists and researchers focused on design, modeling and control of robots for variety of applications in recent years.

Historically, robots can be classified into two categories: stationary robots and non-stationary or mobile robots. Stationary robots are manipulator robots that can move in a fixed frame with a limited work envelope and use to bring the end effector to the desired position with the desired orientation. Unlike stationary robots, mobile robots can move around their environment and are not fixed to a single physical location. Mobile robots can be air, submarine or land. Among the terrestrial robots, we find the mobile robots with wheels, which is the object of our work. Wheeled mobile robots (WMRs) have many applications in the real world for military, industrial and commercial purposes as a means of logistics [1], agriculture or passenger transport [2], inspection, surveillance and specialized / non-specialized operations because of their efficiency and flexibility. Wheeled mobile robots are also useful for performing so-called 4D (Difficult, Dull, Dangerous and Dirty) tasks such as handling radioactive materials, decontaminating nuclear reactors, demining, inspecting tunnels, etc. In addition, WMR is an excellent platform for testing a variety of educational and research applications.

There are researches covering this topic since the nineteen-seventies [3], so it is a mature field with lots of published algorithms and available tools. However, real applications in real conditions remain to be a challenge.

In general, the autonomous navigation of WMRs is found to be a crucial area of research among the various problems of this category of robots.

Indeed, the various navigation problems deal with WMR autonomous navigation under different assumptions (known model, precise location, non-slip ground) whereas an important aspect for



each realistic navigation task is to take into consideration real situations and scenarios (parameters uncertainties, modeling errors). Furthermore, due to their dynamic constraints, mobile robots suffer from wheels slipping leading to significant position drift (inaccurate localization). The actuators can be affected by the interaction robot/environment, which lead to positioning errors.

In fact, in this work, robust navigation algorithms based on type 1 and type 2 fuzzy logic controllers with deep robustness analysis are investigated in this thesis. We have implemented some navigation tasks using kinematic and dynamic model of the pioneer 3AT robot available at the Autonomous Intelligent Vehicle Laboratory (LVAI) of the Polytechnic Military School.

Many advanced robustness tests (modeling errors, localization uncertainties, loss of efficiency of motors, sliding ground) are established to compare the efficiency of each controller. Many scenarios are considered, moving to a goal, moving to multiple waypoints with and without parameters uncertainties and track trajectory. Type 1 and interval type 2 fuzzy logic controllers are validated using V-REP robotic simulator on MATLAB® environment and real scenario are considered.

The obtained results show clearly the performances of the interval type 2 fuzzy logic controller, comparing to the fuzzy type 1 controller especially when significant uncertainties are considered.

## **Problem Statement**

Wheeled mobile robots (WMRs) control problems have been intensively studied in recent years; and apparently, most problems have been properly addressed [4] [5].

However, most existing works assume that WMRs satisfy the non-slipping and non-skidding conditions. In reality, these assumptions cannot be met due to tire deformation and other reasons; hence, stability and control performance of these existing controllers are not guaranteed in real navigation.

Thus, the autonomous navigation of mobile robotics remains an essential field of research; in particular, when real scenarios are considered (parameter uncertainties, modeling errors, etc.) taking into account several sources of errors. These errors can be summarized as follows:

- Sensor measurements are usually noisy due to the instruments. Generally, a robot is equipped with internal sensors like encoder and inertial sensors and external sensors as laser sensors, infrared sensors and cameras. Measured data from these sensors are generally uncertain and noisy [6] [7];
- Assembly errors that include linear and angular errors produced during the assembly of the various robot mechanical components; Interaction robot/environment can affect the actuators, which leads to positioning errors [6].

In order to overcome those uncertainties and to develop a robust, flexible and on-line controller for navigation, type 1 fuzzy logic controller has been used in [8] [9] [10].

However, when using type 1 fuzzy logic controller for mobile robot navigation all the aforementioned uncertainties are multiplied over fuzzification, inference and defuzzification [11] [12]. Those uncertainties can degrade the performances of the mobile robot navigation controller.

### **Motivations**

First point, the concept of uncertainty is posed in almost any complex system including wheeled mobile robots as an outstanding instance of dynamical robotics systems. As suggested by the name, uncertainty, is some missing information that is beyond the knowledge of human thus we may tend to handle it properly to minimize the side-effects through the control process.

Type 2 fuzzy logic has shown its superiority over traditional fuzzy logic when dealing with uncertainty. Type 2 fuzzy logic controllers are however newer and more promising approaches that have been recently applied to various fields due to their significant contribution especially when noise (as an important instance of uncertainty) emerges. During the design of Type 1 fuzzy logic systems, we presume that we are almost certain about the fuzzy membership functions, which is not true in many cases. Thus, type 2 FLS as a more realistic approach dealing with practical applications might have a lot to offer. Type 2 fuzzy logic takes into account a higher level of uncertainty, in other words, the membership grade for a type 2 fuzzy variable is no longer a crisp number but rather is itself a type 1 linguistic term.

Type 2 fuzzy logic controller has been used by researchers to overcome some of type 1 fuzzy logic limitations as shown in [13] [14], then, many researchers have explored the type 2 fuzzy logic controllers in various applications [15] [16] [17].

In fact, type 2 fuzzy sets were initially introduced by Zadeh [18]. Hence, a simplified version of general type 2 set called interval type 2 fuzzy set is widely used [19] [20]. This kind of set has membership grades that are crisp interval sets bounded in  $[0,1]$ . In this case, the uncertainty is represented as a 2D bounded region that is called the Footprint of Uncertainty. Various researchers have explored the advantages of interval type 2 fuzzy sets [21] [22]. In mobile robotics, some researchers have explored mobile robots control using interval type 2 fuzzy logic [21] [23] [24]. In [25], Hagrais presented an interval type 2 fuzzy logic controller to command a robot in indoor and outdoor unstructured environment. A mobile robot was tested under different sources of non-systematic errors. The results showed that type 2 fuzzy logic outperforms its type 1 counterpart. In [22], an interval type 2 fuzzy logic was proposed for the control of a robot tracking a mobile object in the context of robot soccer games. In this game, the robot has to track a ball.

In second point, in recent years, many researchers, all over the world, have paid attention to solving the tracking problems of WMRs by employing various techniques with the assumption “pure rolling without slip” being always satisfied.

However, in practice, the assumption “pure rolling without slip” is often violated due to various factors such as slippery floor, external forces, and so on. The wheel slip is one of the reasons making the tracking performance of non-holonomic WMRs reduce considerably. Therefore, if one wants the tracking performance of the WMRs to be improved in such context, then control methods having the ability to overcome the undesired effects of the wheel slips must be taken into account.

In this context, we propose: First, the full dynamic model with wheel slip consideration. Second, a robust solution based on the type 2 fuzzy logic controller. This latter is implemented and compared to the type 1 fuzzy logic controller. The performances of the proposed controllers are evaluated by simulations and validated using V-REP robotic simulator on MATLAB® environment and real scenario are considered.

## **Contributions**

The main contributions of this thesis are: firstly, the use of a dynamic model of the WMR taking into account wheel slip. Second, the implementation of a fuzzy logic controller type 2 to solve several navigation problems in different scenarios. Third, the deep robustness and evaluation of the controllers face modeling errors, uncertainties localization (odometry), loss of efficiency of motors and wheel slip. Finally, both controllers are validated and compared using the V-REP robot simulator on MATLAB®.

## **Structure of the thesis**

The remainder of this thesis is organized as follows: The first Chapter is divided into three distinct parts: The first is a reminder of the generalities of mobile robotics and navigation methods. The second is a bibliographic search of previous DDWMR and robust control work. The third is a literature review on robustness analysis in mobile robotics. In Chapter II, uncertainties and errors modelling for a mobile robot are explained by the study of many advanced robustness tests (modeling errors, location uncertainties, loss of engine efficiency, slippery ground) are carried out. In Chapter III, different models of the DDWMR are presented: kinematic model, dynamic model with and without wheel slip then the DC motor model. The theory of type 1 fuzzy logic controller and interval type 2 fuzzy logic controller are presented in Chapter IV with the development of a multi-waypoints navigation algorithm using both controllers. Simulation results and robustness evaluation are presented in Chapter V using Kinematic, dynamic as well as the V-REP robotic simulator on the MATLAB® environment. Finally, conclusion and future works are presented.

## **Publications list**

### Conferences Paper

1. Y.Saidi, M.Tadjine and A.Nemra “Robust Waypoints Navigation Using Fuzzy Type 2 Controller”, 1st Algerian Multi-Conference on Computer, Electrical and Electronic Engineering 2017, April 24-27, 2017 in USTHB, Algiers.
2. Y.Saidi, M.Tadjine and A.Nemra, “Robust Waypoints Navigation Using Fuzzy Type 2 Controller”, International Conference on Advanced Electrical Engineering 2019, November 19-21, "National Library, Elhamma" in Algiers, Algeria.

## Journal Paper

- Y.Saidi, A.Nemra and M.Tadjine. "Robust mobile robot navigation using fuzzy type 2 with wheel slip dynamic modeling and parameters uncertainties." *International Journal of Modelling and Simulation* vol. 40, no. 6 (2020): pp. 397-420.

**CHAPTER I :**  
**RELATED WORKS**

## **Chapter I. RELATED WORKS**

### **I.1. Introduction**

This dissertation covers a broad range of research areas such as modeling of DDWMMR robot, wheel slip phenomenon, DDWMMR robust control as well as robustness analysis. Thus, it requires broad literature survey encompassing multidisciplinary areas.

In this chapter, we examine a relevant state-of-the-art for mobile robot and navigation techniques as well as more generally the research on the WMR and precisely the DDWMMR. Finally, a literature research on the robustness analysis for mobile robot navigation will be investigated.

### **I.2. Mobile Robotics and Navigation Methods**

In recent years, a plethora of research has been carried out on the control problem of the autonomous mobile robotic systems. This is mainly due to the growing application of these systems both in industrial and service environments. Some typical applications of such systems are for instance order-pick robots in automated warehouses, post-delivery robots in office buildings and deep sea exploration robots. Different kinds of robots can be used in these applications [26]. In rough terrains, walking robots are usually preferred. On smoother surfaces, wheeled mobile robots have the advantage because they are much faster and more agile. Other kinds of systems such as unmanned aerial vehicles or unmanned under water vehicles are used when we need to maneuver in three-dimensional spaces. Wheeled mobile robots are the type of mobile robotic systems that we are considering in this thesis because they are most widely used among the class of mobile robots. This is due to their fast maneuvering, simple controllers and energy saving characteristics.

#### **I.2.1. Current and emerging areas of application**

The mobile robots are largely used in many areas (domestic, industrial, military ...). In the last decade the use of mobile robots in strategic domains (Nuclear, Spatial, Submarine...) has been increased significantly, firstly, to save the human life and secondly to improve the required performances. Whether it is a civilian or a military robot, there are now robots capable of amazing feats in many sectors:

- **Nuclear robotic:** nuclear robots can be used for maintenance, dismantling of facilities, decontamination, inspection and also for responding to accidents, etc....(Figure I-1)



**Figure I-1 : Nuclear Robotic**

· **Spatial robotic:** space robots were used for lens placement on the Hubble telescope, Martian exploration (Figure I-2) as well as the small robot Sojourner (a Mars rover robot that landed on July 4, 1997 in the region of 'Ares Vallis planet, and was part of the Mars Pathfinder mission as shown in Figure I-3, etc....



**Figure I-2 : Mars: NASA's OPPORTUNITY robot.**





**Figure I-3: Sojourner rover**

· **Submarine robotic:** these are robots intended for the inspection and repair of offshore structures, TITANIC exploration, seabed mapping, outlets (water, sewers), laying of telecommunications and electrical cables, etc ... (Figure I-4)



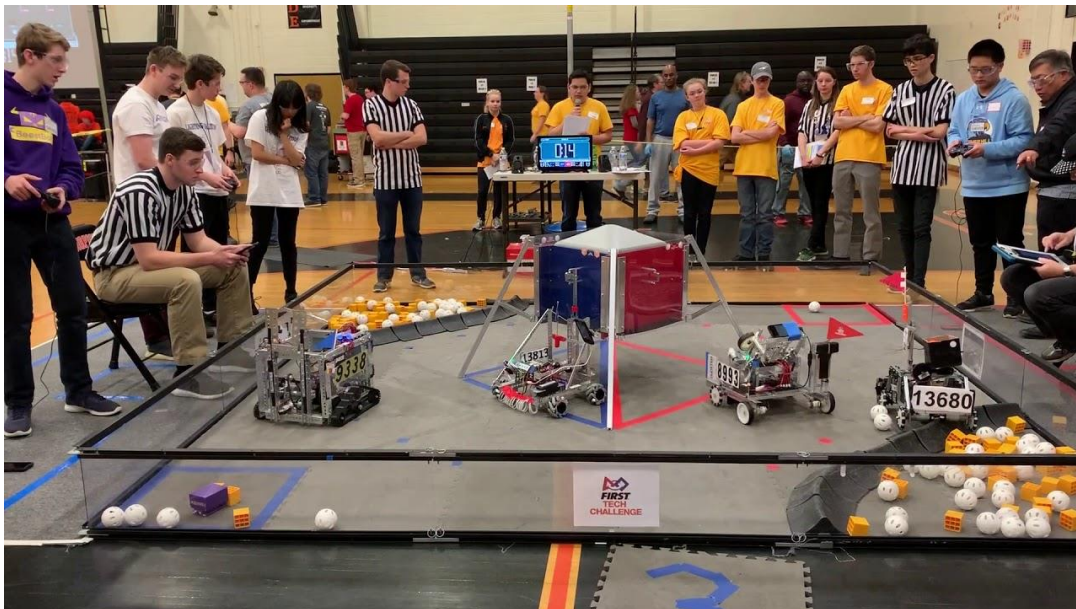
**Figure I-4 : Aquanaut: the autonomous shape shifting submarine robot**

· **Agriculture robot:** fruit picking robots, planters, robotic weeding, guiding agricultural vehicles, automatic milking of dairy cows, etc. (Figure I-5).



**Figure I-5 : Robotic Agriculture**

· **Fun activities:** competition robots ("Robocup") that mobilize many researchers and students over the world, etc. (Figure I-6).



**Figure I-6 : Robotics Competition**

· **Health:** robot for intelligent wheelchairs, cardiac, ocular, brain surgery, training applications in which we associate virtual reality and assisted gestures robotics (Figure I-7).



**Figure I-7 : Robotics Health**

· **Automatic Vehicles:** Assistance with driving (using largely the results of work on mobile robots in location, obstacles avoidance, motion planning), small flying vehicles or drones (airships, airplanes, helicopters) for military applications, automatic mapping, high-voltage line inspection or mountain accident detection [27] (Figure I-8).



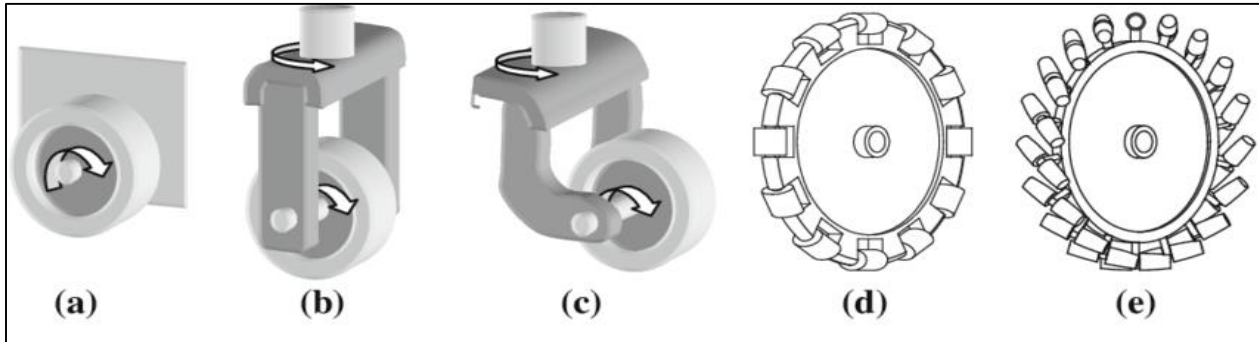
**Figure I-8 : Self Driving and Drone Technologies Services**

### **I.2.2. Types of mobility systems on solid ground**

System mobility is the combination of wheels choice of and their arrangement, which gives a robot its own mode of locomotion. There are mainly three types of wheels for mobile robots (Figure I-9):

- fixed wheels whose axis of rotation, for fixed direction, passes through the center of wheels (Figure I-9.a);
- centered orientable wheels for which the axis of orientation passes through the center of the wheel (figure I-9.b)

- Off-center steerable wheels, often called idler wheels, for which the axis of orientation does not pass through the center of the wheel (figure I-9.c).
- Swedish wheels, wheels with several rolling directions, (figure I-9.d and e).

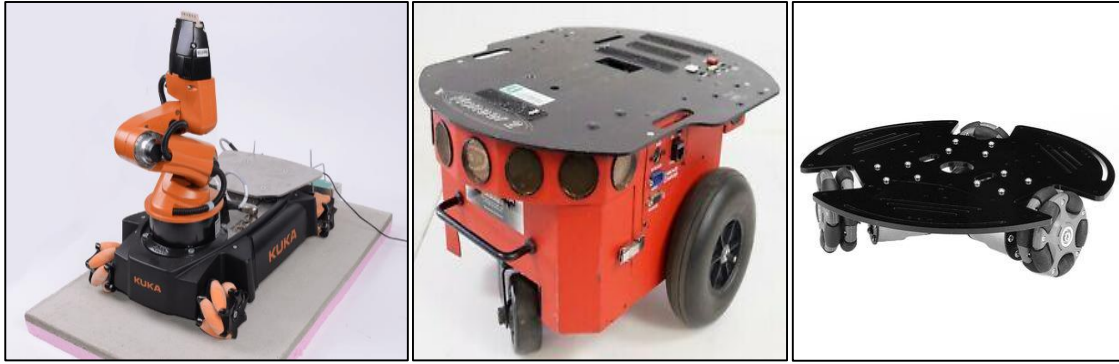


**Figure I-9: Conventional wheels: (a) Fixed wheel. (b) Centered orientable wheel. (c) Off-centered orientable wheel. (d) Swedish wheel 908. (e) Swedish wheel 458**

Obviously, for a given set of wheels, wrong arrangement does not lead to a viable solution. Moreover, it can limit the mobility of the robot or cause possible blockages. For example, a robot with two non-parallel fixed wheels could not go in a straight line!

Mobile robots can be classified into:

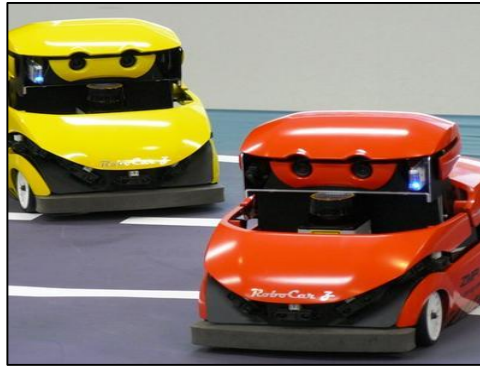
- **Wheeled Mobiles:** They are the most popular mobile robots for reasons of simplicity of design and control. Except in the case of a particular mechanical structure, the displacement is made only according to the movement of the wheels, we find:
  - Omni-directional wheeled robots (Figure I-10.a).
  - Robots with differential wheels (Figure I-10.b).
  - Robots type "tricycle": equipped with a fixed rear axle provided with two non-steerable wheels and a steerable center wheel (Figure I-10.c).
  - Robots "car" type: non-steerable rear axle with two non-rotating wheels with free rotation and two front wheels centering and steerable (Figure I-10.d).



(a)

(b)

(c)



(d)

**Figure I-10 : Different type of wheeled mobile robots: (a) The KUKA Youbot Omnidirectional robot, (b) Pioneer 2DX: differential wheeled mobile robot, (c) Tricycle robot, (d) Robot type car**

➤ **Crawler Mobiles:** These are robots that have the best grip on the ground and are used when the ground is disturbed, mainly outdoors. The control is carried out by imposing a speed difference to the right and left tracks (Figure I-11).



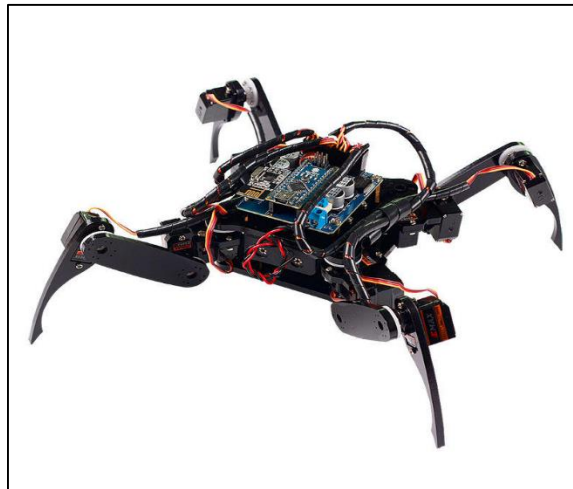
**Figure I-11: Crawler mobile robot**

➤ **Mobile hands:** most of the time, this kind of robots is used on ground with big difference of amplitude where it is necessary to choose points of support. The design and control of such mechanisms is complex (Figure I-12).



**Figure I-12: mobile robot with 6 legs**

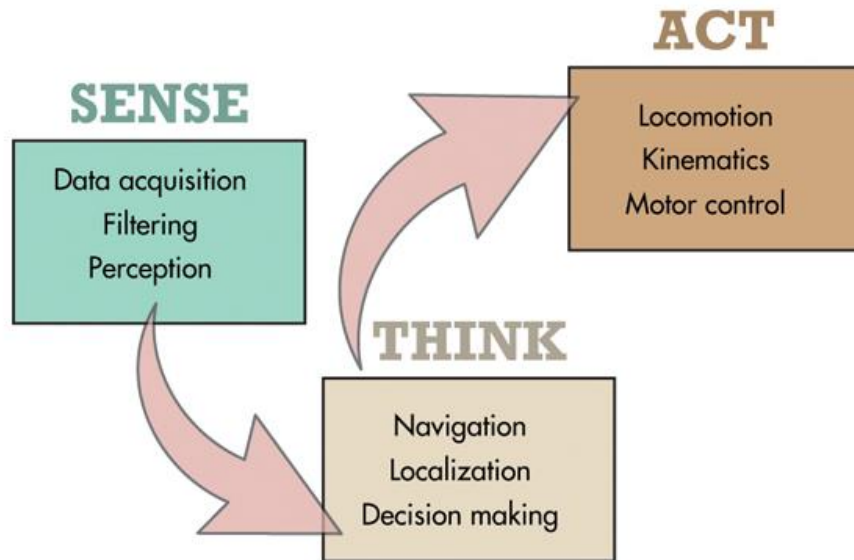
➤ **Mobile moving by crawling:** These robots are used for the progression in galleries or pipes (Figure I-13).



**Figure I-13: Crawling Quadruped Robot**

### **I.2.3. Robot architectures**

Mobile Robot architecture is a mechanical device that performs tasks according to a program that has been assigned to it. Generally, robots have an architecture that can be schematized in most cases as follows (Figure I-14):



**Figure I-14 Architecture of mobile robot**

#### **I.2.4. Perception**

The following points can summarize the perception of the environment by mobile robots:

- Defined by the set of measurement acquisition and information processing functions, perception allows environment analysis and / or modeling, in order to support the decision-making and control generation.
- The mobile robot uses, to perceive their environment, environmental sensors, also called sensors that can be used alone or in combination.

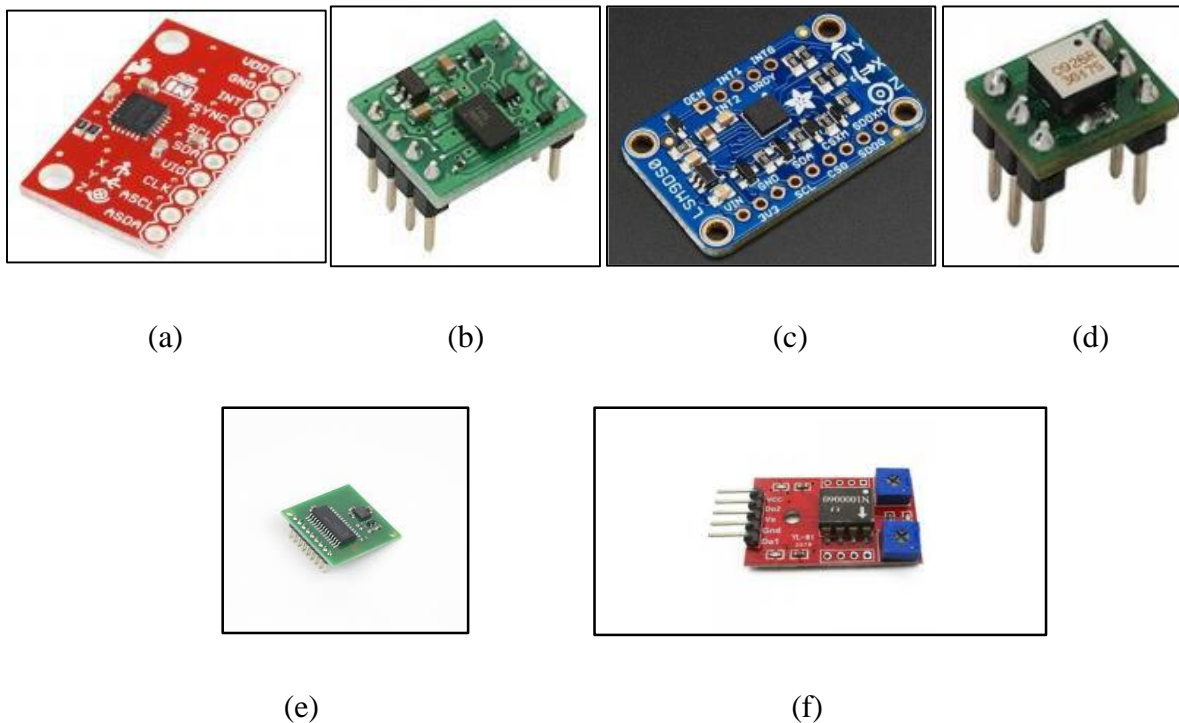
##### **I.2.4.1. Classification of sensors in robotics**

- Internal sensors (proprioceptive)

They are sensors capable of measuring the own (internal) information of the robot. Thanks to these sensors, the robot can determine the state of some of its components (kinematic, batteries, actuators, etc.). It can deduce its state in the environment [27].

For a mobile robot, internal sensors are the sensors of heading, speed, altitude by means of gyrocompasses, accelerometers, depth gauge, odometer, etc.

- **Accelerometers:** allow the measurement of the acceleration which, integrated twice, makes it possible to estimate the linear displacement of the vehicle (Figure I-15.b).
- **Gyroscopes:** these devices allow measuring the orientation rate. There are mechanical gyroscopes that use the inertial properties of matter and laser gyroscopes that use the properties of coherent light (laser gyro) (Figure I-15.a).
- **Gyrometers:** is an angular velocity sensor. The integration of this rotation speed measurement makes it possible to obtain an estimate of the heading angle of a vehicle (Figure I-15.c).
- **Magnetic Compasses:** they allow the absolute measurement of the heading relative to the direction of the geographical north and are sensitive to the surrounding magnetic masses. It is impossible to use them inside a building (Figure I-15.e).
- **Inclinometers:** they measure angles of attitude (pitch and roll) on the principle of pendulum accelerometers and are sensitive to Earth's gravity but also to any external acceleration applied to them (movements of the machine, vibrations, shocks) (Figure I-15.f).



**Figure I-15 : Different type of internal sensors : (a) 3-axis gyroscope and accelerometer - MPU-6050, (b) MMA7455L accelerometer, (c) LSM9DS0 9-DOF Accel / Mag / Gyro / Temp Card, (d) 4 axis accelerometer and tilt sensor, (e) magnetic compass, (f) angle inclinometer sensor**



- External sensors (exteroceptive)

These sensors are capable of measuring information from the environment (external measurements). They allow the robot to model and map its external environment.

They deliver information about the environment (recognition, model) or interactions between the robot and its environment (position, force, and obstacle) [27].

We can find as an exteroceptive sensor:

#### ➤ **GPS (Global Positioning System)**

Acronym standing for Global Positioning System. Satellite geolocation system. The network of 24 satellites (plus 4 satellites in reserve) currently in operation, developed by the American army, is made available to civilians. It allows determining the geographic coordinates of any point located on the surface of the globe. Its precision can reach 1 meter. The GPS is used in association with a map to find your way around and position yourself: hiking, sailing, trekking...

#### **How GPS works**

The operating principle of GPS is based on measuring the distance of a receiver from several satellites (the satellites are distributed in such a way that 4 to 8 of them are always visible). Each satellite emits a signal, picked up on Earth by the receiver, thus making it possible to measure very precisely the distance between the transmitter and the receiver thanks to the travel time.

With the reception of signals from four satellites (three to obtain the point of intersection of the three spheres, a fourth for time synchronization), the mobile receiver is able to calculate its geographical position by triangulation.

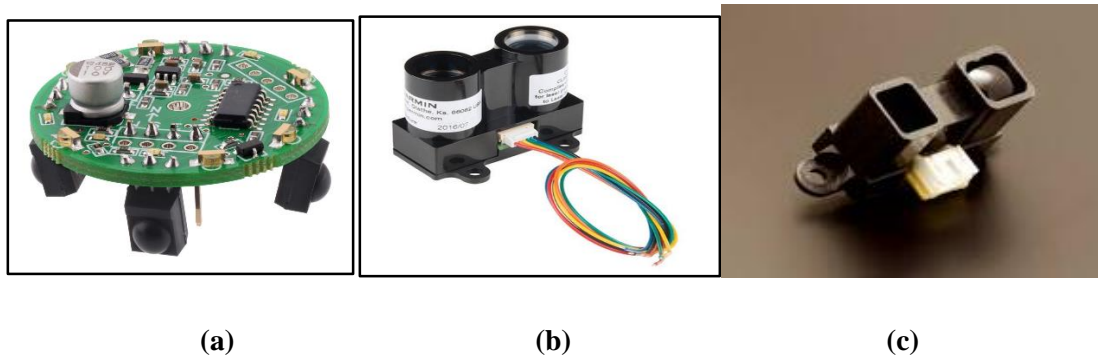
#### **GPS accuracy**

Degraded voluntarily by the American army until 2000, the accuracy of GPS is today of the order of a meter, but depends on the equipment used and the number of satellites in visibility.

➤ **Artificial Beacon Detectors:** This involves using fixed elements whose relative positions are known and using them passively or actively, communicating with the robot or not, in order to determine the absolute position of the robot. The most common artificial beacons are passive optical retro-reflectors that have the property of returning light in the direction of incidence. The reading of these beacons is generally done using a laser beam transceiver (Figure I-16.a).

➤ **Telemeters:** Their principle consists of measuring the time taken by an electromagnetic or acoustic wave (Ultrasonic, Infrared and Visible) to measure the distance which separates the sensor from the target on which this wave is reflected (Figure I-16.b).

➤ **Optical telemeters:** The optical waves used in telemetry are produced by a laser diode operating in continuous or pulsed mode to emit a generally monochromatic light beam in the red ( $\lambda = 670$  nm), the infrared or the near infrared ( $780 \text{ nm} < \lambda < 850 \text{ nm}$ ). The spatial coherence of the laser light makes it possible to obtain beams of very low divergence and high luminance. In robotics, we distinguish between impulse telemetry and phase difference telemetry. The acquisition of 2D or 3D distance images requires the use of a mechanical system (rotating mirror) which allows the laser beam to perform a plane or spatial scan (in site and in azimuth) of the scene (Figure I-16.c).

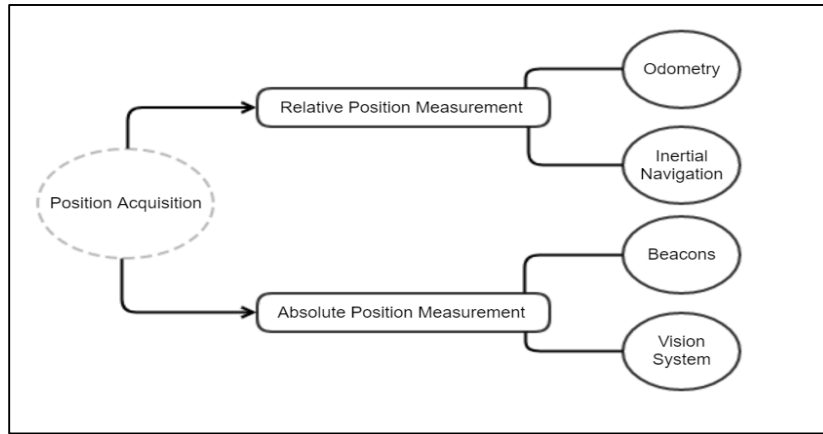


**Figure I-16 : A set of external sensors: (a) Pololu IR Beacon Transceiver, (b) LIDAR Lite v3 Laser Telemeter, (c) An USB Optical Telemeter,**

### I.2.5. Localization

Localization is a set of sensors and techniques allowing the vehicle to navigate autonomously or semi-autonomously in its environment [27]. Localization consists in finding or tracking the current position of a robot with known maps or dynamic environment using inaccurate, noisy and sensitive sensors. Generally, you cannot measure the position directly. We must rather estimate it. We find the relative location and the absolute location as shown in the figure I-17.

- **Relative localization:** allows the vehicle to localize itself in the local frame "*dead reckoning*" using only the measurements of its own movements provided by its proprioceptive sensors.
- **Absolute localization:** uses measurements of exteroceptive sensors to estimate the position of the vehicle in a global reference.



**Figure I-17: Localization Methods**

The table I-1 is a summary of the characteristics of the localization techniques

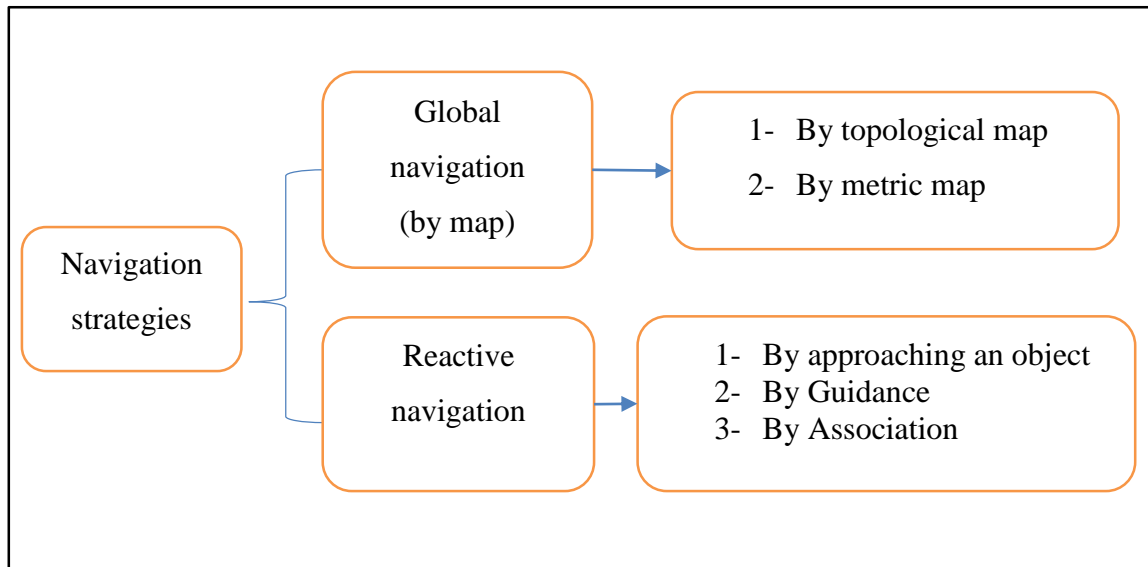
**Table I-1 Summary of characteristics of localization techniques**

<b>Technique</b>	<b>Localization type</b>	<b>Area of use</b>	<b>Implementation</b>	<b>Performance</b>	<b>Span</b>
Odometry on drive wheels	Relative: heading and position measurement.	Interior and outside on flat terrain	Simple	Strong drift Errors on uneven ground	unlimited
Fiber optic gyroscope	Relative: course measurement	Indifferent	Simple	Slight drift a few ° / h	unlimited
inclinometers	Absolute: Pitch and roll measurement	Smooth floor Uniform movement	Simple	Good if no parasitic accelerations Error <0.1°	0° to 90°
Magnetic compass	Absolute: heading measurement	Outside. Medium free of magnetic disturbances	Simple	Error: a few tenths of a degree	unlimited
Strap-type inertial attitude system	Relative: measurement of the three attitude angles	Indifferent	Simple	Accuracy: a few tenths of a degree Low drift: a few degrees / h	Unlimited

GPS	Absolute : position measurement (x, y, z)	Outside	Simple	Accuracy: some cm to some meters in differential mode	unlimited
2D localization on artificial optical beacons	2D absolute: position and heading measurement	inside	Installation and modeling of the tag field	Error 1 to 5 cm in position and <1 ° in orientation	<20 m
3D localization on artificial optical beacons	Absolute 3D: measure of position (x, y, z) and attitude ( $\phi, \psi, \theta$ )	Outside	Installation and modeling of the tag field	Accuracy: depending on the distance of the beacons	100m to 300m in distance and 10° to 30° in inclination
Map matching	Absolute: position and course measurement	Interior Relatively structured environmen t	Prior acquisition of a reference card	Error: <20 cm in position and <5 ° in orientation	Function of range finder
Dynamic vision	Absolute or relative 3D: measurement to within a scale factor	Indifferent	Complex algorithms. High computing time		
stereovision	Absolute or relative 3D	Indifferent	Complex algorithms. High computing time		

### I.2.6. Mobile Robot Navigation

For several decades, various researchers and scientists have provided numerous methodologies on navigational approaches. Navigation strategies that allow a mobile robot to move to a goal are extremely diverse. Figure I-18; illustrate the navigation strategies classification given by a Trullier and Meyer [28], which has the advantage of distinguishing between strategies with and without internal models. This classification has five categories, from the simplest to the most complex:

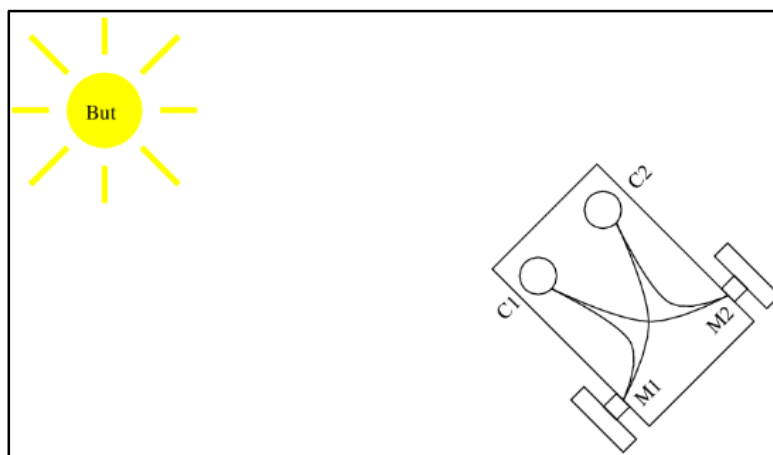


**Figure I-18: Navigation strategies**

### **Reactive navigation**

#### **A. Approaching an object**

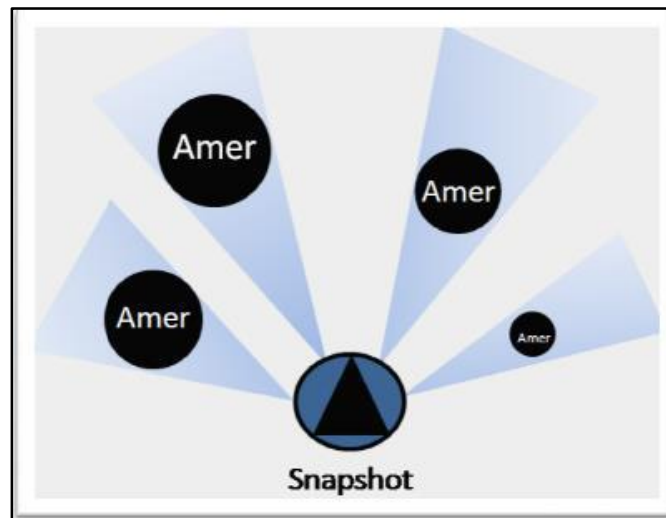
This basic ability allows the robot to move towards an object visible from the current position of the robot. It is generally achieved by a gradient ascent based on the perception of the object, as in the famous example of Valentino Braitenberg's vehicles [29], which use two light sensors to reach or flee a light source. This strategy uses reflex actions, in which each perception is directly associated with an action (Figure I-19).



**Figure I-19: In Braitenberg vehicles, the speed of each of the robot's two motors depends on the values from two sensors that detect the light emitted by the goal**

## B. Guidance

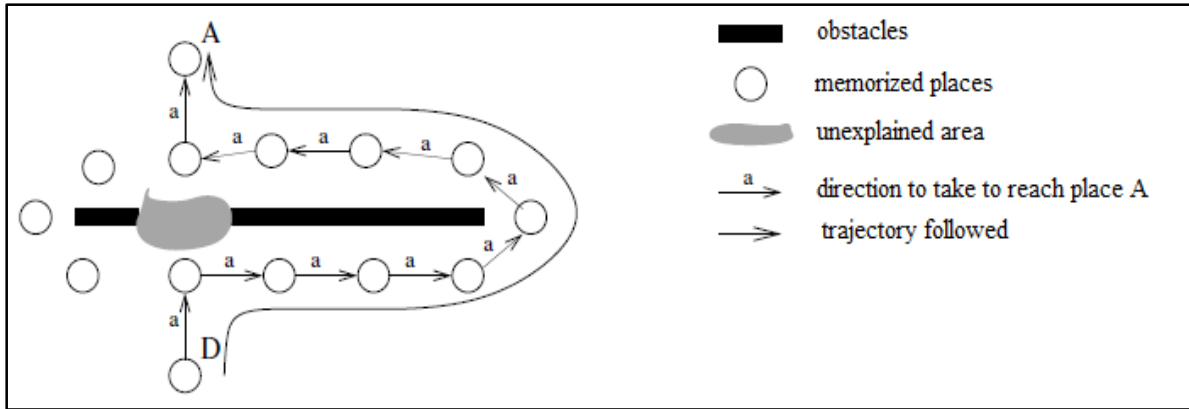
This ability makes it possible to reach a goal, which is not a directly visible object, but a point in space characterized by the spatial configuration. The navigation strategy, often a gradient descent as well, is then to head in the direction that allows reproducing the desired configuration (Figure I-20).



**Figure I-20: Reactive navigation: Guidance**

## C. Action associated with a place

This ability is the first ability performing a global navigation. In this case, the mobile robot can reach a goal or landmarks with invisible location or characteristics. It requires an internal representation of the environment, which consists in defining places as areas of space in which perceptions remain similar, and in associating an action to be carried out at each of these places (see figure I-21). The sequence of actions associated with each of the recognized places defines a route that makes it possible to reach the goal. These models therefore allow greater autonomy but are limited to a fixed goal. A road that makes it possible to reach a goal cannot in fact be used to reach a different goal. Changing goals will teach you a new route, independent of the routes to reach the other goals.



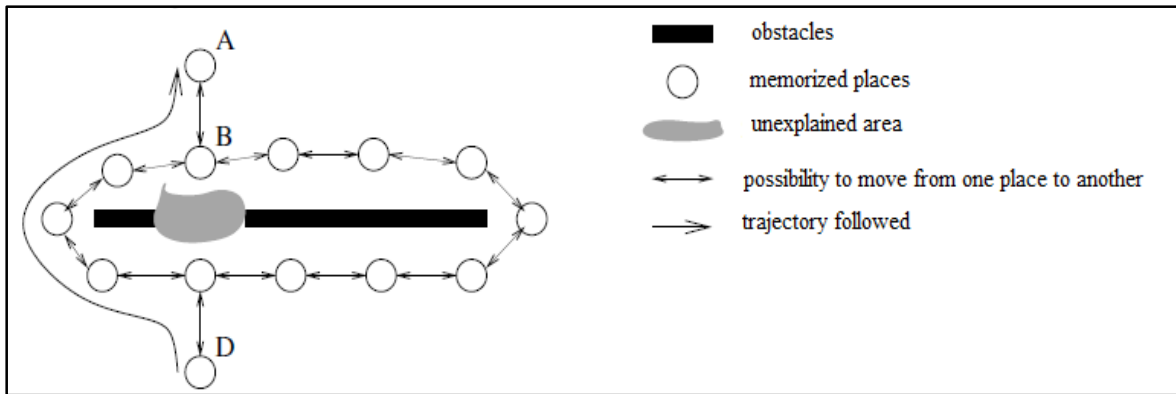
**Figure I-21: Action associated with a place [27].**

In each place, represented by a circle, the action to be taken to reach goal A is represented by an arrow indicating the direction to follow from this place. This strategy makes it possible to reach a distant goal in the environment but is based on fixed paths. In this example, the path from location D to location A and passing through the right of the obstacle has been learned. Joining place A from place D can only be made by this path. The shortcut on the left path, for example, cannot be used.

## Global navigation

### A. Topological navigation

This capacity is an extension of the previous one, which memorizes in the internal model the spatial relationships between the different places. These relationships indicate the possibility of moving from one place to another, but are no longer associated with a particular goal. Thus, the internal model is a graph, which makes it possible to calculate different paths between two arbitrary places. However, this model only allows planning of trips among known places and along known paths (see figure I-22).



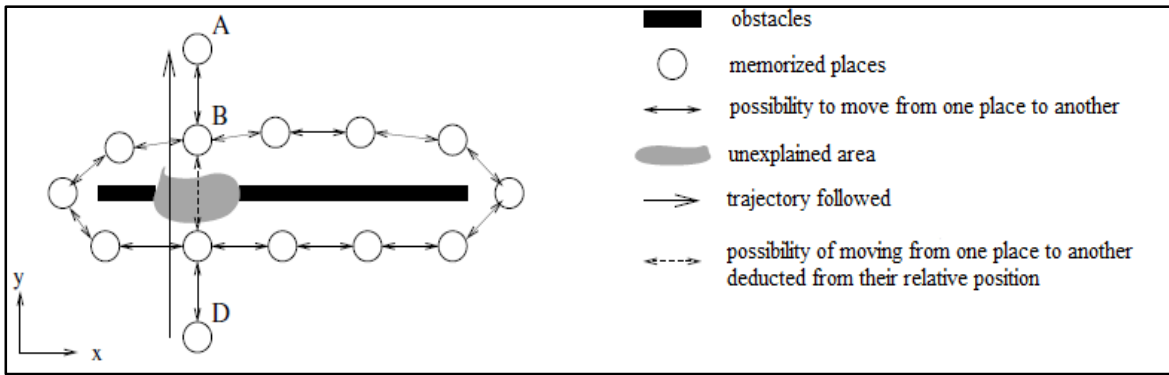
**Figure I-22: Topological navigation [27].**

This strategy allows you to memorize a set of places and the possibilities of moving from one to another, regardless of any goal. To reach a goal, there must be a planning stage, which makes it possible to find, among all the possible paths, the path joining the goal. In our example, the shortest path between D and A can then be calculated, but only among the places and paths already known. This strategy allows, for example, bypassing the obstacle on the left but does not allow crossing it in a straight line from D to A.

### **B. Metric navigation**

This capacity is an extension of the previous one because it allows the robot to plan paths within unexplored areas of its environment. To do this, it memorizes the relative metric positions of the different places, in addition to the possibility of switching from one to the other. These relative positions allow, by simple composition of vectors, to calculate a trajectory going from one place to another, even if the possibility of this displacement has not been memorized in the form of a link (cf. figure I-23).





**Figure I-23: Metric navigation [27].**

This strategy makes it possible to calculate the shortest path between two memorized places, even making it possible to plan shortcuts within unexplored areas of the environment. For this, the map memorizes the relative metric position of each of the places visited by the robot. Thus, it is possible to plan a trip between two places, even if the possibility of this trip is not recorded in the map. In this example, this strategy allows you to go from place A to place D by crossing the unexplored area.

The models in the first three categories use reflex actions to guide the robot and are essentially differentiated by the type of perceptions used to trigger these actions. They are grouped under the generic term of reactive navigation.

They can be very simple, do not require a global model of the environment but have a field of application often restricted. Behaviors of this type, however, remain essential in modern robots because of their simplicity, they are generally executed very quickly and they make it possible to perform low-level tasks, such as the avoidance of unforeseen obstacles, essential to the safety of a robot.

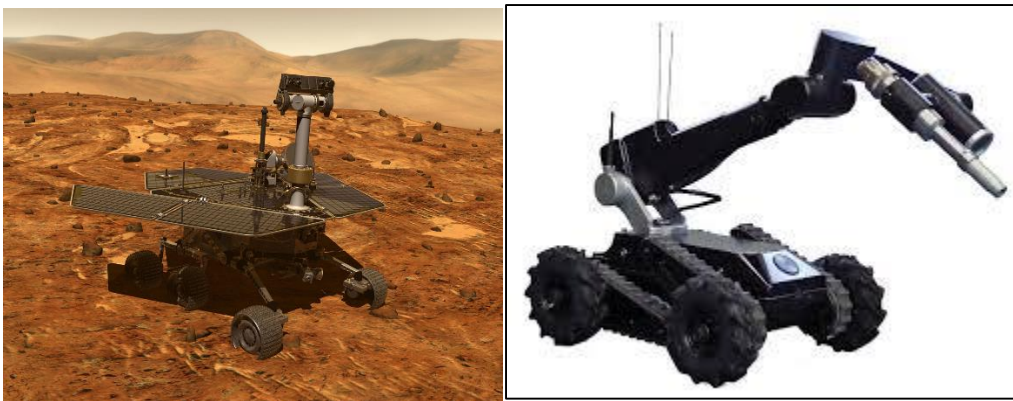
### **I.3. Wheeled mobile robot (WMR)**

#### **I.3.1. Definition, current applications, and future potentials**

Unlike the majority of industrial robots that can only move about a fixed frame in a specific workspace, the WMR has a distinct feature of moving around freely within its predefined workspace to fulfill a desired task. The mobility of WMR makes it suitable for a variety of applications in structured as well as in unstructured environments. For examples, Spirit, the

NASA's Mars rovers [30] have successfully demonstrated its ability to achieve the mission goals in exploring and running experiments on the red planet (Figure I-24.a). In military and high-risk hazardous environments, AB Precision Ltd [31] has developed Cyclops, a miniature remotely operated vehicle that has been in use in many military and law enforcement organizations worldwide (Figure I-24.b). It provides distinct advantages over human operators to complete critical missions in a safe manner.

The list goes as the WMR can also be found in other field of applications such as in mining, transportation, entertainment and so on. The ever-increasing demand and applications of WMRs justify the research needs and potentials of this very fascinating topic. We should expect WMR in the future to have stronger autonomous capabilities and higher agility, be able to self-learn and reliable for continuous operation regardless of time and environment.



(a)

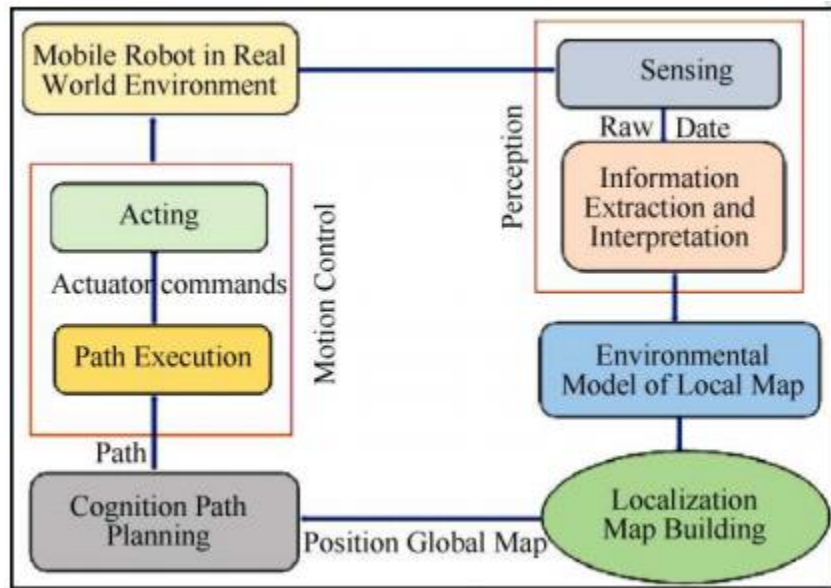
(b)

**Figure I-24: (a) Spirit – NASA Solar System Exploration. (b) Cyclops Mk4D – Miniature Remotely Operated Vehicle (MROV).**

### **I.3.2. Research on WMR - modeling, planning and control**

In general, the research on WMR can be divided into several components namely: WMR modeling, localization, planning and navigation strategies as well as mobility and communication system [32].

The relationship between all these components is shown in figure I-25.



**Figure I-25: The relationship between components in the autonomous control application of WMR**

The research in WMR mobility is related to understand the physical mechanics of the robot platform, the model of the interaction between the robot and its environment as well as the overall effect of control algorithm on the WMR. In localization, the research objective is to estimate the position, attitude, velocity and acceleration of the WMR.

Navigation is concerned by the acquisition and response to external sensed information to execute the mission. Meanwhile, research in planning is related to behaviors, trajectories or waypoints generation for the robot mission. Lastly, the goal of communication research is to provide the link between WMR and any remaining elements in the whole system, including system operators or other WMRs.

While the modeling of WMR has been extensively studied from an ideal perspective in which the wheel rolls without slip and the WMR does not move laterally instantaneously, there is little research that models wheel slip and consider the effect of traction force on the motion of the WMR.

Majority of these WMR platforms use standard wheels over omni-directional wheels due to the inherent mechanical simplicity. These WMRs are called non-holonomic mobile robots because of the velocity constraints imposed due to the structure of the wheels. A car is an example of a four-wheel vehicle system that shares many similarities with a WMR system due to the same wheel structure.

#### **I.4. Differential drive wheeled mobile robot (DDWMR)**

Differential drive robots are two-wheeled mobile robots with independent actuators for each wheel. Structural simplicity and low cost of production makes this class of mobile robots very popular for many applications. Controlling differential drive robots has been studied by many researchers due to its connection to the control of under-actuated systems with non-holonomic constraints. We do not intend to review the rich literature of differential drive robots in this manuscript; however, several important studies are cited here to highlight the diversity of control strategies used in this area.

As with other mobile robots, one of the important control problems for differential drive robots is steering the robot on a desired path, which is known as the tracking control. Inspired by the parallel parking problem, the other challenge for wheeled mobile robots is posture stabilization where the control objective is steering the robot from an initial configuration (position and heading) to a final configuration [33] [34]. Consequently using time-varying feedback laws [35] [36] and discontinuous control strategies [37] [38] have been common for stabilization of wheeled mobile robots. Back-stepping methods [39], static feedback linearization [40] and dynamic feedback linearization [41] are the standard nonlinear control techniques that have been used to address the posture stabilization of wheeled mobile robots. On the other hand, robust and adaptive control approaches [42] as well as nonlinear control techniques such as dynamic feedback linearization [43] have been used to tackle the tracking control of wheeled robots.

Although these model based control methods provide rigorous strategies for steering wheeled robots, it is often challenging to generalize them to more realistic cases such as input saturation. Among the papers that have addressed control design considering the bounds on the input magnitudes, using Model Predictive Control (MPC) approaches have been the common trend [44].

Recently developed Constrained Directions method addresses the control of mobile robots by studying the directions a robot can move at each instant of time during the steering process [45]. The fundamental idea of this method is that the directions of motion a robot can achieve are limited by the mechanical constraints imposed by the system model and the boundedness of the input control signals. Having derived the set of reachable directions, Constrained Direction method devises control schemes by employing an efficient search method to choose the best direction of

motion within reachable directions. Since this method does not impose any restrictive assumptions on the robot's model, it can be adopted for steering of any mobile robot model. For example, Constrained Directions method is applied to point-to-point steering of the dynamic model of a two-wheeled differential drive robot under slip in [45], while [46] uses the same concept for steering the kinematic model of the same robot under the assumption of rolling without slipping.

### **I.5. Robustness analysis for mobile robots**

Uncertainty arises if the robot lacks critical information for carrying out its task. It arises from five different factors [47]:

1. **Environments:** Physical worlds are inherently unpredictable. While the degree of uncertainty in well-structured environments such as assembly lines are small. Environments such as highways and private homes are highly dynamic and unpredictable.
2. **Sensors:** Sensors are inherently limited in what they can perceive. Limitations arise from two primary factors. First, range and resolution of a sensor is subject to physical laws. For example, Cameras cannot see through walls, and even within the perceptual range, the spatial resolution of camera images is limited. Second, sensors are subject to noise, which perturbs sensor measurements in unpredictable ways and hence limits the information that can be extracted from sensor measurements.
3. **Robots:** Robot actuation involves motors that are, at least to some extent, unpredictable, due to many effects like control noise and wear-and-tear. Some actuators, such as heavy-duty industrial robot arms, are quite accurate. Others, like low-cost mobile robots, can be extremely inaccurate.
4. **Models:** Models are inherently inaccurate; they are abstractions of the real world. Model errors are a source of uncertainty that has largely been ignored in robotics, despite the fact that most robotic models used in state-of-the-art of robotics systems are rather crude.
5. **Computation:** Robots are real-time systems, which limits the amount of computation that can be carried out.

All of these factors give rise to uncertainty. Traditionally, such uncertainty has mostly been ignored in robotics. However, as robots are moving away from factory floors into increasingly unstructured environments, the ability to cope with uncertainty is critical for building successful robots.

In our work, we take into consideration four aspects of uncertainty, which are:

- Modeling error
- Localization error by odometry
- Motor efficiency
- Wheel slip

### **I.5.1. Modeling error**

Much of the research on the mobile robot focuses on designing navigation methods for a mobile robot under non-holonomic constraints.

In these methods, however, a perfect knowledge of the parameter values of a mobile robot is required. In general, this requirement cannot be fulfilled. In practical situations, it is almost impossible to get exact values of the parameters of a mobile robot. There are relatively few results on the navigation problem for a mobile robot. In addition, there is almost little literature on the robustness of the controller in the presence of uncertainties and modeling errors in the dynamic model of a mobile robot.

By the way, there are many possible methods that can be used even when knowledge about the model is not complete, such as adaptive control methods and robust control methods [48].

In our work, we took two aspects of modeling error, developed as follows:

- 1- An uncertainty about the length  $L$  of the robot

We considered that the geometric dimensions of the robot's length were not precise, more precisely:

$$L = L + \Delta L \quad (I.1)$$

$L$ : Robot length

$\Delta L$  : Deviation along the length of the robot

2- An uncertainty about the radius of a wheel

This uncertainty is due to tire wear (inflation errors, vehicle overloading, driving style, and road condition) to assess the performance of the robot controller.

To do this, we carried out robustness tests taking as the radius of the left wheel

$$R_l = R_l - \Delta R_l \quad (I.2)$$

$R_l$  : Left wheel radius

$\Delta R_l$ : Left wheel radius deviation

### **I.5.2. Localization error by odometry**

Odometry is one of the basic localization systems in any autonomous vehicle [49] [50]. It is based on the use of data from on-board sensors in order to estimate changes in position and orientation from the vehicle itself, and is subsequently used in many autonomous systems to estimate their position relative to a starting location, by integrating sensors measurements. However, this method is sensitive to errors due to the integration over time and the final position is usually not very accurate. Usually, the odometry output of a robot is very poor, it is only valid for a few meters, and needs others sensors to obtain a good localization system. Any small increment in odometry accuracy can improve the whole localization system a lot.

Usually, odometry is used in combination with positioning systems like GPS, lasers, radio frequency markers, natural or artificial beacons, and others [51]. When mechanical odometry is not available, visual odometry can be used [52] to estimate vehicle position from changes between images. A complete sensorial system for an autonomous vehicle is based on multiple sensors combined to get position and orientation [53] [54]. Some algorithms used for this purpose are Kalman filters [55], particle filters based on Montecarlo simulation [56], etc. Sensors excluding odometry usually need external information obtained from the environment, so in many situations these sensors simply do not work correctly. For example, GPS loses coverage when the vehicle does not have a full sky vision [57].

As another examples, if we introduce beacons, we need to structure the whole environment, or a laser needs available features to recognize in the environment, and these features should be in the range of action of the laser (typically between 10–20 m) [58].

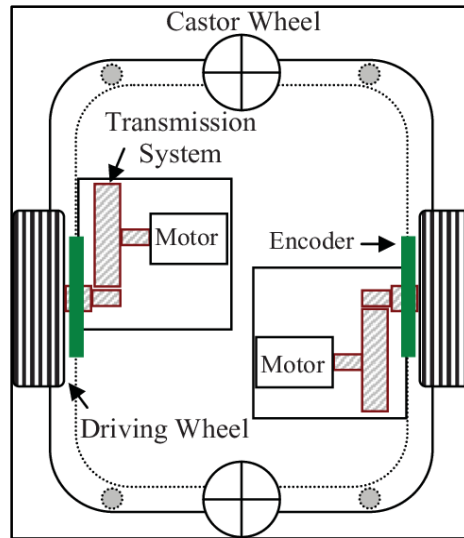
The main advantage of odometry is that all localization information comes from the robot itself.

Odometry information is always available and usually it is the only localization information when other sensors are not able to provide data, so a good odometry based localization system is always necessary and it is usually the first step to localization [59] [60], obstacle detection [61], and navigation [62].

In wheeled robots, odometry is based on the movement of each wheel. A rotation sensor (rotation optical encoder) is attached to each drive wheel of the robot (figure I-26), and, knowing the wheel diameter, it is possible to approximate linear displacement of each wheel. Using each wheel translation and the separation between wheels, position, and orientation of the robot (pose) is calculated. All of the calculation is based on optical encoder information, which obtains, in real time, the rotation angle of each wheel. The sensor and all of the parameters can be affected by errors, so final pose based on odometry usually is very noisy.

The main disadvantage of odometry is incremental error; odometry starts in a known pose, and this pose is updated with small increments using the integration of information acquired from sensors. Errors grow very fast due to the integration of sensor data, so a continuous calibration system is crucial.





**Figure I-26: Main components of differential drive mobile robot**

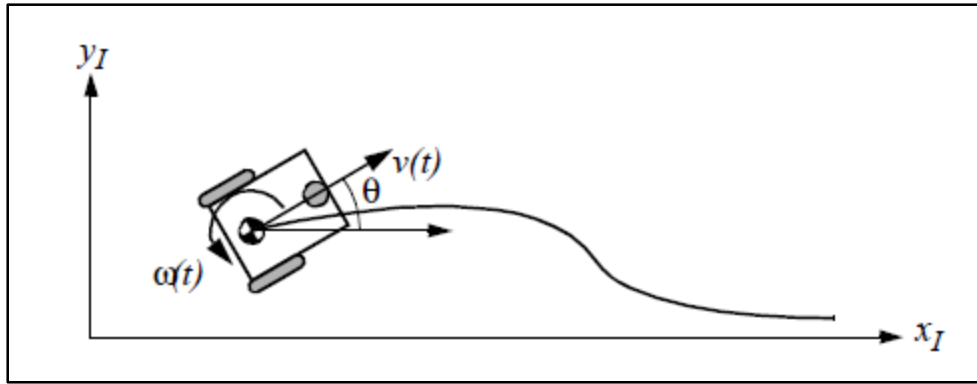
In the following, we will concentrate on odometry based on the wheel sensor readings of a differential drive robot only.

There are many sources of odometric error, from environmental factors to resolution:

- Limited resolution during integration (time increments, measurement resolution, etc.)
- Misalignment of the wheels (deterministic)
- Unequal wheel diameter (deterministic)
- Variation in the contact point of the wheel
- Unequal floor contact (slipping, non-planar surface, etc.)

Some of the errors might be deterministic (systematic), thus, they can be eliminated by proper calibration of the system. However, there are still a number of non-deterministic (random) errors, which remain, leading to uncertainties in position estimation over time. From a geometric point of view, one can classify the errors into three types:

- Range error: integrated path length (distance) of the robots movement: Sum of the wheel movements
- Turn error: similar to range error, but for turns: Difference of the wheel motions
- Drift error: difference in the error of the wheels leads to an error in the robot's angular orientation



**Figure I-27: Movement of a differential drive robot**

Over long periods of time, turn and drift errors far outweigh range errors, since their contribution to the overall position error is nonlinear. Consider a robot, whose position is initially perfectly well known, moving forward in a straight line along the x-axis. The error in the y position introduced by a move of d meters will have a component of, which can be quite large as the angular error  $\Delta\theta$  grows. Over time, as a mobile robot moves about the environment (figure I-27), the rotational error between its internal reference frame and its original reference frame grows quickly. As the robot moves away from the origin of these reference frames, the resulting linear error in position grows quite large. It is instructive to establish an error model for odometric accuracy and see how the errors propagate over time.

Generally, the pose (position) of a robot is represented by the vector:

$$p = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \quad (\text{I.3})$$

For a differential drive robot, the position can be estimated starting from a known position by integrating the movement (summing the incremental travel distances). For a discrete system with a fixed sampling interval  $\Delta t$  the incremental travel distances ( $\Delta x$ ;  $\Delta y$ ;  $\Delta\theta$ ) (figure I-28) are:

$$\Delta x = \Delta s \cos(\theta + \Delta\theta/2) \quad (\text{I.4})$$

$$\Delta y = \Delta s \sin(\theta + \Delta\theta/2) \quad (\text{I.5})$$

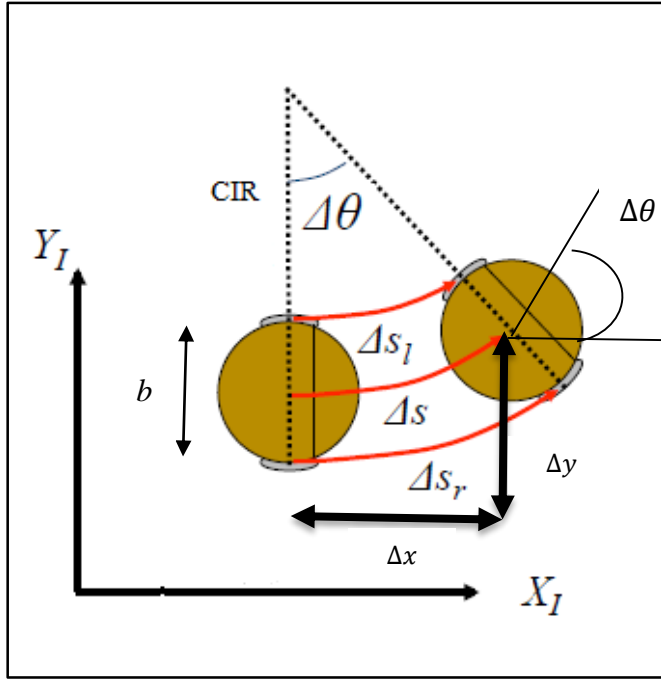
$$\Delta\theta = \frac{\Delta s_r - \Delta s_l}{b} \quad (\text{I.6})$$

$$\Delta s = \frac{\Delta s_r + \Delta s_l}{2} \quad (I.7)$$

$(\Delta x; \Delta y; \Delta\theta)$ : Path traveled in the last sampling interval

$\Delta s_r; \Delta s_l$ : Traveled distances for right and left wheel respectively

$b$ : Distance between the two wheels of differential drive robot



**Figure I-28: Modeling of the odometry of the robot allowing to estimate the position of the robot from measurements of the incremental wheel encoders (CIR = instantaneous center of rotation)**

Thus, we get the updated position  $p'$ :

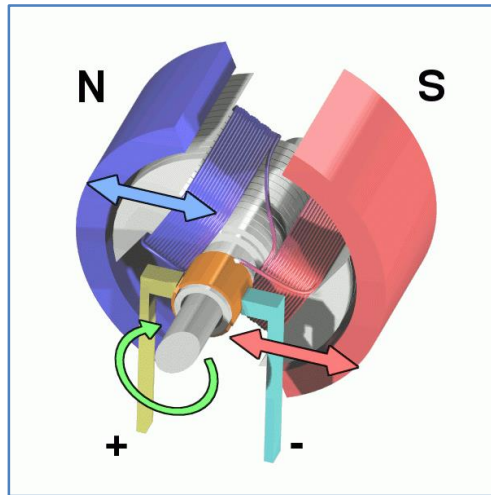
$$p' = \begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = p + \begin{bmatrix} \Delta s \cos(\theta + \Delta\theta/2) \\ \Delta s \sin(\theta + \Delta\theta/2) \\ \Delta\theta \end{bmatrix} = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \begin{bmatrix} \Delta s \cos(\theta + \Delta\theta/2) \\ \Delta s \sin(\theta + \Delta\theta/2) \\ \Delta\theta \end{bmatrix} \quad (I.8)$$

By using the relation for  $(\Delta s; \Delta\theta)$  of equations (I.7) and (I.8) we further obtain the basic equation for odometric position update (for differential drive robots) (figure I-28):

$$p' = f(x, y, \theta, \Delta s_r, \Delta s_l) = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \begin{bmatrix} \frac{\Delta s_r + \Delta s_l}{2} \cos(\theta + \Delta\theta/2) \\ \frac{\Delta s_r + \Delta s_l}{2} \sin(\theta + \Delta\theta/2) \\ \frac{\Delta s_r - \Delta s_l}{b} \end{bmatrix} \quad (I.9)$$

### I.5.3. Motors efficiency

As the field of mobile robotics is growing, so is the demand for lightweight and energy-efficient actuators. Energy efficiency is important because it allows for a longer autonomy of the robot and reduced weight of the battery pack, again decreasing energy demand and possibly increasing dexterity.



**Figure I-29: DC Motor**

Many designs feature DC motors (figure I-29), which, on top of their easy implementation and flexibility, offer the advantage of having high maximum efficiency values. However, unlike most other industrial applications in which motors are working close to their rated point of operation, robots typically require the motor to be operated at a variable speed and load. In the latter case, additional inertial loads have to be overcome, and the motor will no longer be able to operate at its maximum efficiency. Moreover, in lightweight applications, motors are often pushed to their torque limits, operating far away from their most efficient region.

In the past years, many papers have appeared in which actuators for mobile applications are optimized for energy efficiency. Very often, calculations are based entirely on the energy consumption at the output shaft of the motor [63] [64] [65] [66] [67], which implies that the motor

efficiency's dependency on torque and speed and its inertia are neglected. Some authors also consider electrical power consumption by introducing a DC motor efficiency model. Typically, the model includes resistive losses [68] [69] and sometimes losses proportional to motor speed [70] [71]. In those papers, which utilize a DC motor model, inertia of the motor is usually included, but gearbox inertia is rarely taken into account [68]. In many actuator systems, however, the reflected inertia of the motor and gearbox is much larger than the link inertia [72], which implies that generally these inertias cannot be neglected. This is recognized in [73], where the authors present a motor-gearbox selection method that includes inertias and a DC motor model. Other evidence of the importance of DC motor models can be found in Brown and Ulsoy [71]. In an optimization on a robot arm, the authors claim that 2/3 of the energy savings they accomplished could be attributed to a more efficient use of the motor.

Electric motor efficiency is the measure of the ability of an electric motor to convert electrical energy to mechanical energy; i.e., kilowatts of electric power are supplied to the motor at its electrical terminals, and the horsepower of mechanical energy is taken out of the motor at the rotating shaft. Therefore, the only power absorbed by the electric motor is the losses incurred in making the conversion from electrical to mechanical energy. Thus, the motor efficiency can be expressed as

$$Efficiency = \frac{\text{mechanical energy out}}{\text{electrical energy in}} \times 100\% \quad (I.8)$$

But *Mechanical energy out = electrical energy in – motor losses*

Or *Electrical energy in = mechanical energy out – motor losses*

Therefore, to reduce the electric power consumption for a given mechanical energy out, the motor losses must be reduced and the electric motor efficiency increased.

#### **I.5.4. Wheel slip**

In recent years, control problems for wheeled mobile robots (WMR) are considered remarkable, due to their inherent nonlinear properties such as nonholonomic constraints and their wide applicability in various areas. Many researchers, all over the world, have paid attention to

solving the tracking problems of WMRs by employing various techniques with the assumption “pure rolling without slip” being always satisfied.

However, in practice, the assumption “pure rolling without slip” is often violated due to various factors such as slippery floor, external forces, and so on. The wheel slip is one of the reasons making the tracking performance of nonholonomic WMRs reduce considerably. Therefore, if one wants the tracking performance of the WMRs to be improved in such context, then control methods having the ability to overcome the undesired effects of the wheel slips must be taken into account.

Several research results have been published for trajectory tracking of nonholonomic WMRs subject to the wheel slips. In particular, an adaptive tracking control method by means of neural networks was proposed in [74] in order to overcome the harmful effect of the wheel slips. By employing gyro-sensors and encoders, the slip ratios were calculated in [75] [76].

Then the control approaches to compensate the wheel slips were proposed in these reports. The work in [77] developed a robust controller dealing with not only slip-kinematics but also slip dynamics by employing the framework of differential flatness. The authors in [78] modeled overall the dynamics of a WMR subject to the wheel slips. Next, they proposed a discontinuous control technique for regulation control task and a SMC technique for sharp turning control task.

On the other hand, the neural network (NN) has been utilized as one of the intelligent techniques to enhance the performance of closed-loop control systems. Unlike classification applications, the NN in feedback control application seems to be part of the closed-loop control system. For this reason, it is useful in order to have a NN closed-loop control system with on-line learning algorithms, a new strategy to navigate to a known target location in an unknown environment using a combination of the “go-to-goal” approach and reinforcement learning with biologically realistic spiking neural networks is proposed in [79]. About the works in [80], a neural network-based adaptive sliding mode control (ASMC) method for tracking of a non-holonomic wheeled mobile robot (WMR) subject to unknown wheel slips, model uncertainties and unknown bounded disturbances have been proposed.

Other works like [81], proposed a command by sliding mode, the dynamic model of a wheeled mobile robot (WMR) is derived by assuming the longitudinal and lateral slip of the wheels.

## **I.6. Conclusion**

This chapter provides an overview of mobile robotics in general and the DDWMR more specifically. We find that DDWMR has seen great interest in the last decade; this is mainly due to the growing advances in instrumentation and calculator technologies and the advantages that WMR offers over other mobile robot models. This chapter presents a state of the art on modeling, navigation and robustness analysis of DDWMR robot it constitutes a solid framework for novel contributions and original developments.

**CHAPTER II:**  
**UNCERTAINTIES AND ERRORS MODELLING FOR**  
**A MOBILE ROBOT**



## Chapter II. UNCERTAINTIES AND ERRORS MODELLING FOR A MOBILE ROBOT

### II.1. Introduction

In mobile robotics, the dynamic system to be controlled, includes vehicle dynamics, wheel and ground properties, as well as sensors and actuators properties, is non-linear. In addition, it is very uncertain and its parameters can vary over time, depending on the terrain encountered, tire wear, brakes, etc. To deal against these uncertainties and possible faults that can affect the mobile robot, several types of robust and adaptive control are implemented. A suitable controller for mobile robot navigation requires a deep robustness analysis of robotic system. In this chapter, we will describe some types of faults and uncertainties encountered in the wheeled mobile robot used in our work. The following figure shows these different uncertainties:

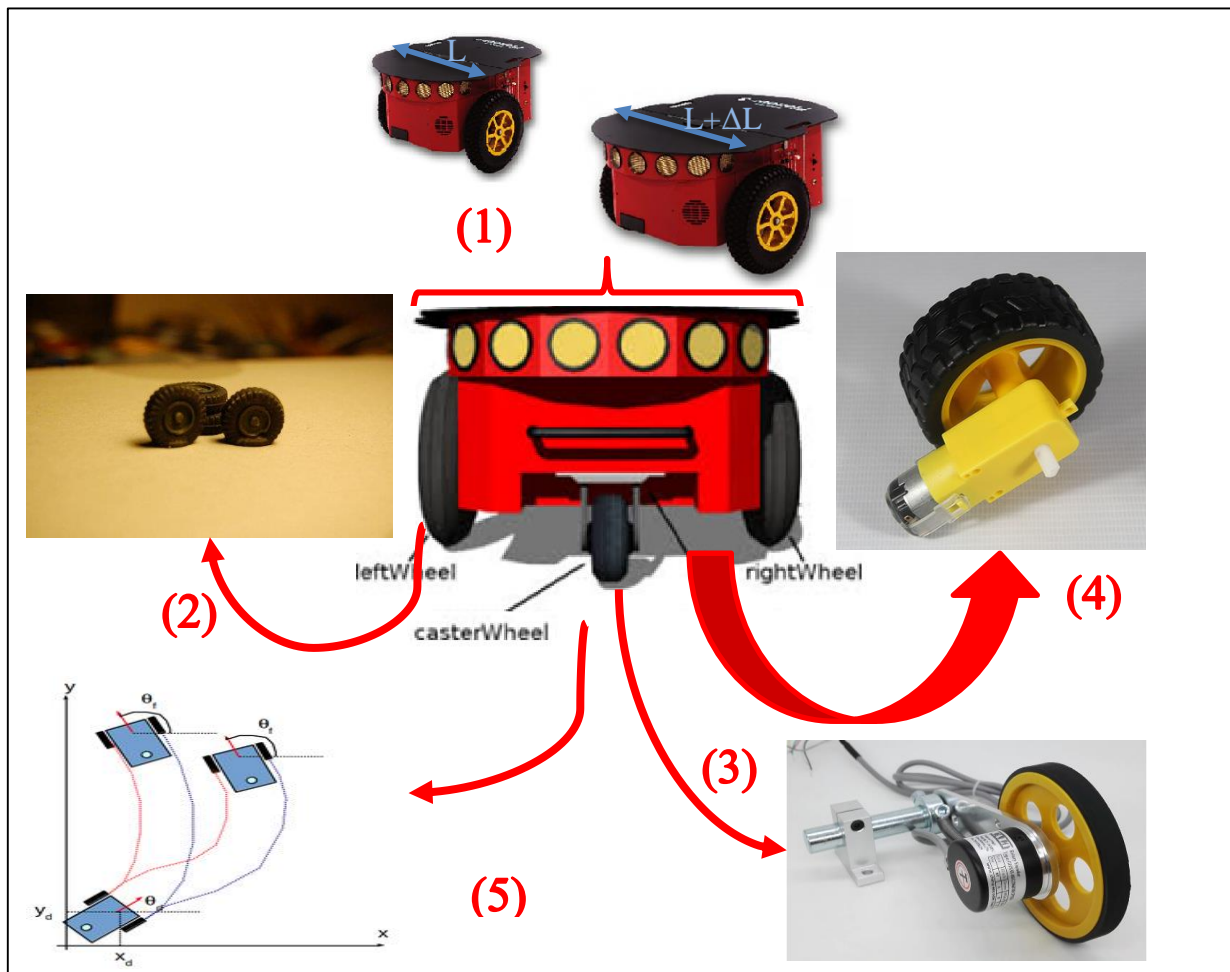


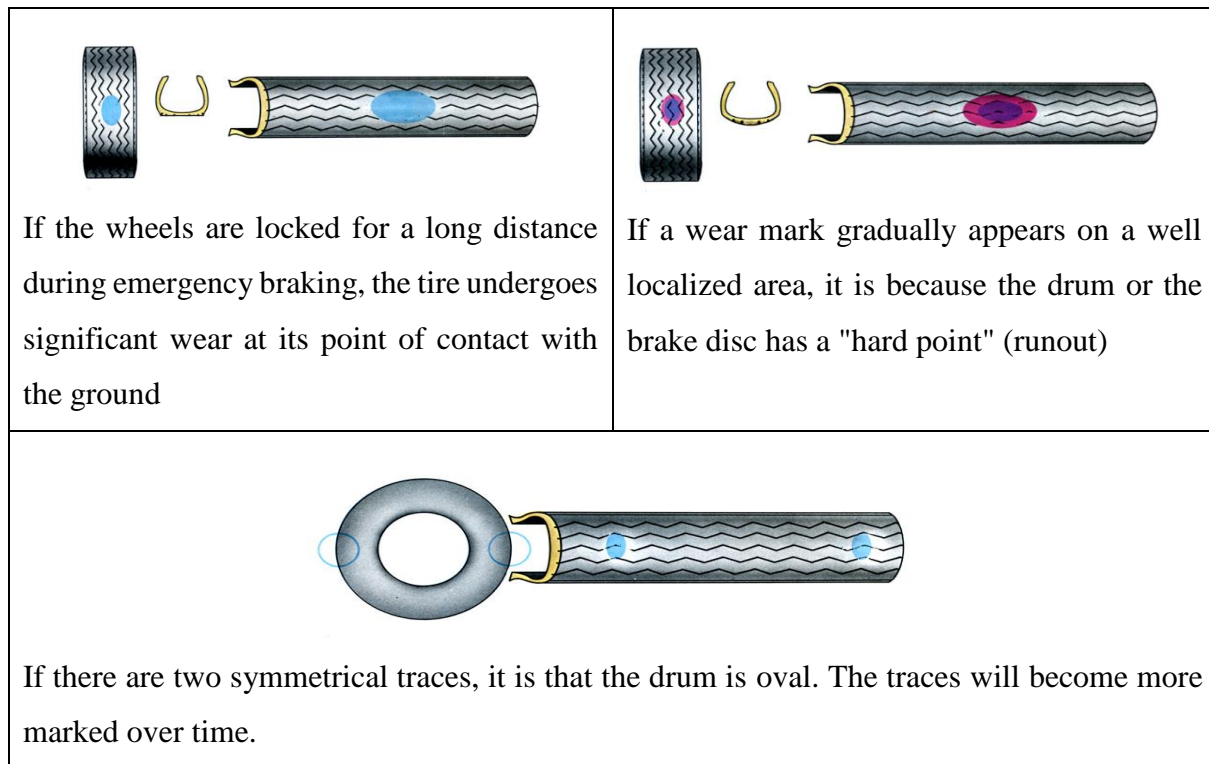
Figure II-1: Different faults and uncertainties applied in this work: (1) an uncertainty on the length  $L$  of the robot model, (2) uncertainty on the radius  $R$  of the left wheel, (3) localization errors, (4) loss of efficiency of motors, (5) wheel slip

## II.2. An uncertainty on the radius $R$ of each wheel

This uncertainty is due to tire wear. Inflation errors, vehicle overload, driving style or road conditions are all causes of motion inaccuracy, which is an important indicator to evaluate the performance of robots.

When driving, the tread of any tire rubs against the road, causing tire wear, which is normally slow and even. It is inevitable and even necessary to transmit forces to the road, whether they are tractive forces, such as acceleration or braking, or drift forces such as when passing through curves. The slip indicates the relative movement of the tire with respect to the road surface, which occurs during the transmission of forces.

### II.2.1. The inflation



#### II.2.1.1. Under-inflation

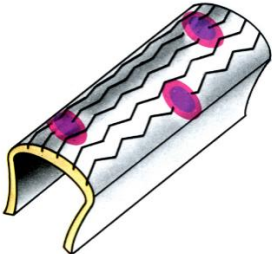

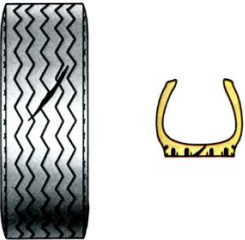
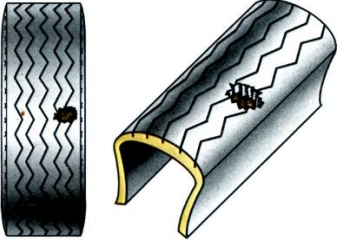
It is an important cause of wear, which cannot be blamed either on the tires or on the design or the state of wear of the vehicle components. We can notice, on the outside of the tire, more wear on the edges of the tread than in the middle. Inside, you can observe a rupture or detachment of the fabrics due to exaggerated bending of the sides or an abnormal heating. The manufacturer

recommends pressures (cold) for normal use and pressures of the order of 0.2 to 0.4 bar kg / cm<sup>2</sup>. When you drive quickly or with a loaded vehicle. Heating an under-inflated tire can cause it to burst.

### II.2.1.2. Over-inflation

It leads to greater external wear on the middle of the tread than on the edges. With radial ply tires, a slight over-inflation is preferable.

### II.2.2. Overload and type of driving

 <p>Irregular marks are a sign of a defect in hunting.</p>	 <p>A deep, localized cut is caused by a sharp object.</p>
 <p>If the cut is deep, water can get inside the tire, between the tire and the inner tube.</p>	 <p>When a piece of rubber comes off, it is a poor quality tire.</p>

### II.2.3. The condition of a vehicle



A highly unbalanced wheel will cause vibration in the steering and marked wear on the tire in a fairly regular ripple shape.	If the wear is less regular, distributed over a larger area, it is then disrupted or worn wheel bearings, but also play in the front axle joints or worn shock absorbers.
--	---

### II.3. Localization error

Localization is one of the most fundamental aspects of a mobile robot. All mobile robot system has to answer the fundamental question, which is “Where Am I”, i.e. the current location & orientation of the robot has to be obtained, so that the robot can easily move from source to destination. There are many localization systems and techniques for mobile robot navigation; however, in the present work we have used odometry to measure the movement of the robot. In this case, data are obtained from an incremental encoder (odometry), which is fitted along with a motor of the mobile robot. Incremental encoders measure the rotation of the wheels, which in turn, calculates robot position and orientation using integration approaches of kinematic model over  $[t_k, t_{k+1}]$ .

#### II.3.1. Odometry

Odometry is the most widely used navigation method for mobile robot positioning because it provides good short-term accuracy. It is inexpensive, and allows very high sampling rates. However, the fundamental idea of odometry is the integration of incremental motion information over time, which leads inevitably to errors accumulation. Particularly, the accumulation of orientation errors will cause large position errors [82], which increase proportionally with the distance travelled by the robot. Nonetheless, most researchers agree that odometry is an important part of a robot navigation system and that navigation tasks will be simplified if odometric accuracy can be improved.

Odometry is the measurement of wheel rotation as a function of time. If the two wheels of the robot are joined to a common axle, the position and orientation of the center of the axle relative to the previous position and orientation can be determined from odometry measurements on both wheels. In practice, optical encoders that are mounted onto both drive wheels feed discrete wheel increment information to the central processor, which in turn continually updates the robot’s state using geometric equations. However, with time, odometric localization accumulates errors in an

unbounded fashion due to wheel slippage, floor roughness and discretized sampling of wheel increments. Many research works have been undergone at both the hardware and theoretical level to improve the reliability of odometry. When trying to measure and reduce odometry errors, it is important to understand the distinction between systematic and non-systematic odometry errors.

Systematic errors are those errors that are inherently part of the robot's kinematic or controller and sensors properties, independently from the robot's environment.

Non-systematic errors are those that depend on the robot's environment and differ from one environment to another. This distinction is important because each one of these two groups affects mobile platforms differently, their remediation differs, and, most important, both groups require different measuring techniques in order to obtain meaningful and comparable experimental data.

We categorize odometry errors as follows:

#### **II.3.1.1. Systematic errors**

- Unequal wheel diameters
- Average of both wheel diameters differs from nominal diameter
- Misalignment of wheels
- Uncertainty about the effective wheelbase (due to non-point wheel contact with the floor)
- Limited encoder resolution
- Limited encoder sampling rate

#### **II.3.1.2. Non-systematic errors**

- Travel over uneven floors
- Travel over unexpected objects on the floor
- Wheel-slippage (slippery floors, over-acceleration, skidding in fast turns, etc.)
- External forces (interaction with external bodies)
- Internal forces (e.g., castor wheels)
- Non-point wheel contact with the floor

## **II.3.2. Robot positioning calculation**

There are many methods for robot positioning that can roughly be categorized into two groups: relative and absolute position measurements. Odometry is one of the relative position measurement methods. This method uses encoders to measure wheel rotation and/or steering orientation.

### **II.3.2.1. Rotational Displacement Equipment**

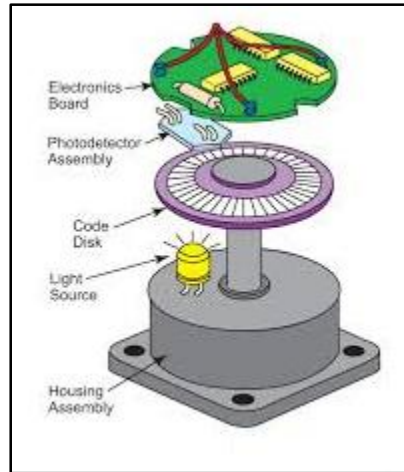
There are different types of rotational displacement and velocity sensors in use today:

1. Brush encoders
2. Potentiometers
3. Resolvers.
4. Optical encoders.
5. Magnetic encoders
6. Inductive encoders
7. Capacitive encoders

For mobile robot applications, incremental optical encoders are the most popular type

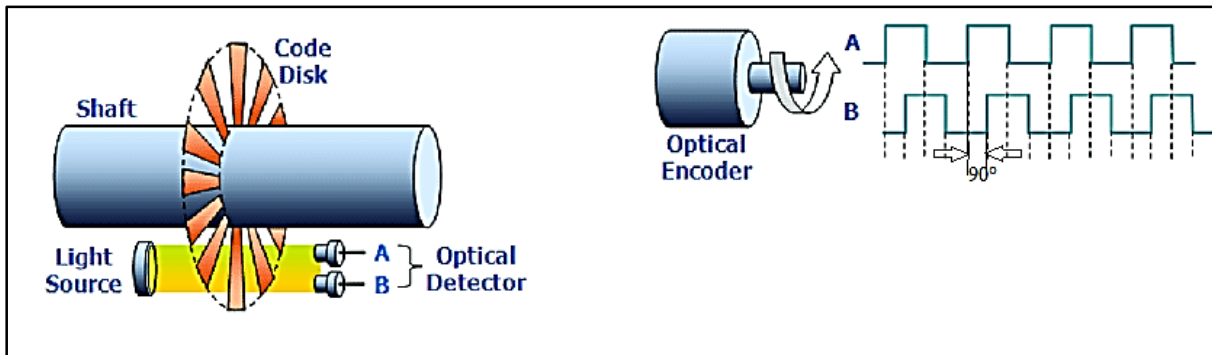
#### ***Incremental Optical Encoders***

An encoder is an electrical mechanical device that converts linear or rotary displacement into digital or pulse signals. The most popular type of encoder is the optical encoder, which consists of a rotating disk, a light source, and a photo detector (light sensor). The disk, which is mounted on the rotating shaft, has patterns of opaque and transparent sectors coded into the disk (see figure II-2). As the disk rotates, these patterns interrupt the light emitted onto the photo detector, generating a digital or pulse signal output



**Figure II-2: Rotary Incremental Optical Encoder**

An incremental encoder generates a pulse for each incremental step in its rotation. Although the incremental encoder does not output absolute position, it can provide high resolution with suitable price. For example, an incremental encoder with a single code track (tachometer encoder), generates a pulse signal whose frequency indicates the velocity of displacement. However, the output of the single-channel encoder does not indicate direction. To determine direction, a two-channel, or quadrature, encoder uses two detectors and two code tracks.



**Figure II-3: Examples of the A pulse and B pulse. If the A pulse occurs before the B pulse, the shaft is turning clockwise, and if the B pulse occurs before the A pulse, the shaft is turning counter-clockwise**

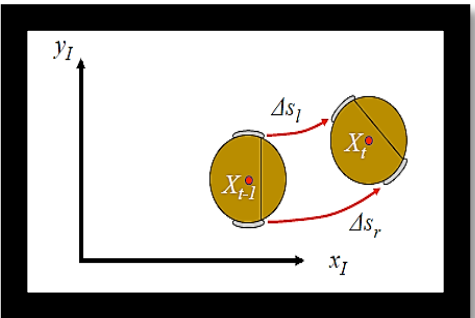
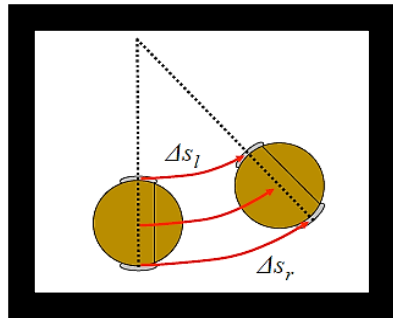
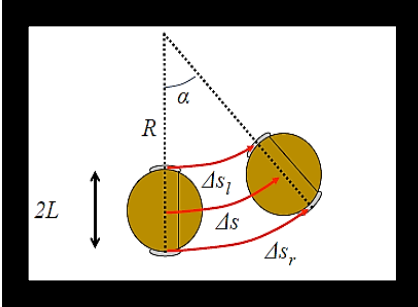
The most common type of incremental encoder uses two output channels (A and B) to estimate velocity and position as shown in figure II-3. Using two code tracks with sectors positioned  $90^\circ$  out of phase; the two output channels of the quadrature encoder indicate both position and direction of rotation. If A leads B, for example, the disk is rotating in a clockwise direction. If B leads A,

then the disk is rotating in a counter-clockwise direction. Therefore, by monitoring both the number of pulses and the relative phase of signals A and B, you can track both the position and direction of rotation.

In addition, some quadrature detectors include a third output channel, called a zero or reference signal, which supplies a single pulse per revolution. This single pulse can be used for precise determination of a reference position.

### II.3.3. Modeling motion

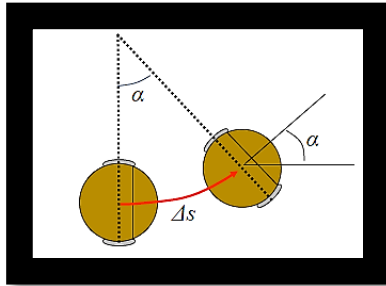
If a robot starts from a position  $X_{t-1}$ , and the right and left wheels move respective distances  $\Delta s_r$  and  $\Delta s_l$ , what is the resulting new position  $X_t$ ? The calculation of  $X_t$  require eight (08) steps

<p>1)</p>  <p>To start, let's model the change in angle <math>\Delta\theta</math> and distance travelled <math>\Delta s</math> by the robot.</p> <p>Assume the robot is travelling on a circular arc of constant radius.</p>	<p>2)</p>  <p>Begin by noting the following holds for circular arcs:</p> $\Delta s_l = R\alpha$ $\Delta s_r = (R + 2L)\alpha$ $\Delta s = (R + L)\alpha$
<p>3)</p>  <p>Now manipulate first two equations:  <math>\Delta s_l = R\alpha</math>, <math>\Delta s_r = (R + 2L)\alpha</math></p>	<p>4) Or, note the distance of the travelled center is simply the average distance of each wheel:</p> $\Delta s = (\Delta s_r + \Delta s_l) / 2$ <p>To calculate the change in angle <math>\Delta\theta</math>, observe that it equals the rotation about the circular arc's center point <math>\Delta\theta = \alpha</math>.</p>



To:  $R\alpha = \Delta s_l$   
 $L\alpha = (\Delta s_r - R\alpha)/2 = \Delta s_r / 2 - \Delta s_l / 2$   
 Substitute this into last equation for  $\Delta s$ :  
 $\Delta s = (R+L)\alpha = R\alpha + L\alpha = \Delta s_l + \Delta s_r / 2 - \Delta s_l / 2$   
 $= \Delta s_l / 2 + \Delta s_r / 2 = (\Delta s_l + \Delta s_r) / 2$

5)



So we solve for  $\alpha$  by equating  $\alpha$  from the first two equations:

$$\Delta s_l = R\alpha$$

$$\Delta s_r = (R+2L)\alpha$$

This results in:

$$\Delta s_l / R = \Delta s_r / (R+2L)$$

$$(R+2L) \Delta s_l = R \Delta s_r$$

$$2L \Delta s_l = R (\Delta s_r - \Delta s_l)$$

Substitute  $R$  into

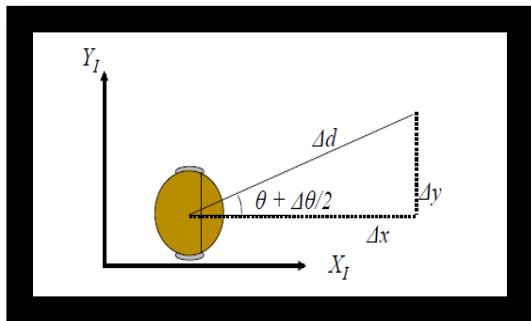
$$\alpha = \Delta s_l / R = \Delta s_l (\Delta s_r - \Delta s_l) / (2L \Delta s_l) = (\Delta s_r - \Delta s_l) / 2L$$

$$\text{So... } \Delta\theta = (\Delta s_r - \Delta s_l) / 2L$$

7) Using Trig:

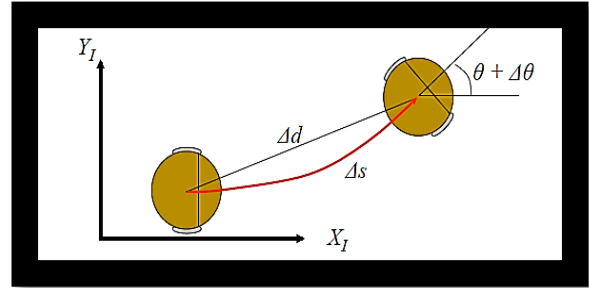
$$\Delta x = \Delta d \cos(\theta + \Delta\theta/2)$$

$$\Delta y = \Delta d \sin(\theta + \Delta\theta/2)$$

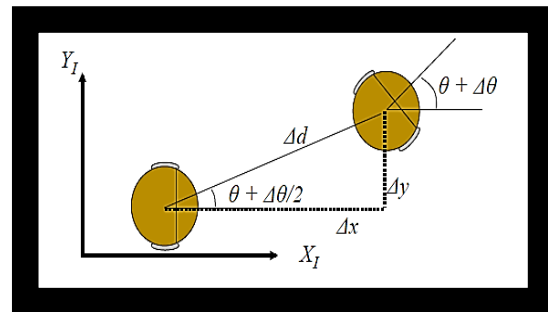


6) Now that we have  $\Delta\theta$  and  $\Delta s$ , we can calculate the position change in global coordinates.

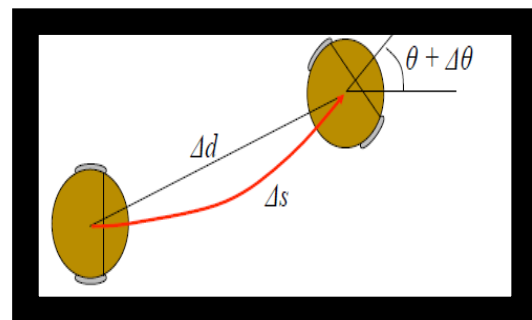
We use a new segment of length  $\Delta d$ .



Now calculate the change in position as a function of  $\Delta d$ .



8) Now if we assume that the motion is small, then we can assume that  $\Delta d \approx \Delta s$ :



So...

$$\Delta x = \Delta s \cos(\theta + \Delta\theta/2)$$

$$\Delta y = \Delta s \sin(\theta + \Delta\theta/2)$$

Summary:

$$\Delta x = \Delta s \cos(\theta + \frac{\Delta\theta}{2}) \quad (\text{II.1})$$

$$\Delta y = \Delta s \sin(\theta + \frac{\Delta\theta}{2}) \quad (\text{II.2})$$

$$\Delta\theta = \Delta s_r - \Delta s_l / b \quad (\text{II.3})$$

$$\Delta s = \Delta s_r + \Delta s_l / 2 \quad (\text{II.4})$$

$$p' = f(x, y, \theta, \Delta s_r, \Delta s_l) = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \begin{bmatrix} ((\Delta s_r + \Delta s_l) / 2) \cos(\theta + (\frac{\Delta s_r - \Delta s_l}{2b})) \\ ((\Delta s_r + \Delta s_l) / 2) \sin(\theta + (\frac{\Delta s_r - \Delta s_l}{2b})) \\ (\Delta s_r - \Delta s_l) / b \end{bmatrix} \quad (\text{II.5})$$

## II.4. Motor efficiency

With the great advancements in mobile robot applications, energy concerns have increased. Most of researches and works done in the field of robotics are developed without considering the robot life duration. Especially, when mobile robots use batteries to power themselves, so, the lifespan” of these robots is limited. The investigations on energy-related concepts are also of great importance for motors efficiency. Motion control and planning of robot have, of course, relation to energy consumption. In the following section, we will highlight the effect of energy consumption on the efficiency of Mobile Robots navigation.

### II.4.1. Energy efficiency in mobile robots

Generally, to evaluate any machine performance, the question is raised to ask about how efficient that machine is. Efficiency is a measure to show, roughly, how much input is utilized to produce the output. We can say that efficiency is formulated as:

$$\text{Efficiency} = \text{Output}/\text{Input} \quad (\text{II.6})$$

So, in the process of designing the machine and its operation, the objective is to have the most transfer of input to output. In other word, the goal is to maximize Efficiency. So, better performances are attained by increasing the ratio *Output/Input* .

Mobility feature requires independent power source rather than fixed sources. So in most of cases, batteries are used to power mobile robots. Other sources can be fuel, for example, for powering

autonomous cars. However, generally, batteries are the main sources of energy safe powering of mobile robots.

It is obvious that batteries have finite limit. Energy stored in a battery is depleted with rate related to the consumption of the device equipped with. So, energy limit should be considered carefully when designing the motion of the mobile robot. Below are some examples of energy efficiency applications in different situations:

- Disasters: mobile robots are distributed to find survivors and maybe also rescue them.
- War zones: unmanned mobile robots are deployed to combat the enemies.
- Social: robots could be responsible of cleaning the floor or assisting people.
- Also imagine the cost of replacing or recharging the battery.

Above situations shows typical scenarios of mobile robots operations. In critical situations of war or disaster, energy management should be optimal in order to elongate lifespan of the battery. Cost of replacing or recharging batteries also show us that proper energy utilization is of great importance. Minimization cost is crucial in designing robot operation. So, from above explanations, we can see that the main goal of robot design is to respect the energy consumption constraint. To have clear view of energy efficiency in mobile robots, we can make a new definition:

$$\textit{Efficiency} = \textit{OutputTask} / \textit{EnergyConsumption} \quad (\text{II.7})$$

“Output Task” can be the distance travelled by the robot, operation time, coverage area of the robots. In other words, these are the objectives put by the developers and designers. Energy Consumption obviously means how the battery power is utilized to power different components of the robot. So, to increase the efficiency of our machine, i.e. the robot, we have to get the maximum tasks accomplishments and in the same time the minimum consumption of energy. As a result, the ultimate goal of energy-efficient motion control of mobile robots is to minimize energy consumption.

## **II.4.2. Energy consumption in mobile robots**

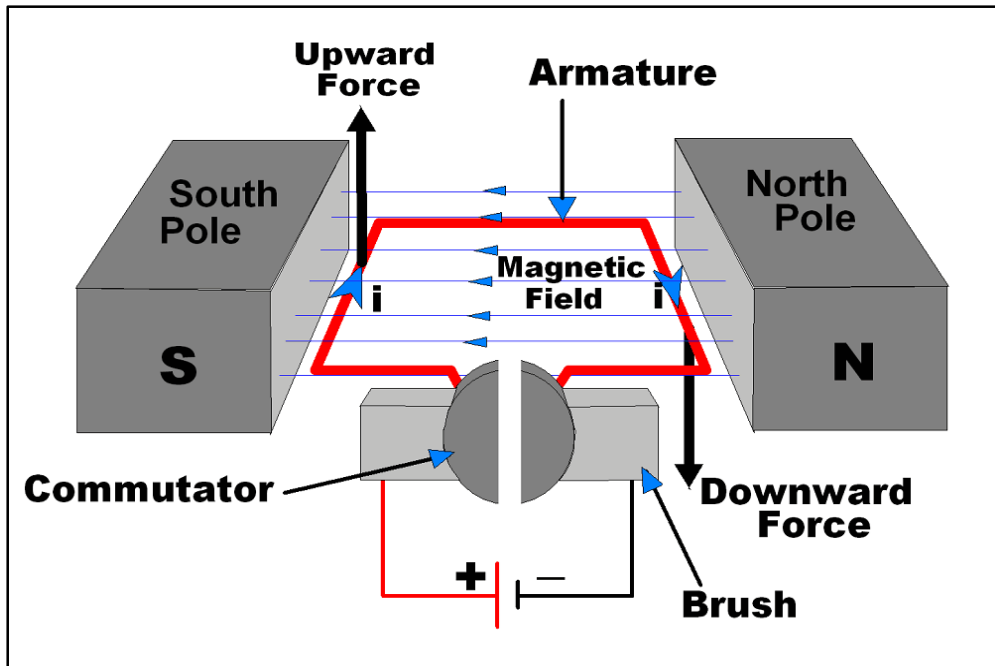
### **II.4.2.1. Motors Energy Consumption**

Here, we will discuss how robot actuators, mostly motors, are consuming energy. Mobile robots in small-scale usually use DC motors as actuators acting on the environment [83]. In general, mobile robots move in their environment via the equipped wheels. Rotation of the wheels is controlled by DC motors. Therefore, motion is directly depending on the DC motors. A typical DC motor is current or voltage supplied depending on the circuitry accompanying the rotary body. So battery consumption is related to “consumption” of the electrical signal required. A DC motor being a physical system, energy loss appears due to, motion, friction or load Inertia or simply due to power consumption of the electronic circuit driving the motor.

However, when a motor is operated at variable load and/or varying speed, its efficiency can drop far below this value [84]. We can define three types of losses:

- Resistive losses: These losses are due to current flowing through the windings, leading to power being dissipated. They depend only on the torque,
- Damping losses: These are the losses that are related to speed only. They represent friction losses in the system (e.g. bearings, graphite brushes. . .),
- Combined losses: These losses include all remaining terms. Usually, combined losses are insignificant with respect to resistive and damping losses.

Figure II-4 shows a schematic of Brushed DC Motor.



**Figure II-4: Schematic of Typical DC Motor**

For some motors, power consumption increases linearly with velocity. In the other hand, others will behave exponentially for high velocities. The power model of a motor depends on motor design and electronic circuit driving the motor. To solve the problem of energy loss in motors, robust and optimal controller should be considered for navigation.

#### **II.4.2.2. Auxiliary Sources of Energy Consumption**

Here in this small part, just a review of other sources of energy consumption will be given. These sources are either minor or unrelated to motion of the robot. In any mobile robot, in addition to actuators, other electrical components exist. Generally, two kinds of components can also consume energy: the sensing elements and robot brain (the microcontrollers and/or the onboard computer).

- Sensors: the sensing elements are essential parts of any robot. Sensors are devices that acquire observations from the surrounding environment. Those could include basic proximity sensors: ultrasonic, laser or infrared based. Vision system, i.e. camera, also is a common sensing system. Sensors consume energy in proportion with the rate of observations. So, generally when considering sensors energy, the decision variable would be the rate of sensing (e.g. frame rate in a camera). A linear relation can sufficiently model energy versus rate.

- Microcontrollers & Computer: these components are responsible of controlling all processes of the robot. Energy consumption of computers depends on the execution of the program. However, the specific relation to consumption is complex. Complexity is due to the inherent complexity of how microprocessors handle algorithms and programs. So, when considered, it is either dropped or included as constant value.

## **II.5. Wheel slip**

It is the forces of tire / road contact that allow the vehicle to move forward brake and "hold the road". It is important for correct and realistic modeling of the dynamic behavior of the vehicle to study, understand and model these efforts in detail.

Within the framework of this study, an important part of the work is devoted to the modeling of the tire / road contact.

### **II.5.1. General information on pneumatic / pavement contact**

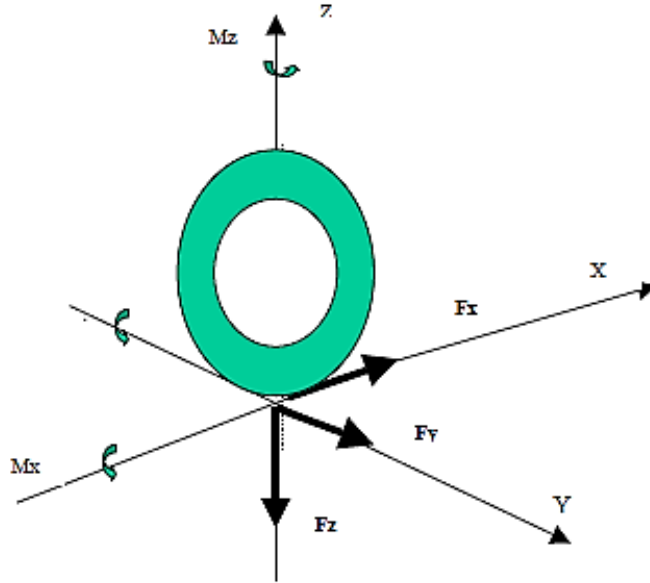
The tire is the interface element between the vehicle and the road. Its ability to transmit forces is essential for good handling. Its damping qualities also constitute a first vibration isolation device against the stresses generated by the road.

In operation, a tire is subjected to several types of force (figure II-5):

- $F_x$ : the longitudinal force developed between the road and the wheel in the X direction,
- $F_y$ : the lateral force developed between the road and the wheel in the Y direction,
- $F_z$ : the normal force on the wheel in Z direction.

A tire is also subjected to three moments, one around each axis:

- $M_x$ : the payment moment around the X-axis,
- $M_y$ : the moment of rolling resistance around the Y-axis,
- $M_z$ : the moment of self-alignment around the Z-axis.



**Figure II-5: Forces and moments on a wheel**

The two main variables, which characterize these forces and moments, are the longitudinal and lateral slipping tire behavior.

### **II.5.2. Longitudinal behavior**

The longitudinal slip rate influences the longitudinal behavior; it is defined as follows:

When an engine torque (acceleration) is applied to a wheel turning on a solid surface, a tensile force appears at the contact surface of the wheel with the road. This surface, subjected to compression forces is deformed, which has the effect of increasing its perimeter. Consequently, the horizontal displacement of the point of contact between the wheel and the road becomes slower. This phenomenon called longitudinal sliding is defined by the following equation:

$$ix\% = \frac{(r\omega - v_r)}{r\omega} \times 100\% \quad (\text{II.8})$$

When

$v_r$  is the linear velocity of translation of the contact point between the wheel and the road,

$\omega$  is the angular velocity of the wheel rotation,

$r$  is the rolling radius when a wheel rolls freely.

The longitudinal forces are directly linked to the direction of travelling, the traveled distance, speed and acceleration. The main physical variable to adjust these forces is longitudinal slipping ( $ix\%$ ).

### II.5.3. Lateral behavior

Lateral sliding (the drift) plays an important role in the lateral behavior of the wheel. It is mainly caused by forces acting on the wheel when entering a turn. They directly influence the directional control of the vehicle.

#### II.5.3.1. The drift angle:

When a rotating wheel is subjected to a lateral force  $F_s$  (Figure II-5), it appears that the contact surface of the tire with the road slides moves following an opposite direction to this force. This lateral force can be linked to a lateral acceleration experienced by the vehicle, for example, during the cornering action. The deformation of the contact surface creates an angle between the longitudinal axis of the wheel and the direction of its movement. This angle  $\alpha$ , called the tire drift angle, is mainly due to the elastic properties of the tire in the lateral direction. As a reaction to these deformations, the tire develops a lateral force  $F_y$  that is a function of the angle of drift (Fig.II-6).

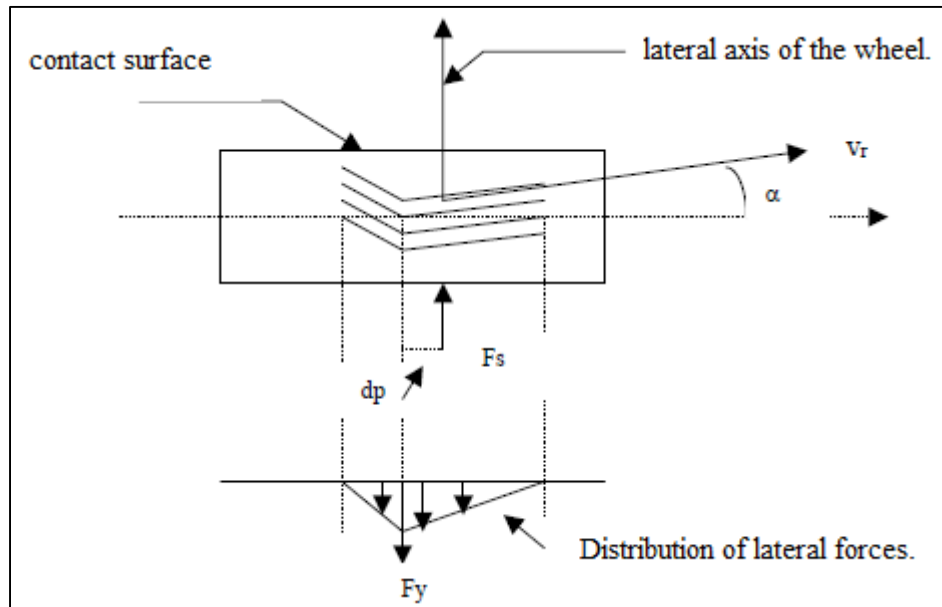


Figure II-6: Lateral deformation of the tire during cornering action



#### II.5.4. Dynamic coupled behavior: lateral - Longitudinal

When a tire is subjected to lateral and longitudinal forces simultaneously, the longitudinal force tends to reduce the lateral force and vice-versa. The reduction in such forces is due, first, to the reduction in the elastic properties of the tire in both directions (longitudinal stiffness and lateral stiffness) and, second, to the decrease in adhesion properties in both directions (longitudinal grip coefficient and lateral grip coefficient).

##### *Lateral slipping and longitudinal slipping:*

In the case where the coupled behavior is taken into account, the lateral sliding and the longitudinal sliding are defined as follows:

Longitudinal sliding is given by the following expression:

$$ix = \frac{r\omega - v_r \cos \alpha}{r\omega} \quad (\text{II.9})$$

$v_r \cos \alpha$  is the component of the linear velocity of the contact point along the longitudinal axis of the wheel.

The lateral slip, defined as the ratio between the component of the linear velocity of the contact point along the lateral axis of the wheel and the linear velocity of the wheel  $r\omega$ . The lateral slip is given by the following expression:

$$iy = \frac{v_r \sin \alpha}{r\omega} \quad (\text{II.10})$$

Thus, a resulting slip can be defined according to the direction of the drift angle  $\alpha$  given by:

$$i = \sqrt{ix^2 + iy^2} \quad (\text{II.11})$$

#### II.5.5. Pneumatic / pavement contact models

Dealing with vehicle dynamics, the tire model has a great importance for moving vehicle behavior investigation. It is the only link between the vehicle chassis and the road. Several models of the tire are available in the literature.

Two major approaches are adopted: empirical methods and theoretical methods. Figure II-7 depicts the various differences of tire modeling (taken from [85]).

Six parameters are considered in [85]:

- Degree of fit: model's accuracy with respect to the design objective,
- Number of full scale tests: amount of tests required to validate the model,
- Complexity of formulations,
- Effort required to design the model,
- Insight in tire behavior: model's ability of predicting the tire behavior,
- Number of special experiments: amount of experiments required to develop the model.

As expected, empirical approaches require an important number of full-scale tests with respect to theoretical approaches. These latter are mainly based on the tire's physical structure theory. Four categories are distinguished in [85]: "from experimental data only", "using similarity method", "through simple physical model", "through complex physical model".

The first category at the extreme left (figure II.7) uses regression procedures to develop mathematical formulations whose parameters fit best the measured data. A well-known empirical model is the Magic Formula [85]. This model provides an excellent fit for tire's efforts curves, which makes it more suitable for vehicle motion simulations. The similarity approach uses simple distortion and re-scaling methods to develop simpler empirical models. This method is particularly useful when fast computations are needed [85]. A good example would be Dugoff's model, which uses a simpler representation of tire deformation while keeping a good representation of combined slip [86]. However, these two categories provide less insight in tire behavior. The relatively simple physical models of the third category are more useful to get better understandings about the tire behavior. In this context, the "brush model" represents a good illustration [85], [87], [88]. Regarding the fourth category, interest is given to tire performances related to its construction. More detailed analysis is required and complex finite element based models are usually adopted [85].

On one hand, empirical models rely on experimental measures to make simulation more accurate, and on the other hand, theoretical models rely on physical models to give more insight about the

tire behavior and improve its construction. No model, however, is designed for control synthesis. Physical models are too complex to be implemented in real-time operations, and empirical models use numerous parameters with poor physical meaning, which make them hard to measure in real time.

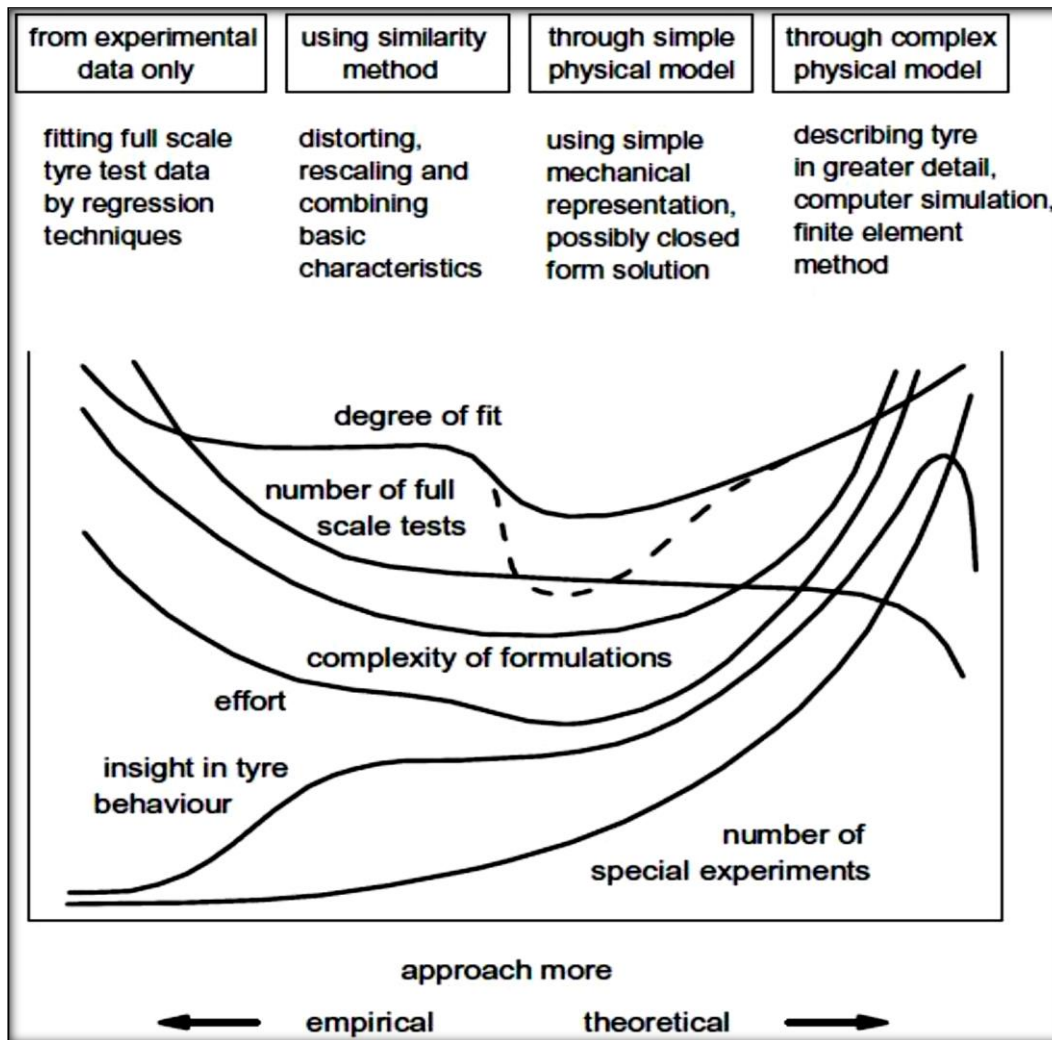


Figure II-7 : Possible Categories of Tire Modeling Approaches [85].

### II.5.5.1. Wheel Slip Modeling

Based on the physical description of a tire as commonly used in the field of automotive engineering [89], a linear tire model with 1st-order dynamics is applied. Note that longitudinal and lateral slip and the resulting forces are assumed independent, i.e. combined slip effects are ignored.

$$f_{lat} = \beta \frac{\delta}{|\rho|} \quad (\text{II.12})$$

$$f_{long} = \alpha \frac{\xi}{|\rho|} \quad (\text{II.13})$$

Where  $\alpha > 0$  and  $\beta < 0$  are constants.

With  $\xi_r, \xi_l$  and  $\delta_r, \delta_l$  as the longitudinal and lateral slip displacement for the right and left wheel respectively.

Moreover, the total longitudinal displacement of the wheel center is defined as:

$$\begin{cases} \rho_r = R\varphi_r + \xi_r \\ \rho_l = R\varphi_l + \xi_l \end{cases} \quad (\text{II.14})$$

## II.6. Conclusion

To cope with the uncertainties that can affect the mobile robot, several types of robust and adaptive controls have been implemented. A suitable controller for mobile robot navigation requires a thorough analysis of the robustness of the robotic system. In this chapter, we have described some types of faults and uncertainties encountered in the wheeled mobile robot used in our work. (modeling error, localization error, motor efficiency and wheel slip).

**CHAPTER III:**  
**SYSTEM MODELING**

## **Chapter III. SYSTEM MODELING**

### **III.1. Introduction**

This chapter deals with motion modeling of differential drive wheeled mobile robot. First, we are interested in Kinematic model that describes geometric relationships that are present in the system by mathematics of robot motion without considering its causes, such as forces or torques. It describes the relationship between input (control) parameters and behavior of a system given by state-space representation. A kinematic model describes system velocities and is presented by a set of differential first-order equations. Second, a dynamic model describes a system motion when forces are applied to the system. This model includes physics of motion where forces, energies, system mass, inertia, and velocity parameters are used. Descriptions of dynamic model are given by second order differential equations.

In the case of a DDWMR, the contact with the environment occurs at the contact point between the wheel outer surface and the ground surface. The interaction between these two surfaces has a significant influence over the dynamic motion of the system, and hence, need to be properly modeled.

In general, there are two major methods for deriving the dynamic equations of mechanical systems namely Newton's method that is directly related to Newton's 2nd law and Lagrange's method that has its root in the classical work of d'Alembert and Lagrange on analytical mechanics. The main difference between the two methods is in dealing with constraint equations. While Newton's method treats each rigid body separately and explicitly models the constraints through the reactive force required to enforce them, Lagrange's provides systematic procedures for eliminating the constraints from the dynamic equations, typically yielding simpler system equations. Thus, it is not surprising that the majority of the conventional WMR models we found in the literature were developed using Lagrange's method. In this chapter, we formulate the dynamic equation of the conventional WMR using Lagrange's method. The Lagrange formulation is presented to provide theoretical background of a non-holonomic DDWMR model with and without slip dynamics.

### **III.2. Coordinates system**

The DDWMR shown in figure III-1 is a typical example of a non holonomic mechanical system. It consists of a vehicle with two driving wheels mounted on the same axis, and a front free

wheel. The motion and orientation are achieved by independent actuators, e.g., DC motors providing the necessary torques to the rear wheels.

$R$  is the radius of each wheel,  $d$  is the distance between the center of mass (point  $D$ ) and mid-point of the axis center of driving wheels (point  $A$ ),  $L$  is each wheel distance to point  $A$ ,  $\dot{\phi}_R$  and  $\dot{\phi}_L$  are the right and left wheel angular velocities respectively.  $C$  is the distance between point  $A$  and instantaneous center of curvature ( $ICC$ ).

At first, two different coordinate systems (frames) were designated: The Inertial Coordinate System: it is considered as the reference frame and is denoted as  $[X_I, Y_I]$ . The Robot Coordinate System (Body frame): is a local frame attached to the DDWMR, and thus, moving with it. This frame is denoted as  $[X_r, Y_r]$ . The two defined frames are shown in Figure III-1.

As shown in Figure III-1, the robot position and orientation in the Inertial Frame is completely specified by the vector

$$q = \begin{bmatrix} x_c \\ y_c \\ \theta \end{bmatrix} \quad (III.1)$$

Where  $x_c, y_c$  are the coordinates of the center of mass of the vehicle, and  $\theta$  is the orientation of the basis  $\{c, x_c, y_c\}$  with respect to the inertial basis.

The important issue that needs to be explained at this stage is the transformation 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:  $X^r = \begin{bmatrix} x^r \\ y^r \\ \theta^r \end{bmatrix}$  and  $X^I = \begin{bmatrix} x^I \\ y^I \\ \theta^I \end{bmatrix}$ .

Then, the transformation from robot frame to the inertial frame can be given as follows:

$$X^I = R(\theta)X^r \quad (III.2)$$

Where  $R(\theta)$  is the orthogonal rotation matrix :

$$R(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (\text{III.3})$$

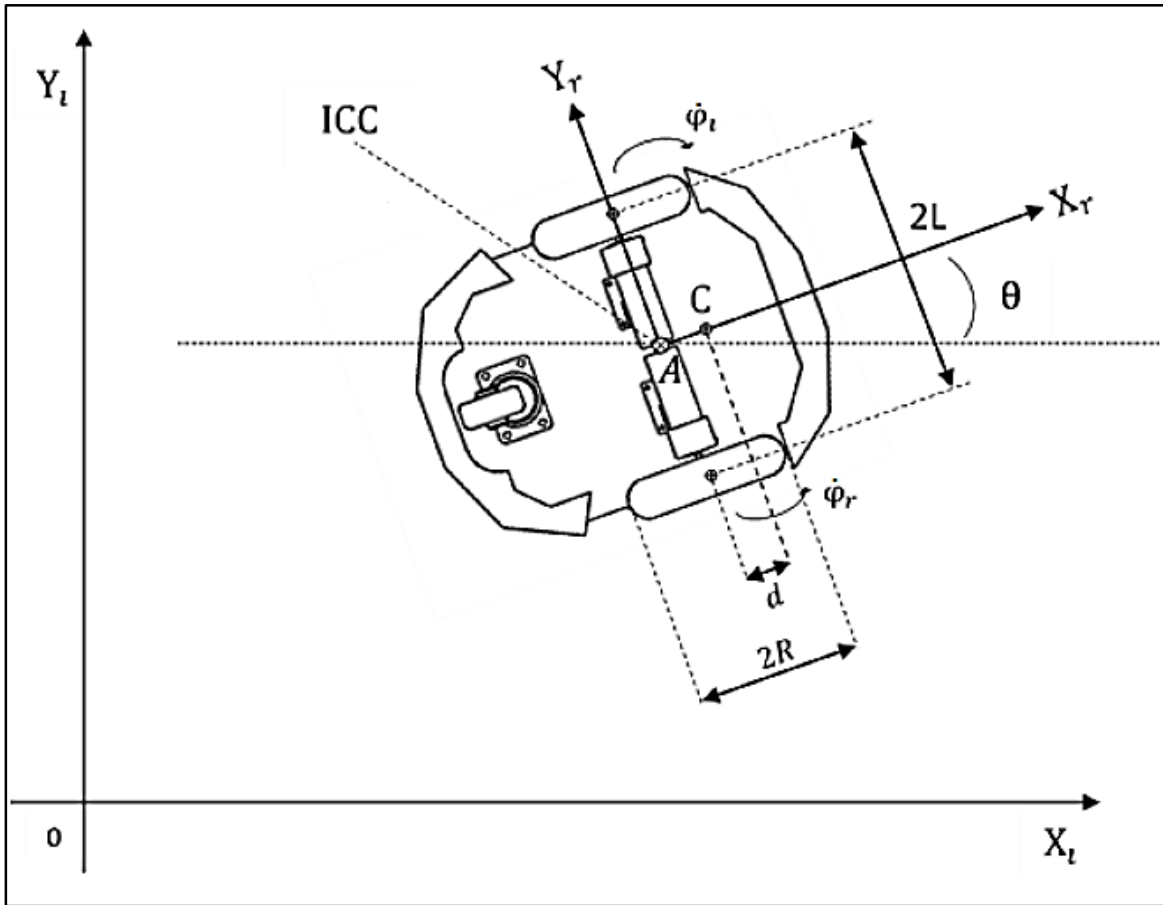


Figure III-1: Differential Drive Wheeled Mobile Robot (DDWMR) [90]

### III.2.1. Kinematic constraints of the Differential-Drive Robot

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

1. 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 along the lateral axis is zero:

$$\dot{y}_A^r = 0 \quad (\text{III.4})$$

The relationship between the speeds of the center of gravity and the center of wheel axis is:



$$\begin{cases} x_C = x_A + d \cos \theta \\ y_C = y_A + d \sin \theta \end{cases} \quad (\text{III.5})$$

$$\begin{cases} \dot{x}_C = \dot{x}_A - d\dot{\theta} \sin \theta \\ \dot{y}_C = \dot{y}_A + d\dot{\theta} \cos \theta \end{cases} \quad (\text{III.6})$$

Using the orthogonal rotation matrix  $R(\theta)$ , the robot velocity in the inertial frame can be given by:

$$\begin{bmatrix} \dot{x}_C^I \\ \dot{y}_C^I \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x}_A^r \\ \dot{y}_A^r = 0 \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} -d \dot{\theta} \sin \theta \\ d \dot{\theta} \cos \theta \\ 0 \end{bmatrix}$$

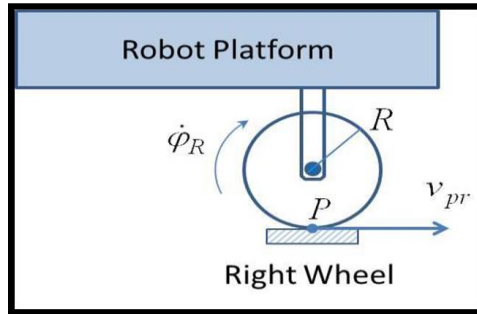
With the non-holonomy equation

$$-\dot{x}_C \sin \theta + \dot{y}_C \cos \theta - d\dot{\theta} = 0 \quad (\text{III.7})$$

### III.2.2. 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 figure III-2. There is no slipping of the wheel in its longitudinal axis ( $x_r$ ) and no skidding in its orthogonal axis ( $y_r$ ). Position and velocity of the contact points in the robot frame are related to the wheel velocities by (III.8) and (III.9) respectively:

$$\begin{cases} v_{pR} = R\dot{\phi}_R \\ v_{pL} = R\dot{\phi}_L \end{cases} \quad (\text{III.8})$$



**Figure III-2: Pure rolling motion constraint.**

$$\begin{bmatrix} x_{pR}^r \\ y_{pR}^r \end{bmatrix} = \begin{bmatrix} 0 \\ -L \end{bmatrix} \quad (\text{III.9})$$

Using the rotation matrix  $R(\theta)$ :

$$\begin{bmatrix} x_{pR}^I \\ y_{pR}^I \end{bmatrix} = \begin{bmatrix} x_A \\ y_A \end{bmatrix} + R(\theta) \begin{bmatrix} 0 \\ -L \end{bmatrix} = \begin{bmatrix} x_A + L \sin \theta \\ y_A - L \cos \theta \end{bmatrix} \quad (\text{III.10})$$

Likewise:  $\begin{bmatrix} x_{pL}^r \\ y_{pL}^r \end{bmatrix} = \begin{bmatrix} 0 \\ L \end{bmatrix}$  (III.11)

$$\begin{bmatrix} x_{pL}^I \\ y_{pL}^I \end{bmatrix} = \begin{bmatrix} x_A \\ y_A \end{bmatrix} + R(\theta) \begin{bmatrix} 0 \\ L \end{bmatrix} = \begin{bmatrix} x_A - L \sin \theta \\ y_A + L \cos \theta \end{bmatrix} \quad (\text{III.12})$$

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

$$\begin{bmatrix} \dot{x}_{pR}^I \\ \dot{y}_{pR}^I \end{bmatrix} = \begin{bmatrix} \dot{x}_A + L \dot{\theta} \cos \theta \\ \dot{y}_A + L \dot{\theta} \sin \theta \end{bmatrix} \quad (\text{III.13})$$

$$\begin{bmatrix} \dot{x}_{pL}^I \\ \dot{y}_{pL}^I \end{bmatrix} = \begin{bmatrix} \dot{x}_A - L \dot{\theta} \cos \theta \\ \dot{y}_A - L \dot{\theta} \sin \theta \end{bmatrix} \quad (\text{III.14})$$

Moreover,

$$\begin{cases} v_{pR} = \dot{x}_{pR} \cos \theta + \dot{y}_{pR} \sin \theta = R \dot{\phi}_R \\ v_{pL} = \dot{x}_{pL} \cos \theta + \dot{y}_{pL} \sin \theta = R \dot{\phi}_L \end{cases} \quad (\text{III.15})$$

Replacing (III.13) and (III.14) in (III.15):

$$\begin{cases} \dot{x}_a \cos \theta + \dot{y}_a \sin \theta + L \dot{\theta} = R \dot{\phi}_R \\ \dot{x}_a \cos \theta + \dot{y}_a \sin \theta - L \dot{\theta} = R \dot{\phi}_L \end{cases} \quad (\text{III.16})$$

Replacing (III.6) in (III.16), rolling constraint equations are formulated as follows:

$$\begin{cases} \dot{x}_C \cos \theta + \dot{y}_C \sin \theta + L \dot{\theta} = R \dot{\phi}_R \\ \dot{x}_C \cos \theta + \dot{y}_C \sin \theta - L \dot{\theta} = R \dot{\phi}_L \end{cases} \quad (\text{III.17})$$

Using the contact points velocities from (III.17) and substituting in  $(x_C, y_C)$  the three constraint equations can be written in the following matrix form:

$$\Lambda(q)\dot{q} = 0 \quad (\text{III.18})$$

Where

$$\Lambda(q) = \begin{bmatrix} -\sin \theta & \cos \theta & -d & 0 & 0 \\ \cos \theta & \sin \theta & L & -R & 0 \\ \cos \theta & \sin \theta & -L & 0 & -R \end{bmatrix} \quad (\text{III.19})$$

And

$$\dot{q} = [\dot{x}_C \quad \dot{y}_C \quad \dot{\theta} \quad \dot{\phi}_R \quad \dot{\phi}_L]^r \quad (\text{III.20})$$

$$\begin{cases} v_R = R\dot{\phi}_R \\ v_L = R\dot{\phi}_L \end{cases} \quad (\text{III.21})$$

Linear velocities (III.21), the constraints matrix  $\Lambda(q)$ (III.19) and the generalized coordinates  $q$  (III.20) will be used in the next section for the DDWMR dynamic modeling.

### III.3. DDWMR Kinematic model

Kinematic is the study of the motion without considering the forces. The purpose of the kinematic modelling is to derive robot velocities as a function of the driving wheels velocities in predefined constraints. Robot's wheels have same angular speed according to the instantaneous curvature center. So the right wheel and the left wheel velocity relation can be obtained as below [91]

$$\omega(ICC + L) = v_L, \omega(ICC - L) = v_R \rightarrow ICC = L(v_R + v_L)/(v_R - v_L) \quad (\text{III.22})$$

The linear and the angular velocity of the robot as follow, respectively;

$$v = \omega * ICC = (v_R + v_L)/2 = R(\dot{\phi}_R + \dot{\phi}_L)/2 \quad (\text{III.23})$$

$$\dot{\theta} = \omega = (v_R - v_L)/2L = R(\dot{\phi}_R - \dot{\phi}_L)/2L \quad (\text{III.24})$$

From (III.23) and (III.24) and (III.16), we get:

$$\begin{bmatrix} \dot{x}_A \\ \dot{y}_A \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \frac{R}{2} \cos \theta & \frac{R}{2} \cos \theta \\ \frac{R}{2} \sin \theta & \frac{R}{2} \sin \theta \\ \frac{R}{2L} & -\frac{R}{2L} \end{bmatrix} \begin{bmatrix} \dot{\phi}_R \\ \dot{\phi}_L \end{bmatrix} \quad (\text{III.25})$$

Replacing in (III.6), we obtain:

$$\begin{bmatrix} \dot{x}_C \\ \dot{y}_C \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & -d \sin \theta \\ \sin \theta & d \cos \theta \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (\text{III.26})$$

Replacing (III.23) and (III.24) in (III.26), Robot's velocity according to the center of mass C of the inertial coordinate system is given by:

$$\begin{bmatrix} \dot{x}_c \\ \dot{y}_c \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \left(\frac{R}{2}\right) \left(\cos \theta - \left(\frac{d}{L}\right) \sin \theta\right) & \left(\frac{R}{2}\right) \left(\cos \theta + \left(\frac{d}{L}\right) \sin \theta\right) \\ \left(\frac{R}{2}\right) \left(\sin \theta + \left(\frac{d}{L}\right) \cos \theta\right) & \left(\frac{R}{2}\right) \left(\sin \theta - \left(\frac{d}{L}\right) \cos \theta\right) \\ \frac{R}{2L} & -\frac{R}{2L} \end{bmatrix} \begin{bmatrix} \dot{\phi}_R \\ \dot{\phi}_L \end{bmatrix} \quad (\text{III.27})$$

Equation (III.27) represents the forward kinematic model of the DDWMR.

#### III.4. Dynamic modeling of the DDWMR without wheel slips

Unlike kinematics where the forces are not taken into consideration, dynamics is the study of the motion of a mechanical system taking into consideration the different forces that affect its motion. The dynamic model is very important for DDWMR simulation as well as for control and navigation algorithms design.

The most popular approaches to design the dynamic model of DDWMR, are either the Lagrangian approach [92] [93] or the Newton-Euler approach [94] [95]. Other formalisms such as the Kane's method have been also suggested as viable approaches of DDMR modeling [96]. In the Newton Euler method, one has to take into account two kinds of forces applied to a system: the given forces and the constraint forces. The given forces include the externally impressed forces by the actuators while the constraint forces are the forces of interaction between the robot platform and ground through wheels. Moreover, in a system with interconnected elements, the components may interact with each other through gears, springs, and frictional elements. Therefore, we need to take into

account all of these forces. It is clear that the Newtonian approach includes a few practical difficulties since in most cases these forces are not easily quantifiable.

The methodology developed by Lagrange overcomes these problems by expressing the forces in terms of system energies, i.e., the kinetic energy and the potential energy, which are scalar quantities easily expressible in terms of system coordinates. The derivation of the Lagrange equations requires also that the generalized coordinates be independent. This will be the method developed in this thesis.

A non-holonomic DDWMR with  $n$  generalized coordinates  $(q_1, q_2, \dots, q_n)$  and subject to  $m$  constraints can be described by the following equations of motion:

$$M(q)\ddot{q} + V(q, \dot{q})\dot{q} + F(\dot{q}) + G(q) + \tau_d = B(q)\tau - \Lambda^T(q)\lambda \quad (\text{III.28})$$

Where:

$M(q)$ : is an  $n \times n$  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,  $\tau_d$  is the vector of bounded unknown disturbances including unstructured unmodeled dynamics,  $B(q)$  is the input matrix,  $\tau$  is the input vector,  $\Lambda^T(q)$  is the matrix associated with the kinematic constraints, and  $\lambda$  is the Lagrange multipliers vector [96].

### **Lagrange dynamic approach**

Lagrange dynamic approach is a very powerful method for formulating the equations of motion of dynamic systems. This method, which was introduced by Lagrange, is used to systematically derive equations of motion by considering kinetic and potential energies of the given system.

The Lagrange equation can be written in the following form:

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} = F - \Lambda^T(q)\lambda \quad (\text{III.29})$$

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,  $\Lambda$  is the constraints matrix, and  $\lambda$  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 DDWMR motion. Furthermore, since the DDWMR is moving in the  $XI-YI$  plane, the potential energy of the DDWMR is considered to be zero.

For the DDWMR, the generalized coordinates are selected as follows:

$$q = [x_c \quad y_c \quad \theta \quad \varphi_R \quad \varphi_L]^T \quad (\text{III.30})$$

The kinetic energies of the DDWMR is the sum of the kinetic energy of the mobile robot without wheels plus the kinetic energies of the wheels and actuators.

The kinetic energy of the robot platform is

$$T_c = \frac{1}{2} m_c v_c^2 + \frac{1}{2} I_c \dot{\theta}^2 \quad (\text{III.31})$$

While the kinetic energy of the right and left wheel is

$$T_{wR} = \frac{1}{2} m_w v_{wR}^2 + \frac{1}{2} I_m \dot{\theta}^2 + \frac{1}{2} I_w \dot{\varphi}_R^2 \quad (\text{III.32})$$

$$T_{wL} = \frac{1}{2} m_w v_{wL}^2 + \frac{1}{2} I_m \dot{\theta}^2 + \frac{1}{2} I_w \dot{\varphi}_L^2 \quad (\text{III.33})$$

where,  $m_c$  is the DDWMR mass without the driving wheels and actuators (DC motors),  $m_w$  is the mass of each driving wheel (with actuator),  $I_c$  is the moment of inertia of the DDWMR following the vertical axis through the center of mass,  $I_w$  and  $I_m$  are the moment of inertia of each driving wheel (with actuator) around the wheel axis, and the moment of inertia of each driving wheel with a motor around the wheel diameter respectively.

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

$$v_i^2 = \dot{x}_i^2 + \dot{y}_i^2 \quad (\text{III.34})$$

The Xi and Yi components of the center of mass and wheels are given by (III.5), (III.10) and (III.12)

We can write:

$$\begin{cases} x_{pR} = x_{wR} \\ y_{pR} = y_{wR} \end{cases}, \quad \begin{cases} x_{pL} = x_{wL} \\ y_{pL} = y_{wL} \end{cases}$$

Using (III.31), (III.32) and (III.33) along with (III.6), (III.13) and (III.14) the total kinetic energy of the DDWMR is:

$$T = \frac{1}{2}m(\dot{x}_c^2 + \dot{y}_c^2) - (m - m_c)d^2\dot{\theta}^2 + \frac{1}{2}I_w(\dot{\phi}_R^2 + \dot{\phi}_L^2) + \frac{1}{2}I\dot{\theta}^2 \quad (\text{III.35})$$

Where the following new parameters are introduced:

$m = m_c + 2m_w$  is the total mass of the robot,  $I = I_c + 2m_w(L^2 + d^2) + 2I_m$  is the total equivalent inertia.

Using (III.29) along with the Lagrangian function,  $L=T$  the equations of motion of the DDWMR are given by

$$m\ddot{x}_c + 2dm_w(\ddot{\theta} \sin \theta + \dot{\theta}^2 \cos \theta) = \lambda_1 \sin \theta - \lambda_2 \cos \theta - \lambda_3 \cos \theta \quad (\text{III.36})$$

$$m\ddot{y}_c - 2dm_w(\ddot{\theta} \cos \theta - \dot{\theta}^2 \sin \theta) = -\lambda_1 \cos \theta - \lambda_2 \sin \theta - \lambda_3 \sin \theta \quad (\text{III.37})$$

$$2dm_w(\ddot{x}_c \sin \theta - \ddot{y}_c \cos \theta) + I\ddot{\theta} = \lambda_1 d - \lambda_2 L + \lambda_3 L \quad (\text{III.38})$$

$$I_w\ddot{\phi}_R = \tau_R + \lambda_2 R \quad (\text{III.39})$$

$$I_w\ddot{\phi}_L = \tau_L + \lambda_3 R \quad (\text{III.40})$$

Now, the obtained equations of motion (III.36) and (III.40) can be represented in the general form given by (III.28) as follows:

$$M(q)\ddot{q} + V(q, \dot{q})\dot{q} = B(q)\tau - \Lambda^T(q)\lambda \quad (\text{III.41})$$

Where

$$M(q) = \begin{bmatrix} m & 0 & 2dm_w \sin \theta & 0 & 0 \\ 0 & m & -2dm_w \cos \theta & 0 & 0 \\ 2dm_w \sin \theta & -2dm_w \cos \theta & I & 0 & 0 \\ 0 & 0 & 0 & I_w & 0 \\ 0 & 0 & 0 & 0 & I_w \end{bmatrix}, V(q, \dot{q}) = \begin{bmatrix} 0 & 0 & 2dm_w \dot{\theta} \cos \theta & 0 & 0 \\ 0 & 0 & 2dm_w \dot{\theta} \sin \theta & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$B(q) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\Lambda^T(q)\lambda = \begin{bmatrix} -\sin \theta & \cos \theta & \cos \theta \\ \cos \theta & \sin \theta & \sin \theta \\ 0 & L & -L \\ 0 & -R & 0 \\ 0 & 0 & -R \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix}$$

Next, the system described by (III.41) is transformed into an alternative form, which is more convenient for the purpose of control and simulation. The main aim is to eliminate the constraint term  $\Lambda^T(q)\lambda$  in (III.41) since the Lagrange multipliers  $\lambda_i$  are unknown.

This is done first by defining the reduced vector:

$$\eta = \begin{bmatrix} \dot{\phi}_R \\ \dot{\phi}_L \end{bmatrix} \quad (\text{III.42})$$

Next, by expressing the generalized coordinates velocities using the forward kinematic model given by (III.27). Then we can obtain:

$$\begin{bmatrix} \dot{x}_c \\ \dot{y}_c \\ \dot{\theta} \\ \dot{\phi}_R \\ \dot{\phi}_L \end{bmatrix} = \frac{1}{2} \begin{bmatrix} R(\cos \theta - \left(\frac{d}{L}\right) \sin \theta) & R(\cos \theta + \left(\frac{d}{L}\right) \sin \theta) \\ R(\sin \theta + \left(\frac{d}{L}\right) \cos \theta) & R(\sin \theta - \left(\frac{d}{L}\right) \cos \theta) \\ \frac{R}{L} & -\frac{R}{L} \\ 0 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} \dot{\phi}_R \\ \dot{\phi}_L \end{bmatrix} \quad (\text{III.43})$$

This can be written in the form

$$\dot{q} = S(q)\eta \quad (\text{III.44})$$



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

$$S^T(q)\Lambda^T(q) = 0 \quad (\text{III.45})$$

Taking the time derivative of (III.44) gives:

$$\ddot{q} = \dot{S}(q)\eta + S(q)\dot{\eta} \quad (\text{III.46})$$

By substituting (III.44) and (III.46) in the main (III.41) we obtain

$$M(q)[\dot{S}(q)\eta + S(q)\dot{\eta}] + V(q, \dot{q})[S(q)\eta] = B(q)\tau - \Lambda^T(q)\lambda \quad (\text{III.47})$$

Next, rearranging the equation and multiplying both sides by matrix  $S(q)$  leads to (III.48).

$$S^T(q)M(q)S(q)\dot{\eta} + S^T(q)[M(q)\dot{S}(q) + V(q, \dot{q})S(q)]\eta = S^T(q)B(q)\tau - S^T(q)\Lambda^T(q)\lambda \quad (\text{III.48})$$

Where the last term is identical to zero (III.45). Now defining the new matrices:

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

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

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

The dynamic equations are reduced to the form

$$\bar{M}(q)\dot{\eta} + \bar{V}(q, \dot{q})\eta = \bar{B}(q)\tau \quad (\text{III.49})$$

Where

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

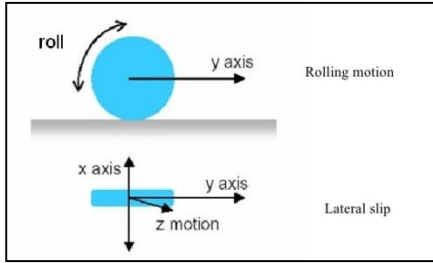
$$\bar{V}(q, \dot{q}) = \begin{bmatrix} 0 & \frac{R^2}{2L} dm_c \dot{\theta} \\ -\frac{R^2}{2L} dm_c \dot{\theta} & 0 \end{bmatrix} \text{ and } \bar{B}(q) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Equation (III.49) shows that the DDWMR dynamics are expressed only as a function of the right and left wheel angular velocities ( $\dot{\phi}_R, \dot{\phi}_L$ ), the robot angular velocity  $\dot{\theta}$  and the driving motor torques ( $\tau_R, \tau_L$ ). The equation of motion (III.49) can be also transformed into an alternative form, which is represented by linear, and angular velocities ( $v, \omega$ ) of the DDWMR. Using the kinematic model equations (III.23) and (III.24), it can be, easily, shown that the motion equation (III.49) can be rearranged in the following compact form:

$$\begin{cases} \left(m + \frac{2I_w}{R^2}\right) \dot{v} - m_c d \omega^2 = \frac{1}{R} (\tau_R + \tau_L) \\ \left(d^2(m_c - 2m_w) + I + \frac{2L^2}{R^2} I_w\right) \dot{\omega} + m_c d \omega v = \frac{L}{R} (\tau_R - \tau_L) \end{cases} \quad (\text{III.50})$$

### III.5. Dynamic modeling of the DDWMR with wheel slips

In this work, we want to investigate the navigation problem of a non-holonomic DDWMR when the ideal no-slip condition does not hold true. Therefore, we want to include wheel slipping into the dynamics model of the system [96]. We start by introducing  $\xi_r, \xi_l$  and  $\delta_r, \delta_l$  as the longitudinal and lateral slip displacement for the right and left wheel respectively (figure III-3).



**Figure III-3: Rolling and lateral slip motions of wheel.**

We then define a new state to represent the total longitudinal displacement of the wheel center as

$$\begin{cases} \rho_r = R\phi_R + \xi_r \\ \rho_l = R\phi_L + \xi_l \end{cases} \quad (\text{III.51})$$

And our new generalized coordinates considering wheel slips become:

$$q = [x_C \quad y_C \quad \theta \quad \delta_r \quad \delta_l \quad \rho_r \quad \rho_l \quad \varphi_R \quad \varphi_L]^T \quad (\text{III.52})$$

The new non-holonomic longitudinal constraints of the robot then become

$$\begin{cases} \dot{\rho}_r = \dot{x}_C \cos \theta + \dot{y}_C \sin \theta + L \dot{\theta} \\ \dot{\rho}_l = \dot{x}_C \cos \theta + \dot{y}_C \sin \theta - L \dot{\theta} \end{cases} \quad (\text{III.53})$$

Since the two driving wheels are rigidly connected with the robot body and do not have relative motion along lateral direction, their lateral slips have to be same  $\delta_r = \delta_l$ , as seen below.

$$\begin{cases} \dot{\delta}_r = -\dot{x}_C \sin \theta + \dot{y}_C \cos \theta - d \dot{\theta} \\ \dot{\delta}_l = -\dot{x}_C \sin \theta + \dot{y}_C \cos \theta - d \dot{\theta} \end{cases} \quad (\text{III.54})$$

Writing above new constraints in the form of (III.18) and we will have matrix as

$$\Lambda(q) = \begin{bmatrix} -\sin \theta & \cos \theta & -d & -1 & 0 & 0 & 0 & 0 & 0 \\ -\sin \theta & \cos \theta & -d & 0 & -1 & 0 & 0 & 0 & 0 \\ \cos \theta & \sin \theta & L & -R & 0 & -1 & 0 & 0 & 0 \\ \cos \theta & \sin \theta & -L & 0 & -R & 0 & -1 & 0 & 0 \end{bmatrix} \quad (\text{III.55})$$

Then we can find matrix S and vector, which fulfill (III.44) and (III.45), as follows

$$S(q) = \begin{bmatrix} -\sin \theta & \frac{L \cos \theta - d \sin \theta}{2L} & \frac{L \cos \theta + d \sin \theta}{2L} & 0 & 0 \\ \cos \theta & \frac{d \cos \theta + L \sin \theta}{2L} & \frac{-d \cos \theta + L \sin \theta}{2L} & 0 & 0 \\ 0 & \frac{1}{2L} & -\frac{1}{2L} & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{III.56})$$

$$\eta(t) = [\dot{\delta}_l \quad \dot{\rho}_r \quad \dot{\rho}_l \quad \dot{\varphi}_R \quad \dot{\varphi}_L]^T \quad (\text{III.57})$$

By applying the methodology developed by Lagrange, under the rolling hypothesis in the presence of slip and the influence of the traction forces on the robot, the equation of motion (III.41) can be rewritten in the following form:

$$M(q)\ddot{q} + V(q, \dot{q})\dot{q} = B(q)\tau + F(\dot{q}) - \Lambda^T(q) \quad (\text{III.58})$$

After taking the proper derivatives, by substituting (III.44) in the dynamic model of the robot (III.58). The equation (III.29) can generate equations of motion that can be written in the following form:

$$M(q)[\dot{S}(q)\eta + S(q)\dot{\eta}] + V(q, \dot{q}) = B(q)\tau + F(\dot{q}) - \Lambda^T(q)\lambda \quad (\text{III.59})$$

Where

$$M(q) = \text{diag}([m_c \quad m_c \quad I_c + 2I_m \quad m_w \quad m_w \quad m_w \quad m_w \quad I_w \quad I_w])$$

$$V = [0 \quad 0 \quad 0 \quad m_w\dot{\theta}\dot{\rho}_r \quad m_w\dot{\theta}\dot{\rho}_l \quad -m_w\dot{\theta}\dot{\delta}_r \quad -m_w\dot{\theta}\dot{\delta}_l \quad 0 \quad 0]^T$$

$$\lambda = [\lambda_1 \quad \lambda_2 \quad \lambda_3 \quad \lambda_4]$$

$$B = [0_{2 \times 7} \quad I_{2 \times 2}]^T$$

$$F(\dot{q}) = [0 \quad 0 \quad 0 \quad f_{latr} \quad f_{latl} \quad f_{longr} \quad f_{longl} \quad -Rf_{longr} \quad -Rf_{longl}]$$

We multiply the equation (III.59) by  $S^T(q)$  as follows

$$S^T(q)M(q)S(q)\dot{\eta} + (S^T(q)M(q)\dot{S}(q))\eta + S^T(q)V(q, \dot{q}) = S^T(q)B(q)\tau + S^T(q)F(\dot{q}) - S^T(q)\Lambda^T(q)\lambda \quad (\text{III.60})$$

Where the last term is identical to zero (III.45), we can simplify the equation (III.60) such as

$$(S^TMS)\dot{\eta}(t) + (S^TMS)\eta(t) + S^TV = S^TB\tau + S^TF \quad (\text{III.61})$$

By writing (III.61) in another form, one can separate the equations of torques as follows:

$$\left\{ \begin{array}{l} (\overline{S^TMS})\dot{\bar{\eta}}(t) + (\overline{S^TMS})\bar{\eta}(t) + \overline{S^TV} = \overline{S^TF} \\ I_w\ddot{\phi}_r = \tau_R - f_{longr}R \\ I_w\ddot{\phi}_l = \tau_L - f_{longl}R \end{array} \right. \quad (\text{III.62})$$

In this work, we assume the traction force can be linearly approximated as follows, [96]

$$f_{lat} = \beta \frac{\dot{\delta}}{|\dot{\rho}|}, f_{long} = \alpha \frac{\dot{\xi}}{|\dot{\rho}|}$$

Where  $\alpha > 0$  and  $\beta < 0$  are constants.

### III.6. Driving wheel DC motor modeling

The DC motors, which are generally used to drive the wheels of a differential drive mobile robot system, are considered the servo actuators. They were used to drive our mobile robot. DC motors can be controlled with voltage, so the following equations are used for simulations [97]

$$e_a = K_b \omega_m \quad (\text{III.63})$$

$$v_a = R_a i_a + L_a \frac{di_a}{dt} + e_a \quad (\text{III.64})$$

$$\tau_m = K_t i_a \quad (\text{III.65})$$

$$\tau_m = J \frac{d\omega_m}{dt} \quad (\text{III.66})$$

$$\tau = N \tau_m \quad (\text{III.67})$$

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 fem,  $\omega_m$  is the rotor angular speed,  $\tau_m$  is the motor torque,  $(K_t, K_b)$  are the torque constant and back fem constant respectively,  $N$  is the gear ratio,  $J$  is moment of inertia of the motor shaft and  $\tau$  is the output torque applied to the wheel.

Since, in the DDWMR, the motors are mechanically coupled to the robot wheels through the gears, the equations of motion of the motors are linked directly with the mechanical dynamics of the DDWMR. Therefore, each DC motor will have:

$$\begin{cases} \omega_{mR} = N \dot{\phi}_R \\ \omega_{mL} = N \dot{\phi}_L \end{cases} \quad (\text{III.68})$$

Setting  $U_a(s) = E_a(s)$  as control input, then the transfer from the control input to the angular speed output as follow can be expressed in Laplace as follows:

$$\Omega(s) = H(s)U_a(s) \quad (\text{III.69})$$

After development, the transfer function of the DC motor can be given by:

$$H(s) = \frac{K_m}{\frac{JT_a}{K_aK_tK_b}s^2 + \frac{J}{K_aK_tK_b}s + 1} \quad (\text{III.70})$$

Where  $K_m = \frac{1}{K_b}$ ,  $T_a = \frac{L_a}{R_a}$ ,  $K_a = \frac{1}{R_a}$

In equation 70, the  $s^2$  coefficient is very small and can be neglected because of the small coefficient  $T_a$  compared to the  $s$  coefficient. Therefore, the transfer function (III.70) can be approximated as follows:

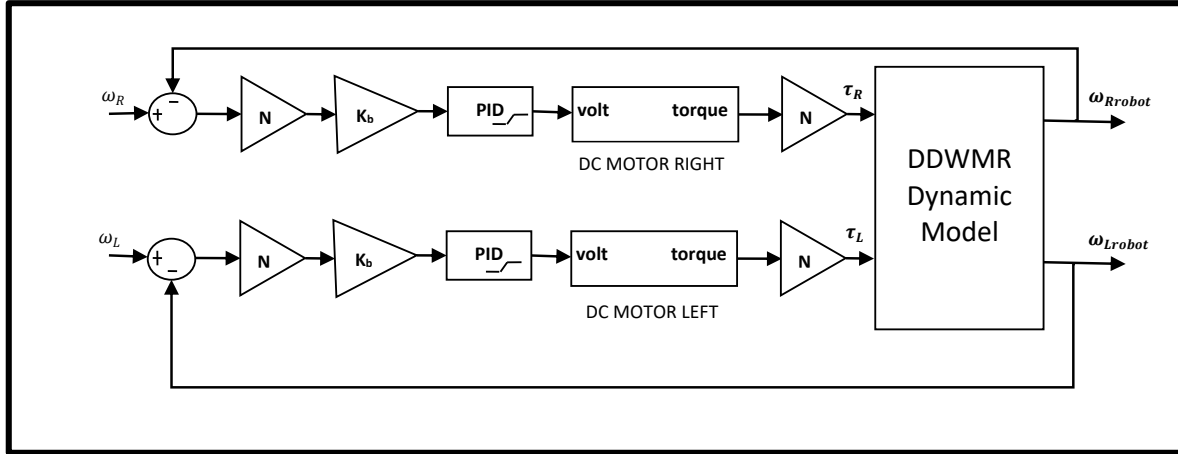
$$H(s) = \frac{K_m}{\tau s + 1} \quad (\text{III.71})$$

according to [97], due to the identification of the parameters, the following transfer functions corresponding to each wheel (left and right) of the control system were obtained and adapted in our work

$$H_L = \frac{2.6}{0.28s + 1} \quad (\text{III.72})$$

$$H_R = \frac{2.7}{0.3s + 1} \quad (\text{III.73})$$

Figure III-4 illustrate the loop of control for DC motors (right and left) angular velocity using PID controller based on the DDWRM dynamic model.



**Figure III-4: Velocity control of DDWMR dynamic model with actuators**

Table III-I describes the parameters of the PIONEER 3DX mobile robot that we used in this work

**Table III-1: Parameters of robot**

Robot		
Term	Unit	Value
$m_c$	kg	27
$m_w$	kg	0.5
$I_c$	kgm <sup>2</sup>	0.732
$I_w$	kgm <sup>2</sup>	0.0025
$I_m$	kgm <sup>2</sup>	0.0012
d	m	0.0
R	m	0.0975
L	m	0.164
N	—	53

### III.7. Conclusion

In this chapter, first the kinematic model, which describes the speeds of the system, is presented by a set of first order differential equations. Second, the dynamic model is described by movement when forces are applied to the system, the Lagrange formulation is presented to provide theoretical background of a non-holonomic DDWMR model with and without slip dynamics. Finally, the parameters necessary for the simulation have been presented.

**CHAPTER IV:**  
**FUZZY TYPE 1 AND INTERVAL TYPE 2**  
**CONTROLLERS**



## **Chapter IV. FUZZY TYPE 1 AND INTERVAL TYPE 2 CONTROLLERS**

### **IV.1. Introduction**

Navigation remains a crucial problem in many applications (surveillance, reconnaissance, cartography, etc.). Many solutions are proposed in the literature to allow a mobile robot to navigate to several waypoints autonomously. However, most of these solutions require an accurate dynamic model with an accurate positioning system and do not take into consideration the appropriate trajectory for optimal navigation, nor the effect of the type of soil on navigation performance especially when the odometry is used as the main sensor. Thus, we offer an efficient solution for the navigation of DDWMR waypoints. The solution must be robust face modeling errors, parameter uncertainties and wheel slip depending on the type of soil. We have chosen a solution based on fuzzy logic, which does not require knowledge of the model in order to overcome these various uncertainties and overcome the complexity of modeling the robot, the ground and the sensors. For this, two fuzzy controllers are implemented and compared. In this chapter, we do this from a high-level perspective in order to give a feel for the nature of fuzzy sets and their applications.

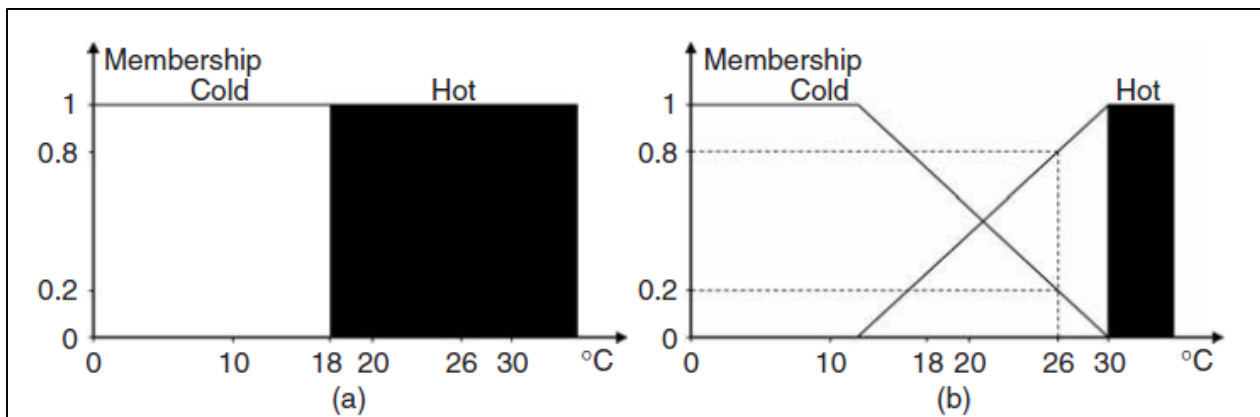
### **IV.2. Overview of fuzzy logic controllers**

#### **IV.2.1. What is a type-1 fuzzy set?**

Fuzzy logic is the logic that deals with fuzzy sets [98]. The concepts of fuzzy sets and fuzzy logic are used in fuzzy control. The compositional rule of fuzzy inference is particularly applicable in the context of fuzzy control. A fuzzy set is a set that does not have a sharp (crisp) boundary. In other words, there is a softness associated with the membership function of fuzzy sets (Figure IV-1).

Suppose that a group of people is asked about the temperature values they associate with the linguistic concepts Hot and Cold. If crisp sets are employed, as shown in figure IV-1a, then a threshold must be chosen above which temperature values are considered Hot and below which they are considered Cold. Reaching a consensus about such a threshold is difficult, and even if an agreement can be reached—for example,  $18^{\circ}\text{C}$ —, is it reasonable to conclude that  $17.99999^{\circ}\text{C}$  is Cold whereas  $18.00001^{\circ}\text{C}$  is Hot?

In figure IV-1b, no sharp boundaries exist between the two sets and that each value on the horizontal axis may simultaneously belong to more than one type 1 fuzzy set but with different degrees of membership. For example, 26°C, which is in the crisp Hot set with a membership degree of 1.0 (figure IV-1a), is now in that set to 0.8 degree, but is also in the Cold set with degree 0.2 (figure IV-1b). Type 1 fuzzy sets provide a means for calculating intermediate values between the crisp values associated with being absolutely true (1) or absolutely false (0). Those values range between 0 and 1 (and can include them); thus, it can be said that a fuzzy set allows the calculation of shades of gray between white and black (or true and false). The smooth transition that occurs between type 1 fuzzy sets gives a good decision response for a type 1 fuzzy logic control system face noises and other uncertainties [98].



**Figure IV-1: Representing Cold and Hot using (a) crisp sets, and (b) type-1 fuzzy sets [98]**

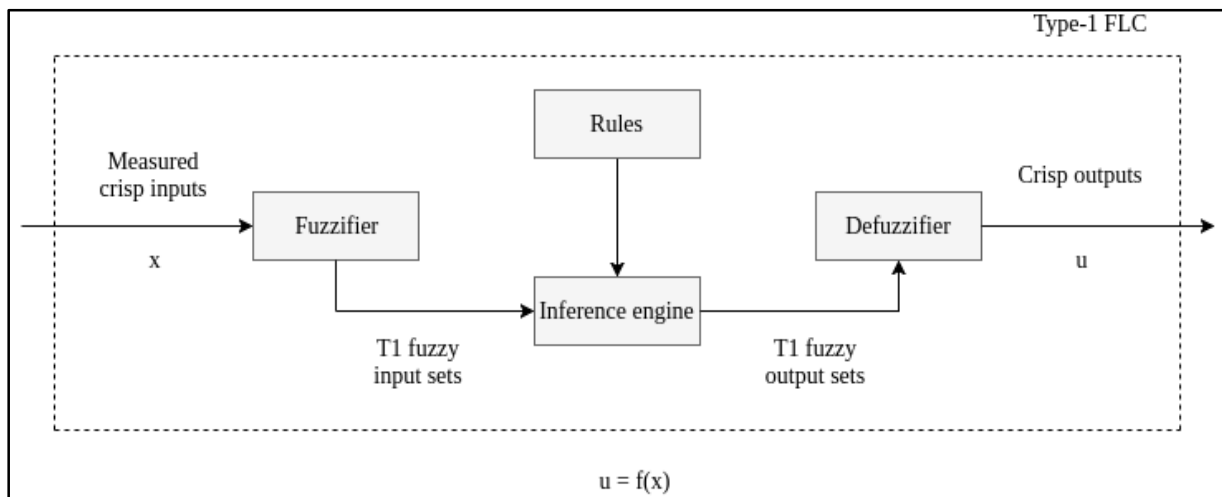
#### **IV.2.2. What is a type 1 fuzzy logic controller?**

With the advent of type 2 fuzzy sets and type 2 fuzzy logic control, it has become necessary to distinguish between type 2 fuzzy logic control and all earlier fuzzy logic control that uses type 1 fuzzy sets. Fuzzy logic control aims to mimic the process followed by the human mind when performing control actions. In its attempt to mimic human control actions, a type 1 FLC, whose structure is shown in figure VI-2, is composed of four main components: fuzzifier, rules, inference engine, and defuzzifier, where the operation of each component is summarized as follows:

- The fuzzifier maps each measured numerical input variable into a fuzzy set. One motivation for doing this is that measurements may be corrupted by noise and are somewhat uncertain (even after filtering). So, for example, a measured temperature of 26°C may be modeled as a triangular type-

1 fuzzy set that is symmetrically centered around 26°C, where the base of the triangle is related to the uncertainty of this measurement. If, however, one believes that there is no measurement uncertainty, then the measurements can be modeled as crisp sets.

- Fuzzy Rules have an if–then structure, for example: *If Temperature is Low and Pressure is High, then Fan Speed is Low.* Each IF part of a fuzzy rule is called its antecedent and the THEN part of a rule is called its consequent. Rules relate input fuzzy sets to output fuzzy sets. All of the fuzzy rules are collected into a rule base.



**Figure IV-2: General structure of a type1 FLC.**

The heavy lines with arrows indicate the path taken by signals during the actual operation of the FLC. Rules are used during FLC design and are activated by the inference engine during the actual operation of the FLC.

- The inference engine decides which rules from the rule base are fired and what their degrees of firing are, by using the fuzzy sets provided to it from the fuzzifier as well as some mathematics about fuzzy sets. The inference engine may also combine each rule’s degree of firing with that rule’s consequent fuzzy set to produce the rule’s output fuzzy set (i.e., its fired-rule output set), and then combine all of those sets (across all of the fired rules) to produce an aggregated fuzzy output set using the mathematics of fuzzy sets; or it may send each rule’s degree of firing directly to the defuzzifier where they are all aggregated in a different way.

- The defuzzifier receives either the aggregated fuzzy output sets from the inference engine or the degrees of firing for each rule plus some information about each consequent fuzzy set, and then processes this data to produce crisp (numerical) outputs that are then passed to the physical actuators that control the actual plant.

In general, real-world control systems, such as fuzzy logic control systems, are affected by the following uncertainties:

- Uncertainties about the FLC inputs: For instance, sensor measurements can be affected by high noise levels and changing observation conditions such as changing environmental conditions, for example, wind, rain, humidity, and so forth. In addition to measurement noise, other possible inputs to the FLC, such as those estimated by an observer or computed using a process model, can also be imprecise and exhibit uncertainty.
- Uncertainties about control outputs that can occur because of changes in an actuator's characteristics due to wear and tear as well as environmental changes.
- Uncertainties about the change in operating conditions of the controller, such as changes in a plant's parameters.
- Uncertainties due to disturbances acting upon the system when those disturbances cannot be measured, for example, wind buffeting an airplane.
- Linguistic uncertainties because the meaning of words that are used in antecedents and consequents linguistic labels can be uncertain, some words mean different things to different FLC designers.
- In addition, experts do not always agree and they often provide different consequents for the same antecedents. A survey of experts will usually lead to a histogram of possibilities for the consequent of a rule; this histogram represents the uncertainty about the consequent of a rule.

There are two widely used architectures for a type 1 FLC that mainly differ in their fuzzy rule consequents. Those architectures are:

- Mamdani FLC, developed by Mamdani and Assilian (1975) in which the antecedents and consequents of the rules are linguistic terms, for example: *If  $x_1$  is Low and  $x_2$  is High, then  $u$  is Low*. The linguistic labels for antecedents and consequents in a Mamdani FLC are represented by type 1 fuzzy sets.

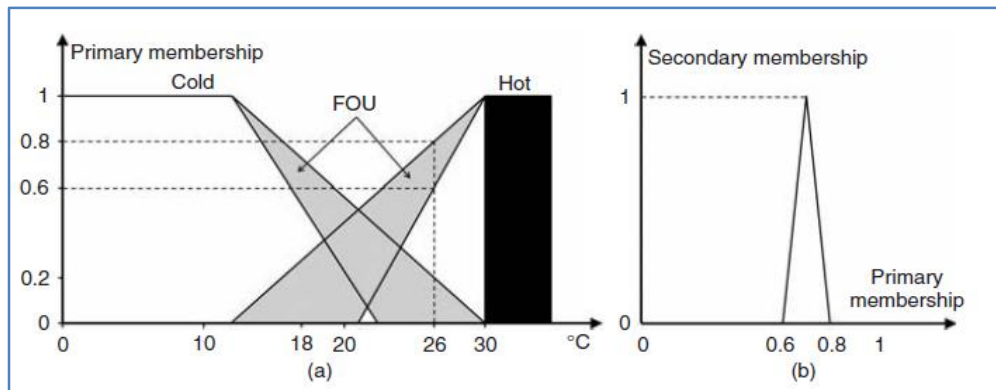
- Takagi–Sugeno (TS) FLC or Takagi–Sugeno–Kang (TSK) FLC (Takagi and Sugeno, 1985) in which the antecedents of the rules are also linguistic terms (modeled as type 1 fuzzy sets), but each rule’s consequent is modeled as a mathematical function of the input variables [98].

### IV.2.3. What is a type-2 fuzzy set?

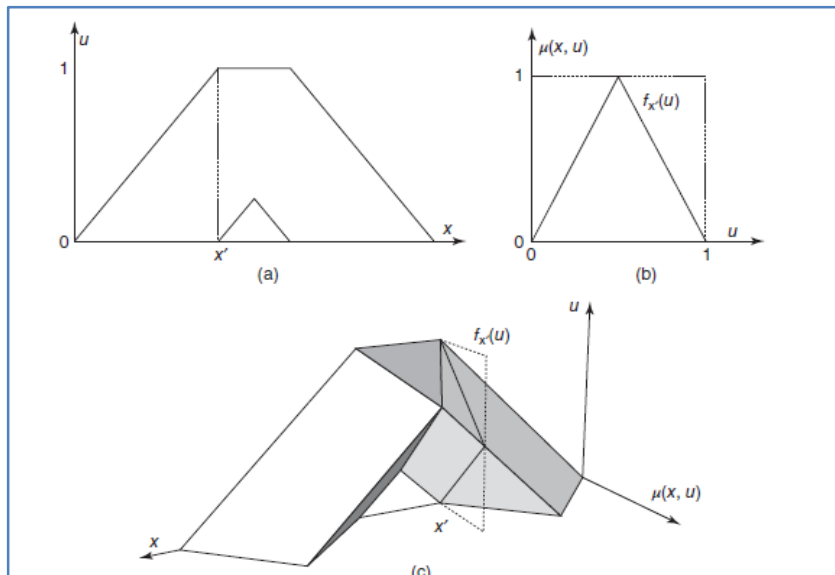
Because type 1 fuzzy sets (e.g., as in figure VI-1b) are themselves crisp and precise (i.e., their membership functions are supposed known perfectly), this does not allow for any uncertainties about membership values, which is a potential shortcoming when using of such fuzzy sets. A type-2 fuzzy set is characterized by a fuzzy membership function, that is, the membership value for each element of this set is itself a fuzzy set in  $[0,1]$ . The membership functions of type 2 fuzzy sets are three dimensional (3D) and include a footprint of uncertainty (FOU) (which is shaded in gray in figure VI-3a). It is the new third dimension of Type 2 fuzzy sets (e.g., figure VI-4c) and its FOU that provide additional degrees of freedom that make it possible to directly model and handle membership function uncertainties.

In figure VI-3a, we observe that the 26°C membership value in Hot is no longer a crisp value of 0.8 (as was the case in figure VI-1b); instead, it is a function that takes values from 0.6 to 0.8 in the primary membership domain, and maps them into a triangular distribution in the third dimension (figure VI-3b), called a secondary membership function. This triangular secondary membership function weights the interval  $[0.6, 0.8]$  more strongly over its middle values and less strongly away from those middle values. Of course, other weightings are possible, including equal weightings, in which case the type 2 fuzzy set is called an interval type 2 fuzzy set (IT2 fuzzy set). Being able to choose different kinds of secondary membership functions demonstrates one of the flexibilities of type 2 fuzzy sets. Figure VI-4c depicts the 3D MF of a general type 2 fuzzy set whose secondary MFs  $[f_x(u)]$  are triangles. When the secondary membership values equal 1 for all the primary membership values (as in the dashed curve in figure VI-4b), this results in an interval-valued secondary membership function, and, as just mentioned, the resulting type 2 fuzzy set is called an

interval type 2 fuzzy set. In figure VI-4c,  $\mu(x, u)$  denotes the membership function value at  $(x, u)$ . Figure VI-5 depicts the FOU of an interval type 2 fuzzy set for Low. The three dashed functions that are embedded within that FOU are type 1 fuzzy sets. Clearly, one can cover this FOU with a multitude of such type 1 fuzzy sets. It is important is interpreting an interval type 2 fuzzy set as the aggregation of amultitude of type 1 fuzzy sets. This suggests that type 1 fuzzy sets and everything that is already known about them can be used in derivations involving interval type 2 fuzzy sets [98].



**Figure IV-3: Type-2 fuzzy sets: (a) FOU and a primary membership and (b) a triangle secondary membership function [98].**



**Figure IV-4: (a) FOU with primary membership (dashed) at  $x'$ , (b) two possible secondary membership functions (triangle in solid line and interval in dashed line) associated with  $x'$ , and, (c) the resulting 3D type-2 fuzzy set.**

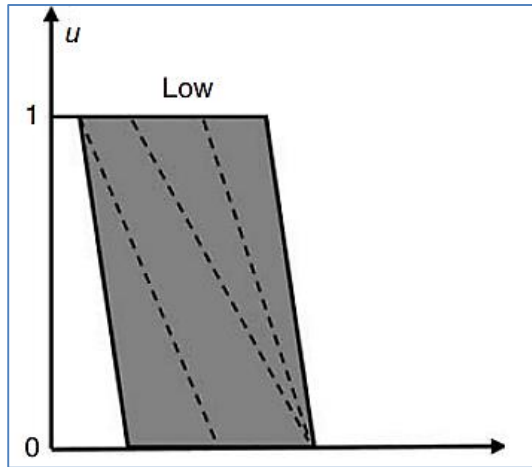


Figure IV-5: Three type-2 fuzzy sets that are embedded in the FOU of Low

#### IV.2.4. What is a type 2 fuzzy logic controller?

A type 2 FLC is depicted in figure VI-6. It contains five components: fuzzifier, rules, inference engine, type reducer, and defuzzifier. In a type 2 FLC the inputs and/or outputs are represented by type 2 FSs, and it operates as follows: crisp inputs, obtained from input sensors, are fuzzified into input type 2 fuzzy sets, which then activate an inference engine that uses the same rules used in a type 1 FLC to produce output type 2 fuzzy sets. These are then processed by a type reducer that projects the type 2 fuzzy sets into a type 1 fuzzy set (this step is called type reduction) after which that type 1 fuzzy set is defuzzified to produce a crisp output that, for example, can be used as the command to an actuator in the control system. Type reduction followed by defuzzification is usually referred to as output processing [98].

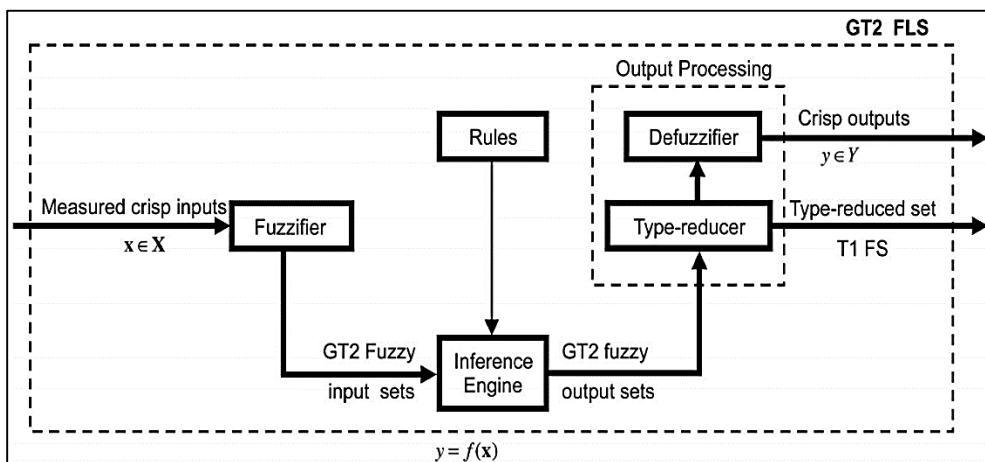


Figure IV-6: Overview of the architecture of a Type 2 FLC.

The heavy lines with arrows indicate the path taken by signals during the actual operation of the FLC. Rules are used during the design of the FLC and are activated by the inference engine during the actual operation of the FLC.

In a type 2 FLC all uncertainties are modeled by the type 2 fuzzy sets membership functions in the antecedents and/or consequents of the rules, as well as by the kind of fuzzifier. As we have explained, a type 2 fuzzy set can be considered as a collection of many embedded type 1 FLCs whose crisp output is obtained by aggregating the outputs of all the embedded type 1 FLCs. Consequently, a type 2 FLC has the potential to outperform a type 1 FLC under certain conditions because it deals with uncertainties by aggregating a multitude of embedded type 1 FLCs. Just as a type 1 FLC is a variable structure controller so is a type 2 FLC, and just as a type 1 FLC has two architectures, Mamdani and TSK, a type 2 FLC also has those two architectures. In a type 2 Mamdani or TSK FLC, the fuzzy sets are type 2.

### IV.3. Theory of interval type 2 fuzzy logic controller

#### IV.3.1. Type-2 fuzzy sets

A type 2 fuzzy set (T2-FS), denoted  $\tilde{A}$ , is characterized by a type 2 membership function  $\mu_{\tilde{A}}(x, v)$ , where  $x \in X$  and  $v \in J_x \subseteq [0,1]$  [13]

$$\tilde{A} = \{((x, v), \mu_{\tilde{A}}(x, v)) \mid \forall x \in X, \forall v \in J_x \subseteq [0,1]\} \quad (IV.1)$$

In which  $0 \leq \mu_{\tilde{A}}(x, v) \leq 1$ .  $\tilde{A}$  can also be expressed as follows:

$$\tilde{A} = \int_{x \in X} \int_{v \in J_x} \mu_{\tilde{A}}(x, v) / (x, v) J_x \subseteq [0,1] \quad (IV.2)$$

Where  $\int$  denotes union over all admissible  $x$  and  $v$ .  $\int$  is replaced by  $\sum$  when the universe of discourse is discrete. For the above definition, the first restriction that  $0 \leq \mu_{\tilde{A}}(x, v) \leq 1$  is consistent with the fact that the amplitudes of a membership function should lie between or be equal to 0 and 1. The second restriction that  $\forall v \in J_x \subseteq [0,1]$  is consistent with the type-1 constraint that  $0 \leq \mu_A(x) \leq 1$ , i.e., when uncertainties disappear a type-2 membership function must be reduced to a type-1 membership function, in which case the variable  $v$  is equal to  $\mu_A(x)$  and  $0 \leq \mu_A(x) \leq 1$ .



### IV.3.2. Interval Type-2 Fuzzy Sets

An interval type 2 fuzzy set (IT2-FS)  $\tilde{A}$  is characterized as [13]:

$$\tilde{A} = \int_{x \in X} \int_{v \in J_x \subseteq [0,1]} 1/(x, v) = \int_{x \in X} \left[ \int_{v \in J_x \subseteq [0,1]} 1/v \right] / x \quad (IV.3)$$

Where  $x$  is the primary variable and  $x \in X$ ;  $v$  is the secondary variable,  $v \in V$  and it has domain  $J_x$  at each  $x \in X$ ;  $J_x$  is called the primary membership of  $x$  and is defined in equation (IV.7) and, the secondary grades of  $\tilde{A}$  all is equal to 1. The union of all the primary memberships for fuzzy set  $\tilde{A}$  is called the footprint of uncertainty (FOU) of  $\tilde{A}$  (see Fig. IV-7),

$$FOU(\tilde{A}) = \cup_{\forall x \in X} J_x = \{(x, v): v \in J_x \subseteq [0,1]\} \quad (IV.4)$$

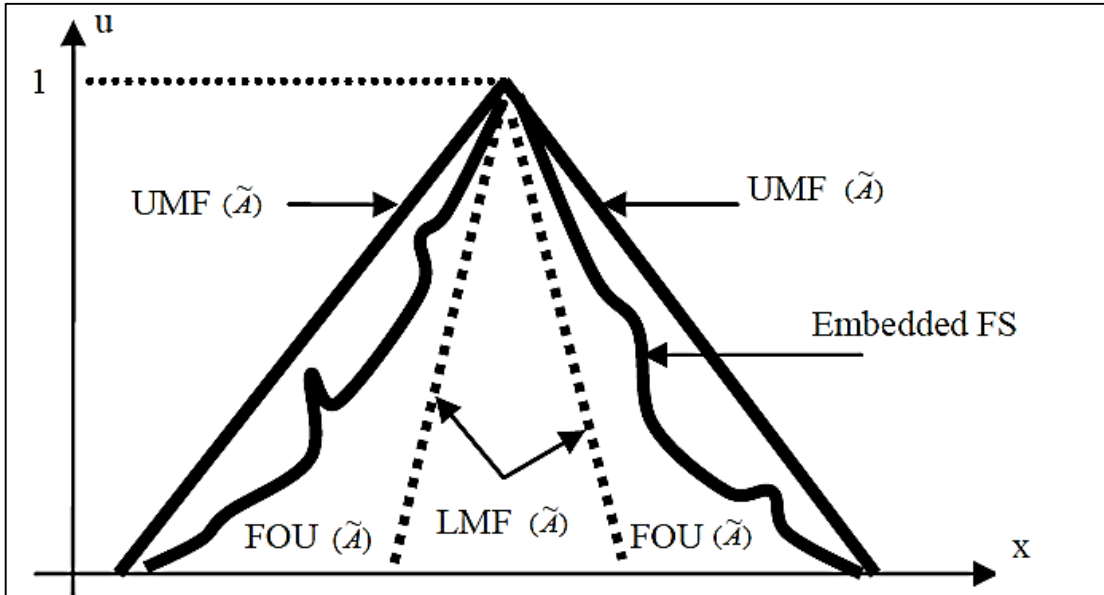


Figure IV-7: Interval type 2 fuzzy set

The upper membership function (UMF) and lower membership function (LMF) of  $\tilde{A}$  are two type-1 MFs that bound the FOU. The UMF is associated with the upper bound of  $FOU(\tilde{A})$  and is denoted  $\bar{\mu}_{\tilde{A}}(x)$ ,  $\forall x \in X$ , and the LMF is associated with the lower bound of  $FOU(\tilde{A})$  and is denoted  $\underline{\mu}_{\tilde{A}}(x)$ ,  $\forall x \in X$ :

$$\bar{\mu}_{\tilde{A}}(x) = \overline{FOU(\tilde{A})} \forall x \in X \quad (IV.5)$$

$$\underline{\mu}_{\tilde{A}}(x) = \underline{FOU}(\tilde{A}) \forall x \in X \quad (IV.6)$$

Note that  $J_x$  is an interval set:

$$J_x = \{(x, v) : v \in [\underline{\mu}_{\tilde{A}}(x), \bar{\mu}_{\tilde{A}}(x)]\} \quad (IV.7)$$

In this work, the input and output variables will be represented by interval type 2 fuzzy sets.

### IV.3.3. Interval Type-2 Fuzzy Logic Controller

The basics of fuzzy logic do not change from type 1 to type 2 sets, and in general, will not change for any type-n [99]. A higher-type number just indicates a higher "degree of fuzziness". Since a higher type changes the nature of the membership functions, the operation that depend on the membership functions change too, however, the basic principle of fuzzy logic are independent of the nature of membership functions and hence, do not change [100].

The interval type 2 FLC (IT2 FLC) contains four components fuzzifier, inference engine, rule base, and output processing that is inter-connected [101]. The interval type 2 FLC works as follows [102]: the crisp input is first fuzzified into input interval type 2 fuzzy sets. The input interval type 2 fuzzy sets then activate the inference engine and the rule base to produce output interval type 2 fuzzy sets. The interval type 2 FLC rules will remain the same as in type 1 FLC, but the antecedents and/or the consequent will be represented by interval type 2 fuzzy sets. The interval type 2 fuzzy outputs of the inference engine are then processed by the type reducer, which combines the output sets and performs a centroid calculation that leads to type 1 fuzzy sets called the type-reduced sets. After the type reduction process, the type-reduced sets are defuzzified (by taking the average of the type-reduced set) to obtain crisp outputs.

### IV.3.4. Computations in an interval type 2 FLC

#### IV.3.4.1. Interval type 2 FLC Fuzzification and Inference

The calculations of the interval type 2 FLC are shown in figure VI-8. When  $x_1 = x'_1$ , the vertical line at  $x'_1$  intersects  $FOU(\tilde{A}_1)$  everywhere in the interval  $[\underline{\mu}_{\tilde{A}_1}(x'_1), \bar{\mu}_{\tilde{A}_1}(x'_1)]$ : and,

when  $x_2 = x'_2$ , the vertical line at  $x'_2$  intersects  $FOU(\tilde{A}_2)$  everywhere in the interval  $[\underline{\mu}_{\tilde{A}_2}(x'_2), \bar{\mu}_{\tilde{A}_2}(x'_2)]$

Two firing levels are then computed, a lower firing level,  $\underline{f}(x')$ , and an upper firing level  $\bar{f}(x')$ , where  $\underline{f}(x') = \min[\underline{\mu}_{\tilde{A}_1}(x'_1), \underline{\mu}_{\tilde{A}_2}(x'_2)]$  and  $\bar{f}(x') = \min[\bar{\mu}_{\tilde{A}_1}(x'_1), \bar{\mu}_{\tilde{A}_2}(x'_2)]$ .

The main thing to observe from this figure is that the result of input and antecedent operations is the firing interval  $F(x')$ , where  $F(x') = [\underline{f}(x'), \bar{f}(x')]$ ,  $\underline{f}(x')$  is t-normed with  $LMF(\tilde{B})$  and  $\bar{f}(x')$  is t-normed with  $UMF(\tilde{B})$ . When  $FOU(\tilde{B})$  is triangular, and the t-norm is minimum, the resulting fired-rule FOU is the trapezoidal FOU as shown in figure (IV.8).

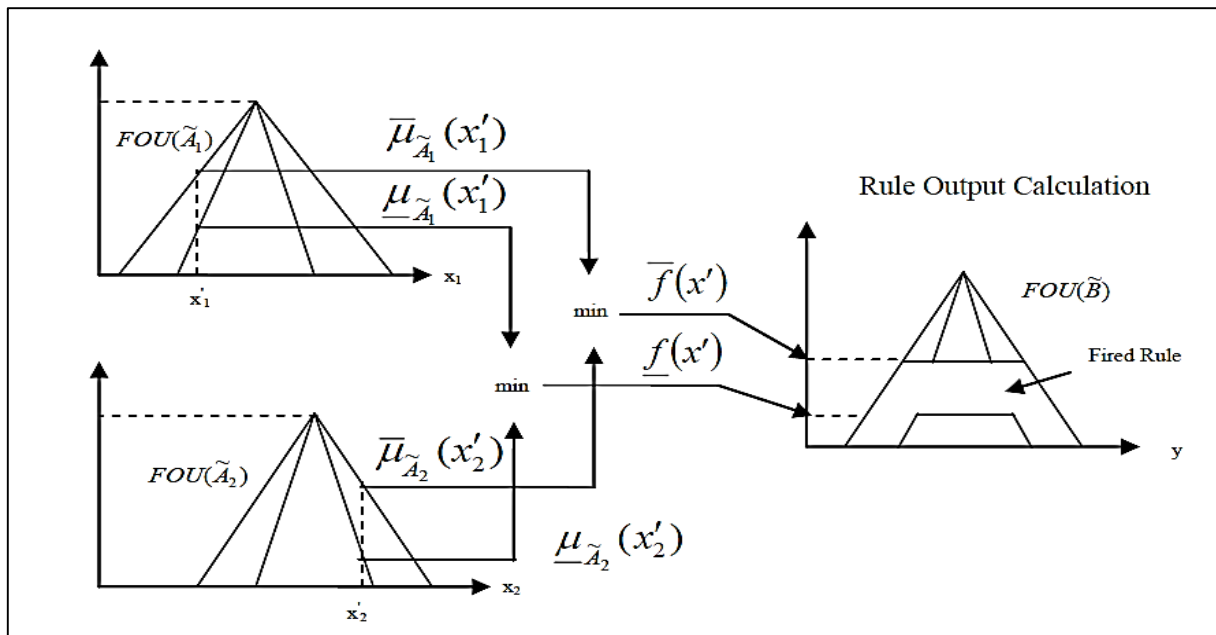


Figure IV-8: Interval type 2 FLC fuzzification and inference

#### IV.3.4.2. Type-reduction

In the interval type 2 FLC the output sets are type 2, so we have to use extended version of type 1 defuzzification methods. The extended defuzzification operation in the type 2 case gives a type 1 fuzzy set at the output. Since this operation takes us from the type 2 output sets of the interval type 2 FLC to a type 1 fuzzy set, we call this operation type-reduction and call the type 1 set so obtained a type-reduced set [99]. There are several methods of type-reduction.

In the version of the IT2-FLS toolbox that we used, the following Type reduction and defuzzification methods are supported:

- 1) Karnik-Mendel Algorithm (KM)
- 2) Enhanced KM Algorithm (EKM)
- 3) Iterative Algorithm with Stop Condition (IASC)
- 4) Enhanced IASC (EIASC)
- 5) Enhanced Opposite Direction Searching Algorithm (EODS)
- 6) Wu-Mendel Uncertainty Bound Method (WM)
- 7) Nie-Tan Method (NT)
- 8) Begian-Melek-Mendel Method (BMM)

IT2 FLSs have demonstrated better ability to handle uncertainties than their T1 counterparts in many applications; however, their high computational cost may prevent them from accessing some cost-sensitive real-world applications. In the article [103], which provides a comprehensive overview and comparison of categories of methods to reduce the cost of calculating FSL IT2. The first category: (EKM) (IASC) (EIASC) (EODS) includes improvements to the KM algorithms. Experiments have shown that they are generally all faster than KM algorithms; among them, EODS algorithms are the fastest for practical IT2 FLS. Furthermore, EIASC algorithms, which are much simpler than EODS algorithms and are at most 1.2 times slower, may also be preferred by practitioners for ease of understanding and implementation. The second category includes reciprocating type reducers, which have a closed form representation and, therefore, are more convenient for analysis. Experiments have shown that these methods are generally faster than KM algorithms; among them, the WM and NT methods are the fastest. The BMM method may also be preferred because its properties, eg, stability and robustness, have been extensively studied.

If more computational cost savings are desired, alternative TR algorithms such as WM, NT, BMM method can be considered as they are consistently faster than EODS algorithms and their outputs are close to outputs of KM algorithms. In our work, we used the NT method.

#### **IV.3.4.3. Defuzzification**

Defuzzification method used to convert type-reduced set to crisp output.

#### **IV.4. Waypoint navigation**

Traditional human used waypoints are mountains, waterways, oasis, buildings, highways and railways and other reference objects. Typical waypoints on waters are buoys, lighthouses, anchorages and harbors. Typical waypoints in the city are important buildings, skyscrapers, churches, supermarkets, factories, stadiums, bridges, tunnels and parks. There are waypoints for practical considerations made by industries, e.g. radio masts, beacons and satellites.

Nowadays the conception of waypoints has reached the abstraction level, which minimizes the requests for visible environmental features. For example, artificial airways also known as highways-in-the-sky has no visible form at all. Those are series of waypoints used to define invisible navigation routes that a pilot navigates through. Abstract waypoints are maintained by radio masts on the ground and satellite based technology of Global Positioning System.

Generally, a single waypoint is a set of coordinates, which indicates an important point in the environment. Waypoints are defined in a 2D space by longitude and latitude. Some air navigation systems incorporate the third coordinate of height. The fourth coordinate of time is added in case an outer space waypoint is to be specified.

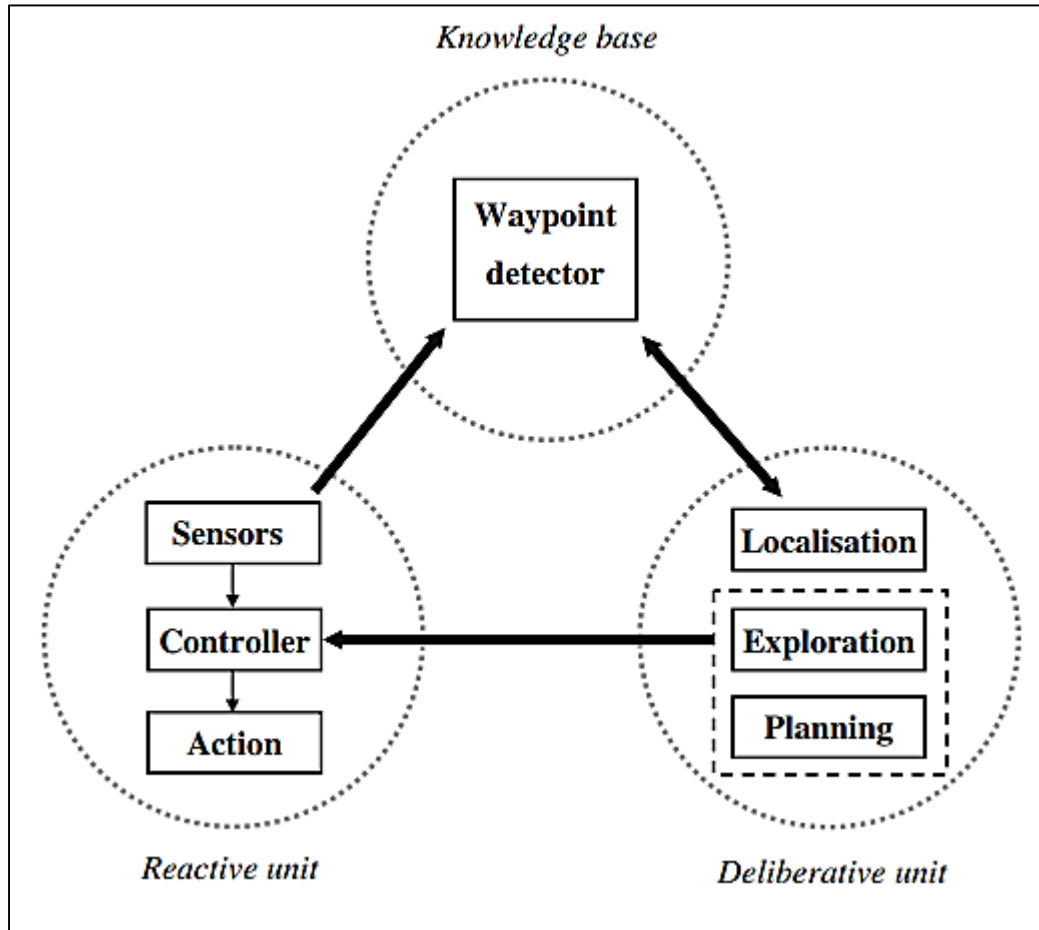
Waypoints are named this way because waypoints are used to "point out the way" to the goal from the robot's initial position. When navigating through waypoints the robot has an initial instruction about the actions that has to be taken when a specific waypoint is reached.

Actions may imply some modification of the robot's motion, like a direction, speed, acceleration or height when navigating an aircraft. Typically, the modification leads to reaching the next waypoint.

The waypoint navigation system shown in figure IV-9 is composed of three main units.

1. The reactive unit provides reactive navigation for the robot, which, in the current work is achieved by a fuzzy logic control.
2. The knowledge base contains the robot's acquired knowledge of the environment represented as a set of waypoints and paths between waypoints.

3. The deliberative unit is a high-level control unit used for navigation when existing knowledge of the environment is available. It contains three elements, namely localization, exploration and planning



**Figure IV-9: Block diagram of the waypoint navigation system**

Assuming the waypoint navigation system has no previous knowledge of its environment, it attempts to reach a goal under reactive control, while continuously transmitting sensor information and its current action to the knowledge base. The waypoint in the current system is defined as a location where the robot changes its behavior as a result of reacting to the perceived environment.

Identified waypoints are entered into the knowledge base and are then available to the deliberative unit for use in exploration and planning. Suitable waypoints for the environment can be obtained following off-line simulation or generated on-line either as a result of executing previous navigation tasks or by purposely invoking exploration. Information transfer between the waypoint

detector and the localization unit is two-way, since localization needs access to the stored waypoints in order to instruct the waypoint detector to remove duplicate entries. The solution to a navigation task produced by the planner is a path defined by a sequence of waypoints.

#### **IV.4.1. Reactive Unit**

The function of the reactive unit is to control the robot when navigating towards a given waypoint or when exploring previously uncharted regions of the environment.

#### **IV.4.2. Waypoint Knowledge Base**

To be able to plan future movements in an autonomous and intelligent manner, a robot requires a form of memory to record where it has already been.

#### **IV.4.3. Deliberative Unit**

The deliberative control system contains three sub-units, namely localization, exploration and planning.

##### **IV.4.3.1. Localization**

The knowledge base of waypoints may be interrogated to provide the location of the robot with respect to previously discovered waypoints.

##### **IV.4.3.2. Exploration**

In order to generate waypoints for use in planning, exploration of the environment is required. Given the practical task of moving from a start position to a goal point, two practical approaches to the generation of waypoints have been implemented. The first approach is appropriate when the robot is introduced to a new environment and the assumption is made that it is completely known. The movement to the goal point, is then simulated off-line and the waypoints so generated can be used to plan the best path. The second approach is to determine suitable paths using the waypoints already entered into the knowledge base arising from previous navigation tasks or planned explorative movements. In this case, no a priori knowledge of the environment itself is needed.

#### **IV.4.3.3. Planning**

The introduction of waypoints as part of the navigation process gives the opportunity to search for a feasible path using only the recorded collection of waypoints rather than attempting to search the whole environment.

In our work, we considered four waypoints. We have considered the initial waypoint1 whose coordinates: ( $x_{\text{start}} = 2$  meter,  $y_{\text{start}} = 0$  meter) and the final waypoint4 whose coordinates: ( $x_{\text{final}} = 0$  meter,  $y_{\text{final}} = 0$  meter) pass through two waypoints. The navigator receives the series of waypoints. Its role is to generate a trajectory, which is monitored based on the concept of fuzzy logic.



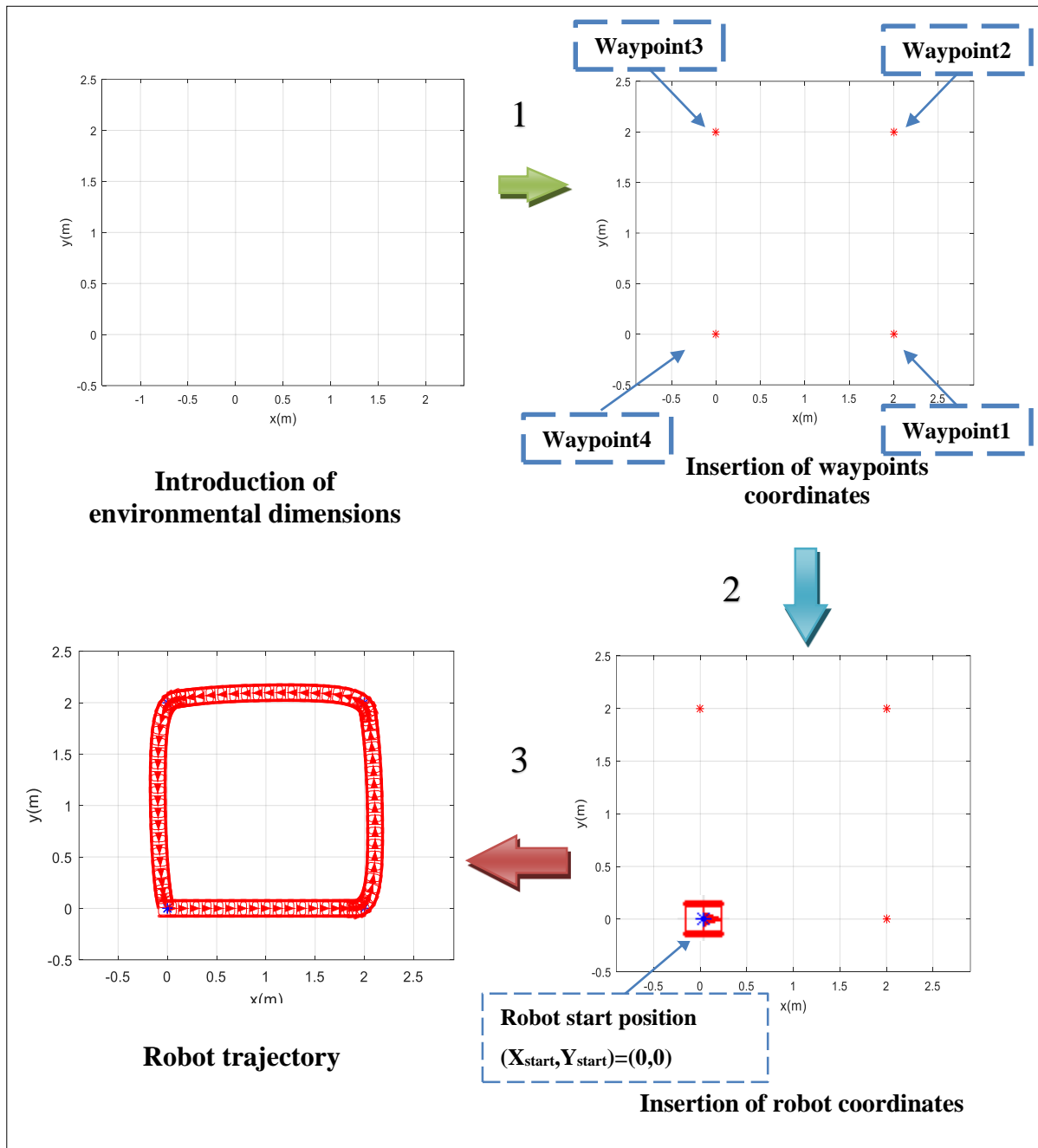
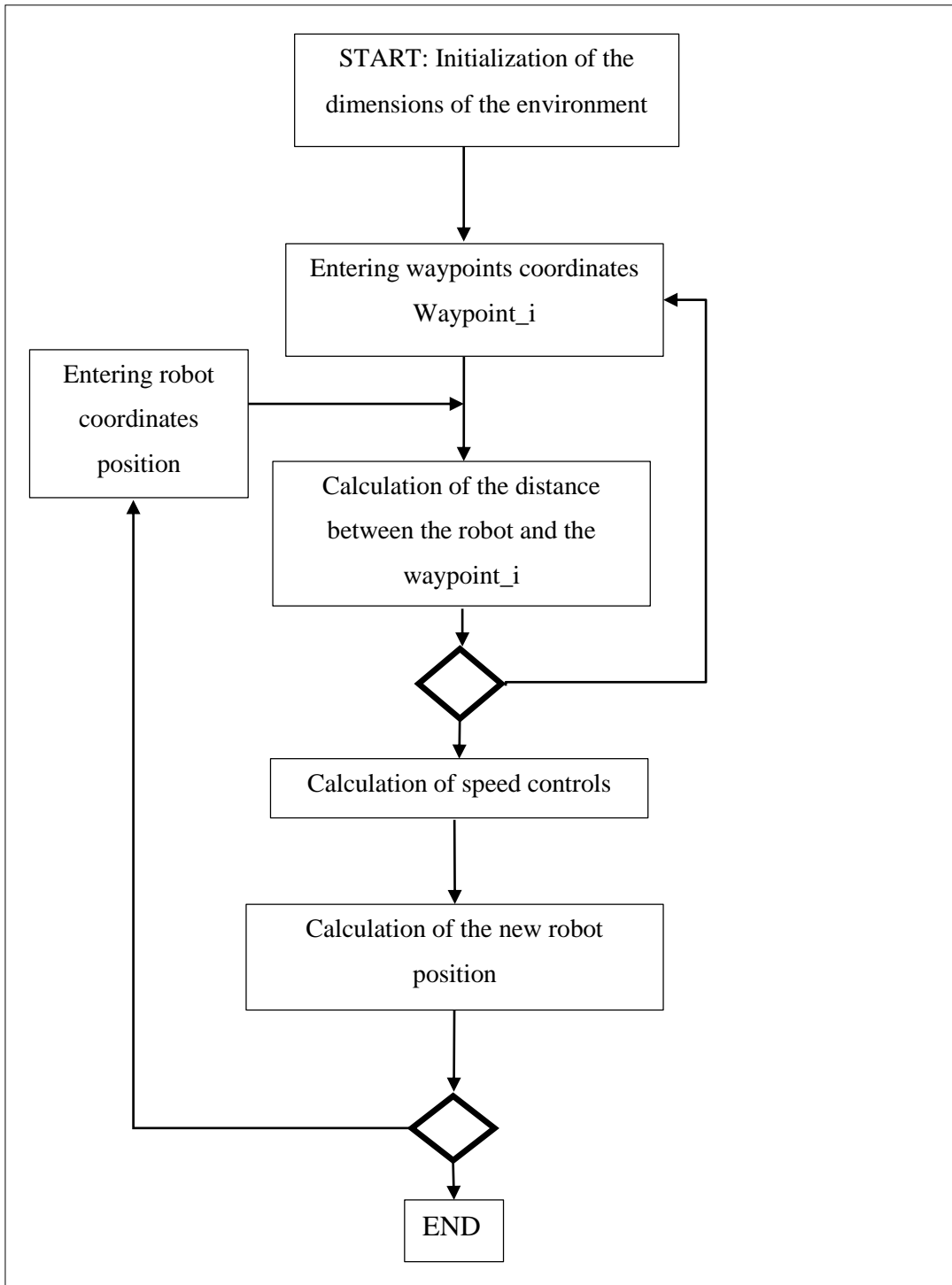


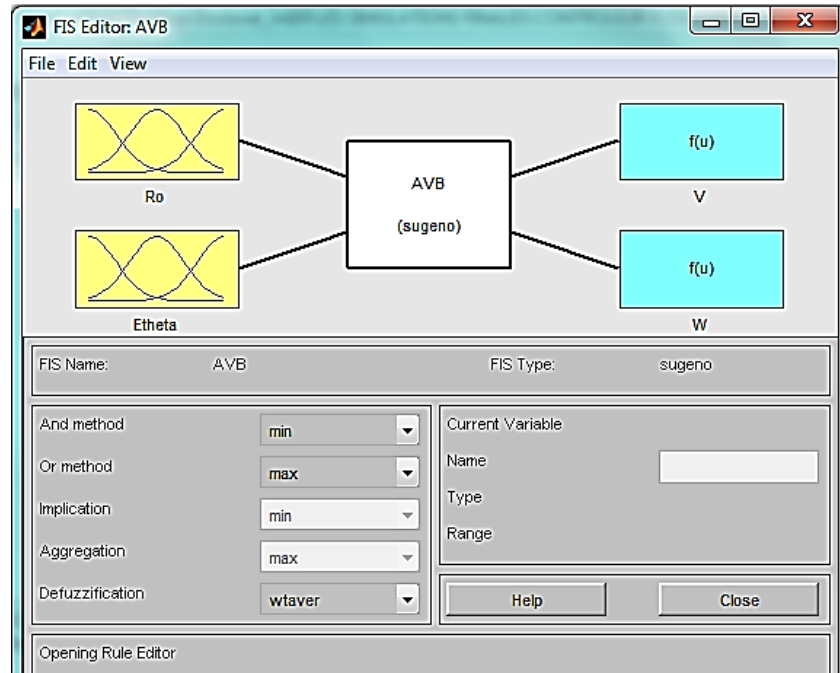
Figure IV-10: Waypoints Navigator



**Figure IV-11: Waypoint navigation diagram**

## IV.5. Navigation using Type 1 Fuzzy controller

A type 1 fuzzy logic controller is implemented to solve the waypoints navigation problem. For that purpose, a MIMO (two-input two-output) Takagi-Sugeno controller is implemented (figure IV-9) using MATLAB® Fuzzy Logic Toolbox [104].



**Figure IV-12: Internal structure of Type1 Fuzzy Logic Controller**

At each sample time, robot position and orientation errors ( $\Delta x, \Delta y, \Delta \theta$ ) are calculated (by comparing current and desired pose using odometry). Distance error ( $Ro$ ) and orientation error ( $Etheta$ ) are used to be the inputs of the Type 1 Fuzzy controller (figure IV.9). This controller is able to deliver appropriate action for mobile robot, translational and angular velocity ( $V, W$ ) respectively. To control the system, we consider the following ranges for the two errors  $Ro$  and  $Etheta$ :  $0 \leq Ro \leq 1.5(m)$  and  $-\pi \leq Etheta \leq \pi(rad)$ . For the controller output, maximum translational and angular velocities are given respectively by  $V_{max} = 1.2 m/s$ ;  $W_{max} = \pm 1.4 rad/s$ .

### IV.5.1. Type 1 fuzzy logic controller structure

To solve the waypoints navigation problem, the first input of the FLC controller (error distance ( $R_o$ )) is represented by five membership functions: Z (Zero), S (Small), M (Medium), B (Big), and VB (Very Big) (figure IV-10).

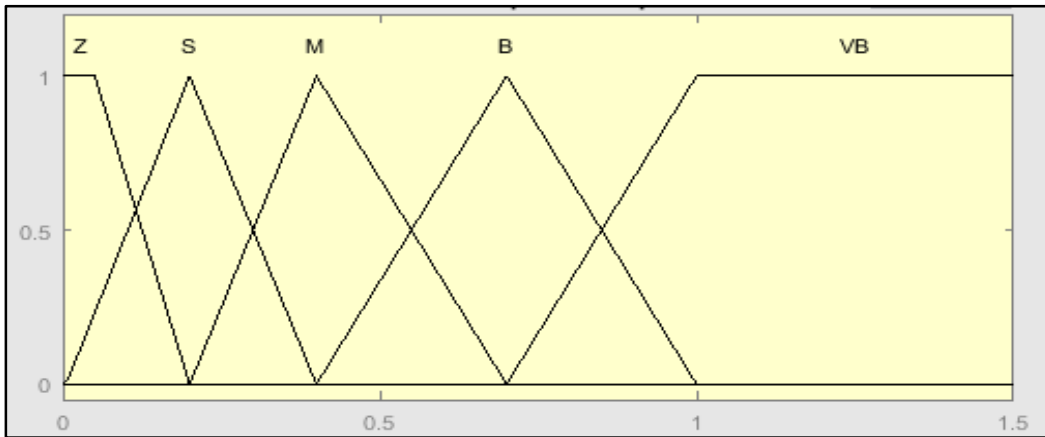


Figure IV-13: Input variable "Ro"

The second input (angular error ( $E_{theta}$ )), is represented by seven membership functions: NB (Negative Big), NM (Negative Medium), NS (Negative Small), Z (Zero), PS (Positive Small), PM (Positive Medium) and PB (Positive Big), (figure IV-11).

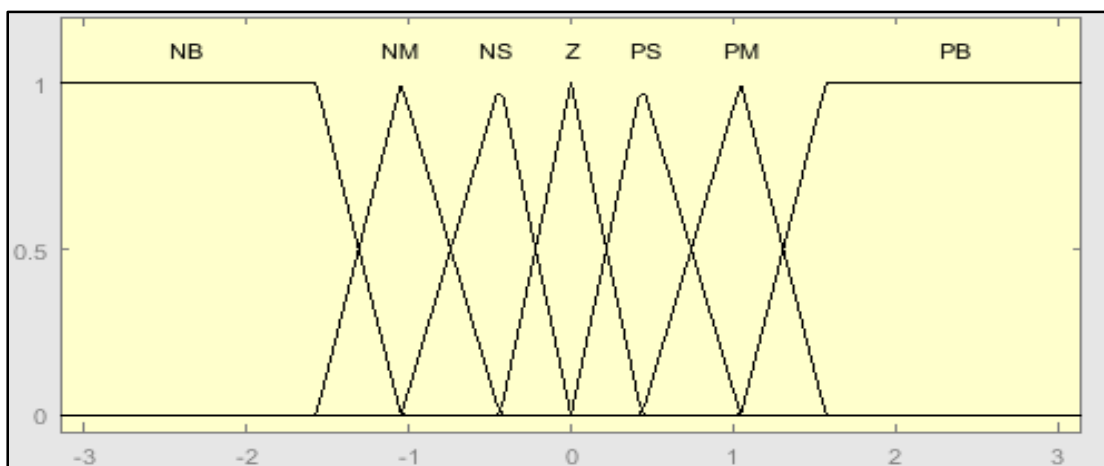


Figure IV-14: Input variable "Etheta"

The translational velocity, which is one of the outputs of the controller, is described with the following linguistic variables: S (Stop), SL (Slow), N (Normal) and F (Full).

The angular velocity is described with the following linguistic variables: Right\_Full (RF), Right (R), Right\_Slow (RS), No\_Rotation (NR), Left\_Slow (LS), Left (L), Left\_Full (LF). The parameters defining the outputs of the TKS-FLC controller are listed in Table VI-1 and Table IV-2.

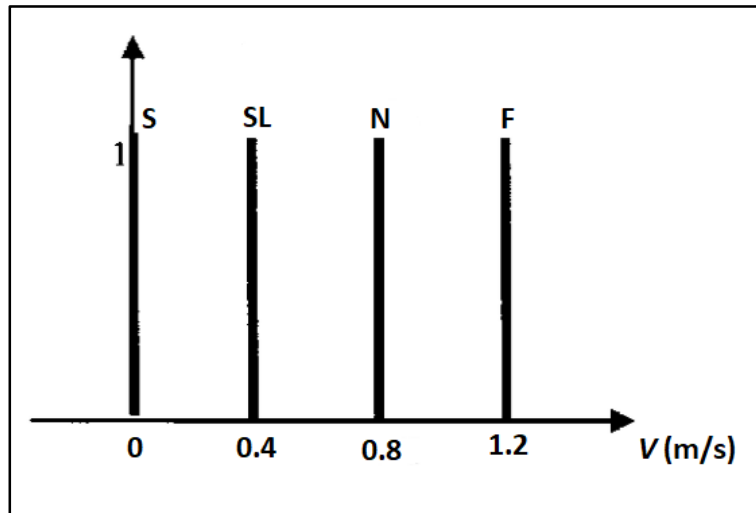


Figure IV-15: Parameters for output: The translational velocity

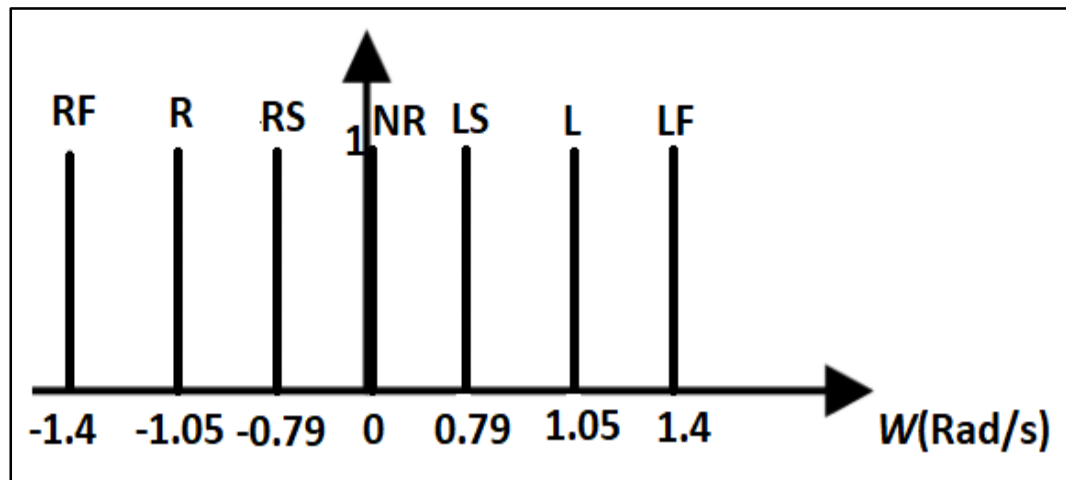


Figure IV-16: Parameters for output: The angular velocity

#### IV.5.2. Type 1 FLC Rule Base

The main step to design a fuzzy logic controller is the determination of the rules base (Inference System). This step requires good expertise of the operator. The inference matrix of the proposed controller is illustrated in Table IV-2. This table is developed based on 35 IF-THEN fuzzy

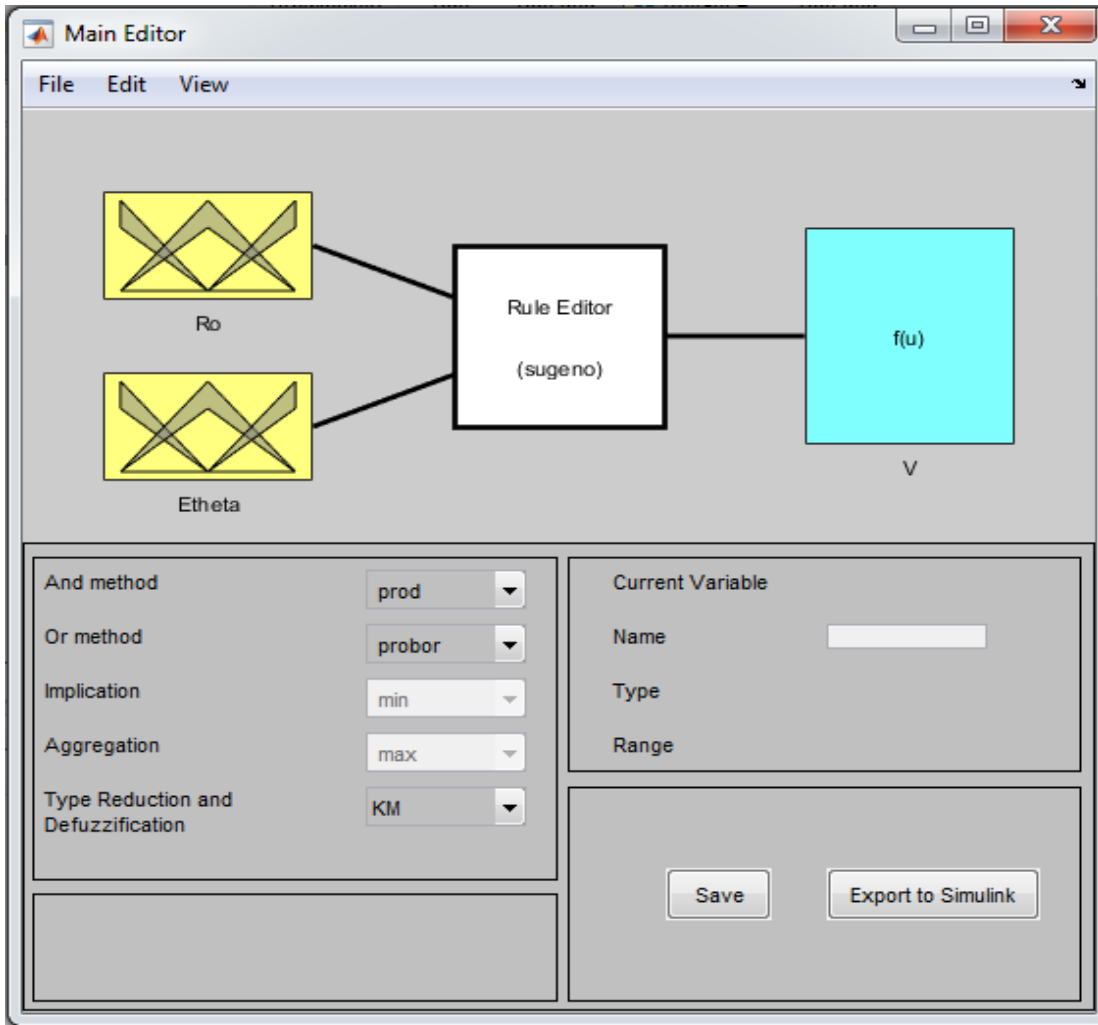
rules. . For example, the first rule is: IF (Angular error is Big Negative and Distances is Zero) THEN (translational Velocity is Slow and Rotational Velocity is Left). This rule allows the robot to change its direction, meanwhile the robot try to keep tracking the desired position.

**Table IV-1: Rules of translational velocity (V) and angular velocity (W)**

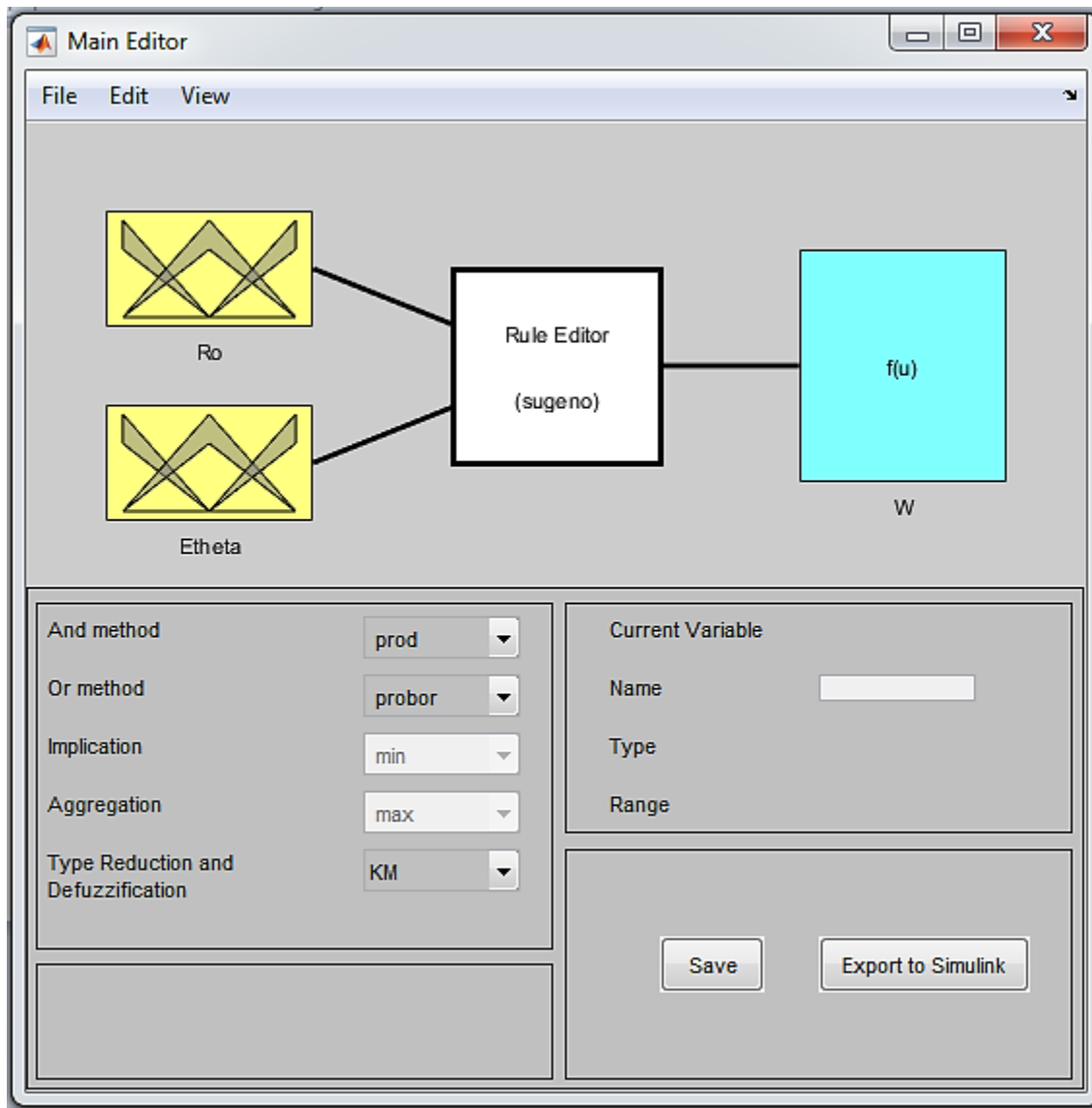
VARIABLES			Error of Angle ( <i>Etheta</i> )						
			<i>NB</i>	<i>NM</i>	<i>NS</i>	<i>Z</i>	<i>PS</i>	<i>PM</i>	<i>PB</i>
Distance robot/goal ( <i>Ro</i> )	<i>Z</i>	<i>V</i>	S	S	S	S	S	S	S
		<i>W</i>	L	LS	NR	NR	NR	RS	R
	<i>S</i>	<i>V</i>	SL	SL	N	N	N	SL	SL
		<i>W</i>	LF	L	LS	NR	RS	R	RF
	<i>M</i>	<i>V</i>	N	N	N	F	N	N	N
		<i>W</i>	LF	LF	LF	NR	RF	RF	RF
	<i>B</i>	<i>V</i>	F	F	F	F	F	F	F
		<i>W</i>	LF	LF	LF	NR	RF	RF	RF
	<i>VB</i>	<i>V</i>	SL	SL	N	F	N	SL	SL
		<i>W</i>	LF	LF	L	NR	R	RF	RF

#### IV.6. Waypoint Navigation using interval type 2 Fuzzy logic controller

In the design of type 2 fuzzy logic controller, we have extended the proposed type 1 controller to a type 2 using the Interval Type 2 Fuzzy Logic System (IT2-FLS v1.1) Matlab/Simulink Toolbox [105]. The inputs of the controllers are the distance robot/goal (*Ro*) and the error of angle (*Etheta*). The first controller delivers appropriate translational velocity (*V*) for mobile robot, when the second controller delivers angular velocity (*W*) as shown in figure IV-15 and figure IV-16.



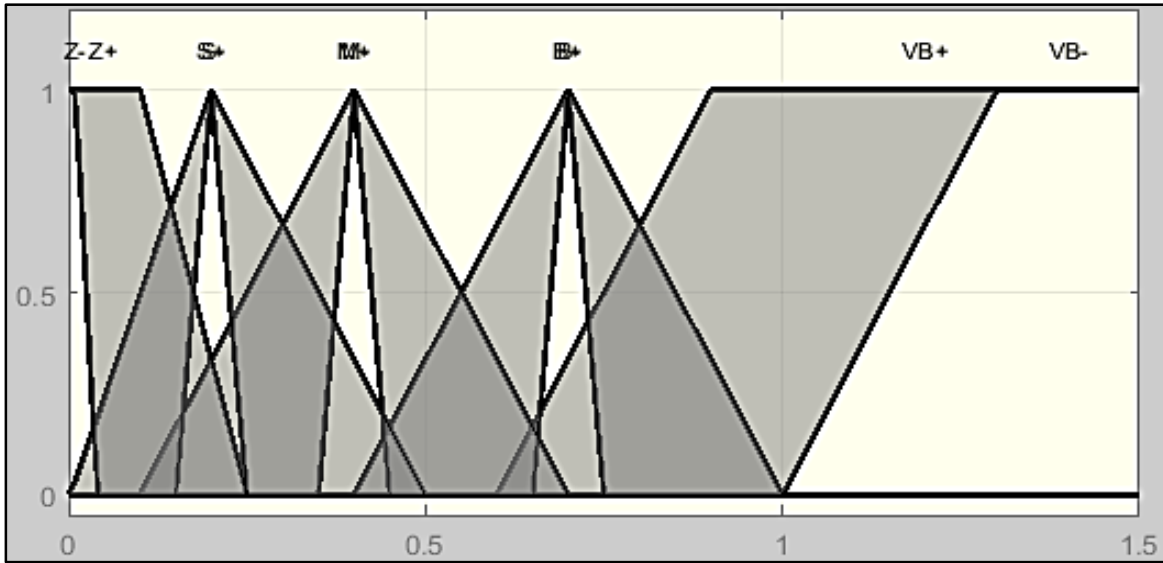
**Figure IV-17: Type 2 fuzzy translational velocity control**



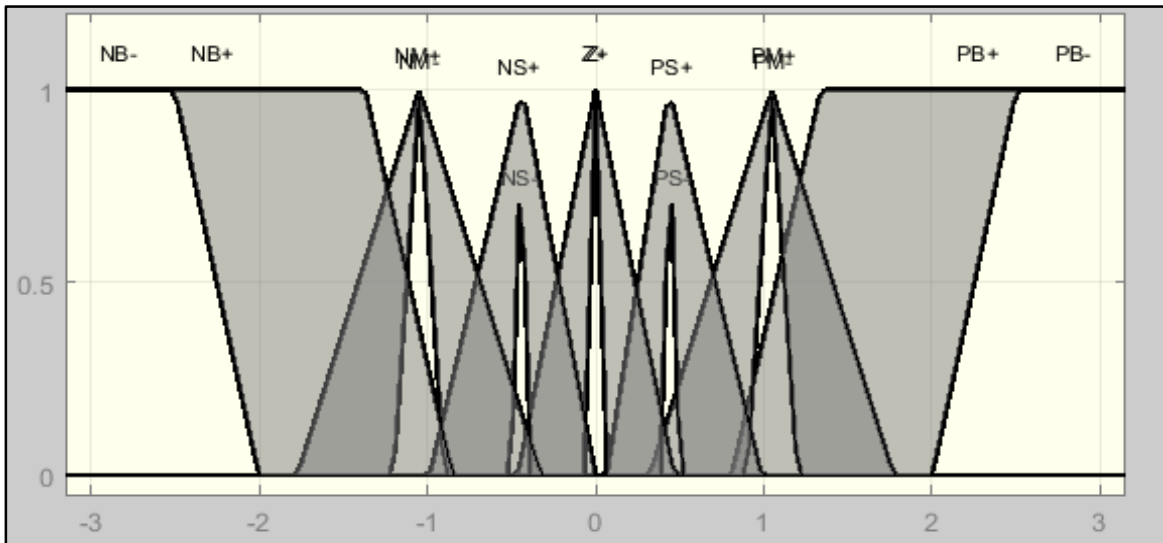
**Figure IV-18: Type 2 fuzzy angular velocity control**

The membership functions of the proposed controller inputs are represented in figure IV-17 and figure IV-18:





**Figure IV-19: Input variable "Ro"**



**Figure IV-20: Input variable "Etheta"**

Similarly to the type 1 fuzzy logic controller, we propose 35 (IF- THEN fuzzy rules) for the type 2 fuzzy controller. The output processing block is computed by Takagi-Sugeno "som/prod" inference method, and "NT" type for reduction and defuzzification.

#### **IV.7. Conclusion**

In this chapter, we propose an efficient solution for DDWMR waypoints navigation that is robust face modeling errors and parameters uncertainties. For this purpose, two controllers are developed and implemented.

Due to the efficiency of fuzzy controllers in mobile robot navigation, we have presented in this chapter a fuzzy logic controller for multiple waypoints navigation. Two types of FLC are illustrated. Firstly, we have presented an overview of type 1 fuzzy logic controllers and their limitations. We then presented the theory of type 2 and interval type 2 fuzzy logic controller. Finally, we have developed two fuzzy controllers: type 1 and interval type 2 fuzzy logic controller for multiple waypoints navigation.

**CHAPTER V :**  
**RESULTS AND DISCUSSION**

## Chapter V. RESULTS AND DISCUSSION

### V.1. Introduction

In this work, we have implemented some navigation tasks on the mobile robot pioneer 3AT available at the robotics laboratory of the Polytechnic Military School.

Precisely, we made a deep comparison between type 1 FLC and Interval type 2 FLC in term of precision, robustness and computational time. To evaluate the performances of the aforementioned controllers, three scenarios are considered; firstly, go to goal problem, secondly, multiple waypoints navigation, thirdly, robustness analysis under modeling error and parameters uncertainties that can affect the mobile robot. In the third scenario, several disturbances are considered:

- Modeling errors: deformation of one of the wheels ( $\Delta R$ ) and uncertainty in the mobile robot length ( $\Delta L$ ) of the mobile robot.
- Localization errors: we consider odometry measurements on both wheels. With time, odometric localization accumulates errors in an unbounded fashion due to wheel slippage, floor roughness and discretized sampling of wheel increments.
- Loss of efficiency of motors: when a motor is operated at variable load and/or varying speed, its efficiency can drop far below nominal value.
- Wheel slip: in practice, the assumption “pure rolling without slip” is often violated due to various factors such as slippery floor, external forces, and so on. The wheel slip is one of the reasons making the tracking performance of non-holonomic WMRs reduce considerably. Thus control methods having the ability to overcome the undesired effects of the wheel slips must be taken into account.

In order to consider all these faults and uncertainties for mobile robot during navigation, in our study, we have used the Virtual Robot Experimentation Platform (V-REP) as physical simulator, which provides an easy and intuitive environment to create our own virtual platform with. The V-REP includes many popular robots, objects, structures, actuators and sensors. Furthermore, it provides the possibility to simulate all the faults and uncertainties illustrated in figure II-1.

In this chapter, simulation results and discussion are considered following three scenarios given below:

- Navigation using the kinematic model of the DDWMR.
- Navigation using the dynamic model of DDWMR.
- Navigation using VREP simulator.

## V.2. Simulation using DDWMR kinematic model

In the first time, we consider only the kinematic model of the mobile robot. Perfect velocity tracking is assumed to generate the actual vehicle control inputs as shown in figure V-1.

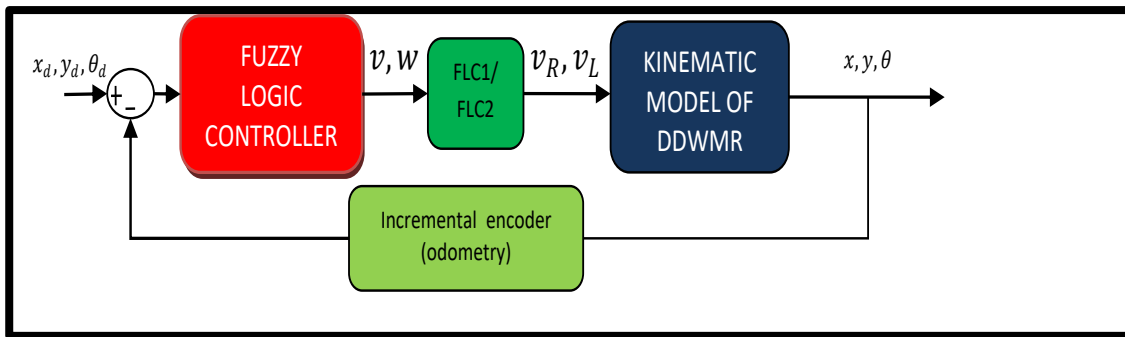
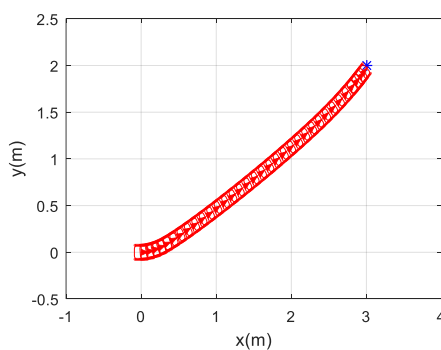
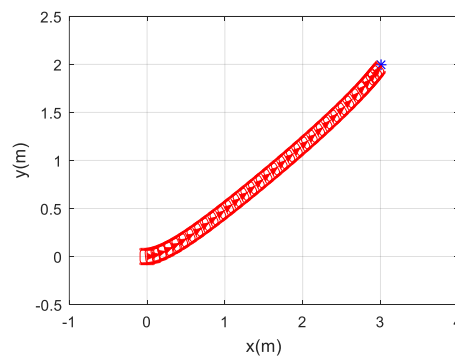


Figure V-1: Global scheme of control with kinematic model of DDWMR

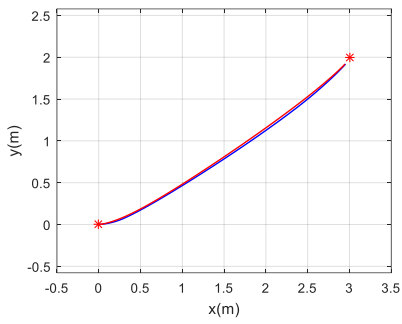
### V.2.1. Scenario 1: Go to goal (single waypoint)



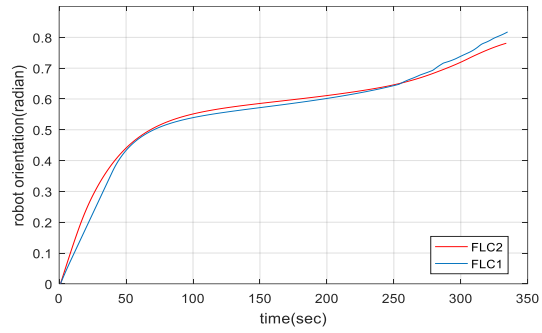
(a) Result of Go to goal with controller FLC1.



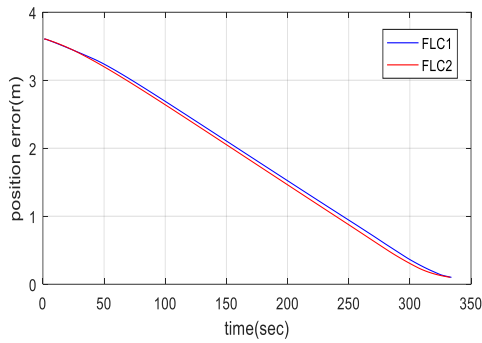
(b) Result of Go to goal with controller FLC2.



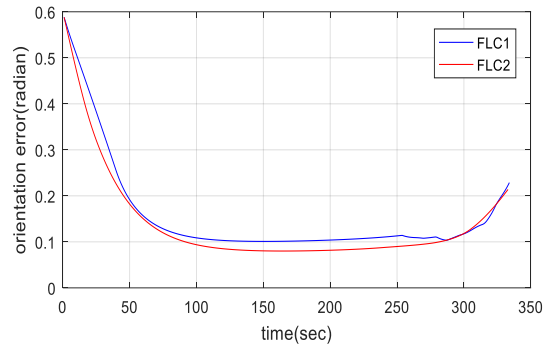
(c) Robot x/y position (m).



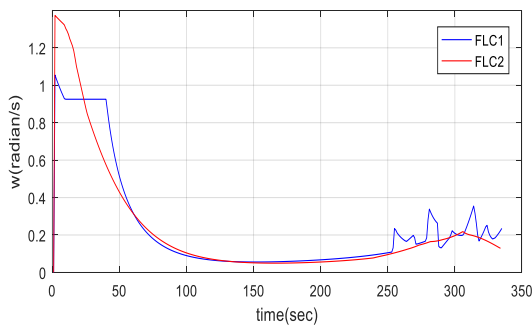
(d) Robot orientation (radian).



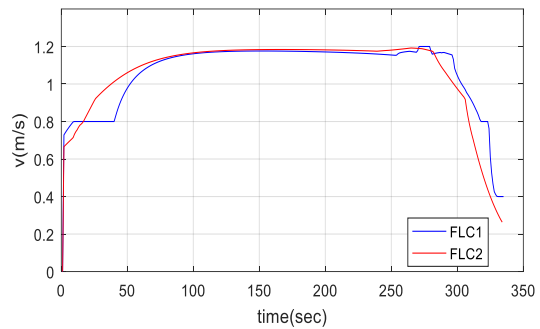
(e) Tracking position errors (m).



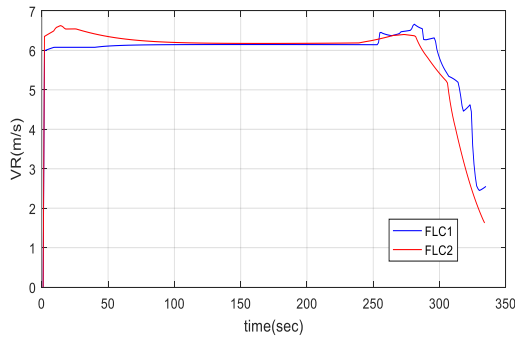
(f) Tracking orientation errors (radian).



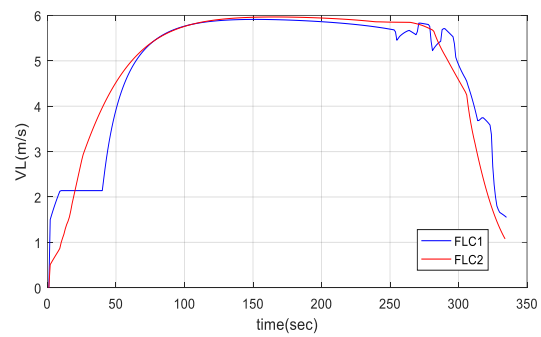
(g) Angular velocity (radian/s).



(h) Linear velocity (m/s).



(i) Linear velocity of the right wheel (m/s).



(j) Linear velocity of the left wheel (m/s).

**Figure V-2: Results obtained for the scenario: Go to goal.**

As can be seen from figure V-2, the two controllers FLC1 and FLC2 are based respectively on type 1 and interval type 2 fuzzy logic controllers provide similar results. Table V-1 confirms these results.

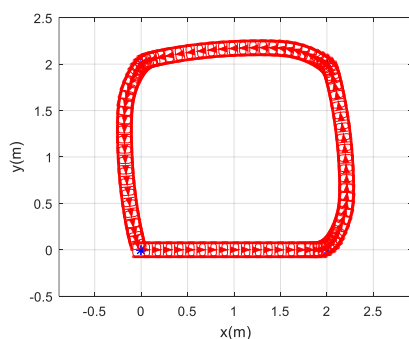
**Table V-1: Computational time comparison for the two controllers for single waypoint**

	<b>FLC1</b>	<b>FLC2</b>
<b>Mean (s)</b>	0.018	0.1306
<b>Std (s)</b>	0.014	0.0174
<b>Number of iterations</b>	334	333

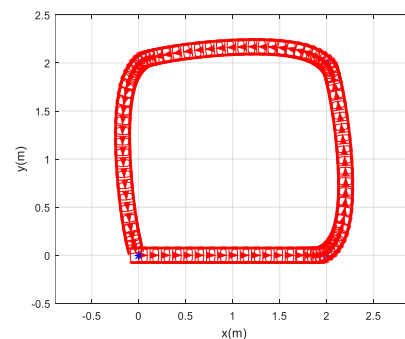
Figure V-2 (a and b), illustrates the obtained results by the two controllers (FLC1 and FLC2) respectively. As can be seen, the mobile robot reaches the goal successfully with both controllers. Similar performances are obtained using T1 and IT2 FLC. Table V-1, confirms the obtained results by comparing the computational time of the two controllers using an i3 processor with 6Go of RAM.

### V.2.2. Scenario 2: Multiple waypoints

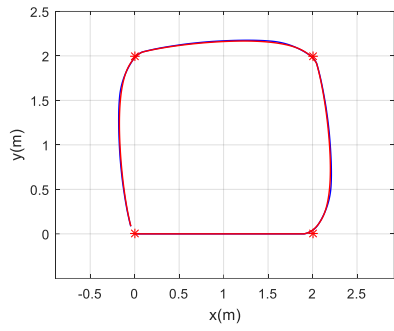
In this simulation, the mobile robot should navigate to multiple waypoints. Four waypoints are considered to compare the behavior of the two controllers. From figure V-3, (a, b, c and d) the two controllers navigate to different waypoints successfully with similar performances even the interval type 2 controller contains additional step of type reduction. This result is confirmed by the table V-2 that compare the computational time of the two controllers.



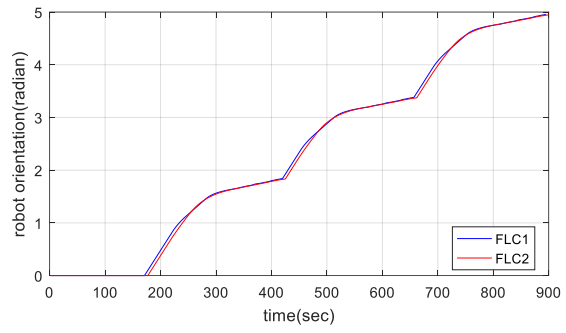
(a) Waypoints navigation with with FLC1 controller.



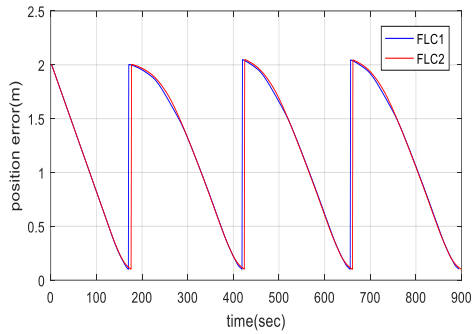
(b) Waypoints navigation with FLC2 controller.



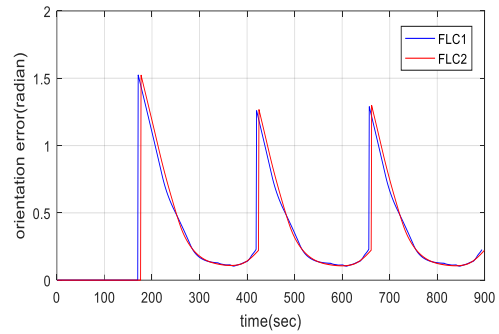
(c) Robot x/y position (m).



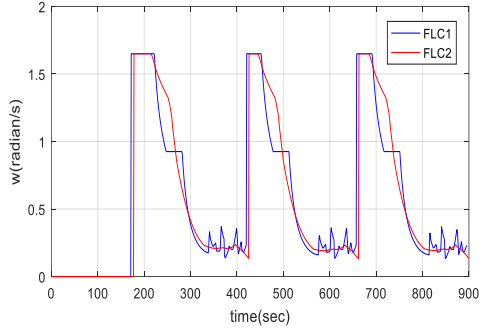
(d) Robot orientation (radian).



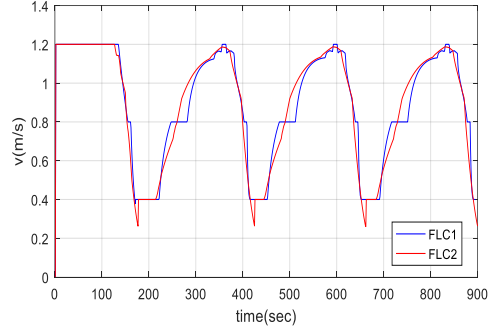
(e) Tracking errors in position (m).



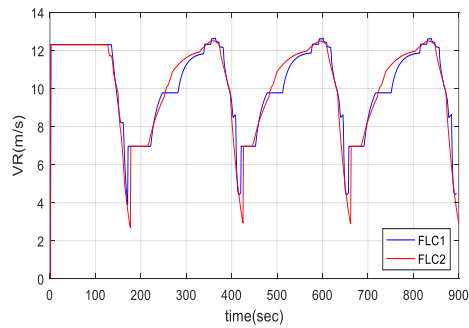
(f) Tracking errors in orientation (radian).



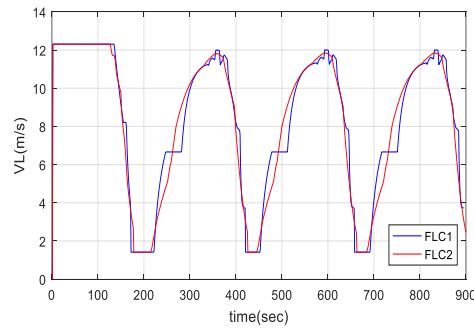
(g) Angular velocity (radian/s).



(h) Linear velocity (m/s).



(i) Linear velocity of the right wheel (m/s).



(j) Linear velocity of the left wheel (m/s).

**Figure V-3: Results obtained for the scenario: Multiple waypoints navigation.**



**Table V-2: Computational time comparison for the two controllers for multiple waypoints**

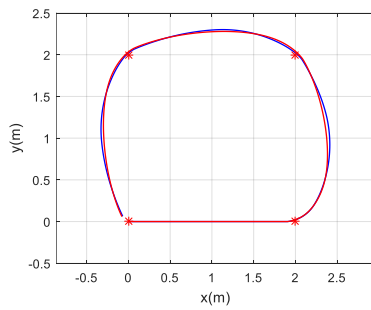
	<b>FLC1</b>	<b>FLC2</b>
<b>Mean (s)</b>	0.018	0.1285
<b>Std (s)</b>	0.0132	0.0184
<b>Number of iterations</b>	894	899

### V.2.3. Scenario 3 : Robustness analysis

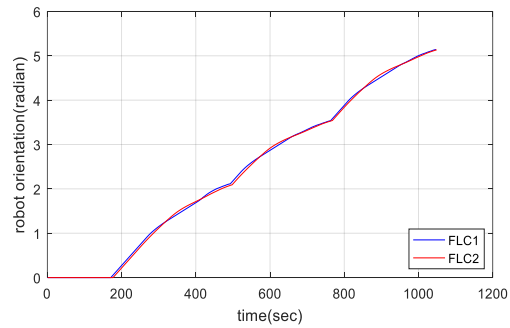
In this simulation, robustness of both controllers is evaluated face parameters uncertainties, modeling errors as well as localization errors.

#### V.2.3.1. Modeling error

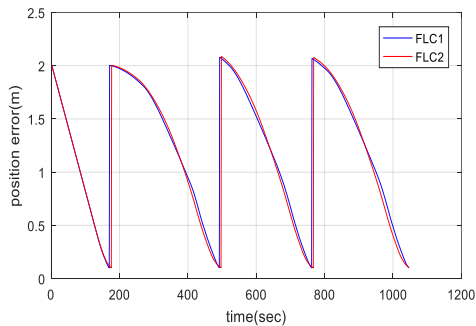
In the first scenario, we consider an uncertainty on the length L of the robot model:  $\Delta L = 30\%$  was added in order to analyze the robustness of the controllers face modeling errors, the results are obtained in the figure V-4.



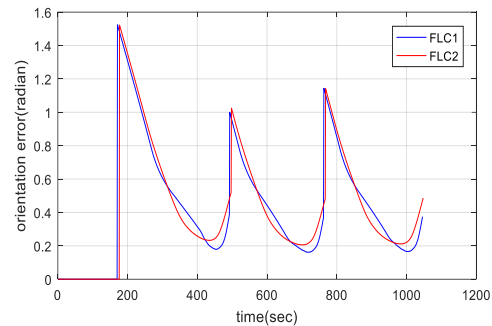
(a) Robot x/y position (m).



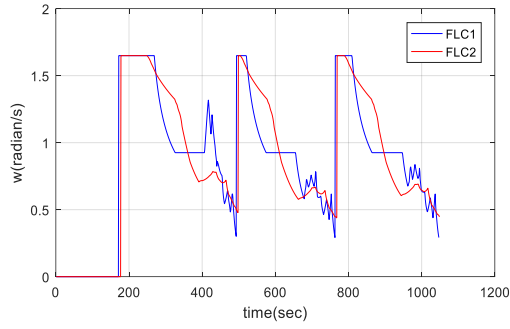
(b) Robot orientation (radian).



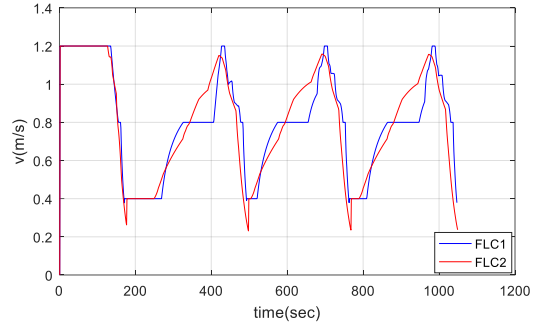
(c) Tracking errors in position (m).



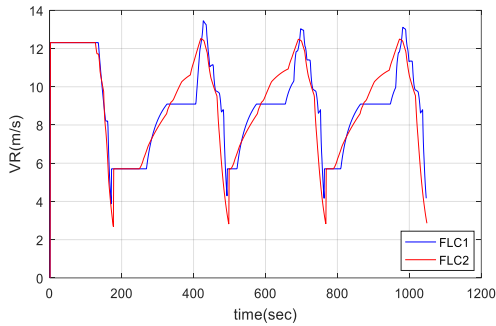
(d) Tracking errors in orientation (radian).



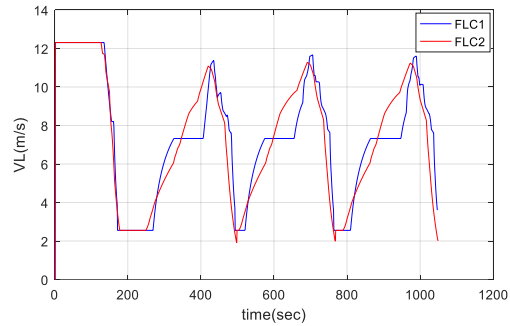
(e) Angular velocity (radian/s).



(f) Linear velocity (m/s).



(g) Linear velocity of the right wheel (m/s).

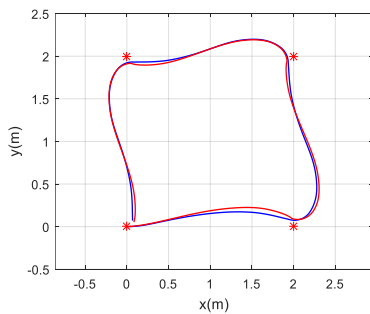


(h) Linear velocity of the left wheel (m/s).

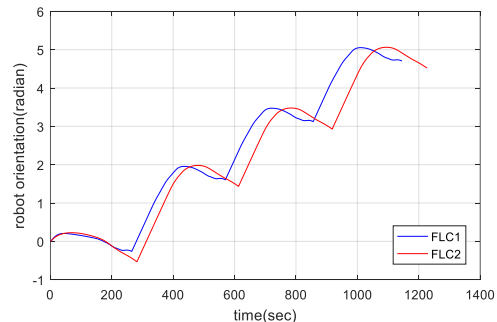
**Figure V-4: Results obtained for the scenario: Multiple waypoints navigation ( $\Delta L = 30\%$ ).**

From figure V-4, (a, b, c and d) the two controllers navigate to different waypoints successfully and similarly. We note that the fact of adding this uncertainty affect the two fuzzy controllers which drifts from its initial trajectory especially when new waypoint is considered.

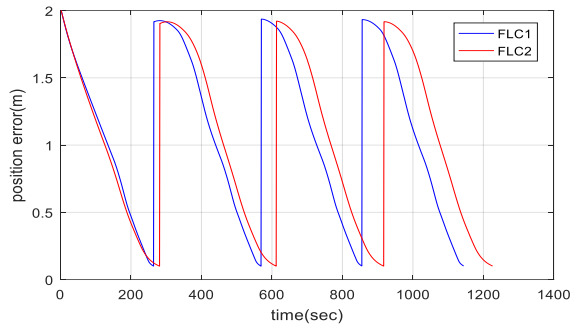
In order to observe well the effect of error modeling on the performances of waypoints navigation, we consider in the following experience another level of uncertainty on the radius  $R$  of the left wheel of the mobile robot with  $\Delta R = 25\%$  The obtained results are illustrated in figure V-5.



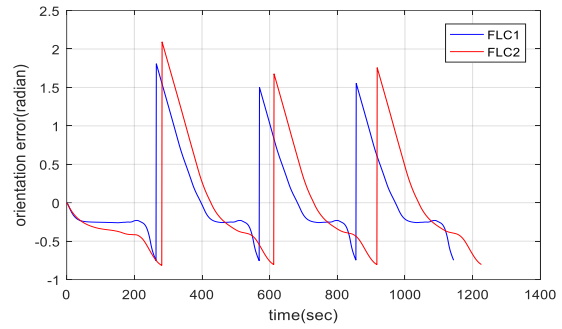
(a) Robot x/y position (m).



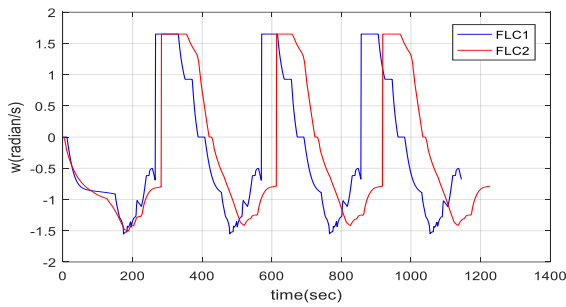
(b) Robot orientation (radian).



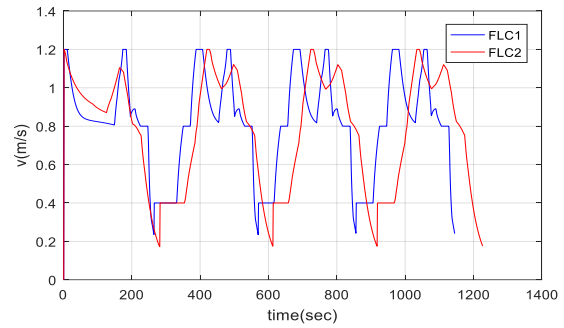
(c) Tracking errors in position (m).



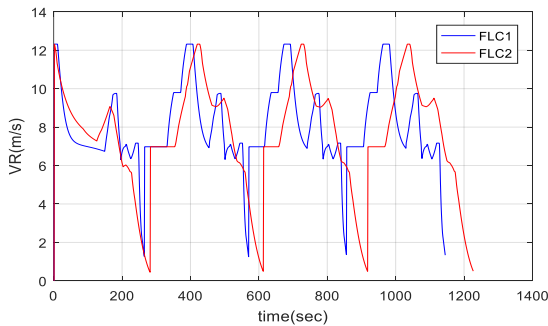
(d) Tracking errors in orientation (radian).



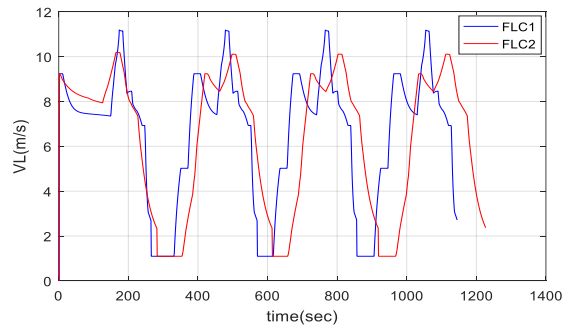
(e) Angular velocity (radian/s).



(f) Linear velocity (m/s).



(g) Linear velocity of the right wheel (m/s)



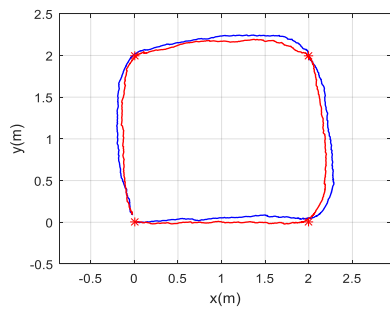
(h) Linear velocity of the left wheel (m/s).

**Figure V-5: Results obtained for the scenario: Multiple waypoints navigation( $\Delta R = 25\%$ ).**

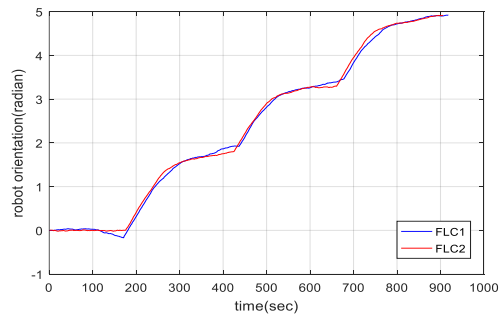
From figure V-5, (a, b, c and d) the two controllers navigate to different waypoints successfully and quiet similarly. We note that the fact of adding this uncertainty affect the two fuzzy controllers which drifts from its initial trajectory especially when new waypoint is considered. This is confirmed by the increase of the navigation time for 900 s without uncertainty to 1200 s with radius uncertainty( $\Delta R = 25\%$ ).

### V.2.3.2. Localization error

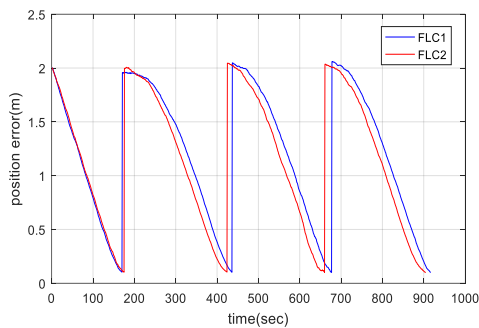
In this section, the robustness of controllers is evaluated face localization error. In order to better observe the effect of localization error on the performances of waypoints navigation, we consider that the differential drive mobile robot has an optical encoder for each wheel. The overall scheme for the localization and position control of a differential drive mobile is shown in figure V-1. An odometry module for a mobile robot with noisy position estimation is used in this experiment.



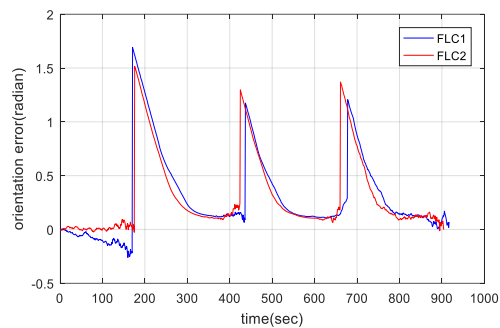
(a) Robot x/y position (m).



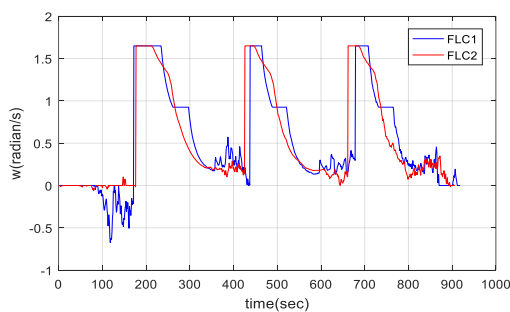
(b) Robot orientation (radian).



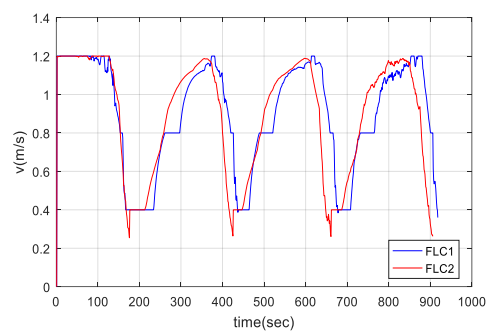
(c) Tracking errors in position (m).



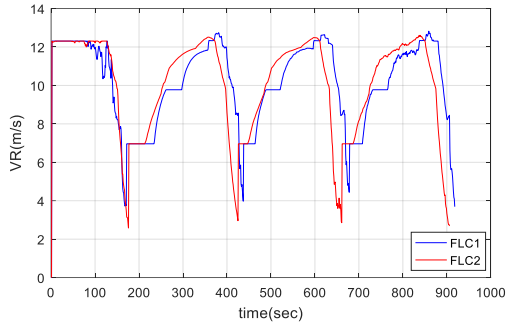
(d) Tracking errors in orientation (radian).



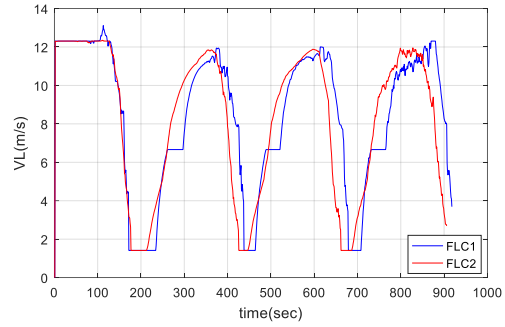
(e) Angular velocity (radian/s).



(f) Linear velocity (m/s).



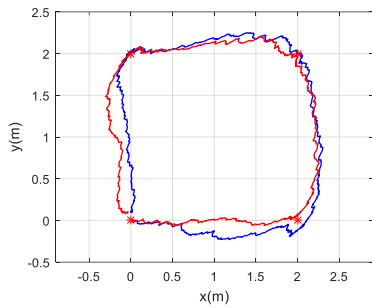
(g) Linear velocity of the right wheel (m/s)



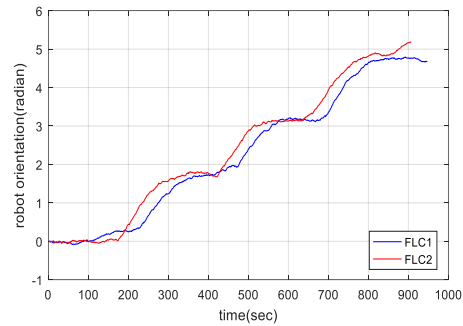
(h) Linear velocity of the left wheel (m/s).

**Figure V-6: Results obtained for the scenario: Multiple waypoints (uncertainties of localization  $\alpha = 0.003$ ).**

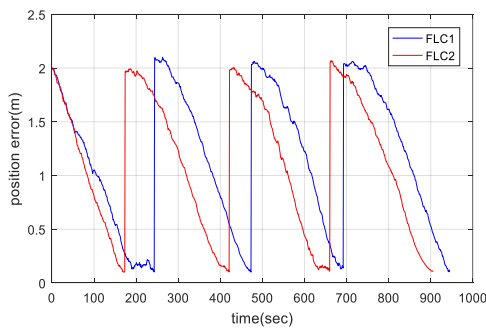
From figure V-6, (a, b, c and d) the two controllers navigate to different waypoints quiet similarly. We note that the fact of considering noisy position affect the two fuzzy controllers (Figure V-6 e and f) especially when new waypoint is considered.



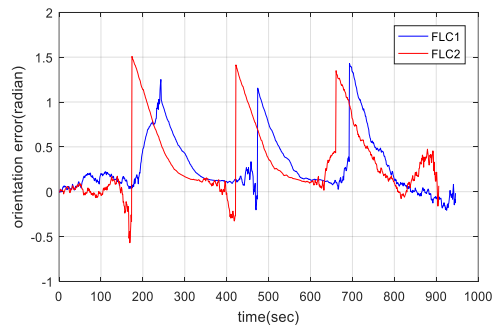
(a) Robot x/y position (m).



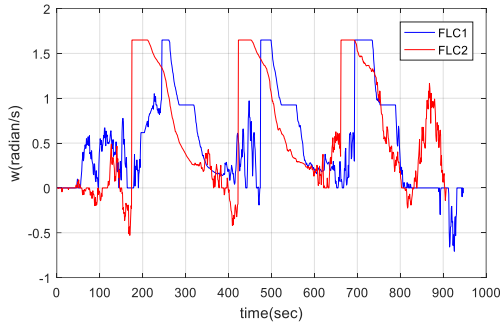
(b) Robot orientation (radian).



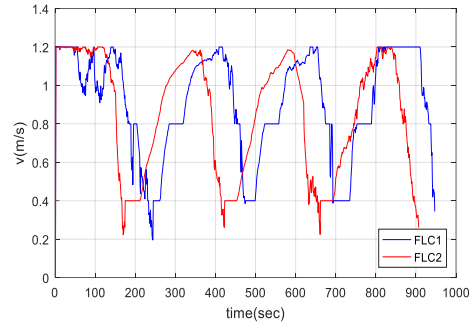
(c) Tracking errors in position (m).



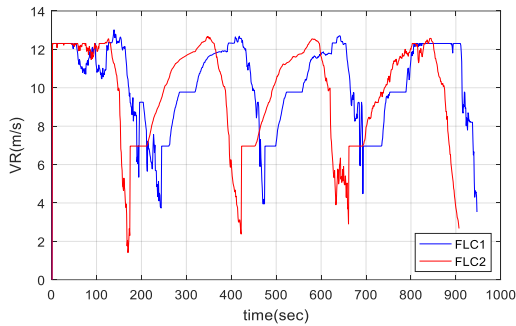
(d) Tracking errors in orientation (radian).



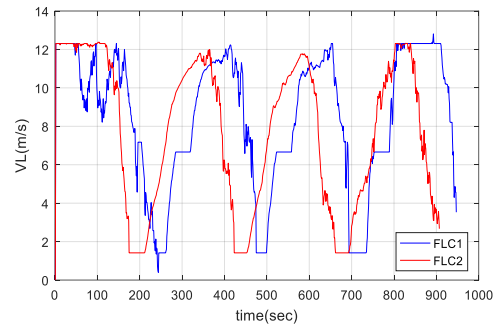
(e) Angular velocity (radian/s).



(f) Linear velocity (m/s).



(g) Linear velocity of the right wheel (m/s)



(h) Linear velocity of the left wheel (m/s).

**Figure V-7: Results obtained for the scenario: Multiple waypoints navigation (uncertainties of localization  $\alpha = 0.009$ ).**

From figure V-7, (a, b, c and d) the two controllers navigate to different waypoints following noisy trajectories especially with the FLC1. Also, we note from (Figure V-7 e and f) that the fact of considering noisy position affect significantly the angular velocity provided by FLC1 and FLC2 especially when new waypoint is considered. We can observe that the FLC2 require less navigation time comparing to the FLC1 to navigate to different waypoints.

### V.3. Results of dynamic model

In the previous work, the implemented controllers consider only the kinematic model of the mobile robot and “perfect velocity” tracking is assumed to generate the current vehicle control inputs. There are many problems with this approach; first, the perfect velocity tracking assumption does not hold in practice, second, disturbances are ignored. Thus, take in consideration the full dynamic model is very important for realistic simulation of the aforementioned controllers. We take into consideration the same scenarios. The overall control scheme of a dynamic model of DDWMR is shown in figure V-8.

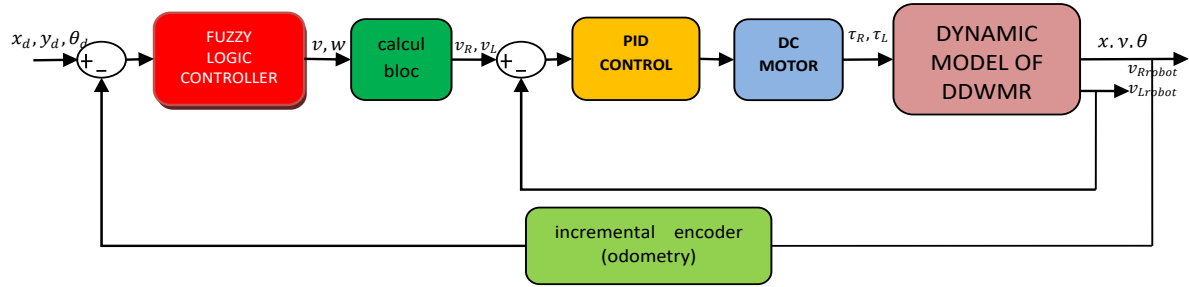
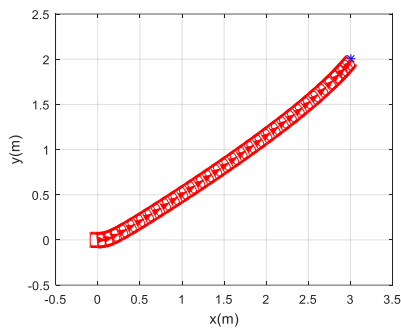
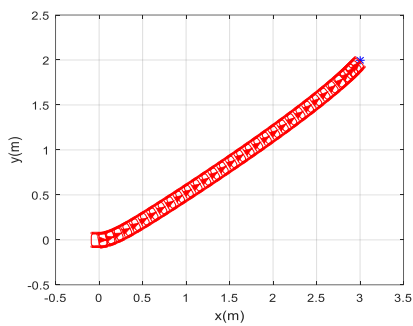


Figure V-8: Global scheme of control with dynamic model of DDWMR.

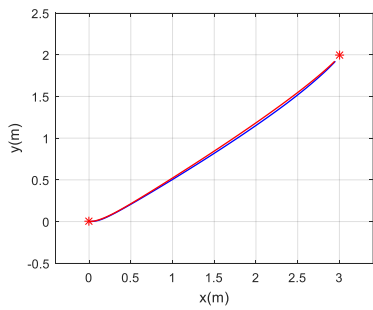
### V.3.1. Scenario 1: Go to goal (single waypoint)



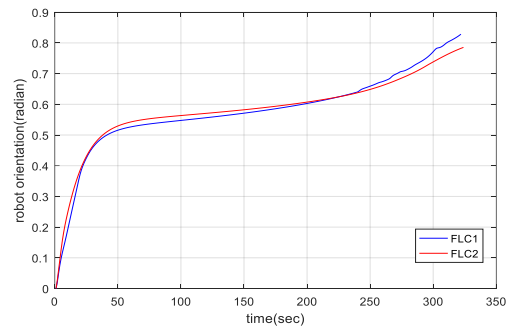
(a) Go to a goal navigation with FLC1 controller.



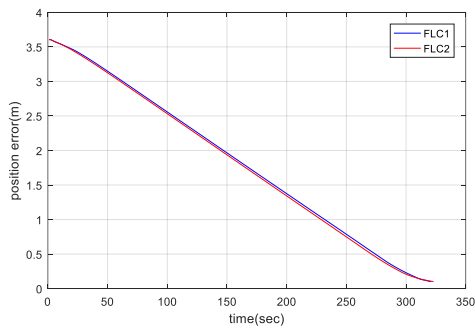
(b) Go to a goal navigation with FLC2 controller.



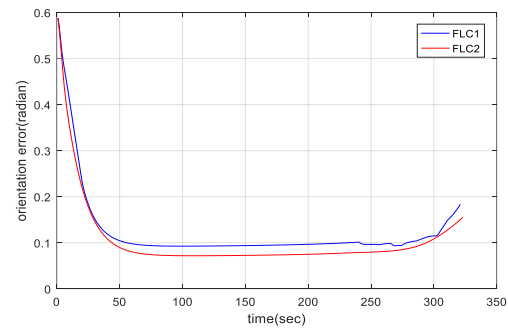
(c) Robot x/y position (m).



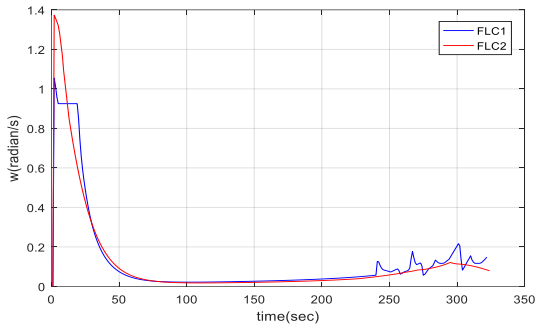
(d) Robot orientation (radian).



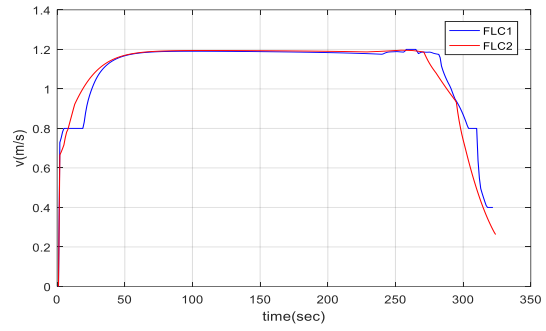
(e) Tracking errors in position (m).



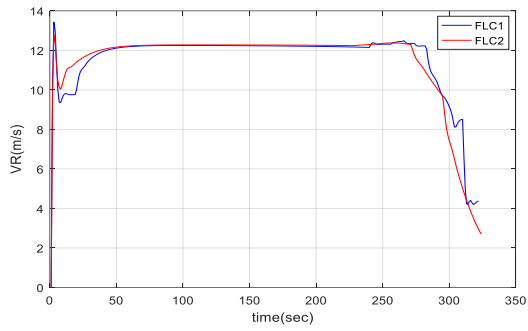
(f) Tracking errors in orientation (radian).



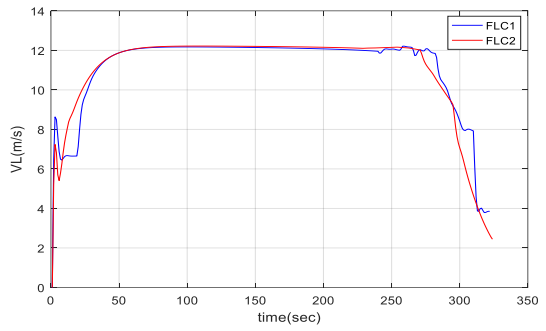
(g) Angular velocity (radian/s).



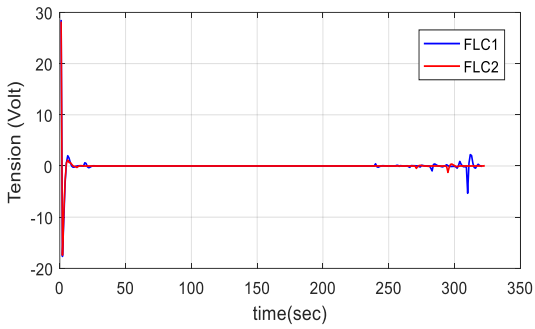
(h) Linear velocity (m/s).



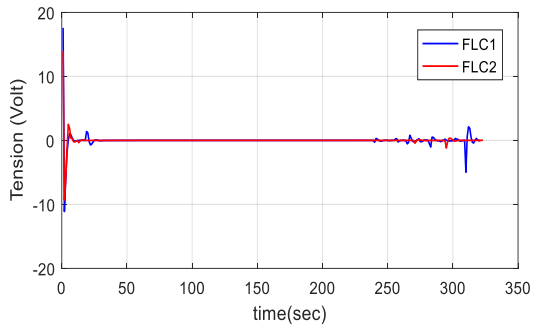
(i) Linear velocity of the right wheel (m/s).



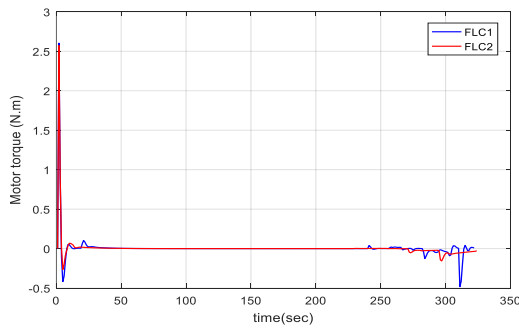
(j) Linear velocity of the left wheel (m/s).



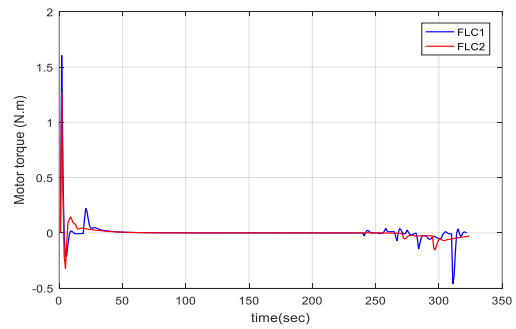
(k) Right motor tension (volt).



(l) Left motor tension (volt).



(m) Right motor torque (N.m).



(n) Left motor torque (N.m).

**Figure V-9: Results obtained for the scenario: Go to goal.**



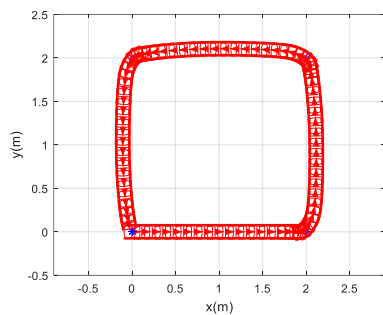
As can be seen from figure V-9, the two controllers FLC1 and FLC2 provide similar results. The table V-3 confirms this result, where computational time of both controllers is compared.

**Table V-3: Computational time comparison for the two controllers.**

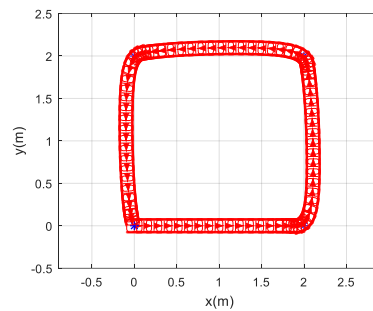
	<b>FLC1</b>	<b>FLC2</b>
<b>Mean (s)</b>	0.0179	0.1272
<b>Std (s)</b>	0.0133	0.0194
<b>Number of iterations</b>	321	323

According to table V-3, the required time of the mobile robot to reach the final goal by the FLC1 controller is shorter than that required for FLC2 controller and this is explained by the fact that the latter has the additional step of reducing type.

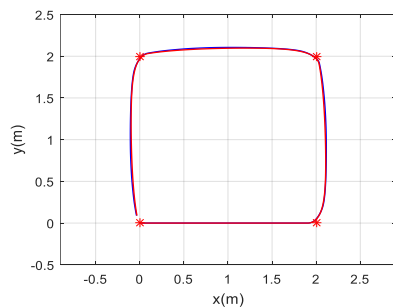
### V.3.2. Scenario 2 : Multiple waypoints navigation



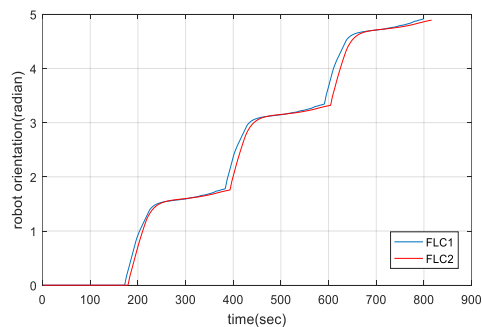
(a) Waypoints navigation with FLC1 controller



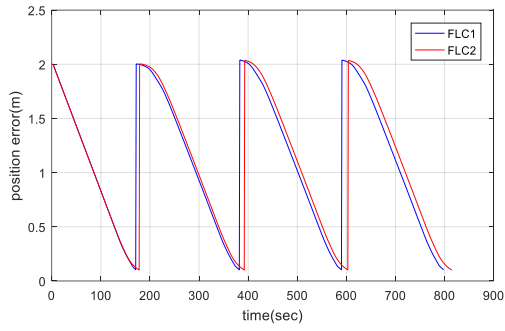
(b) Waypoints navigation with FLC2 controller .



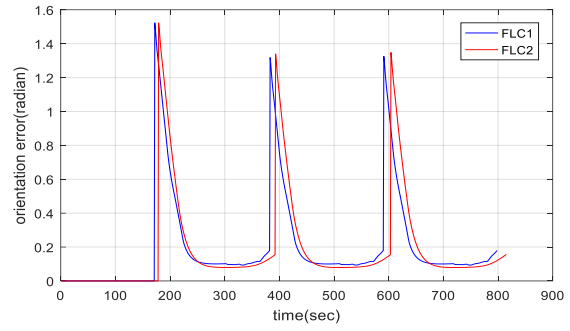
(c) Robot x/y position (m)



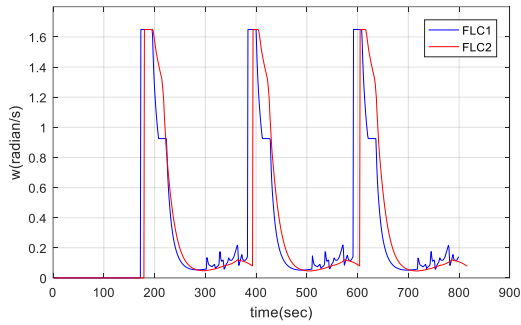
(d) Robot orientation (radian).



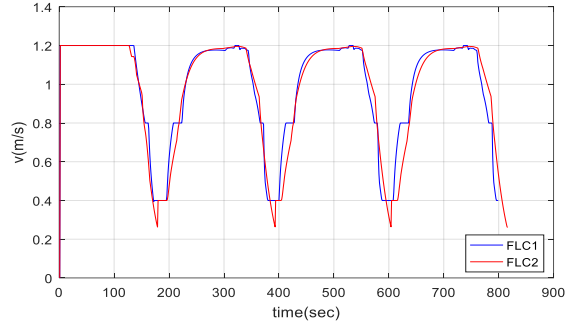
(e) Tracking errors in position (m).



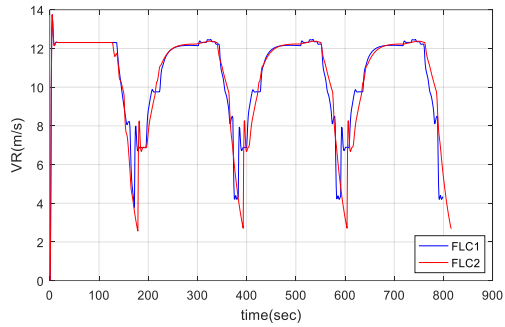
(f) Tracking errors in orientation (radian).



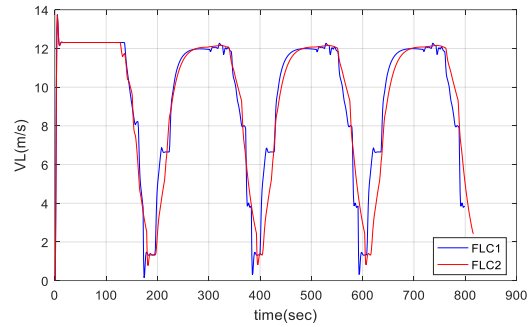
(g) Angular velocity (radian/s).



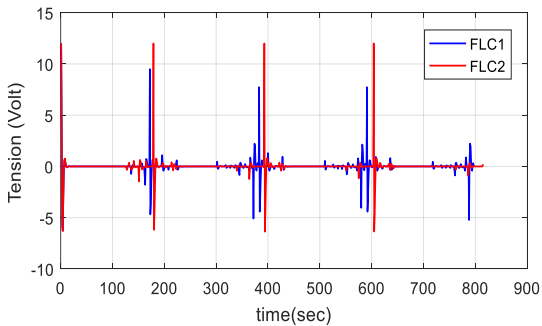
(h) Linear velocity (m/s).



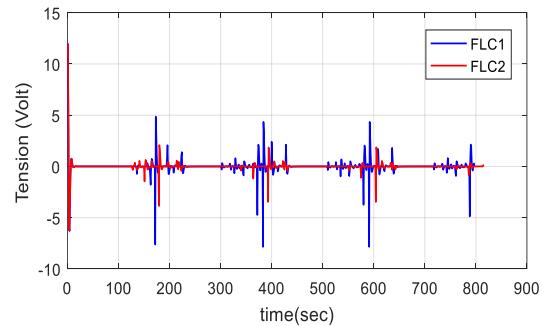
(i) Linear velocity of the right wheel (m/s).



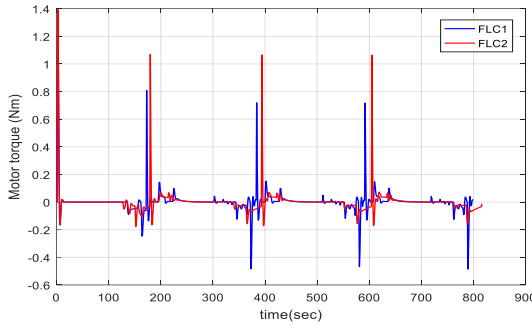
(j) Linear velocity of the left wheel (m/s).



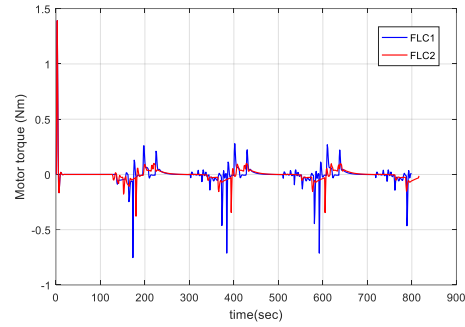
(k) Right motor tension (volt).



(l) Left motor tension (volt).



(m) Right motor torque (N.m).



(n) Left motor torque (N.m).

**Figure V-10: Results obtained for the scenario: Multiple waypoints navigation.**

**Table V-4: Computational time comparison for the two controllers for multiple waypoints.**

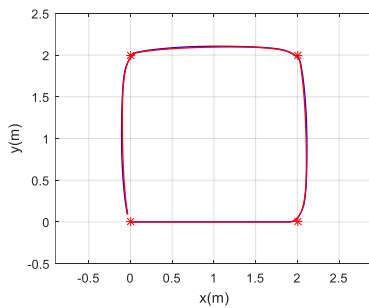
	<b>FLC1</b>	<b>FLC2</b>
<b>Mean (s)</b>	0.0187	0.1307
<b>Std (s)</b>	0.014	0.0174
<b>Number of iterations</b>	797	814

From figure V-10, (a, b, c and d) the two controllers navigate to different waypoints successfully and similarly. According to table V-4, the required time of the mobile robot to reach the final goal by FLC1 controller is less than the required time by FLC2 controller and this is explained also by the fact that the latter has the additional step of reducing type.

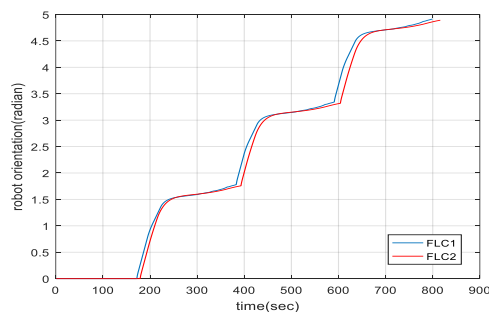
### V.3.3. Scenario 3 : Robustness analysis

#### V.3.3.1. Modeling error

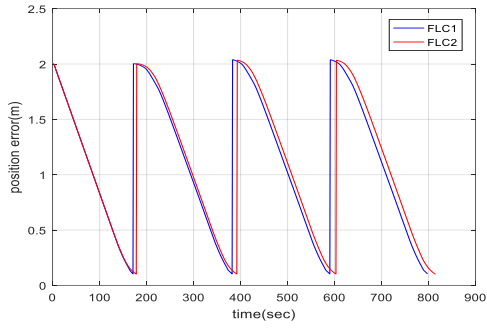
In this simulation, an uncertainty on the length  $L$  of the mobile robot model:  $\Delta L = 30\%$  was added in order to analyze the robustness of the controllers face modeling errors, the results are obtained in the following figures:



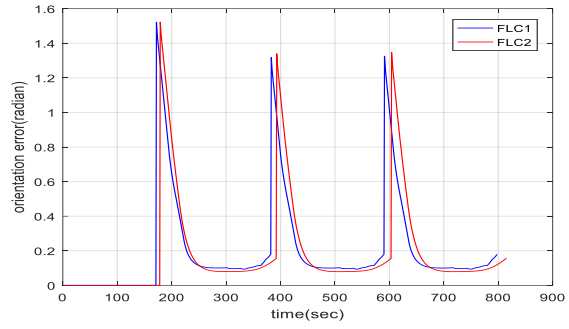
(a) Robot x/y position (m).



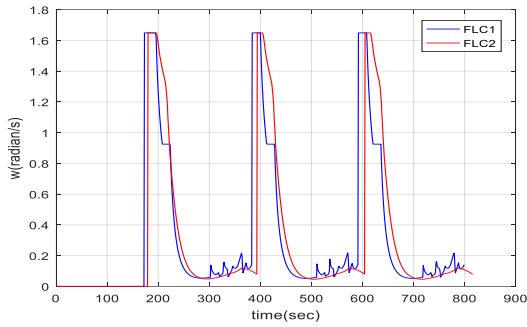
(b) Robot orientation (radian).



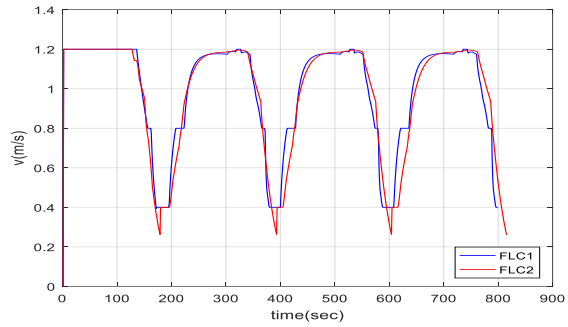
(c) Tracking errors in position (m).



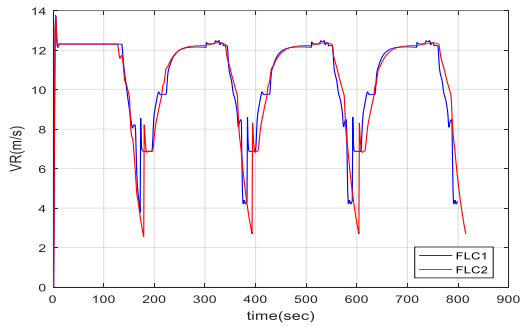
(d) Tracking errors in orientation (radian).



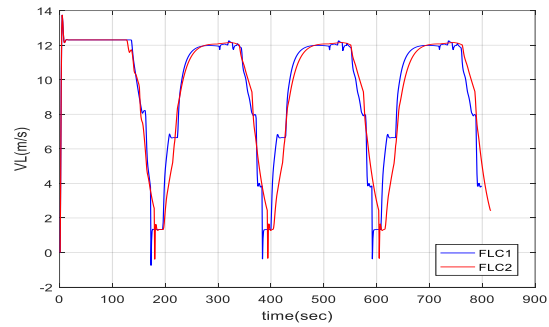
(e) Angular velocity (radian/s).



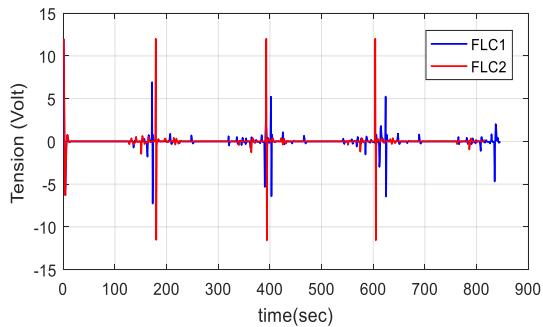
(f) Linear velocity (m/s).



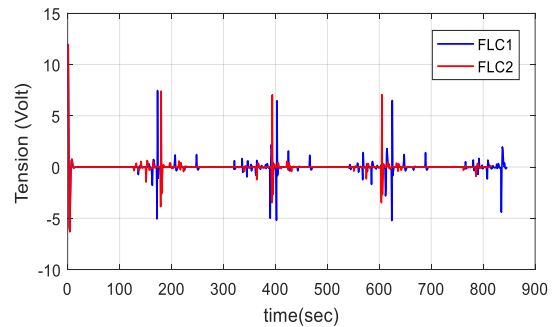
(g) Linear velocity of the right wheel (m/s).



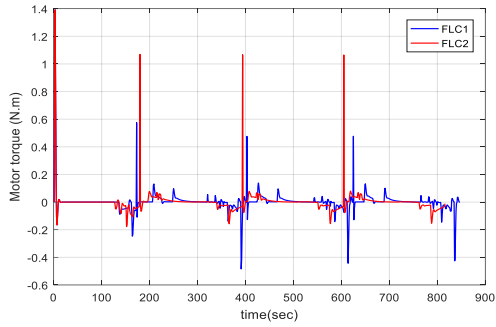
(h) Linear velocity of the left wheel (m/s).



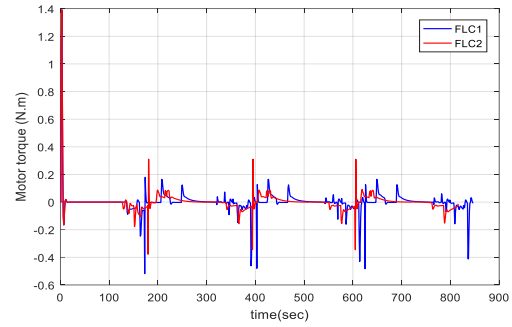
(k) Right motor tension (volt).



(l) Left motor tension (volt).



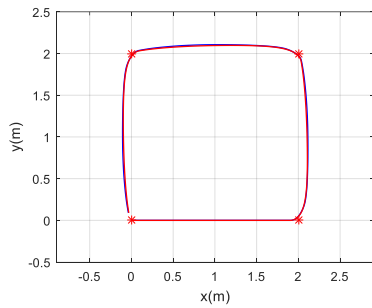
(m) Right motor torque (N.m)



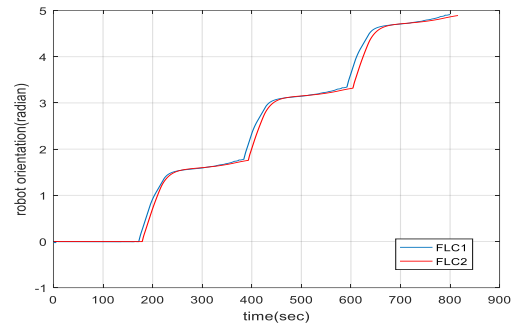
(n) Left motor torque (N.m).

**Figure V-11: Results obtained for the scenario: Multiple waypoints navigation ( $\Delta L = 30\%$ ).**

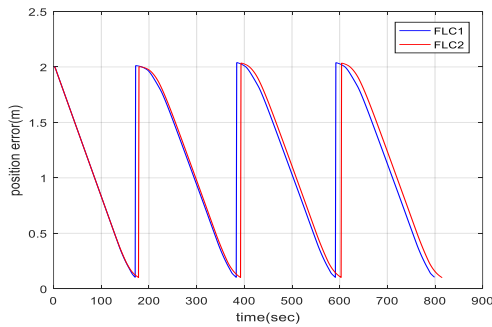
As shown in Figure V-11, both controllers presents suitable robustness face robot length uncertainty ( $\Delta L = 30\%$ ). We consider in the next experience a significant uncertainty on the radius R of the right wheel of the robot with  $\Delta RL = 25\%$ . The obtained results are illustrated in figure V-12.



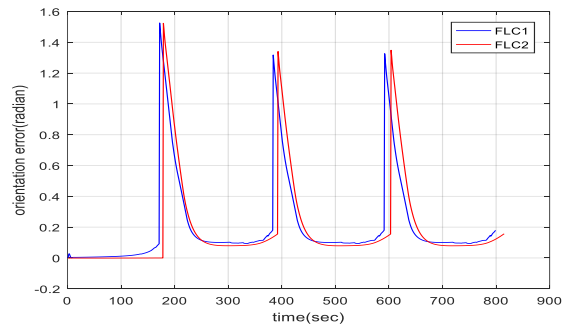
(a) Robot x/y position (m).



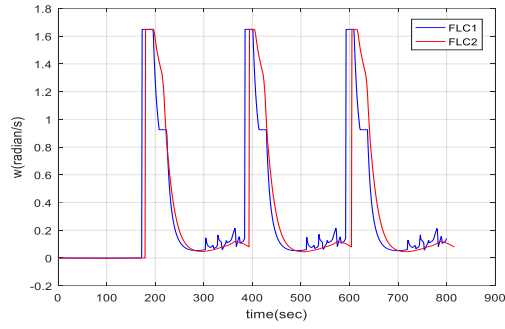
(b) Robot orientation (radian).



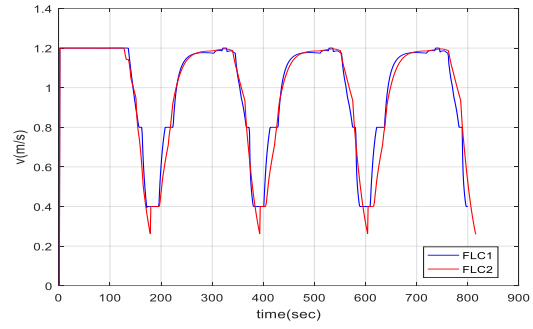
(c) Tracking errors in position (m).



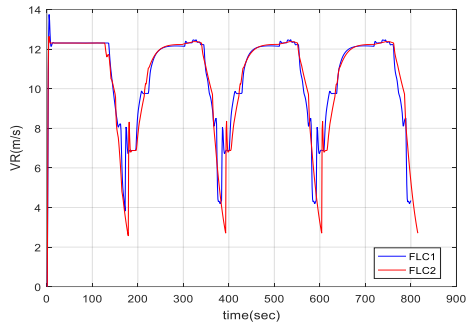
(d) Tracking errors in orientation (radian)



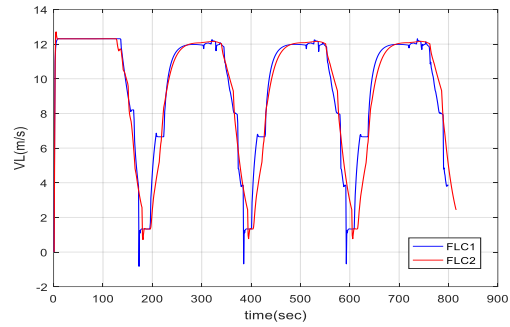
(e) Angular velocity (radian/s).



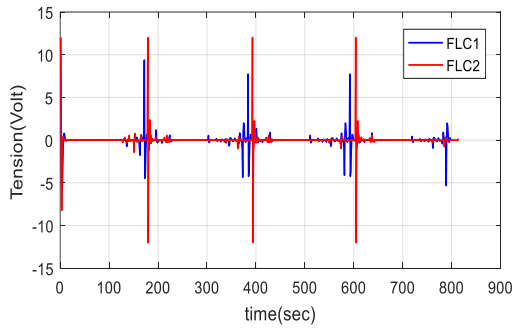
(f) Linear velocity (m/s).



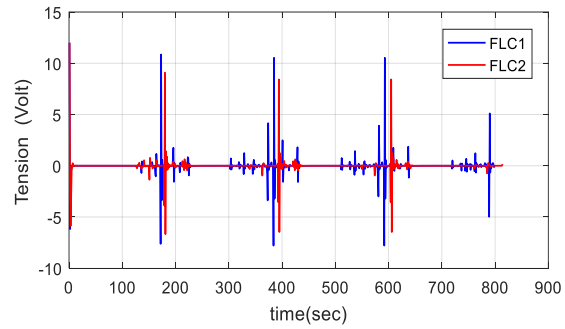
(g) Linear velocity of the right wheel (m/s).



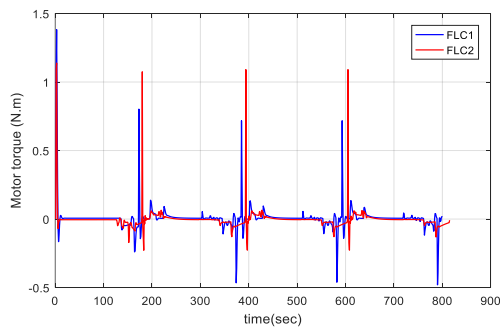
(h) Linear velocity of the left wheel (m/s).



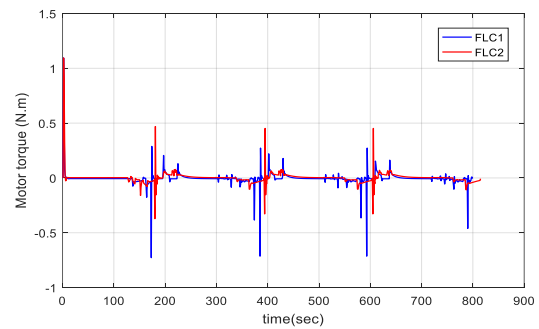
(k) Right motor tension (volt).



(l) Left motor tension (volt).



(m) Right motor torque (N.m).

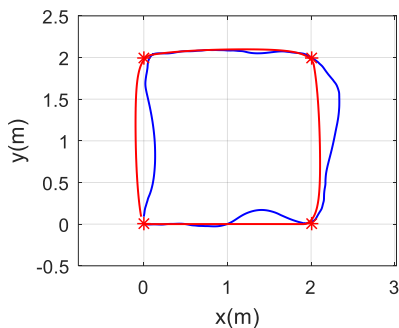


(n) Left motor torque (N.m).

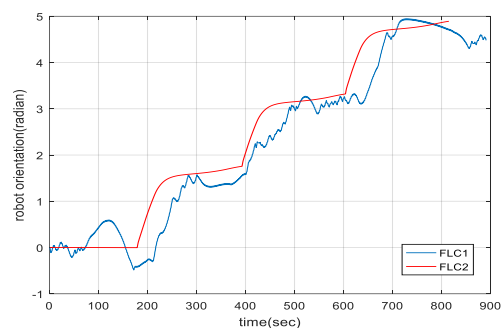
**Figure V-12: Results obtained for the scenario: Multiple waypoints navigation ( $\Delta R = 25\%$ ).**

From figure V-12, (a, b, c and d) the two controllers navigate to different waypoints successfully and quite similarly.

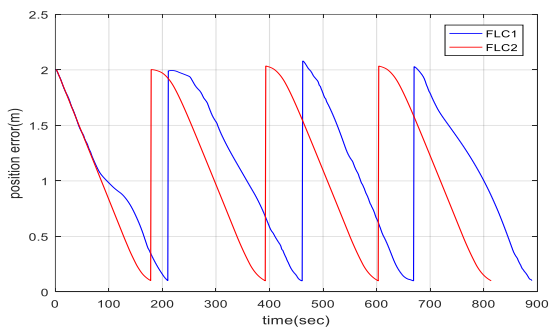
In this simulation, we increase the uncertainty of the right wheel radius  $R$  with  $\Delta RL = 50\%$ . The obtained results are illustrated in the following figures. As can be seen, the performances of the FLC1 based on type 1 fuzzy controller decreases significantly, in the other hand; good performances are obtained by the FLC2 based on interval type 2 fuzzy controller. Moreover, the mobile robot using FLC2 requires less navigation time to achieve different waypoints (Figure V-13.c and d). Also, less energy is required by the FLC2 comparing to the FLC 1 as illustrated in (Figure V-13.k and l).



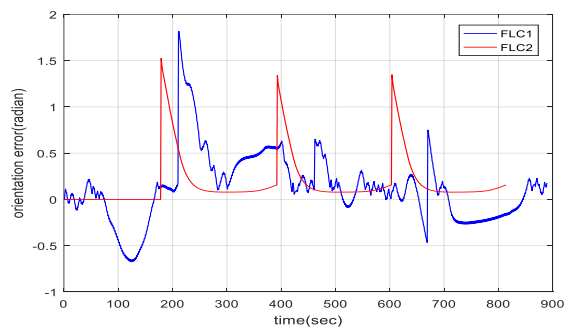
(a) Robot x/y position (m).



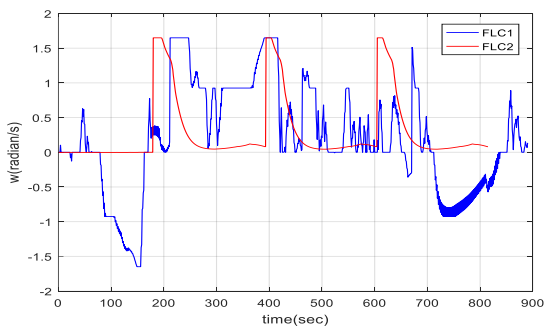
(b) Robot orientation (radian).



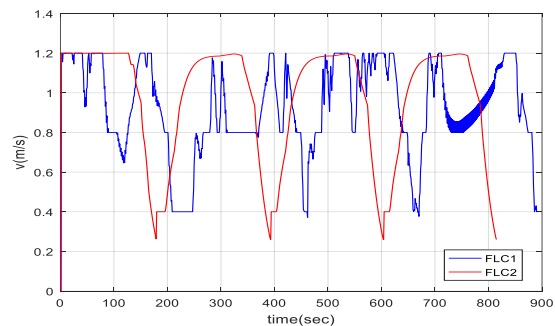
(c) Tracking errors in position (m).



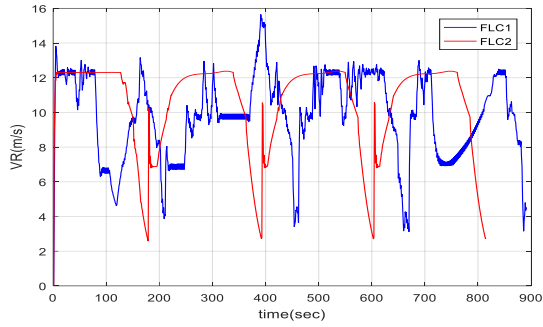
(d) Tracking errors in orientation (radian).



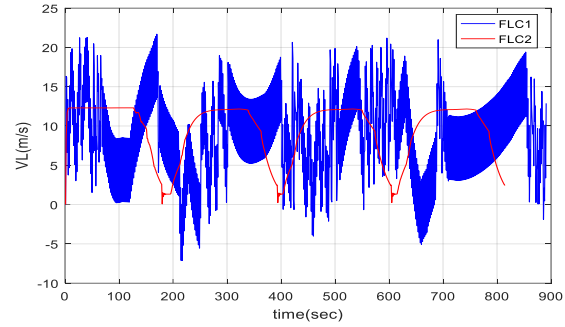
(e) Angular velocity (radian/s).



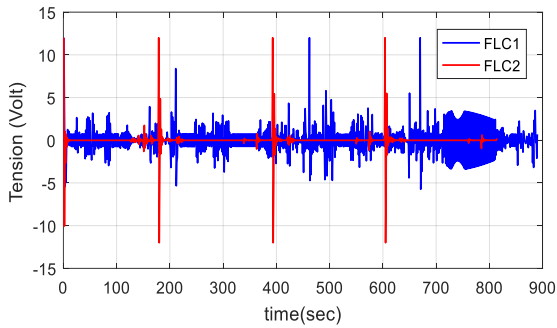
(f) Linear velocity (m/s).



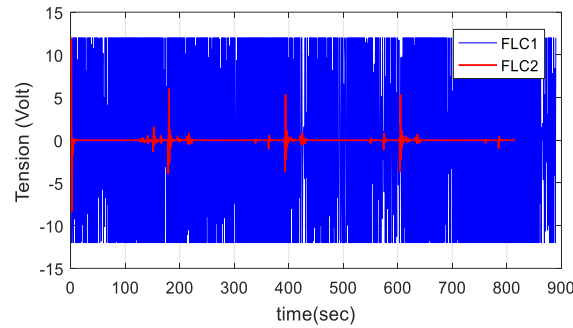
(g) Linear velocity of the right wheel (m/s)



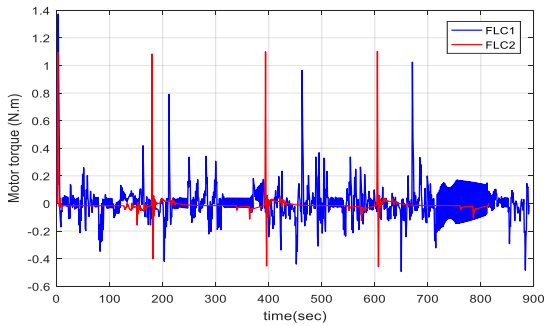
(h) Linear velocity of the left wheel (m/s).



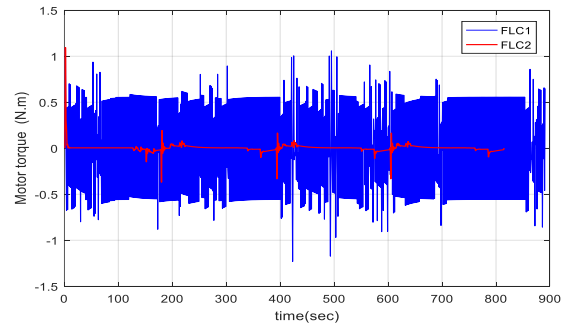
(k) Right motor tension (volt).



(l) Left motor tension (volt).



(m) Right motor torque (N.m).

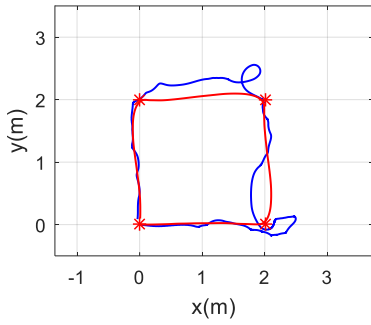


(n) Left motor torque (N.m).

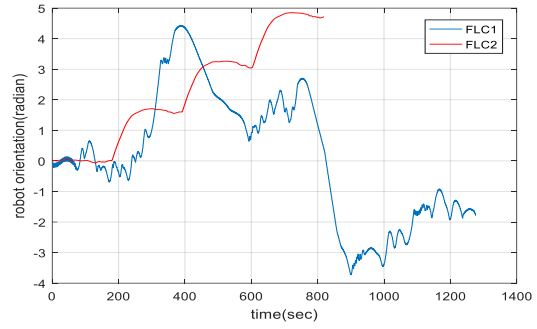
**Figure V-13: Results obtained for the scenario: Multiple waypoints navigation ( $\Delta R = 50\%$ ).**

In this simulation, we increase the uncertainty of the right wheel radius  $R$  with  $\Delta R = 60\%$ . The obtained results are illustrated in the figure V-14. As can be seen, the performances of the FLC1 based on type 1 fuzzy controller decreases significantly (long trajectory, increase of navigation time, more energy required), in the other hand; the FLC2 maintains suitable performances for good navigation. These results confirm the robustness of the Type 2 Fuzzy Logic controller face significant parameters uncertainties.

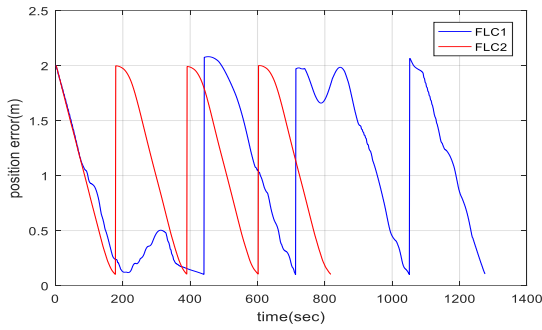




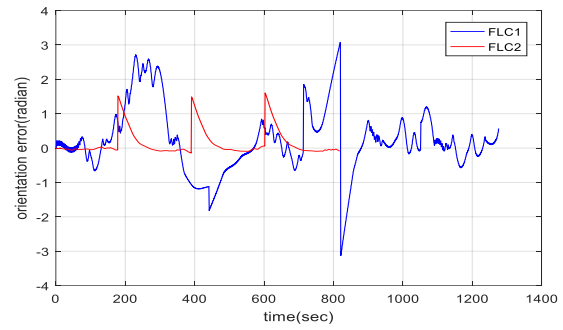
(a) Robot x/y position (m).



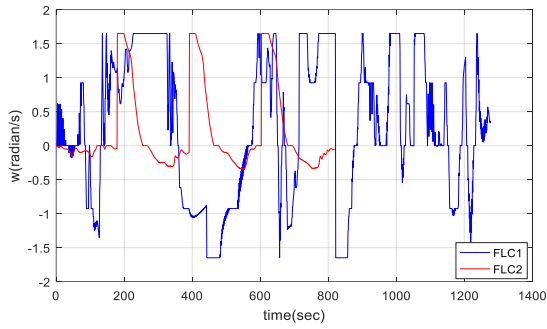
(b) Robot orientation (radian).



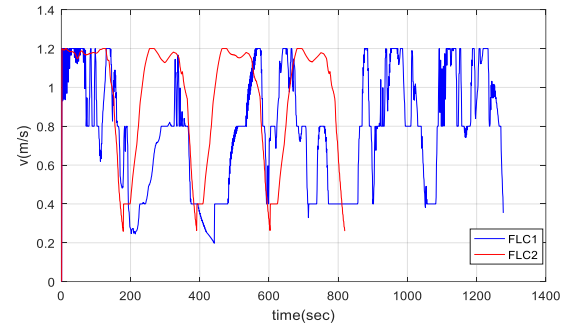
(c) Tracking errors in position (m).



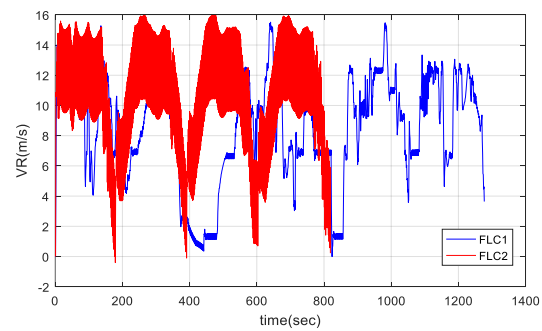
(d) Tracking errors in orientation (radian).



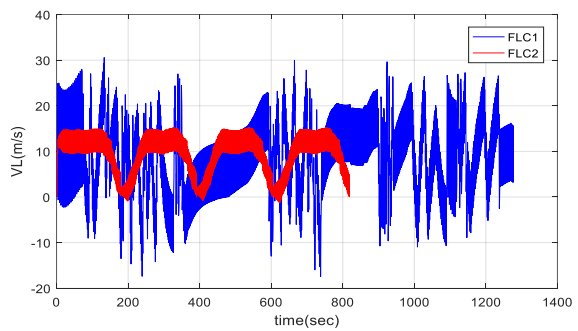
(e) Angular velocity (radian/s).



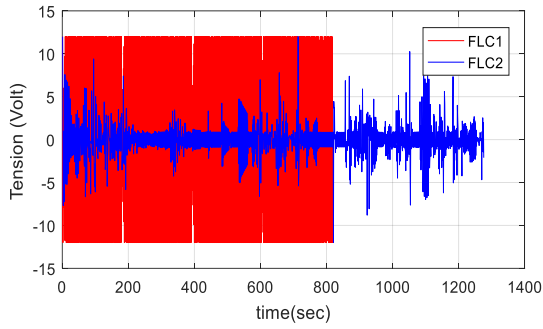
(f) Linear velocity (m/s).



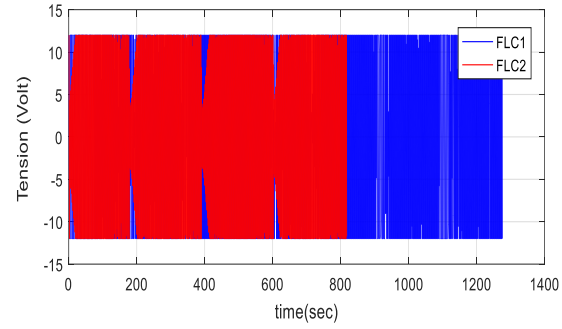
(g) Linear velocity of the right wheel (m/s).



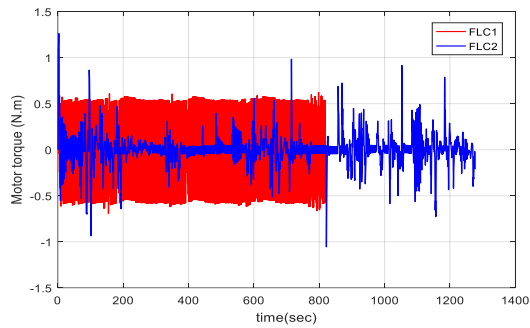
(h) Linear velocity of the left wheel (m/s).



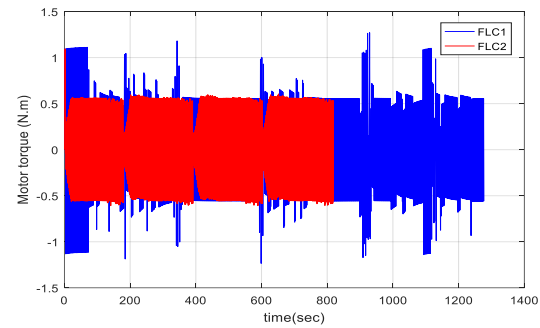
(k) Right motor tension (volt).



(l) Left motor tension (volt).



(m) Right motor torque (N.m).



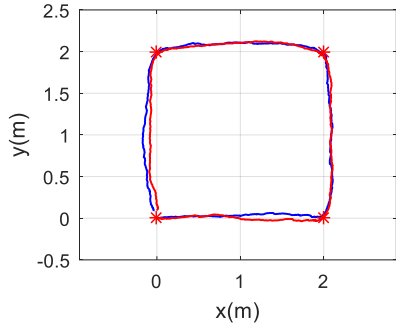
(n) Left motor torque (N.m).

**Figure V-14: Results obtained for the scenario: Multiple waypoints navigation( $\Delta R_L = 60\%$ ).**

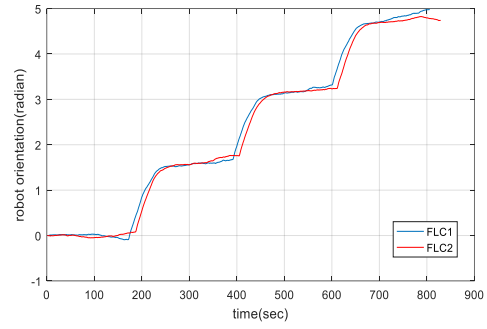
### V.3.3.2. Localization errors

In this section, the robustness of controllers is evaluated face localization error using the full dynamic model of the robot. In order to better observe the effect of localization error on the performances of waypoints navigation, we consider that the differential drive mobile robot has an optical encoder for each wheel. The overall scheme for the localization and position control of a differential drive mobile robot is shown in figure V-8, an odometry module for a mobile robot with noisy position estimation is considered.

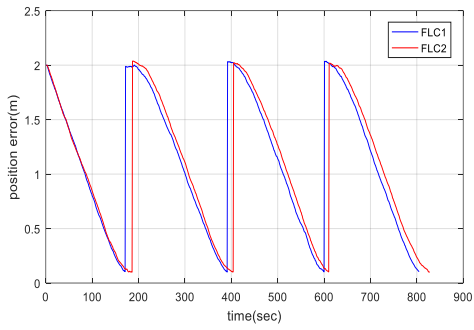
As can be seen from Figure V-15, both controllers provide good results.



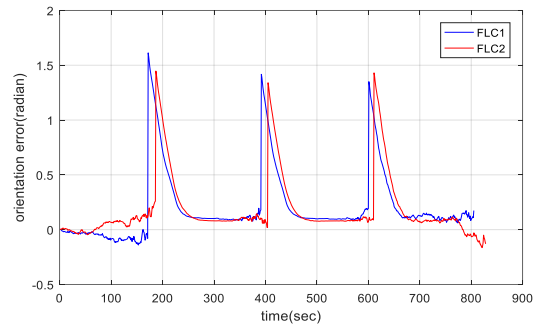
(a) Robot x/y position (m).



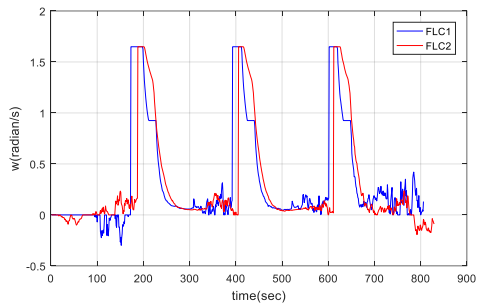
(b) Robot orientation (radian).



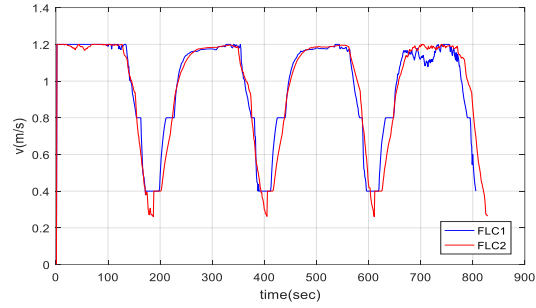
(c) Tracking errors in position (m).



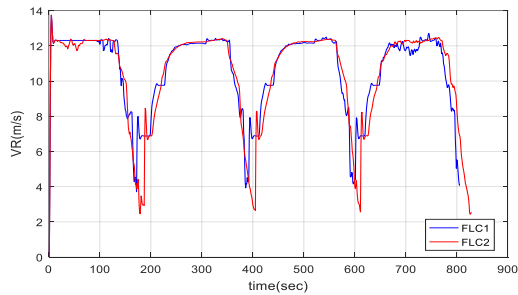
(d) Tracking errors in orientation (radian).



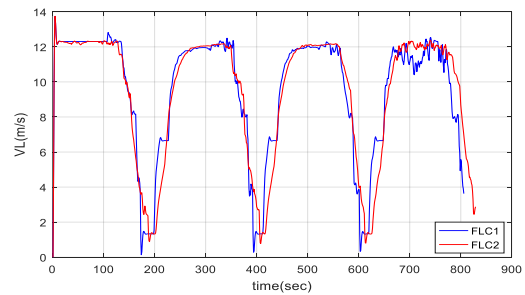
(e) Angular velocity (radian/s).



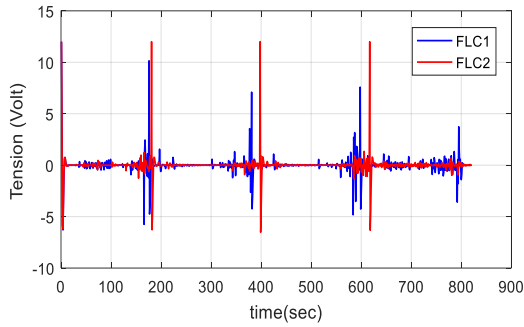
(f) Linear velocity (m/s).



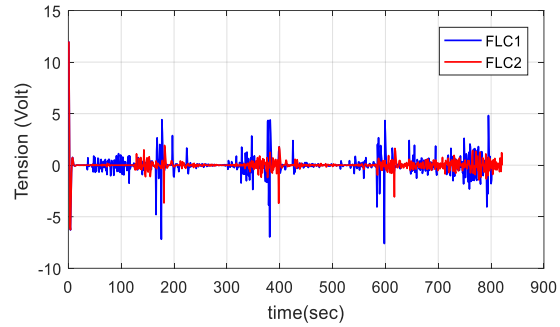
(g) Linear velocity of the right wheel (m/s)



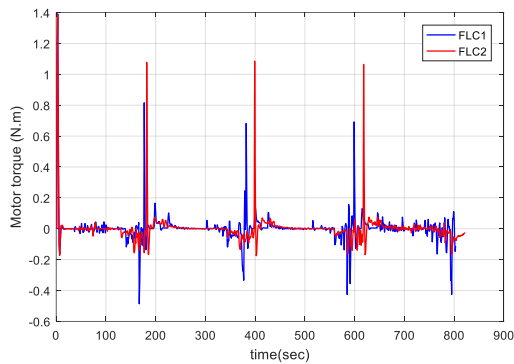
(h) Linear velocity of the left wheel (m/s).



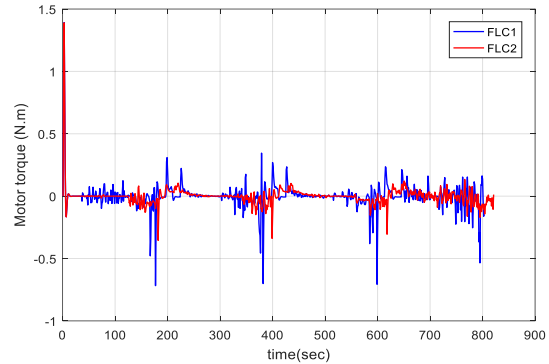
(k) Right motor tension (volt).



(l) Left motor tension (volt).



(m) Right motor torque (N.m).

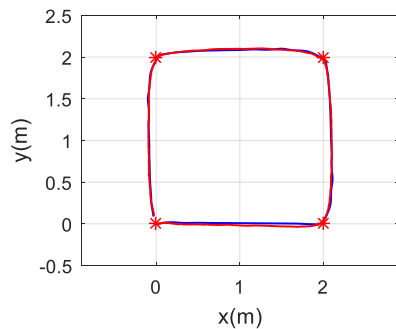


(n) Left motor torque (N.m).

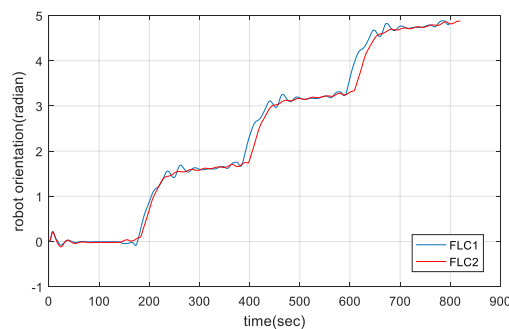
**Figure V-15: Results obtained for the scenario: Multiple waypoints navigation (uncertainties of localization  $\alpha = 0.003$ ).**

### V.3.3.3. Loss of efficiency

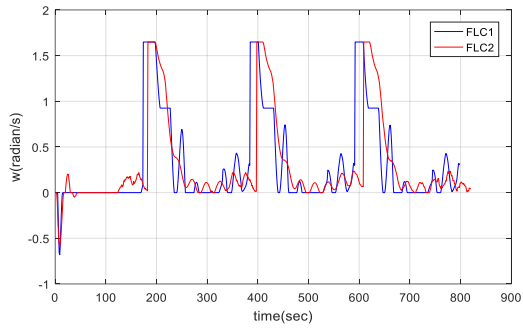
In this section, the robustness of controllers is evaluated face loss of efficiency of one or two motors using the full dynamic model of the robot. In order to better observe the effect of Loss of efficiency on the performances of waypoints navigation we consider significant decrease of DC motor torque. The loss of efficiency of left and right motors can be modeled by two coefficients  $\alpha_L$  and  $\alpha_R \in [0, 1]$  respectively.



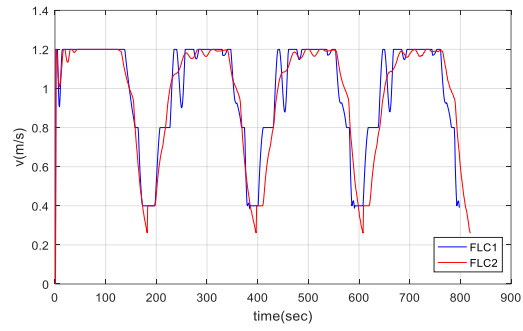
(a) Robot x/y position (m).



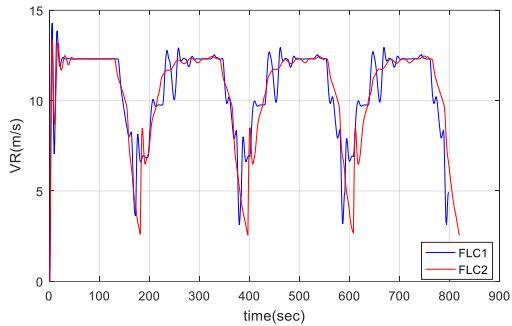
(b) Robot orientation (radian).



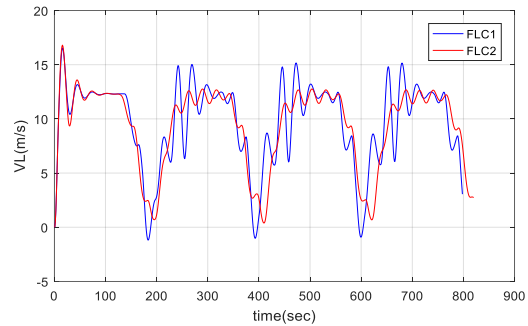
(c) Angular velocity (radian/s).



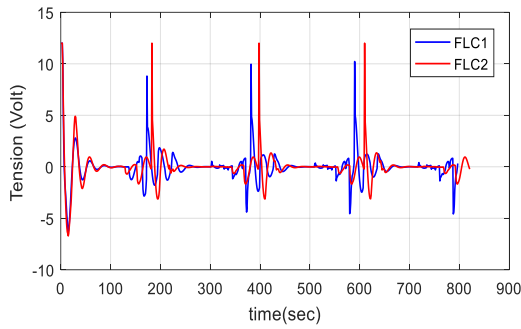
(d) Linear velocity (m/s).



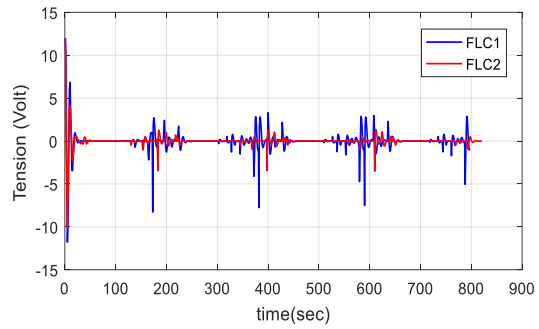
(e) Linear velocity of the right wheel (m/s).



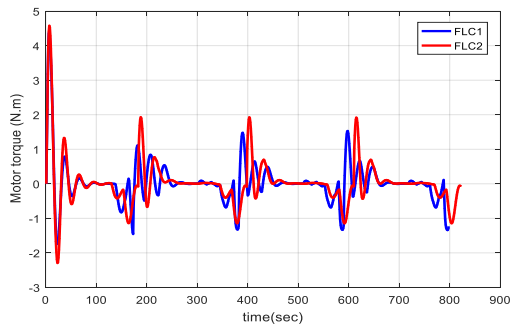
(f) Linear velocity of the left wheel (m/s).



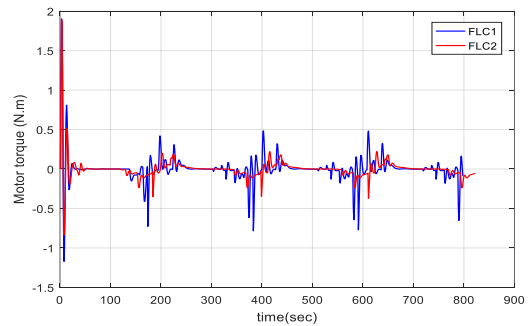
(k) Right motor tension (volt).



(l) Left motor tension (volt).



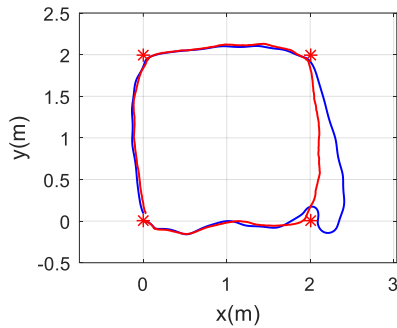
(m) Right motor torque (N.m).



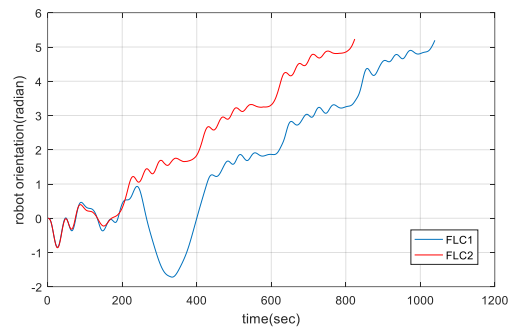
(n) Left motor torque (N.m).

**Figure V-16: Results obtained for the scenario: Multiple waypoints navigation (loss of efficiency of motors  $\alpha_L = 0.6$ ,  $\alpha_R = 0.1$ ).**

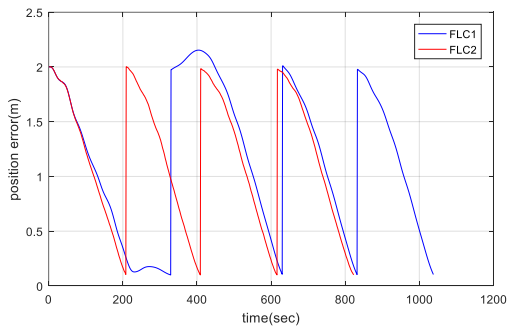
Figure V-17 represents the obtained results with significant loss of efficiency ( $\alpha_L=0.05$ ,  $\alpha_R=0.005$ ). As can be seen from this figure, FLC2 (IT2 FLC) performs much better than FLC1 (T1 FLC) (red trajectory in figure V-17 (a)). Moreover, the FLC2 based on type 2 fuzzy controller requires less time to achieve the different waypoints comparing to the FLC1 (Figure V-17 (c, d)).



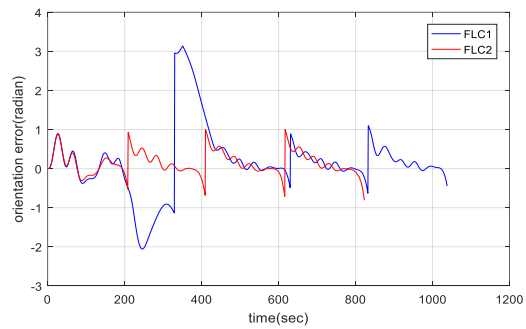
(a) Robot x/y position (m).



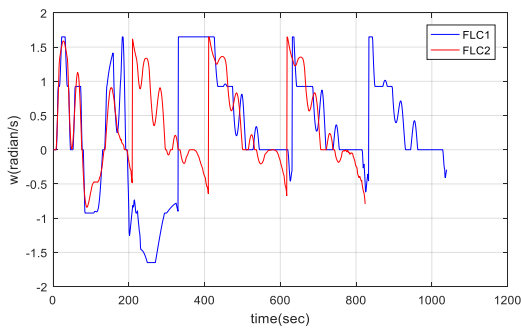
(b) Robot orientation (radian).



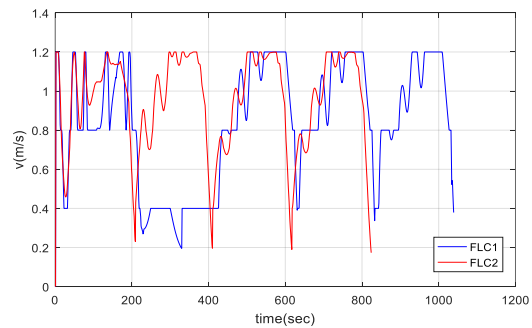
(c) Tracking errors in position (m).



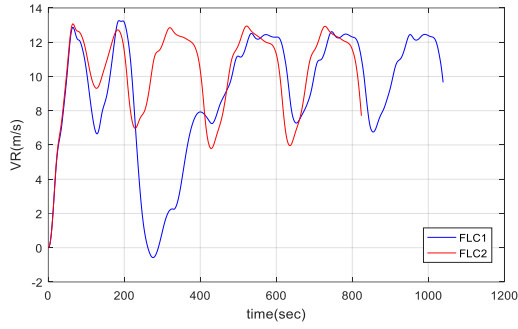
(d) Tracking errors in orientation (radian).



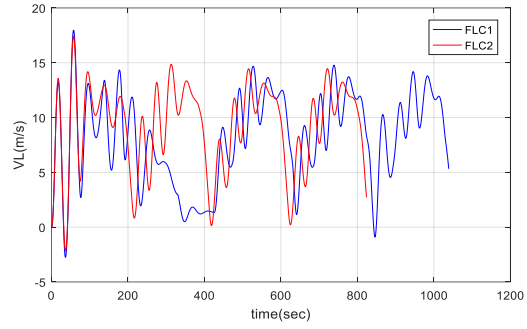
(e) Angular velocity (radian/s).



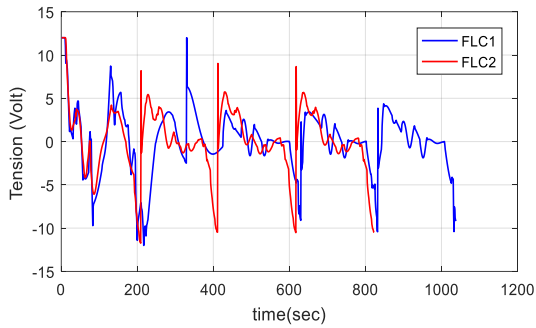
(f) Linear velocity (m/s).



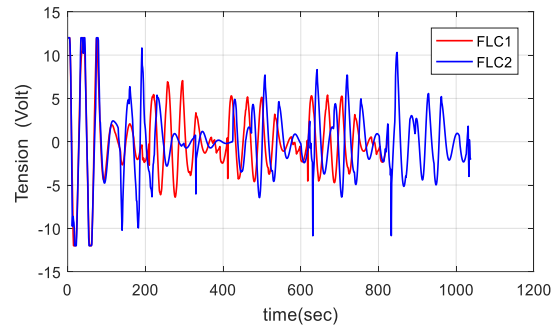
(g) Linear velocity of the right wheel (m/s).



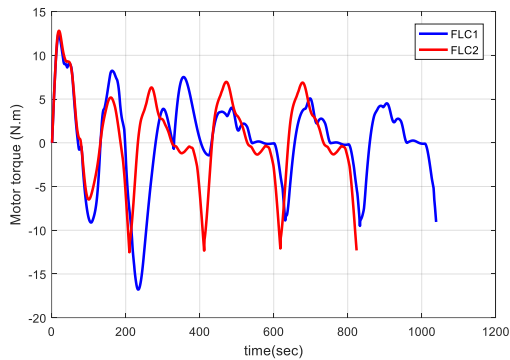
(h) Linear velocity of the left wheel (m/s).



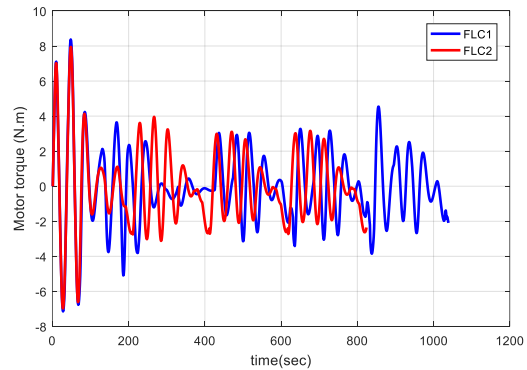
(k) Right motor tension (volt).



(l) Left motor tension (volt).



(m) Right motor torque (N.m).



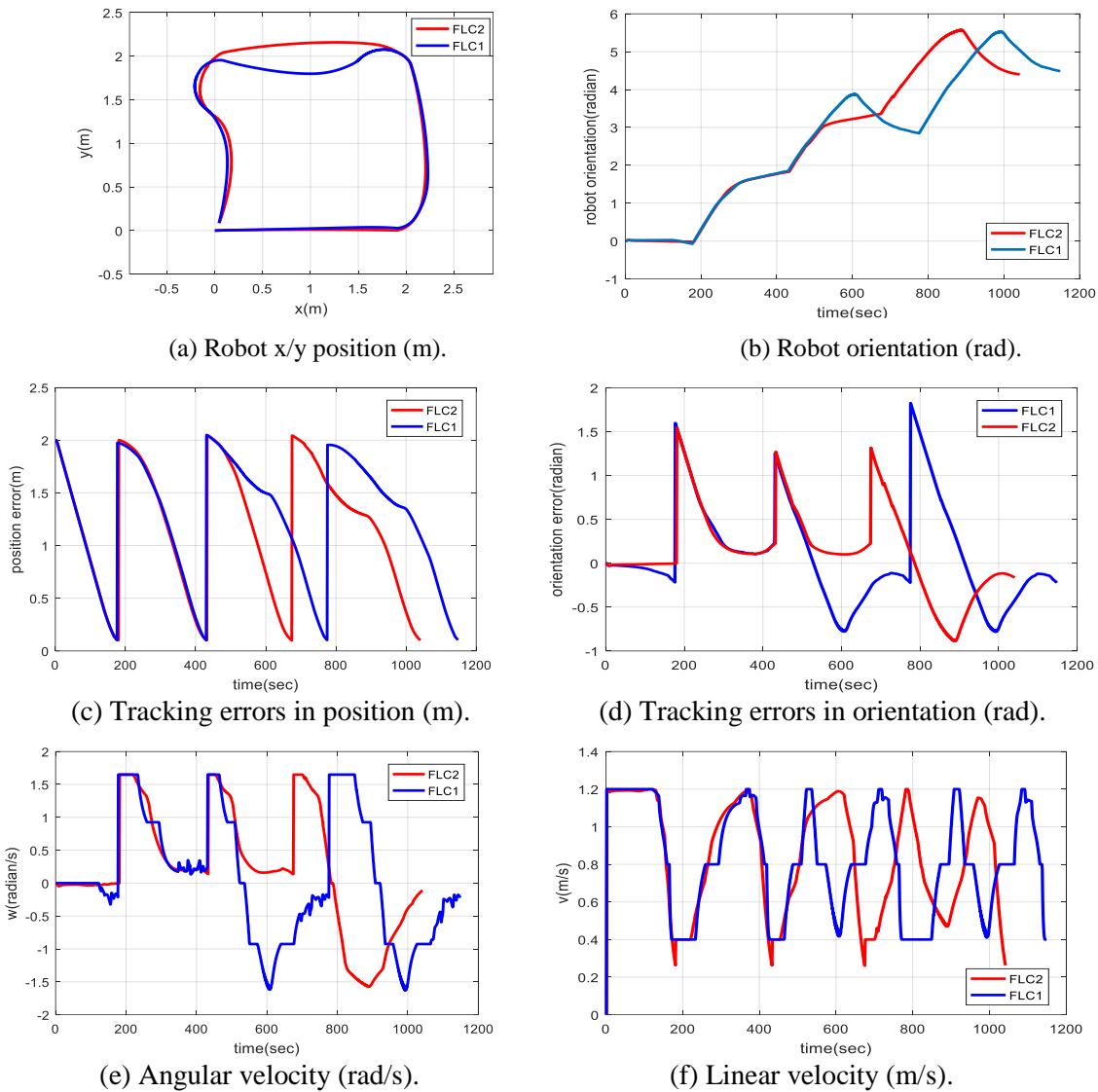
(n) Left motor torque (N.m).

**Figure V-17: Results obtained for the scenario: Multiple waypoints navigation (loss of efficiency of motors  $\alpha_L = 0.05, \alpha_R = 0.005$ ).**

### V.3.3.4. Wheel slip

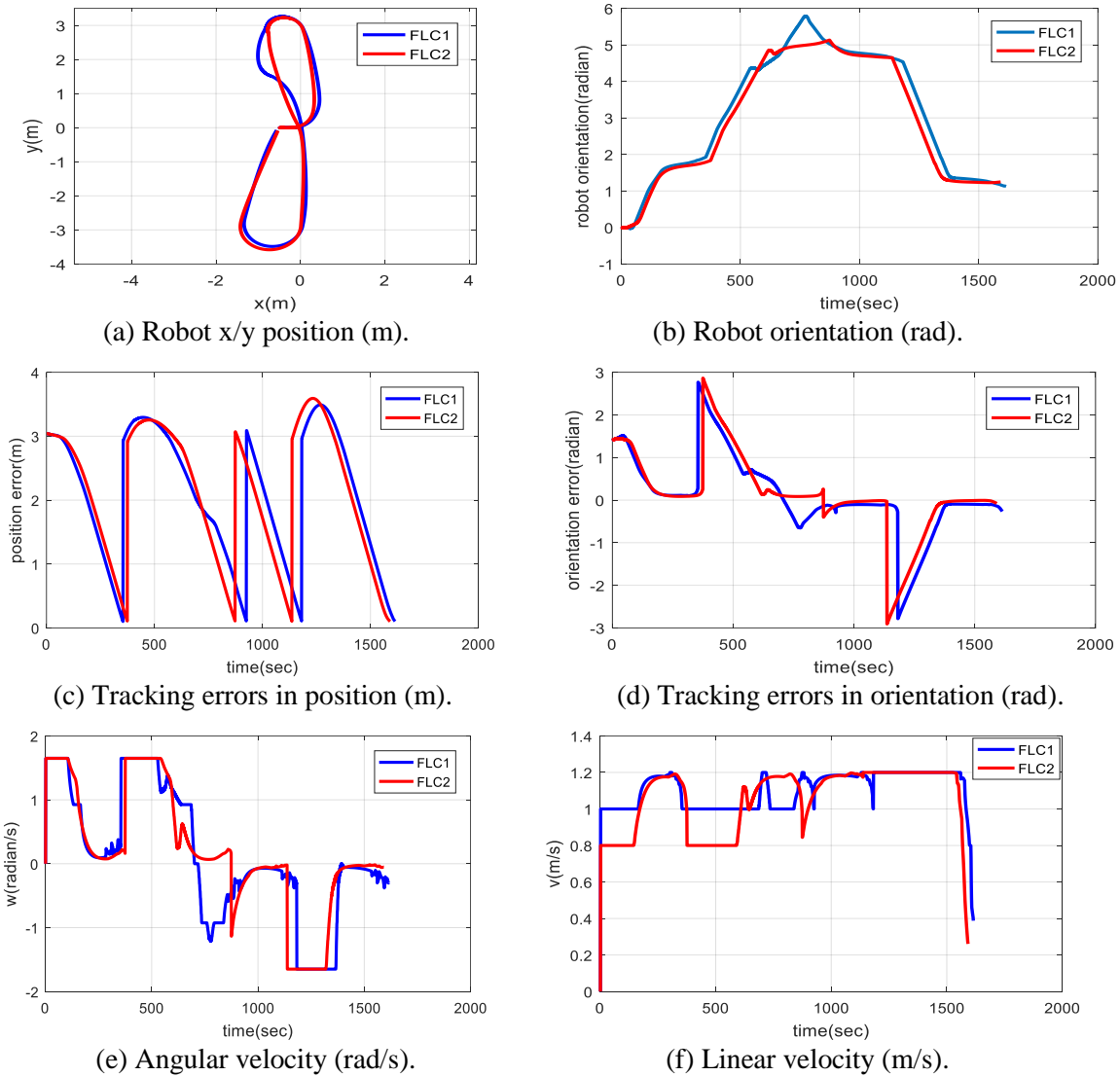
In this section, we present simulation results to evaluate the performances of the two controllers on the WMR model that include wheel slip dynamics. The proposed controller is validated based on waypoints navigation problem that is subject to longitudinal slips (for both wheels). Two scenarios are considered with different parameters, in the first case  $\alpha_{right} = 1$  (right wheel),  $\alpha_{left} = 3$  (left wheel), the obtained results are illustrated in figure V-18. As can

be seen from this figure, the Interval Type 2 FLC maintains good performances even with the presence of longitudinal slips (for both wheels), in the other hand, the performances of the Type 1 FLC are reduced (poor precision). In the second case, we choose other waypoints, significant longitudinal slips is considered  $\alpha_{right} = 20, \alpha_{left} = 5$ , the obtained results are illustrated in figure V-19. In this case the Interval Type 2 FLC maintains good performances even with the presence of longitudinal slips (for both wheels), when the performances of the Type 1 FLC are reduced (poor precision).



**Figure V-18: Results obtained with DDWMR navigation with slip dynamics ( $\alpha_L = 3, \alpha_R = 1$ ).**





**Figure V-19: Results obtained with DDWMR navigation with slip dynamics ( $\alpha_L = 5, \alpha_R = 20$ ).**

## V.4. V-REP MODEL AND MATLAB/SIMULINK INTERFACING

### V.4.1. Virtual robot experimentation platform (V-REP)

In our study, we have used the Virtual Robot Experimentation Platform (V-REP), as physical simulator, which provides an easy and intuitive environment to create a virtual platform and to include some popular robots, objects, structures, actuators and sensors.

Virtual Robot Experimental Platform (V-REP) is the product of Coppelia Robotics [106] developed for general purpose robot simulation. The main characteristics of this simulator are: a customized user interface and a modular structure integrated development environment. Modularity is in high

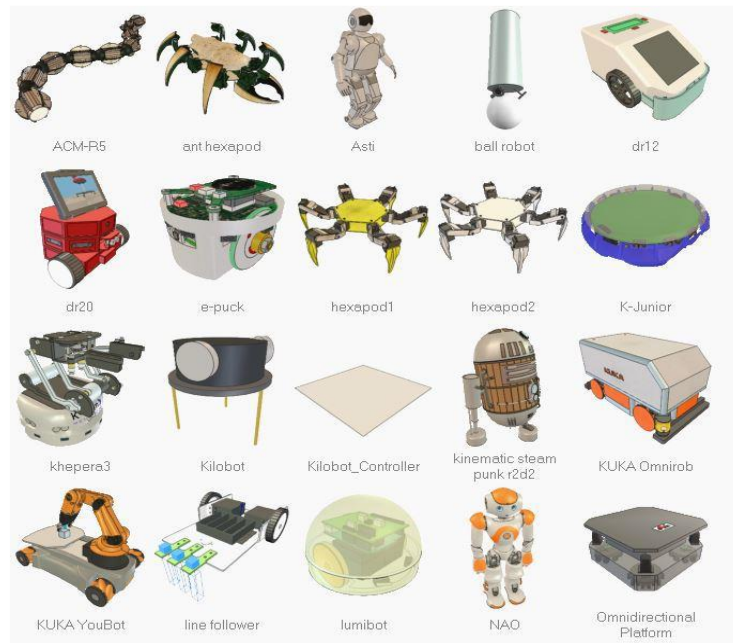
level both for simulation objects and control methods. The easy use of the development environment inside the simulator allows the creation of different robots and simulation scenarios. This capability permits the fast prototyping, algorithm design and implementation.

V-REP simulation scene component or model is composed by the following objects:

- **Shapes:** Shapes are triangular faced rigid mesh objects. These objects can be used in collision detections against other collidable objects and minimum distance calculations with other measurable objects. Shapes also can be detected by proximity and vision sensors.
- **Joints:** Joints are the tools used for building mechanisms and moving object, which has at least one Degree of Freedom (DOF). There are four joints types, which are: revolute, prismatic, spherical and screw joints. The operation modes of joints are passive mode, inverse kinematic mode, dependent mode, motion mode and finally torque or force mode. Dynamic model of actuator can be modeled by enabling torque mode or force mode.
- **Proximity sensors:** From ultrasonic to infrared, nearly all type of proximity sensors can be modeled to simulate proximity sensors. They estimate an exact distance from sensor to any detectable entity that interferes with its detection volume.
- **Vision sensors:** Provide all renderable objects in simulation scene (colors, objects sizes, depth maps, etc.) and extract complex image information. A built-in filter and image processing functions simplify the use of vision sensors in simulation.
- **Force sensors:** These sensors measure transmitted force and torque values between two or more objects. The force sensor principle can be modeled as real one, so that, they can even be broken under overshoot force or torque values.
- **Graphs:** Graphs objects are used to record, visualize and export data from simulation. The graphs feature in V-REP are very powerful, so that time graphs, x/y graphs and 3D graphs can be generated for data types applied to specific objects to be recorded.

- **Cameras:** Cameras are objects that can be used to monitor a simulation from different viewpoints. It can either add multiple view windows in one view window or attach each view to separate windows.
- **Lights:** Lights are the objects that light the simulation scene and directly influence camera and vision sensors.
- **Paths:** Paths are objects that define a rotational, translational or combined path or trajectory in space.
- **Dummies:** A dummy is a type of object that can be defined as a reference frame or point of orientation attached to the object. They are useful especially for path-trajectory planning and following. Dummies are generally multipurpose helper object in combination with other objects. It must be noticed that alone they are not so useful.
- **Mills:** Using mills, almost, any type of cutting volumes as long as they are convex can be modeled. Mills always have a convex cutting volume; however, they can be combined to generate a non-convex cutting volume or more complex volumes.

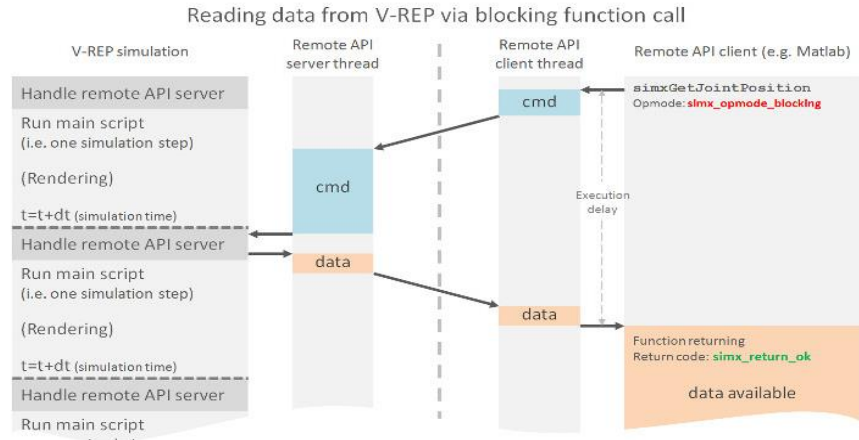
The combination of above described scene objects allow the creation of complex sensors (accelerometer, gyroscope, GPS, Kinect, etc.), and complex models from manipulators to wheeled robots (figure V-20). There is a wide sensor and robot model library in VREP environment that can be added easily to the scene. In addition, it must be noticed that these models are fully customizable.



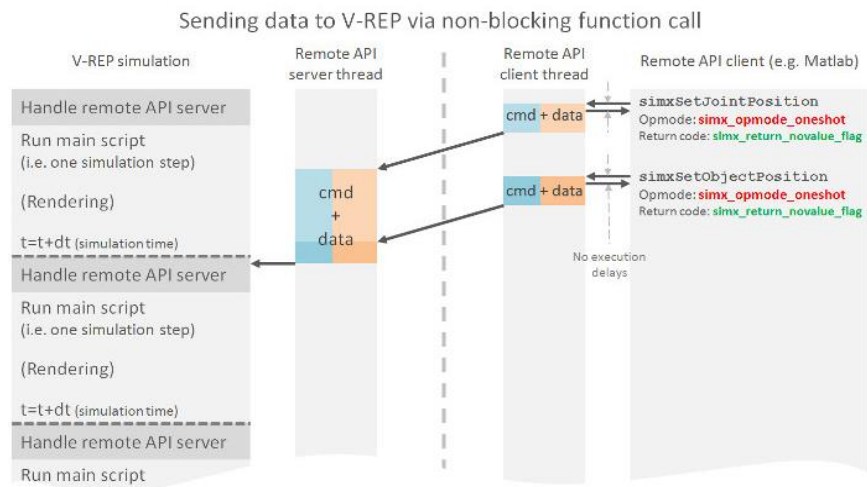
**Figure V-20: The built-in V-REP robot models**

There are various control mechanisms to manage the behavior of each simulation objects. These controllers can be implemented not only inside of the simulation environment but also outside of the simulation environment. The main internal control mechanism is the use of child scripts, which can be associated with any element in the scene.

For simulator in the loop configuration tests, V-REP offers also a method to control the simulation from outside the simulator by external implied controller algorithm. The controller, developed in remote API interface in V-REP, communicates with the simulation scene using a communication socket. It is composed by a remote API server services and a remote API clients. The client side can be developed in C/C++, Python, Java, Matlab or Urbi languages, also it can be embedded in any software running on remote control hardware or real robots, and it allows remote function calling, as well as fast data streaming. Functions support two calling methods to be adapted to any configuration: blocking, waiting until the server replies, or non-blocking, reading streamed commands from a buffer.... The schematic of this communication mode is shown in figure V.21. Plugins implement the API server inside V-REP for providing a simulation process with standard LUA commands. So they, generally, are used in combination with scripts. On the other hand, if there is need for either fast calculation case (compiled languages most of the time are faster than scripts) or an interface to a real device (e.g. real robot), the plugging provides special functionality.



(a)



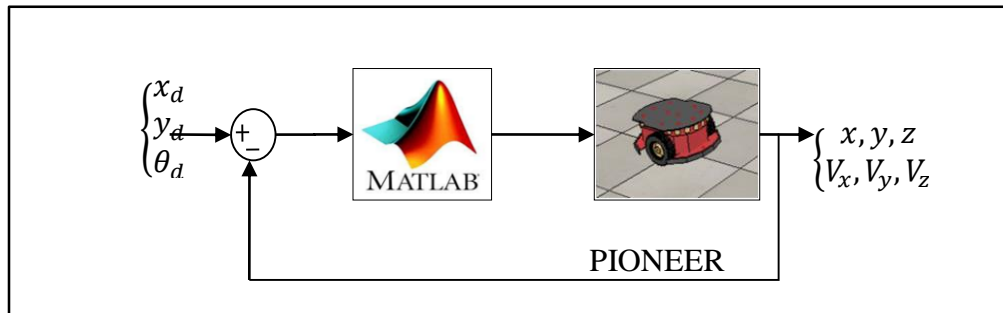
(b)

**Figure V-21: The remote API communication modes: (a) blocking function call, (b) non-blocking function call [22].**

The interaction between objects in the simulation scene is calculated by various calculation modes. V-REP's dynamics module currently supports four different physics engines: the Bullet physics library [23], the Open Dynamics Engine [24], the Vortex Dynamics engine [25] and the Newton Dynamics engine [26]. At any time, it is easy to switch from one engine to the other quickly according to the simulation needs. The reason for this diversity in physics engine support is that physics simulation is a complex task, which can be achieved with various degrees of precision, speed, or with support of diverse features.

#### V.4.2. Matlab/Simulink and V-REP Interfacing

V-REP offers a remote API allowing to control a simulation from an external application. The V-REP remote API is composed by approximately one hundred functions that can be called from Matlab program. In our application the command and controller part are developed in Matlab when actuation and physical interaction part are created in V-REP. The general scheme of developed simulator can be seen in figure V.22.



**Figure V-22: General simulator operation diagram**

The control data is sent to Matlab where the entire communication process is completed. In Matlab remote, API functions and Matlab functions are integrated. Remote APIs interact with V-REP over socket communication to dramatically reduce network delays and load. Synchronization with V-REP is also performed in this part of the simulation. In Matlab, the fuzzy controller gets commands, makes corrections, and sends that data to VREP to control left and right velocities of the PIONEER 3DX robot's wheel motors. After that, PIONEER 3DX status data will be sent to the correction block in Matlab. In the main MATLAB, V-REP script gets the engine speeds from sends these values to the appropriate child model LUA scripts. Meanwhile, the main LUA script reads the odometer data from child scripts, obtains the absolute positions  $(x, y, z)$  and orientations  $(\varphi, \theta, \psi)$  with the Euler angles and linear velocities and sends them to the Matlab application. Visualization of processed data is done in both Matlab and V-REP applications.

#### V.5. V-REP Validation

The aforementioned controllers FLC1 and FLC2 are validated and compared using V-REP simulator. This latter provides realistic simulation environment, moreover it gives more options to evaluate the performances of the both controllers under real scenarios (parameters uncertainties, localization error and loss of efficiency ...).

### V.5.1. Scenario 1: Go to goal (single waypoint)

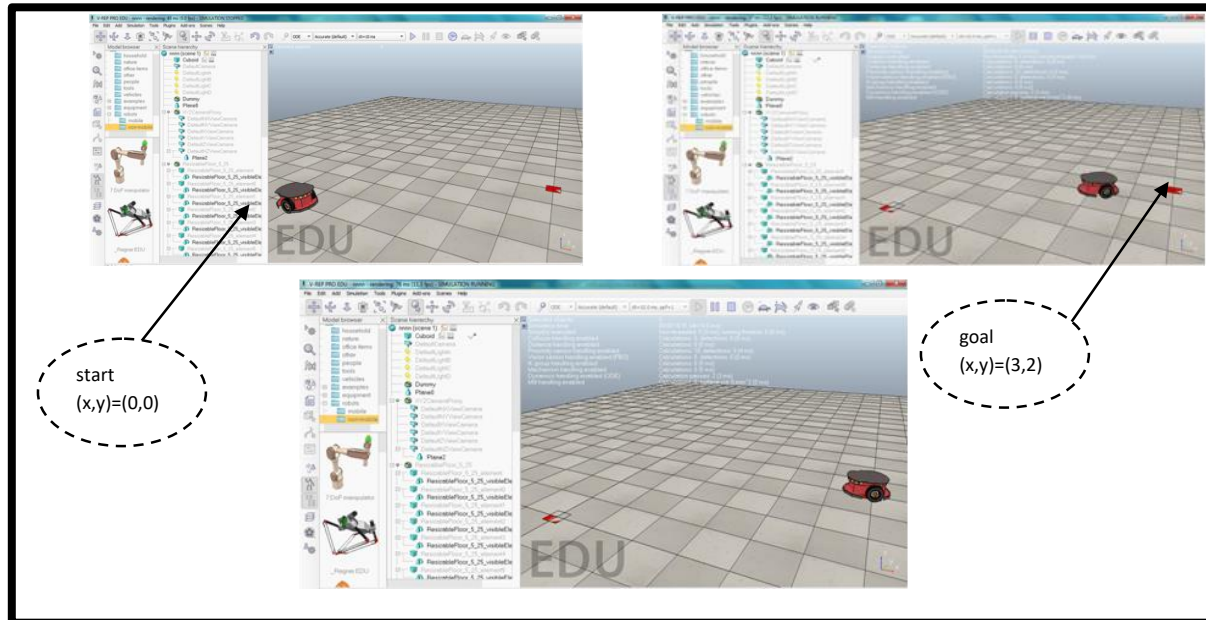
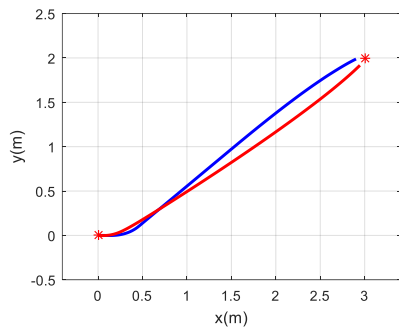
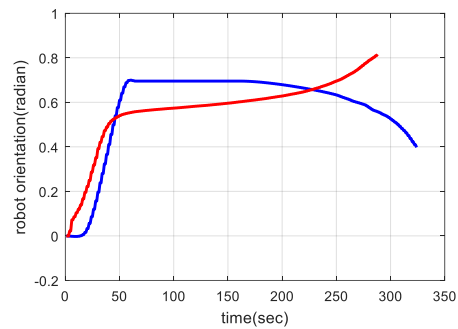


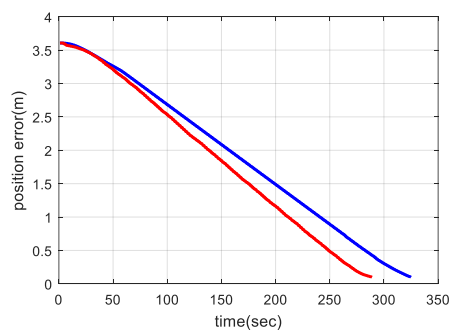
Figure V-23: PIONEER 3DX in V-REP environment with single waypoint.



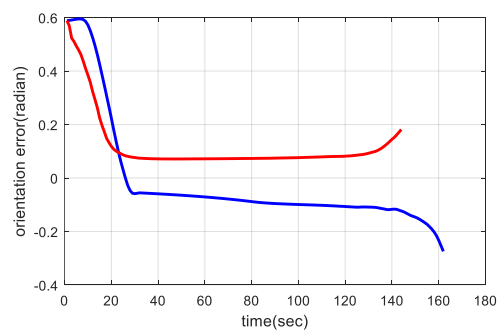
(a) Robot x/y position (m).



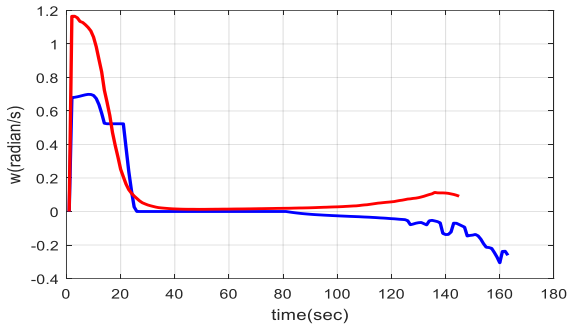
(b) Robot orientation (radian).



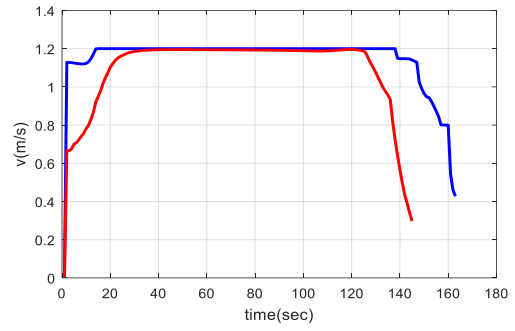
(c) Tracking errors in position (m).



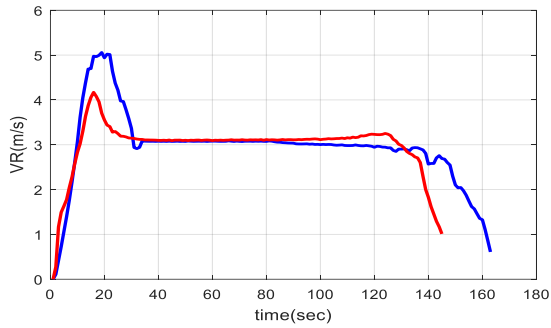
(d) Tracking errors in orientation (radian).



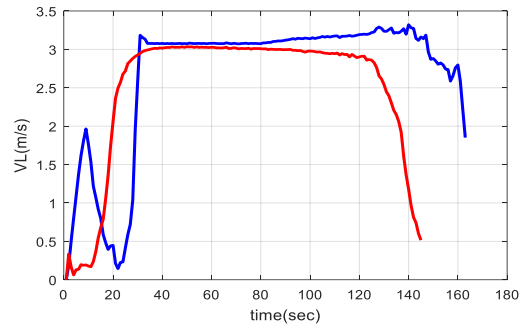
(e) Angular velocity (radian/s).



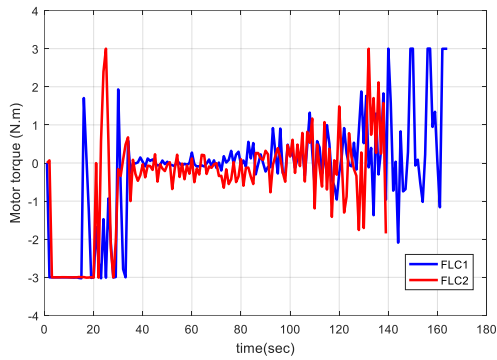
(f) Linear velocity (m/s).



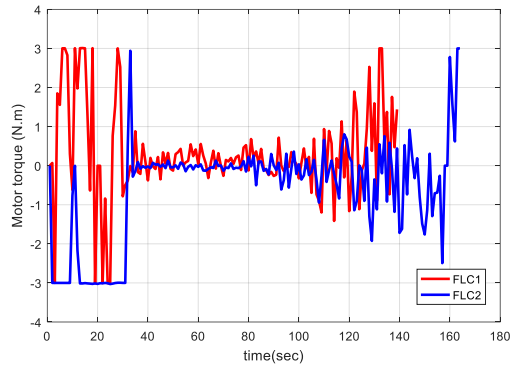
(g) Linear velocity of the right wheel (m/s).



(h) Linear velocity of the left wheel (m/s).



(i) Right motor torque (N.m).



(j) Left motor torque (N.m).

**Figure V-24: V-REP: Results obtained for the scenario: Go to goal.**

In the first scenario, the go to goal behavior is considered using PIONEER 3DX robot from VREP simulator (figure V-23). From figure V-24, both controllers provides similar results with less navigation time by the FLC2 controller.



## V.5.2. Scenario 2 : Multiple waypoints navigation

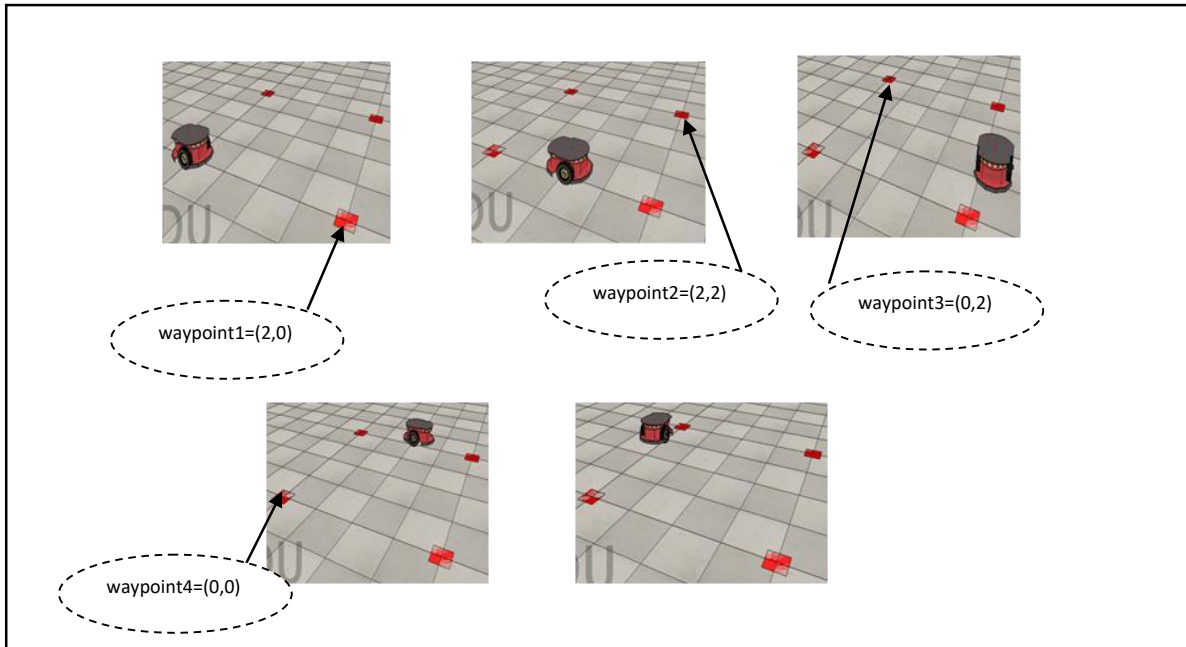
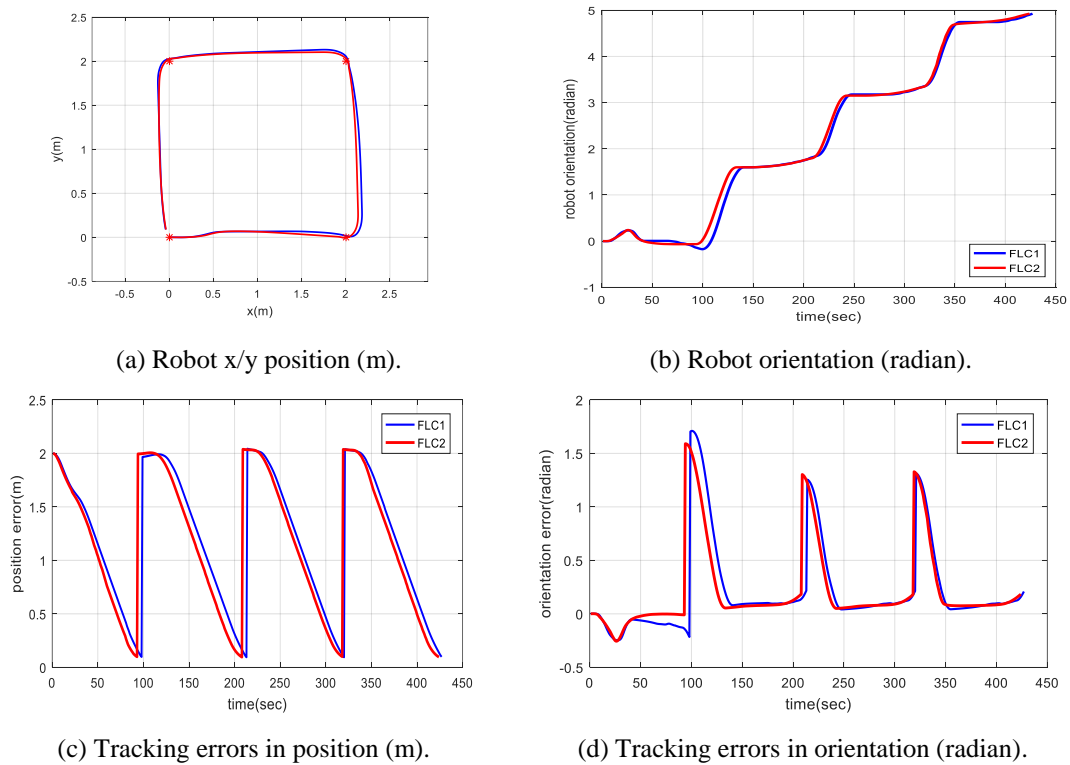
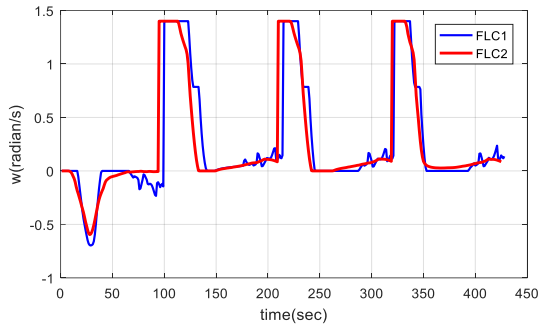
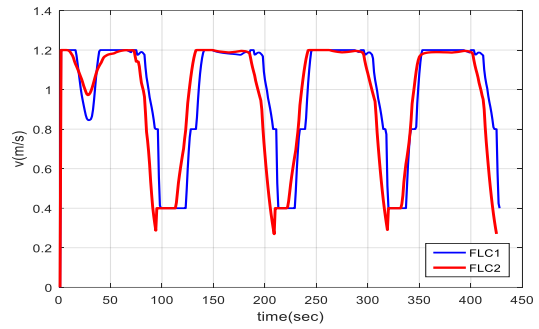


Figure V-25: PIONEER 3DX in V-REP environment with multiple waypoints navigation

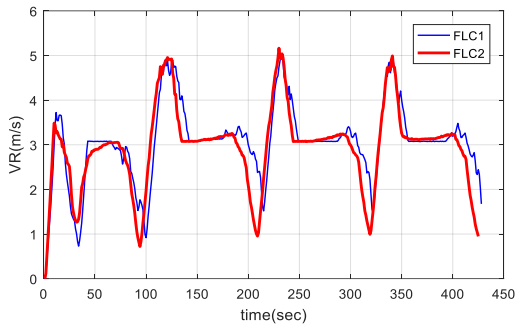




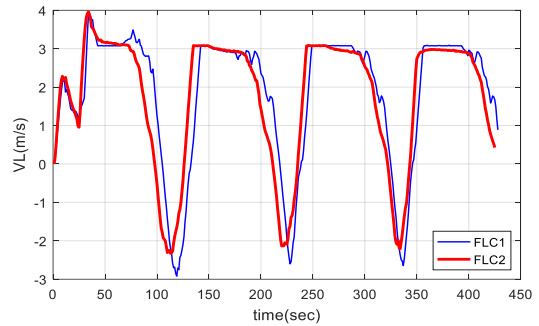
(e) Angular velocity (radian/s)



(f) Linear velocity (m/s).



(g) Linear velocity of the right wheel (m/s)



(h) Linear velocity of the left wheel (m/s).

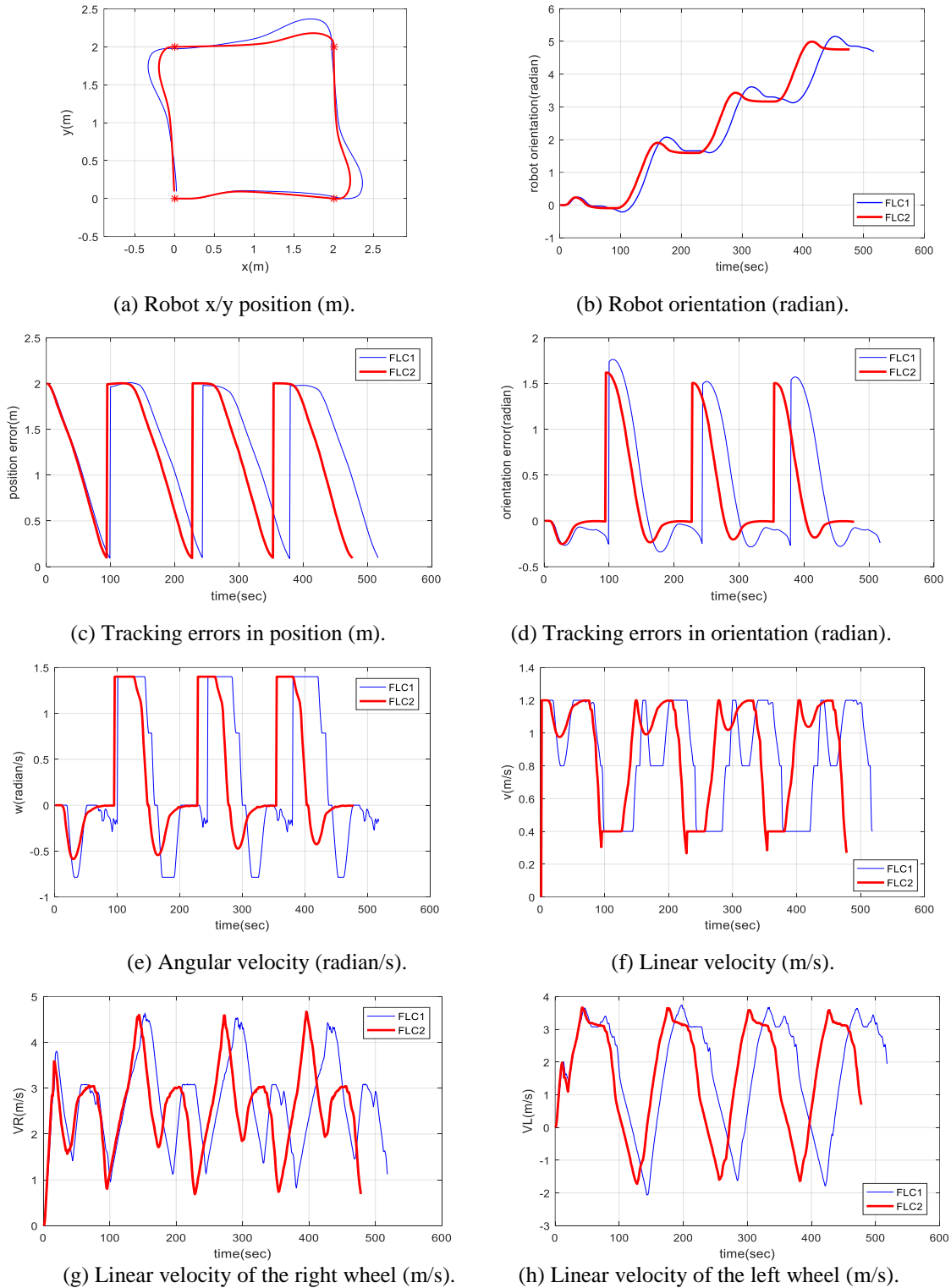
**Figure V-26: V-REP: Results obtained for the scenario: Multiple waypoints navigation**

In this simulation, the mobile robot should navigate to multiple waypoints. Four waypoints are considered to compare the behavior of the two controllers. From figure V-26, the two controllers navigate to different waypoints successfully and similarly even the type 2 controller contains additional step of type reduction.

### V.5.3. Scenario 3 : Robustness analysis

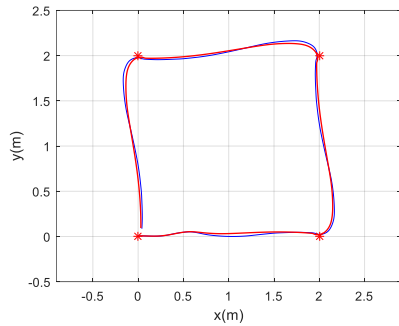
#### V.5.3.1. Modeling errors

In this simulation, controller robustness is evaluated face parameters uncertainties. In this case using PIONEER 3DX robot from VREP simulator, in the first scenario we consider an uncertainty on the length  $L$  of the mobile robot:  $\Delta L = 30\%$  was added in order to analyze the robustness of the controllers face modeling errors. Illustrated results in figure V-27 confirm the robustness of interval type 2 controller against type 1 controller.

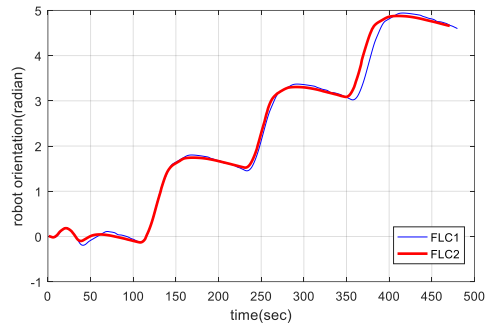


**Figure V-27: V-REP: Results obtained for the scenario: Multiple waypoints navigation ( $\Delta L = 30\%$ ).**

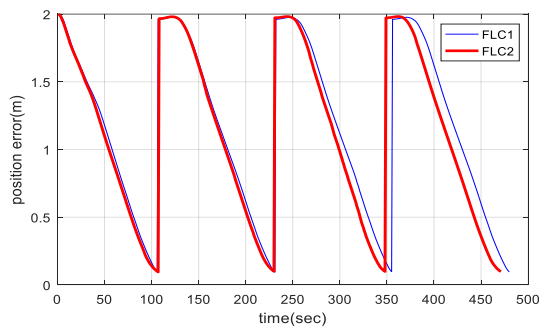
In this simulation, controller robustness is evaluated face parameters uncertainties; we consider an uncertainty on the left radius wheel with  $\Delta RL = 15\%$  (Figure V-28) then  $\Delta RL = 25\%$  (Figure V-29). As Illustrated in these figures, both controllers present similar performances face radius uncertainty with slight navigation time improvement by the FLC2.



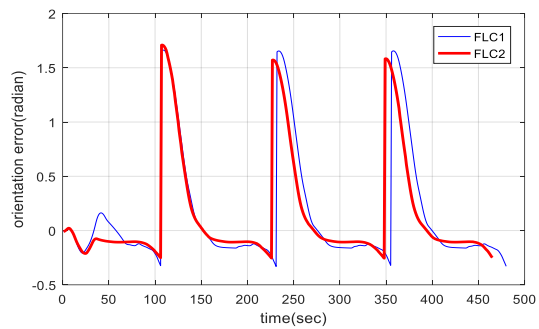
(a) Robot x/y position (m).



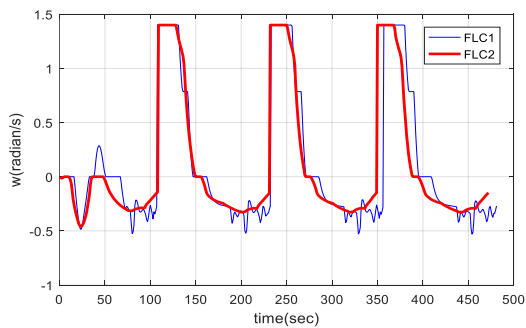
(b) Robot orientation (radian).



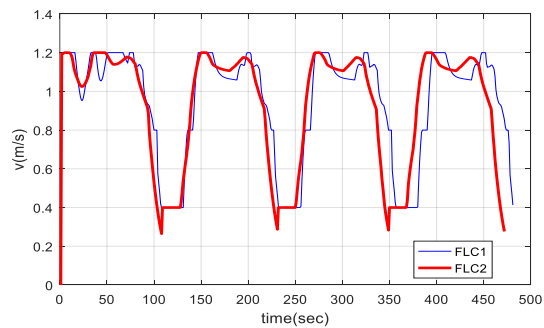
(c) Tracking errors in position (m).



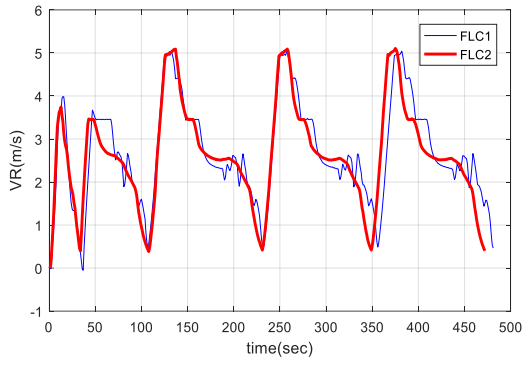
(d) Tracking errors in orientation (radian).



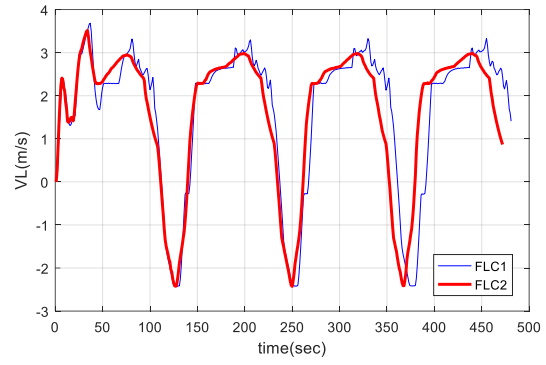
(e) Angular velocity (radian/s)



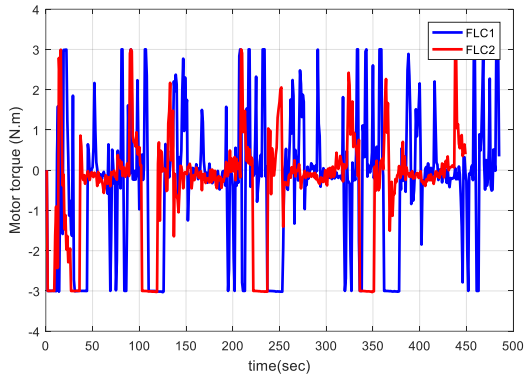
(f) Linear velocity (m/s).



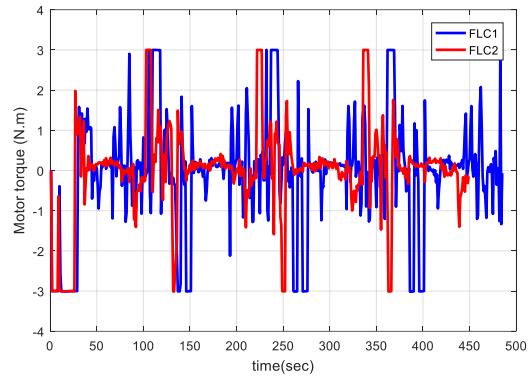
(g) Linear velocity of the right wheel (m/s).



(h) Linear velocity of the left wheel (m/s).

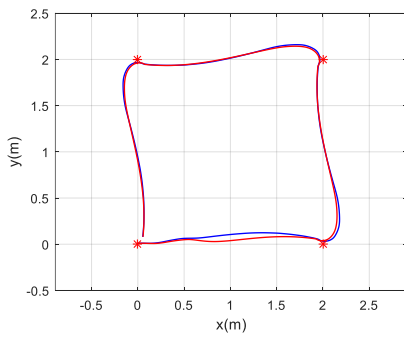


(i) Right motor torque (N.m).

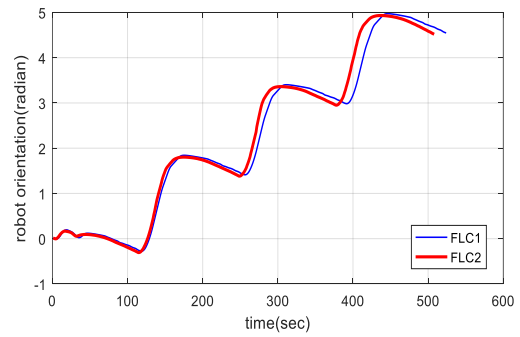


(j) Left motor torque (N.m).

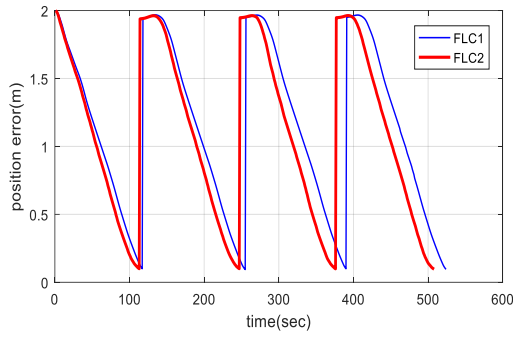
**Figure V-28: V-REP: Results obtained for the scenario: Multiple waypoints navigation ( $\Delta R = 15\%$ ).**



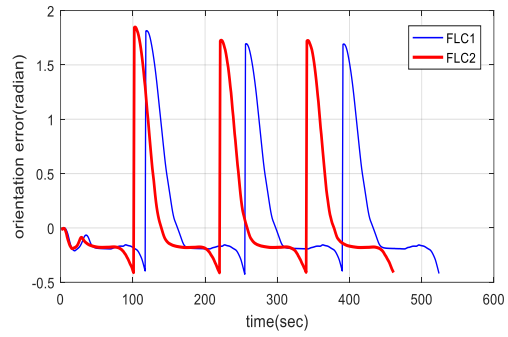
(a) Robot x/y position (m).



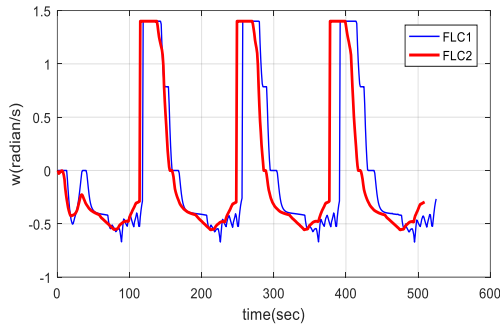
(b) Robot orientation (radian).



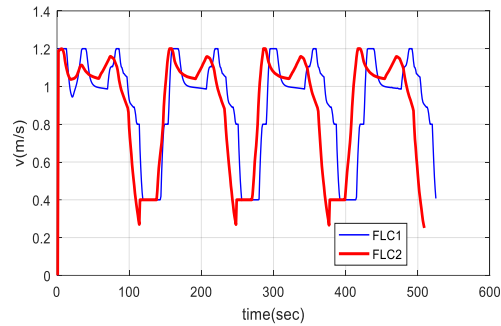
(c) Tracking errors in position (m).



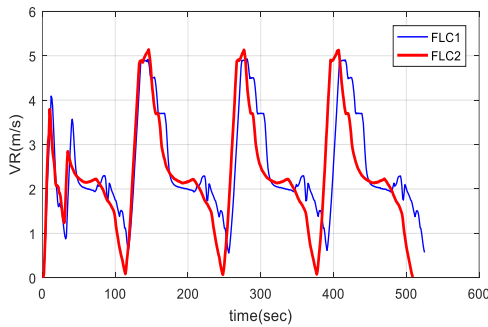
(d) Tracking errors in orientation (radian).



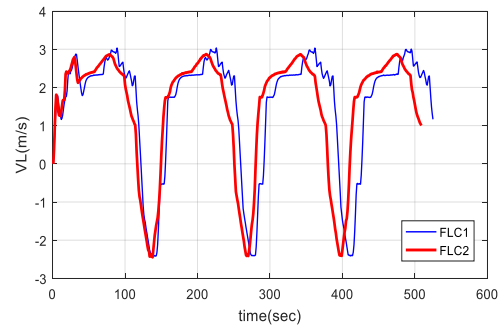
(e) Angular velocity (radian/s)



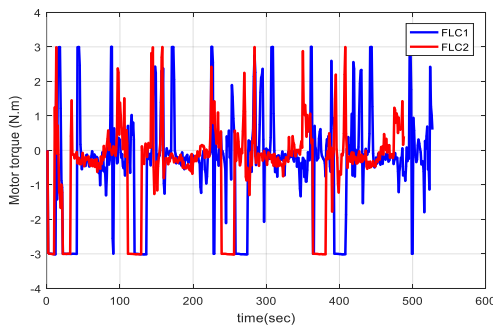
(f) Linear velocity (m/s).



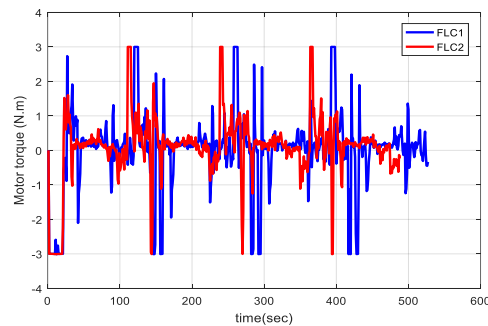
(g) Linear velocity of the right wheel (m/s).



(h) Linear velocity of the left wheel (m/s).



(i) Right motor torque (N.m).



(j) Left motor torque (N.m).

**Figure V-29: V-REP: Results obtained for the scenario: Multiple waypoints navigation ( $\Delta R = 25\%$ ).**

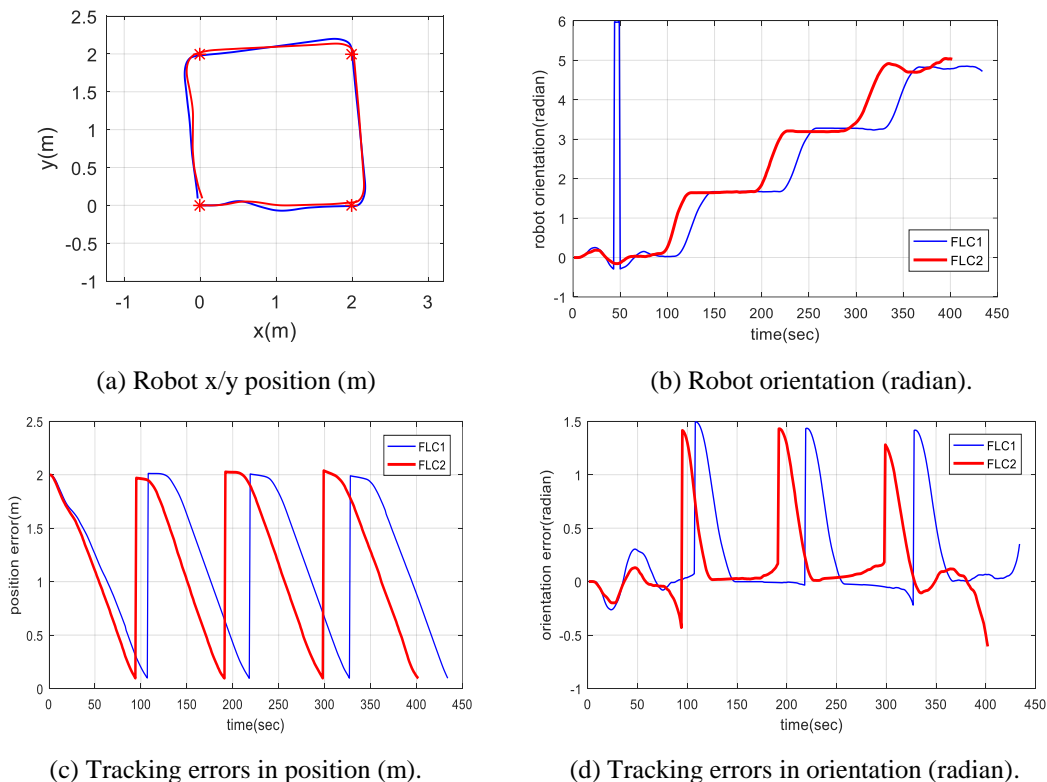
### V.5.3.2. Localization error

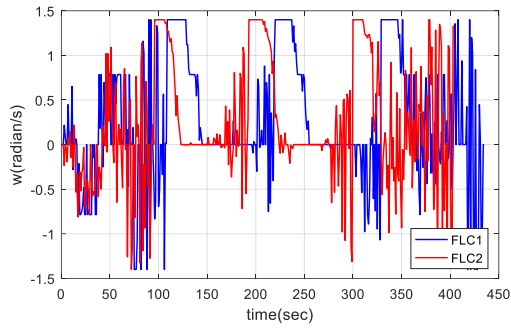
In this section, the robustness of controllers is evaluated face localization error using V-REP simulator. In order to better observe the effect of localization error on the performances of waypoints navigation, we consider noisy odometry position.

We consider :  $Position(x,y,\theta) = Position(x,y,\theta) + \sigma_{noise} \cdot randn$

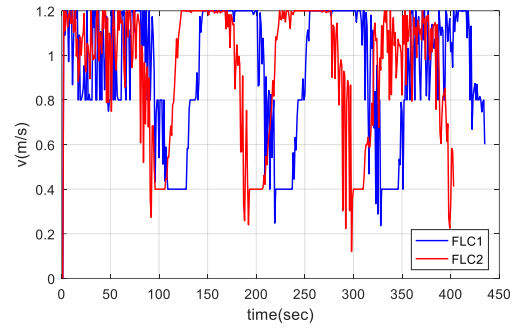
The overall scheme for the localization and position control of a differential drive mobile robot is shown in figure V-8.

Figure V-30 and figure V-31 illustrate the obtained results for waypoint navigation using both controllers using noisy position with  $\sigma_{noise} = 0.1$  and  $\sigma_{noise} = 0.3$  respectively. From these figures, we observe that both controllers provide good similar results with small position error. However, when this error increases (Figure V-31) the FLC1 performances decrease significantly (poor precision with considerable navigation time). In the other hand, the FLC2 based on interval Type 2 Fuzzy Logic Controller maintains good performances of navigation with suitable navigation time.

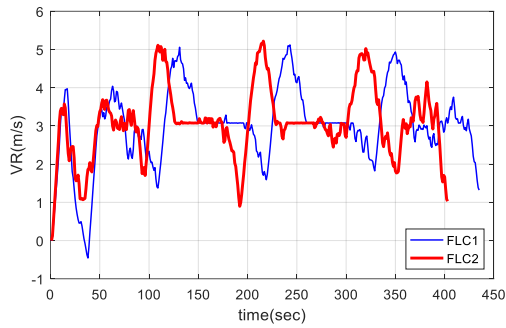




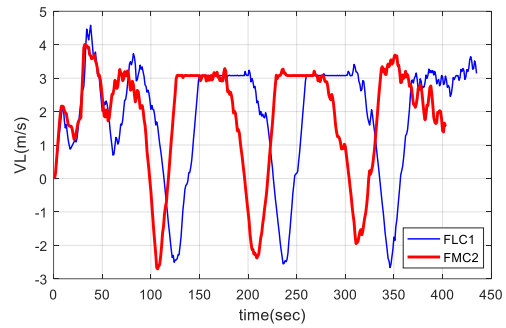
(e) Angular velocity (radian/s)



(f) Linear velocity (m/s).

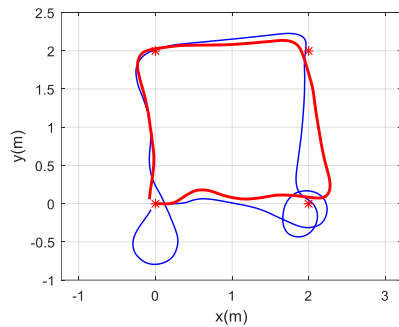


(g) Linear velocity of the right wheel (m/s).

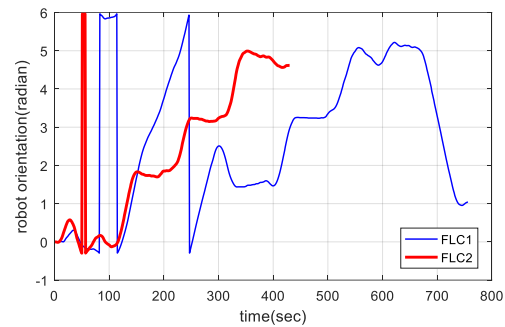


(h) Linear velocity of the left wheel (m/s).

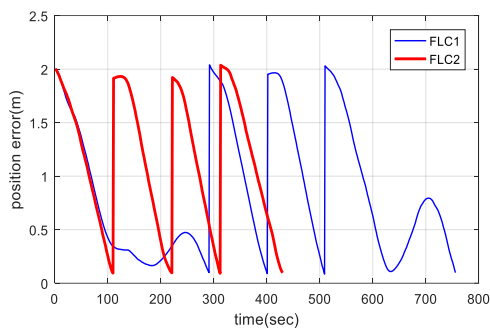
**Figure V-30: V-REP: Results obtained for the scenario: Multiple waypoints navigation (uncertainties of localization  $\alpha = 0.1$ ).**



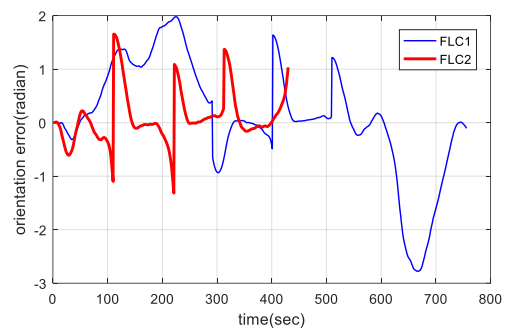
(a) Robot x/y position (m)



(b) Robot orientation (radian).

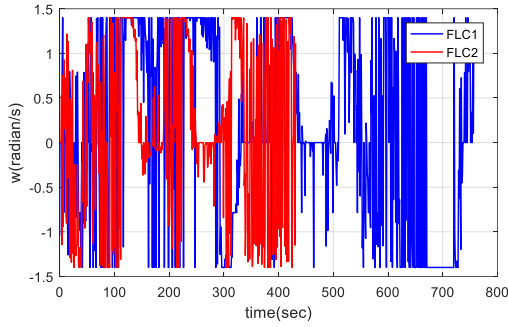


(c) Tracking errors in position (m).

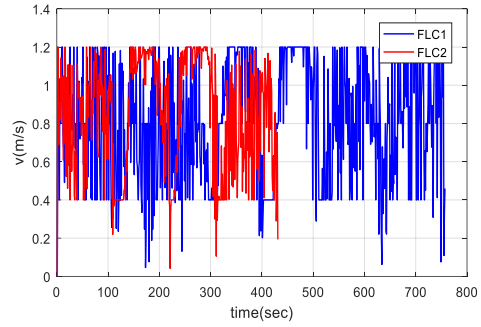


(d) Tracking errors in orientation (radian).

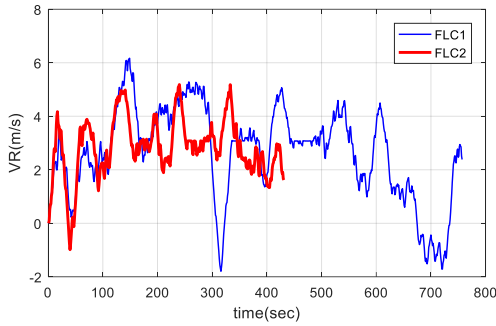




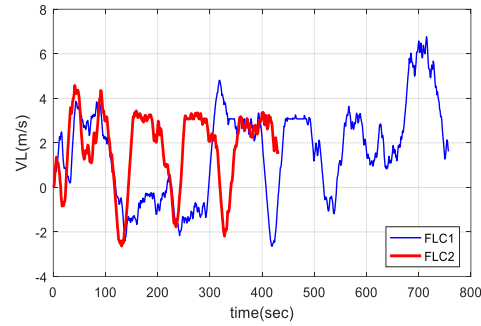
(e) Angular velocity (radian/s).



(f) Linear velocity (m/s).



(g) Linear velocity of the right wheel (m/s)



(h) Linear velocity of the left wheel (m/s).

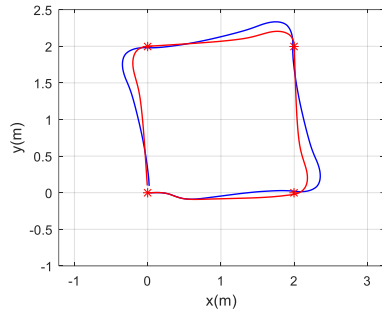
**Figure V-31: V-REP: Results obtained for the scenario: Multiple waypoints navigation (uncertainties of localization  $\alpha = 0.3$ ).**

### V.5.3.3. Loss of efficiency

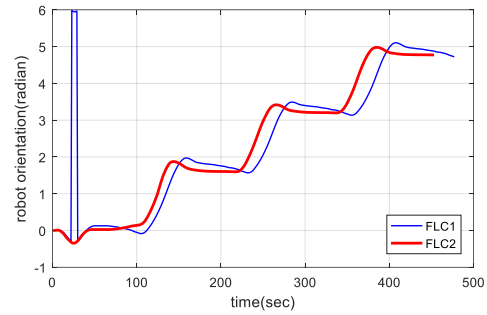
In this section the robustness of controllers is evaluated face loss of efficiency of one or two motors using V-REP simulator, in this experiment we consider medium ( $\alpha_L = 0.5, \alpha_R = 1$ ) and high ( $\alpha_L = 0.1, \alpha_R = 0.75$ ) loss of efficiency.

Figure V-32 represents the obtained results with medium loss of efficiency, from this figure, we observe that both controllers maintain a good robustness even the left motor loss 50% of its efficiency. The FLC2 presents quiet good results comparing to the FLC1.

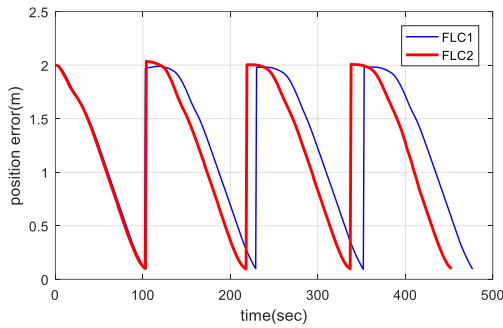
Figure V-33 illustrates the obtained results face high loss of efficiency, in this case the right motors loss 25% of its efficiency, when the left motor loss 90%. Poor performances are obtained by both controllers, the mobile robot reaches waypoints following long path with considerable navigation time.



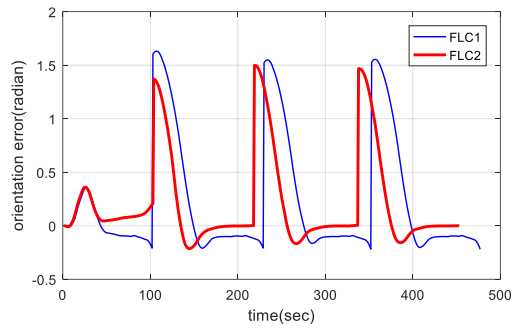
(a) Robot x/y position (m).



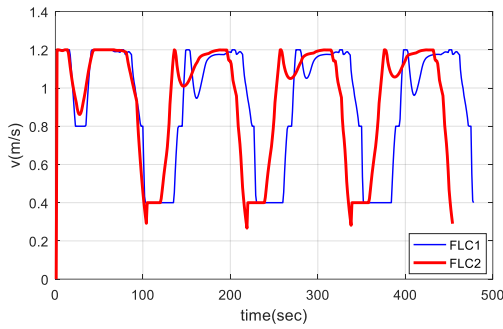
(b) Robot orientation (radian).



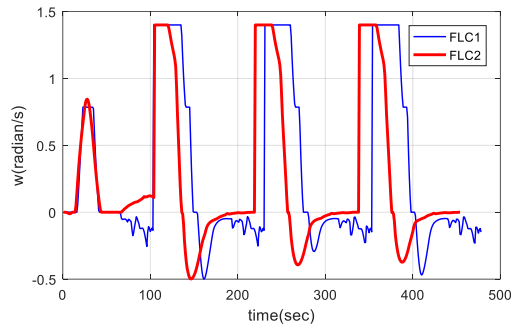
(c) Tracking errors in position (m).



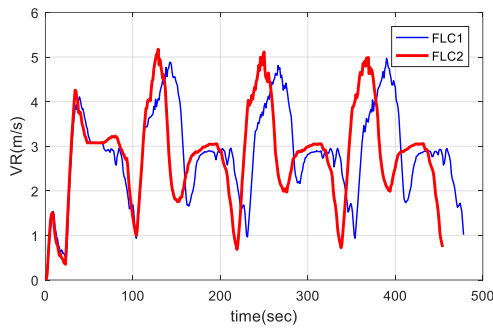
(d) Tracking errors in orientation (radian).



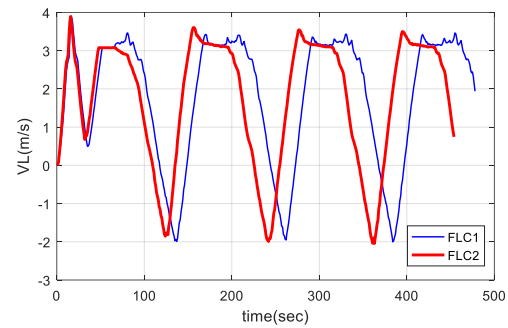
(e) Angular velocity (radian/s)



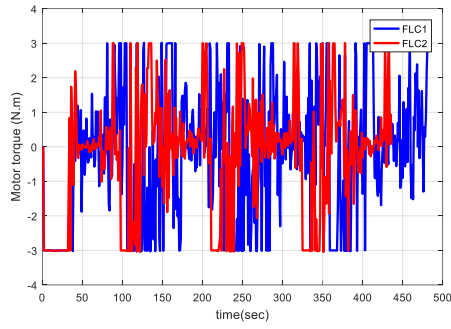
(f) Linear velocity (m/s).



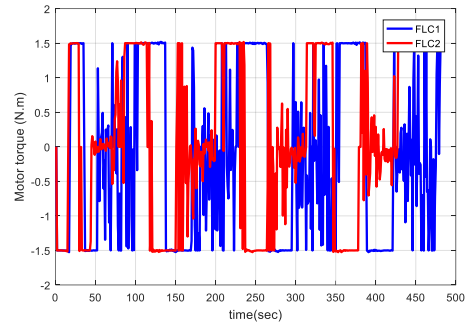
(g) Linear velocity of the right wheel (m/s)



(h) Linear velocity of the left wheel (m/s).

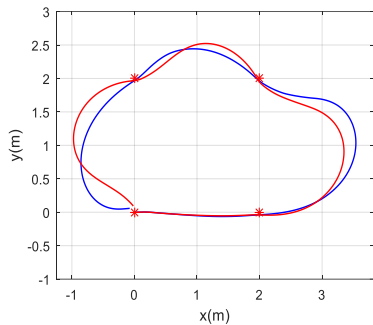


(i) Right motor torque (N.m).

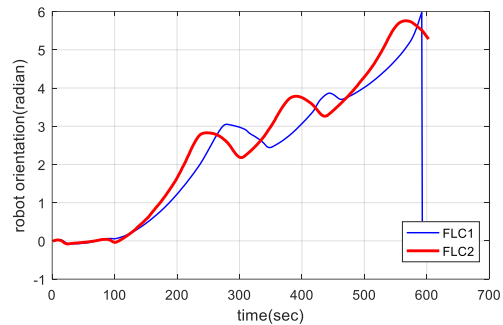


(j) Left motor torque (N.m).

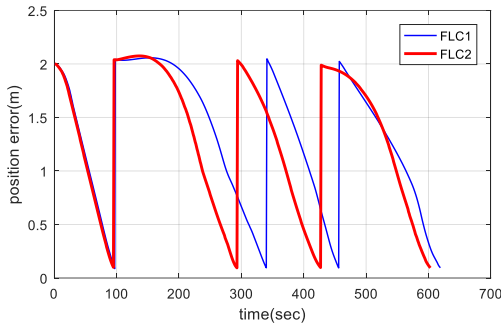
**Figure V-32: V-REP: Results obtained for the scenario: Multiple waypoints navigation (loss of efficiency of motors  $\alpha R = 1, \alpha L = 0.5$ ).**



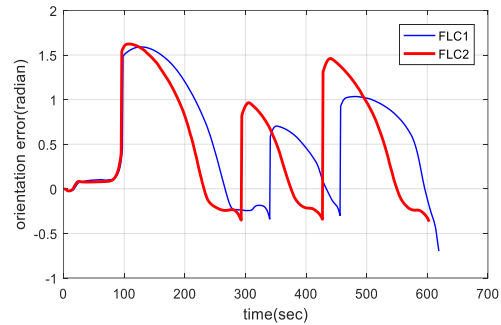
(a) Robot x/y position (m).



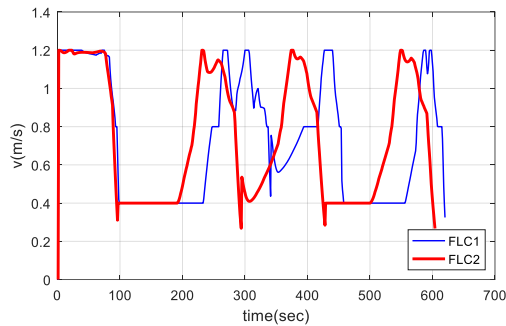
(b) Robot orientation (radian).



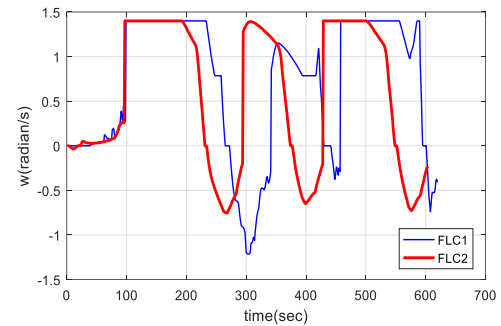
(c) Tracking errors in position (m).



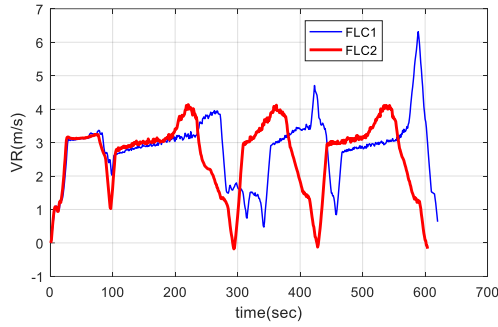
(d) Tracking errors in orientation (radian).



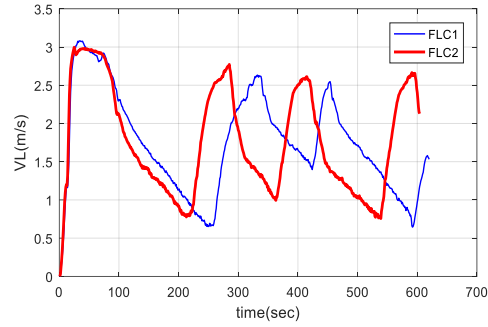
(e) Angular velocity (radian/s)



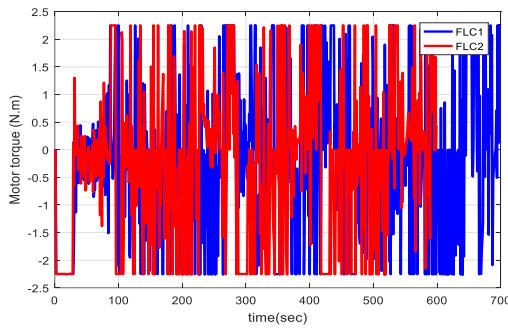
(f) Linear velocity (m/s).



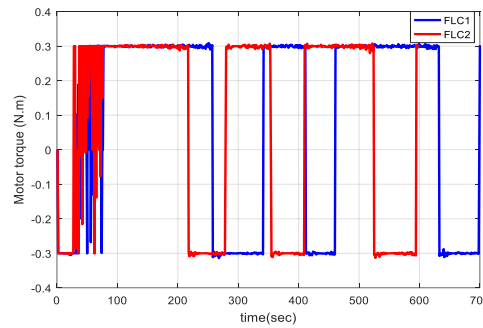
(g) Linear velocity of the right wheel (m/s).



(h) Linear velocity of the left wheel (m/s).



(i) Right motor torque (N.m).



(j) Left motor torque (N.m).

**Figure V-33: V-REP: Results obtained for the scenario: Multiple waypoints navigation (loss of efficiency of motors  $\alpha_R = 0.75, \alpha_L = 0.1$ ).**

Table V-5 summarizes simulation results:

**Table V-5: Summary on the results of fuzzy controllers**

	<b>Scenario 1: Single Waypoint</b>	<b>Scenario 2: Multiple Waypoints</b>	<b>Scenario 3: Robustness Analysis</b>
<b>Kinematic model</b>	The two controllers navigate go to goal successfully and similarly	The two controllers navigate to different waypoints successfully and similarly	<p><math>\Delta L = 30\%</math>: The two controllers navigate to different waypoints successfully and similarly, this uncertainty affect the two fuzzy controllers, which drifts from its initial trajectory especially when new waypoint is considered.</p> <p><math>\Delta R_L = 25\%</math>: Same result as the previous one.</p> <p><b>Localization error:</b> the two controllers navigate to different waypoints following noisy trajectories especially with the FLC1. The FLC2 require less navigation time comparing to the FLC1.</p>

<p><b>Dynamic model</b></p>	<p>The two controllers FLC1 and FLC2 give similar results</p>	<p>The two controllers navigate to different waypoints successfully and similarly. The FLC1 require less navigation time comparing to the FLC2.</p>	<p><math>\Delta L=30\%</math>: both controllers presents suitable robustness face robot length uncertainty</p> <p><math>\Delta R_L = 25\%</math>: The two controllers navigate to different waypoints successfully and similarly, the two fuzzy controllers drifts from its initial trajectory especially when new waypoint is considered.</p> <p><math>\Delta R_L = 50\%</math>: The performances of the FLC1 decreases significantly, good performances are obtained by the FLC2. The FLC2 comparing to the FLC1 requires the mobile robot using FLC2 requires less navigation time to achieve different waypoints, less energy.</p> <p><math>\Delta R_L = 60\%</math>: The performances of the FLC1 decreases significantly (long trajectory, increase of navigation time, more energy required), the FLC2 maintains suitable performances for good navigation. These results confirm the robustness of the interval type 2 FLC face significant parameters uncertainties.</p> <p><b>Localization error:</b> both controllers provide good results.</p> <p><b>Loss of efficiency (<math>\alpha_L = 0.6, \alpha_R = 0.1</math>):</b> both controllers provide similar and good results.</p> <p><b>Loss of efficiency (<math>\alpha_L = 0.05, \alpha_R = 0.005</math>):</b> FLC2 performs much better than FLC1, FLC2 needs less time to achieve the different waypoints comparing to the FLC1.</p> <p>Wheel slip: FLC2 maintains good performances even with the presence of longitudinal slips (for both wheels), the performances of FLC1 are reduced.</p>
-----------------------------	---	---	--

<b>VREP validation</b>	Both controllers provide similar results with less navigation time by the FLC2.	The two controllers navigate to different waypoints successfully and similarly	<p><math>\Delta L=30\%</math>: both controllers present suitable robustness face robot length uncertainty.</p> <p><math>\Delta R_L = 15\%</math> and <math>\Delta R_L = 25\%</math>: Both controllers present similar performances face radius uncertainty. With slight navigation time improvement by the FLC2.</p> <p>Localization error: both controllers provide good similar results with small position error. When this error increases, the FLC1 performances decrease significantly (poor precision and considerable navigation time). The FLC2 maintains good performances of navigation in suitable navigation time.</p> <p>Loss of efficiency of motors: both controllers maintain a good robustness. The FLC2 presents quiet good results comparing to the FLC1.</p>
------------------------	---	--	---

A summary of performances analysis for both controllers is illustrated in table V-6 bellows. These parameters were obtained by the following calculation:

$$Error\ of\ distance\ (\%) = (sum\ (distance)/time)/100 \quad (V.1)$$

$$With: distance = \sqrt{(x_{robot} - x_{waypoint_i})^2 + (y_{robot} - y_{waypoint_i})^2}$$

*i*: Number of waypoints

$$Error\ of\ orientation\ (\%) = (sum\ (orientation)/time)/100 \quad (V.2)$$

$$Effective\ torque\ of\ the\ motor = \sqrt{(sum(motor\ torque)^2/time)} \quad (V.3)$$

$$Torque\ max\ of\ the\ motor = \max(motor\ torque) \quad (V.4)$$

Fives scenarios of navigation using both controllers are implemented and compared. As can be concluded from this table, the Type 2 FLC (FLC2) performs much better comparing to the FLC1. Controllers' performances evaluation face parameters uncertainties, loss of efficiency and wheel slip show the efficiency of the FLC 2 (good precision with small effective torque within suitable

simulation time) comparing to the FLC1. However, the presence of wheel slip leads to quiet similar poor performances for both controllers.

**Table V-6: Robustness analysis and controller performances**

Parameters of uncertainties	Parameters quantification	FLC1	FLC2
<b>Uncertainty on the radius R of the left wheel of the robot</b> $\Delta R_L = 25\%$ .	Time (sec)	799	815
	Error of distance (%)	0.184	0.027
	Error of orientation (%)	2.064	0.154
	Effective torque of the right motor (N.m)	0.100	0.088
	Effective torque of the left motor (N.m)	0.073	0.064
	Torque max of the right motor (N.m)	1.382	1.094
	Torque max of the left motor (N.m)	1.094	1.094
<b>Uncertainty on the radius R of the left wheel of the robot</b> $\Delta R_L = 50\%$ .	Time (sec)	890	814
	Error of distance (%)	9.837	0.629
	Error of orientation (%)	29.95	2.344
	effective torque of the right motor (N.m)	0.14	0.085
	effective torque of the left motor (N.m)	0.551	0.051
	torque max of the right motor (N.m)	1.372	1.101
	torque max of the left motor (N.m)	1.094	1.094
<b>Localization error</b> $\alpha = 0.003$	Time (sec)	809	817
	Error of distance (%)	0.282	0.374
	Error of orientation (%)	3.604	2.316
	effective torque of the right motor (N.m)	0.099	0.106
	effective torque of the left motor (N.m)	0.107	0.086
	torque max of the right motor (N.m)	1.391	1.391
	torque max of the left motor (N.m)	1.391	1.391
<b>Loss of efficiency1</b> $(\alpha_L = 0.1, \alpha_R = 0.6)$	Time (sec)	797	820
	Error of distance (%)	0.825	0.269
	Error of orientation (%)	1.117	4.450
	effective torque of the right motor (N.m)	0.183	0.156
	effective torque of the left motor (N.m)	1.109	0.684
	torque max of the right motor (N.m)	1.900	1.875
	torque max of the left motor (N.m)	4.389	4.565
<b>Loss of efficiency2</b> $(\alpha_L = 0.05,$	Time (sec)	1038	823
	Error of distance (%)	30.84	8.613

$\alpha_R = 0.005$ .	Error of orientation (%)	12.12	23.301
	effective torque of the right motor (N.m)	5.206	4.967
	effective torque of the left motor (N.m)	2.181	2.210
	torque max of the right motor (N.m)	12.59	12.809
	torque max of the left motor (N.m)	8.372	7.943
<b>Wheel slip1</b> $\alpha_{right} = 1,$ $\alpha_{left} = 3$	Time (sec)	1148	1041
	Error of distance (%)	77.82	38.727
	Error of orientation (%)	29.28	2.157
	effective torque of the right motor (N.m)	0.799	0.832
	effective torque of the left motor (N.m)	0.522	0.689
	torque max of the right motor (N.m)	2.486	3.074
	torque max of the left motor (N.m)	2.122	2.532
<b>Wheel slip2</b> $\alpha_{right} = 20,$ $\alpha_{left} = 5$	Time (sec)	1614	1591
	Error of distance (%)	315.3	282.21
	Error of orientation (%)	10.47	58.51
	effective torque of the right motor (N.m)	1.342	1.147
	effective torque of the left motor (N.m)	1.243	1.070
	torque max of the right motor (N.m)	5.219	4.884
	torque max of the left motor (N.m)	5.650	5.8488

## V.6. Conclusion

In this chapter, we made a comparison of type 1 FLC (FLC1) and Interval type 2 FLC (FLC2) to evaluate the performances of these controllers. This chapter includes also the description of V-REP simulator and Matlab/Simulink, these latter will be used to validate and evaluate all the aforementioned scenarios: The command and controller part are developed in Matlab when actuation and physical interaction part are created in V-REP. Three scenarios were considered; firstly, go to goal problem, secondly, multiple waypoints navigation, thirdly, robustness analysis under modeling error and parameters uncertainties. Many uncertainties are considered a deformation of the radius of one wheel (R), uncertainty of the mobile robot length (L), uncertainty of the localization, loss of efficiency of motors and Wheel slip. Both controllers are evaluated and validated using three simulation scenarios as follows:

- The kinematic model of the DDWMR.
- The dynamic model of DDWMR.
- VREP Validation.



## GENERAL CONCLUSION

## GENERAL CONCLUSION

In the majority of the research on the control of nonholonomic mobile robot, exact kinematic and dynamic models have been used. Such an assumption is, however, far from realistic in practice. An important issue for a real mobile robot design is the robustness consideration against possible modeling errors and uncertainties. The ideal constraints are never strictly satisfied, because in practical for the mobile robot, for various reasons there is violation of such assumptions. In practical standpoint models deviate from real systems due to perturbations such skidding of wheels and lateral slips (which violate the ideal nonholonomic constraints), unmodeled dynamics, parametric uncertainties in the models. In this context, we provide an efficient waypoint navigation solution for the DDWMR that is robust to face modeling errors and parameter uncertainties.

First, we gave an overview of mobile robotics in general and DDWMR in particular. DDWMR has generated great interest over the past decade; this is mainly due to the increasing advancements in instrumentation and computation technologies and the advantages that WMR offers over other models of mobile robots. We presented a state of the art on the modeling, navigation and robustness analysis of the DDWMR robot.

Then we described certain types of uncertainties encountered in the mobile robot that we used and modeled in our work (modeling error, location error, motor efficiency and wheel slip). Through the description of V-REP and Matlab / Simulink and V-REP Interfacing defining our application in our work.

We have moved on to modeling the DDWMR; first, the kinematic model, then the full dynamic model of this mobile robot is developed with and without considering the wheel slip, which is very important when using odometry for locating the robot, this is by using the formulation of Lagrange. With the parameters necessary for the simulation which were presented.

We provide an efficient waypoint navigation solution for the DDWMR that is robust to face modeling errors and parameter uncertainties. For this, two controllers are developed and implemented. Due to the effectiveness of fuzzy controllers in the navigation of mobile robots, we have presented an overview of type 1 fuzzy logic controllers and their limitations. We then presented the theory of the type 2 fuzzy logic controller and interval type 2.

In this work, two controllers (type 1 fuzzy logic and type 2 interval controllers) are implemented and compared to evaluate their performance with DDWMMR navigation under realistic conditions.

An in-depth analysis of the robustness of the two controllers is carried out under modeling error, parameter uncertainties, localization of errors, loss of efficiency and wheel slip. The proposed controllers are validated using the VREP simulator. The type 2 fuzzy controller compared to the type 1 fuzzy controller, especially when large errors and uncertainties are considered, performs well.

In addition, the type 2 fuzzy controller can maintain proper navigation performance even in the presence of longitudinal skidding of the wheels. The type 2 fuzzy controller performance decreases when large wheel pads are taken into account. As a solution, an adaptive controller based on the estimation of the ground model will be studied and developed in our next work.

## BIBLIOGRAPHY

## BIBLIOGRAPHY

- [1] T. Niemueller, G. Lakemeyer and A. Ferrein, "The RoboCup logistics league as a benchmark for planning in robotics," in *In Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS)—WS on Planning and Robotics (PlanRob)*, Jerusalem, Israel, 7–11 June 2015 .
  
- [2] V. Milanés and L. Bergasa, "Introduction to the special issue on," *New trends towards automatic vehicle control and perception systems. Sensors 2013*, vol. 13, pp. 5712-5719, 2013.
  
- [3] N. Nilsson, "A Mobile Automaton: An Application of Artificial Intelligence Techniques.," in *In Proceedings of the 1st International Joint Conference on Artificial intelligence (IJCAI'69)*, Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 1969; pp. 509–520, Washington, DC, USA, 7–9 May 1969.
  
- [4] J. YongZhai and S. Zbao, "Adaptive sliding mode trajectory tracking control of wheeled mobile robots," *International Journal of Control*, 2018.
  
- [5] J. Keighobadi and M. Menbaj, "From Nonlinear to Fuzzy Approaches in Trajectory Tracking Control of Wheeled Mobile Robots," *Asian Journal of Control*, vol. 14, no. 4, pp. 960-973, 2012.
  
- [6] J. Borenstein, "Experimental results from internal odometry error correction with the OmniMate mobile robot," *IEEE Transactions on Robotics and Automation* , vol. 14, no. 6, pp. 963-969, 1998.

- [7] D. Guha, P. Roy and S. Banerjee, "Grasshopper optimization algorithm-scaled fractional-order PI-D controller applied to reduced-order model of load frequency control," *Int J Model Simulat*, pp. 1-26, 2019.
- [8] M. Faisal, R. Hedjar and M. a. a. Al Sulaiman, "Fuzzy logic navigation and obstacle avoidance by a mobile robot in an unknown dynamic environment," *Int J Adv Rob Syst.*, vol. 10, no. 1, pp. 1-7, 2013.
- [9] M. Masmoudi, N. Krichen and M. Masmoudi, "Fuzzy logic controllers design for omnidirectional mobile robot navigation," *Appl Soft Comput*, vol. 49, pp. 901-919, 2016.
- [10] S. Das and A. Akella, "A fuzzy logic-based frequency control scheme for an isolated AC coupled PV-windbattery hybrid system.," *Int J Model Simulat*, pp. 1-13, 2019.
- [11] A. Khan and S. Kumar, "T-S fuzzy modeling and predictive control and synchronization of chaotic satellite systems," *Int J Model Simulat*, vol. 39, no. 3, p. 203–213, 2019.
- [12] M. Ebaid and A. Al-fahoum, "Design of fuel control system using fuzzy logic for a pre-designed radial gas turbine driving directly high-speed permanent magnet alternator," *Int J Model Simulat*, vol. 36, no. 4, p. 145–161, 2016.
- [13] J. Mendel and R. John, "Type-2 fuzzy sets made simple," *IEEE Trans Fuzzy Syst*, vol. 10, no. 2, p. 117–127., 2002.
- [14] J. Mendel and R. John, "A fundamental decomposition of type-2 fuzzy sets," *Proceedings of IFSA World Congress and 20th NAFIPS International Conference*, vol. 4, pp. 1896-1901, 2001.

- [15] R. John, P. Innocent and M. Barnes, "Neuro-fuzzy clustering of radiographictibia image data using type 2 fuzzy sets," *Inf Sci*, vol. 125, no. 1-4, pp. 25-82, 2000.
- [16] S. Abhishekh, S. Gautam and S. Singh, "A refined weighted method for forecasting based on type 2 fuzzy time series," *Int J Model Simulat*, vol. 38, no. 3, pp. 180-188, 2018.
- [17] T. Dereli, A. Baykasoglu and K. a. a. Altun, "Industrial applications of type-2 fuzzy sets and systems:A concise review," *Comput Ind*, vol. 62, no. 2, p. 125–137, 2011.
- [18] L. Zadeh, "The Concept of a Linguistic Variable and Its Application to Approximate Reasoning-i," *Information Sciences*, vol. 8, no. 13, pp. 199-249, 1975.
- [19] N. Karnik and J. Mendel, "Centroid of a type-2 fuzzy set," *Inf Sci*, vol. 132, no. 1–4, p. 195–220, 2001.
- [20] J. Mendel, R. John and F. Liu, "Interval Type-2 Fuzzy Logic Systems Made Simple," *IEEE Trans Fuzzy Syst*, vol. 14, no. 6, pp. 808-821, 2006.
- [21] H. Hagrass, "Type-2 FLCs: A New Generation of Fuzzy Controllers," *IEEE Comput Intel Mag*, vol. 2, no. 1, pp. 30-43, 2007.
- [22] S. Coupland and R. John, "New Geometric Inference Techniques for Type-2 Fuzzy Sets," *Int J Approx Reason*, vol. 49, no. 1, pp. 198-211, 2008.
- [23] N. Baklouti and A. Alimi, "The geometric interval type- 2 fuzzy logic approach in robotic mobile issue," in *IEEE International Conference on Fuzzy Systems*, Jeju Island, South Korea, 2009.

- [24] M. Hsiao and C. Wang, "Fuzzy interval control of mobile robots.," in *2013 International Conference on Advanced Robotics and Intelligent Systems*, Tainan, Taiwan, 2013.
- [25] H. Hagnas, "A hierarchical type-2 fuzzy logic control architecture for autonomous mobile robots," *IEEE Trans Fuzzy Syst*, vol. 12, no. 4, pp. 524-539, 2004.
- [26] E. Wijn, "Unstable behavior of a unicycle mobile robot with tracking control," *DCT rapporten*, 2004.
- [27] D. Filliat, *Robotique Mobile, Doctoral dissertation, EDX.*, 2011.
- [28] O. Trullier and J. Meyer, "Biomimetic navigation models and strategies in animats," *AI Communications*, no. 10, pp. 79-92, 1997.
- [29] V. Braitenberg, "Vehicles : Experiments in Synthetic Psychology," *The MIT Press*, 1986.
- [30] "marsrovers.nasa.gov/," [Online].
- [31] "www.abprecision.co.uk/," [Online].
- [32] "www.prod.sandia.gov/cgi-bin/techlib/access-control.pl/2001/013685.pdf," [Online].
- [33] R. Brockett, "Asymptotic Stability and Feedback Stabilization," *Differential Geometric Control Theory*, vol. 1, no. 27, pp. 181-191, 1983.
- [34] I. Kolmanovsky and M. N.H, "Developments In Nonholonomic Control Problems," *IEEE Control Systems Mag*, vol. 6, no. 15, pp. 20-36, 1995.



- [35] C. Canudas de Wit, H. Khenouf, C. Samson and O. Sordalen, "Nonlinear Control Design For Mobile Robots," *Recent Trends In Mobile Robots, Zheng Y.F., eds., World Scientific, Singapore*, pp. 121-156., 1993.
- [36] Z. Jiang and P. J.B, "Combining Backstepping and Time-Varying Techniques for A New Set of Adaptive Controllers," *Proc. of 33rd IEEE Conf. on Decision and Control, Lake Buena Vista*, , p. 2207–2212, 1994.
- [37] J. Guldner and V. Utkin, "Stabilization of Nonholonomic Mobile Robot Using Lyapunov Functions For Navigation And Sliding Mode Control," *Proc. of 33rd IEEE Conf. on Decision and Control, Lake Buena Vista*, pp. 2967-2972, 1994.
- [38] A. Astolfi, "Discontinuous control of nonholonomic systems," *System and Control Letters*, no. 27, p. 37–45, 1996.
- [39] F. Pourboghrat, "Exponential Stabilization of Nonholonomic Mobile Robots," *Elsevier Journal of Computers & Electrical Engineering*, vol. 5, no. 28, pp. 349-359, 2002.
- [40] G. Oriolo, A. De Luca and M. Vendittelli, "WMR Control via Dynamic Feedback Linearization: Design,Implementation, and Experimental Validation," *IEEE Transactions on Control Systems Technology*, vol. 6, no. 10, pp. 835-852, 2002.
- [41] B. d'Andrea-Novel, G. Bastin and G. Campion, "Dynamic feedback linearization of nonholonomic wheeled mobile robots," *In Proceedings 1992 IEEE International Conference on Robotics and Automation* , pp. 2527-2532, 1992, May.

- [42] Z. Wang, C. Su, T. Lee and S. Ge, "Robust adaptive control of a wheeled mobile robot violating the pure nonholonomic constraint," *In ICARCV 2004 8th Control, Automation, Robotics and Vision Conference*, vol. 2, pp. 987-992, 2004, December.
- [43] A. De Luca and M. Di Benedetto, "Control of nonholonomic systems via dynamic compensation.," *Kybernetika*, vol. 6, no. 29, pp. 593-608, 1993.
- [44] G. Klančar and I. Škrjanc, "Tracking-error model-based predictive control for mobile robots in real time," *Robotics and autonomous systems*, vol. 6, no. 55, pp. 460-469., 2007.
- [45] R. Soltani-Zarrin and S. Jayasuriya, "Constrained directions as a path planning algorithm for mobile robots under slip and actuator limitations.," *In 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems* , pp. 2395-2400, 2014, September.
- [46] R. Soltani-Zarrin, A. Zeiaee and S. Jayasuriya, " Pointwise Angle Minimization: A Method for Guiding Wheeled Robots Based on Constrained Directions.," *In ASME 2014 Dynamic Systems and Control Conference. American Society of Mechanical Engineers Digit*, 2014.
- [47] S. Thrun, W. Burgard and D. Fox, *Probabilistic Robotics*, USA: MIT Press: Cambridge, MA, 2005.
- [48] J. H. Shin, K. B. Park, S. W. Kim and J. J. Lee, "Robust adaptive control for robot manipulators using regressor-based form," *In Proceedings of IEEE International Conference on Systems, Man and Cybernetics* , vol. 3, pp. 2063-2068, (1994, October).
- [49] A. Geiger, P. Lenz and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," *In 2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3354-3361, (2012, June).

- [50] J. Sanchez, F. Denis, P. Checchin, F. Dupont and L. Trassoudaine, "Global registration of 3D LiDAR point clouds based on scene features: Application to structured environments.," *Remote Sensing*, vol. 9, no. 10, p. 1014, 2017.
- [51] F. Espinosa, C. Santos, M. Marrón-Romera, D. Pizarro, F. Valdés and J. Dongil, "Odometry and laser scanner fusion based on a discrete extended kalman filter for robotic platooning guidance.," *Sensors*, vol. 11, no. 9, pp. 8339-8357., 2011.
- [52] D. Scaramuzza and F. Fraundorfer, "Visual Odometry Part I: The First 30 Years and Fundamentals," *IEEE Robot. Autom. Mag.*, vol. 18, no. 80–92, 2011.
- [53] M. Montemerlo, J. Becker, S. Bhat, H. Dahlkamp, D. Dolgov, S. Ettinger, D. Haehnel, T. Hilden, G. Hoffmann and B. Huhnke, "Junior: The stanford entry in the urban challenge," *J. Field Robot*, vol. 25, p. 569–597, 2008.
- [54] R. Vivacqua, R. Vassallo and F. Martins, "A low cost sensors approach for accurate vehicle localization and autonomous driving application," *Sensors*, vol. 17, no. 10, p. 2359, 2017.
- [55] L. C. Bento, U. Nunes, F. Moita and A. Surrecio, " Sensor fusion for precise autonomous vehicle navigation in outdoor semi-structured environments," *In Proceedings. 2005 IEEE Intelligent Transportation Systems*, pp. 245-250, (2005, September).
- [56] D. Perea, J. Hernández-Aceituno, A. Morell, J. Toledo, A. Hamilton and L. Acosta, "MCL with sensor fusion based on a weighting mechanism versus a particle generation approach," *In 16th International IEEE Conference on Intelligent Transportation Systems (ITSC), The Hague, The Netherlands*, p. 166–171, (2013, October).
- [57] L. Wei, C. Cappelle and Y. & Ruichek, "Unscented information filter based multi-sensor data fusion using stereo camera, laser range finder and GPS receiver for vehicle

- localization," *In 2011 14th International IEEE Conference on Intelligent on Intelligent Transportation Systems (ITSC), Washington, DC, USA* , p. 1923–1928, (2011, October).
- [58] M. A. Moreira, H. N. Machado, C. F. Mendonca and G. A. Pereira, "Mobile robot outdoor localization using planar beacons and visual improved odometry," *In 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, p. 2468–2473, (2007, October).
- [59] R. Arnay, L. Acosta, M. Sigut and J. Toledo, "Ant colony optimisation algorithm for detection and tracking of non-structured roads," *Electronics Letters*, vol. 14, no. 12, pp. 725-727, 2008.
- [60] J. Hernandez-Aceituno, R. Arnay, J. Toledo and L. Acosta, "Using kinect on an autonomous vehicle for outdoors obstacle detection," *IEEE Sensors Journal*, vol. 16, no. 10, pp. 3603-3610, 2016.
- [61] N. Morales, J. T. Toledo, L. Acosta and R. Arnay, "Real-time adaptive obstacle detection based on an image database.," *Computer vision and image Understanding*, vol. 115, no. 9, pp. 1273-1287, 2011.
- [62] N. Morales, J. Toledo and L. Acosta, "Generating automatic road network definition files for unstructured areas using a multiclass support vector machine," *Information Sciences*, , vol. 329, pp. 105-124., 2016.
- [63] B. Vanderborght, R. Van Ham, D. Lefeber, T. Sugar and K. Hollander, "Comparison of mechanical design and energy consumption of adaptable, passive-compliant actuators.," *The International Journal of Robotics Research* , vol. 1, no. 28, pp. 90-103, 2009.

- [64] U. Mettin, P. La Hera, L. Freidovich and A. Shiriaev, "Parallel elastic actuators as a control tool for preplanned trajectories of underactuated mechanical systems," *The International Journal of Robotics Research*, vol. 9, no. 29, 2010.
- [65] S. Wang, W. van Dijk and H. van der Kooij, "Spring uses in exoskeleton actuation design," *In: Rehabilitation Robotics (ICORR), 2011 IEEE International Conference on*, pp. 1-6, 2011.
- [66] A. Jafari, N. Tsagarakis, I. Sardellitti and D. Caldwell, "How design can affect the energy required to regulate the stiffness in variable stiffness actuators," *In: Robotics and Automation (ICRA), 2012 IEEE International Conference on*, 2012.
- [67] M. Grimmer, M. Eslamy, S. Gliech and A. Seyfarth, "A comparison of parallel- and series elastic elements in an actuator for mimicking human ankle joint in walking and running," *In: Robotics and Automation (ICRA), 2012 IEEE International Conference on*, 2012.
- [68] E. Rouse, L. Mooney, E. Martinez-Villalpando and H. Herr, "Clutchable series-elastic actuator: Design of a robotic knee prosthesis for minimum energy consumption," *In: Rehabilitation Robotics (ICORR), 2013 IEEE International Conference on*, pp. 1-6, 2013.
- [69] S. Au and H. Herr, "Powered ankle-foot prosthesis," *Robotics Automation Magazine, IEEE*, vol. 3, no. 15, pp. 52-59, 2008.
- [70] M. Holgate, J. Hitt, R. Bellman, T. Sugar and K. Hollander, "The sparky (spring ankle with regenerative kinetics) project: Choosing a dc motor based actuation method," *In: 2nd IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechanics. IEEE*, pp. 163-168, 2008.

- [71] W. Brown and A. Ulsoy, *In: Robotics and Automation (ICRA), 2011 IEEE International Conference on*, 2011.
- [72] M. Zinn, B. Roth, O. Khatib and J. K. Salisbury, "A new actuation approach for human friendly robot design," *The International Journal of Robotics Research* , Vols. 4-5, no. 23 , pp. 379-398, 2004.
- [73] F. Roos, H. Johansson and J. Wikander, "Optimal selection of motor and gearhead in mechatronic applications," *Mechatronics* , vol. 1, no. 16, pp. 63-72, 2006.
- [74] H. Gao, X. Song, L. Ding, K. Xia, N. Li and Z. Deng, "Adaptive motion control of wheeled mobile robot with unknown slippage," *Int. J. Control*, vol. 87, no. 8, pp. 1513-1522, 2014.
- [75] M. Seyr and S. Jakubek, "Proprioceptive navigation, slip Estimation and slip control for autonomous wheeled mobile robot," in *IEEE Conf. on Robotics, Automation and Mechatronics*, 2006.
- [76] L. Chang Boon and W. Danwei, "Integrated Estimation for Wheeled Mobile Robot posture, velocities, and wheel skidding perturbations," in *Proc. ICRA*,, Roma, Italy, 2007.
- [77] J. Ryu and S. Agrawal, "Differential flatness-based robust control of mobile robots in the presence of slip, Vol 30, pp.," *Int. J. Robot. Res*, vol. 30, pp. 463-475, 2011.
- [78] Y. Tian and N. Sarkar, "Control of a mobile robot subject to wheel slip," *J. Intell. Robot. Syst*, vol. 74, pp. 915-929, 2014.

- [79] A. Mahadevuni and P. Li, "Navigating mobile robots to target in near shortest time using reinforcement learning with spiking neural networks.," in *In 2017 International Joint Conference on Neural Networks (IJCNN)*, 2017.
- [80] T. Nguyen, K. Nguyentien and P. Do T, "Neural network-based adaptive sliding mode control method for tracking of a nonholonomic wheeled mobile robot with unknown wheel slips, model uncertainties, and unknown bounded external disturbances.," *Acta Polytechnica Hungarica*, vol. 15, no. 2, pp. 103-23, 2018.
- [81] N. Ghobadi and S. Dehkordi, "Dynamic modeling and sliding mode control of a wheeled mobile robot assuming lateral and longitudinal slip of wheels.," in *In 2019 7th International Conference on Robotics and Mechatronics (ICRoM)*, 2019.
- [82] J. Borenstein, "Internal Correction of Dead-reckoning Errors with the Compliant Linkage Vehicle," *Journal of Robotic Systems*, vol. 12, no. 4, pp. 257-273, 1995.
- [83] T. Verstraten, G. Mathijssen, R. Furnemont, B. Vanderborght and D. Lefeber, "Modeling and dimensioning of geared DC motors for energy efficiency: Comparison between theory and experiments," *Mechatronics*, no. 30, pp. 198-213, 2015.
- [84] S. Dereyne, K. Stockman, S. Derammelaere and P. Defreyne, *In: Energy Efficiency in Motor Driven Systems (EEMODS), 2011 7th International Conference on*, 2011.
- [85] H. Pacejka, *Tyre and Vehicle Dynamics*, Butterworth-Heinemann: Second Edition. Elsevier, 2006.
- [86] H. Dugoff, P. Fancher and L. Segel, "Tire Performance Characteristics Affecting Vehicle Response to Steering and Braking Control Inputs," CST 460, Michigan, Aug 1969.

- [87] J. Svendenius, "Tire Models for Use in Braking Applications," 2003.
- [88] J. Svendenius and B. Wittenmark, "Brush Tire Model with Increased Flexibility," in *2003 European Control Conference (ECC), Cambridge, UK*, pp. 1863-1868, 2003.
- [89] H. Pacejka, *Tire and Vehicle Dynamics*, Oxford: Butterworth-Heinemann, 2002.
- [90] F. Demirbaş and M. Kalyoncu, "Differential drive mobile robot trajectory tracking with using pid and kinematic based backstepping controller.," *Selçuk Üniversitesi Mühendislik, Bilim ve Teknoloji Dergisi*, vol. 5, no. 1, pp. 1-15, 2017.
- [91] T. Mac, C. Copot, R. De Keyser, T. T. D and T. Vu, "MIMO Fuzzy Control for Autonomous Mobile Robot," *Journal of Automation and Control Engineering*, vol. 4, no. 1, pp. 65-70, 2016.
- [92] T. Fukao, H. Nakagawa and N. Adachi, "Adaptive tracking control of a non-holonomic mobile robot," *IEEE Trans Rob Autom*, vol. 16, no. 5, pp. 609-615, 2000.
- [93] X. Yun and Y. Yamamoto, "Internal dynamics of a wheeled mobile robot," in *In Proceedings of 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '93)*, Yokohama, Japan, 1993.
- [94] R. DeSantis, "Modeling and path-tracking control of a mobile wheeled robot with a differential drive," *Robotica*, vol. 13, no. 4, pp. 401-410, 1995.
- [95] K. Thanjavur and R. Rajagopalan, "Ease of dynamic modeling of wheeled mobile robots (WMRs) using Kane's approach," in *In: Proceedings of International Conference on Robotics and Automation*, Albuquerque (NM), 2002.



- [96] Y. Tian, N. Sidek and N. Sarkar, "Modeling and control of a non-holonomic wheeled mobile robot with wheel slip dynamics," *Symposium on Computational Intelligence in Control and Automation; IEEE*, 2009.
- [97] R. Solea, A. Filipescu, A. Filipescu, E. Minca and S. Filipescu, "Wheelchair control and navigation based on kinematic model and iris movement.," *In 2015 IEEE 7th International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE*, pp. 78-83, 2015.
- [98] J. Mendel, H. Hagra, W. W. Tan, W. W. Melek and H. Ying, Introduction to type-2 fuzzy logic control: theory and applications, John Wiley & Sons, 2014.
- [99] N. Karnik, "Type-2 Fuzzy Logic Systems," in *Ph.D. Dissertation*, Los Angeles, CA, University of Southern California, 1998.
- [100] O. Castillo and P. Melin, "Type-2 Fuzzy Logic: Theory and Application," *Stufuzz* 223, pp. 29-43, 2008.
- [101] J. Mendel, "Type-2 Fuzzy Sets and Systems: An overview," *IEEE Computational Intelligence Magazine*, pp. 20-29, 2007.
- [102] E. Jammeh, M. Fleury, C. Wagner, H. Hagra and M. Ghanbari, "Interval Type-2 Fuzzy Logic Congestion Control for Video Streaming Across IP Networks," *IEEE Trans. Fuzzy Syst*, vol. 17, no. 5, 2009.
- [103] D. Wu, "Approaches for reducing the computational cost of interval type-2 fuzzy logic systems: overview and comparisons," *IEEE Transactions on Fuzzy Systems*, vol. 21, no. 1, pp. 80-99, 2012.

- [104] J. Roger Jang and N. Gulley, "MATLAB: Fuzzy logic toolbox user's guide: Version 1," *Math Works*, 1997.
- [105] A. Taskin and T. Kumbasar, "An open source Matlab/Simulink toolbox for interval type-2 Fuzzy logic systems," in *IEEE Symposium Series on Computational Intelligence*, Cape Town, South Africa, 2015.
- [106] "V-REP: Virtual Robot Experimentation Platform," Coppelia Robotics, 02 2014. [Online]. Available: <http://www.coppeliarobotics.com>.
- [107] "www.angelusresearch.com," [Online].
- [108] R. Volpe, "Navigation results from desert field tests of the Rocky 7 Mars rover prototype," *Int. Journal of Robotics Research*, vol. 7, no. 18, pp. 669-683, 1999.
- [109] A. Angelova, L. Matthies, D. Helmick, G. Sibley and P. Perona, "Learning to predict slip for ground robots," in *Proc. of IEEE Int. Conf. on Robotics and Automation*, 2006.
- [110] A. Salerno and J. Angeles, "A new family of two-wheeled mobile robots: modeling and controllability robotics," *IEEE Transactions on Robotics and Automation*, vol. 1, no. 23, pp. 169-173, 2007.
- [111] M. Aicardi, G. Casalino, A. Bicchi and A. Balestrino, "Closed loop steering of unicycle like vehicles via Lyapunov techniques.," *Robotics & Automation Magazine, IEEE*, no. 2, p. 27–35, 1995.

- [112] A. Conceicao, H. Oliveira, A. Sousa e Silva, D. Oliveira and A. Moreira, "A nonlinear model predictive control of an omni-directional mobile robot," *IEEE Int. Symposium on Industrial Electronics ISIE07*, p. 2161 – 2166, 2007.
- [113] C. Ward and K. Iagnemma, "Model-based wheel slip detection for outdoor mobile robots," *IEEE Int. Conf. on Robotics and Automation*, p. 2724 – 2729, 2007.
- [114] D. Stonier, C. Se-Hyoung, C. Sung-Lok, N. Kuppaswamy and K. Jong-Hwan, "Nonlinear slip dynamics for an omniwheel mobile robot platform," in *IEEE Int. Conf. on Robotics and Automation*,, 2007.
- [115] D. Helmick, Y. Cheng, D. Clouse, L. Matthies and S. Roumeliotis, "Path following using visual odometry for a Mars rover in high-slip environments," in *Proc. IEEE Aerospace Conference*, 2005.
- [116] D. Helmick, Y. Cheng, D. Clouse, M. Bajracharya and L. Matthies, "Slip compensation for a Mars rover," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS05*, 2005.
- [117] G. Ishigami, K. Nagatani and K. Yoshida, "Path planning for planetary exploration rovers and its evaluation based on wheel slip dynamics," *Proc. of IEEE Int. Conf. on Robotics and Automation*, 2007.
- [118] I. Motte and G. Campion, "A Slow manifold approach for the control of mobile robots not satisfying the kinematic constraints," *IEEE Trans. on Robotics and Automation*,, vol. 6, no. 16, pp. 875-880, 2000.
- [119] M. Eghtesad and D. Neacsulescu, "Study of the internal dynamics of an autonomous mobile robot," *Robotics and Autonomous Systems*, vol. 54, pp. 342-349, 2006.

- [120] M. Tarokh and G. McDermott, "Kinematics modeling and analyses of articulated rover," *IEEE Trans. on Robotics*, vol. 4, no. 21, pp. 539-553, 2005.
- [121] N. Sarkar, X. Yun and V. Kumar, "Dynamic path following: a new control algorithm for mobile robots," in *Proc. of the 32nd IEEE Conference on Decision and Control*, 1993.
- [122] R. Balakrishna and A. Ghosal, "Modeling of slip for wheeled mobile robot," *IEEE Trans. on Robotics and Automation*, vol. 1, no. 11, pp. 126-132, 1995.
- [123] L. Ray, "Nonlinear tire force estimation and road friction identification: simulation and experiments," *Journal of Automatica*, vol. 10, no. 33, pp. 1819-1833, 1997.
- [124] S. Jung and T. Hsia, "Explicit lateral force control of an autonomous mobile robot with slip", in *EEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS05*, 2005.
- [125] W. Dixon, D. Dawson and E. Zergeroglu, "Robust control of a mobile robot system with kinematic disturbance," *IEEE Int. Conference on Control Applications*, pp. 437-442, 2000.
- [126] W. Lin, L. Chang and P. Yang, "Adaptive critic anti-slip control of wheeled autonomous robot," *Control Theory & Applications, IET*, vol. 1, no. 1, pp. 51-57, 2007.
- [127] X. Zhu, G. Dong, D. Hu and Z. Cai, "Robust tracking control of wheeled mobile robots not satisfying nonholonomic constraints," in *Proc. of the 6th Int. Conf. on Intelligent Systems Design and Applications, ISDA06*, 2006.
- [128] Y. Liyong and X. Wei, "An adaptive tracking method for non-holonomic wheeled mobile robots," in *Control Conference, CCC07*, Chinese, 2007.

- [129] R. Fierro and F. L. Lewis, "Control of a Nonholonomic Mobile Robot: Backstepping Kinematics into Dynamics," *Journal of Robotic Systems*, vol. 3, no. 14, pp. 149-163, 1997.
- [130] G. Klancar, D. Matko and S. Blazic, "Wheeled Mobile Robots Control in a Linear Platoon," *Journal of Intelligent Robot System*, no. 54, p. 709–731, 2009.
- [131] Q. Zhang, J. Shippen and B. Jones, "Robust backstepping and neural network control of a low-quality nonholonomic mobile robot," *International Journal of Machine Tools and Manufacture*, vol. 7, no. 39, p. 1117–1134, 1999.
- [132] K. Choi, S. J. Yoo, J. B. Park and Y. H. Choi, "Adaptive formation control in absence of leader's velocity information," *IET control theory & applications*, vol. 4, no. 4, p. 521–528, 2010.
- [133] 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 Engineering Practice*, vol. 11, no. 16, p. 1354–1363, 2008.
- [134] R. Rashid, I. amvazuthi, M. Beam and M. Arrofiq, "Fuzzy-based navigation and control of a non-holonomic mobile robot," *arXiv preprint arXiv:1003.4081*, 2010.
- [135] S. Liu, Q. Yu, W. Lin and S. Yang, "Tracking control of mobile robots based on improved rbf neural networks," *In Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, p. 1879–1884, 2006.
- [136] C. Chin-Yang, S. Tzuu-Hseng, Y. Yang-Chieh and C. Chang, "Design and implementation of an adaptive sliding mode dynamic controller for wheeled mobile robots," *Mechatronics*, no. 19, p. 156–166, 2009.

- [137] P. Bong Seok, Y. Sung Jin, P. Jin Bae and Y. H. Choi, "A Simple Adaptive Control Approach for Trajectory Tracking of Electrically Driven Nonholonomic Mobile Robots," *IEEE Transactions on Control Systems Technology*, vol. 5, no. 18, p. 1199–1205, 2010.
- [138] R. Fierro and F. Lewis, "Control of a nonholonomic mobile robot using neural networks.," *Neural Networks, IEEE Transactions on*, vol. 4, no. 9, p. 589–600, 1998.
- [139] N. Sarkar, X. Yun and V. Kumar, "Control of mechanical systems with rolling constraints application to dynamic control of mobile robots," *The International Journal of Robotics Research*, vol. 1, no. 13, pp. 55-69, 1994.
- [140] C. Oh, M. Kim, J. Lee and J. Lee, "Control of mobile robots using RBF network," *In Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003,IEEE/RSJ International Conference on*, vol. 4, p. 3528–3533, 2003.
- [141] M. Bugeja, S. Fabri and L. Camilleri, "Dual adaptive dynamic control of mobile robots using neural networks," *Systems, Man, and Cybernetics, Part B: Cybernetics IEEE Transactions on*, , vol. 1, no. 39, pp. 129-141, 2009.
- [142] D. Parhi and M. Singh, "Intelligent fuzzy interface technique for the control of an autonomous mobile robot," *Proceedings of the Institution of Mechanical Engineers,Part C: Journal of Mechanical Engineering Science*, vol. 11, no. 222, p. 2281–2292., 2008.
- [143] V. Peri and D. Simon, "Fuzzy logic control for an autonomous robot," *In Fuzzy Information Processing Society, 2005. NAFIPS 2005. Annual Meeting of the North American*, p. 337–342, 2005.

- [144] T. Das, I. Kar and S. Chaudhury, "Simple neuron-based adaptive controller for a nonholonomic mobile robot including actuator dynamics," *Neurocomputing*, vol. 16, no. 69, p. 2140–2151., 2006.
- [145] K. Young, V. Utkin and U. Ozguner, "A control engineer's guide to sliding mode control," *Control Systems Technology, IEEE Transactions on*, vol. 3, no. 7, p. 328–342., 1999.
- [146] A. Bloch and S. Drakunov, "Stabilization and tracking in the nonholonomic integrator via sliding modes," *Systems & Control Letters*, vol. 2, no. 29, p. 91–99., 1996.
- [147] J. Yang and J. Kim, "Sliding mode control for trajectory tracking of nonholonomic wheeled mobile robots," *Robotics and Automation, IEEE Transactions on*, vol. 3, no. 15, p. 578–587., 1999.
- [148] D. Chwa, "Sliding-mode tracking control of nonholonomic wheeled mobile robots in polar coordinates," *Control Systems Technology, IEEE Transactions on*, vol. 4, no. 12, p. 637–644, 2004.
- [149] T. Li and Y. Huang, "MIMO adaptive fuzzy terminal sliding-mode controller for robotic manipulators," *Information Sciences*, , vol. 23, no. 180, p. 4641–4660., 2010.
- [150] H. Mehrjerdi, Y. Zhang and M. Saad, "Adaptive exponential sliding mode control for dynamic tracking of a nonholonomic mobile robot," *In Intelligent Robotics and Applications. Springer.*, p. 643–652, 2012.
- [151] B. Park, S. Yoo, J. B. Park and Y. Choi, " Adaptive neural sliding mode control of nonholonomic wheeled mobile robots with model uncertainty," *Control Systems Technology, IEEE Transactions on*, vol. 1, no. 17, p. 207–214., 2009.

- [152] F. Rossomando, "Sliding Mode Control for Trajectory Tracking of a Nonholonomic Mobile Robot using Adaptive Neural Network," *Journal of Control Engineering and Applied Informatics*, , vol. 1, no. 16, p. 12–21., 2014.
- [153] C. Chen, T. Li and Y. Yeh, "EP-based kinematic control and adaptive fuzzy sliding-mode dynamic control for wheeled mobile robots," *Information Sciences*, vol. 1, no. 179, p. 180–195., 2009.
- [154] O. Castillo, Introduction to type-2 fuzzy logic control. In : Type-2 Fuzzy Logic in Intelligent Control Applications., Berlin, Heidelberg: Springer, 2012, pp. 3-5.
- [155] H. Shu, Q. Liang and J. Gao, "Wireless Sensor Network Lifetime Analysis Using Interval Type-2 Fuzzy Logic Systems," *IEEE Trans. Fuzzy Syst*, vol. 16, no. 2, 2008.
- [156] J. Jang and N. Gulley, "MATLAB fuzzy logic toolbox," *The MathWorks Inc., Natick, MA.*, 1997.
- [157] F. L. Lewis, A. Yesildirek and K. Liu, "Multilayer neural-net robot controller with guaranteed tracking performance," *IEEE Trans. Neural Networks*, vol. 7, pp. 1-12, 1996.
- [158] S. Yoo, Y. Choi and J. Park, "Generalized predictive control based on self-recurrent wavelet neural network for stable path tracking of mobile robots: Adaptive learning rate approach," *IEEE Trans. Circuit Syst. I, Reg Papers*, vol. 53, no. 6, pp. 1381-1394, 2006.
- [159] S. Milos, "Roadmap methods vs. cell decomposition in robot motion planning.," in *In: Proceeding of the 6th WSEAS international conference on signal processing, robotics and automation. World Scientific and Engineering Academy and Society (WSEAS); .*, 2007.



- [160] W. Regli, "Robot Lab: robot path planning," Lecture notes of department of computer science., Drexel University, Oct 2007..
- [161] T. Schwartz and M. Sharir, "On the "Piano Movers" Problem I. The case of a twodimensional rigid polygonal body moving amidst polygonal barriers," *Commun Pure Appl Math*, vol. 3, no. 36, pp. 345-398, 1983.
- [162] M. Weigl, B. Siemiaatkkowska, A. Sikorski and A. Borkowski, "Grid-based mapping for autonomous mobile robot.," *Robot Autonom Syst* , vol. 1, no. 11, pp. 13-21, 1993.
- [163] D. Zhu and J. Latombe, "New heuristic algorithms for efficient hierarchical path planning," *IEEE Trans Robot Autom*, vol. 1, no. 7, pp. 9-20, 1991.
- [164] Z. R. Conte G, "Hierarchical path planning in a multi-robot environment with a simple navigation function," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 4, no. 25, pp. 651-654, 1995.
- [165] H. Samet, ""An overview of quadtree" octrees and related hierarchical data structures," *NATO ASI Series, F40*, 1988.
- [166] H. Noborio, T. Naniwa and S. Arimoto, "A quadtree-based path-planning algorithm for a mobile robot," *J Robot Syst* , vol. 4, no. 7, pp. 555-574, 1990.
- [167] Choset, H. Burdick and Joel, "Sensor-based exploration: the hierarchical generalized Voronoi graph," *Int J Robot Res* , vol. 2, no. 19, pp. 96-125, 2000.

- [168] L. Lulu and A. Elnagar, "A comparative study between visibility-based roadmap path planning algorithms," *In: 2005 IEEE/RSJ international conference on intelligent robots and systems*, pp. 3263-3268, 2005.
- [169] M. Berg, M. Kreveld, M. Overmars and O. Schwarzkopf, "Computational geometry," *Springer Berlin Heidelberg*, Vols. 1-17, 2000.
- [170] H. M. Choset, S. Hutchinson, K. M. Lynch, G. Kantor, W. Burgard, L. E. Kavraki and S. Thrun, "Principles of robot motion: theory, algorithms, and implementation," *MIT press*, 2005.
- [171] O. Takahashi and R. J. Schilling, "Motion planning in a plane using generalized Voronoi diagrams," *IEEE Transactions on robotics and automation*, vol. 2, no. 5, pp. 143-150, 1989.
- [172] C. Ó'Dúnlaing and C. K. Yap, "A "retraction" method for planning the motion of a disc.," *Journal of Algorithms*, vol. 1, no. 6, pp. 104-111, 1985.
- [173] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *In Autonomous robot vehicles*, New York, NY, 1986.
- [174] H. J. Bremermann, *The evolution of intelligence: The nervous system as a model of its environment.*, University of Washington, Department of Mathematics., 1958.
- [175] J. H. Holland, *Adaptation in natural and artificial systems.*, 1992.
- [176] L. A. Zadeh, "Fuzzy sets as a basis for a theory of possibility.," *Fuzzy sets and systems*, vol. 1, no. 1, pp. 3-28, 1978.

- [177] J. K. Pothal and D. R. Parhi, "Navigation of multiple mobile robots in a highly clutter terrains using adaptive neuro-fuzzy inference system," *Robotics and Autonomous Systems*, no. 72, pp. 48-58, 2015.
- [178] X. S. Yang, "Nature-inspired metaheuristic algorithms," *Luniver press*, 2010.
- [179] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory.," in *In MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, 1995, October.
- [180] M. Dorigo and L. M. Gambardella, "Ant colony system: a cooperative learning approach to the traveling salesman problem," *IEEE Transactions on evolutionary computation*, vol. 1, no. 1, pp. 53-66, 1997.
- [181] K. M. Passino, "Biomimicry of bacterial foraging for distributed optimization and control," *IEEE control systems magazine*, vol. 3, no. 22, pp. 52-67, 2002.
- [182] L. Tan, H. Wang, C. Yang and B. Niu, "A multi-objective optimization method based on discrete bacterial algorithm for environmental/economic power dispatch," *Natural computing*, vol. 4, no. 16, pp. 549-565, 2017.
- [183] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," *Technical report-tr06*, 2005.
- [184] X. Yang and S. Deb, "Cuckoo search via Levy flights," in *In: Nature and biologically inspired computing*, 2009.

- [185] M. Eusuff and K. Lansey, "Optimization of water distribution network design using the shuffled frog leaping algorithm.," *J Water Resour Plan Manag*, vol. 3, no. 129, pp. 210-225, 2003.
- [186] P. Mohanty and D. Parhi, "A new efficient optimal path planner for mobile robot based on Invasive Weed Optimization algorithm," *Front Mech Eng*, vol. 4, no. 9, pp. 317-330, 2014.
- [187] P. D. Kundu S, "Navigation of underwater robot based on dynamically adaptive harmony search algorithm," *Memetic Computing* , vol. 2, no. 8, pp. 125-146, 2016.
- [188] J. Guo, G. Yu and G. Coi, "The path planning for mobile robot based on bat algorithm," *Int J Autom Control* , vol. 1, pp. 50-60, 2015.
- [189] D. Parhi and S. Kundu, "Navigational control of underwater mobile robot using dynamic differential evolution approach," *Proc IME M J Eng Marit Environ*, vol. 1, no. 231, pp. 284-301, 2017.
- [190] A. Savkin and C. Wang, "A simple biologically-inspired algorithm for collision free navigation of a unicycle-like robot in dynamic environments with moving obstacles," *Robotica*, vol. 6, no. 31, pp. 993-1001, 2013.
- [191] H. Thanh, N. Phi and S. K Hong, "Simple nonlinear control of quadcopter for collision avoidance based on geometric approach in static environment," *Int J Adv Robot Syst*, vol. 2, no. 15, 2018.
- [192] B. K. Patle, A. Pandey, D. R. K. Parhi and A. Jagadeesh, "A review: On path planning strategies for navigation of mobile robot.," *Defence Technology*, 2019.

[193] O. Trullier, S. I. Wiener, A. Berthoz and J. A. Meyer, "Biologically based artificial navigation systems: Review and prospects," *Progress in neurobiology*, vol. 51, no. 5, pp. 483-544, 1997.

[194] D. Filliat, Robotique mobile. Doctoral dissertation, EDX,, 2011.