

**MULTI-ROBOT FORMATION WITH THE CLUSTER
SPACE REPRESENTATION**

BY

MOHAMMAD TARIQ MOUSA NASIR

A Dissertation Presented to the
DEANSHIP OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

In

SYSTEMS & CONTROL ENGINEERING

APRIL 2017

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS
DHAHRAN 31261, SAUDI ARABIA

DEANSHIP OF GRADUATE STUDIES

This thesis, written by **MOHAMMAD TARIQ MOUSA NASIR** under the direction of his thesis adviser and approved by his thesis committee, has been presented to and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of **DOCTOR OF PHILOSOPHY IN SYSTEMS & CONTROL ENGINEERING**.

Dissertation Committee



Dr. Sami El-Ferik (Adviser)



Dr. Lahouari Cheded (Member)



Dr. M. El-Shafie (Member)




Dr. M A Abido (Member)



Dr. M Hawwa (Member)



Dr. Hesham K. Al-Fares
Department Chairman


Dr. Salam A. Zummo
Dean of Graduate Studies



4/6/17
Date

©Mohammad Tariq Mousa Nasir
2017

To my Family

ACKNOWLEDGMENTS

In the name of Allah, the Most Gracious and the Most Merciful Alhamdulillah, all praises to Allah for the strengths and His blessing in completing this thesis. Special appreciation goes to my supervisor, Dr Sami El-Ferik, for his supervision and constant support. His invaluable help of constructive comments and suggestions throughout the experimental and thesis works have contributed to the success of this research. My acknowledgment also goes to all the committee members Dr Mustafa Elshafie, Dr M A Abido, Dr Lahouari Cheded, Dr M Hawwa for their valuable comments and ideas. And thanks to Dr Uthman Baroudi for his support. Sincere thanks to all my friends and colleagues at KFUPM for their kindness and moral support during my study. Thanks for the friendship and memories. And the last but not the least thanks to my beloved parents for teaching me the value of the knowledge. And many thanks to my beloved wife for supporting me during my study. To those who indirectly contributed in this research, your kindness means a lot to me. Thank you very much.

TABLE OF CONTENTS

ACKNOWLEDGEMENT	iii
LIST OF TABLES	viii
LIST OF FIGURES	ix
LIST OF ABBREVIATIONS	xii
ABSTRACT (ENGLISH)	xiii
ABSTRACT (ARABIC)	xv
CHAPTER 1 INTRODUCTION	1
1.1 Preliminaries	3
1.1.1 Classification of multi robot control approaches	3
1.2 Literature Survey	5
1.3 Problem statement	9
1.4 Steps and control methodology	10
1.5 Significance of Research	11
1.6 Experimental setup	11
1.7 Thesis organization	13
CHAPTER 2 CLUSTER SPACE REPRESENTATION OF NON HOLONOMIC ROBOTS	15
2.1 Preliminaries	15

2.1.1	Nonholonomic robot dynamics	16
2.1.2	Overview of cluster space framework	16
2.1.3	Three robot cluster	18
2.2	Cluster space dynamics of nonholonomic robots	19
2.2.1	Single nonholonomic robot modeling	19
2.2.2	Cluster space modeling	20
CHAPTER 3 BEHAVIORAL INTELLIGENT KINEMATIC		
CONTROLLERS		24
3.1	Control Architecture	25
3.2	The Fuzzy controller design:	28
3.3	BAFC structure	31
3.4	Stability Proof	33
3.5	Simulation Results	35
3.5.1	Two-robot simulation	35
3.5.2	Three-robot simulation	36
3.6	Experimental setup	38
3.6.1	Experiment test	38
3.7	Chapter summary	39
CHAPTER 4 SLIDING MODE CONTROLLER FOR NON-		
HOLONOMIC MULTI-ROBOT CLUSTER		46
4.1	Controller design	48
4.1.1	The cluster profile errors	48
4.1.2	Sliding mode derivation	49
4.1.3	Adaptive sliding mode	53
4.1.4	Control diagram	55
4.2	Simulation Results	55
4.2.1	Two robot simulation	55
4.2.2	Simulation results of 3-robot system	59
4.3	Experimental application	61

4.3.1	Experiment test	62
4.4	Chapter summary	66
CHAPTER 5 FUZZY ADAPTIVE SLIDING MODE CLUSTER		
SPACE CONTROL OF A NONHOLONOMIC MULTI-ROBOT		
SYSTEM		68
5.1	Introduction	68
5.2	Controller design	70
5.2.1	The cluster profile errors	70
5.2.2	Sliding mode derivation	72
5.2.3	Fuzzy Adaptive sliding mode	75
5.2.4	Control diagram	79
5.3	Simulation Results	79
5.3.1	Two-robot simulation	80
5.3.2	Simulation results of 3-robot system	82
5.4	Experimental application	84
5.4.1	Experiment test	87
5.5	Chapter summary	87
CHAPTER 6 MPC CONTROLLER FOR ACTUATOR SATURA-		
TION ISSUE WITH CLUSTER SPACE		90
6.1	Introduction	90
6.2	Differential Robot velocity constraints	91
6.2.1	Diamond shape velocity constraint	91
6.3	Controller design	93
6.3.1	The cluster profile errors	93
6.3.2	Kinematic Controller	94
6.3.3	MPC Controller	95
6.3.4	Control diagram	97
6.4	Simulation Results	98
6.4.1	Two robot simulation	98

6.5	Experimental application	99
6.5.1	Experiment test	102
6.6	Chapter summary	108
CHAPTER 7 CONCLUSION AND FUTURE WORK		110
REFERENCES		112
VITAE		126
Appendices		127
.1	A Behavioral Adaptive Fuzzy controller of multi robots in a cluster space	128
.2	Adaptive sliding-mode cluster space control of a non-holonomic multi-robot system with applications	140

LIST OF TABLES

1.1	Literature survey	9
3.1	The fuzzy rules of ΔK_p provided that P is low	30
3.2	The fuzzy rules of ΔK_d Provided that P is low	30
3.3	The Fuzzy Rules of P	30
5.1	The fuzzy rules of γ_i	75

LIST OF FIGURES

1.1	Lego EV3 WMRs	13
2.1	A WMR model, where q is the WMR's heading angle, point C is the robots center of gravity, d is the distance between the wheels line and C and ρ is the wheel radius	16
2.2	Cluster space and robot space variables	17
2.3	3-robot system configuration	18
3.1	Cluster space control architecture where Δr is the robot commands and J is the Jacobian Matrix	25
3.2	Fuzzy control parts	29
3.3	The member ship function of normalized $E, \dot{E}, \Delta K_p, \Delta K_d$	31
3.4	The Membership function for the Priority input P	31
3.5	The BAFC structure for a two-robot cluster	32
3.6	A comparison between the classical PD controller and BAFC. The solid blue lines are the system response with classical PD controller, and the dashed red lines are BAFC responses.	36
3.7	Robots motion with Fuzzy adaptive cluster controller	37
3.8	Simulation results for cluster space control of a 3-robot system	41
3.9	comparison between the control signal from the classical controller and the BAFC.	42
3.10	experiment with two WMRs	43
3.11	The Experimental Robot moving profile with standard PD controller	44
3.12	Experimental Robots moving profile with Fuzzy PD controller	44

3.13	Experimental results	45
4.1	Cluster location errors	49
4.2	The proposed adaptive SMC diagram	55
4.3	Adaptive SMC control path tracking profile	57
4.4	Comparison between SMC and Adaptive SMC with disturbance injected after 20 sec	58
4.5	The adaptive parameters response with no disturbance case	58
4.6	3-robot system configuration	59
4.7	Adaptive SMC control path tracking profile	60
4.8	Using an integrator to change an SMC command to a velocity command calculation and the inner loop diagram	62
4.9	Experimental cluster space control on a 2-robot system	63
4.10	Experimental movement trajectory of a cluster of two robot with SMC	64
4.11	Experimental movement trajectory of a cluster of two robot with adaptive SMC	65
4.12	The experimental desired and actual cluster dynamics states x_c, y_c, Q_c and d_c , and a comparison between the SMC and the adaptive SMC	66
5.1	Cluster location errors	71
5.2	Fuzzy structure	76
5.3	The membership function for the s_i and \dot{s}_i	76
5.4	The membership function for the output γ_i	77
5.5	The fuzzy surface	78
5.6	The proposed fuzzy adaptive SMC diagram	79
5.7	Fuzzy Adaptive SMC control path tracking profile	81
5.8	Comparison between Adaptive SMC and standard SMC and fuzzy adaptive SMC with disturbance injected after 10 sec	82
5.9	3-robot system configuration	83

5.10 Adaptive SMC control path tracking profile	84
5.11 Using an integrator to change a fuzzy SMC command to a velocity command calculation and the inner loop diagram	86
5.12 Experimental cluster space control on a 2-robot system	88
5.13 Experimental profile tracking results of a two robot cluster with fuzzy adaptive SMC	88
5.14 A comparison between the tracking errors with adaptive SMC and fuzzy adaptive SMC	89
6.1 Differential drive velocities domain's	92
6.2 Cluster location errors	94
6.3 The proposed MPC diagram	98
6.4 MPC control path tracking profile	100
6.5 Comparison between kinimatic with and without saturation and MPC with saturation	101
6.6 MPC velocity error is filtered using an integrator to get a velocity command that is sent to the robot inner loop diagram	102
6.7 Experimental cluster space control on a 2-robot system	104
6.8 Experimental movement trajectory of a cluster of two robot with Kinimatic controller and without velocity saturation	105
6.9 Experimental movement trajectory of a cluster of two robot with Kinimatic controller and robot velocity saturation	106
6.10 Experimental movement trajectory of a cluster of two robot with MPC and robot velocity saturation	107
6.11 The experimental trajectory errors of the cluster dynamics states x_c, y_c, Q_c and d_c , and a comparison between the kinimatic with/without velocity saturation and MPC with velocity satura- tion	108

THESIS ABSTRACT

NAME: Mohammad Tariq Mousa Nasir

TITLE OF STUDY: Multi-Robot Formation with the Cluster Space Representation

MAJOR FIELD: Systems & Control Engineering

DATE OF DEGREE: APRIL 2017

A cluster space control provides a simple concept for controlling and maintaining a multi-robot formation. Model uncertainty, velocity saturation, and model non-linearity are critical challenges in such multi-robot systems. Designing a robust and an adaptive controller is fueled by these challenges. In this thesis, the multi non-holonomic robot formation with a cluster space approach was studied, by deriving the multi non-holonomic robot cluster dynamic model. Then based on this model, adaptive sliding mode control (SMC) algorithms were developed to overcome the model's uncertainty and nonlinearity. These robust SMC algorithms are the normal SMC, the adaptive SMC, and the artificial fuzzy adaptive SMC. The proposed approaches are studied, based on their responses' speed and performance while the existence of an external disturbance. Also, this thesis studies the effect of

the robot velocity saturation and the limits on the cluster space. These limits are represented as diamond and elliptical models, then based on these models a model predictive controller (MPC) is used to minimize the saturation effects by tuning the formation controller gains. The proposed algorithms were validated through simulation and real experimentation and the results show considerable improvements.

ملخص الرسالة

الاسم الكامل: محمد طارق موسى نصر

عنوان الرسالة: التحكم بتشكيل مجموعة من الروبوتات بواسطة مبدأ التحكم الجمعي

التخصص: هندسة النظم

تاريخ الدرجة العلمية: 4/2017

يعتبر التحكم الجمعي من أبسط طرق التحكم في تشكيل مجموعة من الروبوتات. تعتبر مشكلة اللاتحديد واللاخطية في نماذج الروبوتات بالإضافة الى أن سرعة محدودة القيمة من التحديات المؤثرة في التحكم بمجموعة من الروبوتات مما يجعل تصميم متحكم فعال ومتأقلم مهم في مثل هذه الحالة. في هذه الأطروحة العلمية نتناول التحكم بتشكيل مجموعة من الروبوتات محددة الحركة بواسطة مبدأ التحكم الجمعي، بحيث يتم اشتقاق النموذج الرياضي الذي يمثل ديناميكية الحركة لها ثمبناء على هذا النموذج يتم اشتقاق وتصميم متحكم فعال ومتأقلم من نوع المجال المنزلق ثم تطويره بإضافة المنطق المشوش يعمل على تعديل قيم هذا المتحكم بما يتلائم من اللاتحديد والمؤثرات الخارجية للنظام. ثم تم فحص وتجريب هذه الطرق في التحكم بواسطة النمذجة والتجربة العملية، وتمت مقارنة النتائج ببعضها من حيث سرعة الاستجابة والكفاءة بوجود المؤثرات الخارجية. وفي هذه الأطروحة تم تصميم متحكم يعتمد على توقع استجابة النموذج الديناميكي من أجل حل اشكالية الحد الاعلى للسرعة وتأثيرها على حركة مجموعة من الروبوتات. جميع النظريات المطروحة تم اختبارها ببرامج المحاكاة ثم التجربة العملية

LIST OF ABBREVIATIONS

x	:	Robot displacement in horizontal axis in meters
y	:	Robot displacement in vertical axis in meters.
q	:	Robot heading angle
WMR	:	Wheeled mobile robot.
v	:	Robot heading velocity.
w	:	Robot rotation velocity.
d	:	Distance between the robot center of gravity and the midpoint on the line linking between the robot wheels.
ρ	:	Robot wheel radius.
C	:	Robot center of gravity.
Q_c	:	Cluster heading angle.
d_c	:	Distance between the two robot in two robot cluster.
x_c, y_c	:	Cluster center
c	:	Cluster states

\mathbf{r}	:	Robot space
\mathbf{J}	:	Jacobian matrix
\mathbf{q}_c	:	Distance between robot 1 and robot 2 in three robot cluster.
\mathbf{p}_c	:	Distance between robot 1 and robot 3 in three robot cluster.
β_c	:	Angle between q_c and p_c lines in the three robot cluster.
$\mathbf{M}(\mathbf{r})$:	Robot inertia matrix
$\mathbf{b}(\mathbf{r}, \dot{\mathbf{r}})$:	Combination of coriolis and centripetal and friction forces.
$\mathbf{g}(\mathbf{r})$:	Gravity forces.
$\boldsymbol{\tau}$:	Motor torque
λ	:	Lagrangian multiplier.
$\mathbf{A}(\mathbf{r})$:	Robot motion constraints matrix.
DOF	:	Degree of freedom
BAFC	:	Behavior and adaptive fuzzy control algorithm
P	:	Priority of the BFAC.
$\Delta\mathbf{K}_p$:	Fuzzy tuning propositional gain
$\Delta\mathbf{K}_d$:	Fuzzy tuning derivative gain

E	:	Robot error signal.
Q_d	:	Robot heading angle error.
k_{ij}	:	Controller constant gains.
E_c	:	Cluster error
\dot{C}_{cmd}	:	Cluster velocity command.
c_d	:	Desired cluster path
s	:	Sliding surface function
δ	:	Deviation error from sliding surface .
erf	:	Error function
w_r, w_l	:	Right and left wheels velocities

CHAPTER 1

INTRODUCTION

Cooperative control of multi-agent systems has attracted considerable research interest over the last decade, owing to the augmented capabilities that such systems offer during automation tasks. Some of these capabilities include increased coverage, speed, repeatability, precision, redundancy and strength, as well as the ability to withstand extreme conditions [1]. This allows multi-robot systems to be used in many automation applications, including sensor deployment, scouting, fire fighting, rescue and recovery and military applications, as well as environmental protection and surveys, such as oil spill disasters. One potential industrial application is seismic sensor deployment robots: a group of robot will deploy more than 70 thousand sensors daily, instead of deploying them manually as with the traditional method. As a research group we have a pending patent for this application.

In all of these applications simultaneous motion coordination and formation control is one of the key challenges when using multi-robot systems. Formation

control is crucial, especially when the sensors coverage and capabilities are limited. In such a case, formation allows each robot to focus its sensors towards a certain portion of the area of interest [2]. For example, a robot-scout benefits from a formation by directing the sensors of each robot towards achieving the maximum coverage area [3].

From the literature, there are three main control frameworks used in robot formation: leader-follower, null space, and cluster space. The well known leader-follower concept has been extensively studied to design control strategies for robot formation, where the follower robots have to follow a virtual position relative to the leader (for example see [4–6]). Recently, several related research issues have been investigated. For instance, optimizing the path planning in a leader-follower formation with obstacle avoidance and its suitability for real-time implementation has been presented in [7]. Model uncertainty was addressed in [8–10]. The null space approach is a task-based formation control concept, where each requirement is considered as a task. For example the spacing between the robot is considered to be a task; the center of the group is another task, and the robot target following is a separate task, and so on. See [11–14].

The third control method uses the cluster space approach, where the group of robots are considered as one entity, called a cluster. This cluster has its own dynamic states (called the cluster space). The cluster states are a function of the robots states (called the robot space). The control commands are calculated at the cluster level. Based on that, cluster commands are translated to robot space

commands by applying inverse kinematics and using the specific Jacobean matrix. Thus, each robot has its own command derived from the clusters command. Therefore, using the cluster space framework makes the control design simpler, as opposed to dealing with numerous robot entities as in the virtual-leader concept (see [15–19]). Many research questions were addressed in the literature; the obstacle avoidance problem was studied by [1, 16]; a behavioral intelligent controller was proposed by [20]; and the cluster space approach was applied to vessel control for a military purposes in [21–23], (these vessels have nonholonomic dynamics). However, designing a model-based controller for a group of nonholonomic robots is still an active field of research.

Nonholonomic robots are robots that have constraints on their velocities; for more see [24]. Most mobile robots are nonholonomic. For example, the two-wheel differential robot is considered to be a nonholonomic robot because it only moves toward the direction of its heading angle.

1.1 Preliminaries

1.1.1 Classification of multi robot control approaches

The existing literature can be classified into two categories based on command source

Centralised Centralized Controllers: information from all of the nodes is collected at a single end (or a central processor), the appropriate control input for

each node is computed and transmitted to the corresponding nodes.

Decentralized Distributed Controllers: rather than having a central processor, each node has its own implementation of the controller. A single node can exchange information with its neighboring nodes only.

On the other hand, we can classify the existing work based on the control concept into three classes:

Leader and follower control concept A virtual vehicle is constructed such that its trajectory converges with the reference trajectory of the follower. Position tracking control is designed for the follower to track the virtual vehicle.

Cluster space approach The cluster space approach considers the group of robots to be one entity and calculates the control commands in the cluster level so the group of robots can be conducted as one big robot. After that, these cluster commands are translated into robot space commands by applying inverse kinematics and using the specific Jacobean matrix. Thus, each robot has its own command which is derived from the cluster command.

Null space approach This approach proposes a task based controller for the group of robots. These tasks are the obstacle avoidance task, distribution on a certain format task, the target tracking task, and the last task is maintaining equal distance between each other. They are performed according to the desired priority. As a result, the highest priority task will have the highest control weight.

1.2 Literature Survey

According to table 1.1 the cluster space approach is still new and needs further research. Many challenges need more investigation with the cluster space approach, such as using non-linear robots, model uncertainty, the need for an adaptive control algorithm and intelligent controllers. In this section, a survey of these challenges will be presented, starting with the case for using the nonholonomic robot type in a multi-robot formation.

Nonholonomic robots are robots that have constraints on their velocities; for more see [24]. Most mobile robots are nonholonomic. For example, the two-wheel differential robot is considered to be a nonholonomic robot because it only moves toward the direction of its heading angle. Also this velocity constraint is considered a nonlinearity and it makes controlling such robots more challenging.

In the literature, the problem of multi-nonholonomic cluster formation control has been tackled by adding a fast inner control loop to change the robot heading angle toward the desired motion profile, while the outer controller handles the formation and tracking tasks [1,25]. However, having two control loops makes the system more complicated, and gives a generally slower time response. The outer controller always assumes that the robot is heading to the target (which is not always true) causing the tracking performance to decrease.

Moreover, the majority of the proposed controllers were velocity-based controllers; therefore, the acceleration is not considered as a state to be controlled, and this leads to neglecting the uncertainty in the robots mass and inertia. To

solve this issue a recent approach [23] proposed a dynamic-based controller for the cluster space approach, where the acceleration and dynamic model of the cluster space robots were considered in the controller design. In [23] a feedback linearization algorithm was proposed. However, the proposed approach assumed that the robots were holonomic robots.

Control of single nonholonomic robots has seen extensive research activity during the past few years. On the other hand, the cluster space control of multi-nonholonomic robots is still under investigation and, to our knowledge, a robust model-based control of such a system has not been addressed.

Behavior control methods are developed to tackle complex control problems that autonomous robots encounter in an unfamiliar real-world environment [26]. The behavior controller has a general set of constraints that allow robots to react in a certain domain [27]. Based on these constraints the robot will select the appropriate response, which is called a 'behavior', task or routine. A hierarchy of distributed behaviors was tackled in the literature to fulfill a given goal with different scenarios. In order to switch between these behaviors, a fuzzy logic technique was used since it does not need a precise model and it is based on logistic commands that make it suitable for representing the behavior selection criteria or constraints [26, 28, 29]. For example, the behavior-based control has been implemented for a soccer playing robot in [29] and was used in navigation and coordination control in [26, 28, 30, 31]. However, these methods were not applied to multi-robot clusters.

A behavioral fuzzy controller for null space was studied in [32]. However, to our knowledge, it has never been implemented in a cluster space concept. Also, in the literature, a classical PID controller was proposed with the behavior-based controller. Issues like disturbance effects, robot nonlinearity, and actuator saturation were not considered. The nonlinearity issue is important, especially when using a group of nonlinear robots. It is important to note that the actuator saturation issue can make the formation of a group of robots unstable. One of the recommended solutions to deal with these issues is the use of fuzzy adaptive controllers [33–36].

Still, the robots velocity saturation is a practical challenge that affects the formation control in a cluster space of differential drive robots. The usage of the Model Predictive Controller (MPC) is a preferred approach in the literature to overcome actuator saturation and model uncertainties where the cluster space is considered to be a highly non-linear and coupled system. Previous work tackled the issue of saturation with other formation concepts, for example, the leader-follower approach [37] proposed an MPC for the virtual leader approach. [38, 39] proposed an MPC framework with a neural network for compressing the robot communication packets. Also, the null space concept [13] proposed an intelligent method to deal with the saturation by a saturation management technique where the higher priority task will be served. However, this method used a one dimension saturation limits. The saturation limit is usually presented in one dimension [40] which is not an exact saturation representation of the differential drive robot

saturation, a recent work [41] presented the velocity saturation of the differential drive robot as a diamond shape domain. Based on that domain and using the proposed MPC a new approach is presented in this work to overcome the issue of velocity saturation.

Previous works applied cluster space with different types of robots, i.e. the unmanned vessel fleets [16], [21], Aerial robots [18] and wheeled land robots.

Also, in order to overcome the environmental obstacles and enemies, formation shape switching for the multi-robot was proposed. This method was proposed for the leader-follower approach (e.g. [42–44]). This may also be applied to the cluster concept.

Practical networking issues and their impacts were also studied such as limited bandwidth [38], time delay [45], packet loss, and maximum coverage area with wireless networks [46–48]. Considerable work has been done in these fields considering the leader-follower approach. However, there is a gap in the cluster space approach.

Disturbance and uncertainties are common challenges for the control designer; these issues are usually unmeasured and may cause instability. Adaptive and robust controllers, such as the fuzzy sliding mode controllers are used to solve such challenges [8]. Developing a hybrid intelligent approach to a multi-robot system is shown in [49]. Another work [9] tackles the uncertainties of the leader-follower approach.

Table 1.1: Literature survey

Research directions	Approaches		
	Leader follower	Cluster space	Null space
Singularity problems	-	[50]	-
Uncertainty issues	[8, 9]	-	[51]
Network issues (limited bandwidth, Time delay, Area coverage)	[38, 48]	-	[52]
Formation Shapes Switching and fault tolerance	[42, 44]	-	[53]
Actuator saturation	[41]	-	[13]
Model based controllers	[54]	[23]	[55]
Adaptive controllers	[8–10]	-	[32]
Under actuated robot	[56]	-	[11]
Obstacle avoidance	[57, 58]	[1, 16]	[59]

1.3 Problem statement

The literature survey shows that the cluster approach is still under investigation.

Consequently, many issues need to be investigated:

- The derivation of the dynamic model for a non-holonomic multi-robot was tackled in this thesis.
- Studying and developing a hybrid intelligent controller that makes the

cluster space approach more robust to disturbances and the systems non-linearity.

- Investigating the problem of actuator saturation that may lead to unstable cluster.
- Designing a Robust controller to tackle the uncertainties in the system.

1.4 Steps and control methodology

- Studying the dynamics of the cluster space with the non-holonomic robots group, starting from the robot space to find the cluster space dynamics. This can be implemented by extending the work in [23], in which the dynamics model of the cluster space is derived in a Lagrangian framework. An extension was done by developing an adaptive robust sliding mode controller for the formation control of the cluster.
- Extending the SMC into a fuzzy adaptive SMC. Adding the fuzzy adaptive term has improved the SMC performance, the response will be faster than the standard SMC and the adaptive SMC.
- Investigating the actuator saturation problem; From the literature, a graphical method has been used with a leader-follower to study the actuator saturation problem [41]. By extending this approach to the cluster space method, we ended up with a model predictive controller (MPC) that can deal with the saturation problems.

- A behavioral adaptive intelligent controller is proposed in order to improve the cluster space dynamics. This behavioral controller will give a higher priority to the formation task than the to target following task.

1.5 Significance of Research

The major contributions of this thesis can be summarized in the following points:

- Developing a behavior-based formation controller with the cluster space.
- Extending the dynamic model of the cluster of robots to a cluster of non-holonomic robots.
- Developing an SMC and an adaptive SMC for the cluster formation to overcome the nonlinearity, model uncertainty, and external disturbances effects.
- Improving the SMC by adding to it the fuzzy adaptive controller. As a result, the response of the fuzzy adaptive SMC is faster than the standard adaptive SMC and the SMC.
- Investigating the velocity saturation of the robots in the cluster space and developing an MPC controller to minimize its effect.

1.6 Experimental setup

Two Lego EV3 WMRs were used in the experiments to validate and implement the control strategy. The WMRs (see Figure 1.1) are equipped with a 32-bit, 48

Mhz ARM9 CPU with 16MB flash memory and 64MB RAM, Bluetooth and Wi-Fi transceivers, and two servo motors with encoders with 1 degrees of resolution. A PC interface with the SIMULINK program is also required to transmit the control signal by means of wi-fi protocol. SIMULINK has a powerful feature called External Mode. This feature is useful for on-line monitoring and tuning of the EV3 WMR's controller. A two-level control structure is used: a high-level and a low-level. The high-level controller is the sliding mode controller, operating in the central PC, which sends and receives command /data to and from the WMRs low-level controller. The low-level controller is a PID inner loop for controlling wheel speeds. Based on the proposed control algorithm, the central PC receives the location feedback from each robot; then the PC calculates the error and control signals and sends the velocity commands to each robot. The low-level controller on the robot receives the commands from the PC and relays these signals to the motors. Then the encoders provide measurements for the feedback. The actual time of a one-loop process depends on the robot sampling time (set to 25ms) plus the wi-fi delay time, which is dependent on the computer speed and network usage. The robot localization is achieved by using the encoders only. The WMRs use EV3 servo motors that have a gear reduction mechanism in order to increase the torque and decrease the maximum output speed. However, this gear mechanism has a backlash issue, which introduces a nonlinear behavior due to small gaps between the mating gear teeth. Once the servo motor changes its direction the backlash effect occurs, causing the servo to have a certain rotation

without being translated to actual wheel rotation. This issue can be mitigated by adding backlash compensation. Thus, when the servo motor changes its rotation direction a certain value is subtracted from the encoder reading.



Figure 1.1: Lego EV3 WMRs

1.7 Thesis organization

The rest of the thesis is as follows: chapter 2 is a preliminary, discussing the cluster space formation framework and the nonholonomic robot model used, and it proposes the nonholonomic cluster model. The third chapter proposes the behavioral kinematic controller. The fourth chapter proposes the sliding mode controller and gives the simulation results, as well as experimental results. Chapter 5 presents the potential of using fuzzy adaptive SMC and shows the experimental and simulation results. Chapter 6 discusses the velocity saturation issue and proposes a model based controller and then showing experimental and simulation results. Chapter 7 is the conclusion and future work. Finally, it gives the appendices of

the published journal papers from the thesis work.

CHAPTER 2

CLUSTER SPACE REPRESENTATION OF NON HOLONOMIC ROBOTS

2.1 Preliminaries

In this section fundamental definitions and mathematical models are presented; the Nonholonomic robot kinematic model is presented, followed by the mathematical representation of the cluster framework, after that the dynamical models for a single nonholonomic robot and the general cluster model with nonholonomic robots are presented.

2.1.1 Nonholonomic robot dynamics

We consider a wheeled mobile robot (WMR), with two driven wheels and a passive caster wheel, whose schematic model is shown in 2.1.

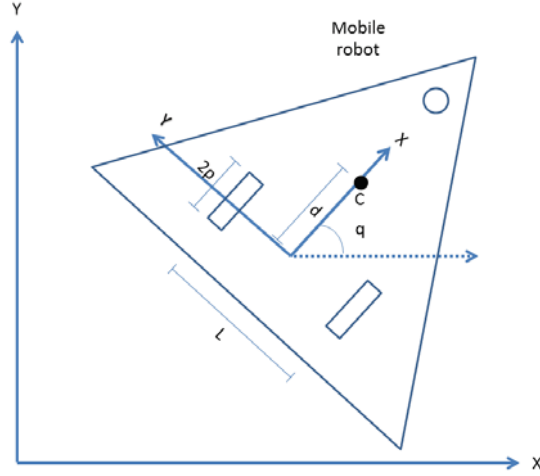


Figure 2.1: A WMR model, where q is the WMR's heading angle, point C is the robots center of gravity, d is the distance between the wheels line and C and ρ is the wheel radius

The state-space model of the considered kinematic vehicle with the associated nonholonomic constraints (rolling with no slipping) is given by equation 2.1, where v and w are the heading and rotational velocities variables:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} \cos(q) & 0 \\ \sin(q) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix}. \quad (2.1)$$

2.1.2 Overview of cluster space framework

In order to implement the cluster space for two robots, an appropriate set of cluster variables are chosen to represent the shape of the cluster. As shown in

Figure 2.2, the proposed cluster variables are $c = (x_c, y_c, Q_c, d_c, q_1, q_2)$ and the corresponding robot space are $r_1 = (x_1, y_1, q_1)$ and $r_2 = (x_2, y_2, q_2)$. The following equations 2.2 show the relation between the cluster space and the robot space variables, which can be presented as $c = f(r)$. Similar developments are found in [19] .

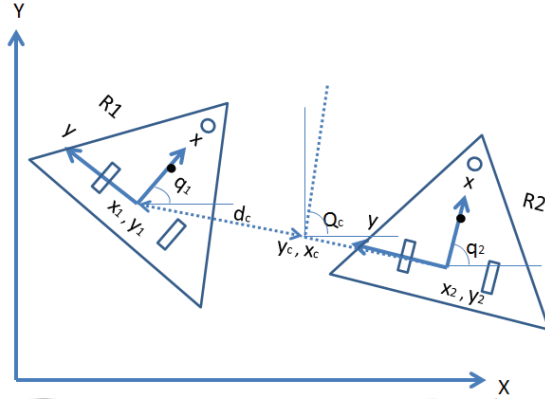


Figure 2.2: Cluster space and robot space variables

$$\begin{aligned}
 x_c &= \frac{x_1 + x_2}{2}, \\
 y_c &= \frac{y_1 + y_2}{2}, \\
 Q_c &= \tan_2^{-1}(y_1 - y_2, x_1 - x_2) + \frac{\pi}{2}, \\
 d_c &= \frac{1}{2} \sqrt{(y_1 - y_2)^2 + (x_1 - x_2)^2},
 \end{aligned} \tag{2.2}$$

Where (x_c, y_c) is the center of the cluster; Q_c is the cluster heading and d_c is the spacing between the robots from the cluster center. The corresponding Jacobian matrix is :

$$J = \begin{bmatrix} 0.5 & 0 & 0 & 0.5 & 0 & 0 \\ 0 & 0.5 & 0 & 0 & 0.5 & 0 \\ \frac{-(y_1-y_2)}{\epsilon_1} & \frac{(x_1-x_2)}{\epsilon_1} & 0 & \frac{(y_1-y_2)}{\epsilon_1} & \frac{-(x_1-x_2)}{\epsilon_1} & 0 \\ \frac{(x_1-x_2)}{\epsilon_2} & \frac{(y_1-y_2)}{\epsilon_2} & 0 & \frac{-(x_1-x_2)}{\epsilon_2} & \frac{-(y_1-y_2)}{\epsilon_2} & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

where $\epsilon_1 = (x_1 - x_2)^2 + (y_1 - y_2)^2$ and $\epsilon_2 = 2\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$, and $\dot{c} = J\dot{r}$

2.1.3 Three robot cluster

In this subsection a three-robot cluster was developed. The selected cluster variables are $(x_c, y_c, Q_c, \beta_c, q_c, p_c, q_1, q_2, q_3)$; see Figure (2.3) and for more details refer to [17].

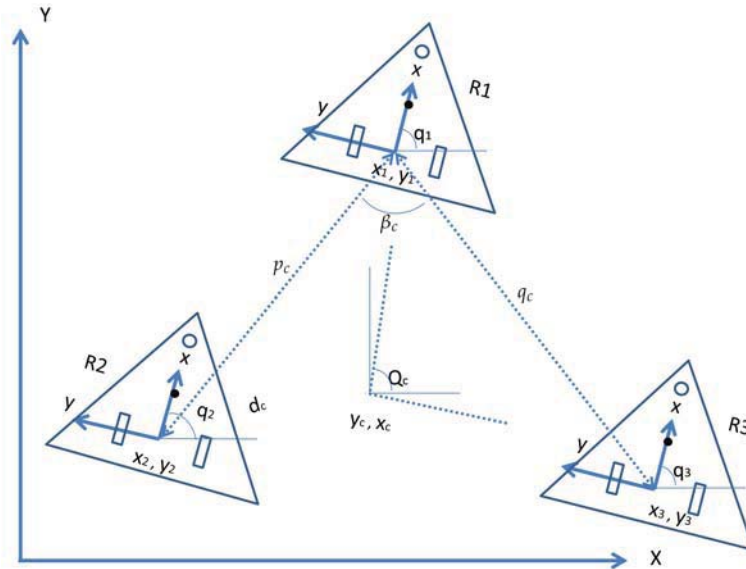


Figure 2.3: 3-robot system configuration

2.2 Cluster space dynamics of nonholonomic robots

In this section, the model representation of the cluster of nonholonomic robots will be derived. In particular, the coupling between the cluster states will be highlighted, owing to its importance in the overall control design.

2.2.1 Single nonholonomic robot modeling

The starting point will be the classical modeling of a single nonholonomic robot in equation 2.3, this model is a modified version of [60] model. Since the local axis is relocated in the modified version to be at the center between the wheels. Followed by a generalization to address the case of n robot and the cluster space model.

$$M(r)\ddot{r} + b(r, \dot{r}) + g(r) + \tau_d = B(r)\tau - A^T(r)\lambda. \quad (2.3)$$

According to Figure(2.1), $r = [x, y, q]^T$ and $M(r) \in \mathfrak{R}^{3 \times 3}$ denotes the positive definite symmetric inertia matrix; $b(r, \dot{r}) \in \mathfrak{R}^{1 \times 3}$ is a combination of Coriolis, centripetal and friction terms; $g(r) \in \mathfrak{R}^{3 \times 1}$ represents the gravitational forces; τ_d is the bounded unknown disturbances, and $\tau \in \mathfrak{R}^{2 \times 1}$ is the motors' torque vector. $A(r) \in \mathfrak{R}^{1 \times 3}$ represents the constraint matrix that is multiplied with the Lagrange multiplier $\lambda \in \mathfrak{R}^{1 \times 1}$, and the constraint equation is $A(r)\dot{r} = 0$, where

$$M(r) = \begin{bmatrix} m & 0 & -md \sin(q) \\ 0 & m & md \cos(q) \\ -md \sin(q) & md \cos(q) & I + md^2 \end{bmatrix},$$

$$b(r, \dot{r}) = \begin{bmatrix} 0 & 0 & md\dot{q}^2 \cos(q) \\ 0 & 0 & md\dot{q}^2 \sin(q) \\ 0 & 0 & 0 \end{bmatrix}, g(r) = 0, B(r) = \frac{1}{2\rho} \begin{bmatrix} \cos(q) & \cos(q) \\ \sin(q) & \sin(q) \\ L & -L \end{bmatrix}$$

$$\tau = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \text{ and } A^T(r) = \begin{bmatrix} -\sin(q) \\ \cos(q) \\ 0 \end{bmatrix}.$$

Let $S_r(r) \in \mathfrak{R}^{3 \times 2}$ be a full rank matrix such that $S_r^T(r) A^T(r) = 0$ such that

$$S_r(r) = \begin{bmatrix} \cos(q) & 0 \\ \sin(q) & 0 \\ 0 & 1 \end{bmatrix}.$$

And in the case of having $i \in \mathfrak{R} = 1, 2, \dots, n$ robots we have different models such that:

$$M_i(r_i)\ddot{r}_i + b_i(r_i, \dot{r}_i) + g_i(r_i) + \tau_d = B_i(r_i)\tau_i - A_i^T(r_i)\lambda_i.$$

And $S_{r_i}(r_i) \in \mathfrak{R}^{3 \times 2}$ be a full rank matrix such that $S_{r_i}^T(r_i)A_i^T(r_i) = 0$

2.2.2 Cluster space modeling

To select the cluster space states, several conditions should be considered. The cluster states should describe the function of the application, such as formation control. The number of cluster Degrees of Freedom (DOF) should be equal to

the number of the robot space DOF. The cluster dynamics can be calculated by transforming the robot space dynamics through the Jacobian matrix by $\dot{c} = J(r)\dot{r}$.

Cluster space dynamics Starting from the robot space dynamics with n robots, equation 2.4 is given as .

$$\overline{M}(r)\ddot{r} + \overline{b}(r, \dot{r}) + \overline{g}(r) + \tau_d = \overline{B}(r)\overline{\tau} - \overline{A}^T\overline{\lambda}, \quad (2.4)$$

where

$$\overline{M}(r) = \begin{bmatrix} M_1 & 0^{3 \times 3} & \dots & 0^{3 \times 3} \\ \bullet & M_2 & 0^{3 \times 3} & \vdots \\ \bullet & \bullet & \ddots & 0^{3 \times 3} \\ \bullet & \bullet & \bullet & M_n \end{bmatrix},$$

$$\overline{b}(r, \dot{r}) = \begin{bmatrix} b_1(r_1, \dot{r}_1) \\ \vdots \\ b_n(r_n, \dot{r}_n) \end{bmatrix},$$

$$\overline{B}(r) = \begin{bmatrix} B_1 & 0^{3 \times 2} & \dots & 0^{3 \times 2} \\ \bullet & B_2 & 0^{3 \times 2} & \vdots \\ \bullet & \bullet & \ddots & 0^{3 \times 2} \\ \bullet & \bullet & \bullet & B_n \end{bmatrix},$$

$$\bar{A}^T(r) = \begin{bmatrix} A_1^T & 0^{3 \times 1} & \dots & 0^{3 \times 1} \\ \bullet & A_2^T & 0^{3 \times 1} & \vdots \\ \bullet & \bullet & \ddots & 0^{3 \times 1} \\ \bullet & \bullet & \bullet & A_n^T \end{bmatrix},$$

$$\bar{\lambda} = [\lambda_1 \cdots \lambda_n]^T,$$

$$\bar{\tau} = [\tau_1 \cdots \tau_n].$$

Starting with robot space dynamics, the holonomic cluster dynamics in equation 2.5 were derived by [23]; based on that the nonholonomic robot cluster dynamics are found in equation 2.6, and the coupling between the cluster states can be represented in $\mu(c, \dot{c})$

$$\Lambda(c)\ddot{c} + \mu(c, \dot{c}) + p(c) + \tau_d = \beta_h(c)\bar{\tau}. \quad (2.5)$$

$$\Lambda(c)\ddot{c} + \mu(c, \dot{c}) + p(c) + \tau_d = \beta(c)\bar{\tau} - \alpha^T(c)\bar{\lambda}, \quad (2.6)$$

where

$$\Lambda(c) = J^{-T}(r)\bar{M}(r)J^{-1}(r),$$

$$\mu(c, \dot{c}) = J^{-T}(r)\bar{b}(r, \dot{r}) - \Lambda(c)\dot{J}(r, \dot{r})\dot{r},$$

$$p(c) = J^{-T}(r)g(r), \quad \beta(c) = J^{-T}(r)\bar{B}(r),$$

$$\alpha^T(c) = J^{-T}(r)\bar{A}^T(r).$$

and the constraint equation will be $\alpha^T(c)\dot{c} = 0$; let

$$S_c(c) = J \cdot \begin{bmatrix} S_{r1} & 0^{3 \times 2} & \dots & 0^{3 \times 2} \\ \bullet & S_{r2} & 0^{3 \times 2} & \vdots \\ \bullet & \bullet & \ddots & 0^{3 \times 2} \\ \bullet & \bullet & \bullet & S_{rn} \end{bmatrix},$$

such that $S_c^T(c)\alpha^T(c) = 0$ and accordingly it is possible to find an auxiliary $V(t)$ such that $\dot{c} = S_c(c)V(t)$ where $V(t) = [v_1, w_1, \dots, v_i, w_i, \dots, v_n, w_n]$ and v_i, w_i are the heading and the angular velocities of the nonholonomic robots.

Following the derivation we get equation 2.7

$$\ddot{c} = S_c(c)\dot{V}(t) + \dot{S}_c(c)V(t). \quad (2.7)$$

and by multiplying both sides of equation 2.7 with $S_c^T(c)$ we arrive at equation 2.8

$$H\dot{V} + E + \tau_d = \bar{\tau}, \quad (2.8)$$

where

$$H = (S_c^T(c)\beta(c))^{-1} S_c^T(c)\Lambda(c)S_c(c),$$

$$E = (S_c^T(c)\beta(c))^{-1} S_c^T(c) \left(\Lambda(c)\dot{S}_c(c)V(t) + \mu(c, \dot{c}) + P(c) \right),$$

$$\tau_d = Hf, \quad f \in \mathfrak{R}^{2 \cdot n \times 1}, \quad f = [f_{11}, f_{21}, f_{12}, f_{22}, \dots, f_{1 \cdot n}, f_{2 \cdot n}].$$

Now we have a reduced order dynamic equation for the cluster space with no Lagrange multiplier term.

CHAPTER 3

BEHAVIORAL INTELLIGENT KINEMATIC CONTROLLERS

Motivated by the gaps in the literature and the appealing nature of fuzzy approach, this chapter proposes a novel behavior and adaptive fuzzy control algorithm (BAFC) for cluster space control. The proposed study considers two competing behaviors, which are target following and formation shape preservation. The algorithm is simple, easy to implement, and its control approach performs tasks based on their importance. In this proposed novel BAFC, the position error and its rate of change are both considered as inputs to the fuzzy logic control tuning algorithm. This will in turn improve the controllers dynamical performance in an adaptive manner. The task-based control algorithms consider the robots actuator power allocated to one task at a time. This may help in solving the actuator saturation issue.

In this chapter, non-holonomic robots are considered owing to the fact that

the majority of wheeled robots are constrained in motion (wheels rotate without slipping). The implementation of the new control approach on Lego EV3 WMRs wheeled robot is also presented. The control strategy is implemented using Simulink. Real-time communication between robots and controller is established through a Wi-Fi link.

3.1 Control Architecture

As shown in Figure 3.1, the proposed cluster space controller, which consists of a closed loop controller with an adaptive fuzzy tuner that changes the controller parameters.

This scheme measures the robots states and converts it into cluster space states. The conversion is implemented by comparing cluster position and velocities with the desired trajectory values and outputting cluster velocities. These output cluster velocities are then translated into commands and sent to the robots. The following steps show the control procedure;

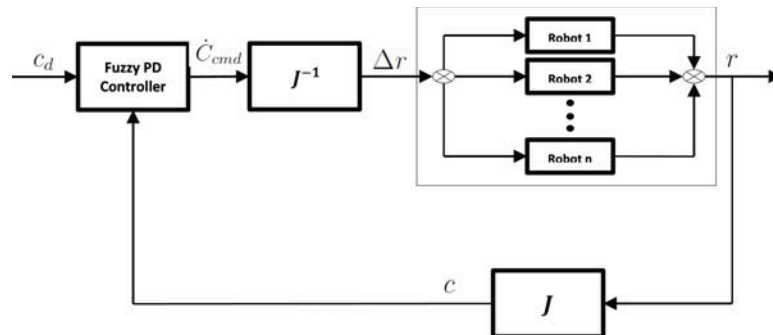


Figure 3.1: Cluster space control architecture where Δr is the robot commands and J is the Jacobian Matrix

Step 1: Calculating the error in the cluster variables using equation 3.1

$$E_c = \begin{bmatrix} X_{cd} - X_c \\ Y_{cd} - Y_c \\ Q_{cd} - Q_c \\ d_{cd} - d_c \end{bmatrix} \quad (3.1)$$

Step 2: differentiate the error using equation 3.2

$$\dot{E}_c = \frac{dE_c}{dt} \quad (3.2)$$

Where E_c and \dot{E}_c are (4×1) vectors,

Step 3: Adapt the values of K_p diagonal matrix (4×4) , and K_d diagonal matrix (4×4) by implementing the fuzzy functions as given in equations 3.3 and 3.4:

$$K_p = \bar{K}_p + \Delta K_p \left[E_c, \dot{E}_c \right]_{fuzzy} \quad (3.3)$$

$$K_d = \bar{K}_d + \Delta K_d \left[E_c, \dot{E}_c \right]_{fuzzy} \quad (3.4)$$

Where K_p and K_d are the proportional and integral constant controller gains and

$$\Delta K_p \left[E_c, \dot{E}_c \right]_{fuzzy} = [\Delta K_{pXc}, \Delta K_{pYc}, \Delta K_{pQc}, \Delta K_{pdc}],$$

$$\Delta K_d \left[E_c, \dot{E}_c \right]_{fuzzy} = [\Delta K_{dXc}, \Delta K_{dYc}, \Delta K_{dQc}, \Delta K_{ddc}],$$

$$K_p = [K_{pXc}, K_{pYc}, K_{pQc}, K_{pdc}],$$

$$K_d = [K_{dXc}, K_{dYc}, K_{dQc}, K_{ddc}],$$

$$\bar{K}_p = [\bar{K}_{pXc}, \bar{K}_{pYc}, \bar{K}_{pQc}, \bar{K}_{pdc}],$$

$$\bar{K}_d = [\bar{K}_{dXc}, \bar{K}_{dYc}, \bar{K}_{dQc}, \bar{K}_{ddc}].$$

Step 4: applying the Cluster fuzzy PD controller commands using 3.5

$$\dot{C}_{cmd} = K_p E_c + K_d \dot{E}_c \quad (3.5)$$

Where $\dot{C}_{cmd} = [\Delta X_c, \Delta Y_c, \Delta Q_c, \Delta d_c]^T$. Note that ΔQ_c should be in the range of $[-\pi, \pi]$.

Step 5: The cluster controller commands are translated into robot velocities by calculating the velocity inverse kinematics using the Jacobian matrix as in equations 3.6: $\bar{\Delta r} = J^{-1} \Delta c$

$$\begin{aligned} \Delta X_1 &= \Delta X_c + \Delta d \cos(Q_c - \frac{\pi}{2}) - d_c \Delta Q_c \sin(Q_c - \frac{\pi}{2}) \\ \Delta Y_1 &= \Delta Y_c + \Delta d \sin(Q_c - \frac{\pi}{2}) + d_c \Delta Q_c \cos(Q_c - \frac{\pi}{2}) \\ \Delta X_2 &= \Delta X_c - \Delta d \cos(Q_c - \frac{\pi}{2}) - d_c \Delta Q_c \sin(Q_c - \frac{\pi}{2}) \\ \Delta Y_2 &= \Delta Y_c - \Delta d \sin(Q_c - \frac{\pi}{2}) + d_c \Delta Q_c \cos(Q_c - \frac{\pi}{2}) \end{aligned} \quad (3.6)$$

Step 6: The Low level controller (WMRs controller):

$$\begin{aligned}
 E_{i1} &= \Delta X_i, \\
 E_{i2} &= \Delta Y_i, \\
 E_{i3} &= Q_c - Q_{id}, \\
 E_{i4} &= Q_{id} - q_i.
 \end{aligned} \tag{3.7}$$

Where $i=1,2$ indicates the robot index.

$$Q_{id} = \tan^{-1}\left(\frac{E_{i2}}{E_{i1}}\right) \tag{3.8}$$

The robot commands can be calculated using equation 3.9

$$\begin{aligned}
 u_{i1} &= \sqrt{E_{i1}^2 + E_{i2}^2} \cos(E_{i4}), \\
 u_{i2} &= k_{i1}E_{i3} + k_{i2}E_{i4}.
 \end{aligned} \tag{3.9}$$

Where $0 < k_i$, and u_{i1} is the head speed and u_{i2} is the orientation speed.

3.2 The Fuzzy controller design:

As described in Figure 3.2, the body of the fuzzy controllers consists of

1. Input fuzzification (crisp-to-fuzzy conversion),
2. Fuzzy rule base (linguistic knowledge base),
3. Inference engine and Output defuzzification (fuzzy-to-crisp conversion).

The fuzzy inference engine simulates the fuzzy rules using the input variables.

In order to achieve the foregoing objectives, two main methods are generally used: Mamdani’s method and Sugeno method. Mamdani is the first known fuzzy inference system, which consists of fuzzification, rule evaluation, aggregation of the rule outputs, and defuzzification. The main difference between Mamdani-method and Sugeno-method is how the crisp output is generated from the fuzzy inputs. While Mamdani uses the defuzzification of a fuzzy output, Sugeno uses a weighted average function to get the crisp output.

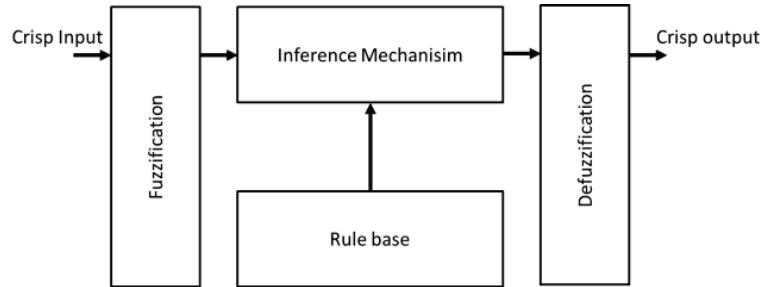


Figure 3.2: Fuzzy control parts

In this work, Mamdani’s method has been adopted, which is the commonly used inference engine. The proposed fuzzy approach starts with applying the fuzzyfication step to get the fuzzy values from the crisp inputs. These fuzzy values are represented in the fuzzy membership functions (see Figure 3.3 and 3.4). Next, the fuzzy rules are simulated on the fuzzy inputs as shown in Tables 3.1, 3.2 and 3.3, such that N (Negative), P (Positive), S (Small), M (Medium) and B (Big). In this case, the fuzzy inputs are the error terms, derivatives of the errors on the cluster space and the priority value P . The fuzzy outputs are the controller tuning parameters ΔK_p and ΔK_d [61].

Table 3.1: The fuzzy rules of ΔK_p provided that P is low

ΔK_p		\vec{E}_c						
		NB	NM	NS	Z	PS	PM	PB
E_c	NB	PB	PB	PM	PM	PS	Z	Z
	NM	PB	PB	PM	PS	PS	Z	NS
	NS	PM	PM	PM	PS	Z	NS	NS
	Z	PM	PN	PS	Z	NS	NM	NM
	PS	PS	PS	Z	NS	NS	NM	NM
	PM	PS	Z	NS	NM	NM	NM	NB
	PB	Z	Z	NM	NM	NM	NB	NB

Table 3.2: The fuzzy rules of ΔK_d Provided that P is low

ΔK_d		\vec{E}_c						
		NB	NM	NS	Z	PS	PM	PB
E_c	NB	PS	NS	NB	NB	NB	NM	PS
	NM	PS	NS	NB	NM	NM	NS	Z
	NS	Z	NS	NM	NM	NS	NS	Z
	Z	Z	NS	NS	NS	NS	NS	Z
	PS	PB	PS	PS	PS	PS	PS	PB
	PM	PB	PM	PM	PM	PS	PS	PB
	PB	PB	PM	PM	PM	PS	PS	PB

Table 3.3: The Fuzzy Rules of P

$\Delta K_p, \Delta K_d$	P	
	Low	High
The fuzzy values	Rules form tables 3.1 and 3.2	NB

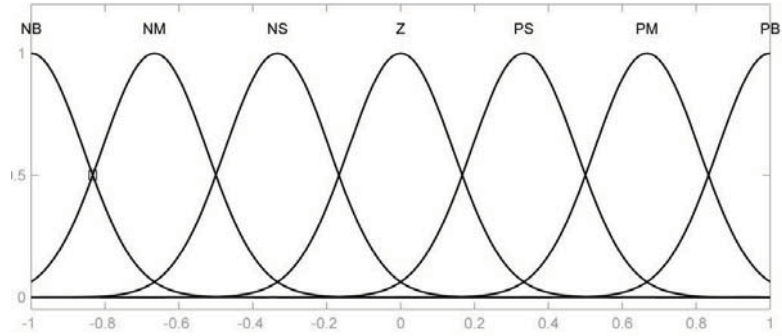


Figure 3.3: The member ship function of normalized $E, \dot{E}, \Delta K_p, \Delta K_d$

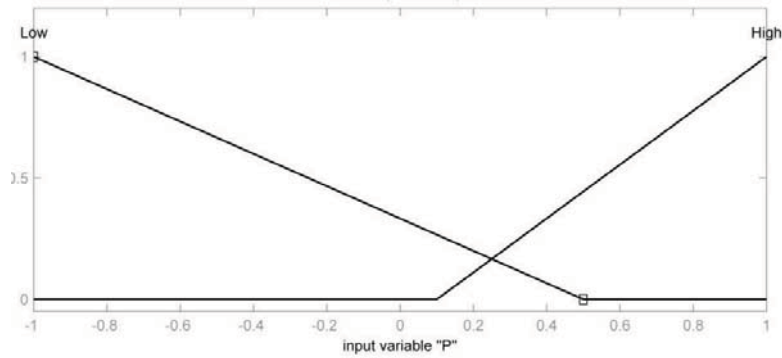


Figure 3.4: The Membership function for the Priority input P

3.3 BAFC structure

Figure 3.5 shows the BAFC graphical structure for a two robot-cluster presented in equations (3.3 and 3.4). The priority is achieved by using the max function. The higher control gain from the higher priority states will be passed. Therefore, if the output of the max is high then the fuzzy model will output a low control gain. And if no high gain is applied to the higher priority states, then the max function will return low signal. By so doing, the fuzzy model will give this state the priority to adapt its errors. The small triangles are the scaling constants that may be selected by trial and error or by applying an evolutionary approach such as Genetic Algorithm.

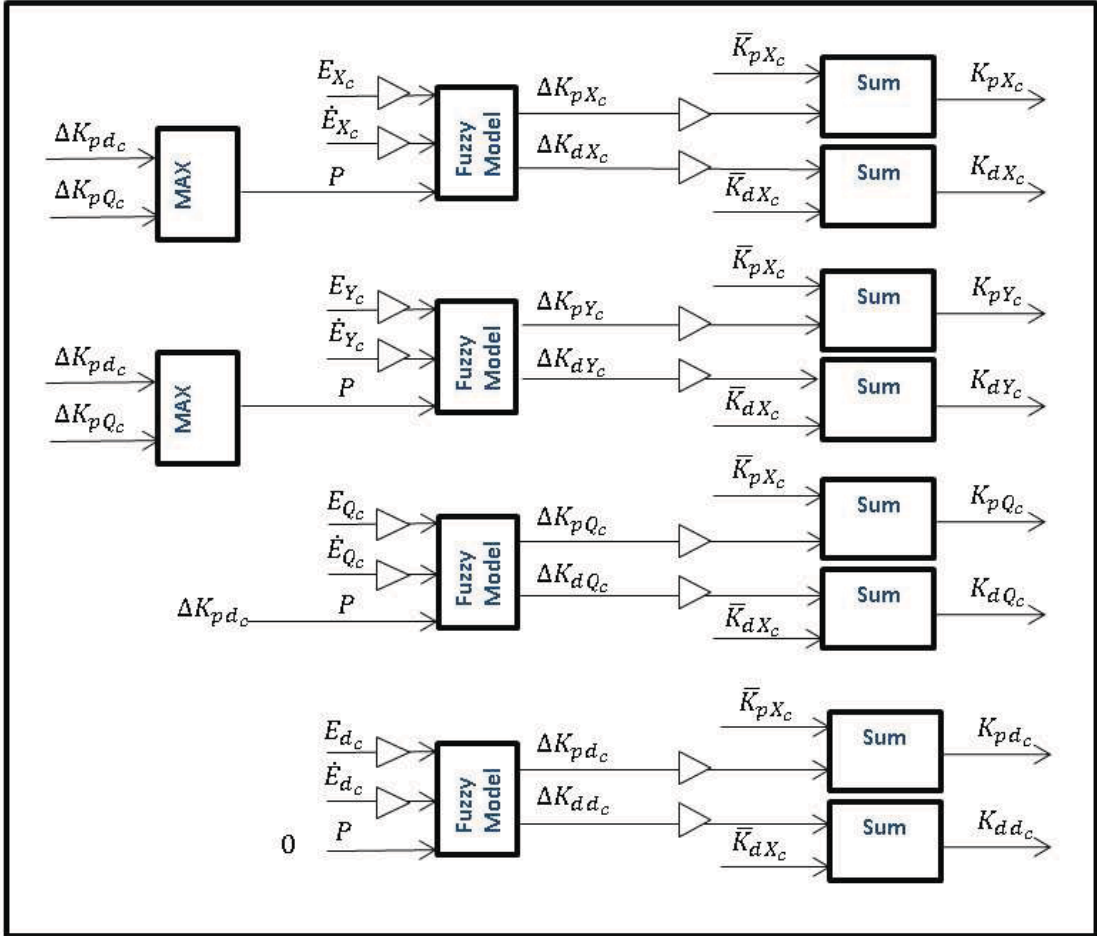


Figure 3.5: The BAFC structure for a two-robot cluster

Remark 1 1 *PD controller is used in this case as an example of the classical controllers. classical controller, especially PID type, can also be selected.*

Remark 2 2 *The proposed fuzzy model is a hybrid adaptive system. This means that the fuzzy model is simulated in a discrete time, while the robots dynamic model is simulated in the continuous time. Therefore, the fuzzy model should be slower than the robots model during the simulation.*

3.4 Stability Proof

For the stability test Lyapunov theory is used as follows: Let

$$E_c = C_d - C. \quad (3.10)$$

$$\dot{E}_c = \dot{C}_d - \dot{C}. \quad (3.11)$$

and

$$C_{cmd} = \int_0^t \left(K_p E_c + K_d \dot{E}_c d\tau \right) \quad (3.12)$$

Theorem Consider the mobile non-holonomic system (1). If the control command defined in 3.12 is applied to the mobile robot then the position and the velocity tracking errors in equations (3.10, 3.11) converge to zero.

Proof

Starting from equations (8)-(10), the cluster states error and its derivative E_c, \dot{E}_c are defined as in as in equations 3.10,3.11 and 3.12.

$\dot{C}_{cmd} = K_p E_c + K_d \dot{E}_c$ where $\bar{K}_p, \bar{K}_d > 0$ and $\left[E_c, \dot{E}_c \right]_{fuzzy} \leq \bar{K}_p, \bar{K}_d$ Substituting \dot{E}_c from equation 3.12 into equation 3.5 leads to

$$\dot{E}_c = \dot{C}_d - K_p (C_d - C) K_d (\dot{C}_d - \dot{C}) \quad (3.13)$$

$$\dot{E}_c = \frac{\dot{C}_d - K_p E_c}{1 + K_d} \quad (3.14)$$

$$\dot{L} = E_c^T \dot{E}_c,$$

And $\dot{c} = K_p E_c + K_d \dot{E}_c$. Then by substitution we arrive at:

$$\dot{E}_c = (\dot{c}_d - K_p E_c - K_d \dot{E}_c)$$

$$\text{Then } \dot{L} = (E_c^T \dot{c}_d - E_c^T (-K_p E_c - K_d \dot{E}_c)).$$

Now the equilibrium point is , so by considering the following Lyapunov function candidate

$$V = \frac{1}{2} E_c^T E_c$$

With $V(\vec{0}) = 0$, Computing the derivative dV/dt

$$\frac{dV}{dt} = E_c^T \dot{E}_c = \frac{E_c^T \dot{C}_c - E_c^T K_p E_c}{1 + K_d}$$

According to Lyapunov theorem, the system is stable if $V(\vec{0}) = 0$, $dV(\vec{0})/dt = 0$ and $dV/dt < 0$ Since $-E_c^T K_p E_c < -\lambda_{min}(K_p) \|E_c\|^2$ and $1 + K_p > 0$ where

$\lambda_{min}(K_p)$ is the minimum eigenvalues of the controller gain K_p , then $dV/dt < 0$ is true when equation 3.15 is satisfied, and accordingly the system is stable.

$$\|E_c^T\| < \frac{\|\dot{C}_d\|}{\lambda(K_p)} \quad (3.15)$$

Remark 3 *3 The above proof is for general PD control gains. When using fuzzy logic tuning, the PD controller becomes nonlinear. A necessary and sufficient condition for stability is to always verify that $K_p > 0$ and $(I + K_d) > 0$. However, in practice, actuator saturation or heterogeneous characteristics of the robots may lead to instability. The first issue is well known in the literature. The latter is due to the coupling created within the cluster between the robots and the inability of some robots to keep up with the cluster. This issue will be investigated in a future work....*

3.5 Simulation Results

3.5.1 Two-robot simulation

In this case, disturbance was added to the second robot between a time frame of 1 – 3 sec. Also, there is an existence of an initial condition error. Therefore, the controller should overcome two challenges: the initial condition and the disturbances. And also the controller’s objective is to give more priority to the formation shape than to the target-following task, in addition to the adaptability of the controller gain based on the changes in the states errors and errors velocity.

As seen in Figure 3.6, the BAFCs the response of the shape states (d_c and Q_c) are better than the classical controller.

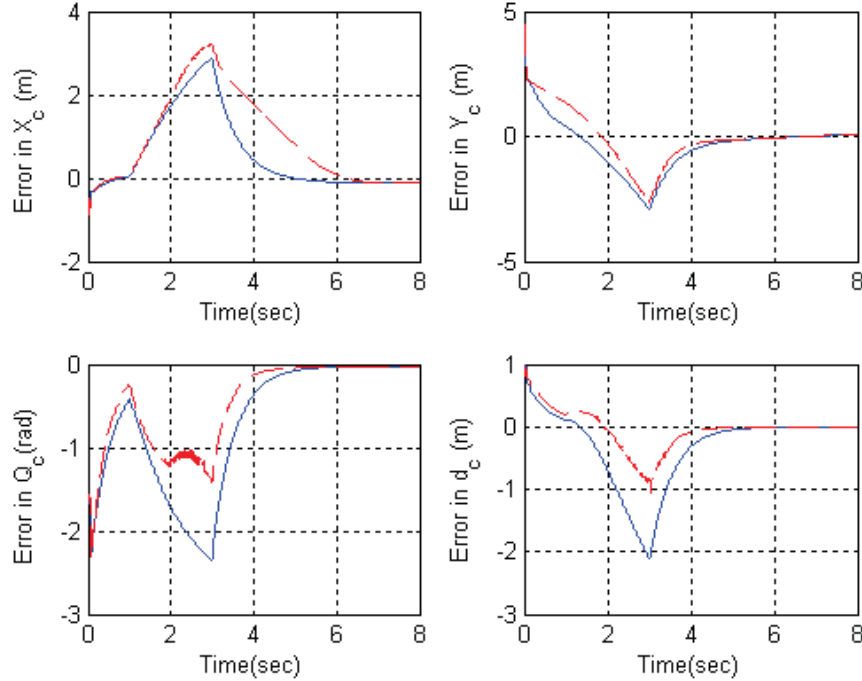


Figure 3.6: A comparison between the classical PD controller and BAFC. The solid blue lines are the system response with classical PD controller, and the dashed red lines are BAFC responses.

3.5.2 Three-robot simulation

The dynamics of three-robot cluster are presented in [17]. The BAFC structure is similar to the one with two robots but with more fuzzy models. In this case, disturbances were added to both robot 2 and robot 3 between 4-8 sec. And also, there is existence of an initial condition error. Therefore, the controller should be able to overcome two challenges: the initial condition error and the disturbances. The controller's objective is to give priority to formation shape over target follow-

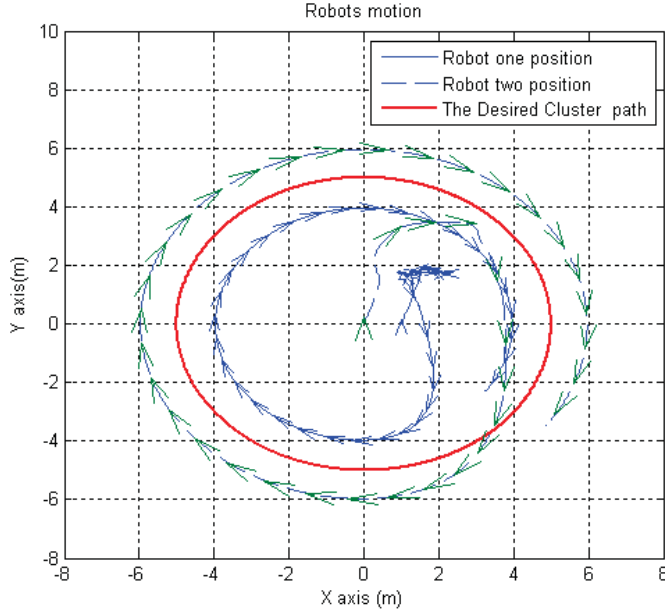


Figure 3.7: Robots motion with Fuzzy adaptive cluster controller

ing. In addition, BAFC adapts the controller gain based on the changes in the states errors and errors in velocity. Figures in 3.8 show the comparison between the classical PD controller and the BAFC. The results show better performance in q_c, p_c states, as depicted in Figure 3.8a, Figure 3.8b). These improvements are apparent in the transient response even with both the initial condition and the disturbance challenges. q_c, p_c are considered as the shape states and they are given the highest priority among the system states followed by β_c, Q_c, X_c and Y_c sequentially. As a result, q_c, p_c should have better responses with BAFC than the classical controller. Other potential advantages of BAFC over the classical approach are the actuators energy consumption and the max control values. As shown in Figure 3.9, the max absolute value of the control signal with the classical controller is 266 and with BAFC is 115. This means that BAFC is better in dealing with actuator saturation than the classical controller, and this will in

turn minimize the effects of actuators saturation problem. Therefore, BAFC requires a smaller actuator to be used in the Robots. This observation is arguably correct because BAFC controller allocates the energy to one objective at a time rather than to two conflicting objectives. This conflict may result to increase of the control energy and may cause instability.

3.6 Experimental setup

The Lego EV3 WMRs are used in the experiments. Those WMRs (see Figure 1.1) have a 32-bit, For more details about the experimental setup refer to section 1.6.

3.6.1 Experiment test

Two WMRs move from initial positions $[x, y, q]$ for the first robot: $[0.2, 0, \pi/2]$ and $[0, 0, \pi/2]$ for the second robot. The desired path is $Q_c = \pi/2, x_c = 0.5, y_c = 0.1t, d_c = 0.3$ in the cluster space that is equal to $x_1 = 0.8, y_1 = 0.1t, q_1 = \pi/2, x_2 = 0.2, y_2 = 0.1t, q_2 = \pi/2$ in the robot space, where t is the time, (see 3.10). In order to show the disturbance effects on each controller a software disturbance is added to Robot 2. This disturbance will hold Robot 2 for 90 sec with no movement see Figure 3.10 (a-c). The comparison now is based on how the other robot (Robot 1) will behave using both approaches (classical PD controller and the proposed BAFC).

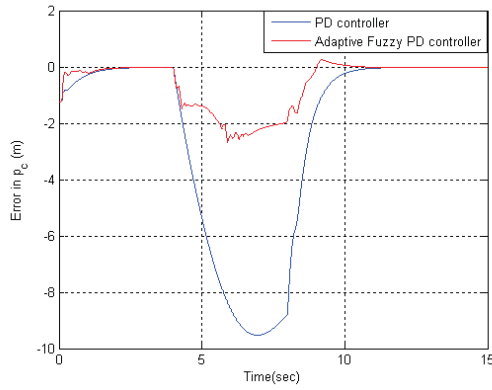
Comparing the standard PD controller response (see Figure 3.11) with the

fuzzy PD controller (see Figure 3.12), the BAFC gives more priority to the shape than the standard PD. Figure 3.13 shows the improved shape dynamics (the cluster angle and the distance between the robots) with the proposed BAFC controller when compared with the standard one. The Odometer uncertainties cause an accumulated error which can be lessened by calibrating the odometry equations [62,63]. Also, adding a sensor-like compass can greatly reduce this error [64].

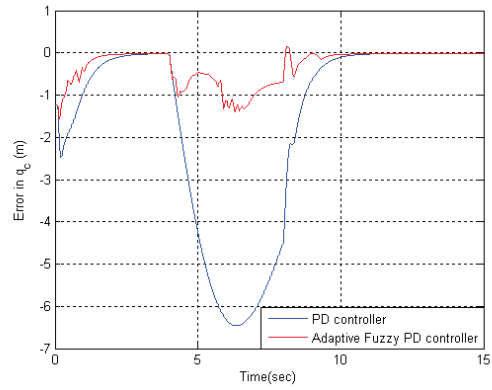
3.7 Chapter summary

Cluster space concept for controlling multi-robot systems is useful in simplifying the formation problem. In this work, an adaptive fuzzy controller is designed to improve the dynamics of the cluster space controllers. The cluster space dynamics were divided into two main groups based on their tasks or behaviors: the formation shape states and the target following states. Therefore, the BAFC gives more priority to the formation shape states than the target following states. BAFC adapts the controller gain based on the states errors and error in velocities. Simulations and experimental results show that the proposed behavioral adaptive technique has a significant potential. Formation dynamics are improved in addition to having lower actuator input in the robot cluster, which helps with actuator saturation issues. Using BAFC, the control designer has the flexibility to select the states priority and design an error-based adaptive controller. There are many extension to this work while keeping its easy implementation features. For instance, future work may tackle clusters with larger number of heterogeneous

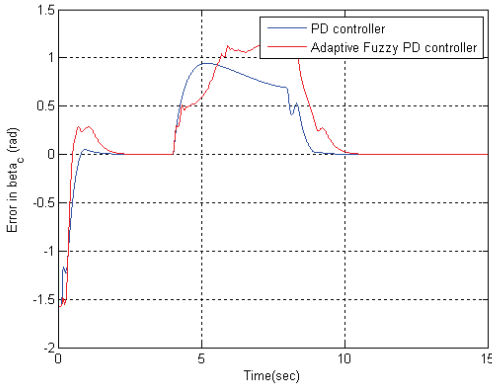
robots and apply intelligent methods to overcome the singularities in the cluster dynamics in addition to addressing the behavior-based obstacle avoidance problem. Fault tolerant cluster control is another area where study of the effect of faults and how to guarantee the performance of the cluster. In this area, division of the cluster to many sub-clusters could be sought. Effect of actuator could be formally addressed and quantified.



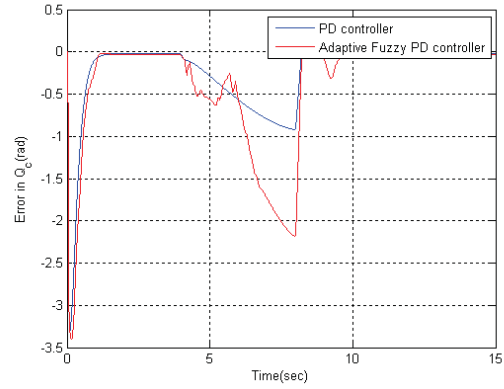
(a) comparison between the BAFC and the standard PD controller with p_c state



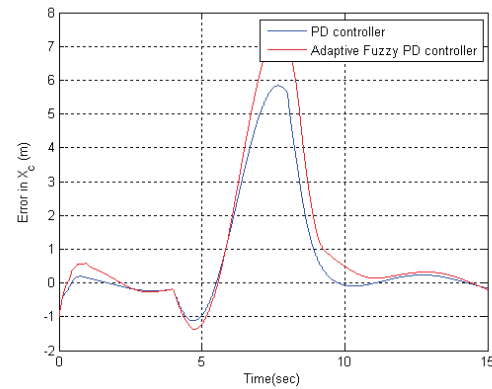
(b) comparison between the BAFC and the standard PD controller with q_c state



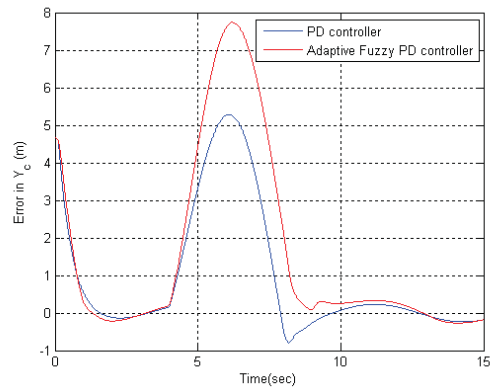
(c) comparison between the BAFC and the standard PD controller with β_c state



(d) comparison between the BAFC and the standard PD controller with Q_c state



(e) : comparison between the BAFC and the standard PD controller with X_c state



(f) comparison between the BAFC and the standard PD controller with Y_c state

Figure 3.8: Simulation results for cluster space control of a 3-robot system

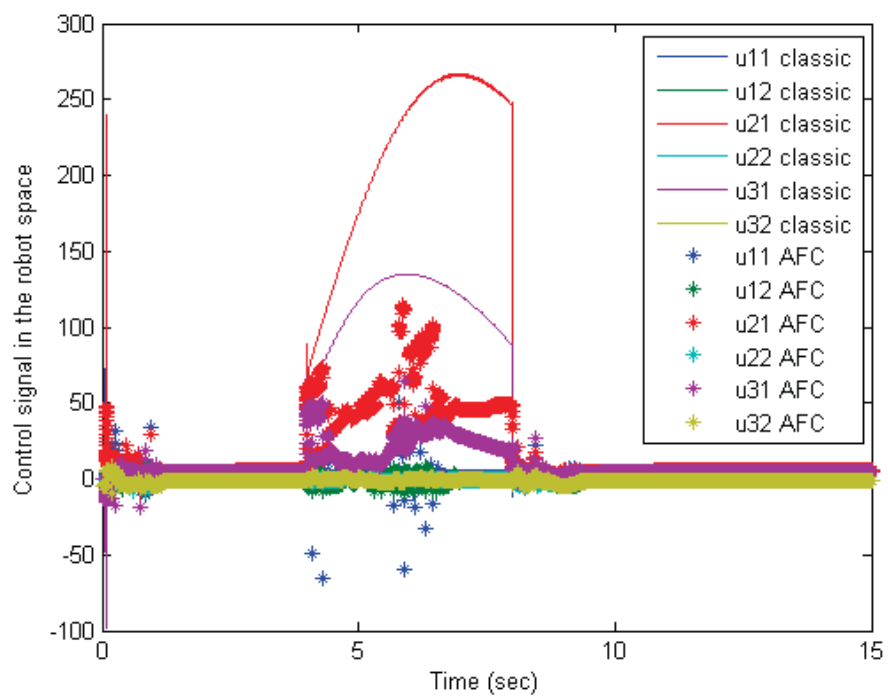
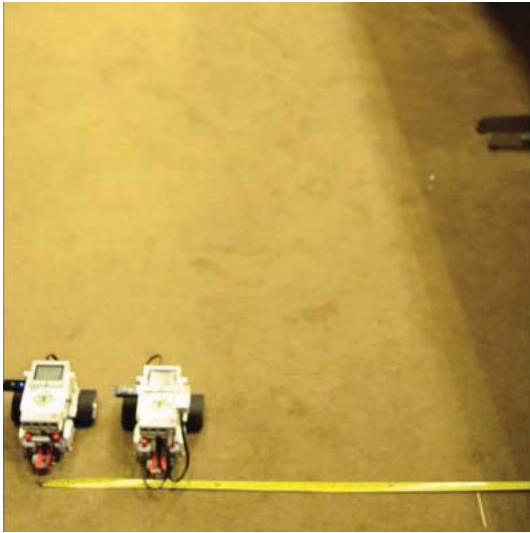
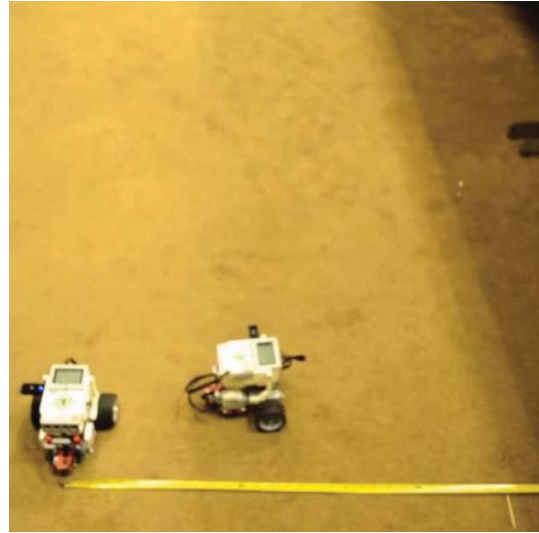


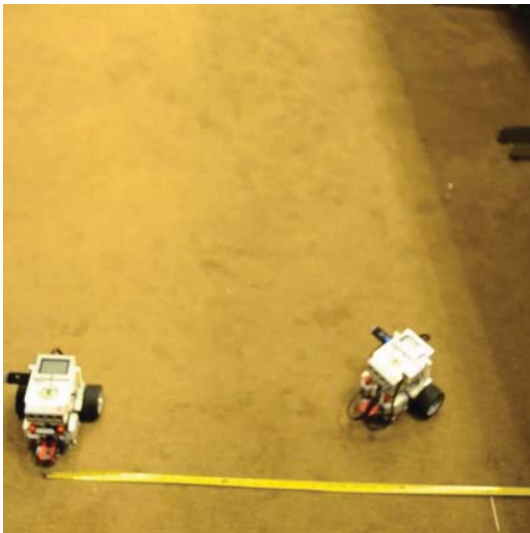
Figure 3.9: comparison between the control signal from the classical controller and the BAFC.



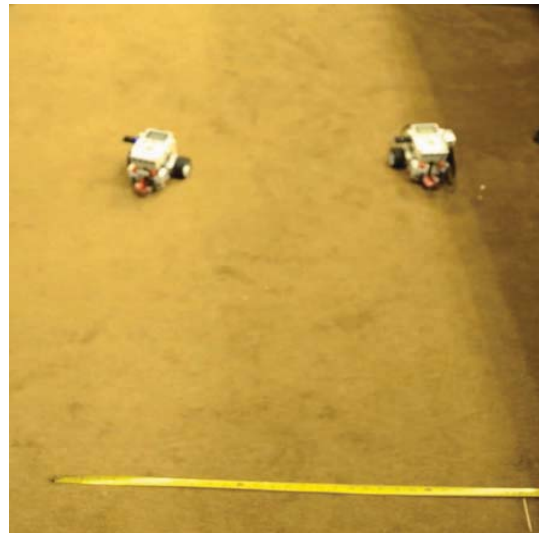
(a) Initial Position



(b) Robot 1 start moving While Robot2 is forced to hold



(c) Robot 1 is waiting and keeping the formation



(d) Robot 2 is working now and the group start moving to the target position

Figure 3.10: experiment with two WMRs

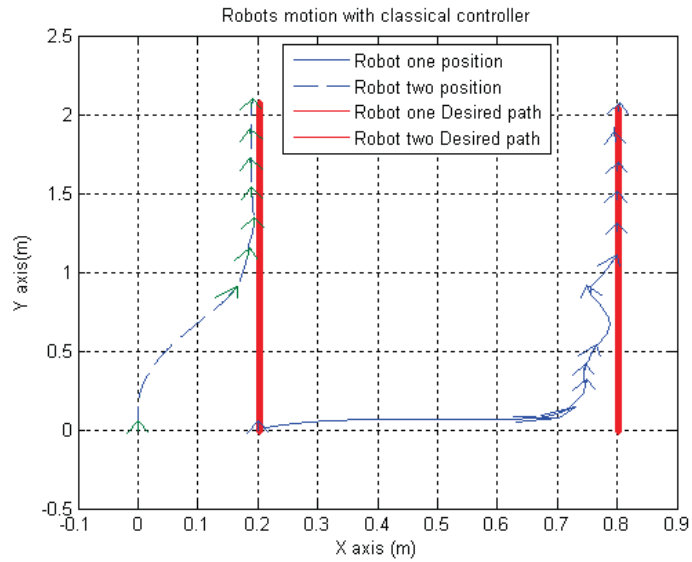


Figure 3.11: The Experimental Robot moving profile with standard PD controller

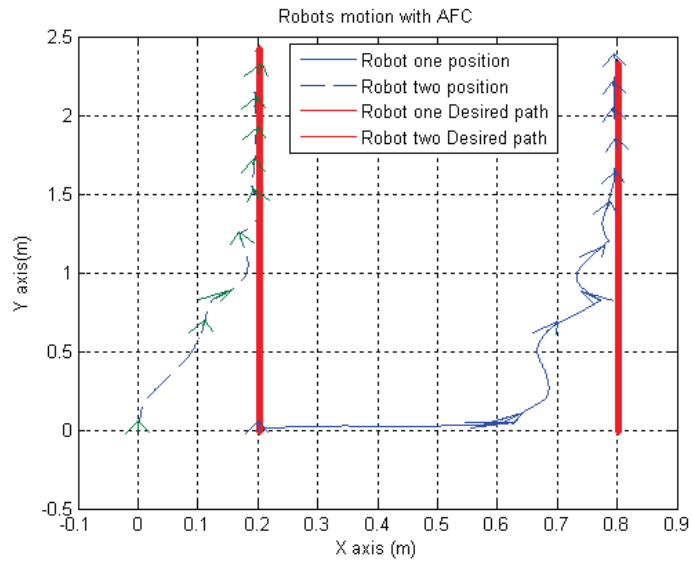


Figure 3.12: Experimental Robots moving profile with Fuzzy PD controller

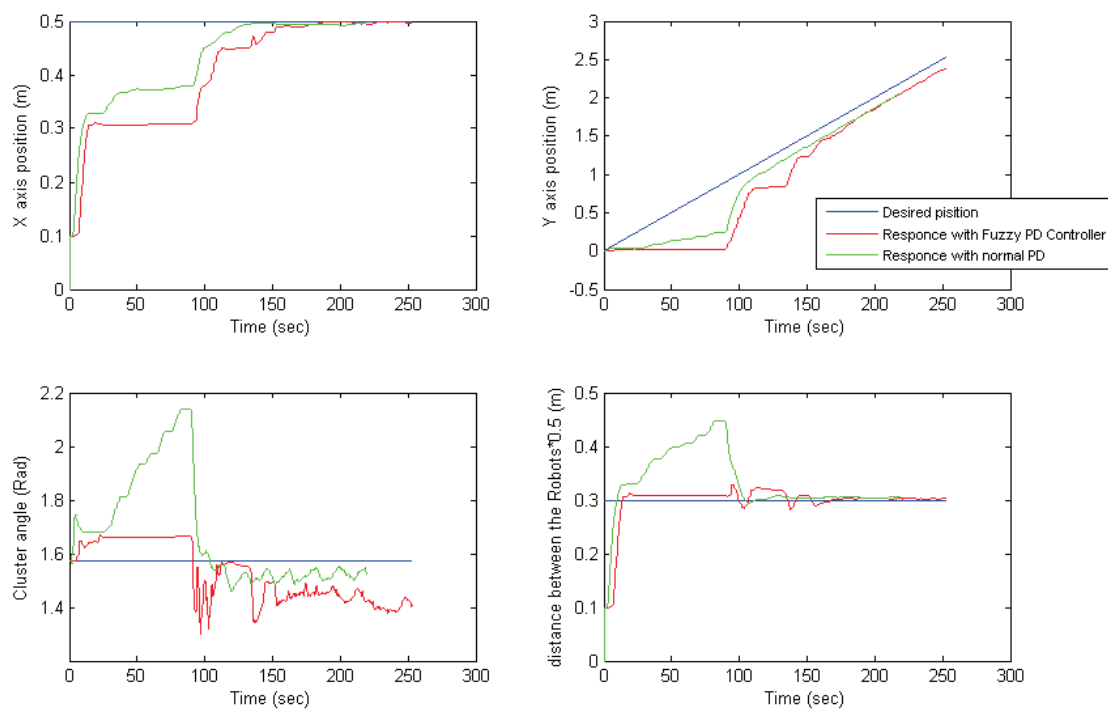


Figure 3.13: Experimental results

CHAPTER 4

SLIDING MODE CONTROLLER FOR NON-HOLONOMIC MULTI-ROBOT CLUSTER

A sliding-mode controller is a nonlinear robust model-based controller, where the system dynamics are forced to stay on a stable surface. This surface is a function of the system states; to guarantee reaching the sliding surface and to address the uncertainty, a robustifying term is added to the controller algorithm. When designing a sliding mode controller for the nonholonomic robot motion, constraints should be considered; hence selecting the sliding surface is not a trivial problem. In [65] a sliding mode controller was developed for a nonholonomic robot in the polar coordinates. The use of polar coordinates can simplify the sliding surface

selection; however this controller has some singularity issues around the origin and adds constraints on the motion postures and velocities. In [66] these constraints were eliminated. However, the proposed controller is still unstable around the origin and it needs to transform the robot coordination to a polar coordination, which is not commonly used in robotics. In order to tackle the singularity issue of the polar coordinate, [67] proposed a sliding mode controller over the Cartesian coordinate. However, because of the sliding surface constraints, the control input has limitations on the mobile robots movement. Recently, a modified version of the last controller was proposed by [68] where an approaching angle sliding surface was proposed to address the control constraints.

In this chapter, the latest adaptive sliding mode controller [69, 70] is extended to a formation control of a group of nonholonomic robots in a cluster space. The proposed adaptive robust controller addresses the issue of model uncertainties, as well as the system's nonlinear dynamics. Those issues are common in real-life applications where there are constraints on the robots motion that is transformed into a nonlinear behavior, while the model parameters either vary in time or are uncertain. The chapter begins with presenting the cluster model of nonholonomic robots [23] and provides the necessary changes to include the nonholonomic case. The chapter reports the experimental implementation of the proposed scheme on a real ground robot connected to a central controller using a Wi-Fi network.

4.1 Controller design

In this section the adaptive SMC is presented: The error signal calculation, then the sliding surface selection, followed by the adaptive SMC and the controller diagram. This development is done for a two-robot cluster as an example, but the same procedure can be done for a cluster with any number of robots.

4.1.1 The cluster profile errors

The formation error in the cluster spaces is found as follows; Starting with defining $c = (Q_c, x_c, y_c, d_c, q_1, q_2)$, $c_d = (Q_{cd}, x_{cd}, y_{cd}, d_{cd}, q_{1d}, q_{2d})$, where c_d is the desired cluster variables and c is the actual cluster variables. The error signal is $\Delta c = W(c_d - c)$ where W is a positive weighting diagonal matrix. The robot space commands are transformed from the cluster space signals to robot space signals by multiplying them with the inverse of the jacobian matrix, as $\overline{\Delta r} = J^{-1} \Delta c$ where $\overline{\Delta r} = [\Delta x_1, \Delta y_1, \Delta \theta_1, \Delta x_2, \Delta y_2, \Delta \theta_2]^T$ and $\overline{\Delta r}_i = [\Delta x_i, \Delta y_i, \Delta \theta_i]^T$, in order to deal with the nonholonomic constraints, the robot space commands are modified as the following; $\Delta r = [\Delta x_1, \Delta y_1, \theta_{1e}, \Delta x_2, \Delta y_2, \theta_{2e}]^T$ and $\Delta r_i = [\Delta x_i, \Delta y_i, \theta_{ie}]^T$ where $\theta_{ie} = \tan_2^{-1}(\frac{\Delta y_i}{\Delta x_i}) - q_i$, then a transformation of the robot commands from a global frame to a robot frame is done by using the following rotational transformation;

$$r_{ie} = \begin{bmatrix} x_{ie} \\ y_{ie} \\ \theta_{ie} \end{bmatrix} = \begin{bmatrix} \cos(q_i) & \sin(q_i) & 0 \\ -\sin(q_i) & \cos(q_i) & 0 \\ 0 & 0 & 1 \end{bmatrix} \Delta r_i.$$

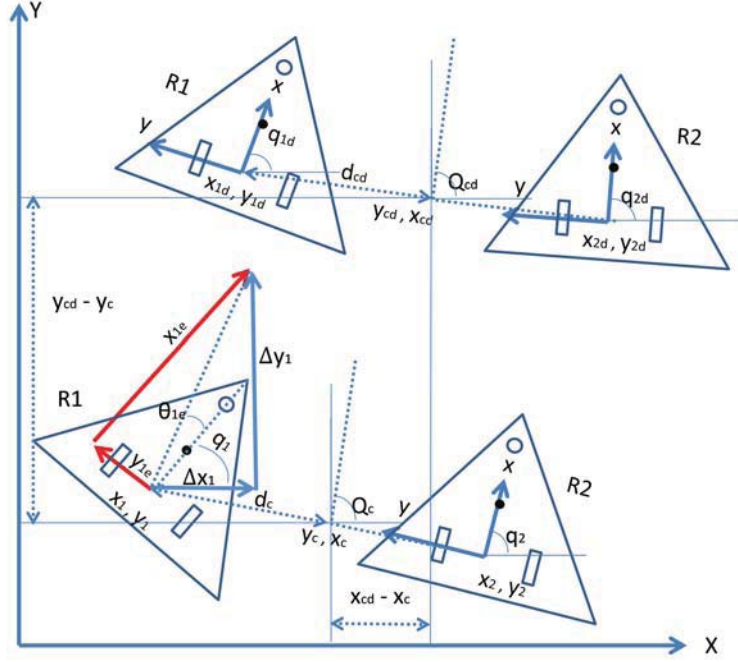


Figure 4.1: Cluster location errors

where $(x_{ie}, y_{ie}$ and $\theta_{ie})$ are shown in Figure 4.1. Then by defining $V_i = [v_i, w_i]^T$, $V_{id} = [v_{id}, w_{id}]^T$ that the v_i, w_i are the actual robot heading and rotational robot's velocities and v_{id}, w_{id} are the desired robot heading and rotational Robot's velocities. It was found in [60,69,70] that the derivatives of the robot profile errors can be found as shown in equation 4.1

$$\dot{r}_{ie} = \begin{bmatrix} \dot{x}_{ie} \\ \dot{y}_{ie} \\ \dot{\theta}_{ie} \end{bmatrix} = \begin{bmatrix} y_{ie}w_i - v_i + v_{id} \cos(\theta_{ie}) \\ -x_{ie}w_i + v_{id} \sin(\theta_{ie}) \\ w_{id} - w_i \end{bmatrix}. \quad (4.1)$$

4.1.2 Sliding mode derivation

In this section, a model-based sliding-mode controller (SMC) is developed. The first step in designing an SMC is selecting a sliding surface where the system

dynamics are stable; this sliding surface is a function of the system states.

Designing the sliding surface for a single robot Starting, from the classical kinematic controller given by [71], we get the following velocity controller in equation 4.2

$$v_c = \begin{bmatrix} v_{1c} \\ w_{1c} \\ \vdots \\ v_{nc} \\ w_{nc} \end{bmatrix} = \begin{bmatrix} v_{1d} \cos(\theta_{1e}) + k_{11}x_{1e} \\ w_{1d} + k_{12}v_{1d}y_{1e} + k_{13}v_{1d} \sin(\theta_{1e}) \\ \vdots \\ v_{nd} \cos(\theta_{ne}) + k_{n1}x_{ne} \\ w_{nd} + k_{n2}v_{nd}y_{ne} + k_{n3}v_{nd} \sin(\theta_{ne}) \end{bmatrix}, \quad (4.2)$$

where $k_{i1}, k_{i2}, k_{i3} > 0, i|i = 1, \dots, n$. Thus, the error in the robots' kinematics is defined as:

$$e_c(t) = [e_{c1}(t), e_{c2}(t), \dots, e_{c2n}]^T = v_c(t) - v(t),$$

$$\dot{e}_c(t) = \dot{v}_c(t) - \dot{v}(t),$$

where

$$v = \begin{bmatrix} v_1 \\ w_1 \\ \vdots \\ v_n \\ w_n \end{bmatrix}.$$

Then a PI-type sliding surface is selected, as in equation 4.3. The adaptive integral component offers a fast convergence of the sliding surface than the standard SMC and a smooth control of the system resulting in zero steady-state error [72].

$$s(t) = \begin{bmatrix} s_1(t) \\ s_2(t) \\ \vdots \\ s_{1 \cdot i}(t) \\ s_{2 \cdot i}(t) \\ \vdots \\ s_{1 \cdot n}(t) \\ s_{2 \cdot n}(t) \end{bmatrix} = e_c(t) + \varphi \int e_c(t) dt, \quad (4.3)$$

Where φ is a positive integer. So, if the system is on the sliding surface $s(t) = 0$, $e_c(t) = -\varphi \int_0^t e_c(t) dt$ and if $t \rightarrow \infty$ then $e_c \rightarrow 0$. In order to have the control signal, the derivation of the sliding surface is found in equation 4.4.

$$\dot{s}(t) = [\dot{v}_c(t) - H^{-1}(\tau_{eq} - E)] + \beta e_c(t) = 0. \quad (4.4)$$

Then by rearranging equation 4.4 we arrive at equation 4.5

$$\tau_{eq} = H [\dot{v}_c(t) + \varphi e_c(t)] + E. \quad (4.5)$$

Now τ_{eq} can keep the system inside the surface. But what if the system dynamics are already outside the surface? In order to guarantee the stability in that case, another control signal τ_r should push the system dynamics onto the sliding surface; this mechanism is called the reachability law or a robustifying term. Now the new control signal τ is a combination of $\tau_{eq} + \tau_r$ as shown in equation 4.6

$$\tau = H [\dot{v}_c(t) + \beta e_c(t) + K \cdot \text{sgn}(s)] + E, \quad (4.6)$$

where $K = \begin{bmatrix} K_1 & 0 & \cdots & 0 \\ 0 & \ddots & 0 & 0 \\ \vdots & 0 & K_i & \vdots \\ 0 & 0 & 0 & K_n \end{bmatrix} \{K_i \mid K_i > 0\}$

and the $\text{sgn}(s(t)) = [\text{sgn}(s_1(t)), \text{sgn}(s_2(t)), \dots, \text{sgn}(s_{2.n})]$

The uncertainty of the system is a function of the disturbance τ_d and the uncertainty in the model itself, such that the error in the dynamical model and the actual model is δ :

$$\delta(t) = \Delta H^{-1}(\tau - E) + H^{-1}(-\Delta E) + H^{-1}\tau_d.$$

Now the dynamic of the system can be written as in equation 4.7

$$\dot{v} = H^{-1}(\tau - E) + \delta. \quad (4.7)$$

However, using the sign function in the reaction law is not preferred for practical work, and this is due to the issue of chattering. Therefore, the error function

$erf(s)$ is used instead.

$$\tau = H [\dot{v}_c(t) + \beta e_c(t) + K \cdot erf(s)] + E. \quad (4.8)$$

4.1.3 Adaptive sliding mode

Theorem 4.1 *Assuming the adaptive law as shown in equation 4.9*

$$\dot{\hat{\gamma}} = \begin{bmatrix} \dot{\hat{\gamma}}_1 & 0 & \cdots & 0 \\ 0 & \dot{\hat{\gamma}}_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \dot{\hat{\gamma}}_i \end{bmatrix} = \begin{bmatrix} \varrho_1 s_1 erf(s_1) & 0 & \cdots & 0 \\ 0 & \varrho_2 s_2 erf(s_2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \varrho_n s_n erf(s_n) \end{bmatrix}, \quad (4.9)$$

where $\varrho_i \in \Re > 0$ and the final controller signal is in equation 4.10

$$\tau = H [\dot{v}_c(t) + \beta e_c(t) + \hat{\gamma} \cdot erf(s)] + E. \quad (4.10)$$

The estimated error is defined as : $\hat{\gamma}(t) = \tilde{\gamma}(t) - \gamma^*$.

Proof Using the Lyapunov function: let $L = L_1 + L_2 + L_3$ such that $L(0) = L_1(0) = L_2(0) = L_3(0) = 0$ and $L, L_1, L_2, L_3 > 0$ for inputs other than 0, where

$$L_1 = \frac{1}{2} \sum_{i=1}^n \left(x_{ie}^2 + y_{ie}^2 + \frac{1 - \cos(\theta_{ie})}{k_{i2}} \right),$$

$$L_2 = \frac{1}{2} s^T(t) s(t) + \frac{1}{2} \sum_{i=1}^n \left(\frac{1}{\varrho_i} \tilde{\gamma}_i^2 \right),$$

$$L_3 = E_c^T E_c.$$

Based on the Lyapunov theory, the system is stable if and only if $\dot{L} < 0$. After derivation we get

$$\dot{L}_1 = \sum_{i=1}^n \left(-k_{i1} x_{ie}^2 - \frac{k_{i3} v_{id} \sin^2(\theta_{ie})}{k_{i2}} \right),$$

$$\dot{L}_2 = s^T \dot{s} + \sum_{i=1}^n \left(\frac{1}{\varrho_i} \tilde{\gamma}_i \dot{\tilde{\gamma}}_i \right).$$

Now $\dot{L}_1 < 0$. And after substituting $\dot{\tilde{\gamma}}$ based on $\dot{\tilde{\gamma}} = \dot{\hat{\gamma}}$.

$$\dot{L}_2 = s^T [-\hat{\gamma} \text{erf}(s) - \delta] + \sum_{i=1}^n \left(\frac{1}{\varrho_i} \tilde{\gamma}_i \dot{\tilde{\gamma}}_i \right),$$

$$\dot{L}_2 = s^T [-(\tilde{\gamma} + \gamma^*) \text{erf}(s) - \delta] + \sum_{i=1}^n \left(\frac{1}{\varrho_i} \tilde{\gamma}_i \dot{\tilde{\gamma}}_i \right),$$

$$\dot{L}_2 = s^T [-\gamma^* \text{erf}(s) - \delta] + \sum_{i=1}^n \left[\tilde{\gamma}_i \left(\frac{1}{\varrho_i} \dot{\tilde{\gamma}}_i - s_i \text{erf}(s_i) \right) \right].$$

And once the adaptive equation is substituted, the resultant \dot{L}_2 is given as

$$\dot{L}_2 = s^T [-\gamma^* \text{erf}(s) - \delta],$$

$$\dot{L}_2 = \sum_1^i s_i [-\gamma_i^* \text{erf}(s_i) - \delta_i] \leq -\sum_1^i |\delta_i| s_i \text{erf}(s_i) (\gamma_i^* - |\delta_i|).$$

And the $\dot{L}_2 < 0$ can be guaranteed by selecting $\gamma^* > |\delta|$.

The L_3 derivation is

$$\dot{L}_3 = E_c^T \dot{E}_c,$$

where $E_c = (c_d - c)$, $\dot{E}_c = (\dot{c}_d - \dot{c})$ and $\dot{c} = W E_c$. By substitution we arrive at:

$$\dot{E}_c = (\dot{c}_d - W E_c)$$

$$\text{Then } \dot{L}_3 = (E_c^T \dot{c}_d - E_c^T W E_c).$$

Knowing that $-E_c^T W E_c < -\lambda_{\min}(W) \|E_c\|^2$, the $\dot{L}_3 < 0$ provided that

$$\|E_c\|^T < \frac{\|\dot{c}_d\|}{\lambda_{\min}(W)}.$$

As a result, the derivative of the Lyapunov function is $\dot{L} < 0$ and the stability is proved.

4.1.4 Control diagram

In this subsection the control diagram is presented. Figure 4.2 shows the adaptive sliding mode controller.

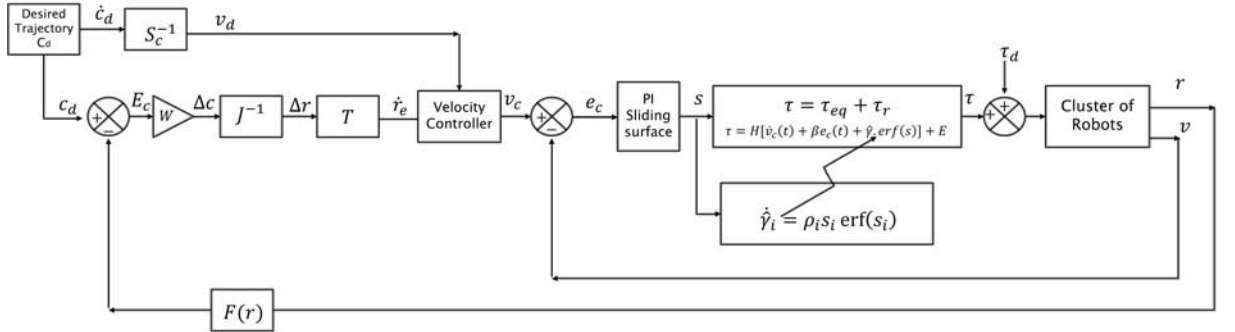


Figure 4.2: The proposed adaptive SMC diagram

4.2 Simulation Results

4.2.1 Two robot simulation

Based on the control law established in Section 4.1.3, a simulation using MATLAB@ is implemented on the cluster of two mobile robots. The cluster space of two nonholonomic robots was used. The sinusoidal desired trajectory was applied as follows:

$$x_{cd} = 5 \cos(0.3t),$$

$$y_{cd} = 15 \sin(0.1t),$$

$$Q_{cd} = \tan^{-1} \left(\frac{dx_{cd}}{dy_{cd}} \right).$$

Let $K = [10, 2, 10, 2]$ and the initial position of the mobile robots are set as $(x_1, y_1, q_1) = (1, 0, 0)$ and $(x_2, y_2, q_2) = (0, 0, 0)$. The simulation results are shown in Figure (4.3) and Figure (4.4). Figure (4.3) illustrates the trajectory tracking result for the sinusoidal function. The actual trajectory reaches the desired line quickly. In Figure (4.4) a disturbance was introduced after 20 sec; the results of the SMC and adaptive SMC were compared by comparing the tracking error in the cluster states x_c, y_c, Q_c and d_c , respectively. And in figure 4.5 the adaptive parameters response's are shown changing in the transient zone, the adapting velocity depends on the adaptation constants. According to the simulation results, the sinusoidal trajectory tracking shows good performance, especially with the adaptive term. This validates the sliding control algorithm by simulation.

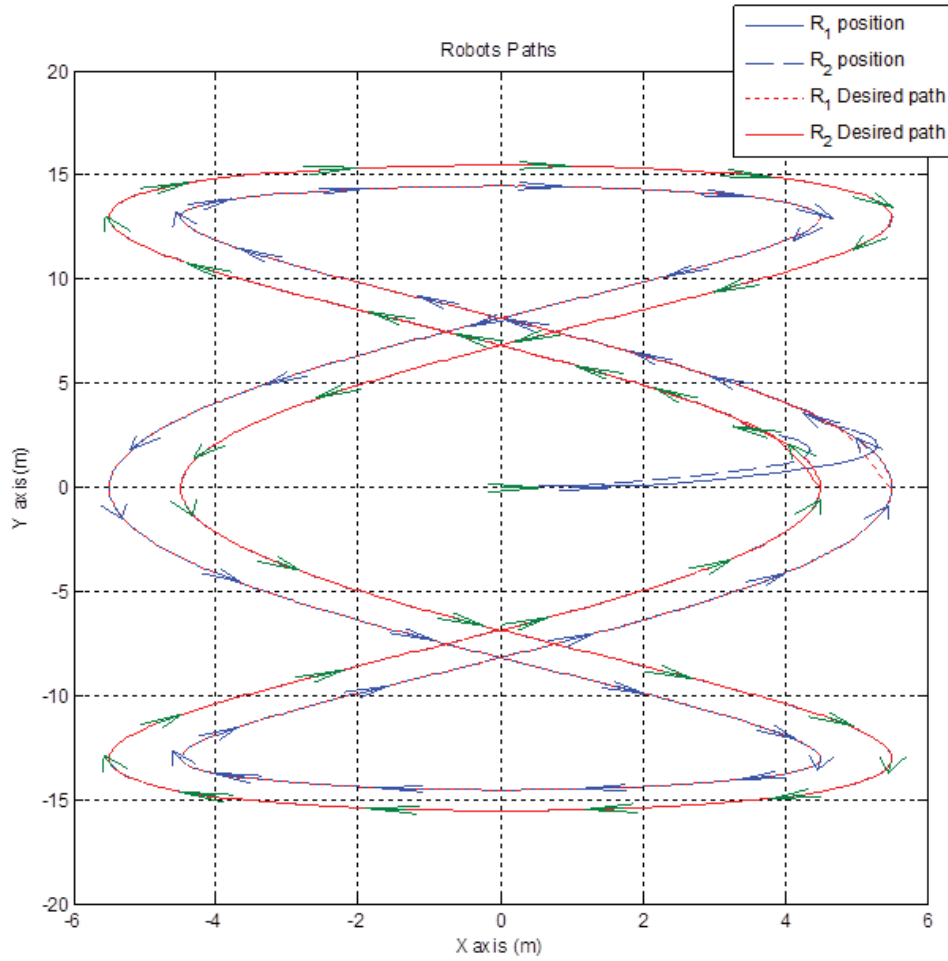


Figure 4.3: Adaptive SMC control path tracking profile

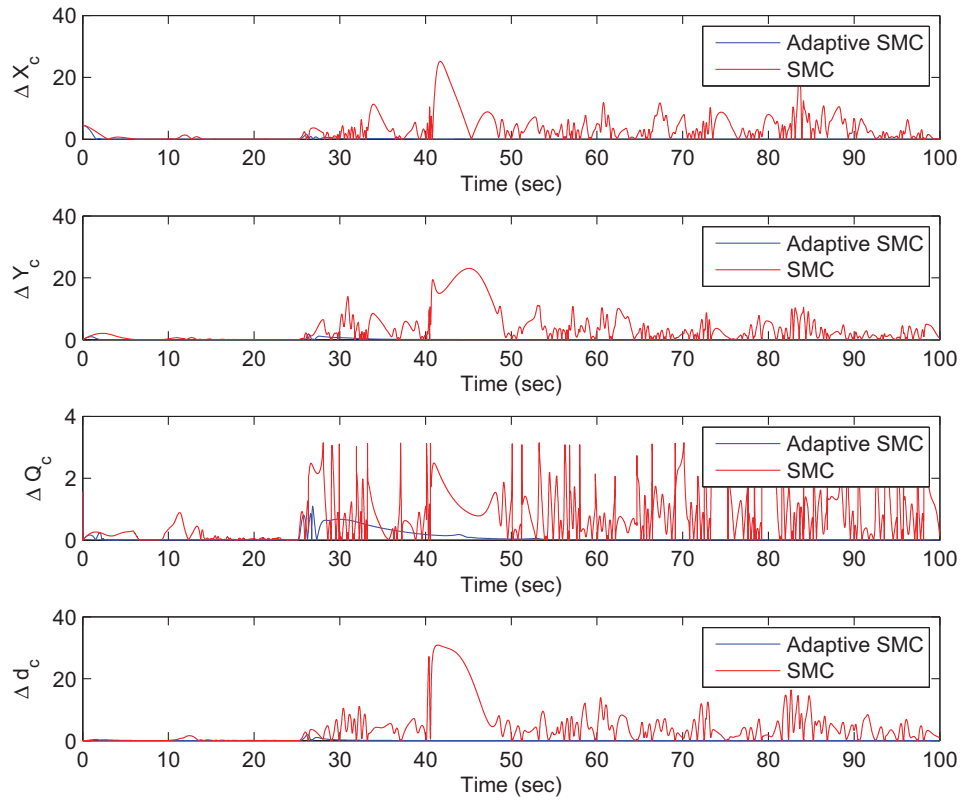


Figure 4.4: Comparison between SMC and Adaptive SMC with disturbance injected after 20 sec

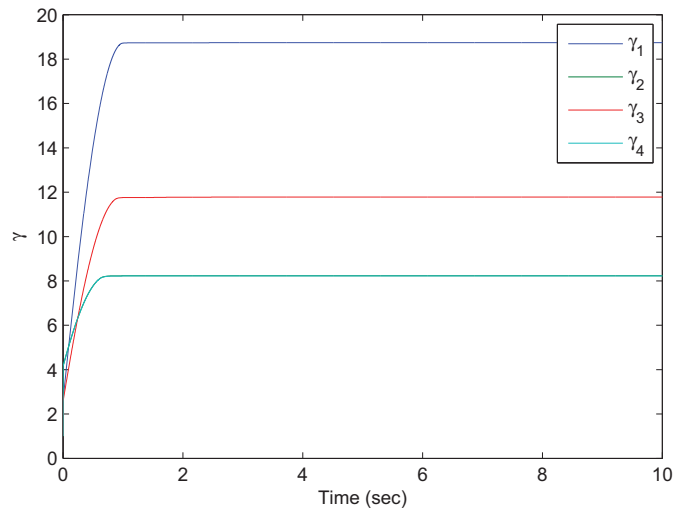


Figure 4.5: The adaptive parameters response with no disturbance case

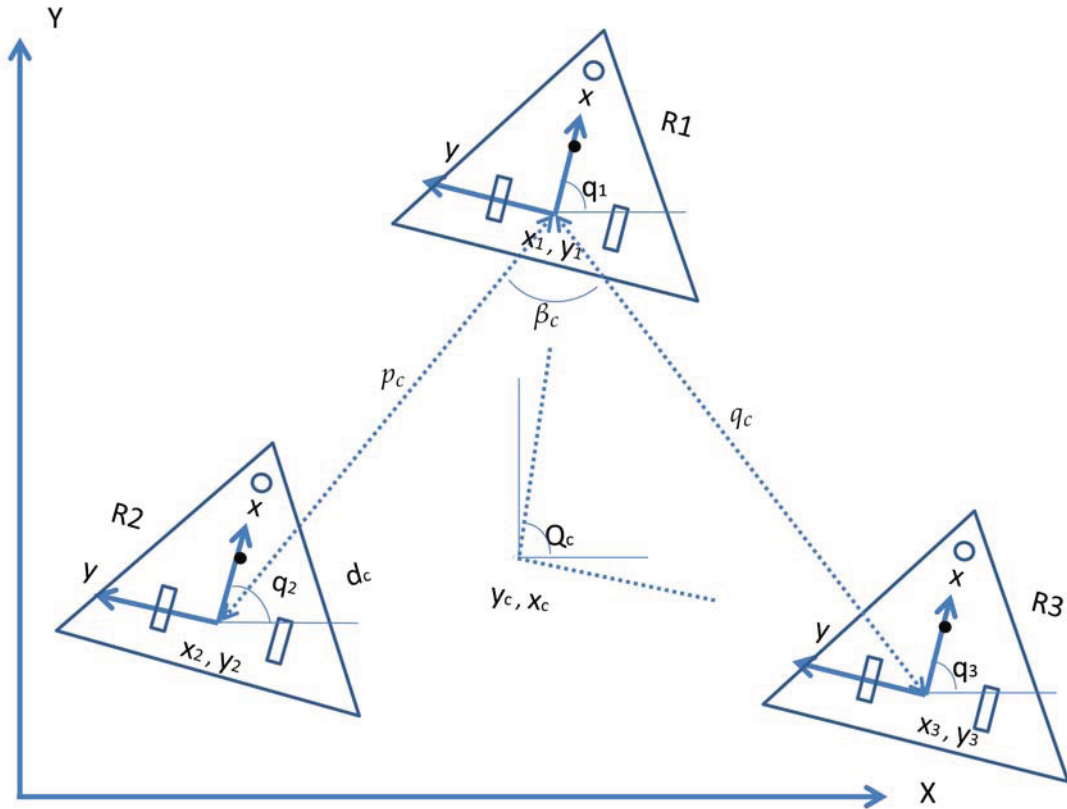


Figure 4.6: 3-robot system configuration

4.2.2 Simulation results of 3-robot system

In this subsection a three-robot cluster was simulated with the same reference trajectory presented in subsection 4.2.1. The selected cluster spaces are $(x_c, y_c, Q_c, \beta_c, q_c, p_c, q_1, q_2, q_3)$; see Figure (4.6) and for more details refer to [17]. The simulation results (see Figure (4.7)) show good performance of the proposed algorithm.

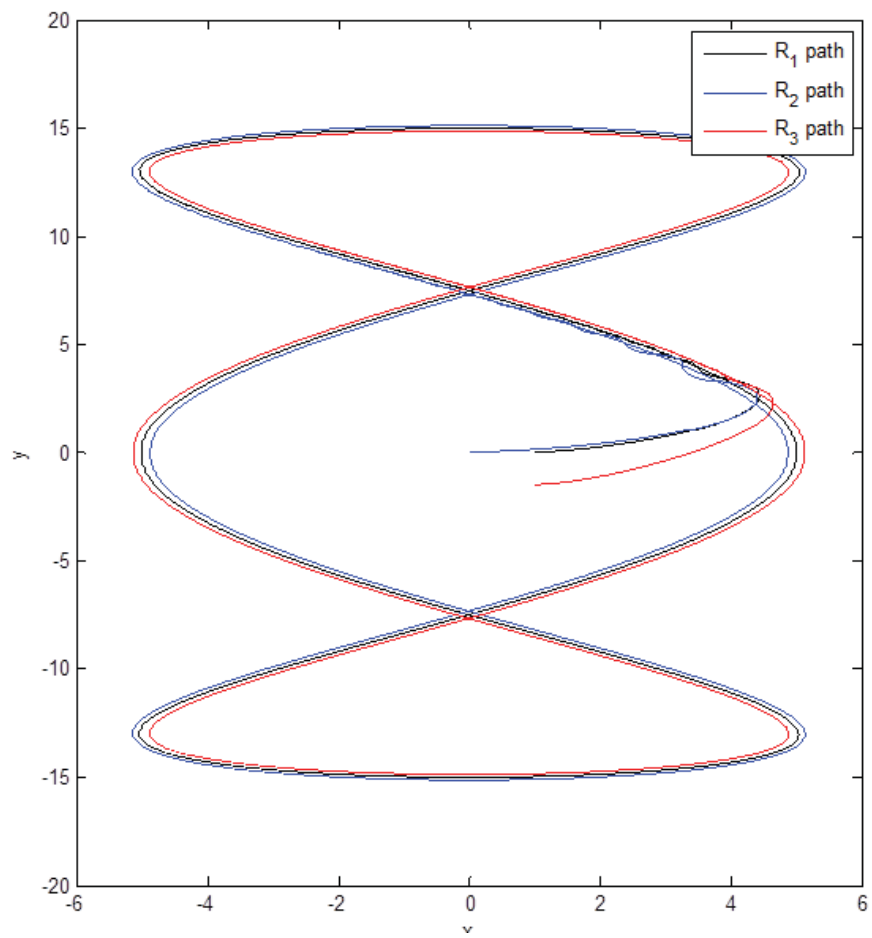


Figure 4.7: Adaptive SMC control path tracking profile

4.3 Experimental application

The Lego EV3 WMRs are used in the experiments. Those WMRs (see Figure 1.1) have a 32-bit, For more details about the experimental setup refer to section 1.6.

Practical issues The output from the centralized controller is the desired wheel's torque, but due to the braking behavior of the EV3 servo motor, (when the motor has zero input) the wheel will stop abruptly. This braking is an additional variable that is not considered when developing the sliding mode controller. To improve the response the following solution has been used: 1) add an integrator filter to the controller output torque, and the output signal of this filter is then considered to be a velocity command; 2) instead of using a voltage or power signal to control the servo motor, the velocity command is sent to an inner speed loop controller. Accordingly, the servo motor will follow a torque equal to the sliding mode torque multiplied by some design constant, which is found through trial and error. For more, see Figure (4.8), where W_{cmd} is the velocity command.



Figure 4.8: Using an integrator to change an SMC command to a velocity command calculation and the inner loop diagram

4.3.1 Experiment test

Two WMRs move from initial positions (x, y, q) , which for the first robot is $(0.2, 0, \frac{\pi}{2})$ and $(0, 0, \frac{\pi}{2})$ for the second robot. The desired path is $Q_c = \frac{\pi}{2}, x_c = 0.3, y_c = 0.1t$ and $d_c = 0.25$ in the cluster space which is equivalent to $x_1 = 0.55, y_1 = 0.1t, q_1 = \frac{\pi}{2}, x_2 = 0.05, y_2 = 0.1t$ and $q_2 = \frac{\pi}{2}$ in the robot space, where t is the time (see Figure 4.9). The dynamic parameters are assumed to be nominal values. so the robot mass is $0.5kg$, the robot inertia is $0.0025 kg.m^2$, the wheel diameter is $56mm$ and the distance between the wheels is $11.8cm$.

The experimental results are shown in Figures (4.10 and 4.11) . Figure (4.10) illustrates the trajectory tracking result for a line trajectory with SMC. The actual trajectory reaches the desired line quickly; however, chattering can be easily

seen in the actual robots' trajectory. This chattering was due to the uncertainty of the robots' model and the untuned controller parameters. Figure (4.11) shows the actual trajectory with the adaptive SMC, and the chattering effects are eliminated with the adaptive controller. Figure (4.12) shows the comparison between the SMC and the adaptive SMC based on the cluster states tracking x_c, y_c, Q_c and d_c , respectively. According to the experimental results for the line trajectory tracking, the trajectory can remain stable and robust despite the uncertainty in the robot's inertia and mass. This validates the proposed algorithms experimentally.

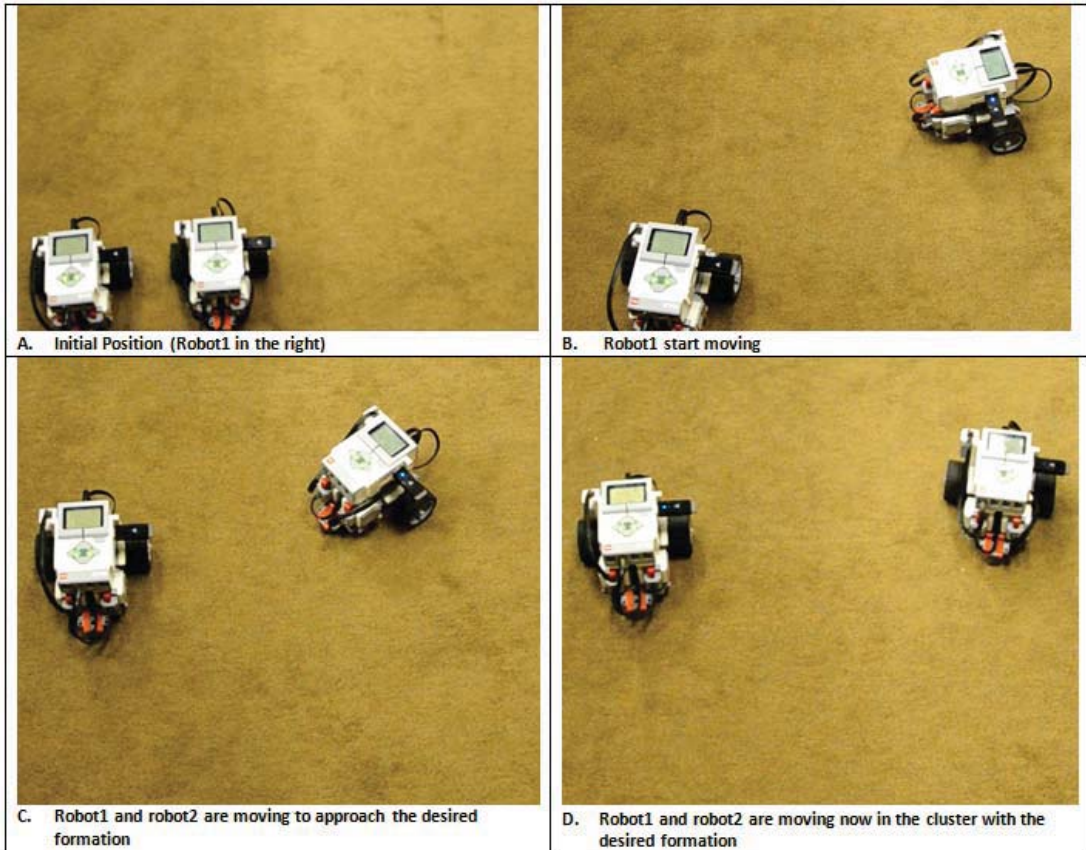


Figure 4.9: Experimental cluster space control on a 2-robot system

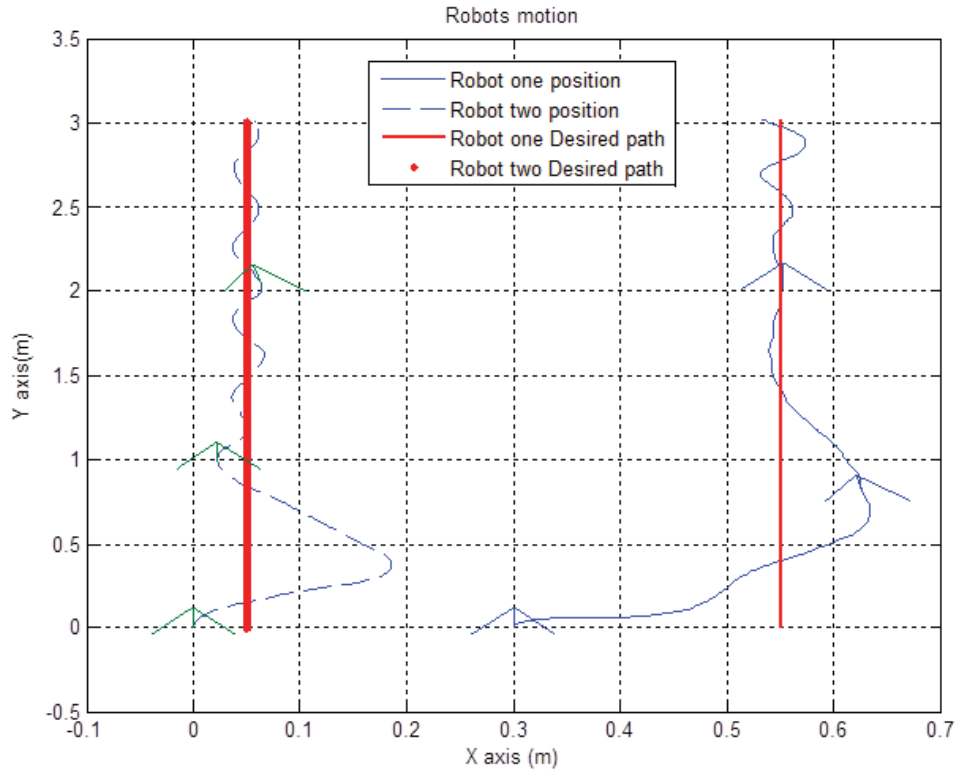


Figure 4.10: Experimental movement trajectory of a cluster of two robot with SMC

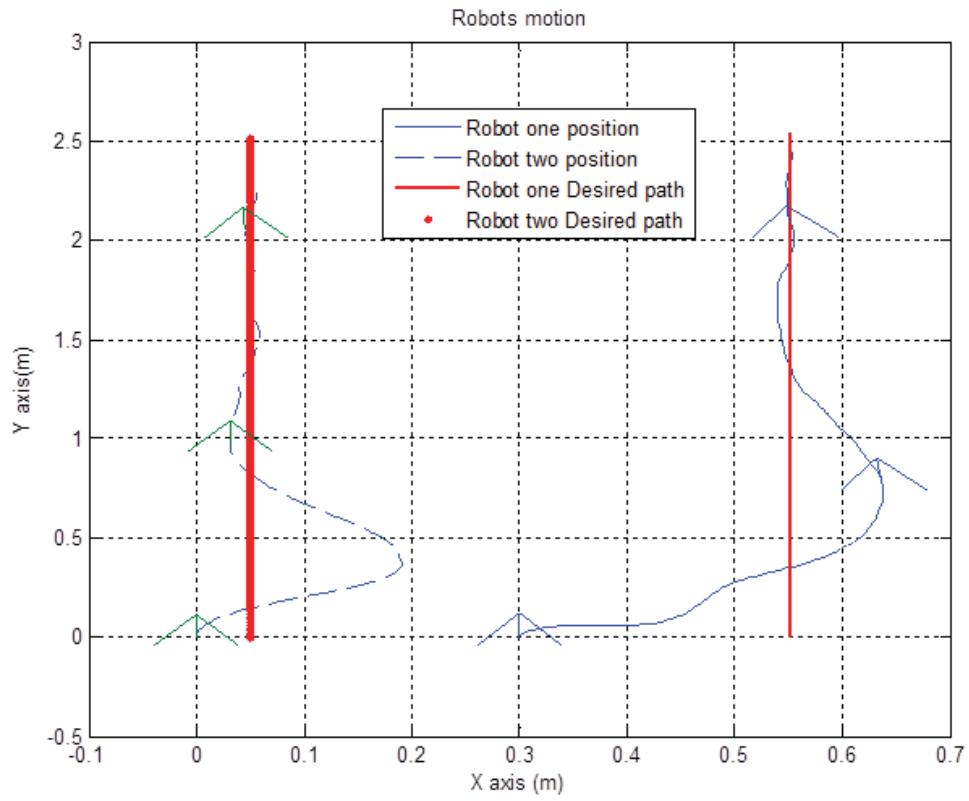


Figure 4.11: Experimental movement trajectory of a cluster of two robot with adaptive SMC

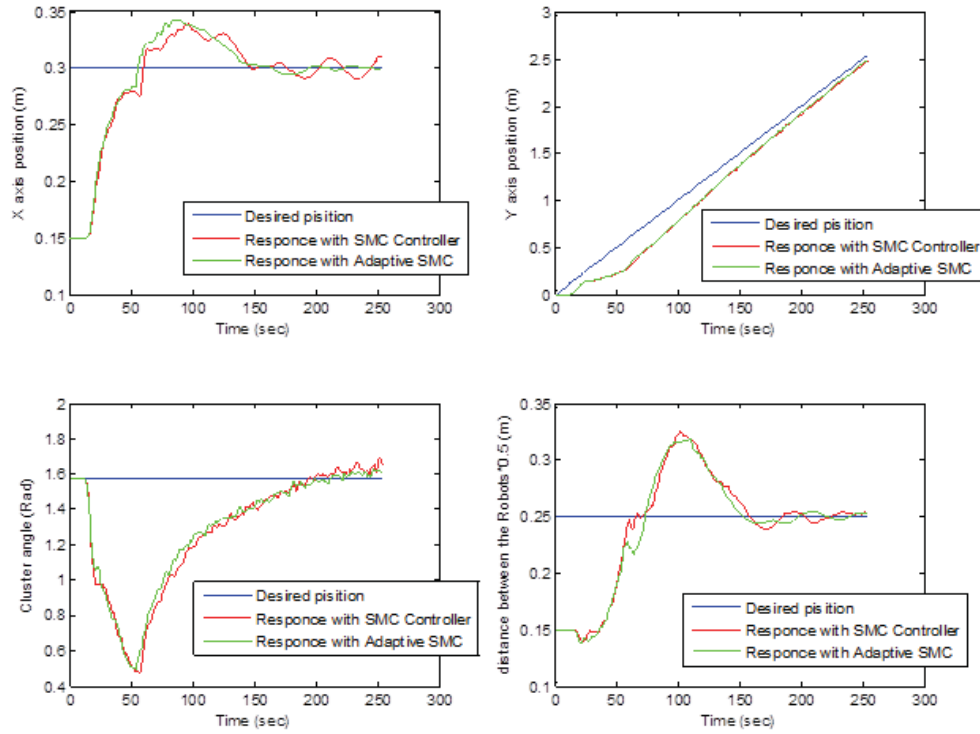


Figure 4.12: The experimental desired and actual cluster dynamics states x_c, y_c, Q_c and d_c , and a comparison between the SMC and the adaptive SMC

4.4 Chapter summary

This chapter proposed a model-based multi nonholonomic robot controller in the cluster space by developing an adaptive sliding mode control algorithm. This includes the presentation of the mathematical model of the nonholonomic multi-robot cluster, and the design of the sliding surface function and of the control law. According to the simulation and experimental results, the proposed adaptive sliding mode control is an important method to deal with a cluster of multi-robots in which uncertainties and nonlinearities exist. In spite of large highly

nonlinear dynamics, the robots' cluster shows that the posture converges to the desired trajectory. Future work may investigate clusters with a larger number of heterogeneous robots and apply intelligent methods like neural networks and fuzzy logic and to overcome the singularities in the cluster dynamics, in addition to addressing the obstacle avoidance problem.

CHAPTER 5

FUZZY ADAPTIVE SLIDING MODE CLUSTER SPACE CONTROL OF A NONHOLONOMIC MULTI-ROBOT SYSTEM

5.1 Introduction

In [73] an adaptive SMC is proposed to overcome the uncertainty issues in the cluster space model. However, this adaptive law is derivative based. and this may lead to making the system sensitive to noise. Furthermore, it results in a time delay to tune the controller parameters. Because of that, we need a more

reliable and faster tuning approach. One major approach is the artificial intelligent fuzzy adaptive method. In this method, the SMC parameters will be tuned using fuzzy rules. From the literature, this technique has been used several times with the sliding mode controller for different robotics applications. [70] proposed a fuzzy adaptive sliding mode controller for the single wheeled mobile robot. In addition, [74] proposed using fuzzy logic to overcome the chattering in the sliding mode controller. Furthermore, the leader-follower concept in [75,76] used a fuzzy sliding mode controller to tackle the model uncertainty issue. [77] proposed using a fuzzy sliding mode to control multi-robot using the graph theory approach. Also [78] used fuzzy logic for tuning the sliding mode controller, in addition to using the error and its derivative as inputs to the fuzzy logic to enhance the performance of the controller.

In this chapter, the latest multi-robot formation control using the adaptive sliding mode controller [73] is extended by adding intelligent fuzzy adaptive sliding mode formation control of a group of nonholonomic robots in a cluster space. This addition is similar to [78] which was used to improve the performance of the SMC. The proposed adaptive robust controller addresses the issue of model uncertainties, as well as the systems nonlinear dynamics. Those issues are common in real-life applications where there are constraints on the robots motion that are transformed into a nonlinear behavior, while the model parameters either vary in time or are uncertain. The chapter reports the experimental implementation of the proposed scheme on a real ground robot connected to a central controller using a

wi-fi network.

5.2 Controller design

In this section the fuzzy adaptive SMC is presented as follows: first the error signal calculation, then the sliding surface selection, followed by the fuzzy adaptive SMC and the controller diagram. As an example, this development is done for a two-robot cluster, while the same procedure is done for a cluster with any number of robots. Regarding the cluster space, dynamic model refers to the authors' previous work [73].

5.2.1 The cluster profile errors

The formation error in the cluster spaces is found by the following: Starting with defining $c = (x_c, y_c, Q_c, d_c, q_1, q_2)$, $c_d = (x_{cd}, y_{cd}, Q_{cd}, d_{cd}, q_{1d}, q_{2d})$, where c_d is the desired cluster spaces and c is the actual cluster spaces. The error signal is $\Delta c = W(c_d - c)$ where W is a positive weighting diagonal matrix. The robot space commands are transformed from the cluster space signals into robot space signals by multiplying them with the inverse of the Jacobian matrix, as $\overline{\Delta r} = J^{-1} \Delta c$ where $\overline{\Delta r} = [\Delta x_1, \Delta y_1, \Delta \theta_1, \Delta x_2, \Delta y_2, \Delta \theta_2]^T$ and $\overline{\Delta r}_i = [\Delta x_i, \Delta y_i, \Delta \theta_i]^T$, in order to deal with the nonholonomic constraints the robot space commands are modified as follows; $\Delta r = [\Delta x_1, \Delta y_1, \theta_{1e}, \Delta x_2, \Delta y_2, \theta_{2e}]^T$ and $\Delta r_i = [\Delta x_i, \Delta y_i, \theta_{ie}]^T$ where $\theta_{ie} = \tan_2^{-1}(\frac{\Delta y_i}{\Delta x_i}) - q_i$ then a transformation of the robot commands from a global frame to a robot frame is done by using the following rotational transformation;

$$r_{ie} = \begin{bmatrix} x_{ie} \\ y_{ie} \\ \theta_{ie} \end{bmatrix} = \begin{bmatrix} \cos(q_i) & \sin(q_i) & 0 \\ -\sin(q_i) & \cos(q_i) & 0 \\ 0 & 0 & 1 \end{bmatrix} \Delta r_i.$$

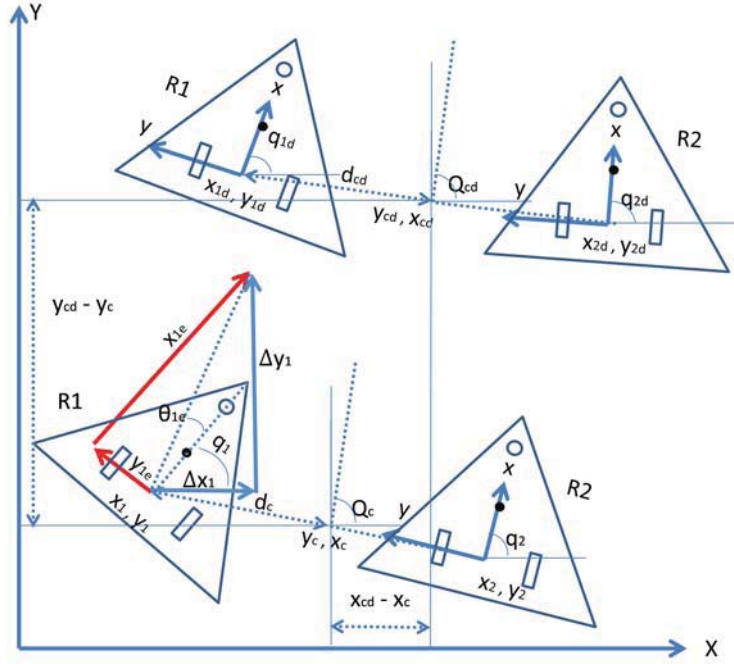


Figure 5.1: Cluster location errors

where $(x_{ie}, y_{ie}$ and $\theta_{ie})$ are shown in Figure 5.1. Then by defining $V_i = [v_i, w_i]^T$, $V_{id} = [v_{id}, w_{id}]^T$ where the v_i, w_i are the actual robot heading and rotational robot's velocities respectively and v_{id}, w_{id} are the desired robot heading and rotational robot's velocities respectively. It was shown in [60, 69, 70] that the derivative of the robot profile errors can be found by equation 5.1

$$\dot{r}_{ie} = \begin{bmatrix} \dot{x}_{ie} \\ \dot{y}_{ie} \\ \dot{\theta}_{ie} \end{bmatrix} = \begin{bmatrix} y_{ie}w_i - v_i + v_{id} \cos(\theta_{ie}) \\ -x_{ie}w_i + v_{id} \sin(\theta_{ie}) \\ w_{id} - w_i \end{bmatrix}. \quad (5.1)$$

5.2.2 Sliding mode derivation

In this section, a model-based sliding mode controller (SMC) is developed. The first step in designing an SMC is selecting a sliding surface where the system dynamics are stable; this sliding surface is a function of the system states.

Designing the sliding surface for a single robot Starting from the classical kinematic controller given by [71], we get the following velocity controller in equation 5.2

$$v_c = \begin{bmatrix} v_{1c} \\ w_{1c} \\ \vdots \\ v_{nc} \\ w_{nc} \end{bmatrix} = \begin{bmatrix} v_{1d} \cos(\theta_{1e}) + k_{11}x_{1e} \\ w_{1d} + k_{12}v_{1d}y_{1e} + k_{13}v_{1d} \sin(\theta_{1e}) \\ \vdots \\ v_{nd} \cos(\theta_{ne}) + k_{n1}x_{ne} \\ w_{nd} + k_{n2}v_{nd}y_{ne} + k_{n3}v_{nd} \sin(\theta_{ne}) \end{bmatrix}, \quad (5.2)$$

where $k_{i1}, k_{i2}, k_{i3} > 0, i|i = 1, \dots, n$. Thus, the error in the robots' kinematics is defined as:

$$e_c(t) = [e_{c1}(t), e_{c2}(t), \dots, e_{c2 \cdot n}]^T = v_c(t) - v(t),$$

$$\dot{e}_c(t) = \dot{v}_c(t) - \dot{v}(t),$$

where

$$v = \begin{bmatrix} v_1 \\ w_1 \\ \vdots \\ v_n \\ w_n \end{bmatrix}.$$

Then a PI-type sliding surface is selected, as in equation 5.3. The adaptive integral component offers a faster convergence of the sliding surface than the standard SMC and a smooth control of the system resulting in a zero steady-state error [72].

$$s(t) = \begin{bmatrix} s_1(t) \\ s_2(t) \\ \vdots \\ s_{1 \cdot i}(t) \\ s_{2 \cdot i}(t) \\ \vdots \\ s_{1 \cdot n}(t) \\ s_{2 \cdot n}(t) \end{bmatrix} = e_c(t) + \beta \int e_c(t) dt, \quad (5.3)$$

Where β is a positive integer. Consequently, if the system is on the sliding surface $s(t) = 0$, $e_c(t) = -\beta \int_0^t e_c(t) dt$ and if $t \rightarrow \infty$ then $e_c \rightarrow 0$. In order to have the control signal, the derivation of the sliding surface is found in equation 5.4.

$$\dot{s}(t) = [\dot{v}_c(t) - H^{-1}(\tau_{eq} - E)] + \beta e_c(t) = 0. \quad (5.4)$$

Then by rearranging equation 5.4 we arrive at equation 5.5

$$\tau_{eq} = H [\dot{v}_c(t) + \beta e_c(t)] + E. \quad (5.5)$$

Now τ_{eq} can keep the system inside the surface. But what if the system dynamics are already outside the surface? In order to guarantee the stability in that case, another control signal τ_r should push the system dynamics onto the sliding surface; this mechanism is called the reachability law or a robustifying term. Now the new control signal τ is a combination of $\tau_{eq} + \tau_r$ as shown in equation 5.6

$$\tau = H [\dot{v}_c(t) + \beta e_c(t) + K \cdot \text{sgn}(s)] + E, \quad (5.6)$$

where $K = \begin{bmatrix} K_1 & 0 & \cdots & 0 \\ 0 & \ddots & 0 & 0 \\ \vdots & 0 & K_i & \vdots \\ 0 & 0 & 0 & K_n \end{bmatrix} \{K_i \mid K_i > 0\}$

and the $\text{sgn}(s(t)) = [\text{sgn}(s_1(t)), \text{sgn}(s_2(t)), \dots, \text{sgn}(s_{2-n})]$

The uncertainty of the system is a function in the disturbance τ_d and the uncertainty in the model itself, such that:

$$\delta(t) = \Delta H^{-1}(\tau - E) + H^{-1}(-\Delta E) + H^{-1}\tau_d.$$

Now the dynamic of the system can be written as in equation 5.7

$$\dot{v} = H^{-1}(\tau - E) + \delta. \quad (5.7)$$

However, using the sign function in the reaction law is not preferred for practical work, and this is due to the issue of chattering. Therefore, the error function $erf(s)$ is used instead.

$$\tau = H [\dot{v}_c(t) + \beta e_c(t) + K \cdot erf(s)] + E. \quad (5.8)$$

5.2.3 Fuzzy Adaptive sliding mode

Theorem 5.1 *Assuming the fuzzy adaptive law as in equation 5.9*

$$\gamma = \begin{bmatrix} \gamma_1 & 0 & \cdots & 0 \\ 0 & \gamma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \gamma_i \end{bmatrix}, \quad (5.9)$$

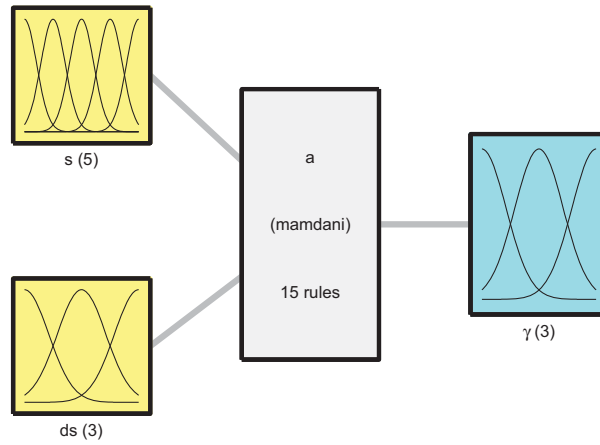
where $\gamma_i > 0$ and function of s_i, \dot{s}_i using the fuzzy rule in table 5.1. The final controller signal is given in equation 5.10

Table 5.1: The fuzzy rules of γ_i

γ_i		s_i				
		NB	NS	Z	PS	PB
\dot{s}_i	N	KB	KB	KM	KM	KM
	Z	KB	KM	KS	KM	KB
	P	KM	KM	KM	KB	KB

Figure 5.2 shows the fuzzy logic structure of the γ function, and the input member-ship functions for both s_i and \dot{s}_i are shown in Figure 5.3, and the mem-

membership function for the output γ are shown in Figure 5.4, the names of the membership function are similar to the names in chapter 3.



System a: 2 inputs, 1 outputs, 15 rules

Figure 5.2: Fuzzy structure

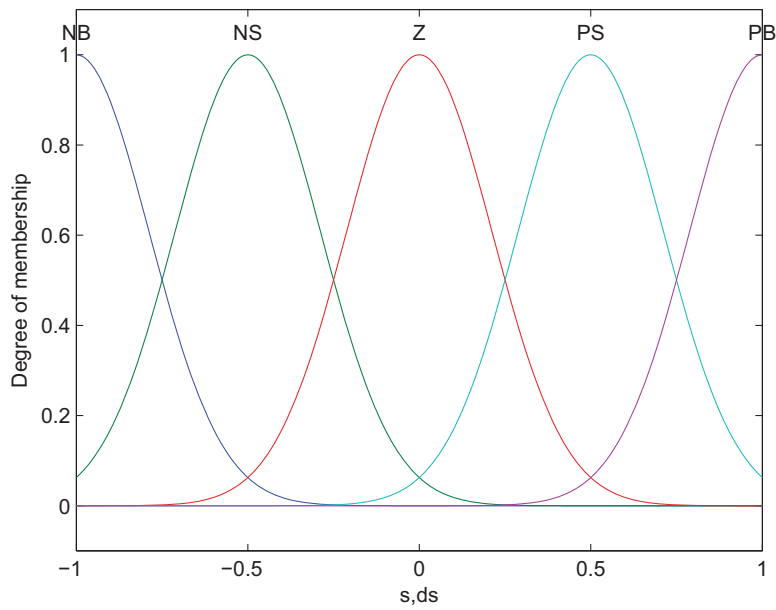


Figure 5.3: The membership function for the s_i and \dot{s}_i

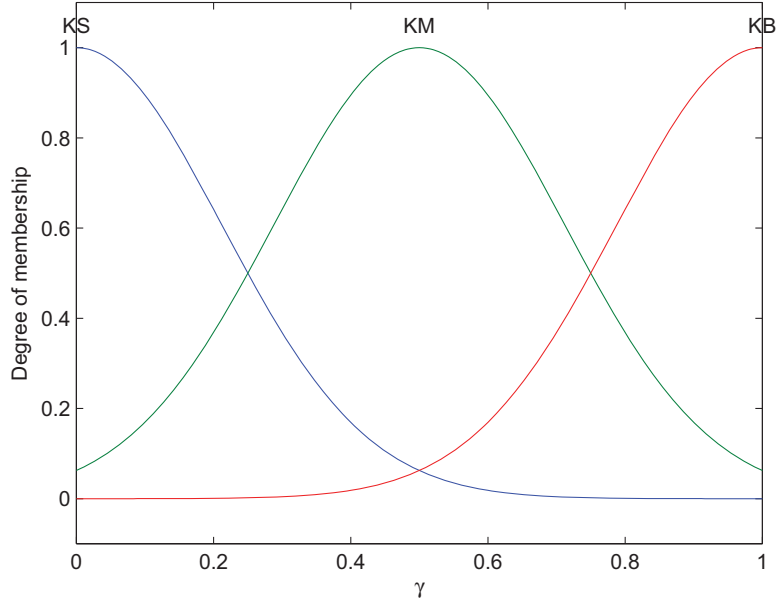


Figure 5.4: The membership function for the output γ_i

$$\tau = H [\dot{v}_c(t) + \beta e_c(t) + \gamma \cdot \text{erf}(s)] + E. \quad (5.10)$$

Proof Using the Lyapunov function: let $L = L_1 + L_2 + L_3$ such that $L(0) = L_1(0) = L_2(0) = L_3(0) = 0$ and $L, L_1, L_2, L_3 > 0$ for inputs other than 0, where

$$L_1 = \frac{1}{2} \sum_{i=1}^n \left(x_{ie}^2 + y_{ie}^2 + \frac{1 - \cos(\theta_{ie})}{k_{i2}} \right),$$

$$L_2 = \frac{1}{2} s^T(t) s(t) + \frac{1}{2} \sum_{i=1}^n (\gamma_i^2),$$

$$L_3 = E_c^T E_c.$$

Based on the Lyapunov theory, the system is stable if and only if $\dot{L} < 0$. After derivation we get

$$\dot{L}_1 = \sum_{i=1}^n \left(-k_{i1}x_{ie}^2 - \frac{k_{i3}v_{id}\sin^2(\theta_{ie})}{k_{i2}} \right),$$

$$\dot{L}_2 = s^T \dot{s}.$$

Now $\dot{L}_1 < 0$. And knowing that γ_i is always positive according to its membership function.

$$\dot{L}_2 = s^T [-\gamma \text{erf}(s) - \delta],$$

And the $\dot{L}_2 < 0$ can be guaranteed by selecting $\gamma > |\delta|$. which is obtained by the fuzzy rules

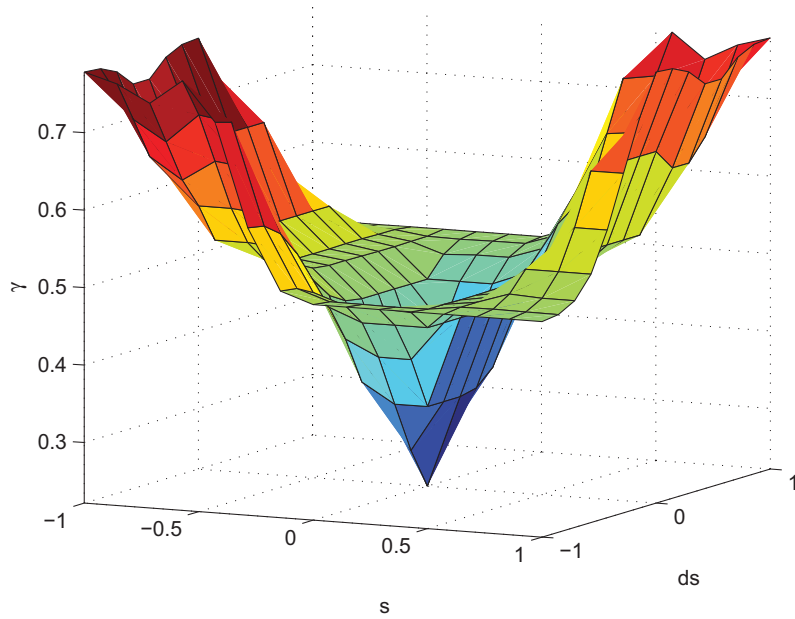


Figure 5.5: The fuzzy surface

The L_3 derivation is

$$\dot{L}_3 = E_c^T \dot{E}_c,$$

where $E_c = (c_d - c)$, $\dot{E}_c = (\dot{c}_d - \dot{c})$ and $\dot{c} = W E_c$. By substitution we arrive at:

$$\dot{E}_c = (\dot{c}_d - WE_c)$$

$$\text{Then } \dot{L}_3 = (E_c^T \dot{c}_d - E_c^T WE_c).$$

Knowing that $-E_c^T WE_c < -\lambda_{\min}(W)\|E_c\|^2$, the $\dot{L}_3 < 0$ provided that

$$\|E_c\|^T < \frac{\|\dot{c}_d\|}{\lambda_{\min}(W)}.$$

As a result, the derivative of the Lyapunov function is $\dot{L} < 0$ and the stability is proved.

5.2.4 Control diagram

In this subsection the control diagram is presented. Figure 5.6 shows the adaptive sliding mode controller.

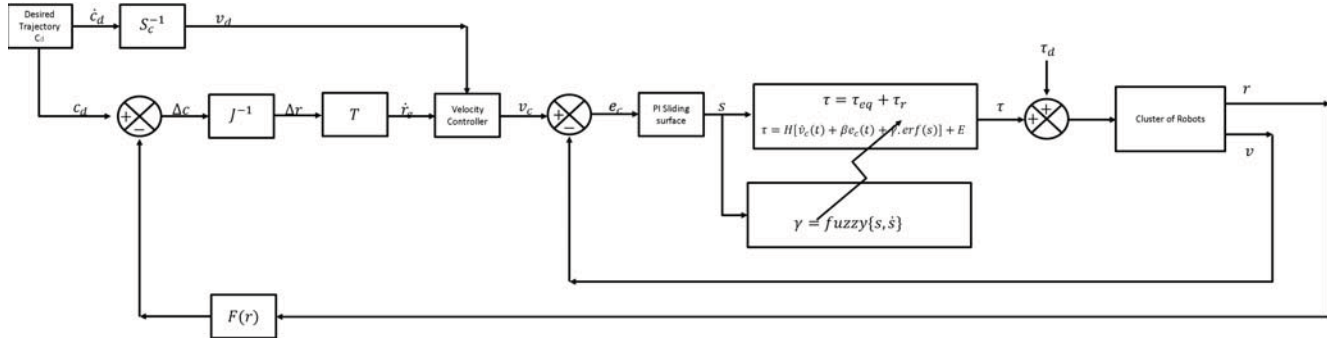


Figure 5.6: The proposed fuzzy adaptive SMC diagram

5.3 Simulation Results

In this section the simulation validation is presented, starting with the two-robot cluster simulation and ending with three-robot simulation. The simulation was done using matlab.

5.3.1 Two-robot simulation

Based on the control law established in Section 5.2.3, a simulation using MATLAB@ is implemented on the cluster of two mobile robots. The cluster space of two nonholonomic robots was used. The sinusoidal desired trajectory was applied as follows:

$$x_{cd} = 5 \cos(0.3t),$$

$$y_{cd} = 15 \sin(0.1t),$$

$$Q_{cd} = \tan^{-1} \left(\frac{dx_{cd}}{dy_{cd}} \right).$$

Let $K = [10, 2, 10, 2]$ and the initial position of the mobile robots are set as $(x_1, y_1, q_1) = (1, 0, 0)$ and $(x_2, y_2, q_2) = (0, 0, 0)$. The simulation results are shown in Figure (5.7) and Figure (5.8). Figure (5.7) illustrates the trajectory tracking result for the sinusoidal function. The actual trajectory reaches the desired line quickly. In Figure (5.8) a disturbance was introduced after 20 sec; the performances of the SMC, adaptive SMC and fuzzy adaptive SMC were compared to each other by comparing the tracking error in the cluster states x_c, y_c, Q_c and d_c , respectively. According to the simulation results, the sinusoidal trajectory tracking performs well, especially with the adaptive term. This validates the sliding control algorithm by simulation.

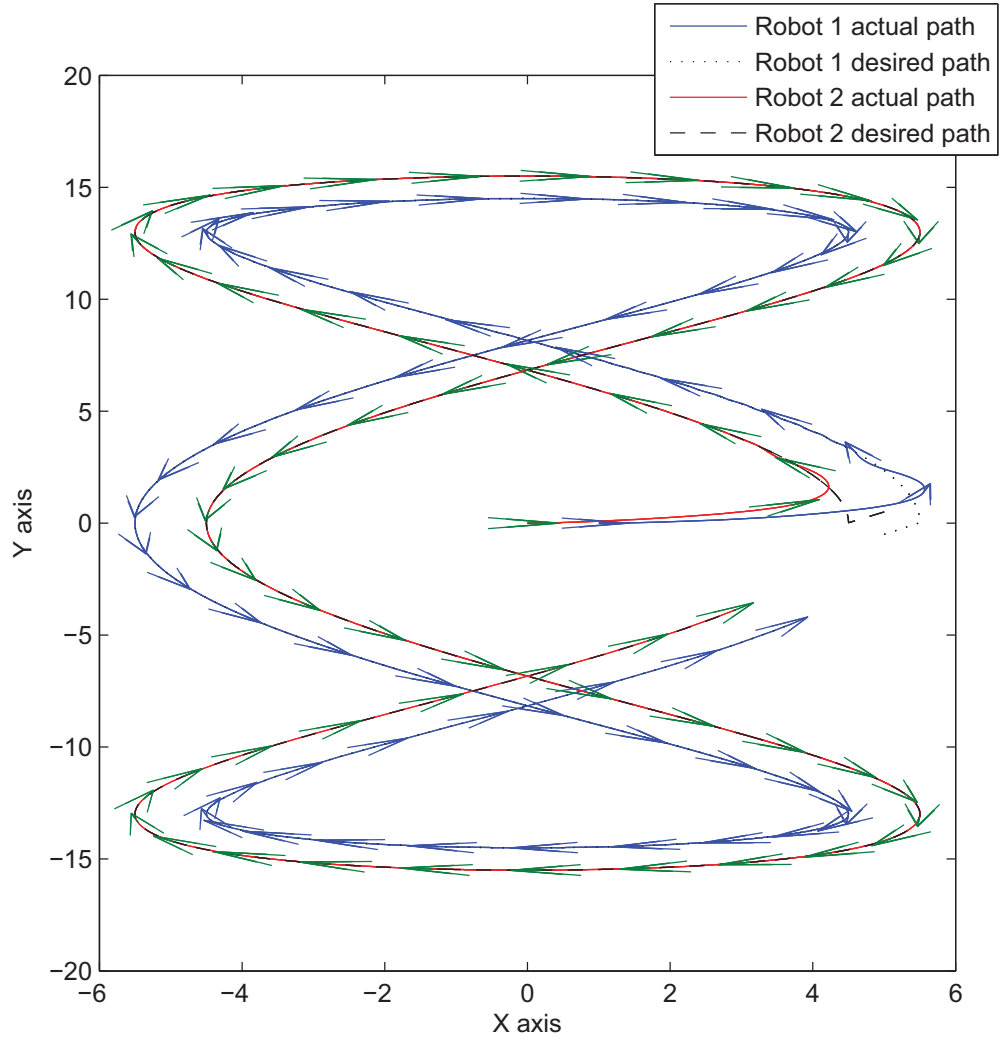


Figure 5.7: Fuzzy Adaptive SMC control path tracking profile

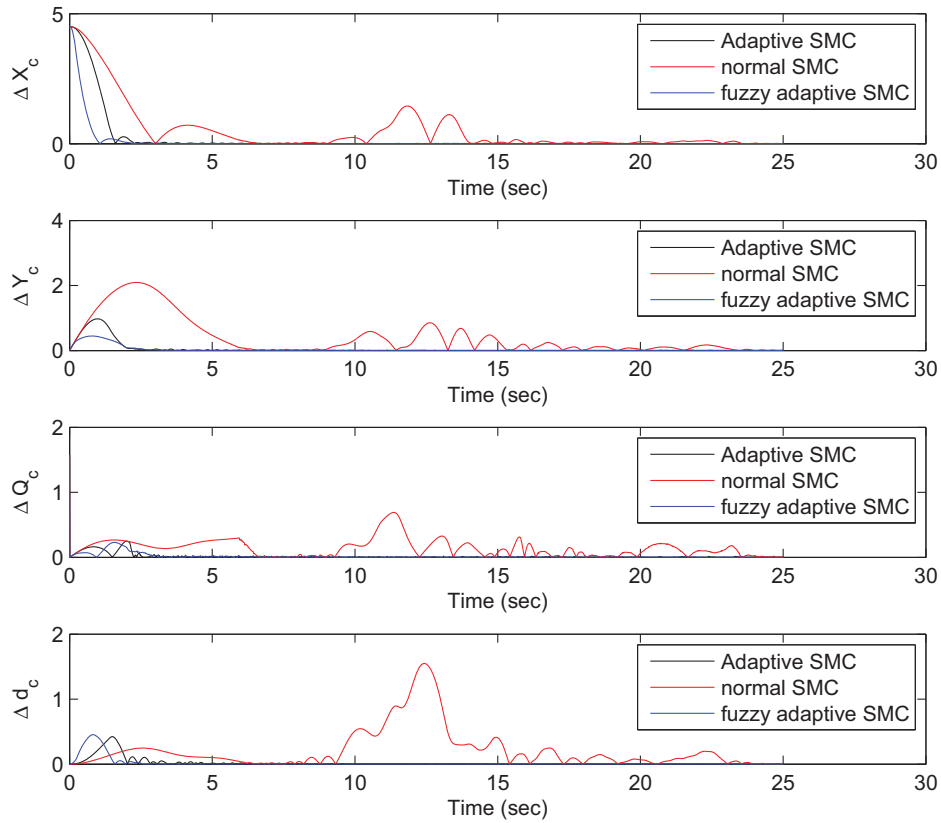


Figure 5.8: Comparison between Adaptive SMC and standard SMC and fuzzy adaptive SMC with disturbance injected after 10 sec

5.3.2 Simulation results of 3-robot system

In this subsection a three-robot cluster was simulated with the same reference trajectory presented in subsection 5.3.1. The selected cluster spaces are $(x_c, y_c, Q_c, \beta_c, q_c, p_c, q_1, q_2, q_3)$; see Figure (5.9) and for more details refer to [17]. The simulation results (see Figure (5.10)) show that the fuzzy SMC is at least 30% faster than the adaptive SMC.

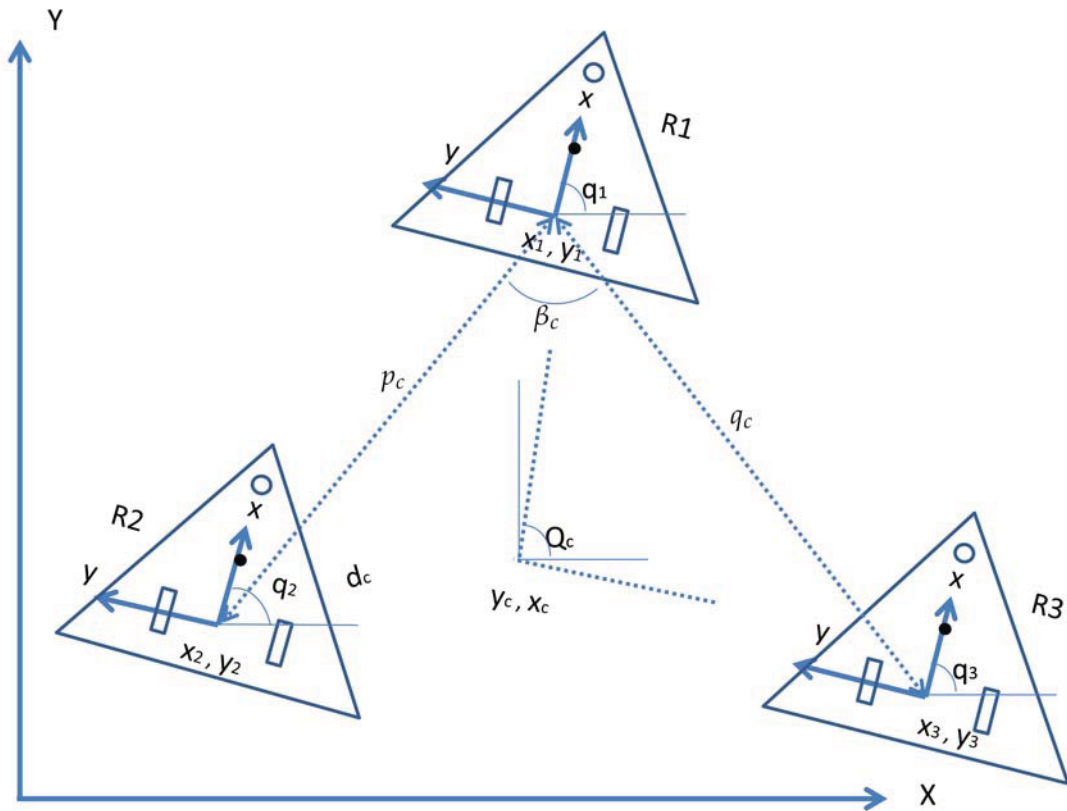


Figure 5.9: 3-robot system configuration

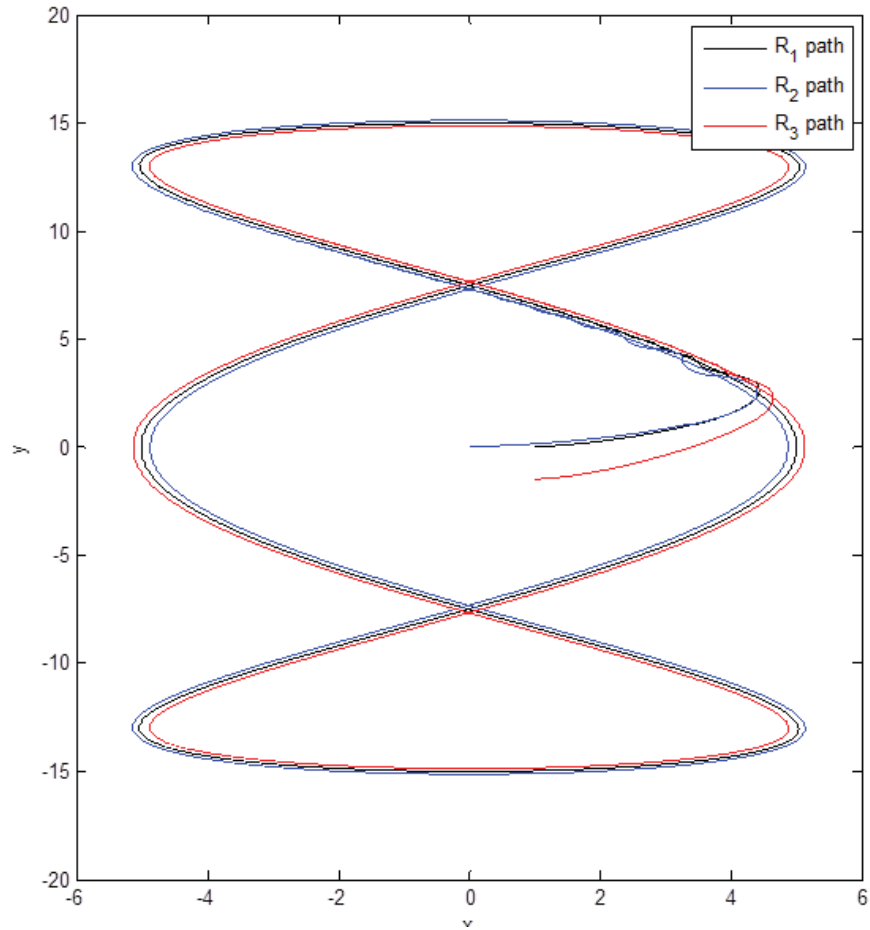


Figure 5.10: Adaptive SMC control path tracking profile

5.4 Experimental application

Lego EV3 WMRs were used in the experiments to validate and implement the control strategy. The WMRs (see Figure 5.11) are equipped with a 32-bit, 48 Mhz ARM9 CPU with 16MB flash memory and 64MB RAM, Bluetooth and wi-fi transceivers, and two servo motors with encoders with 1 degrees of resolution. A PC interface with the SIMULINK program is also required to transmit the control signal by means of wi-fi protocol. SIMULINK has a powerful feature

called External Mode. This feature is useful for on-line monitoring and tuning of the EV3 WMR's controller. A two-level control structure is used: a high-level and low-level. The high-level controller is the sliding mode controller, operating in the central PC, which sends and receives commands /data to and from the WMRs low-level controller. The low-level controller is a PID inner loop for controlling wheel speeds.

Based on the proposed control algorithm, the central PC receives the location feedback from each robot; then the PC calculates the error and control signals and sends the velocity commands to each robot. The low-level controller on the robot receives the commands from the PC and relays these signals to the motors. The encoders provide measurements for the feedback. The actual time of a one-loop process depends on the robot sampling time (set to 25ms) plus the wi-fi delay time, which it is dependent on the computer speed and network usage. The robot localization is achieved by using the encoders only. The WMRs use EV3 servo motors that have a gear reduction mechanism in order to increase the torque and decrease the maximum output speed. However, this gear mechanism has a backlash issue, which introduces a nonlinear behavior due to small gaps between the mating gear teeth. Once the servo motor changes its direction the backlash effect occurs, causing the servo to have a certain rotation without being translated to actual wheel rotation. This issue can be mitigated by adding backlash compensation. Thus, when the servo motor changes its rotation direction a certain value found by trial and error is subtracted from the encoder reading.

Practical issues The output from the centralized controller is the desired wheel's torque, but due to the braking behavior of the EV3 servo motor, (when the motor has zero input) the wheel will stop abruptly. This braking is an additional variable that is not considered when developing the sliding mode controller. To improve the response the following solution has been used: 1) add an integrator filter to the controller output torque, and the output signal of this filter is then considered to be a velocity command; 2) instead of using a voltage or power signal to control the servo motor, the velocity command is sent to an inner speed loop controller. Accordingly, the servo motor will follow a torque equal to the sliding mode torque multiplied by some design constant, which is found through trial and error. For more, see Figure (5.11), where W_{cmd} is the velocity command.



Figure 5.11: Using an integrator to change a fuzzy SMC command to a velocity command calculation and the inner loop diagram

5.4.1 Experiment test

Two WMRs move from their initial positions (x, y, q) , which for the first robot is $(0.2, 0, \frac{\pi}{2})$ and $(0, 0, \frac{\pi}{2})$ for the second robot. The desired path is $Q_c = \frac{\pi}{2}, x_c = 0.3, y_c = 0.1t$ and $d_c = 0.25$ in the cluster space which is equivalent to $x_1 = 0.55, y_1 = 0.1t, q_1 = \frac{\pi}{2}, x_2 = 0.05, y_2 = 0.1t$ and $q_2 = \frac{\pi}{2}$ in the robot space, where t is the time (see Figure 5.12). The dynamic parameters are assumed to be nominal values. So the robot mass is $0.5kg$, the robot inertia is $0.0025 kg.m^2$, the wheel diameter is $56mm$ and the distance between the wheels is $11.8cm$.

The experimental results are shown in Figures (5.13 and 5.14) . Figure (5.13) illustrates the trajectory tracking result for a line trajectory with fuzzy adaptive SMC. The actual trajectory reaches the desired line quickly. Figure (5.14) shows the comparison between the fuzzy adaptive SMC and the adaptive SMC based on the cluster states tracking x_c, y_c, Q_c and d_c , respectively. According to the experimental results for the line trajectory tracking, the trajectory can remain stable and robust despite uncertainty in the robot's inertia and mass. and due to the fast fuzzy logic controller tuning the performance of the adaptive fuzzy SMC is faster than the standard adaptive SMC. This validates the proposed algorithms experimentally.

5.5 Chapter summary

This chapter proposed a model-based multi nonholonomic robot controller in the cluster space by developing a fuzzy adaptive sliding mode control algorithm. This

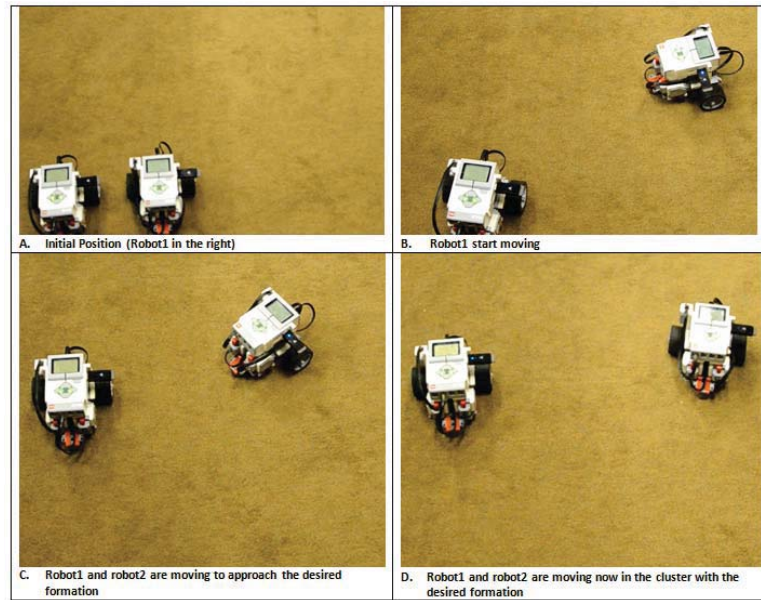


Figure 5.12: Experimental cluster space control on a 2-robot system

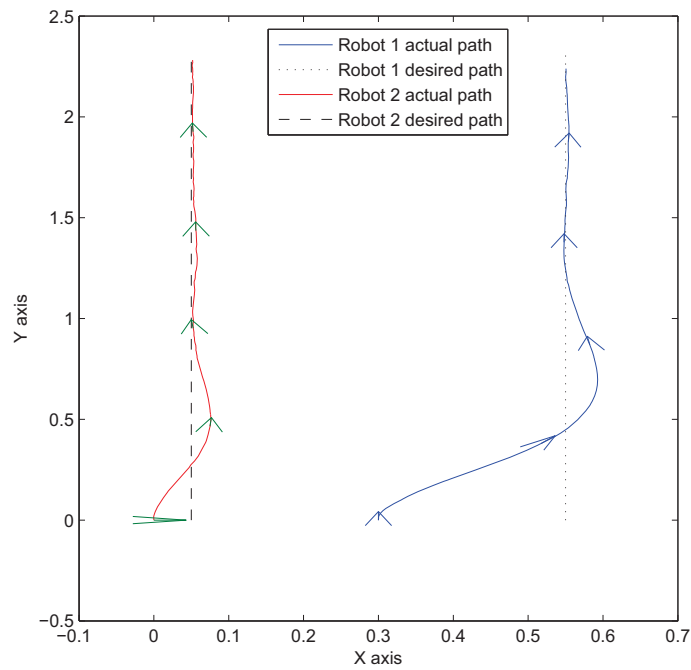


Figure 5.13: Experimental profile tracking results of a two robot cluster with fuzzy adaptive SMC

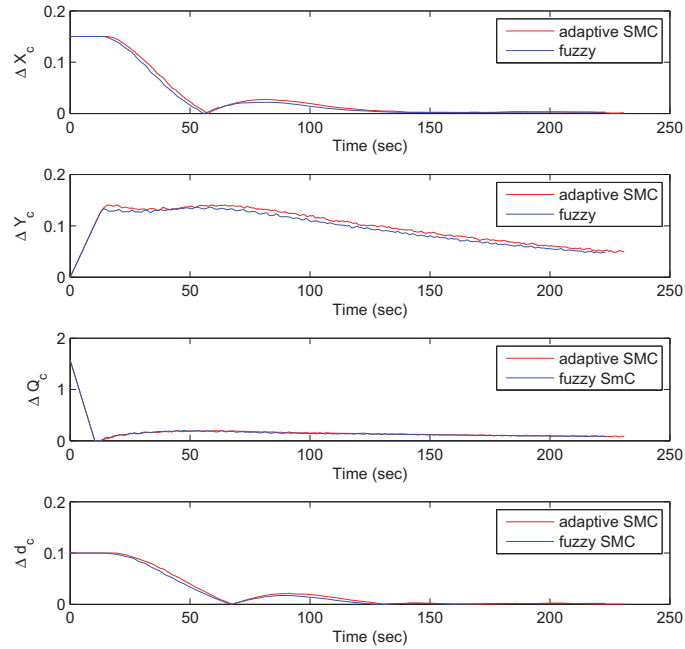


Figure 5.14: A comparison between the tracking errors with adaptive SMC and fuzzy adaptive SMC

includes the design of fuzzy logic algorithm and of the control law. According to the simulation and the experimental results, the proposed fuzzy adaptive sliding mode control is an important method to deal with a cluster of multi-robots and it is superior to the adaptive sliding mode controller [73] in tuning speed and performance, (as well as uncertainties and nonlinearities which exist). Future work may investigate clusters with a larger number of heterogeneous robots and apply intelligent methods like neural networks and fuzzy logic to overcome the singularities in the cluster dynamics.

CHAPTER 6

MPC CONTROLLER FOR ACTUATOR SATURATION ISSUE WITH CLUSTER SPACE

6.1 Introduction

In this work, the MPC is used in order to achieve the maximum usage of the velocity domain for stabilizing the multi-robot formation shape and then carrying out the target following task. This is done by giving more weight to the cluster formation shape states over the target following states. The chapter reports an experimental implementation of the proposed scheme on a real ground robot connected to a central controller using a wi-fi network.

The remainder of this chapter is organized as follows: section 6.2, the robot velocity saturation is discussed. Then the model-based controller is proposed. The

simulation results are discussed in Section 6.4 and the experimental validation and results are presented in section 6.5. And ending up with concludes the chapter.

6.2 Differential Robot velocity constraints

6.2.1 Diamond shape velocity constraint

As shown in Figure 2.1, our nonholonomic robots' differential drive type is considered in this work. Let w_r and w_l represent the right and left wheel angular velocities. It is assumed that the wheels velocities are subjected to $|w_{r/l}| \leq \alpha$ with α being the angular velocity limit. The nonholonomic differential drive robot has the heading velocity v and the angular velocity w which are related to the wheels' angular velocities by

$$v = (w_r + w_l)/2, \quad w = (w_r - w_l)/(2\vartheta) \quad (6.1)$$

Accordingly the heading and the angular velocities are bounded by

$$|v|/\alpha + |\vartheta w|\alpha \leq 1 \quad (6.2)$$

where $\vartheta = L/2$ the half of the distance between the wheels. Figure 6.1a illustrates the velocities domain in a diamond shape.

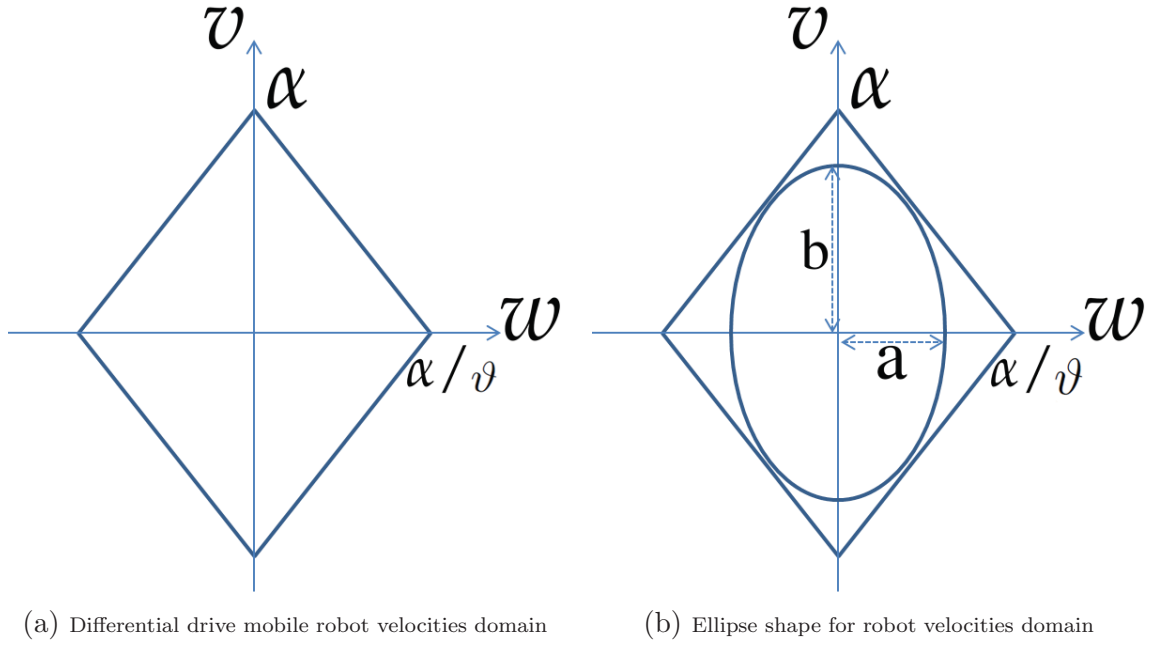


Figure 6.1: Differential drive velocities domain's

However the diamond shape has sharp corners and the dv/dw is undefined on these sharp corners, which may affect the optimization search. consequently for simplicity and for improving the optimization process speed an ellipse shape equation (6.3) is proposed instead of the diamond shape. See figure 6.1b

$$(v/b)^2 + (w/a)^2 = 1 \quad (6.3)$$

where b is a constant that is propotional with the max heading velocity, $a = L * b$ and L is defined in figure 2.1

6.3 Controller design

In this section, the MPC controller is presented as follows: first the error signal calculation, then kinematic controller design, followed by the MPC and finally the controller diagram. As an example this development is done for a two-robot cluster, while the same procedure is done for a cluster with any number of robots.

6.3.1 The cluster profile errors

The formation error in the cluster spaces is found as follows; starting with defining $c = (x_c, y_c, Q_c, d_c, q_1, q_2)$, $c_d = (x_{cd}, y_{cd}, Q_{cd}, d_{cd}, q_{1d}, q_{2d})$, where c_d is the desired cluster space and c is the actual cluster spaces. The error signal is $\Delta c = W(c_d - c)$ where W is a positive weighting diagonal matrix. The robot space commands are transformed from the cluster space signals to robot space signals by multiplying them with the inverse of the Jacobian matrix, as $\overline{\Delta r} = J^{-1} \Delta c$ where $\overline{\Delta r} = [\Delta x_1, \Delta y_1, \Delta \theta_1, \Delta x_2, \Delta y_2, \Delta \theta_2]^T$ and $\overline{\Delta r}_i = [\Delta x_i, \Delta y_i, \Delta \theta_i]^T$, in order to deal with the nonholonomic constraints the robot space commands are modified as follows; $\Delta r = [\Delta x_1, \Delta y_1, \theta_{1e}, \Delta x_2, \Delta y_2, \theta_{2e}]^T$ and $\Delta r_i = [\Delta x_i, \Delta y_i, \theta_{ie}]^T$ where $\theta_{ie} = \tan_2^{-1}(\frac{\Delta y_i}{\Delta x_i}) - q_i$ then a transformation of the robot commands from a global frame to a robot frame is achieved by using the following rotational transformation;

$$r_{ie} = \begin{bmatrix} x_{ie} \\ y_{ie} \\ \theta_{ie} \end{bmatrix} = \begin{bmatrix} \cos(q_i) & \sin(q_i) & 0 \\ -\sin(q_i) & \cos(q_i) & 0 \\ 0 & 0 & 1 \end{bmatrix} \Delta r_i.$$

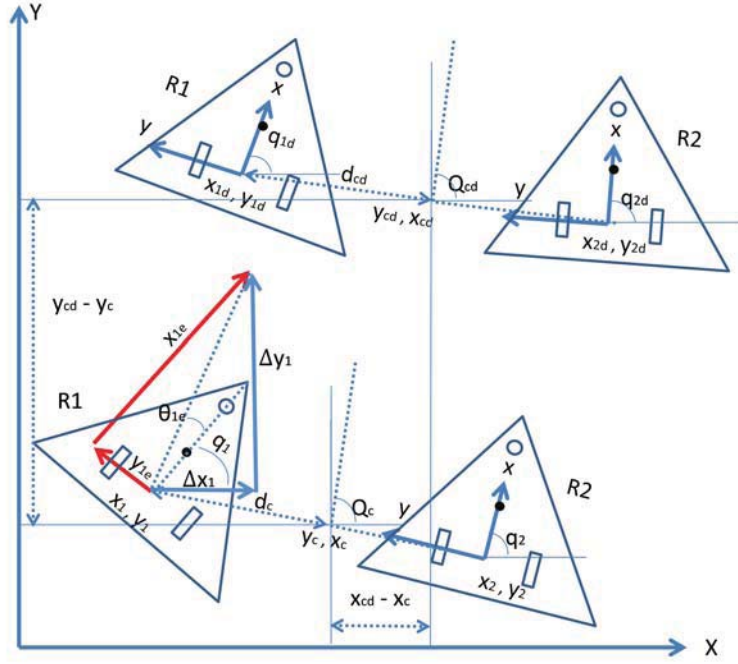


Figure 6.2: Cluster location errors

where $(x_{ie}, y_{ie}$ and θ_{ie}) are shown in Figure 6.2. It was found in [60, 69, 70] that the derivative of the robot profile errors can be found by equation 6.4.

$$\dot{r}_{ie} = \begin{bmatrix} \dot{x}_{ie} \\ \dot{y}_{ie} \\ \dot{\theta}_{ie} \end{bmatrix} = \begin{bmatrix} y_{ie}w_i - v_i + v_{id} \cos(\theta_{ie}) \\ -x_{ie}w_i + v_{id} \sin(\theta_{ie}) \\ w_{id} - w_i \end{bmatrix}. \quad (6.4)$$

6.3.2 Kinematic Controller

In this section, a kinematic-based controller is presented. Starting, from the classical kinematic controller given by [70, 73, 79], we get the following velocity

controller in equation 6.5.

$$v_c = \begin{bmatrix} v_{1c} \\ w_{1c} \\ \vdots \\ v_{nc} \\ w_{nc} \end{bmatrix} = \begin{bmatrix} v_{1d} \cos(\theta_{1e}) + k_{11}x_{1e} \\ w_{1d} + k_{12}v_{1d}y_{1e} + k_{13}v_{1d} \sin(\theta_{1e}) \\ \vdots \\ v_{nd} \cos(\theta_{ne}) + k_{n1}x_{ne} \\ w_{nd} + k_{n2}v_{nd}y_{ne} + k_{n3}v_{nd} \sin(\theta_{ne}) \end{bmatrix}, \quad (6.5)$$

where $k_{i1}, k_{i2}, k_{i3} > 0, i|i = 1, \dots, n$. Thus, the error in the robots' kinematics is defined as:

$$e_c(t) = [e_{c1}(t), e_{c2}(t), \dots, e_{c2.n}]^T = v_c(t) - v(t),$$

$$\dot{e}_c(t) = \dot{v}_c(t) - \dot{v}(t),$$

where

$$v = \begin{bmatrix} v_1 \\ w_1 \\ \vdots \\ v_n \\ w_n \end{bmatrix}.$$

6.3.3 MPC Controller

The cluster error signal is $E_c = c_d - c$, however, due to the nonholonomic constraints the cluster angular errors $q_{id} - q_i$ are replaced with θ_{ie} that represent the stabilizing angular errors for the nonholonomic model. As a result of this, the

new cluster error matrix is

$$Ec_n = (Ec, \theta_{1e}, \theta_{2e})$$

$$\text{minimize } H = \sum_{\tau=t}^{t+T} |Ec_n(X(\tau), U(\tau))|. \quad (6.6)$$

Subjected to $c(t+1) = f(X(t), U(t))$ and equation (6.3), where $X(t) = [c, \dot{c}]$ and $U(t) = [Ec(t), \dot{E}c(t)]$, and T is the time interval. So the optimization function will minimize H based on the selection of the control command $v_c(t)$ shown in equation 6.5.

Stability Proof :

Using the Lyapunov function: let $L = L_1 + L_2$ such that $L(0) = L_1(0) = L_2(0) = 0$ and $L, L_1, L_2 > 0$ for inputs other than 0, where L_1 is the Lyapunov function for the robot level controller, and the L_2 is the Lyapunov function for the cluster controller.

$$L_1 = \frac{1}{2} \sum_{i=1}^n \left(x_{ie}^2 + y_{ie}^2 + \frac{1 - \cos(\theta_{ie})}{k_{i2}} \right),$$

$$L_2 = E_c^T E_c.$$

And the command will be generated in the cluster space controller then transformed to robot space controller by using the Jacobian matrix such that $\dot{r}_e = J^{-1} \Delta c$.

Based on the Lyapunov theory, the system is stable if and only if $\dot{L} < 0$. After derivation we get

$$\dot{L}_1 = \sum_{i=1}^n \left(-k_{i1} x_{ie}^2 - \frac{k_{i3} v_{id} \sin^2(\theta_{ie})}{k_{i2}} \right),$$

Now $\dot{L}_1 \leq 0$ provided that $v_{id} > 0$. The L_2 derivation is

$$\dot{L}_2 = E_c^T \dot{E}_c,$$

where $E_c = (c_d - c)$, $\dot{E}_c = (\dot{c}_d - \dot{c})$ and $\dot{c} = W E_c$. By substitution we arrive at:

$$\dot{E}_c = (\dot{c}_d - W E_c)$$

$$\text{Then } \dot{L}_2 = (E_c^T \dot{c}_d - E_c^T W E_c).$$

Knowing that $-E_c^T W E_c < -\lambda_{\min}(W) \|E_c\|^2$, the $\dot{L}_2 < 0$ provided that $\|E_c\|^T < \frac{\|\dot{c}_d\|}{\lambda_{\min}(W)}$. And the Jacobian matrix $J(t)$ is not singular at t

As a result, the derivative of the Lyapunov function is $\dot{L} < 0$ and this completes the proof.

6.3.4 Control diagram

In this subsection the control diagram is presented. Figure 6.3 shows the MPC controller.

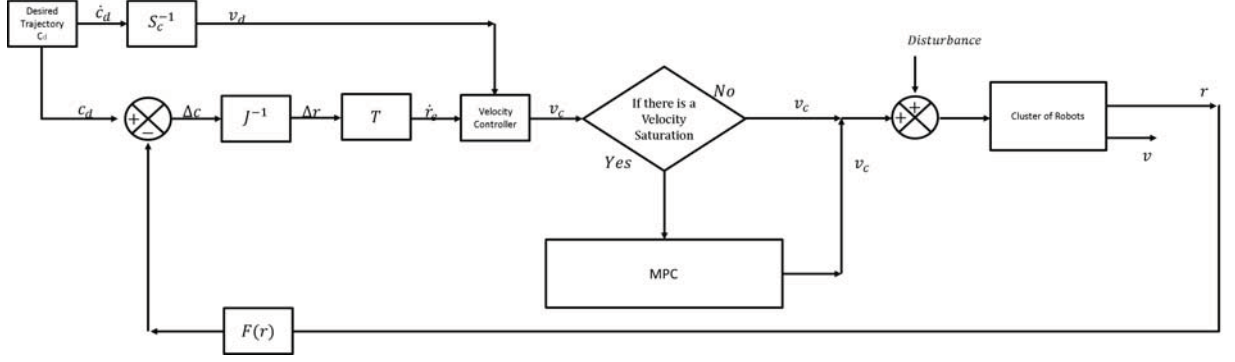


Figure 6.3: The proposed MPC diagram

6.4 Simulation Results

6.4.1 Two robot simulation

Based on the control law established in Section 6.3.4, a simulation using MATLAB@ is implemented on the cluster of two mobile robots. The cluster space of two nonholonomic robots was used. The desired sinusoidal trajectory was applied as follows:

$$x_{cd} = 10 \cos(0.03t),$$

$$y_{cd} = 30 \sin(0.01t),$$

$$Q_{cd} = \tan^{-1} \left(\frac{dx_{cd}}{dy_{cd}} \right).$$

Let $k_j i = [0.1, 5, 0.1, 0.1, 5, 0.1]$ and the initial position of the mobile robots are set as $(x_1, y_1, q_1) = (1, 0, 0)$ and $(x_2, y_2, q_2) = (0, 0, 0)$. The simulation results are shown in Figure (6.4) and Figure (6.5). Figure (6.4) illustrates the trajectory tracking result for the sinusoidal function. The actual trajectory with the MPC reaches the desired line quickly, however, the robots were late when the profile in-

cludes turns. In Figure (6.5) a comparison is made with three cases. The first one shows the profile error of the cluster response with a normal kinematic controller and zero velocity saturation, the second one presents the response with velocity saturation, and the third one shows the proposed MPC with the existence of the saturation. The results of the MPC and the kinematic controller are compared to each other based on the tracking error in the cluster states x_c, y_c, Q_c and d_c , respectively. This shows an improvement with the MPC over the kinematic controller with the existence of the saturation. According to the simulation results, the sinusoidal trajectory tracking shows better response with MPC in the saturation case compared with the kinematic controller. This simulation validates the MPC control algorithm.

6.5 Experimental application

Lego EV3 WMRs were used in the experiments to validate and implement the control strategy. The WMRs (see Figure 6.6) are equipped with a 32-bit, 48 Mhz ARM9 CPU with 16MB flash memory and 64MB RAM, Bluetooth and wi-fi transceivers, and two servo motors with encoders with 1 degree of resolution. A PC interface with the SIMULINK program is also required to transmit the control signal by means of wi-fi protocol. SIMULINK has a powerful feature called 'External Mode'. This feature is useful for on-line monitoring and tuning of the EV3 WMR's controller. A two-level control structure is used: high-level and low-level. The high-level controller is the sliding mode controller, performed in

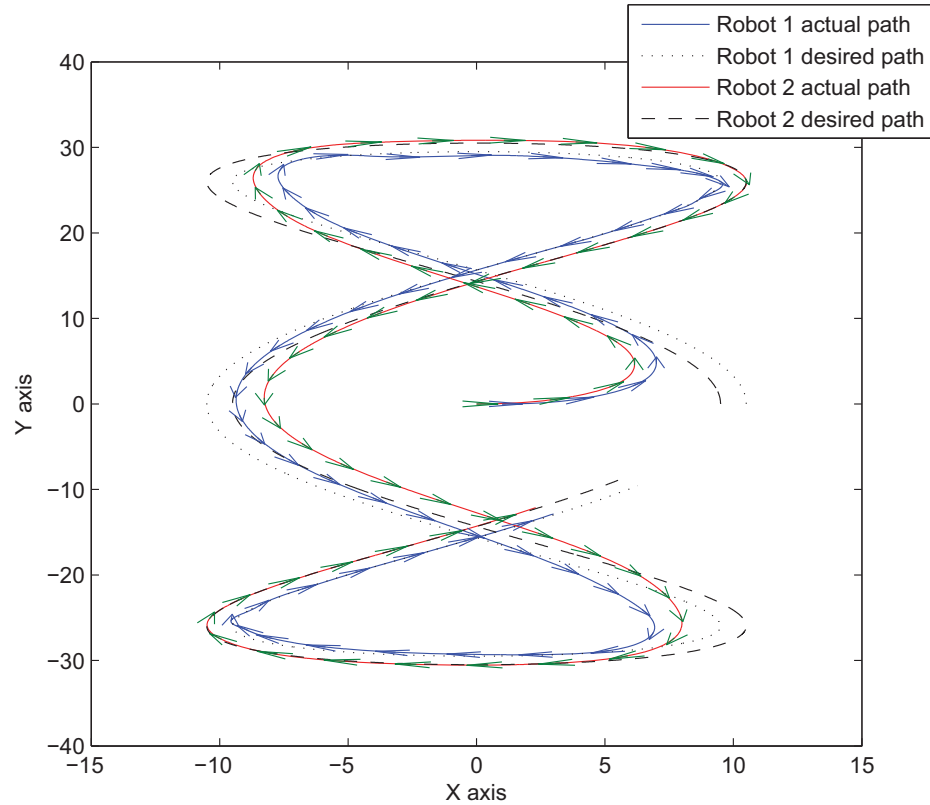


Figure 6.4: MPC control path tracking profile

the central PC, which sends and receives commands /data to and from the WMRs low-level controller. The low-level controller is a PID inner loop for controlling wheel speeds. The 2014 version MatLab was used. However, as far as the authors know, the newer versions of MatLab don't allow for the simultaneous operation of the multi-robot with the simulink.

Based on the proposed control algorithm, the central PC receives the location feedback from each robot; then the PC calculates the error and control signals and sends the velocity commands to each robot. The low-level controller on the robot receives the commands from the PC and relays these signals to the motors. The encoders provide measurements for the feedback. The actual time of a one-loop

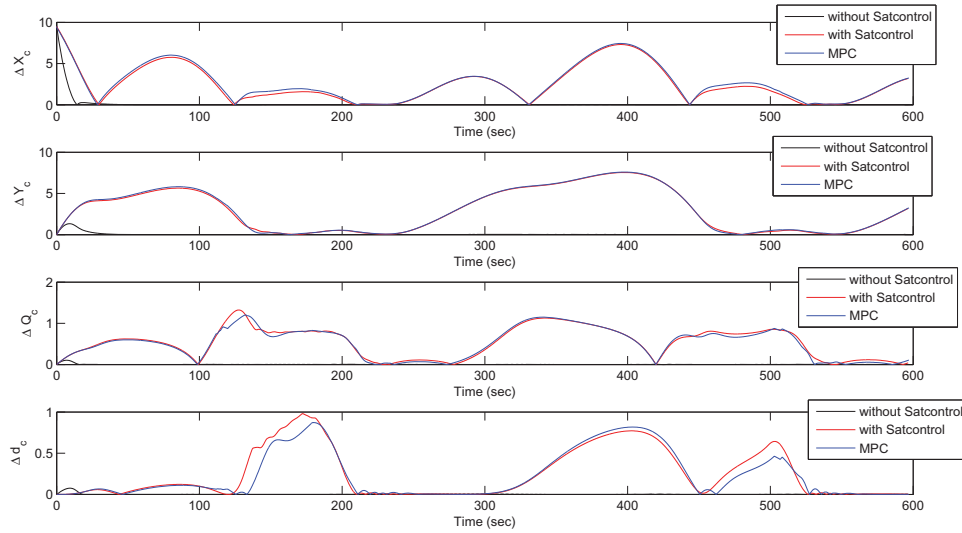


Figure 6.5: Comparison between kinematic with and without saturation and MPC with saturation

process depends on the robot sampling time (set to 50ms) plus the wi-fi delay time, which is dependent on the computer speed and network usage. The robot localization is achieved by using the encoders only. The WMRs use EV3 servo motors that have a gear reduction mechanism in order to increase the torque and decrease the maximum output speed. However, this gear mechanism has a backlash issue, which introduces a nonlinear behavior due to small gaps between the mating gear teeth. Once the servo motor changes its direction the backlash effect occurs, causing the servo to have a certain degree of rotation without this being translated to actual wheel rotation. This issue can be mitigated by adding backlash compensation. Thus, when the servo motor changes its rotation direction a certain value is subtracted from the encoder reading. For more details about the practical issues please refer to the author's work [73]



Figure 6.6: MPC velocity error is filtered using an integrator to get a velocity command that is sent to the robot inner loop diagram

6.5.1 Experiment test

Two WMRs move from their initial positions (x, y, q) , which for the first robot is $(0.2, 0, \frac{\pi}{2})$ and $(0, 0, \frac{\pi}{2})$ for the second robot. The desired path is $Q_c = \frac{\pi}{2}, x_c = 0.3, y_c = 0.01t$ and $d_c = 0.25$ in the cluster space which is equivalent to $x_1 = 0.55, y_1 = 0.01t, q_1 = \frac{\pi}{2}, x_2 = 0.05, y_2 = 0.01t$ and $q_2 = \frac{\pi}{2}$ in the robot space, where t is the time (see Figure 6.7). The dynamic parameters are assumed to be nominal values. The robot mass is $0.5kg$, the robot inertia is $0.0025 kg.m^2$, the wheel diameter is $56mm$ and the distance between the wheels is $11.8cm$.

In order to minimize the processing time during the practical experiment, the developed MPC predicts only one step ahead and calculates the optimum controller gain accordingly. The number of controller variables is four which helps to

decrease the processing time. The experimental results are shown in Figure (??). Figure (6.8) illustrates the trajectory tracking result for a line trajectory with a kinematic controller with out velocity saturation. The actual trajectory reaches the desired line quickly; however, Figure (6.9) shows the actual trajectory of the cluster driven by the kinematic controller with velocity saturation. Figure 6.10 also shows the actual trajectory of the cluster with the MPC controller and it is very apparent that the MPC shows a better formation recovery than the classical controller. Figure (6.11) shows the comparison between these three controller based on the errors in the cluster states tracking x_c, y_c, Q_c and d_c , respectively. The MPC gives priority to the formation recovery over the target following task. Due to this the MPC is better in the case of saturation , for example the error in x axis converge 50% faster with MPC compared with the kinematic controller . This validates the proposed algorithms experimentally.

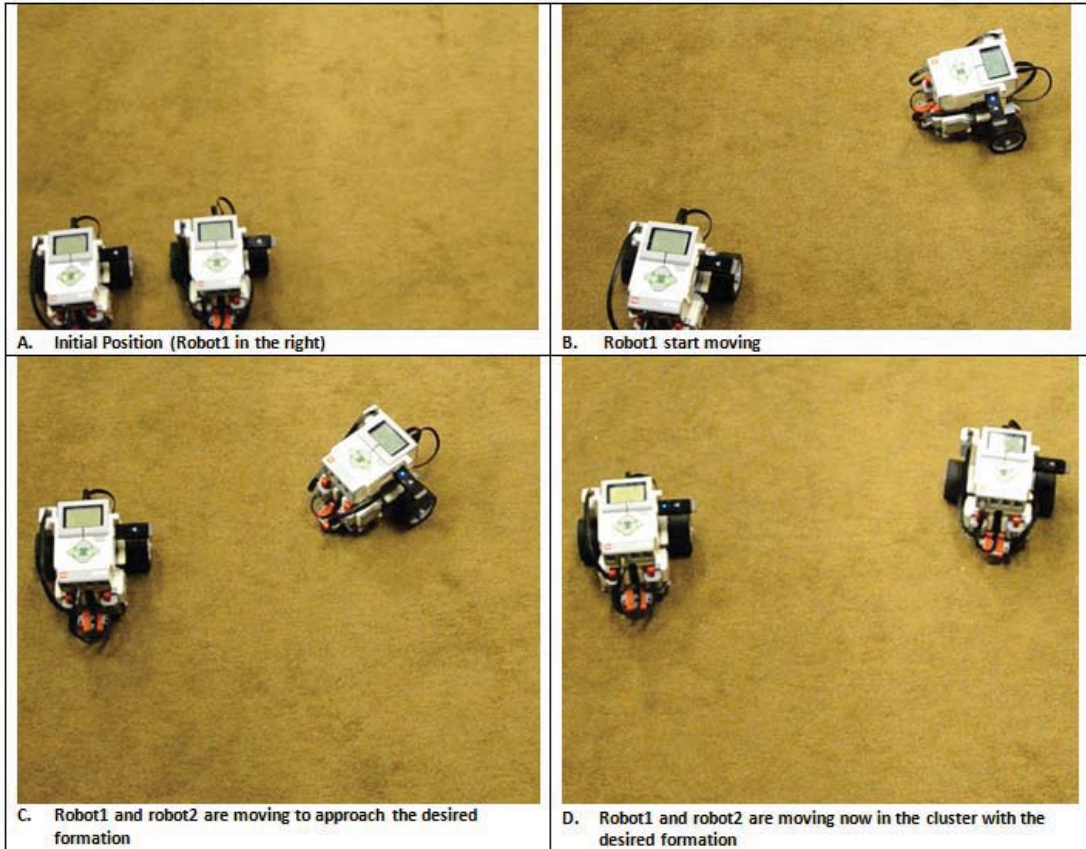


Figure 6.7: Experimental cluster space control on a 2-robot system

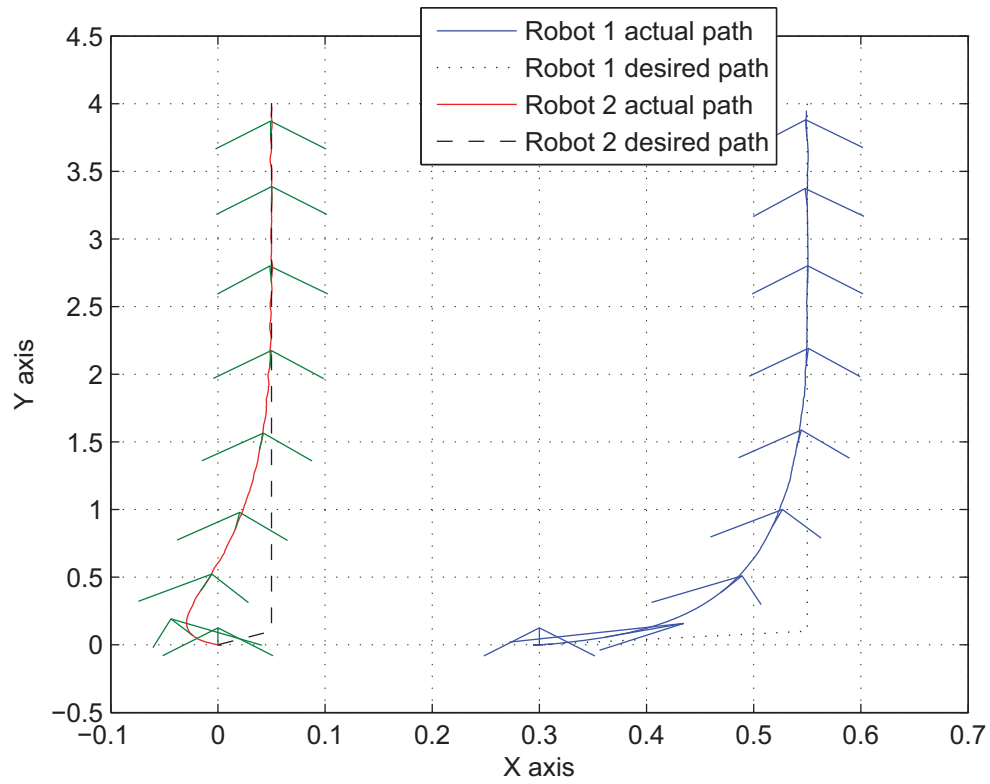


Figure 6.8: Experimental movement trajectory of a cluster of two robot with Kinimatic controller and without velocity saturation

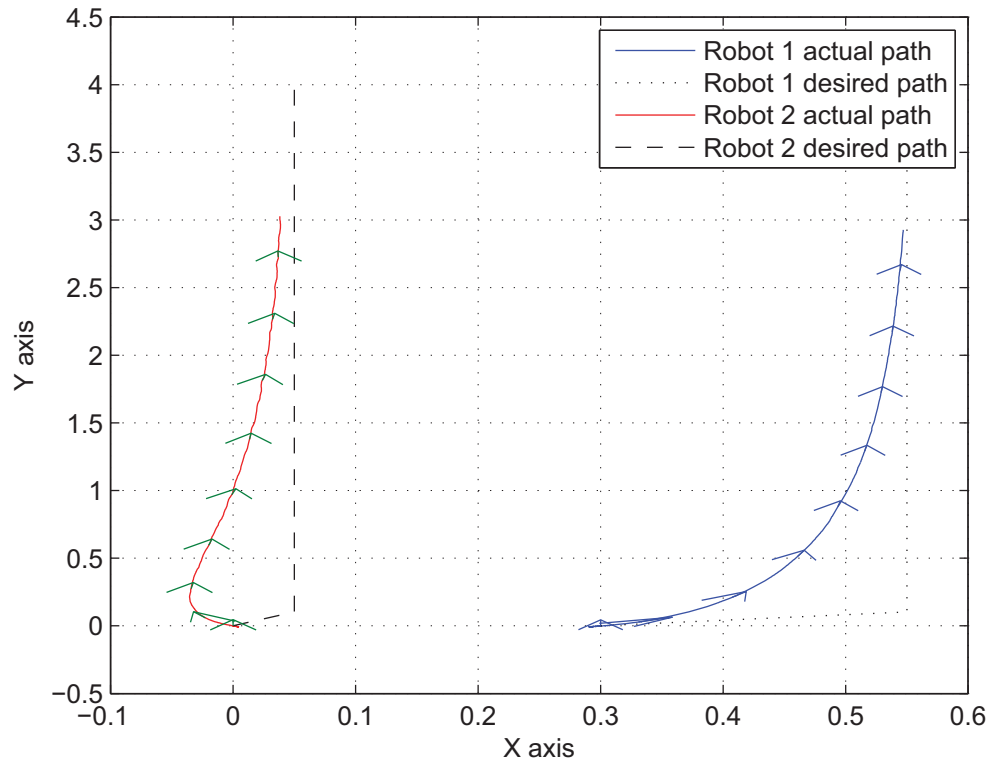


Figure 6.9: Experimental movement trajectory of a cluster of two robot with Kinimatic controller and robot velocity saturation

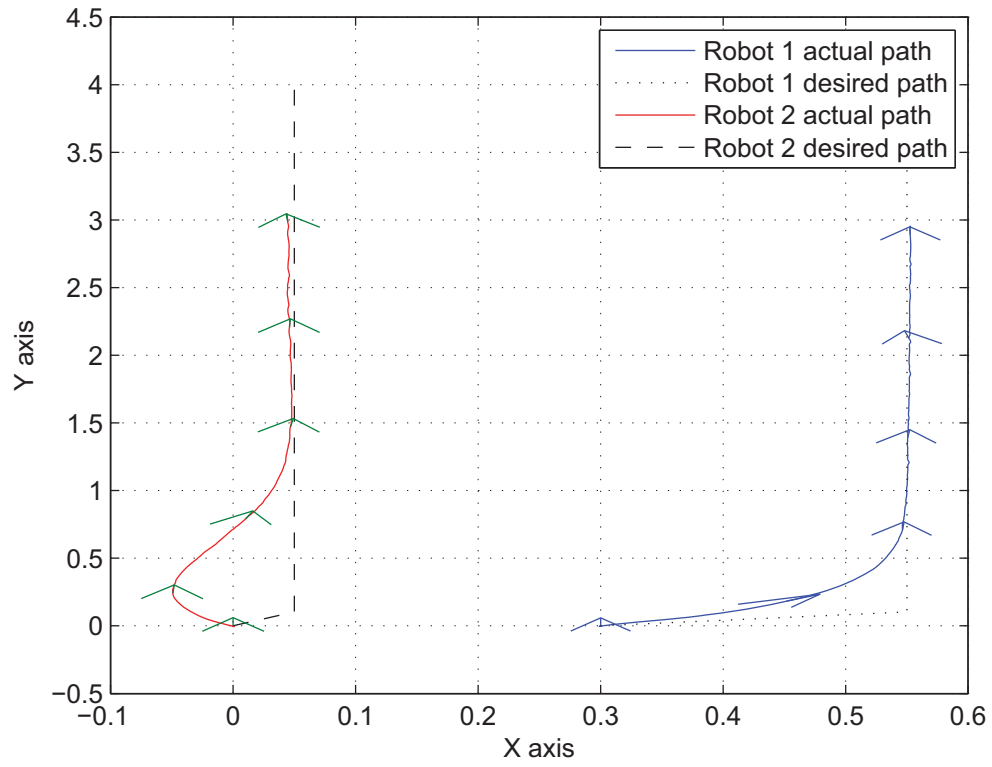


Figure 6.10: Experimental movement trajectory of a cluster of two robot with MPC and robot velocity saturation

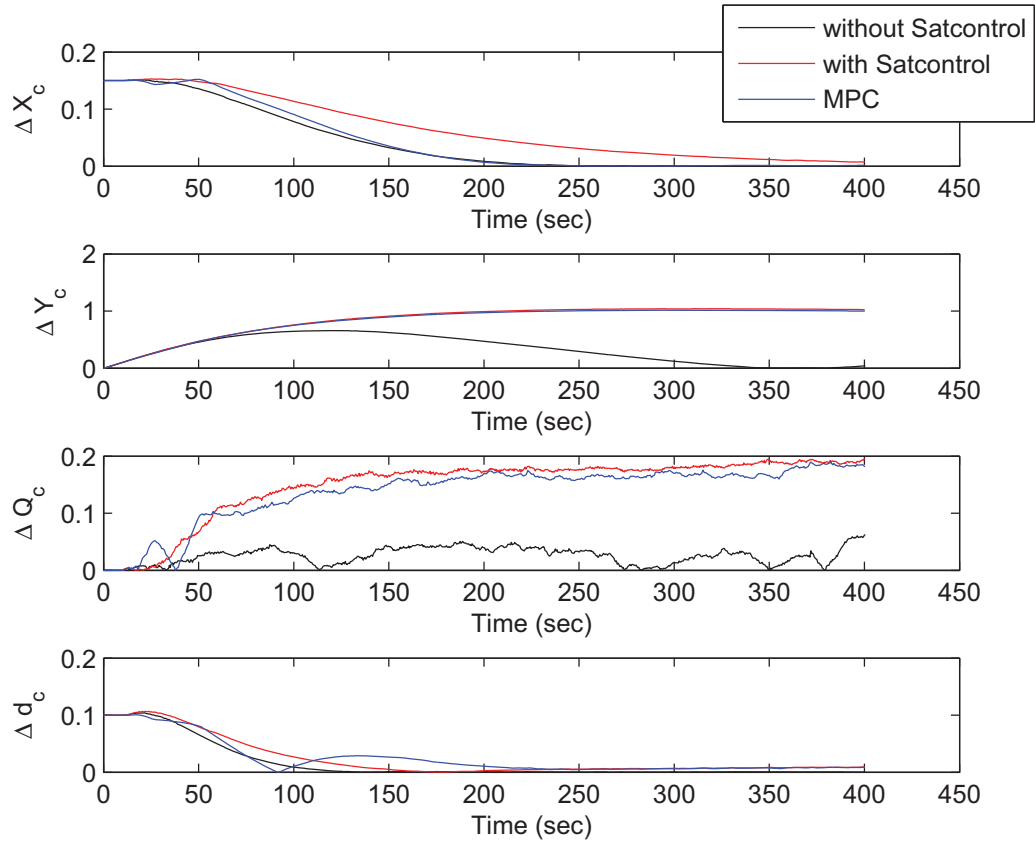


Figure 6.11: The experimental trajectory errors of the cluster dynamics states x_c, y_c, Q_c and d_c , and a comparison between the kinematic with/without velocity saturation and MPC with velocity saturation

6.6 Chapter summary

This chapter proposed a model predictive controller for a multi-differential drive robot in the cluster space. This controller overcomes the velocity saturation issue, using an optimization algorithm the controller weights are tuned to minimize the cluster formation and target following tasks. According to the simulation and experimental results, the proposed MPC is an important method to deal with a cluster of multi-robots in which the velocity's saturation and model nonlinearities

exist. In spite of an effective velocity saturation, the robots' cluster shows that the posture converges to the desired trajectory better than with the classical controller.

CHAPTER 7

CONCLUSION AND FUTURE WORK

In this thesis, the multi nonholonomic robot formation with cluster space concept was studied. This study includes the dynamic model derivation and formation controllers derivations. Four different controllers were presented in this work. The first one is the behavioral adaptive controller which is used to give a higher priority to the formation task over the target following task. This controller was developed with an artificial fuzzy logic controller that allows it to take a behavioral action such as tuning the formation controller based on the priority list. The second controller is the robust sliding mode controller. This controller is robust against the model uncertainties and nonlinearities. Furthermore, for auto tuning this SMC an adaptive law is added, the resultant adaptive SMC shows improved performance with external disturbances. Also, an artificial fuzzy logic adaptive law was added to the SMC in order to get a faster tuning SMC than

the standard adaptive SMC. The fifth controller is a model predictive controller, which was developed to tackle the velocity saturation effects of the nonholonomic robots. The MPC optimizes the kinematic controller gains so that the effect of saturation is minimized.

The developed controllers were validated by simulation and experimental tests. The results show a considerable improvement in their usage.

Future work may be conducted to tackle the communication issues such as the limited bandwidth and wireless coverage issues. Additional future work should be conducted to tackle the case of a non homogeneous group of robots. For example, the case where a ground robot and drones are in the same cluster. Finally, the formation switching of cluster shapes and its application should be studied.

REFERENCES

- [1] Ignacio Mas and Christopher Kitts. Obstacle avoidance policies for cluster space control of nonholonomic multirobot systems. *Mechatronics, IEEE/ASME Transactions on*, 17(6):1068–1079, 2012.
- [2] Tucker Balch and Ronald C. Arkin. Behavior-based formation control for multirobot teams. *Robotics and Automation, IEEE Transactions on*, 14(6):926–939, 1998.
- [3] J.-M. Luna, Rafael Fierro, Chaouki Abdallah, and John Wood. An adaptive coverage control algorithm for deployment of nonholonomic mobile sensors. In *Decision and Control (CDC), 2010 49th IEEE Conference on*, pages 1250–1256. IEEE, 2010.
- [4] Naomi Ehrich Leonard and Edward Fiorelli. Virtual leaders, artificial potentials and coordinated control of groups. In *Decision and Control, 2001. Proceedings of the 40th IEEE Conference on*, volume 3, pages 2968–2973. IEEE, 2001.

- [5] Amit Ailon and Ilan Zohar. Control strategies for driving a group of non-holonomic kinematic mobile robots in formation along a time-parameterized path. *Mechatronics, IEEE/ASME Transactions on*, 17(2):326–336, 2012.
- [6] Young-Ho Lee, Shin-Guk Kim, Tae-Yong Kuc, Jong-Koo Park, Sang-Hoon Ji, Yong-Seon Moon, and Young-Jo Cho. Virtual target tracking of mobile robot and its application to formation control. *International Journal of Control, Automation and Systems*, 12(2):390–398, 2014.
- [7] Ali Noormohammadi Asl, Mohammad Bagher Menhaj, and Atena Sajedin. Control of leader-follower formation and path planning of mobile robots using Asexual Reproduction Optimization (ARO). *Applied Soft Computing*, 14, Part C:563–576, January 2014.
- [8] Yeong-Hwa Chang, Cheng-Yuan Yang, Wei-Shou Chan, Hung-Wei Lin, and Chia-Wen Chang. Adaptive Fuzzy Sliding-Mode Formation Controller Design for Multi-Robot Dynamic Systems. *International Journal of Fuzzy Systems*, 16(1):121, 2014.
- [9] Xiaoyu Cai and M. de Queiroz. Adaptive rigidity-based formation control of uncertain multi-robotic vehicles. In *American Control Conference (ACC), 2014*, pages 293–298, June 2014.
- [10] Yan Dong Li, Ling Zhu, and Ming Sun. Adaptive neural-network control of mobile robot formations including actuator dynamics. *Applied Mechanics and Materials*, 303:1768–1773, 2013.

- [11] Filippo Arrichiello, Stefano Chiaverini, and Thor I. Fossen. Formation control of underactuated surface vessels using the null-space-based behavioral control. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 5942–5947. IEEE, 2006.
- [12] F. Arrichiello, J. Das, H. Heidarsson, S. Chiaverini, and G. S. Sukhatme. Experiments in autonomous navigation with an under-actuated surface vessel via the Null-Space based Behavioral control. In *Advanced Intelligent Mechatronics, 2009. AIM 2009. IEEE/ASME International Conference on*, pages 362–367. IEEE, 2009.
- [13] Filippo Arrichiello, Stefano Chiaverini, Giovanni Indiveri, and Paola Pedone. The null-space-based behavioral control for mobile robots with velocity actuator saturations. *The International Journal of Robotics Research*, 2010.
- [14] G. Antonelli, F. Arrichiello, and S. Chiaverini. The Entrapment/Escorting Mission. *IEEE Robotics Automation Magazine*, 15(1):22–29, March 2008.
- [15] P. Mahacek, I Mas, O. Petrovic, J. Acain, and C. Kitts. Cluster space control of a 2-robot system as applied to Autonomous Surface Vessels. In *OCEANS 2008*, pages 1–5, September 2008.
- [16] Paul Mahacek, Ignacio Mas, and Christopher Kitts. Cluster space control of autonomous surface vessels utilizing obstacle avoidance and shielding techniques. In *Autonomous Underwater Vehicles (AUV), 2010 IEEE/OES*, pages 1–5. IEEE, 2010.

- [17] Ignacio Mas, Ognjen Petrovic, and Christopher Kitts. Cluster space specification and control of a 3-robot mobile system. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 3763–3768. IEEE, 2008.
- [18] Michael Seamus Agnew, Patrick Dal Canto, Christopher A. Kitts, and Steve Li. Cluster space control of aerial robots. In *Advanced Intelligent Mechatronics (AIM), 2010 IEEE/ASME International Conference on*, pages 1305–1310. IEEE, 2010.
- [19] Christopher A. Kitts and Ignacio Mas. Cluster space specification and control of mobile multirobot systems. *Mechatronics, IEEE/ASME Transactions on*, 14(2):207–218, 2009.
- [20] Sami El Ferik, Mohammad Tariq Nasir, and Uthman Baroudi. A Behavioral Adaptive Fuzzy controller of multi robots in a cluster space. *Applied Soft Computing*, 44:117–127, July 2016.
- [21] Paul Mahacek, Christopher A. Kitts, and Ignacio Mas. Dynamic guarding of marine assets through cluster control of automated surface vessel fleets. *Mechatronics, IEEE/ASME Transactions on*, 17(1):65–75, 2012.
- [22] Christopher Kitts, Paul Mahacek, Thomas Adamek, and Ignacio Mas. Experiments in the control and application of Automated Surface Vessel fleets. In *OCEANS 2011*, pages 1–7. IEEE, 2011.

- [23] I. Mas and C.A. Kitts. Dynamic Control of Mobile Multirobot Systems: The Cluster Space Formulation. *IEEE Access*, 2:558–570, 2014.
- [24] Jean-Paul Laumond, S. Sekhavat, and F. Lamiroux. *Guidelines in nonholonomic motion planning for mobile robots*. Springer, 1998.
- [25] Christopher A. Kitts, Kyle Stanhouse, and Piya Chindaphorn. Cluster space collision avoidance for mobile two-robot systems. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 1941–1948. IEEE, 2009.
- [26] Petru Rusu, Emil M. Petriu, Thomas E. Whalen, Aurel Cornell, and Hans JW Spoelder. Behavior-based neuro-fuzzy controller for mobile robot navigation. *IEEE Transactions on Instrumentation and Measurement*, 52(4):1335–1340, 2003.
- [27] Maja J. Mataric. Behaviour-based control: Examples from navigation, learning, and group behaviour. *Journal of Experimental & Theoretical Artificial Intelligence*, 9(2-3):323–336, 1997.
- [28] Eddie Tunstel. Coordination of distributed fuzzy behaviors in mobile robot control. In *Systems, Man and Cybernetics, 1995. Intelligent Systems for the 21st Century., IEEE International Conference on*, volume 5, pages 4009–4014. IEEE, 1995.

- [29] Prahlad Vadakkepat, Ooi Chia Miin, Xiao Peng, and Tong Heng Lee. Fuzzy behavior-based control of mobile robots. *Fuzzy Systems, IEEE Transactions on*, 12(4):559–565, 2004.
- [30] Eugenio Aguirre and Antonio Gonzalez. Fuzzy behaviors for mobile robot navigation: design, coordination and fusion. *International Journal of Approximate Reasoning*, 25(3):255–289, 2000.
- [31] Kiyotaka Izumi and Keigo Watanabe. Fuzzy behavior-based control trained by module learning to acquire the adaptive behaviors of mobile robots. *Mathematics and Computers in Simulation*, 51(3):233–243, 2000.
- [32] Alessandro Marino, Fabrizio Caccavale, Lynne E. Parker, and Gianluca Antonelli. Fuzzy behavioral control for multi-robot border patrol. In *Control and Automation, 2009. MED'09. 17th Mediterranean Conference on*, pages 246–251. IEEE, 2009.
- [33] Saroj Kumar Pradhan, Dayal Ramakrushna Parhi, and Anup Kumar Panda. Fuzzy logic techniques for navigation of several mobile robots. *Applied Soft Computing*, 9(1):290–304, January 2009.
- [34] Rajibul Huq, George K. I. Mann, and Raymond G. Gosine. Mobile robot navigation using motor schema and fuzzy context dependent behavior modulation. *Applied Soft Computing*, 8(1):422–436, January 2008.

- [35] Dayal Ramakrushna Parhi, Saroj Kumar Pradhan, Anup Kumar Panda, and Rabindra Kumar Behera. The stable and precise motion control for multiple mobile robots. *Applied Soft Computing*, 9(2):477–487, March 2009.
- [36] Dayal Ramakrushna Parhi and Jagadish Chandra Mohanta. Navigational control of several mobile robotic agents using Petri-potential-fuzzy hybrid controller. *Applied Soft Computing*, 11(4):3546–3557, June 2011.
- [37] Dongbing Gu and Huosheng Hu. A model predictive controller for robots to follow a virtual leader. *Robotica*, 27(06):905–913, 2009.
- [38] S. El-Ferik, B.A. Siddiqui, and F.L. Lewis. Distributed nonlinear MPC formation control with limited bandwidth. In *American Control Conference (ACC), 2013*, pages 6388–6393, June 2013.
- [39] Sami El-Ferik, Bilal A Siddiqui, and Frank L Lewis. Distributed nonlinear mpc of multi-agent systems with data compression and random delays-extended version. *arXiv preprint arXiv:1504.03494*, 2015.
- [40] D Kostić, Sisdarmanto Adinandra, Jurjen Caarls, Nathan van de Wouw, and Henk Nijmeijer. Saturated control of time-varying formations and trajectory tracking for unicycle multi-agent systems. In *49th IEEE Conference on Decision and Control (CDC)*, pages 4054–4059. IEEE, 2010.
- [41] Xiaohan Chen and Yingmin Jia. Input-constrained formation control of differential-drive mobile robots: geometric analysis and optimisation. *IET Control Theory Applications*, 8(7):522–533, May 2014.

- [42] Ibrahim MH Sanhoury, Shamsudin HM Amin, and Abdul Rashid Husain. Switching Between Formation in a Moving Shape for Multi-Robots via Synchronization Approach. *Procedia Engineering*, 41:678–684, 2012.
- [43] Xiaoli Wang, Wei Ni, and Xinsheng Wang. Leader-following formation of switching multirobot systems via internal model. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 42(3):817–826, 2012.
- [44] Yanyan Dai, SukGyu Lee, Yoon-Gu Kim, and Sung-Gil Wee. A switching formation strategy for obstacle avoidance of multi-robot system. In *Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), 2014 IEEE 4th Annual International Conference on*, pages 457–462. IEEE, 2014.
- [45] Wei Zhu and Daizhan Cheng. Leader-following consensus of second-order agents with multiple time-varying delays. *Automatica*, 46(12):1994–1999, 2010.
- [46] Enric Xargay, Ronald Choe, Naira Hovakimyan, and Isaac Kaminer. Multi-leader coordination algorithm for networks with switching topology and quantized information. *Automatica*, 50(3):841–851, 2014.
- [47] Hongjie Li, Lihua Xu, Libin Xiao, and Li Lin. Second-order leader-following consensus of nonlinear multi-agent systems via adaptive pinning control. In *Control and Decision Conference (2014 CCDC), The 26th Chinese*, pages 586–591. IEEE, 2014.

- [48] Qiang Song, Jinde Cao, and Wenwu Yu. Second-order leader-following consensus of nonlinear multi-agent systems via pinning control. *Systems & Control Letters*, 59(9):553–562, 2010.
- [49] Richard J. Duro, Manuel Graa, and Javier de Lope. On the potential contributions of hybrid intelligent approaches to multicomponent robotic system development. *Information Sciences*, 180(14):2635–2648, 2010.
- [50] Ignacio Mas, Jose Acain, Ognjen Petrovic, and Christopher Kitts. Error characterization in the vicinity of singularities in multi-robot cluster space control. In *Robotics and Biomimetics, 2008. ROBIO 2008. IEEE International Conference on*, pages 1911–1917. IEEE, 2009.
- [51] Ngoc Dung Vuong, Chongyou Ma, and Marcelo H. Ang Jr. Sensitivity of Task Space Performance to Null Space Control in Presence of Model Uncertainties. In *Experimental Robotics*, pages 923–934. Springer, 2014.
- [52] Luca Rosario Buonocore, Vincenzo Lippiello, Sabato Manfredi, Fabio Ruggiero, and Bruno Siciliano. Effects of Packet Losses on Formation Control of Unmanned Aerial Vehicles. In *World Congress*, volume 19, pages 1234–1240, 2014.
- [53] Filippo Arrichiello, Alessandro Marino, and Francesco Pierri. Distributed fault-tolerant strategy for networked robots with both cooperative and reactive controls. In *Information and Automation (ICIA), 2014 IEEE International Conference on*, pages 677–682. IEEE, 2014.

- [54] Hong Shi, Long Wang, and Tianguang Chu. Virtual leader approach to coordinated control of multiple mobile agents with asymmetric interactions. *Physica D: Nonlinear Phenomena*, 213(1):51–65, 2006.
- [55] Gianluca Antonelli, Filippo Arrichiello, and Stefano Chiaverini. Stability analysis for the null-space-based behavioral control for multi-robot systems. In *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*, pages 2463–2468. IEEE, 2008.
- [56] Rongxin Cui, Shuzhi Sam Ge, Bernard Voon Ee How, and Yoo Sang Choo. Leaderfollower formation control of underactuated autonomous underwater vehicles. *Ocean Engineering*, 37(17):1491–1502, 2010.
- [57] Mojtaba Naderi Soorki, Heidar Ali Talebi, and Seyed Kamaloddin Yadavar Nikravesh. A robust dynamic leader-follower formation control with active obstacle avoidance. In *Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on*, pages 1932–1937. IEEE, 2011.
- [58] Jawhar Ghommam, Hasan Mehrjerdi, and Maarouf Saad. Leader-follower formation control of nonholonomic robots with fuzzy logic based approach for obstacle avoidance. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 2340–2345. IEEE, 2011.
- [59] Filippo Arrichiello, Stefano Chiaverini, and Vaibhav Kumar Mehta. Experiments of obstacles and collision avoidance with a distributed multi-robot

- system. In *Information and Automation (ICIA), 2012 International Conference on*, pages 727–732. IEEE, 2012.
- [60] R. Fierro and F.L. Lewis. Control of a nonholonomic mobile robot using neural networks. *IEEE Transactions on Neural Networks*, 9(4):589–600, July 1998.
- [61] Shaojuan Yu, Xu Ding, Lin Li, Feng Pan, Shibo Xiong, and Yanhong Bai. Fuzzy PID-type Iterative learning control For Electro-hydraulic servo system. In *Computational Aspects of Social Networks (CASoN), 2010 International Conference on*, pages 671–674. IEEE, 2010.
- [62] C.M. Wang. Location estimation and uncertainty analysis for mobile robots. In *, 1988 IEEE International Conference on Robotics and Automation, 1988. Proceedings*, pages 1231–1235, April 1988.
- [63] G. Antonelli, S. Chiaverini, and G. Fusco. A calibration method for odometry of mobile robots based on the least-squares technique: theory and experimental validation. *IEEE Transactions on Robotics*, 21(5):994–1004, October 2005.
- [64] Hans-J. von der Hardt, Philippe Arnould, Didier Wolf, and Michel Dufaut. A method of mobile robot localisation by fusion of odometric and magnetometric data. *The International Journal of Advanced Manufacturing Technology*, 9(1):65–69, January 1994.

- [65] Jung-Min Yang and Jong-Hwan Kim. Sliding mode control for trajectory tracking of nonholonomic wheeled mobile robots. *IEEE Transactions on Robotics and Automation*, 15(3):578–587, June 1999.
- [66] Dongkyoung Chwa. Sliding-mode tracking control of nonholonomic wheeled mobile robots in polar coordinates. *Control Systems Technology, IEEE Transactions on*, 12(4):637–644, 2004.
- [67] Jun Ho Lee, Cong Lin, Hoon Lim, and Jang Myung Lee. Sliding mode control for trajectory tracking of mobile robot in the RFID sensor space. *International Journal of Control, Automation and Systems*, 7(3):429–435, 2009.
- [68] Jun Ku Lee, Yoon Ho Choi, and Jin Bae Park. Sliding mode tracking control of mobile robots with approach angle in cartesian coordinates. *International Journal of Control, Automation and Systems*, 13(3):718–724, February 2015.
- [69] Yasmine Koubaa, Mohamed Boukattaya, and Tarak Dammak. Adaptive Sliding-Mode Dynamic Control For Path Tracking of Nonholonomic Wheeled Mobile Robot. 2015.
- [70] Chih-Yang Chen, Tzue-Hseng S. Li, and Ying-Chieh Yeh. EP-based kinematic control and adaptive fuzzy sliding-mode dynamic control for wheeled mobile robots. *Information Sciences*, 179(12):180–195, January 2009.

- [71] Jian Chen, Dong Sun, Jie Yang, and Haoyao Chen. A leader-follower formation control of multiple nonholonomic mobile robots incorporating receding-horizon scheme. *The International Journal of Robotics Research*, 2009.
- [72] Jin Li and Liu Yang. Adaptive pi-based sliding mode control for nanopositioning of piezoelectric actuators. *Mathematical Problems in Engineering*, 2014, 2014.
- [73] Mohammad Tariq Nasir and Sami El-Ferik. Adaptive sliding-mode cluster space control of a non-holonomic multi-robot system with applications. *IET Control Theory & Applications*, 2016.
- [74] Mohammad Mahdi Ebrahimi, Farzin Piltan, Mansour Bazregar, and AliReza Nabaee. Artificial chattering free on-line modified sliding mode algorithm: Applied in continuum robot manipulator. *International Journal of Information Engineering and Electronic Business*, 5(5):57, 2013.
- [75] Yeong-Hwa Chang, Cheng-Yuan Yang, Wei-Shou Chan, Hung-Wei Lin, and Chia-Wen Chang. Adaptive fuzzy sliding-mode formation controller design for multi-robot dynamic systems. *International Journal of Fuzzy Systems*, 16(1):121, 2014.
- [76] Hsiu-Ming Wu, Mansour Karkoub, and Chih-Lyang Hwang. Mixed fuzzy sliding-mode tracking with backstepping formation control for multi-nonholonomic mobile robots subject to uncertainties. *Journal of Intelligent & Robotic Systems*, 79(1):73–86, 2015.

- [77] Yeong-Hwa Chang, Chia-Wen Chang, Chun-Lin Chen, and Chin-Wang Tao. Fuzzy sliding-mode formation control for multirobot systems: Design and implementation. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 42(2):444–457, 2012.
- [78] H Javaheri and GR Vossoughi. Sliding mode control with online fuzzy tuning: Application to a robot manipulator. In *Mechatronics and Automation, 2005 IEEE International Conference*, volume 3, pages 1357–1362. IEEE, 2005.
- [79] R. Fierro and Frank L. Lewis. Control of a nonholonomic mobile robot using neural networks. *Neural Networks, IEEE Transactions on*, 9(4):589–600, 1998.

Vitae

- Name: Mohammad Tariq Mousa Nasir
- Nationality: Jordanian
- Date of Birth:7/2/1986
- Email: *eng.m.naser@gmail.com*
- Permenant Address: Po box 1116, Zarqa 13110, Jordan

Appendices

**.1 A Behavioral Adaptive Fuzzy controller of
multi robots in a cluster space**



A Behavioral Adaptive Fuzzy controller of multi robots in a cluster space



Sami El Ferik*, Mohammad Tariq Nasir, Uthman Baroudi

King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia

ARTICLE INFO

Article history:

Received 17 May 2015

Received in revised form 4 March 2016

Accepted 17 March 2016

Available online 4 April 2016

Keywords:

Cluster space
Behavioral control
Fuzzy adaptive
Multi-robots

ABSTRACT

Cooperation between autonomous robot vehicles holds several promising advantages like robustness, adaptability, configurability, and scalability. Coordination between the different robots and the individual relative motion represent both the main challenges especially when dealing with formation control and maintenance. Cluster space control provides a simple concept for controlling multi-agent formation. In the classical approach, formation control is the unique task for the multi-agent system. In this paper, the development and application of a novel Behavioral Adaptive Fuzzy-based Cluster Space Control (BAFC) to non-holonomic robots is presented. By applying a fuzzy priority control approach, BAFC deals with two conflicting tasks: formation maintenance and target following. Using priority rules, the fuzzy approach is used to adapt the controller and therefore the behavior of the system, taking into accounts the errors in the formation states and the target following states. The control approach is easy to implement and has been implemented in this paper using SIMULINK real-time platform. The communication between the different agents and the controller is established through Wi-Fi link. Both simulation and experimental results demonstrate the behavioral response where the robot performs the higher priority tasks first. This new approach shows a great performance with a lower control signal when benchmarked with previously known results in the literature.

© 2016 Elsevier B.V. All rights reserved.

1. Introduction

In addition to their ability to withstand extreme environment conditions, robotic systems can provide more speed, precision, repeatability and strength to any automated tasks such as sensor deployment, mapping, rescue like firefighting and surveillance to name just a few. However, a vast majority of applications use robots with individually assigned tasks. Hence, each robot operates independently from all other robots and is dedicated to achieve a particular task. Interest is growing in the use of multi-robot systems to push forward the limits of capabilities and improve individual work efficiency of each robot. The use of multi-robot systems provides great features such as flexibility, configurability, redundancy, increased coverage, throughput, and spatially diverse functionalities [1].

Multi-agent systems have many features but also several challenges. One of the key challenges they are facing is the simultaneous coordination of motion and formation control. Formation control is

important especially when sensors' coverage and/or capabilities are limited. In the case of sensors coverage, formation allows each robot to concentrate its sensors on a region of the area of interest [2]. For example, robot scout benefits from a formation by directing the sensors of each robot towards achieving maximum area coverage [3]. Several approaches have been proposed to address formation control. The concept of virtual leader in the leader-follower framework has been extensively studied in designing control strategies (see for example Refs. [4–6]). Comparison of different methods to optimize the path planning in leader-follower formation with obstacle avoidance and their suitability for real-time implementation has been presented in Ref. [7].

Cluster-space is another approach for formation control and maintenance (see Refs. [8–12]). This control method considers the group of robots as single entity and therefore, calculates the control commands at the cluster level. The main aspect of such approach is that the group of robots is perceived as one big robot. Once determined, cluster commands are translated to robot space commands by applying inverse kinematics and using the appropriate Jacobean matrix allowing each robot to have its own command that is derived from the cluster's command. By doing so, the control process is made simpler than dealing with many robot entities as in the virtual-leader framework.

* Corresponding author.

E-mail addresses: selferik@kfupm.edu.sa, sami.elferik@gmail.com (S. El Ferik), mtnasir@kfupm.edu.sa (M. Tariq Nasir), ubaroudi@kfupm.edu.sa (U. Baroudi).

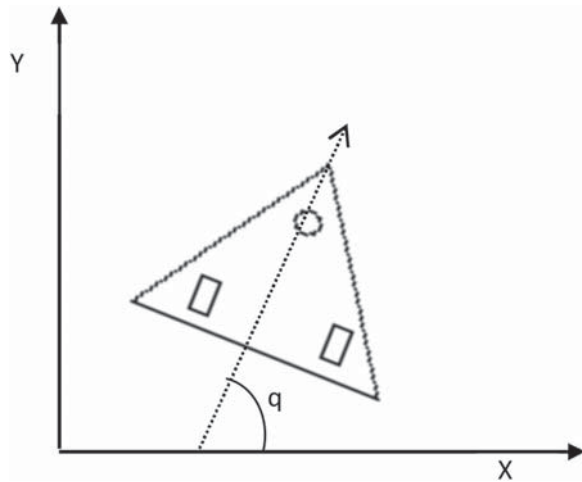


Fig. 1. WMR model, q is the WMRs heading angle, and X, Y are the robot center location.

Cluster space control can be centralized or decentralized. A centralized cluster space control is currently being implemented. Decentralized implementations are also possible [13]. However, the cluster space controller has its own challenges such as singularity problems—due to the Jacobean matrix of the cluster-to-robot space becoming singular [14], collision avoidance issues [15], and also the effects of the actuator nonlinearities on robot mobility which has the potential to put all the cluster in an unstable mode. Another similar approach is the null space algorithm. This is a task-based control algorithm for a group of multi-robots [16,17]. Null space approach is similar to the cluster space in the sense that it considers the robot group as one entity, but has some differences in the mathematical representation of the group dynamics.

Design of intelligent and adaptive controllers are crucial to improve the performance of the cluster while addressing issues like nonlinearity, uncertainty, faults, and external disturbances. Fuzzy control method is a promising intelligent method since it does not require the availability of a precise model. For instance, The study in Ref. [18] developed a grey-prediction self-organizing fuzzy controller to address active suspension systems control. Experimental results demonstrate the effectiveness of the approach when compared to self-organizing fuzzy controllers. Neuro-fuzzy techniques are used in Ref. [19] to optimize the control of robot navigation through combination of rules. A good review of fuzzy systems and their applications can be found in Ref. [20]. On the other hand,

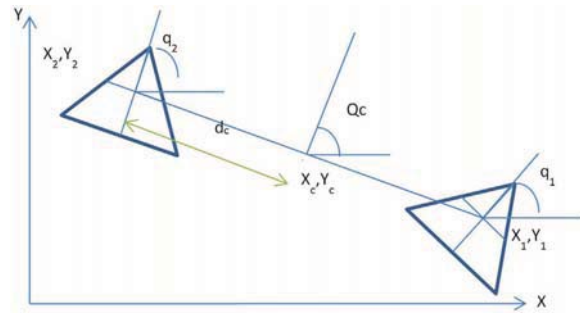


Fig. 2. Cluster space and robot space variables.

behavior control methods have been developed to tackle complex control problems that autonomous robots encounter in an unfamiliar real-world environment [21]. The behavior controller has general set of constraints that allow robots to react in a certain domain [22]. Based on these constraints the robot will select the appropriate response called behavior, task or routine. A hierarchy of distributed behaviors was tackled in the literature to fulfill a given goal with different scenarios. In order to switch between these behaviors, fuzzy logic technique was used since, in addition to the fact that it does not need a precise model, it is based on logistic commands that makes it suitable for representing the behavior selection criteria or constraints [21,23,24]. For example, behavior-based control has been tackled for soccer playing robot in Ref. [24] and was used in navigation and coordination control in Ref. [21,23,25,26]. However, these methods were not applied to multi-robot clusters. Behavioral-fuzzy controller for the null space has been proposed in Ref. [27]. However, to our knowledge it has never been implemented in cluster space control framework. Also, in literature, a classical PID controller was proposed as the behavior-based controller. Issues like disturbance effects, robots nonlinearity or actuator saturation are not yet considered. The nonlinearity issue is important especially when having a group of nonlinear robots dynamics. Equally important, actuator saturation issues can make the formation of the group of robots completely unstable. One of the recommended solutions to deal with these issues is fuzzy adaptive controllers [28–31].

Motivated by the research gaps in the literature and by the appealing nature of the fuzzy approach owing to its features, this paper proposes a novel behavior-fuzzy-based adaptive control algorithm (BAFC) for cluster space control. The present study considers two competing behaviors, which are target following and formation maintenance and control. The algorithm is simple, easy

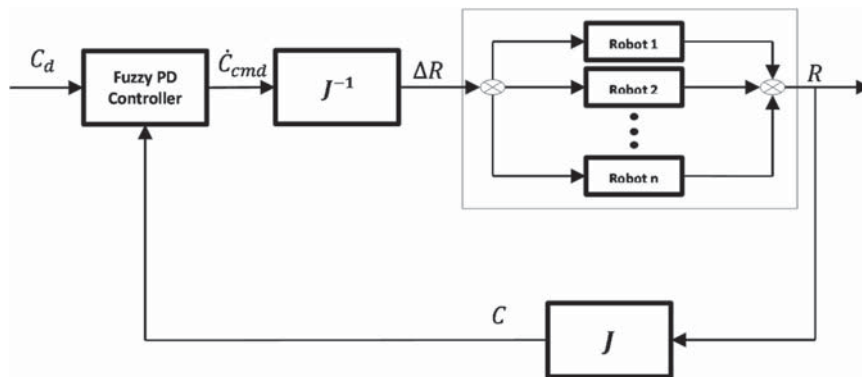


Fig. 3. Cluster space control architecture where ΔR is the robot commands and J is the Jacobian Matrix.

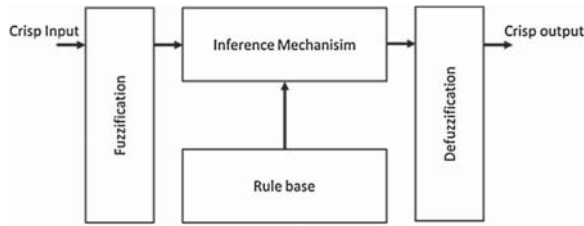


Fig. 4. Fuzzy control parts.

to implement, and its control approach accomplishes tasks based on their level of importance. In this proposed novel BAFC, the position error and its rate of change are both considered as inputs to the fuzzy logic control tuning algorithm. This will in turn improve the controller's performance in an adaptive manner. The task-based control algorithms considers the robots actuator power allocated to one task at a time. This may help in solving the actuator saturation issue.

In addition, cluster space concept has been applied on different types of robots, the unmanned vessel fleets [9,32], aerial robots and ground robots [11]. In this paper, non-holonomic robots are considered owing to the fact that the majority of wheeled robots are constrained in motion (wheels rotate without slipping). The implementation of the new control approach on Lego EV3 WMRs wheeled robot is also presented. The control strategy is implemented using

SIMULINK. Real-time communication between robots and the controller is established through a Wi-Fi link.

The rest of the paper is organized as follows. Section 2 introduces the kinematics and dynamics of non-holonomic robot as well as the definition of multi-robot cluster control. In Section 3, the adaptive fuzzy-based trajectory-tracking controller is presented with its stability proof. The simulation results are discussed in Section 4 and the experimental validation and results are presented in Section 5. Section 6 concludes the paper and discusses future work.

2. Non-holonomic robot dynamics

A wheeled mobile robot (WMR) with two driven wheels (in the rear part) and a passive castor wheel (in the front) is considered. The schematic model is shown in Fig. 1 (see Ref. [5] for more details).

The state-space model of the considered kinematic vehicle with the associated non-holonomic constraints (rolling with no slipping) is given by Eq. (1):

$$\begin{bmatrix} \dot{X}(t) = u_1(t) \cos(\theta(t)) \\ \dot{Y} = u_1(t) \sin(\theta(t)) \\ \dot{q}(t) = u_2(t) \end{bmatrix} \quad (1)$$

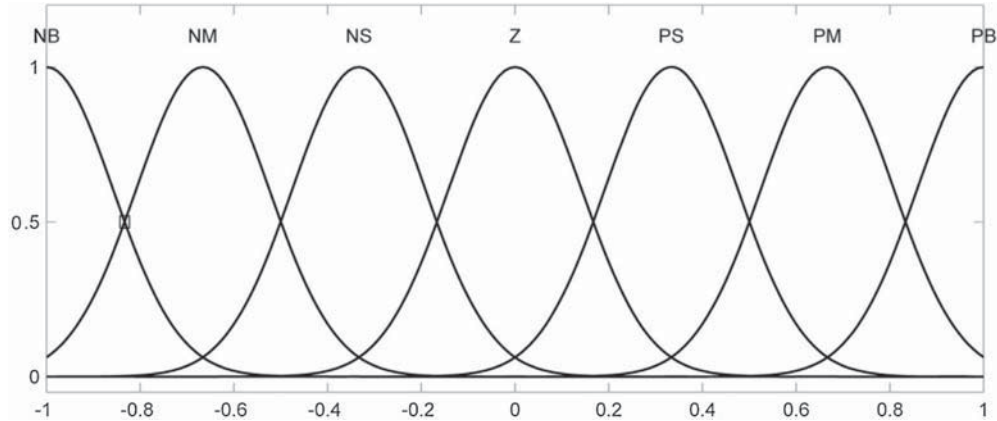


Fig. 5. The membership function of normalized $E, \dot{E}, \Delta K_p, \Delta K_d$.

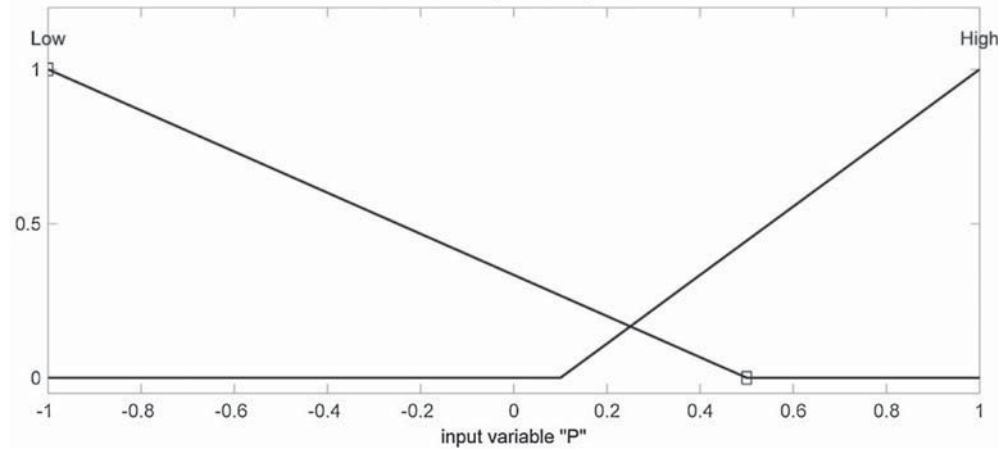


Fig. 6. The Membership function for the Priority input P .

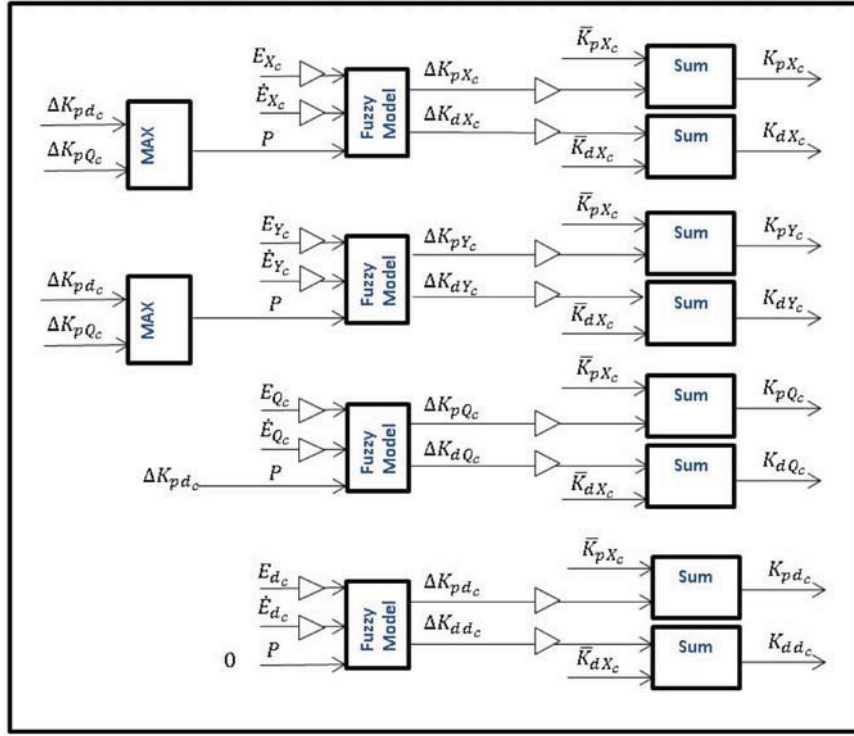


Fig. 7. The BAFC structure for two robot cluster.

2.1. Cluster space design

In order to implement the cluster space for two robots, an appropriate set of cluster variables are chosen to represent the shape of the cluster. As shown in Fig. 2, the proposed cluster variables are (Q_c, X_c, Y_c, d_c) and the corresponding robot space are (X_1, Y_1, q_1) , (X_2, Y_2, q_2) . The following Eqs. (2)–(5) shows the relation between the cluster space and the robot space variables.

$$X_c = \frac{X_1 + X_2}{2} \quad (2)$$

$$Y_c = \frac{Y_1 + Y_2}{2} \quad (3)$$

$$Q_c = \tan^{-1}(Y_1 - Y_2, X_1 - X_2) + \frac{\pi}{2} \quad (4)$$

$$d_c = \frac{1}{2} \sqrt{(Y_1 - Y_2)^2 + (X_1 - X_2)^2} \quad (5)$$

where (X_c, Y_c) is the center of the cluster; Q_c is the cluster heading and d_c is the spacing between the robots from the cluster center.

3. Control architecture

As shown in Fig. 3, the proposed cluster space controller, which is, consists of a closed loop controller with an adaptive fuzzy tuner that changes the controller parameters.

This scheme measures the robots states and converts it into cluster space states. The conversion is implemented by comparing cluster position and velocities with the desired trajectory values and outputting cluster velocities. These output cluster velocities are then translated into commands and sent to the robots. The following steps show the control procedure;

Step 1: calculating the error in the cluster variables using Eq. (6)

$$E_c = \begin{bmatrix} X_{cd} - X_c \\ Y_{cd} - Y_c \\ Q_{cd} - Q_c \\ d_{cd} - d_c \end{bmatrix} \quad (6)$$

Step 2: differentiate the error using Eq. (7)

$$\dot{E}_c = \frac{dE_c}{dt} \quad (7)$$

where E_c and \dot{E}_c are (1×4) vectors,

Step 3: adapt the values of K_p (1×4), and K_d (1×4) by implementing the fuzzy functions as given in Eqs. (8) and (9):

$$K_p = \bar{K}_p + \Delta K_p \{E_c, \dot{E}_c\}_{\text{fuzzy}} \quad (8)$$

$$K_d = \bar{K}_d + \Delta K_d \{E_c, \dot{E}_c\}_{\text{fuzzy}} \quad (9)$$

where \bar{K}_p and \bar{K}_d are the proportional and integral constant controller gains and

$$\Delta K_p \{E_c, \dot{E}_c\}_{\text{fuzzy}} = [\Delta K_{pXc}, \Delta K_{pYc}, \Delta K_{pQc}, \Delta K_{pdc}]$$

$$\Delta K_d \{E_c, \dot{E}_c\}_{\text{fuzzy}} = [\Delta K_{dXc}, \Delta K_{dYc}, \Delta K_{dQc}, \Delta K_{ddc}]$$

$$K_p = [K_{pXc}, K_{pYc}, K_{pQc}, K_{pdc}]$$

$$K_d = [K_{dXc}, K_{dYc}, K_{dQc}, K_{ddc}]$$

$$\bar{K}_p = [\bar{K}_{pXc}, \bar{K}_{pYc}, \bar{K}_{pQc}, \bar{K}_{pdc}]$$

$$\bar{K}_d = [\bar{K}_{dXc}, \bar{K}_{dYc}, \bar{K}_{dQc}, \bar{K}_{ddc}]$$

More details on the fuzzy model structure is presented in Section 3.1.

Step 4: applying the Cluster fuzzy PD controller commands using Eq. (8)

$$\dot{C}_{cmd} = K_p E_c + K_d \dot{E}_c \tag{10}$$

where $\dot{C}_{cmd} = [\Delta X_c, \Delta Y_c, \Delta Q_c, \Delta d_c]^T$. Note that ΔQ_c should be in the range of $[-\pi, \pi]$.

Step 5: the cluster controller commands are translated into robot velocities by calculating the velocity inverse kinematics as in Eqs. (11)–(14):

$$\Delta X_1 = \Delta X_c + \Delta d \cos(Q_c - \frac{\pi}{2}) - d_c \Delta Q_c \sin(Q_c - \frac{\pi}{2}) \tag{11}$$

$$\Delta Y_1 = \Delta Y_c + \Delta d \sin(Q_c - \frac{\pi}{2}) + d_c \Delta Q_c \cos(Q_c - \frac{\pi}{2}) \tag{12}$$

$$\Delta X_2 = \Delta X_c - \Delta d \cos(Q_c - \frac{\pi}{2}) + d_c \Delta Q_c \sin(Q_c - \frac{\pi}{2}) \tag{13}$$

$$\Delta Y_2 = \Delta Y_c - \Delta d \sin(Q_c - \frac{\pi}{2}) - d_c \Delta Q_c \cos(Q_c - \frac{\pi}{2}) \tag{14}$$

Step 6: the low level controller (WMRs controller):

$$E_{i1} = \Delta X_i \tag{15}$$

$$E_{i2} = \Delta Y_i \tag{16}$$

$$E_{i3} = Q_c - Q_{id} \tag{17}$$

$$E_{i4} = Q_{id} - q_i \tag{18}$$

where $i = 1, 2$ indicates the robot index.

$$Q_{id} = \tan^{-1} \left(\frac{E_{i2}}{E_{i1}} \right) \tag{19}$$

The robot commands in Eq. (1) can be calculated using Eqs. (20) and (21)

$$u_{i1} = \sqrt{(E_{i1}^2 + E_{i2}^2)} \cos(E_{i4}) \tag{20}$$

$$u_{i2} = k_i E_{i3} \tag{21}$$

where $0 < k_i$, where u_1 is the head speed and u_2 is the orientation speed.

3.1. The Fuzzy controller design

As described in Fig. 4, the body of the fuzzy controllers consists of

1. Input fuzzification (crisp-to-fuzzy conversion).
2. Fuzzy rule base (linguistic knowledge base).
3. Inference engine and Output defuzzification (fuzzy-to-crisp conversion).

The fuzzy inference engine simulates the fuzzy rules using the input variables. In order to achieve the foregoing objectives, two main methods are generally used: Mamdani’s method and Sugeno’s method. Mamdani’s method is the first known fuzzy inference system, which consists of fuzzification, rule evaluation, aggregation of the rule outputs, and defuzzification. The main difference between Mamdani’s method and Sugeno’s method is in the way crisp output is generated from the fuzzy inputs. While Mamdani’s method uses the defuzzification of a fuzzy output, Sugeno’s method uses a weighted average function to get the crisp output.

In this work, Mamdani’s method has been adopted, which is the commonly used inference engine. The proposed fuzzy approach starts with applying the defuzzification step to get the fuzzy values from the crisp inputs. These fuzzy values are represented in the fuzzy membership functions (see Figs. 5 and 6). Next, the fuzzy rules are simulated on the fuzzy inputs as shown in Tables 1–3. In this case, the fuzzy inputs are the error terms, derivatives of the errors

Table 1
The fuzzy rules of ΔK_p provided that P is low.

ΔK_p	EC							
	NB	NM	NX	Z	PS	PM	PB	
E	NB	PB	PB	PM	PM	PS	Z	Z
	NM	PB	PB	PM	PS	PS	Z	NS
	NS	PM	PM	PM	PS	Z	NS	NS
	Z	PM	PN	PS	Z	NS	NM	NM
	PS	PS	PS	Z	NS	NS	NM	NM
	PM	PS	Z	NS	NM	NM	NM	NB
	PB	Z	Z	NM	NM	NM	NB	NB

Table 2
The fuzzy rules of ΔK_d provided that P is low.

ΔK_d	EC							
	NB	NM	NS	Z	PS	PM	PB	
E	NB	PS	NS	NB	NB	NB	NM	PS
	NM	PS	NS	NB	NM	NM	NS	Z
	NS	Z	NS	NM	NM	NS	NS	Z
	Z	Z	NS	NS	NS	NS	NS	Z
	PS	PB	PS	PS	PS	PS	PS	PB
	PM	PB	PM	PM	PM	PS	PS	PB
	PB	PB	PM	PM	PM	PS	PS	PB

Table 3
The fuzzy rules of P .

Kp/Kd	P	
	Low	High
All the fuzzy values	Rules form Tables 1 and 2	NB

on the cluster space and the priority value P . The fuzzy outputs are the controller tuning parameters ΔK_p and ΔK_d [33].

3.2. BAFC structure

Fig. 7 shows the BAFC graphical structure for two robots cluster presented in Eqs. (8) and (9). The priority is achieved using by the max function. The higher control gain from the higher priority states will be passed. Therefore, if the output of the max is high then the fuzzy model will output a low control gain. And if no high gain is applied to the higher priority states, then the max function will return low signal. By so doing, the fuzzy model will give this state the priority to adapt its errors. The small triangles are the scaling constants that may be selected by try and error or by applying an evolutionary approach such as Genetic Algorithm.

Remark 1. PD controller is used in this case as an example of the classical controllers. Any controller, especially PID type, can also be selected.

Remark 2. The proposed fuzzy model is a hybrid adaptive system. This means that the fuzzy model is simulated in a discrete time, while the robots dynamic model is simulated in the continuous time. Therefore, the fuzzy model should be slower than the robots model during the simulation.

3.3. Stability proof

Let

$$E_c = C_d - C \tag{22a}$$

$$\dot{E}_c = \dot{C}_d - \dot{C} \tag{22b}$$

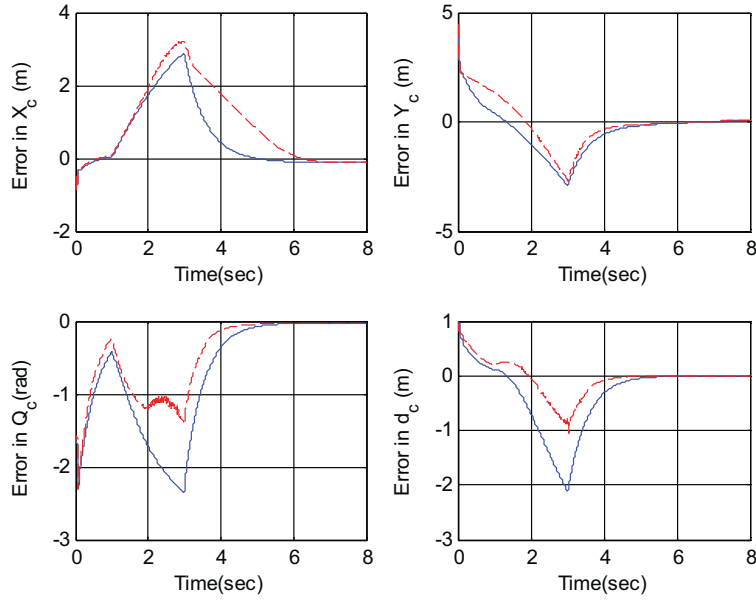


Fig. 8. A comparison between the classical PD controller and BAFC. The solid blue lines are the system response with classical PD controller, and the dashed red lines are BAFC responses. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

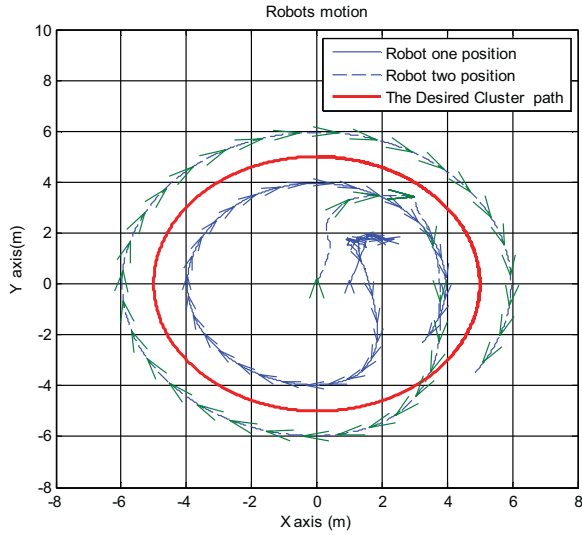


Fig. 9. Robots motion with fuzzy adaptive cluster controller.

and

$$C_{\text{cmd}} = \int_0^t K_p E_c + K_d \dot{E}_c d\tau \quad (23)$$

Theorem 1. Consider the mobile non-holonomic system (1). If the control command defined in Eq. (23) is applied to the mobile robot then the position and the velocity tracking errors (22a) and (22b) converge to zero.

Proof. Starting from Eqs. (8)–(10), the cluster states error and its derivative E_c , \dot{E}_c are defined as in as in Eqs. (22a), (22b) and (23),

$$\dot{C}_{\text{cmd}} = K_p E_c + K_d \dot{E}_c$$

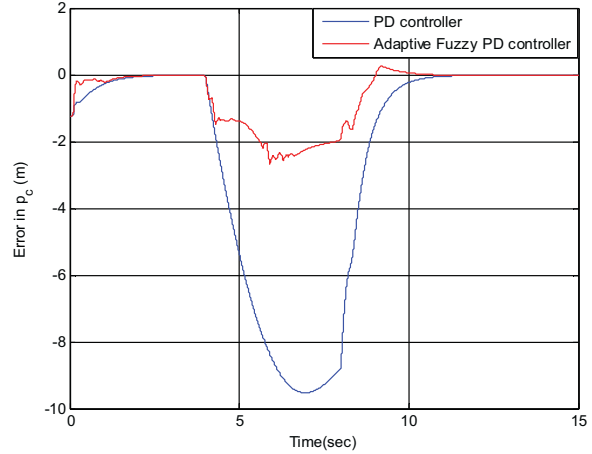


Fig. 10. Comparison between the BAFC and the Normal PD controller with p_c state.

where $\bar{K}_p, \bar{K}_d > 0$ and $\{E_c, \dot{E}_c\}_{\text{fuzzy}} \leq \bar{K}_p, \bar{K}_d$.

Substituting \dot{E}_c from Eq. (23) into Eq. (10) leads to

$$\dot{E}_c = \dot{C}_d - K_p(C_d - C) - K_d(\dot{C}_d - \dot{C})$$

$$\dot{E}_c = \frac{\dot{C}_d - K_p E_c}{1 + K_d} \quad (24)$$

where $\bar{K}_p, \bar{K}_d > 0$. Now the equilibrium point is $E_c = [0 \ 0 \ 0 \ 0]^T$, so by considering the following Lyapunov function candidate

$$V = \frac{1}{2} E_c^T E_c \quad (25)$$

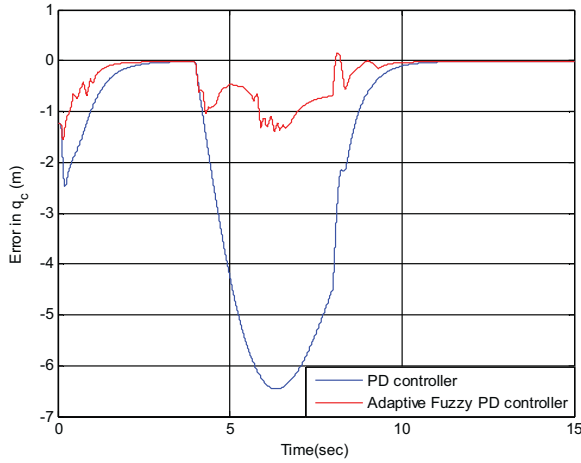


Fig. 11. Comparison between the BAFC and the Normal PD controller with q_c state.

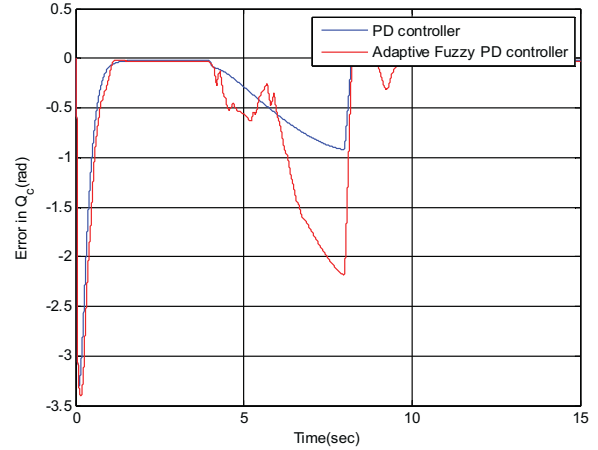


Fig. 13. Comparison between the BAFC and the Normal PD controller with Q_c state.

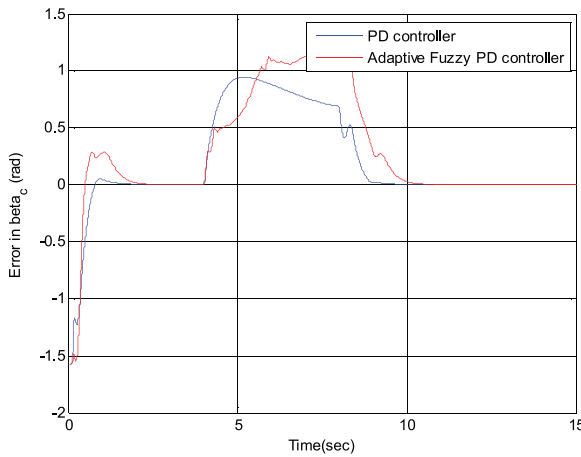


Fig. 12. Comparison between the BAFC and the Normal PD controller with β_c state.

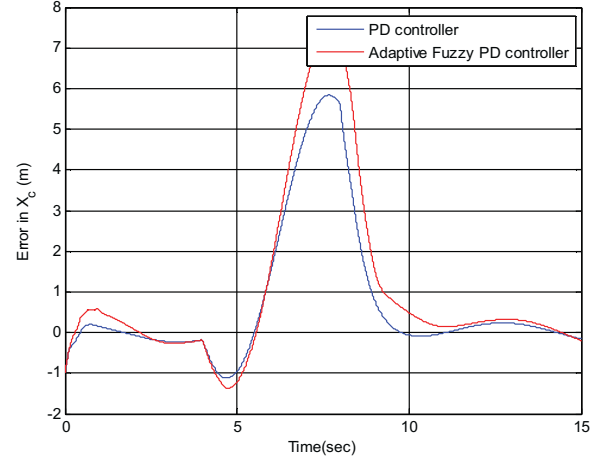


Fig. 14. Comparison between the BAFC and the Normal PD controller with X_c state.

with $V(\vec{0}) = 0$. Computing the derivative $\frac{dV}{dt}$

$$\frac{dV}{dt} = E_c^T \dot{E}_c = \frac{E_c^T \dot{C}_c - E_c^T K_p E_c}{1 + K_d} \quad (26)$$

According to Lyapunov theorem, the system is stable if $V(\vec{0}) = 0$, $dV(\vec{0})/dt = 0$ and $dV/dt < 0$. Since $-E_c^T K_p E_c < -\lambda_{\min}(K_p) \|E_c\|^2$ and $1 + K_d > 0$ where $\lambda_{\min}(K_p)$ is the minimum eigenvalues of the controller gain K_p , then $dV/dt < 0$ is true when Eq. (27) is satisfied, and accordingly the system is stable.

$$\|E_c^T\| < \frac{\|\dot{C}_d\|}{\lambda_{\min}(K_p)} \quad (27)$$

Remark 3. The above proof is for general PD control gains. When using fuzzy logic tuning, the PD controller becomes nonlinear. A necessary and sufficient condition for stability is to always verify that $K_p > 0$ and $(1 + K_d) > 0$. However, in practice, actuator saturation or heterogeneous characteristics of the robots may lead to instability. The first issue is well known in the literature. The latter is due to the coupling created within the cluster between the robots and the inability of some robots to keep up with the cluster. This issue will be investigated in a future work.

4. Simulation results

4.1. Two robot simulation

In this case, disturbance was added to robot 2 during a time frame of 1–3 s. Also, there is an existence of an initial condition error. Therefore, the controller should overcome two challenges: the initial condition and the disturbances. And also the controller's objective is to give more priority to the formation shape than the target following, in addition to the adaptability of the controller gain based on the changes in the states errors and errors velocity. As seen in Fig. 8, the BAFC's response of the shape states (d_c and Q_c) are better than the classical controller (Fig. 9).

4.2. Three robot simulation

The dynamics of three robots cluster are presented in Ref. [10]. The BAFC structure is similar to the one with two robots but with more fuzzy models. In this case, disturbances were added to both robot 2 and robot 3 between 4–8 s. And also, there is existence of an initial condition error. Therefore, the controller should be able to overcome two challenges: the initial condition error and the disturbances. The controller's objective is to give priority to the formation

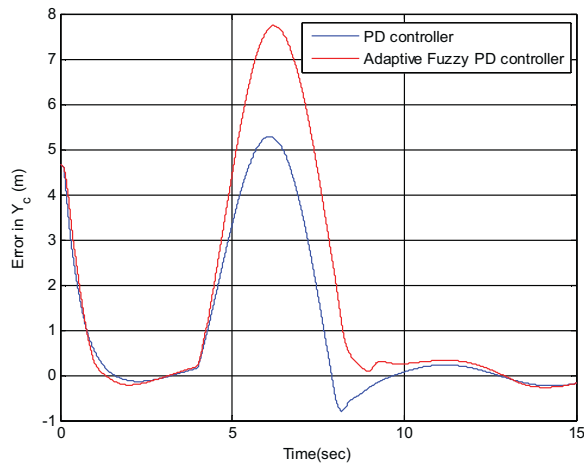


Fig. 15. Comparison between the BAFC and the Normal PD controller with Y_c state.

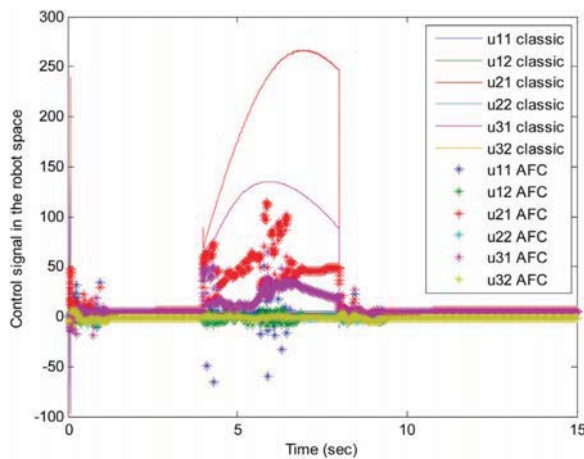


Fig. 16. Comparison between the control signal from the classical controller and the BAFC.

shape more than the target following. In addition, BAFC adapts the controller gain based on the changes in the states errors and errors velocity. Figs. 10–15 show the comparison between the classical controller and the BAFC. The results show better performance in q_c , p_c states, as depicted in Figs. 10 and 11. These improvements are apparent in the transient response even with both the initial condition and the disturbance challenges. q_c , p_c are considered as the shape states and they are given the highest priority among the system states followed by β_c , Q_c , X_c and Y_c sequentially. As a result, q_c , p_c should have better responses with BAFC than the classical controller.

Other potential advantages of BAFC over the classical approach are the actuators energy consumption and the max control values. As shown in Fig. 16, the max absolute value of the control signal with the classical controller is 266 and with BAFC is 115. This means that BAFC is better in dealing with actuator saturation than the classical controller, and this will in turn minimize the effects of actuators saturation problem. Therefore, BAFC requires a smaller actuator to be used in the Robots. This observation is arguably correct because BAFC controller allocates the energy to one objective at a time rather than to two conflicting objectives. This conflict may result to increase of the control energy and may cause instability.



Fig. 17. Lego EV3 WMRs.

5. Experimental setup

The Lego EV3 WMRs are used in the experiments. Those WMRs (see Fig. 17) have a 32-bit, 48 Mhz ARM9CPU with 16MB flash memory and 64MB RAM, Bluetooth and Wi-Fi transceivers, and two servo motors with encoders $\pm 1^\circ$ resolution. A PC interface with SIMULINK program is also required to transmit the control signal by means of Wi-Fi protocol. The SIMULINK has a powerful feature called External Mode. This feature is useful for monitoring and tuning the EV3 WMRs controller online. A Two-level controller structure uses High level and Low level controllers. The high level-controller is in the center PC, which enables the WMRs to send and receive data/command to and from the central PC.

Based on the proposed control algorithm, the central PC receives the location feedback from each robot, it then calculates the error and control signals and sends the command to each robot. The low-level controller on the robot CPU (simply a closed loop speed controller) receives the commands from the PC and relays these signals to the motors. The encoders provide special measurements for the feedback. The actual time of one loop process depends on the robot sampling time (set to 25 ms) and the Wi-Fi delay time, which is dependent on the computer speed and the network usage. The Robot localization is achieved by using the encoders.

5.1. Experiment test

Two WMRs moves from initial positions $[x, y, q]$ for the first robot $[0.2, 0, \frac{\pi}{2}]$ and $[0, 0, \frac{\pi}{2}]$ for the second robot. The desired path is $Q_c = \frac{\pi}{2}$, $x_c = 0.5$, $y_c = 0.1t$, $d_c = 0.3$ in the cluster space that is equal to $x_1 = 0.8$, $y_1 = 0.1t$, $q_1 = \frac{\pi}{2}$, $x_2 = 0.2$, $y_2 = 0.1t$, $q_2 = \frac{\pi}{2}$ in the robot space, where t is the time, (see Fig. 18). In order to show the disturbance effects on each controller a software disturbance is added to Robot 2. This disturbance will hold Robot 2 for 90 s with no movement see Fig. 18(A–C). The comparison now is based on how the other robot (Robot 1) will behave using both approaches (classical PD controller and the proposed BAFC).

Comparing the normal PD controller response (see Fig. 19) with the fuzzy PD controller (see Fig. 20), the BAFC gives more priority to the shape than the normal PD. Fig. 21 shows the improved shape dynamics (the cluster angle and the distance between the robots) with the proposed BAFC controller when compared with the normal one. The Odometer uncertainties cause an accumulated error which can be lessened by calibrating the odometry equations [34,35]. Also, adding a sensor-like compass can greatly reduce this error [36].

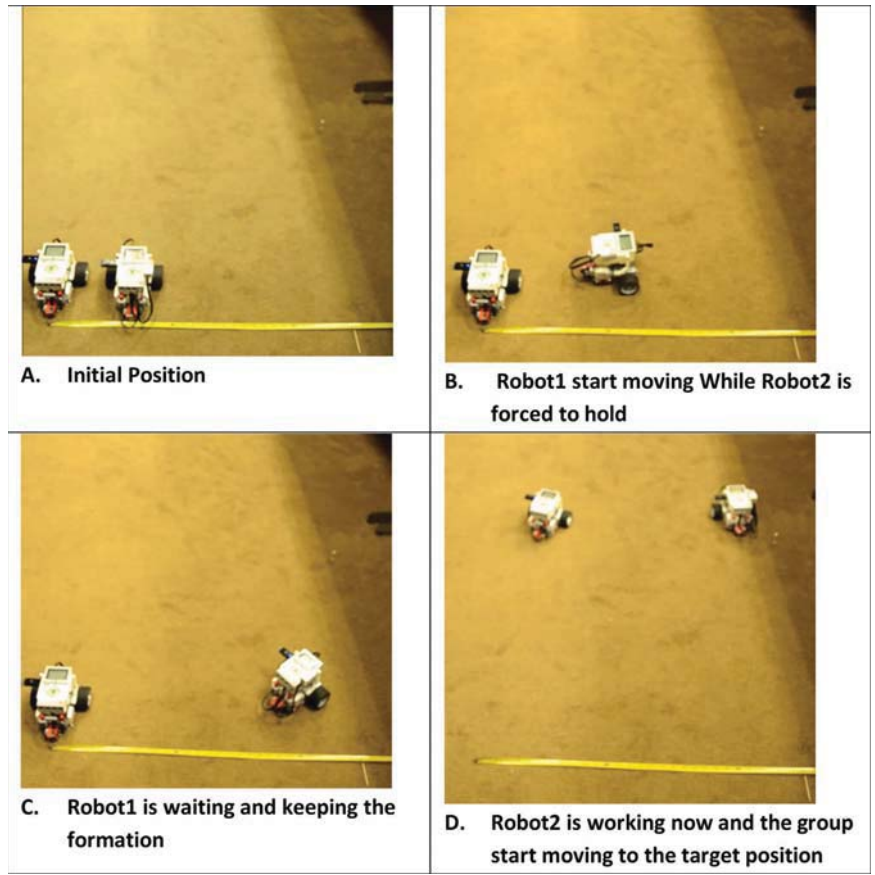


Fig. 18. Experiment with two WMRs.

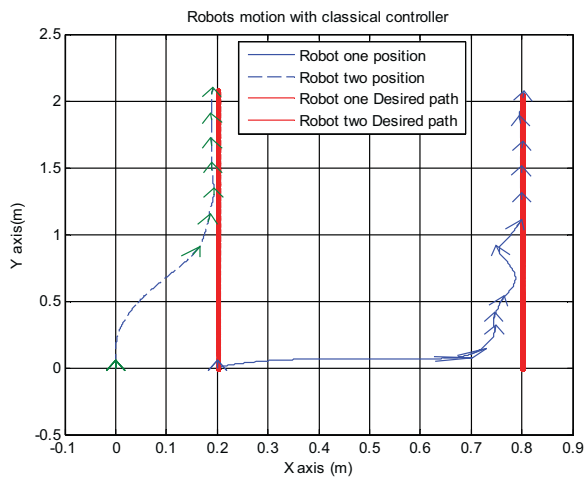


Fig. 19. The experimental robot motion with normal PD controller.

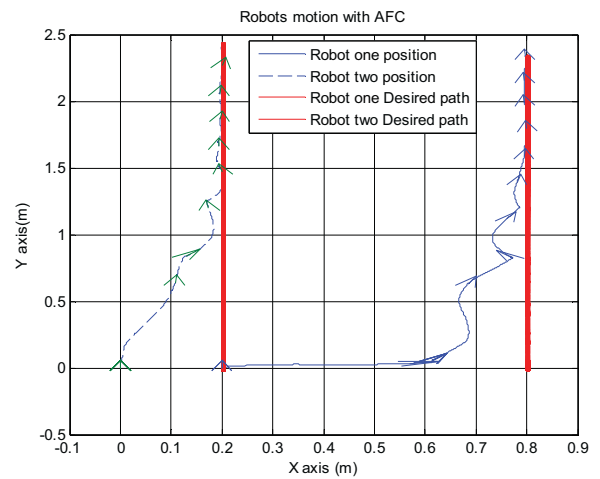


Fig. 20. Experimental robots motions with fuzzy PD controller.

6. Conclusion

Cluster space concept for controlling multi-robot systems is useful in simplifying the formation problem. In this work, an adaptive fuzzy controller is designed to improve the dynamics of the cluster space controllers. The cluster space dynamics were divided into

two main groups based on their tasks or behaviors: the formation shape states and the target following states. Therefore, BAFC gives more priority to the formation shape states than the target following states. BAFC adapts the controller gain based on the states errors and also velocities errors. Simulations and experimental results show that the proposed behavioral adaptive technique has

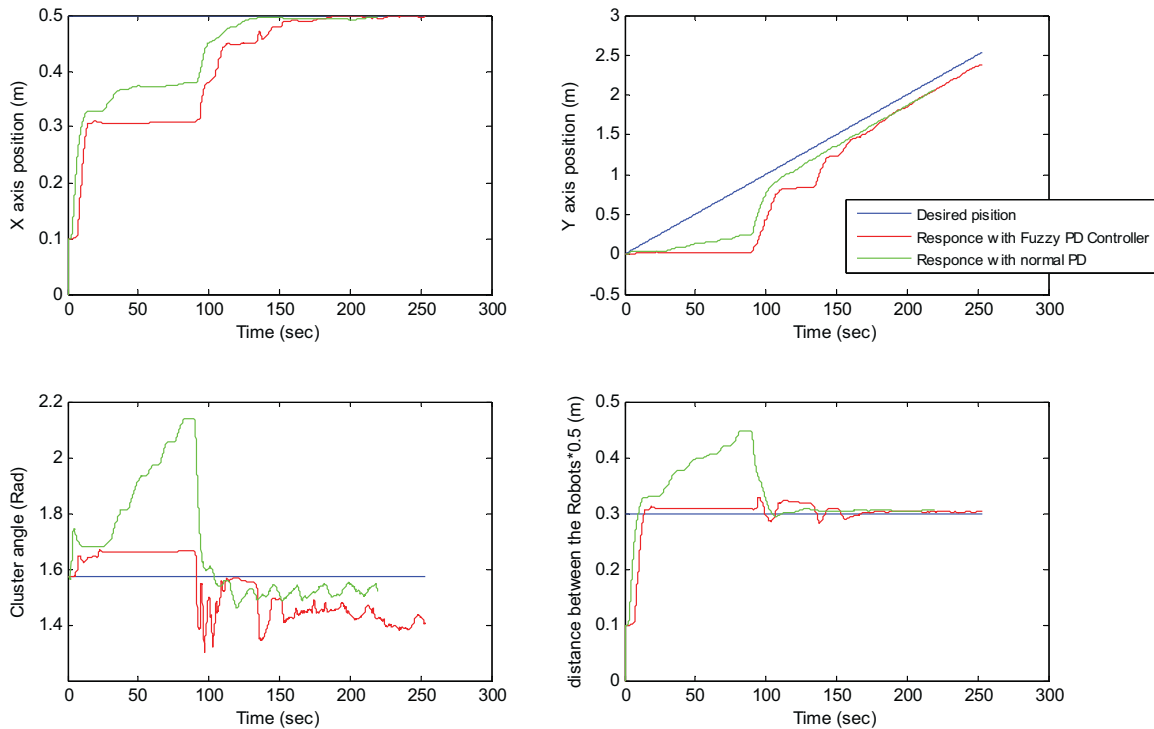


Fig. 21. Experimental results.

a significant potential. Formation dynamics have been improved in addition to having lower actuator input in the robot cluster, which helps with actuator saturation issues. Using BAFC, the control designer has the flexibility to select the states priority and design an error based adaptive controller. There are many extensions that can be considered for this work, while keeping its easy implementation features. For instance, future work may tackle clusters with larger number of heterogeneous robots and apply intelligent methods to overcome the singularities in the cluster dynamics in addition to addressing the behavior-based obstacle avoidance problem. Fault tolerant cluster control is another area where study of the effect of faults and how to guarantee the performance of the cluster. In this area, division of the cluster to many sub-cluster could be sought. Effect of actuator could be formally addressed and quantified.

Acknowledgments

The authors are grateful to the respected anonymous reviewers for their valuable comments that greatly improved the clarity and readability of the paper. The authors would like also to acknowledge the support provided by King Abdulaziz City for Science and Technology (KACST) through the Science & Technology Unit at King Fahd University of Petroleum & Minerals (KFUPM) for funding this work through project No. 11-ELE2147-04, as part of the National Science, Technology and Innovation Plan.

References

- [1] I. Mas, C. Kitts, Obstacle avoidance policies for cluster space control of nonholonomic multirobot systems, *IEEEASME Trans. Mechatron.* 17 (6) (2012) 1068–1079.
- [2] T. Balch, R.C. Arkin, Behavior-based formation control for multirobot teams, *IEEE Trans. Robot. Autom.* 14 (6) (1998) 926–939.
- [3] J.-M. Luna, R. Fierro, C. Abdallah, J. Wood, An adaptive coverage control algorithm for deployment of nonholonomic mobile sensors, 2010 49th IEEE Conference on Decision and Control (CDC) (2010) 1250–1256.
- [4] N.E. Leonard, E. Fiorelli, Virtual leaders, artificial potentials and coordinated control of groups, *Proceedings of the 40th IEEE Conference on Decision and Control*, 2001 vol. 3 (2001) 2968–2973.
- [5] A. Ailon, I. Zohar, Control strategies for driving a group of nonholonomic kinematic mobile robots in formation along a time-parameterized path, *IEEEASME Trans. Mechatron.* 17 (2) (2012) 326–336.
- [6] Y.-H. Lee, S.-G. Kim, T.-Y. Kuc, J.-K. Park, S.-H. Ji, Y.-S. Moon, Y.-J. Cho, Virtual target tracking of mobile robot and its application to formation control, *Int. J. Control Autom. Syst.* 12 (2) (2014) 390–398.
- [7] A. Noormohammadi Asl, M.B. Menhaj, A. Sajedin, Control of leader–follower formation and path planning of mobile robots using Asexual Reproduction Optimization (ARO), *Appl. Soft Comput.* 14 (January Part C) (2014) 563–576.
- [8] P. Mahacek, I. Mas, O. Petrovic, J. Acain, C. Kitts, Cluster space control of a 2-robot system as applied to autonomous surface vessels, *OCEANS 2008* (2008) 1–5.
- [9] P. Mahacek, I. Mas, C. Kitts, Cluster space control of autonomous surface vessels utilizing obstacle avoidance and shielding techniques, *Autonomous Underwater Vehicles (AUV)*, 2010 IEEE/OES (2010) 1–5.
- [10] I. Mas, O. Petrovic, C. Kitts, Cluster space specification and control of a 3-robot mobile system, *IEEE International Conference on Robotics and Automation*, 2008 (ICRA 2008) (2008) 3763–3768.
- [11] M.S. Agnew, P. Dal Canto, C.A. Kitts, S. Li, Cluster space control of aerial robots, 2010 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM) (2010) 1305–1310.
- [12] C.A. Kitts, I. Mas, Cluster space specification and control of mobile multirobot systems, *IEEEASME Trans. Mechatron.* 14 (2) (2009) 207–218.
- [13] I. Mas, C. Kitts, Centralized and decentralized multi-robot control methods using the cluster space control framework, 2010 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM) (2010) 115–122.
- [14] I. Mas, J. Acain, O. Petrovic, C. Kitts, Error characterization in the vicinity of singularities in multi-robot cluster space control, *IEEE International Conference on Robotics and Biomimetics*, 2008 (ROBIO 2008) (2008) 1911–1917.
- [15] C.A. Kitts, K. Stanhouse, P. Chindaphorn, Cluster space collision avoidance for mobile two-robot systems, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009 (IROS 2009) (2009) 1941–1948.
- [16] G. Antonelli, F. Arrichiello, S. Chiverini, The null-space-based behavioral control for autonomous robotic systems, *Intell. Serv. Robot.* 1 (1) (2008) 27–39.

- [17] G. Antonelli, F. Arrichiello, S. Chiaverini, Stability analysis for the null-space-based behavioral control for multi-robot systems, 47th IEEE Conference on Decision and Control, 2008 (CDC 2008) (2008) 2463–2468.
- [18] J. Lin, R.-J. Lian, Design of a grey-prediction self-organizing fuzzy controller for active suspension systems, Appl. Soft Comput. 13 (October (10)) (2013) 4162–4173.
- [19] I. Baturone, A. Gersnoviez, Á. Barriga, Neuro-fuzzy techniques to optimize an FPGA embedded controller for robot navigation, Appl. Soft Comput. 21 (August) (2014) 95–106.
- [20] S. Kar, S. Das, P.K. Ghosh, Applications of neuro fuzzy systems: a brief review and future outline, Appl. Soft Comput. 15 (2014) 243–259.
- [21] P. Rusu, E.M. Petriu, T.E. Whalen, A. Cornell, H.J. Spoelder, Behavior-based neuro-fuzzy controller for mobile robot navigation, IEEE Trans. Instrum. Meas. 52 (4) (2003) 1335–1340.
- [22] M.J. Mataric, Behaviour-based control: examples from navigation, learning, and group behaviour, J. Exp. Theor. Artif. Intell. 9 (2–3) (1997) 323–336.
- [23] E. Tunstel, Coordination of distributed fuzzy behaviors in mobile robot control, IEEE International Conference on Systems, Man and Cybernetics, 1995. Intelligent Systems for the 21st Century 5 (1995) 4009–4014.
- [24] P. Vadakkepat, O.C. Miin, X. Peng, T.H. Lee, Fuzzy behavior-based control of mobile robots, IEEE Trans. Fuzzy Syst. 12 (4) (2004) 559–565.
- [25] E. Aguirre, A. González, Fuzzy behaviors for mobile robot navigation: design, coordination and fusion, Int. J. Approx. Reason. 25 (3) (2000) 255–289.
- [26] K. Izumi, K. Watanabe, Fuzzy behavior-based control trained by module learning to acquire the adaptive behaviors of mobile robots, Math. Comput. Simul. 51 (3) (2000) 233–243.
- [27] A. Marino, F. Caccavale, L.E. Parker, G. Antonelli, Fuzzy behavioral control for multi-robot border patrol, 17th Mediterranean Conference on Control and Automation, 2009 (MED'09) (2009) 246–251.
- [28] S.K. Pradhan, D.R. Parhi, A.K. Panda, Fuzzy logic techniques for navigation of several mobile robots, Appl. Soft Comput. 9 (January (1)) (2009) 290–304.
- [29] R. Huq, G.K.I. Mann, R.G. Gosine, Mobile robot navigation using motor schema and fuzzy context dependent behavior modulation, Appl. Soft Comput. 8 (January (1)) (2008) 422–436.
- [30] D.R. Parhi, S.K. Pradhan, A.K. Panda, R.K. Behera, The stable and precise motion control for multiple mobile robots, Appl. Soft Comput. 9 (March (2)) (2009) 477–487.
- [31] D.R. Parhi, J.C. Mohanta, Navigational control of several mobile robotic agents using Petri-potential-fuzzy hybrid controller, Appl. Soft Comput. 11 (June (4)) (2011) 3546–3557.
- [32] P. Mahacek, C.A. Kitts, I. Mas, Dynamic guarding of marine assets through cluster control of automated surface vessel fleets, IEEEASME Trans. Mechatron. 17 (1) (2012) 65–75.
- [33] S. Yu, X. Ding, L. Li, F. Pan, S. Xiong, Y. Bai, Fuzzy PID-type iterative learning control for electro-hydraulic servo system, International Conference on Computational Aspects of Social Networks (CASoN) (2010) 671–674.
- [34] C.M. Wang, Location estimation and uncertainty analysis for mobile robots, Proceedings, 1988 IEEE International Conference on Robotics and Automation (1988) 1231–1235.
- [35] G. Antonelli, S. Chiaverini, G. Fusco, A calibration method for odometry of mobile robots based on the least-squares technique: theory and experimental validation, IEEE Trans. Robot. 21 (5) (2005) 994–1004.
- [36] H.-J. von der Hardt, P. Arnould, D. Wolf, M. Dufaut, A method of mobile robot localisation by fusion of odometric and magnetometric data, Int. J. Adv. Manuf. Technol. 9 (January (1)) (1994) 65–69.

**.2 Adaptive sliding-mode cluster space control
of a non-holonomic multi-robot system with
applications**

Adaptive sliding-mode cluster space control of a non-holonomic multi-robot system with applications

 ISSN 1751-8644
 Received on 26th August 2016
 Accepted on 28th September 2016
 doi: 10.1049/iet-cta.2016.1110
 www.ietdl.org

 Mohammad Tariq Nasir¹, Sami El-Ferik¹ ✉

¹Department of Systems Engineering, King Fahd University of Petroleum and Minerals, Dhahran, 31261, Saudi Arabia

✉ E-mail: selferik@kfupm.edu.sa

Abstract: The study proposes a novel adaptive and robust model-based sliding-mode controller for multi-robot system in cluster space. The study considers non-holonomic robots, which are commonly used in several real-life applications. The proposed controller is robust to uncertainties and external disturbances. Simulation and experimental tests show the potential of this approach. The experimental validation was done using Lego EV3 robots connected through Wi-Fi link.

1 Introduction

Cooperative control of multi-agent systems has attracted considerable research interest during the past few decades, owing to the augmented capabilities that such systems offer during automation tasks. Some of these capabilities include increased coverage, speed, repeatability, precision, redundancy and strength, as well as the ability to withstand extreme conditions [1]. This allows multi-robot systems to be used in many automation applications, including sensor deployment, scouting, fire fighting, rescue and recovery and military applications, as well as environmental protection and surveys, such as oil spill disasters.

Simultaneous motion coordination and formation control is one of the key challenges in multi-robot systems' behaviour. Formation control is crucial, especially when the sensors' coverage and capabilities are limited. In such a case, formation allows each robot to focus its sensors towards a certain portion of the area of interest [2]. For example, a robot-scout benefits from a formation by directing the sensors of each robot towards achieving a maximum coverage area [3].

From the literature, there are three main control frameworks used in robot formation: leader–follower, null space, and cluster space. The well-known leader–follower concept has been extensively studied to design control strategies for robot formation, where the follower robots should follow a virtual position relative to the leader (for examples, see [4–6]). Recently, several related research issues have been recently investigated. For instance, optimising the path planning in a leader–follower formation with obstacle avoidance and its suitability for real-time implementation has been presented in [7], model uncertainty was addressed in [8–10]. The null space approach is a task-based formation control concept, where each requirement is considered as a task. For example the spacing between the robot is considered as a task; the centre of the group is another task, and the robot target following is a separate task and so on. See [11–14].

The third control method uses the cluster space approach, where the group of robots are considered as one entity, called a cluster. This cluster has its own dynamic states (called the cluster space). The cluster states are a function of the robots' states (called the robot space). The control commands are calculated at the cluster level. Based on that, cluster commands are translated to robot space commands by applying inverse kinematics and using the specific Jacobean matrix. Thus, each robot has its own command derived from the cluster's command. Therefore, using the cluster space framework makes the control design simpler, as opposed to dealing with many robot entities as in the virtual-leader concept (see [15–19]). Many research questions were addressed in the literature; the obstacle avoidance problem was studied by [1, 16]; a behavioural intelligent controller was proposed by El Ferik *et al.*

[20]; the cluster space approach was applied to vessel control for a military purposes in [21–23], and these vessels have non-holonomic dynamics. However, designing a mode-based controller for a group of non-holonomic robots is still an active field of research.

Non-holonomic robots are robots that have constraints on their motion; for more see [24]. Most mobile robots are non-holonomic. For example, the two-wheel differential robot is considered a non-holonomic robot because it only moves towards the direction of its heading angle.

In the literature, the problem of multi-non-holonomic cluster formation control has been tackled by adding a fast inner control loop to change the robot-heading angle towards the desired motion profile, while the outer controller handles the formation and tracking tasks [1, 25]. However, having two control loops makes the system more complicated, gives a generally slower time response, and the outer controller always assumes that the robot is heading to the target, which is not always true-causing the tracking performance to decrease.

Moreover, the majority of the proposed controllers were velocity-based controllers; therefore, the acceleration is not considered as a state to be controlled, and this leads to neglecting the uncertainty in the robot's mass and inertia. To solve this issue a recent approach [23] proposed a dynamic-based controller for the cluster space approach, where the acceleration and dynamic model of the cluster space robots were considered in the controller design. In [23], a feedback linearisation algorithm was proposed. However, the proposed approach assumed that the robots were holonomic robots.

Control of single non-holonomic robots has seen extensive research activity during the past few years. On the other hand, cluster space control of multi-non-holonomic robots is still under investigation and, to our knowledge, a robust model-based control of such a system has not been addressed

A sliding mode controller (SMC) is a non-linear robust model-based controller, where the system dynamics are forced to stay on a stable surface. This surface is a function of the system states; to guarantee reaching the sliding surface and to address the uncertainty, a robustifying term is added to the controller algorithm. When designing a SMC for the non-holonomic robot motion, constraints should be considered; hence selecting the sliding surface is not a trivial problem. In [26], a SMC was developed for a non-holonomic robot in the polar coordinates. The use of polar coordinates can simplify the sliding surface selection; however, this controller has some singularity issues around the origin and adds constraints on the motion postures and velocities. In [27], these constraints were eliminated. However, the proposed controller is still unstable around the origin and it needs to

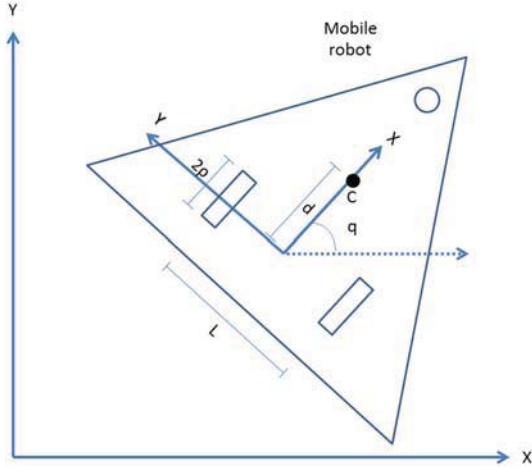


Fig. 1 WMR model, where q is the WMR's heading angle, point c is the robot's centre of gravity, d is the distance between the wheels line and c and p is the wheel radius

transform the robot coordination to a polar coordination, which is not commonly used in robotics. In order to tackle the singularity issue of the polar coordinate [28] proposed a SMC over the Cartesian coordinate. However, because of the sliding surface constraints, the control input has limitations on the mobile robot's movement. Recently, a modified version of the last controller was proposed by Lee *et al.* [29] where an approaching angle sliding surface was proposed to address the control constraints.

In this paper, the latest adaptive SMC [30, 31] is extended to a formation control of a group of non-holonomic robots using cluster control owing to its simple formation control principle feature. The non-holonomic robot considered in this research is commonly used in practice such as unmanned vessels, drones, and mobile robots, this type still needs more investigation in the area of multi-robot formation control. The proposed adaptive robust controller addresses the issue of model uncertainties, as well as the system's non-linear dynamics. Those issues are common in real-life applications where there are constraints on the robot's motion that is transformed into a non-linear behaviour, while the model parameters either time varying or are uncertain. SMC is a robust controller that has the ability to overcome uncertainties and disturbances. However, the closed-loop dynamic suffers from a known chattering problem caused by the signum function which is needed to satisfy the reachability condition. In our work, the error function $\text{erf}(\cdot)$ is used to approximate the signum function. Unlike saturation function, this approximation does not suffer from the boundary layer problem. However, the quality of the approximation depends on the selection of a design parameter (α). Targeting implementation of the algorithm on a real system, the present controller proposes PI-SMC to overcome uncertainties in parameters, uncertainties in the dynamic, as well as external disturbances. PI-SMC has been employed to increase the sensitivity and robustness of the controller to uncertainties and disturbance. Lego EV3 WMRs robots and MATLAB environment have been selected to implement the new control approach. EV3 is a very flexible platform, compatible with MATLAB, and equipped with Wi-Fi and Bluetooth links to connect with the central controller. It allows rapid prototyping.

The paper begins with presenting the cluster model of non-holonomic robots [23] and provides the necessary changes to include the non-holonomic case. The paper reports the experimental implementation of the proposed scheme on a real ground robot connected to a central controller using a Wi-Fi network.

The remainder of the paper is organised as follows: Section 2 introduces the kinematics and dynamics of a non-holonomic robot, and defines multi-robot cluster control. In Section 3, cluster dynamics is derived. In Section 4, the SMC is proposed. The simulation results are discussed in Section 4 and the experimental

validation and results are presented in Section 5. Section 6 concludes the paper and discusses about future work.

2 Preliminaries

In this section, fundamental definitions and mathematical models are presented. The Nonholonomic robot kinematic model is presented, followed by the mathematical representation of the cluster framework, after that the dynamical models for a single non-holonomic robot and the general cluster model with non-holonomic robots are presented.

2.1 Non-holonomic robot dynamics

We consider a wheeled mobile robot (WMR), with two driven wheels and a passive caster wheel, whose schematic model is shown in Fig. 1.

The state-space model of the considered kinematic vehicle with the associated non-holonomic constraints (rolling with no slipping) is given by (1), where v and w are the heading and rotational velocities variables:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} \cos(q) & 0 \\ \sin(q) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix}. \quad (1)$$

2.2 Overview of cluster space framework

In order to implement the cluster space for two robots, an appropriate set of cluster variables are chosen to represent the shape of the cluster. As shown in Fig. 2, the proposed cluster variables are $(Q_c, x_c, y_c, d_c, q_1, q_2)$ and the corresponding robot space are (x_1, y_1, q_1) and (x_2, y_2, q_2) . The following (2) show the relation between the cluster space and the robot space variables, which can be presented as $c = f(r)$. Similar developments are found in [19].

$$\begin{aligned} x_c &= \frac{x_1 + x_2}{2}, \\ y_c &= \frac{y_1 + y_2}{2}, \\ Q_c &= \tan_2^{-1}(y_1 - y_2, x_1 - x_2) + \frac{\pi}{2}, \\ d_c &= \frac{1}{2}\sqrt{(y_1 - y_2)^2 + (x_1 - x_2)^2}, \\ q_1 &= q_1, \\ q_2 &= q_2. \end{aligned} \quad (2)$$

Where (x_c, y_c) is the centre of the cluster; Q_c is the cluster heading and d_c is the spacing between the robots from the cluster centre. The corresponding Jacobian matrix is:

$$J = \begin{bmatrix} 0.5 & 0 & 0 & 0.5 & 0 & 0 \\ 0 & 0.5 & 0 & 0 & 0.5 & 0 \\ \frac{-(y_1 - y_2)}{\epsilon_1} & \frac{(x_1 - x_2)}{\epsilon_1} & 0 & \frac{(y_1 - y_2)}{\epsilon_1} & \frac{-(x_1 - x_2)}{\epsilon_1} & 0 \\ \frac{(x_1 - x_2)}{\epsilon_2} & \frac{(y_1 - y_2)}{\epsilon_2} & 0 & \frac{-(x_1 - x_2)}{\epsilon_2} & \frac{-(y_1 - y_2)}{\epsilon_2} & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\text{where } \epsilon_1 = (x_1 - x_2)^2 + (y_1 - y_2)^2 \quad \text{and} \\ \epsilon_2 = 2\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}.$$

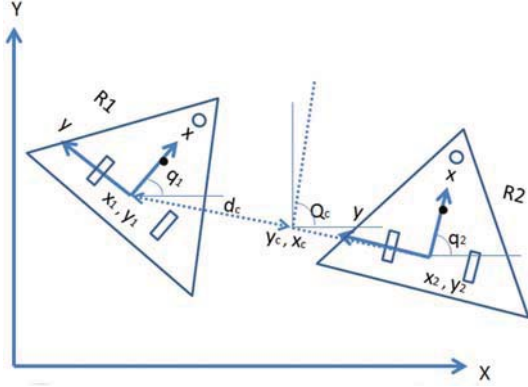


Fig. 2 Cluster space and robot space variables

3 Cluster space dynamics of non-holonomic robots

In this section, the model representation of the cluster of non-holonomic robots will be derived. In particular, the coupling between the cluster states will be highlighted, owing to its importance in the overall control design.

3.1 Single non-holonomic robot modelling

The starting point will be the classical modelling of a single non-holonomic robot in (3), this model is a modified version of [32] model. Since the local axis is relocated in the modified version to be in the centre between the wheels. Followed by a generalisation to address the case of n robot and the cluster space model.

$$M(r)\ddot{r} + b(r, \dot{r}) + g(r) + \tau_d = B(r)\tau - A^T(r)\lambda. \quad (3)$$

According to Fig. 1, $r = [x, y, q]^T$ and $M(r) \in \mathbb{R}^{3 \times 3}$ denotes the positive-definite symmetric inertia matrix; $b(r, \dot{r}) \in \mathbb{R}^{3 \times 3}$ is a combination of Coriolis, centripetal and friction terms; $g(r) \in \mathbb{R}^{3 \times 1}$ represents the gravitational forces; τ_d is the bounded unknown disturbances, and $\tau \in \mathbb{R}^{2 \times 1}$ is the motors' torques. $A(r) \in \mathbb{R}^{1 \times 3}$ represents the constraint matrix that is multiplied with the Lagrange multiplier $\lambda \in \mathbb{R}^{1 \times 1}$, and the constraint equation is $A(r)\dot{r} = 0$, where

$$M(r) = \begin{bmatrix} m & 0 & -m\sin(q) \\ 0 & m & m\cos(q) \\ -m\sin(q) & m\cos(q) & I + md^2 \end{bmatrix},$$

$$b(r, \dot{r}) = \begin{bmatrix} 0 & 0 & m\dot{q}^2\cos(q) \\ 0 & 0 & m\dot{q}^2\sin(q) \\ 0 & 0 & 0 \end{bmatrix}, \quad g(r) = 0,$$

$$B(r) = \frac{1}{2\rho} \begin{bmatrix} \cos(q) & \cos(q) \\ \sin(q) & \sin(q) \\ L & -L \end{bmatrix},$$

$$\tau = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \text{ and } A^T(r) = \begin{bmatrix} -\sin(q) \\ \cos(q) \\ 0 \end{bmatrix}.$$

Let $S_r(r) \in \mathbb{R}^{3 \times 2}$ be a full rank matrix such that $S_r^T(r)A^T(r) = 0$

$$\text{such that } S_r(r) = \begin{bmatrix} \cos(q) & 0 \\ \sin(q) & 0 \\ 0 & 1 \end{bmatrix}.$$

In the case of having $i \in i = 1, 2, \dots, n$ robots we have different models such that:

$$M_i(r_i)\ddot{r}_i + b_i(r_i, \dot{r}_i) + g_i(r_i) + \tau_d = B_i(r_i)\tau_i - A_i^T(r_i)\lambda_i.$$

IET Control Theory Appl.

© The Institution of Engineering and Technology 2016

$S_{r_i}(r_i) \in \mathbb{R}^{3 \times 2}$ be a full-rank matrix such that $S_{r_i}^T(r_i)A_i^T(r_i) = 0$.

3.2 Cluster space modelling

To select the cluster space states, several conditions should be considered. The cluster states should describe the function of the application, such as formation control. The number of cluster degrees of freedom (DOF) should be equal to the number of the robot space DOF. The cluster dynamics can be calculated by transforming the robot space dynamics through the Jacobian matrix by $\dot{c} = J(r)\dot{r}$.

Cluster space dynamics Starting from the robot space dynamics with n robots, (4) is given as.

$$\overline{M}(r)\ddot{r} + \overline{b}(r, \dot{r}) + \overline{g}(r) + \tau_d = \overline{B}(r)\tau - \overline{A}^T\lambda, \quad (4)$$

where

$$\overline{M}(r) = \begin{bmatrix} M_1 & 0^{3 \times 3} & \dots & 0^{3 \times 3} \\ \cdot & M_2 & 0^{3 \times 3} & \vdots \\ \cdot & \cdot & \ddots & 0^{3 \times 3} \\ \cdot & \cdot & \cdot & M_n \end{bmatrix},$$

$$\overline{b}(r, \dot{r}) = \begin{bmatrix} b_1(r_1, \dot{r}_1) \\ \vdots \\ b_n(r_n, \dot{r}_n) \end{bmatrix},$$

$$\overline{B}(r) = \begin{bmatrix} B_1 & 0^{3 \times 2} & \dots & 0^{3 \times 2} \\ \cdot & B_2 & 0^{3 \times 2} & \vdots \\ \cdot & \cdot & \ddots & 0^{3 \times 2} \\ \cdot & \cdot & \cdot & B_n \end{bmatrix},$$

$$\overline{A}^T(r) = \begin{bmatrix} A_1^T & 0^{3 \times 1} & \dots & 0^{3 \times 1} \\ \cdot & A_2^T & 0^{3 \times 1} & \vdots \\ \cdot & \cdot & \ddots & 0^{3 \times 1} \\ \cdot & \cdot & \cdot & A_n^T \end{bmatrix},$$

$$\overline{\lambda} = [\lambda_1 \dots \lambda_n]^T,$$

$$\overline{\tau} = [\tau_1 \dots \tau_n].$$

Starting with robot space dynamics, the holonomic cluster dynamics in (5) were derived by Mas and Kitts[23]; based on that the non-holonomic robot cluster dynamics are found in (6), and the coupling between the cluster states can be represented in $\mu(c, \dot{c})$

$$\Lambda(c)\dot{c} + \mu(c, \dot{c}) + p(c) + \tau_d = \beta_h(c)\overline{\tau}, \quad (5)$$

$$\Lambda(c)\dot{c} + \mu(c, \dot{c}) + p(c) + \tau_d = \beta(c)\overline{\tau} - \alpha^T(c)\overline{\lambda}, \quad (6)$$

where

$$\Lambda(c) = J^{-T}(r)\overline{M}(r)J^{-1}(r),$$

$$\mu(c, \dot{c}) = J^{-T}(r)\overline{b}(r, \dot{r}) - \Lambda(c)j(r, \dot{r})\dot{r},$$

$$p(c) = J^{-T}(r)g(r), \quad \beta(c) = J^{-T}(r)\overline{B}(r),$$

$$\alpha^T(c) = J^{-T}(r)\overline{A}^T(r),$$

and the constraint equation will be $\alpha^T(c)\dot{c} = 0$; let

$$S_c(c) = J \cdot \begin{bmatrix} S_{r1} & 0^{3 \times 2} & \dots & 0^{3 \times 2} \\ \cdot & S_{r2} & 0^{3 \times 2} & \vdots \\ \cdot & \cdot & \ddots & 0^{3 \times 2} \\ \cdot & \cdot & \cdot & S_{rn} \end{bmatrix},$$

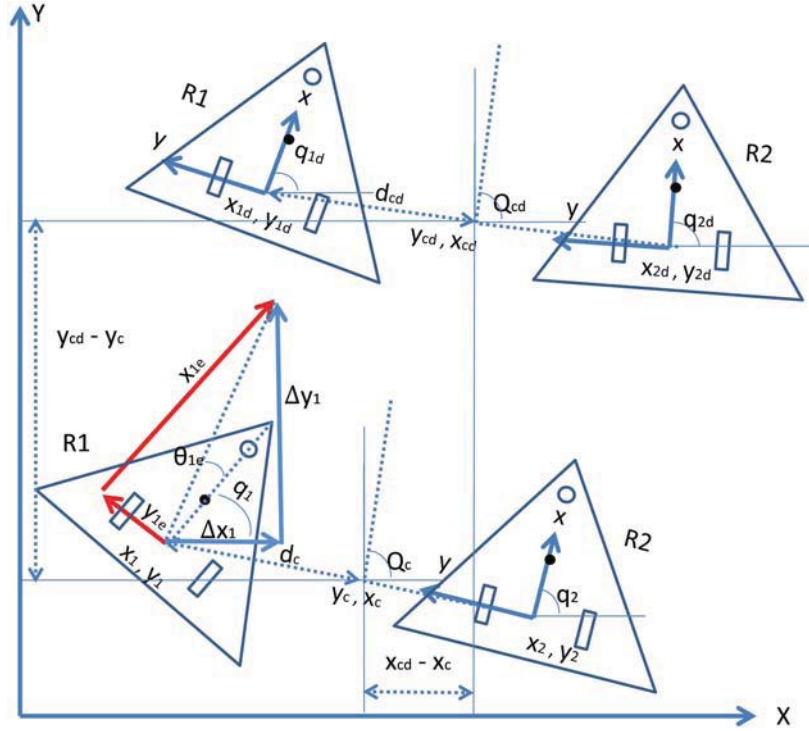


Fig. 3 Cluster location errors

such that $S_c^T(c)\alpha^T(c) = 0$ and accordingly it is possible to find an auxiliary $V(t)$ such that $\dot{c} = S_c(c)V(t)$, where $V(t) = [v_1, w_1, \dots, v_n, w_n, \dots, v_n, w_n]$, and v_i and w_i are the heading and the angular velocities of the non-holonomic robots.

Following the derivation we get (7)

$$\dot{c} = S_c(c)\dot{V}(t) + \dot{S}_c(c)V(t), \quad (7)$$

and by multiplying both sides of (7) with $S_c^T(c)$ we arrive at (8)

$$H\dot{V} + E + \tau_d = \bar{\tau}, \quad (8)$$

where

$$\begin{aligned} H &= (S_c^T(c)\beta(c))^{-1}S_c^T(c)\Lambda(c)S_c(c), \\ E &= (S_c^T(c)\beta(c))^{-1}S_c^T(c)(\Lambda(c)\dot{S}_c(c)V(t) + \mu(c, \dot{c}) + P(c)), \\ \tau_d &= Hf, f \in \mathbb{R}^{2 \cdot n \times 1}, f = [f_{11}, f_{21}, f_{12}, f_{22}, \dots, f_{1n}, f_{2n}]. \end{aligned}$$

Now we have a reduced order dynamic equation for the cluster space with no Lagrange multiplier term.

4 Controller design

In this section, the adaptive SMC is presented. The error signal calculation, then the sliding surface selection, followed by the adaptive SMC and the controller diagram. This development is done for two-robot cluster as an example, which the same procedure is done for a cluster with any number of robots.

4.1 Cluster profile errors

The formation error in the cluster spaces are found as the following. Starting with defining $c = (Q_c, x_c, y_c, d_c, q_1, q_2)$, $c_d = (Q_{cd}, x_{cd}, y_{cd}, d_{cd}, q_{1d}, q_{2d})$, where c_d is the desired cluster spaces and c is the actual cluster spaces. The error signal is

$\Delta c = W(c_d - c)$, where W is a positive weighting diagonal matrix. The robot space commands are transformed from the cluster space signals to robot space signals by multiplying it with the inverse of the Jacobian matrix, as $\bar{\Delta r} = J^{-1}\Delta c$, where $\bar{\Delta r} = [\Delta x_1, \Delta y_1, \Delta \theta_1, \Delta x_2, \Delta y_2, \Delta \theta_2]^T$ and $\bar{\Delta r}_i = [\Delta x_i, \Delta y_i, \Delta \theta_i]^T$, in order to deal with the non-holonomic constraints the robot space commands are modified as the following: $\Delta r = [\Delta x_1, \Delta y_1, \theta_{1e}, \Delta x_2, \Delta y_2, \theta_{2e}]^T$ and $\Delta r_i = [\Delta x_i, \Delta y_i, \theta_{ie}]^T$, where $\theta_{ie} = \tan^{-1}(\Delta y_i / \Delta x_i) - q_i$ then a transformation of the robot commands from a global frame to a robot frame is done by using the following rotational transformation:

$$r_{ie} = \begin{bmatrix} x_{ie} \\ y_{ie} \\ \theta_{ie} \end{bmatrix} = \begin{bmatrix} \cos(q_i) & \sin(q_i) & 0 \\ -\sin(q_i) & \cos(q_i) & 0 \\ 0 & 0 & 1 \end{bmatrix} \Delta r_i$$

where $(x_{ie}, y_{ie}$ and $\theta_{ie})$ are shown in Fig. 3. Then by defining $V_i = [v_i, w_i]^T$, $V_{id} = [v_{id}, w_{id}]^T$ that the v_i, w_i are the actual robot heading and rotational velocities and v_{id}, w_{id} are the desired robot heading and rotational velocities. It was found in [30–32] that the derivative of the robot profile errors can be found by (9)

$$\dot{r}_{ie} = \begin{bmatrix} \dot{x}_{ie} \\ \dot{y}_{ie} \\ \dot{\theta}_{ie} \end{bmatrix} = \begin{bmatrix} y_{ie}w_i - v_i + v_{id}\cos(\theta_{ie}) \\ -x_{ie}w_i + v_{id}\sin(\theta_{ie}) \\ w_{id} - w_i \end{bmatrix}. \quad (9)$$

4.2 Sliding-mode derivation

In this section, a model-based SMC is developed. The first step in designing an SMC is selecting a sliding surface where the system dynamics are stable; this sliding surface is a function of the system states.

For Designing the sliding surface for a single robot: Starting, from the classical kinematic controller given by Chen *et al.* [33], we get the following velocity controller in (10):

$$v_c = \begin{bmatrix} v_{1c} \\ w_{1c} \\ \vdots \\ v_{nc} \\ w_{nc} \end{bmatrix} = \begin{bmatrix} v_{1d} \cos(\theta_{1e}) + k_{11} x_{1e} \\ w_{1d} + k_{12} v_{1d} y_{1e} + k_{13} v_{1d} \sin(\theta_{1e}) \\ \vdots \\ v_{nd} \cos(\theta_{ne}) + k_{n1} x_{ne} \\ w_{nd} + k_{n2} v_{nd} y_{ne} + k_{n3} v_{nd} \sin(\theta_{ne}) \end{bmatrix}, \quad (10)$$

where $k_{i1}, k_{i2}, k_{i3} > 0, i | i = 1, \dots, n$. Thus, the error in the robots' kinematics is defined as:

$$e_c(t) = [e_{c1}(t), e_{c2}(t), \dots, e_{cn}(t)]^T = v_c(t) - v(t),$$

$$\dot{e}_c(t) = \dot{v}_c(t) - \dot{v}(t),$$

where

$$v = \begin{bmatrix} v_1 \\ w_1 \\ \vdots \\ v_n \\ w_n \end{bmatrix}.$$

Then a PI-type sliding surface is selected, as in (11). The adaptive integral component offers a fast convergence of the sliding surface than the normal SMC and a smooth control of the system resulting in zero steady-state error [34].

$$s(t) = \begin{bmatrix} s_1(t) \\ s_2(t) \\ \vdots \\ s_{1,i}(t) \\ s_{2,i}(t) \\ \vdots \\ s_{1,n}(t) \\ s_{2,n}(t) \end{bmatrix} = e_c(t) + \beta \int e_c(t) dt, \quad (11)$$

where β is a positive integer. Hence, if the system is on the sliding surface

$s(t) = 0, e_c(t) = -\beta \int_0^t e_c(t) dt$ and if $t \rightarrow \infty$ then $e_c \rightarrow 0$. In order to have the control signal, the derivation of the sliding surface is found in (12).

$$\dot{s}(t) = [\dot{v}_c(t) - H^{-1}(\bar{\tau} - E)] + \beta e_c(t) = 0. \quad (12)$$

Then by rearranging (12) we arrive at (13)

$$\tau_{eq} = H[\dot{v}_c(t) + \beta e_c(t)] + E. \quad (13)$$

Now τ_{eq} can keep the system inside the surface. But what if the system dynamics are already outside the surface? In order to guarantee the stability in that case, another control signal τ_r should push the system dynamics onto the sliding surface; this mechanism is called the reachability law or a robustifying term. Now the new control signal τ is a combination of $\tau_{eq} + \tau_r$, as shown in (14)

$$\tau_{eq} = H[\dot{v}_c(t) + \beta e_c(t) + K \cdot \text{sgn}(s)] + E, \quad (14)$$

where

$$K = \begin{bmatrix} K_1 & 0 & \dots & 0 \\ 0 & \ddots & 0 & 0 \\ \vdots & 0 & K_i & \vdots \\ 0 & 0 & 0 & K_n \end{bmatrix} \quad \{K_i | K_i > 0\}$$

and the $\text{sgn}(s(t)) = [\text{sgn}(s_1(t)), \text{sgn}(s_2(t)), \dots, \text{sgn}(s_{2,n}(t))]$

The uncertainty of the system is a function in the disturbance τ_d and the uncertainty in the model itself, such that:

$$\delta(t) = \Delta H^{-1}(\tau - E) + H^{-1}(-\Delta E) + H^{-1}\tau_d.$$

Now the dynamic of the system can be written as in (15)

$$\dot{v} = H^{-1}(\tau - E) + \delta. \quad (15)$$

However, using the signum function in the reaction law is not preferred for practical work, and this is due to the issue of chattering. Therefore, the error function $\text{erf}(s)$ is used instead.

$$\tau_{eq} = H[\dot{v}_c(t) + \beta e_c(t) + K \cdot \text{erf}(s)] + E. \quad (16)$$

The error function has the following form:

$$\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^{\alpha x} \exp(-a^2) da.$$

The function has the ability to approximate the $\text{sign}(x)$ function, while guaranteeing a smooth dynamic that allows for a reduction of chattering. The more the gain is selected high the closer is the function to $\text{sign}(x)$. One should note that $|\text{erf}(x)|$ is approximately equal to 1 when $|x| \geq 2/\alpha$. In sliding-mode control, the signum allows the error state vector to reach the sliding surface in a finite time. Thus it is only active to bring the error state dynamic to the sliding surface. Once on the surface, its value is defined as zero and has no impact on the error dynamic to reach zero. The error function ensures the same reachability condition with a smoother behaviour close to the sliding surface. Its use do not impact the convergence of the error dynamic to zero for the function is vanishes at zero.

The advantage of PI sliding mode is its ability to drive the error to zero by driving the states to the sliding surface. In [34], PI-SMC provides the ability to overcome the error caused by approximating the signum function by a saturation function. This is called the boundary layer technique known to reduce chattering. He also needs such controller to increase the robustness and insensitivity to uncertainty.

In this work, the use of the $\text{erf}(\cdot)$ to approximate the signum function does not suffer from the boundary layer problem as the function is smoother. However, the fact that the switching depends on the selection of the parameter as explained previously. As we are seeking implementation of the algorithm on a real system, the controller has to be able to overcome uncertainties in parameters, uncertainties in the dynamic as well as external disturbances. Therefore, in order to increase the sensitivity and robustness of the controller to uncertainties and disturbance, PI has been employed.

4.3 Adaptive sliding mode

Theorem 1: Assuming the adaptive law as in (17)

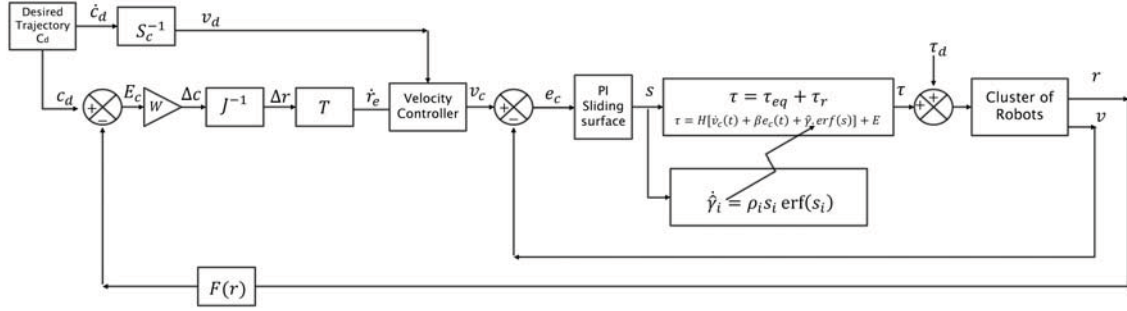


Fig. 4 Proposed adaptive SMC diagram

$$\dot{\gamma} = \begin{bmatrix} \dot{\gamma}_1 & 0 & \cdots & 0 \\ 0 & \dot{\gamma}_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \dot{\gamma}_i \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \dot{\gamma}_n \end{bmatrix} = \begin{bmatrix} \rho_1 s_1 \text{erf}(s_1) & 0 & \cdots & 0 \\ 0 & \rho_2 s_2 \text{erf}(s_2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \rho_n s_n \text{erf}(s_n) \end{bmatrix}, \quad (17)$$

where $\rho_i > 0$ and the final controller signal is in (18)

$$\tau_{eq} = H[\dot{v}_c(t) + \beta e_c(t) + \dot{\gamma} \cdot \text{erf}(s)] + E. \quad (18)$$

The estimate error is define as $\dot{\gamma}(t) = \tilde{\gamma}(t) - \gamma^*$.

Proof: Using the Lyapunov function: let $L = L_1 + L_2 + L_3$ such that $L(0) = L_1(0) = L_2(0) = L_3(0) = 0$ and $L_1, L_2, L_3 > 0$ for inputs other than 0, where

$$L_1 = \frac{1}{2} \sum_{i=1}^n (x_{ie}^2 + y_{ie}^2 + \frac{1 - \cos(\theta_{ie})}{k_{i2}}),$$

$$L_2 = \frac{1}{2} s^T(t) s(t) + \frac{1}{2} \sum_{i=1}^n (\frac{1}{\rho_i} \tilde{\gamma}_i^2),$$

$$L_3 = E_c^T E_c.$$

Based on the Lyapunov theory, the system is stable if and only if $\dot{L} < 0$. After derivation we get

$$\dot{L}_1 = \sum_{i=1}^n (-k_{i1} x_{ie}^2 - \frac{k_{i3} v_{id} \sin^2(\theta_{ie})}{k_{i2}}),$$

$$\dot{L}_2 = s^T \dot{s} + \sum_{i=1}^n (\frac{1}{\rho_i} \tilde{\gamma}_i \dot{\tilde{\gamma}}_i).$$

Now $\dot{L}_1 < 0$, and after substituting $\dot{\gamma}$ based on $\dot{\tilde{\gamma}} = \dot{\gamma} - \dot{\gamma}^*$.

$$\dot{L}_2 = s^T [-\dot{\gamma} \text{erf}(s) - \delta] + \sum_{i=1}^n (\frac{1}{\rho_i} \tilde{\gamma}_i \dot{\tilde{\gamma}}_i),$$

$$\dot{L}_2 = s^T [-(\tilde{\gamma} + \gamma^*) \text{erf}(s) - \delta] + \sum_{i=1}^n (\frac{1}{\rho_i} \tilde{\gamma}_i \dot{\tilde{\gamma}}_i),$$

$$\dot{L}_2 = s^T [-\gamma^* \text{erf}(s) - \delta] + \sum_{i=1}^n [\tilde{\gamma}_i (\frac{1}{\rho_i} \dot{\tilde{\gamma}}_i - s_i \text{erf}(s_i))].$$

Once the adaptive equation is substituted the resultant \dot{L}_2 is given as

$$\dot{L}_2 = s^T [-\gamma^* \text{erf}(s) - \delta],$$

$$\dot{L}_2 = \sum_{i=1}^n s_i [-\gamma_i^* \text{erf}(s_i) - \delta_i] \leq -\sum_{i=1}^n |\delta_i| s_i \text{erf}(s_i) (\gamma_i^* - |\delta_i|).$$

The $\dot{L}_2 < 0$ can be guaranteed by selecting $\gamma^* > |\delta|$.

The L_3 derivation is

$$\dot{L}_3 = E_c^T \dot{E}_c,$$

where $E_c = (c_d - c)$, $\dot{E}_c = (\dot{c}_d - \dot{c})$ and $\dot{c} = WE_c$. By substitution we the arrive at:

$$\dot{E}_c = (\dot{c}_d - WE_c),$$

$$\text{Then } \dot{L}_3 = (E_c^T \dot{c}_d - E_c^T WE_c).$$

Knowing that $-E_c^T WE_c < -\lambda_{\min}(W) \|E_c\|^2$, the $\dot{L}_3 < 0$ provided that

$$\|E_c\|^T < \frac{\|\dot{c}_d\|}{\lambda_{\min}(W)}.$$

As a result, the derivative of the Lyapunov function is $\dot{L} < 0$ and the stability is proved. In addition, since the reachability condition is satisfied, the tracking dynamic will reach the sliding surfaces in a finite time. Thus the sliding variable $s_i \rightarrow 0$ as $t \rightarrow \infty$ and consequently $\dot{\gamma}_i \rightarrow 0$, for $i = 1, \dots, n$. Accordingly, $e_c \rightarrow 0$ and γ_i is bounded and reaches steady-state. This completes the proof. \square

4.4 Control diagram

In this subsection, the control diagram is presented. Fig. 4 shows the adaptive SMC.

5 Simulation results

5.1 Two-robot simulation

Based on the control law established in Section 4.3, a simulation using MATLAB@ is implemented on the cluster of two mobile robots. The cluster space of two non-holonomic robots was used. The sinusoidal desired trajectory was applied as follows:

$$x_{cd} = 5 \cos(0.3t),$$

$$y_{cd} = 15 \sin(0.1t),$$

$$Q_{cd} = \tan^{-1}(\frac{dx_{cd}}{dy_{cd}}).$$

Let $K = [10, 2, 10, 2]$ and the initial position of the mobile robots are set as $(x_1, y_1, q_1) = (1, 0, 0)$ and $(x_2, y_2, q_2) = (0, 0, 0)$. The simulation results are shown in Figs. 5a and b. Fig. 5a illustrates the trajectory tracking result for the sinusoidal function. The actual trajectory reaches the desired line quickly. In Fig. 5b, a disturbance

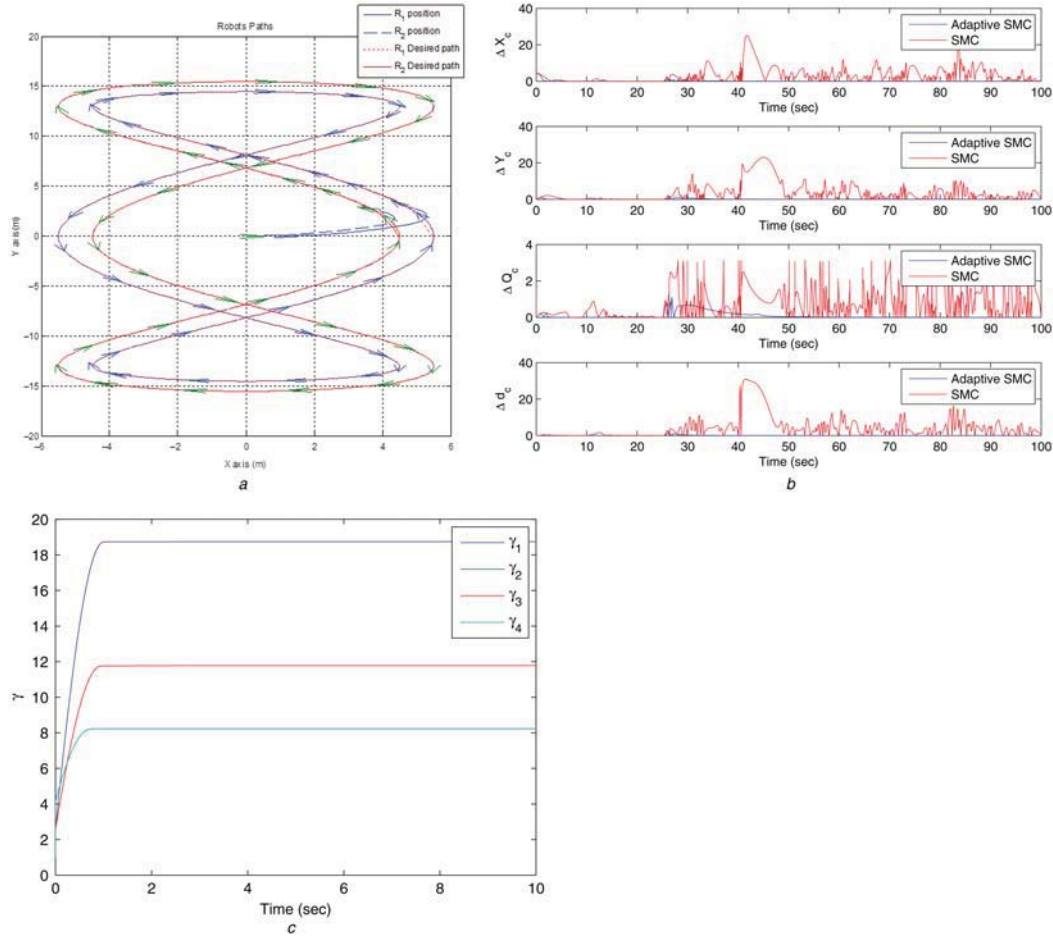


Fig. 5 Cluster space control of a two-robot system

(a) Adaptive SMC control path-tracking profile, (b) Comparison between SMC and adaptive SMC with disturbance injected after 20 s, (c) Adaptive parameters response

was introduced after 20 s; the results of the SMC and adaptive SMC were compared by comparing the tracking error in the cluster states x_c, y_c, Q_c , and d_c , respectively. In Fig. 5c, the adaptive parameter's responses are shown changing in the transient zone, the adapting velocity depends on the adaptive constants. According to the simulation results, the sinusoidal trajectory tracking shows good performance, especially with the adaptive term. This validates the sliding control algorithm by simulation.

5.2 Simulation results of three-robot system

In this subsection, a three-robot cluster was simulated with the same reference trajectory presented in Section 5.1. The selected cluster spaces are $(x_c, y_c, Q_c, \beta_c, q_c, p_c, q_1, q_2, q_3)$; see Fig. 6a and for more details see [17]. The simulation results (see Fig. 6b) show good performance of the proposed algorithm.

6 Experimental application

Lego EV3 WMRs were used in the experiments to validate and implement the control strategy. The WMRs (see Fig. 7) are equipped with a 32-bit, 48 Mhz ARM9 CPU with 16 MB flash memory and 64 MB RAM, Bluetooth and Wi-Fi transceivers, and two servo motors with encoders with 1 degrees of resolution. A PC interface with SIMULINK program is also required to transmit the control signal by means of Wi-Fi protocol. SIMULINK has a powerful feature called External Mode. This feature is useful for online monitoring and tuning of the EV3 WMR's controller. A two-

level control structure is used: high-level and low-level. The high-level controller is the SMC, performed in the central PC, which sends and receives command/data to and from the WMRs low-level controller. The low-level controller is a PID inner loop for controlling wheel speeds.

Based on the proposed control algorithm, the central PC receives the location feedback from each robot; then the PC calculates the error and control signals and sends the velocity commands to each robot. The low-level controller on the robot receives the commands from the PC and relays these signals to the motors. The encoders provide measurements for the feedback. The actual time of a one-loop process depends on the robot sampling time (set to 25 ms) plus the WiFi delay time, which it is dependent on the computer speed and network usage. The robot localisation is achieved by using the encoders only. The WMRs use EV3 servo motors that have a gear reduction mechanism in order to increase the torque and decrease the maximum output speed. However, this gear mechanism has a backlash issue, which introduces a non-linear behaviour due to small gaps between the mating gear teeth. Once the servo motor changes its direction the backlash effect occurs, causing the servo to have a certain rotation without being translated to actual wheel rotation. This issue can be mitigated by adding backlash compensation. Thus, when the servo motor changes its rotation direction a certain value is subtracted from the encoder reading.

Practical issues: The output from the centralised controller is the desired wheel's torque, but due to the braking behaviour of the EV3 servo motor, (when the motor has zero input) the wheel will

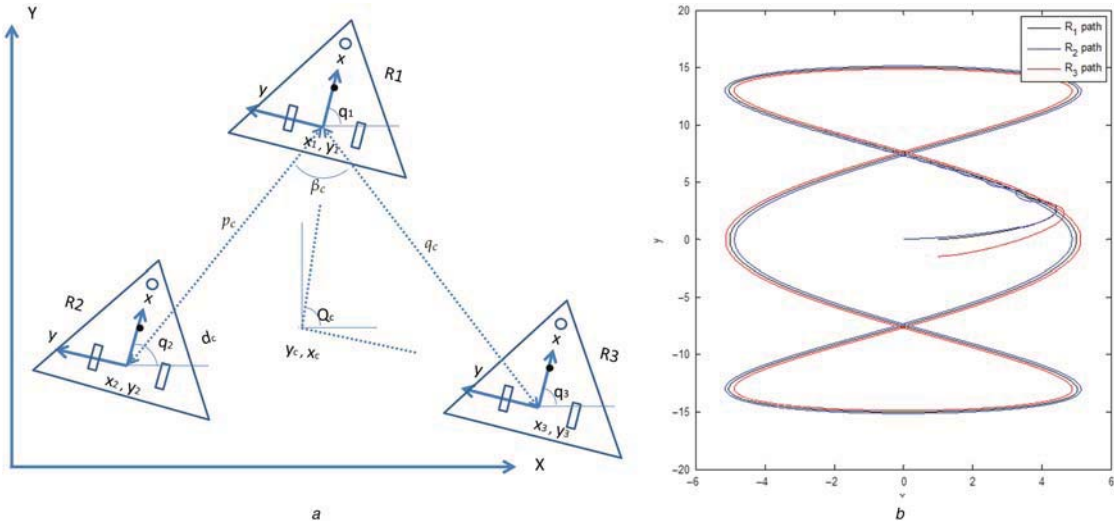


Fig. 6 Cluster space control of a three-robot system
 (a) Three-robot system configuration, (b) Adaptive SMC control path-tracking profile

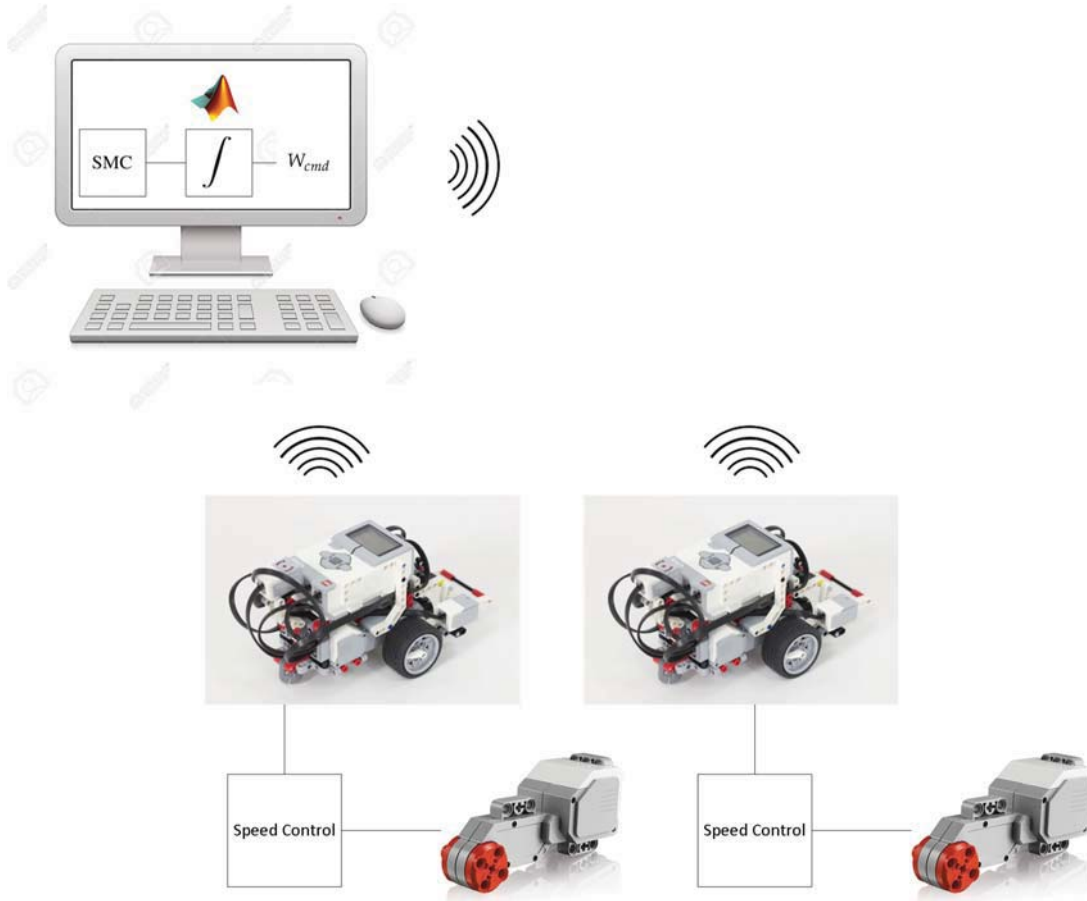


Fig. 7 Using an integrator to change an SMC command to a velocity command calculation and the inner loop diagram

stop abruptly. This braking is an additional variable that is not considered when developing the SMC. To improve the response the following solution has been used: i. add an integrator filter to the controller output torque, and the output signal of this filter is

then considered to be a velocity command; ii. instead of using a voltage or power signal to control the servo motor, the velocity command is sent to an inner speed loop controller. Accordingly, the servo motor will follow a torque equal to the sliding-mode torque

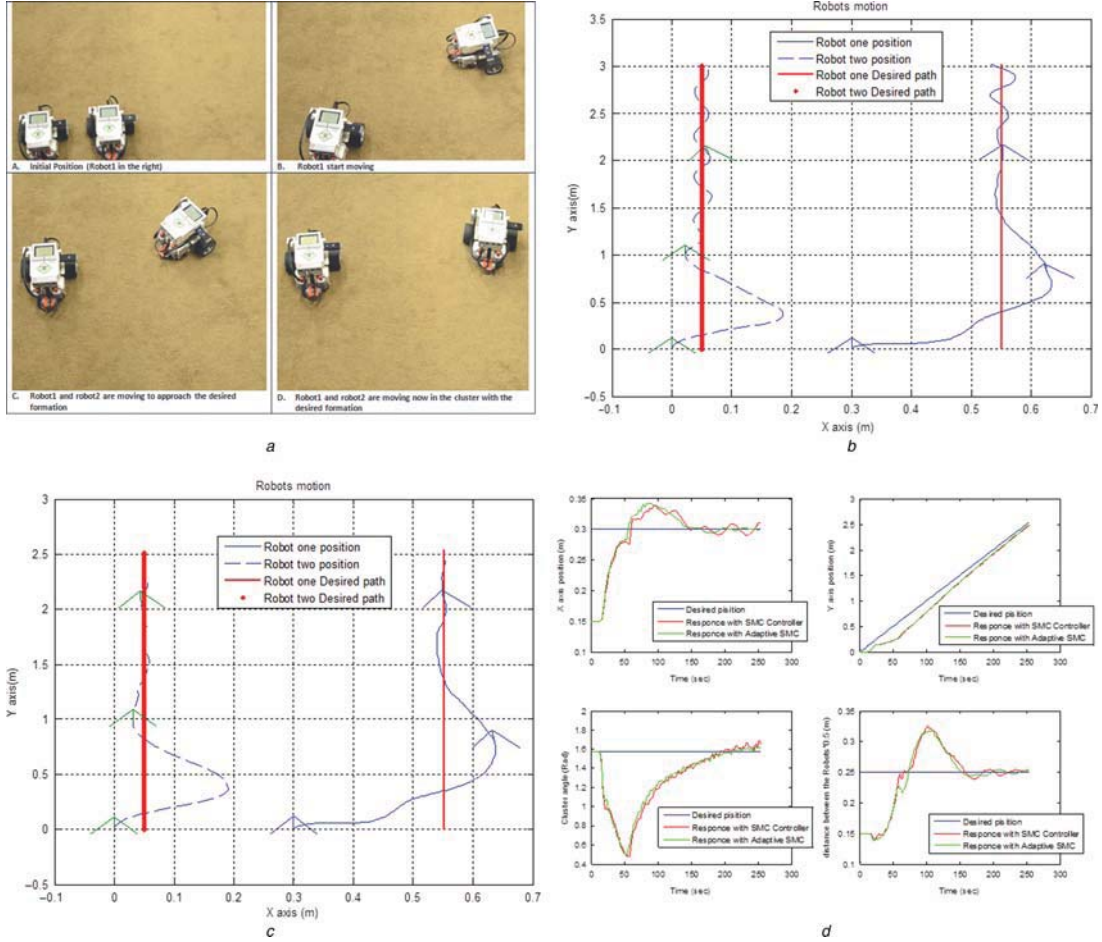


Fig. 8 Experimental results two-robot system
 (a) Experimental cluster space control on a two-robot system, (b) Experimental movement trajectory of a cluster of two robot with SMC, (c) Experimental movement trajectory of a cluster of two robot with adaptive SMC, (d) Experimental desired and actual cluster dynamics states x_c, y_c, Q_c and d_c , and a comparison between the SMC and the adaptive SMC

multiplied by some design constant, which is found through trial and error. For more, see Fig. 7, where W_{cmd} is the velocity command, s is the constant found by trial and error, and W_c is the actual wheel rotational velocity.

6.1 Experiment test

Two WMRs move from initial positions (x, y, q) , which for the first robot is $(0.2, 0, \pi/2)$ and $(0, 0, \pi/2)$ for the second robot. The desired path is $Q_c = \pi/2, x_c = 0.3, y_c = 0.1t$ and $d_c = 0.25$ in the cluster space, which is equivalent to $x_1 = 0.55, y_1 = 0.1t, q_1 = \frac{\pi}{2}, x_2 = 0.05, y_2 = 0.1t$ and $q_2 = \pi/2$ in the robot space, where t is the time (see Fig. 8a). The dynamic parameters are assumed to be nominal values. Hence, the robot mass is 0.5 kg, the robot inertial is 0.0025 kg.m², the wheel diameter is 56 mm, and the distance between the wheels is 11.8 cm.

The experimental results are shown Figs. 8b and c. Fig. 8b illustrates the trajectory tracking result for a line trajectory with SMC. The actual trajectory reaches the desired line quickly; however, chattering can be easily seen in the actual robots' trajectory. This chattering was due to the uncertainty of the robots' model and the untuned controller parameters. Fig. 8c shows the actual trajectory with the adaptive SMC, and the chattering effects are eliminated with the adaptive controller. Fig. 8d shows the comparison between the SMC and the adaptive SMC based on the

cluster states tracking x_c, y_c, Q_c and d_c , respectively. According to the experimental results for the line trajectory tracking, the trajectory can remain stable and robust despite uncertainty in the robot's inertia and mass. This validates the proposed algorithms experimentally.

7 Conclusion

This paper proposed a model-based multi non-holonomic robot controller in the cluster space by developing an adaptive sliding-mode control algorithm. This includes the presentation of the mathematical model of the non-holonomic multi-robot cluster, and the design of the sliding surface function and of the control law. According to the simulation and experimental results, the proposed adaptive sliding mode control is an important method to deal with a cluster of multi-robots in which uncertainties and non-linearities exist. In spite of large highly non-linear dynamics, the robots' cluster shows that the posture converges to the desired trajectory. Future work may investigate clusters with a larger number of heterogeneous robots and apply intelligent methods such as neural networks and fuzzy logic to overcome the singularities in the cluster dynamics, in addition to addressing the obstacle avoidance problem.

8 Acknowledgments

The author(s) would like to acknowledge the support provided by King Abdulaziz City for Science and Technology (KACST) through the Science & Technology Unit at King Fahd University of Petroleum & Minerals (KFUPM) for funding this work through project no. 11-ELE2147-04. as part of the National Science, Technology and Innovation Plan.

9 References

- [1] Mas, I., Kitts, C.: 'Obstacle avoidance policies for cluster space control of nonholonomic multirobot systems', *IEEE/ASME Trans. Mechatronics*, 2012, **17**, (6), pp. 1068–1079
- [2] Balch, T., Arkin, R.C.: 'Behavior-based formation control for multirobot teams', *IEEE Trans. Robot. Autom.*, 1998, **14**, (6), pp. 926–939
- [3] Luna, J.-M., Fierro, R., Abdallah, C., *et al.*: 'An adaptive coverage control algorithm for deployment of nonholonomic mobile sensors'. 2010 49th IEEE Conf. Decision and Control (CDC), 2010, pp. 1250–1256
- [4] Ehrich Leonard, N., Fiorelli, E.: 'Virtual leaders', artificial potentials and coordinated control of groups, 2001'. Proc. 40th IEEE Conf. Decision and Control, 2001, vol. 3, pp. 2968–2973
- [5] Ailon, A., Zohar, I.: 'Control strategies for driving a group of nonholonomic kinematic mobile robots in formation along a time-parameterized path', *IEEE/ASME Trans. Mechatronics*, 2012, **17**, (2), pp. 326–336
- [6] Lee, Y.-H., Kim, S.-G., Kuc, T.-Y., *et al.*: 'Virtual target tracking of mobile robot and its application to formation control', *Int. J. Control, Autom. Syst.*, 2014, **12**, (2), pp. 390–398
- [7] Asl, A.N., Menhaj, M.B., Sajedin, A.: 'Control of leader-follower formation and path planning of mobile robots using asexual reproduction optimization (ARO)', *Appl. Soft Comput.*, 2014, **14**, Part C, pp. 563–576
- [8] Chang, Y.-H., Yang, C.-Y., Chan, W.-S., *et al.*: 'Adaptive fuzzy sliding-mode formation controller design for multi-robot dynamic systems', *Int. J. Fuzzy Syst.*, 2014, **16**, (1), p. 121
- [9] Cai, X., de Queiroz, M.: 'Adaptive rigidity-based formation control of uncertain multi-robotic vehicles'. American Control Conf. (ACC), 2014, June 2014, pp. 293–298
- [10] Li, Y.D., Zhu, L., Sun, M.: 'Adaptive neural-network control of mobile robot formations including actuator dynamics', *Appl. Mech. Mater.*, 2013, **303**, pp. 1768–1773
- [11] Arrichiello, F., Chiaverini, S., Fossen, T.I.: 'Formation control of underactuated surface vessels using the null-space-based behavioral control'. IEEE/RSJ Int. Conf. Intelligent Robots and Systems, 2006, pp. 5942–5947
- [12] Arrichiello, F., Das, J., Heidarsson, H., *et al.*: 'Experiments in autonomous navigation with an under-actuated surface vessel via the null-space based behavioral control'. IEEE/ASME Int. Conf. Advanced Intelligent Mechatronics (AIM), 2009, 2009, pp. 362–367
- [13] Arrichiello, F., Chiaverini, S., Indiveri, G., *et al.*: 'The null-space-based behavioral control for mobile robots with velocity actuator saturations', *Int. J. Robot. Res.*, 2010, pp. 1317–1337
- [14] Antonelli, G., Arrichiello, F., Chiaverini, S.: 'The Entrapment/Escorting mission', *IEEE Robot. Autom. Mag.*, 2008, **15**, (1), pp. 22–29
- [15] Mahacek, P., Mas, I., Petrovic, O., *et al.*: 'Cluster space control of a 2-robot system as applied to autonomous surface vessels'. OCEANS 2008, September 2008, pp. 1–5
- [16] Mahacek, P., Mas, I., Kitts, C.: 'Cluster space control of autonomous surface vessels utilizing obstacle avoidance and shielding techniques'. IEEE/OES Autonomous Underwater Vehicles (AUV), 2010, 2010, pp. 1–5
- [17] Mas, I., Petrovic, O., Kitts, C.: 'Cluster space specification and control of a 3-robot mobile system'. IEEE Int. Conf. Robotics and Automation (ICRA 2008), IEEE, 2008, pp. 3763–3768
- [18] Agnew, M.S., Dal Canto, P., Kitts, C.A., *et al.*: 'Cluster space control of aerial robots'. IEEE/ASME Int. Conf. Advanced Intelligent Mechatronics (AIM), 2010, 2010, pp. 1305–1310
- [19] Kitts, C.A., Mas, I.: 'Cluster space specification and control of mobile multirobot systems', *IEEE/ASME Trans. Mechatronics*, 2009, **14**, (2), pp. 207–218
- [20] El Ferik, S., Nasir, M.T., Baroudi, U.: 'A Behavioral Adaptive Fuzzy controller of multi robots in a cluster space', *Appl. Soft Comput.*, 2016, **44**, pp. 117–127
- [21] Mahacek, P., Kitts, C.A., Mas, I.: 'Dynamic guarding of marine assets through cluster control of automated surface vessel fleets', *IEEE/ASME Trans. Mechatronics*, 2012, **17**, (1), pp. 65–75
- [22] Kitts, C., Mahacek, P., Adamek, T., *et al.*: 'Experiments in the control and application of Automated Surface Vessel fleets'. IEEE OCEANS 2011, 2011, pp. 1–7
- [23] Mas, I., Kitts, C.A.: 'Dynamic control of mobile multirobot systems: the cluster space formulation', *IEEE Access*, 2014, **2**, pp. 558–570
- [24] Laumond, J.-P., Sekhavat, S., Lamiraux, F.: 'Guidelines in nonholonomic motion planning for mobile robots' (Springer, 1998)
- [25] Kitts, C.A., Stanhouse, K., Chindaphorn, P.: 'Cluster space collision avoidance for mobile two-robot systems'. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS 2009), 2009, pp. 1941–1948
- [26] Yang, J.-M., Kim, J.-H.: 'Sliding mode control for trajectory tracking of nonholonomic wheeled mobile robots', *IEEE Trans. Robot. Autom.*, 1999, **15**, (3), pp. 578–587
- [27] Chwa, D.: 'Sliding-mode tracking control of nonholonomic wheeled mobile robots in polar coordinates', *IEEE Trans. Control Syst. Technol.*, 2004, **12**, (4), pp. 637–644
- [28] Lee, J.H., Lin, C., Lim, H., *et al.*: 'Sliding mode control for trajectory tracking of mobile robot in the RFID sensor space', *Int. J. Control, Aut. Syst.*, 2009, **7**, (3), pp. 429–435
- [29] Lee, J.K., Choi, Y.H., Park, J.B.: 'Sliding mode tracking control of mobile robots with approach angle in cartesian coordinates', *Int. J. Control, Autom. Syst.*, 2015, **13**, (3), pp. 718–724
- [30] Koubaa, Y., Boukattaya, M., Dammak, T.: 'Adaptive Sliding-Mode Dynamic Control For Path Tracking of Nonholonomic Wheeled Mobile Robot'. 2015
- [31] Chen, C.-Y., Li, T.-H.S., Yeh, Y.-C.: 'EP-based kinematic control and adaptive fuzzy sliding-mode dynamic control for wheeled mobile robots', *Inf. Sci.*, 2009, **179**, (1–2), pp. 180–195
- [32] Fierro, R., Lewis, F.L.: 'Control of a nonholonomic mobile robot using neural networks', *IEEE Trans. Neural Netw.*, 1998, **9**, (4), pp. 589–600
- [33] Chen, J., Sun, D., Yang, J., *et al.*: 'A leader-follower formation control of multiple nonholonomic mobile robots incorporating receding-horizon scheme', *Int. J. Robot. Res.*, 2009, **29**, (6), pp. 727–747
- [34] Li, J., Yang, L.: 'Adaptive pi-based sliding mode control for nanopositioning of piezoelectric actuators', *Math. Probl. Eng.*, 2014, pp. 1–10