ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

MULTI-AGENT COVERAGE CONTROL WITH ADAPTATION TO PERFORMANCE VARIATIONS AND IMPRECISE LOCALIZATION

Ph.D. THESIS

Mert TURANLI

Department of Control and Automation Engineering

Control and Automation Engineering Programme

JULY 2020

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

MULTI-AGENT COVERAGE CONTROL WITH ADAPTATION TO PERFORMANCE VARIATIONS AND IMPRECISE LOCALIZATION

Ph.D. THESIS

Mert TURANLI
(504112104)

Department of Control and Automation Engineering

Control and Automation Engineering Programme

Thesis Advisor: Prof. Dr. Hakan TEMELTAS

JULY 2020

ISTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

ÇOK ETMENLİ SİSTEMLERDE PERFORMANS DEĞİŞİMLERİNE ADAPTASYONU VE KONUMLAMA BELİRSİZLİĞİNİ GÖZ ÖNÜNE ALAN KAPSAMA KONTROLÜ

DOKTORA TEZİ

Mert TURANLI
(504112104)

Kontrol ve Otomasyon Mühendisliği Anabilim Dalı

Kontrol ve Otomasyon Mühendisliği Programı

Tez Danışmanı: Prof. Dr. Hakan TEMELTAŞ

TEMMUZ 2020

Mert TURANLI, a Ph.D. student of ITU Graduate School of Science Engineering and Technology student ID 504112104 successfully defended the thesis/dissertation entitled “MULTI-AGENT COVERAGE CONTROL WITH ADAPTATION TO PERFORMANCE VARIATIONS AND IMPRECISE LOCALIZATION”, which he prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor : **Prof. Dr. Hakan TEMELTAS** ....
Istanbul Technical University

Jury Members : **Assoc. Prof. Dr. Gulay OKE GUNEL** ....
Istanbul Technical University

Ass. Prof. Dr. Ilker USTOGLU ....
Istanbul Technical University

Assoc. Prof. Dr. Cenk ULU ....
Yildiz Technical University

Ass. Prof. Dr. Janset DASDEMIR ....
Yildiz Technical University

Date of Submission : 22 June 2020
Date of Defense : 24 July 2020

To my family,

FOREWORD

I would like to thank many people who has given support and shown their patience during my thesis. First of all, I would like to thank my advisor Prof. Hakan Temeltas for his technical advices and scientific guidance. Also, I am thankful to my family who have given their support any time I needed.

July 2020

Mert TURANLI
(Control and Automation
Engineer)

TABLE OF CONTENTS

Page

FOREWORD ix
TABLE OF CONTENTS xi
ABBREVIATIONS xiii
SYMBOLS xv
LIST OF TABLES xvii
LIST OF FIGURES xix
SUMMARY xxiii
ÖZET xxv
1. INTRODUCTION 1
  1.1 Purpose and Scope of Thesis 2
    1.1.1 Unique aspect 3
    1.1.2 Impact 5
  1.2 Literature Review 5
  1.3 Hypothesis 14
2. VORONOI BASED COVERAGE CONTROL FOR MULTIPLE AGENTS 15
  2.1 Preliminaries 15
  2.2 Voronoi Tessellations 16
  2.3 Power Voronoi Diagrams 16
  2.4 Guaranteed Voronoi Diagrams 17
  2.5 Guaranteed Voronoi Diagrams of Ellipses 19
  2.6 Guaranteed Power Voronoi Diagrams 21
  2.7 Optimal Coverage Formulation with Locational Optimization 23
  2.8 Discussion 24
3. ADAPTIVE COVERAGE CONTROL WITH IMPRECISE LOCALIZATION 25
  3.1 Coverage Control with GPVDs 25
  3.2 Control Law for Non-holonomic Agents 26
  3.3 Non-holonomic Adaptive Coverage Control 26
  3.4 Stability Analysis 27
  3.5 Discussion 29
4. ADAPTATION TO PERFORMANCE VARIATIONS WITH GPVDs 31
  4.1 Control Laws 31
  4.2 Adaptation Laws 32
  4.3 Stability Analysis 33
  4.4 Discussion 35
5. ADAPTATION TO PERFORMANCE VARIATIONS WITH HNNs UNDER LOCALIZATION UNCERTAINTY 37
  5.1 Hopfield Neural Networks 37
  5.2 Point-Offset Control Law 38
  5.3 HNN Parameter Estimator 38
  5.4 Weight Adaptation Law 39

5.5 Stability Analysis..40
5.6 Agent Disorder 42
5.7 Discussion..43
6. ENERGY-EFFICIENT COVERAGE CONTROL WITH HOPFIELD NETWORKS..45
6.1 Problem Formulation..45
6.2 Energy-Efficient Coverage Optimal Control..46
6.3 Multi-Agent Coverage Control with Different Actuation Capabilities..........46
6.4 Estimating the Performance Parameters with HNNs 47
6.5 Estimating the PD Weights of the Agents..49
6.6 Stability Analysis..49
6.7 Discussion..51
7. SIMULATION RESULTS 53
7.1 Adaptive Coverage Control with Imprecise Localization 53
7.1.1 Stability results..54
7.1.2 Results of the coordination algorithm with GPVD 55
7.2 Adaptation with non-holonomic estimation model 60
7.2.1 Stability results..60
7.2.2 Case study 1 62
7.2.3 Case study 2 65
7.3 Adaptation with Hopfield Neural Networks..67
7.3.1 Results with 5, 10 and 15 agents..68
7.3.2 Effect of Hopfield Network parameter...77
7.3.3 Change of regions with respect to performances 77
7.3.4 Agent disorder..79
7.4 Energy-Efficient adaptation with Hopfield Neural Networks.........................83
7.5 Discussion..89
8. EXPERIMENTS..91
8.1 ROS Implementation..91
8.2 Experiments: Adaptation to Performance Variations with HNNs Under Localization Uncertainty..93
8.2.1 Experimental Setup 93
8.2.2 Experimental Results 94
8.3 Experiments: Energy-Efficient Coverage Control with Hopfield Networks...98
8.4 Discussion..103
9. CONCLUSION..105
REFERENCES..107
APPENDICES 113
APPENDIX A Communication Diagrams of ROS Nodes.................................114
CURRICULUM VITAE 117

ABBREVIATIONS

AWVD	: Additively Weighted Voronoi Diagram
HNN	: Hopfield Neural Network
GVD	: Guaranteed Voronoi Diagram
GPVD	: Guaranteed Power Voronoi Diagram
LIP	: Linear in Parameters
MWVD	: Multiplicatively Weighted Voronoi Diagram
PD	: Power Diagram
VD	: Voronoi Diagram

SYMBOLS

$\boldsymbol{\langle i,j\rangle}$ **:** Bisector of *i* and *j*
$\boldsymbol{A}$: System parameter matrix
$\boldsymbol{B}_i$: Bias of the neuron *i*
$\boldsymbol{B}$: Bias vector
$\boldsymbol{C_i^u}$: Uncertain region for uncertainty circle case
$\boldsymbol{C_i^s}$: Uniform radial sensing footprint
$\boldsymbol{C_i^{gs}}$: Guaranteed sensing region
$\boldsymbol{C}_{\mathrm{V}_i}$: Centroid of the i^{th} Voronoi region
$\boldsymbol{e_i}$: Distance between the i^{th} agent and the centroid
$\boldsymbol{E_i^u}$: Uncertain region for uncertainty ellipse case
$\boldsymbol{f}$: Performance function of $\widehat{K}_i$
$\boldsymbol{I}$: Identity matrix
$\boldsymbol{k}$: Positive gain matrix
$\boldsymbol{k_\Delta}$: Uncertainty matrix
$\boldsymbol{K_i}$: Estimation parameter
$\boldsymbol{\widehat{K}_i}$: Estimated vector
$\boldsymbol{K_p}$: Controller gain matrix
$\boldsymbol{k_\omega}$: Adaptation gain
$\boldsymbol{l}$: Point offset distance
$\boldsymbol{M}$: Number of neurons
$\boldsymbol{M_{V_i}}$: Mass of the i^{th} Voronoi region
$\boldsymbol{N}$: Number of dimensions
$\boldsymbol{N_i}$: Neighbors of the agent *i*
$\boldsymbol{p^0}$: Position in the frame *0*
$\boldsymbol{p_i}$: Position of the i^{th} agent
$\boldsymbol{r_i}$: Uncertainty radius of i^{th} generator point
$\boldsymbol{r_u}$: Uncertainty radius
$\boldsymbol{R_0^c}$: Rotation from frame *0* to *c*
$\boldsymbol{s}_j$: State or output of the neuron *i*
$\boldsymbol{S}$: Convex region
$\boldsymbol{T_0^b}$: Transformation from frame *0* to *b*
$\boldsymbol{u_i}$: Linear control input for the i^{th} agent
$\boldsymbol{U_i}$: Uncertainty region of site point *i*
$\boldsymbol{v_i}$: Control input for the i^{th} agent
$\boldsymbol{V}_i$: Voronoi region of the i^{th} site point
$\boldsymbol{V_i^g}$: Guaranteed voronoi region of site point *i*
$\boldsymbol{w_i}$: The weight of the i^{th} generator point
$\boldsymbol{W}$: Weight matrix
$\boldsymbol{W_i^g}$: Guaranteed power region of i^{th} agent
$\boldsymbol{x_i}$: State of the i^{th} agent
$\boldsymbol{Y}$: Closed loop system matrix

$\boldsymbol{z_i}$: Total input to neuron i
$\boldsymbol{Z}$: Vector function
$\boldsymbol{\alpha}$: HNN parameter
$\boldsymbol{\beta}$: HNN parameter
$\boldsymbol{\gamma}$: Linear control gain
$\boldsymbol{\varsigma}$: Parameter vector
$\boldsymbol{\tilde{\varsigma}_i}$: Parameter estimation error of the i^{th} agent
$\boldsymbol{\theta_{est}}$: Estimation vector
$\boldsymbol{\hat{\theta}_{est}}$: Estimated vector
$\boldsymbol{\lambda}$: Angular control gain
$\boldsymbol{\Xi_{pr}}$: Projection law
$\boldsymbol{\eta}$: Consensus gain
$\boldsymbol{\xi}$: Adaptation gain
$\boldsymbol{\rho}$: Collision radius
$\boldsymbol{\phi}$: Density function
$\boldsymbol{\varphi_i}$: The heading angle of the i^{th} agent
$\boldsymbol{\psi_i}$: The angular error between the i^{th} agent and the centroid
$\boldsymbol{\omega_i}$: Angular control input of the i^{th} agent
$\boldsymbol{\mathfrak{I}}$: Coverage cost function
$\boldsymbol{\mathcal{L}_i}$: Neighbors of the i^{th} agent
$\boldsymbol{\ell}$: Non-decreasing performance function

LIST OF TABLES

Page

Table 7.1 : Region ratios of the agents at the end configuration (with 5 agents)...... 56
Table 7.2 : Region ratios of the agents at the end configuration (with 10 agents).... 57
Table 7.3 : Region ratios of the agents at the end configuration (with 15 agents).... 59
Table 7.4 : Region ratios of the agents at the end configuration (stability results)... 61
Table 7.5 : Region ratios of the agents at the end configuration (case study 1)........ 64
Table 7.6 : Region ratios of the agents at the end configuration (case study 2)........ 66
Table 7.7 : Region ratios of the agents at the end configuration (HNN with 5 agents). 70
Table 7.8 : Region ratios of the agents at the end configuration (HNN with 10 agents). 72
Table 7.9 : Region ratios of the agents at the end configuration (HNN with 15 agents). 76
Table 7.10 : Region ratios of the agents at the end configuration (HNN with 10% degradation)........ 78
Table 7.11 : Region ratios of the agents at the end configuration (HNN with 20% degradation)........ 79
Table 7.12 : Region ratios of the agents at the end configuration (HNN with 30% degradation)........ 79
Table 7.13 : Region ratios of the agents at the end configuration (HNN with agent disorder). 82
Table 7.14 : Region ratios of the agents at the end configurations 86
Table 7.15 : Change of energy consumption with respect to r_i........ 87
Table 7.16 : Change of coverage time with respect to s_i........ 87
Table 8.17 : Region ratios of the agents at the end configuration (with non-holonomic estimator). 92
Table 8.18 : Weights of the agents at the end configuration (with non-holonomic estimator)........ 92
Table 8.19 : Region ratios of the agents at the end configuration (with Hopfield estimator)........ 93
Table 8.20 : Weights of the agents at the end configuration (with Hopfield estimator)........ 93
Table 8.21 : Region ratios of the agents at the end configuration with HNN of three experiments. 97
Table 8.22 : Change of energy consumption with respect to r_i........ 102
Table 8.23 : Change of energy consumption with respect to s_i........ 102
Table 8.24 : Change of region ratios with respect to actuation performances of the agents........ 102

LIST OF FIGURES

Page

Figure 2.1 : Power Voronoi Diagram (all weights are zero) 17
Figure 2.2 : Power Voronoi Diagram ($\omega_1 = 0.5$) 17
Figure 2.3 : Guaranteed Voronoi Diagram of Disks 18
Figure 2.4 : Guaranteed Voronoi Diagram of Disks with Different Radii 18
Figure 2.5 : Example Guaranteed Voronoi Diagram of Ellipses 20
Figure 2.6 : Example Guaranteed Voronoi Diagram of Ellipses 20
Figure 2.7 : Algorithm for GVDs of Ellipses 21
Figure 2.8 : Bisectors of a GVD of Ellipses 21
Figure 2.9 : Example GPVD with $\omega_1 \neq 0$ 22
Figure 2.10 : Example GPVD with different radii 22
Figure 2.11 : Example GPVD with different radii 23
Figure 3.1 : Position of the agent and model parameters 25
Figure 5.1 : Agent disorder algorithm 42
Figure 6.1 : The robot position p_i and centroid location C_{Vi} 45
Figure 7.1 : Robot positions in a simulation frame (red circles show the centroid positions) 53
Figure 7.2 : GPVD in a simulation frame 54
Figure 7.3 : Distance and Angle Errors of 5 Agents 54
Figure 7.4 : Parameter Estimation Errors 55
Figure 7.5 : Consensus Errors 55
Figure 7.6 : The trajectories of the 5 agents for the case $\omega_1 = 0$ (red circles represent the end configurations, black circles show the start positions) 56
Figure 7.7 : The trajectories of 5 agents with $\omega_1 = 0.5$ 57
Figure 7.8 : The trajectories of 10 agents with $\omega_1 = 0$ 58
Figure 7.9 : The trajectories of 10 agents with $\omega_1 = 0.5$ 58
Figure 7.10 : The trajectories of 15 agents with $\omega_1 = 0$ 59
Figure 7.11 : The trajectories of 15 agents with $\omega_1 = 0.5$ 59
Figure 7.12 : Position and Angle Errors 61
Figure 7.13 : Coverage Cost Function 61
Figure 7.14 : Parameter Estimation Error 62
Figure 7.15 : The value of $w_i - f(K_i)$ 62
Figure 7.16 : The weights w_i of the agents 63
Figure 7.17 : Coverage Cost Function (Case Study 1) 63
Figure 7.18 : The value of $w_i - f(K_i)$ 63
Figure 7.19 : The weights w_i of the agents 63
Figure 7.20 : Parameter estimation error 64
Figure 7.21 : The trajectories of the agents for case study 1 (black circles: start configuration, red circles: end configuration) 64
Figure 7.22 : Coverage Cost Function (Case Study 2) 65
Figure 7.23 : The value of w_i-f(K_i) 66

Figure 7.24 : The weights w_i of the agents 66
Figure 7.25 : Parameter estimation error 66
Figure 7.26 : The trajectories of the agents for the case study 2 (black circles: start configuration, red circles: end configuration) 67
Figure 7.27 : Position errors 68
Figure 7.28 : Coverage cost 68
Figure 7.29 : Parameter estimation errors 69
Figure 7.30 : The value of $w_i - f(K_i)$ 69
Figure 7.31 : The weights w_i of the agents 70
Figure 7.32 : The trajectories of the agents 70
Figure 7.33 : Position errors of the agents 71
Figure 7.34 : Coverage cost 71
Figure 7.35 : Parameter estimation errors 72
Figure 7.36 : The value of $\mathrm{w_i}$-$\mathrm{f(K_i)}$ 72
Figure 7.37 : The weights of the agents 73
Figure 7.38 : The trajectories of the agents 73
Figure 7.39 : Position errors 74
Figure 7.40 : Coverage cost 74
Figure 7.41 : Parameter estimation errors 75
Figure 7.42 : The value of $w_i - f(K_i)$ 75
Figure 7.43 : The weights of the agents 75
Figure 7.44 : The trajectories of the agents 76
Figure 7.45 : The effect of the parameter α of the HNN on the convergence speed 77
Figure 7.46 : Blue region: first agent (degraded by 10 percent) 77
Figure 7.47 : Blue region: first agent (degraded by 20 percent) 78
Figure 7.48 : Blue region: first agent (degraded by 30 percent) 78
Figure 7.49 : Position errors 80
Figure 7.50 : Coverage cost 80
Figure 7.51 : Parameter estimation errors 81
Figure 7.52 : The value of $w_i - f(K_i)$ 81
Figure 7.53 : The weights of the agents 81
Figure 7.54 : The trajectories of the agents 82
Figure 7.55 : The end configurations of the agents 82
Figure 7.56 : GPD at the end configuration of the agents 83
Figure 7.57 : Position errors 84
Figure 7.58 : Coverage cost 84
Figure 7.59 : Parameter estimation errors 85
Figure 7.60 : The value of $w_i - f_{perf}(K_i)$ 85
Figure 7.61 : The weights of the agents 85
Figure 7.62 : The trajectories of the agents 86
Figure 7.63 : Energy consumptions of the agents (Simulation #1) 88
Figure 7.64 : Energy consumptions of the agents (Simulation #2) 88
Figure 7.65 : Energy consumptions of the agents (Simulation #3) 88
Figure 7.66 : Total energy consumptions 88
Figure 8.1 : A simulation frame from ROS/Gazebo with Turtlebot 3 robots. The plot window shows the parameter estimation errors. 91
Figure 8.2 : The initial configurations of the robots (ITU Robotics Laboratory) 94
Figure 8.3 : Communication architecture used in the experimental setup 94
Figure 8.4 : Coverage cost 95
Figure 8.5 : Estimation errors 95

Figure 8.6 : The value of $\boldsymbol{wi - f(Ki)}$ 96
Figure 8.7 : The weights of the agents 96
Figure 8.8 : The trajectories of the agents 97
Figure 8.9 : The GPD plot obtained from the first experiment 97
Figure 8.10 : The ROS/Gazebo Simulation Environment 98
Figure 8.11 : The environment in which the experiments were performed 99
Figure 8.12 : The value of cost function J 100
Figure 8.13 : The parameter estimation errors 100
Figure 8.14 : The value of $\boldsymbol{wi - f(Ki)}$ 101
Figure 8.15 : The weights of the agents 101
Figure 8.16 : The trajectories of the agents 102
Figure A.1 : ROS Nodes and Topics for Coverage Control with Non-holonomic Estimator 114
Figure A.2 : ROS Nodes and Topics for Coverage Control with Hopfield Estimator 115

MULTI-AGENT COVERAGE CONTROL WITH ADAPTATION TO PERFORMANCE VARIATIONS AND IMPRECISE LOCALIZATION

SUMMARY

In this thesis, an adaptive collaboration approach for a multi-agent system consisting of nonholonomic wheeled mobile robots is proposed. The positions of the agents are not known precisely but their locations are known to be within uncertainty circles. For the collaboration among the robots, the workspace partitioning algorithm is chosen as Guaranteed Power Voronoi Diagram (GPVD or GPD) which not only takes the localization uncertainty into account but also is capable of changing the regions of the generator points with respect to corresponding weight parameters. Also, the assumption is that the actuation capabilities of the robots are different from each other. The agents do not know those parameters related to their actuation performances beforehand.

The contribution of the thesis is that the performance parameters of the agents are learned online by the proposed adaptive estimator algorithm and Hopfield Neural Network (HNN) estimator under localization uncertainty. The proposed algorithm is based on the coverage control which performs collaboration among the robots by assigning the regions from the workspace according to their actuation performances automatically. The definition of the actuation performances is different capabilities of the agents. The examples of strong actuation performances may include powerful motors and favorable terrain while wheel slip and weak motors can be counted as examples for the weak actuator performances.

The proposed multi-agent collaborative coverage algorithm learns the performance parameters of the robots by using two approaches proposed in the thesis. The first approach is based on an adaptive estimator with a non-holonomic estimation model. The second method uses an HNN estimator. The theoretical proof, analysis and verification of the aforementioned methods are given in the related sections.

After estimating the performance parameters, the weights are calculated using a neighbor based weight estimation algorithm. The weight variables are utilized in the GPD algorithm so that the workspace is partitioned according to the performance parameters of the agents in a guaranteed sense. At the end, the agents take regions from the workspace according to their actuation performances and achieve the optimal collaborative coverage so that the agents with strong actuators take larger regions from the environment than the agents with poor actuators. Thus, the collaborative coverage algorithm enables the robots to deploy themselves to an optimal configuration which minimizes the total coverage cost by taking imprecise localization into account.

Moreover, a multi-agent coverage collaboration method with an energy-efficient optimal coverage control law and Hopfield networks is proposed in the related section. By using the algorithm a trade-off between coverage time and energy consumption among agents can be done. Meanwhile, the collaboration is achieved according to the actuation performances of the agents.

The theoretical results are verified with MATLAB and ROS/Gazebo simulations and experiments that show the efficiency of the algorithm. The ROS implementation of the algorithm is explained. The experimental results are given in the related section.

ÇOK ETMENLİ SİSTEMLERDE PERFORMANS DEĞİŞİMLERİNE ADAPTASYONU VE KONUMLAMA BELİRSİZLİĞİNİ GÖZ ÖNÜNE ALAN KAPSAMA KONTROLÜ

ÖZET

Bu tezde çok etmenli sistemler için uyarlamalı bir işbirliği (kolaborasyon) yöntemi önerilmiştir. Etmenler holonomik olmayan tekerlekli robotlar olarak ele alınmıştır. Robotların konumları tam olarak bilinmemekte, bir belirsizlik yarıçapı içerisinde oldukları varsayılmaktadır.

Voronoi tabanlı koordinasyon ve işbirliği yöntemleri literatürde özellikle son zamanlarda yaygın olarak karşılaşılmakta olan problemlerdir. Voronoi tabanlı kapsama kontrolü, haritayı optimal şekilde, bir konumsal optimizasyon fonksiyonuna göre birden çok etmenin olduğu durumda kapsamayı amaçlar. Etmenlerin haritayı bölgelendirmesi için Voronoi diyagramları kullanılır. Bir yoğunluk fonksiyonu haritadaki her nokta için bilgi dağılımını verecek şekilde tanımlanır. Etmenler Voronoi bölgelendirmesi ve yoğunluk fonksiyonunu kullanarak optimizasyonun çözümüne ulaşırlar. Her bir etmen için bulunan optimal noktalar, bir kontrol kuralı kullanılarak etmenleri görevi yerine getirecek şekilde optimal çözüme ulaştırır.

Literatürde bir çok çeşidi bulunan Voronoi diyagramlarından en basiti olan klasik Voronoi diyagramı oluşturucu noktaların konumlarında belirsizlik olmadığını ve bölgelerin ağırlıklandırılmadığını varsayar. Sweepline algoritması klasik Voronoi diyagramının çiziminde kullanılan ve iyi sonuç veren bir algoritmadır. Eklemeli ağırlıklandırmalı Voronoi diyagramları (AWVD), çarpma ağırlıklandırmalı Voronoi diyagramları (MWVD) ve güç Voronoi diyagramları (PVD ya da PD) Voronoi diyagramlarının çeşitlerindendir. Bu algoritmaların ortak özelliği ağırlıklandırmalı olmalarıdır, bu da oluşturucu noktaların birer ağırlıkları olduğu anlamına gelir. Bu ağırlıklar kullanılarak bölgelerin büyüklükleri değiştirilebilmektedir. Özellikle, güç Voronoi diyagramları, ayırıcı eğrilerinin (bisector curve) klasik Voronoi diyagramlarında olduğu gibi doğru olması nedeniyle çok etmenli robotik yöntemlerinde çokça kullanılmaktadır. Bir diğer tür olan ve bu tezde çokça kullanılacak olan Guaranteed Voronoi diyagramları (GVD), oluşturucu noktaların konumlarının belli birer nokta değil, konumlarında belirsizlik olan noktalar olduğunu varsayar. Bu şekilde konum belirsizliği bir çember ya da elips ile modellenebilir. Bu da oluşan Guaranteed Voronoi bölgelerinin dışında belirsiz ve hiçbir etmene ait olmayan bölgelerin oluşmasına neden olur. Ayrıca, GVD'lerin ayırıcı eğrileri birer hiperboldür. Eğer, literatürde önerildiği gibi GVD'lerin uzaklık fonksiyonları güç uzaklık fonksiyonu olarak tanımlanırsa, bu diyagramlar Guaranteed güç diyagramları (GPD ya da GPVD) adını alır ve güç diyagramlarında olduğu gibi oluşturucu noktaların ağırlık değerleri değiştirilerek noktaların bölgelerinin büyüklükleri

değiştirilebilir ve aynı zamanda konumlama belirsizlikleri göz önüne alınabilir. Bu da çok etmenli robotlu sistemlerde, bir parametreye ya da değişkene göre etmene ilişkin bölgenin büyüklüğünün değiştirilebilmesini olanaklı kılar. Bu tezde de, literatürde de yapıldığı gibi bu ağırlıklar otomatik olarak uyarlamalı bir algoritma ile değiştirilerek etmenlerin çalışma alanlarının eyleyici performansı gibi kestirilen parametrelere göre atanmasını sağlayan yöntemler kullanılmıştır. Bu şekilde robotlar arası işbirliği yapılması sağlanmakta ve eyleyici performansı yüksek robotların daha geniş çalışma alanına sahip olması mümkün hale getirilmektedir.

Önerilen algoritmanın önemli parçalarından bir diğeri ise kestirim algoritmasıdır. Hareketli bir sistemin model kullanılarak parametrelerinin kestirilmesi literatürde uygulaması olan bir yöntemdir. Bu iş için uyarlamalı doğrusal olmayan yöntem, temel alınan bir yayında önerilmiştir. Alternatif olarak bu tezde önerilen Hopfield Yapay Sinir Ağları ile her bir etmenin performans parametreleri kestirilebilmektedir. Yapay Sinir Ağlarını temel alan başka yöntemler ile ilgili olarak, örneğin sınıflandırma yapan ağlar, literatürde bahsedildiği üzere doğruluk ile ilgili sıkıntılar yaşanabilmektedir. Holonomik ve holonomik olmayan modeller ile yapılan kestirimler ile ilgili sonuçlar tez içerisinde verilmiştir. Holonomik modelde robotun noktasal dinamiğe sahip olduğu varsayılmakta ve kontrol kuralı özel bir dönüşüm ile holonomik olmayan robotların kontrol işaretlerini üretebilmektedir. Holonomik olmayan model ile yapılan kestirimde ise, hem kontrol kuralı hem de kestiricinin denklemleri holonomik olmayan tek tekerlekli bir aracın kinematik modeli olarak ele alınmaktadır.

Algoritmaya ilişkin bir diğer parça olan olan kontrol kuralı, holonomik olmayan araçlar için özel olarak türetilmiş bir kontrol kuralı ve holonomik dinamikler için kullanılan bir kontrol kuralının ürettiği kontrol işaretlerinin holonomik olmayan robotlara özel bir dönüşüm ile dönüştürülmesine dayalı olmak üzere iki farklı yöntem olarak incelenmiştir. Her iki kontrol kuralının Lyapunov tabanlı ıspatı yapılmış ve asimptotik kararlılık incelenmiştir. Elde edilen sonuçlara göre konum ile ilgili hatalar ve kestirim hataları sıfıra ilk koşuldan bağımsız bir şekilde asimptotik olarak gitmektedir. Ayrıca etmenlere ilişkin uyarlama kuralı ile elde edilen ağırlıklar da kararlı kalmaktadır. Hopfield ağının asimptotik kararlılığı da Lyapunov tipi bir ıspat ile literatürde yapıldığı gibi incelenmiştir.

Tezin ilk kısmında giriş bilgileri ve literatür taraması verilmiştir. İkinci kısım, çok etmenli sistemlerde Voronoi tabanlı kapsama kontrolü ile ilgili temelleri sunmaktadır. Voronoi diyagramları ve çeşitleri ile birlikte konumsal optimizasyon ile kapsama kontrolüne ilişkin bilgiler ve ıspatlar verilmiştir.

Üçüncü kısım olan konum belirsizliği altında uyarlamalı kapsama kontrolü, önerilen bir çok etmenli koordinasyon algoritmasını açıklamaktadır. Bu algoritma, literatürde var olan uyarlamalı çok etmenli koordinasyon algoritmasını temel almakta ve GPD ile çalışma alanı bölgelendirmesini sağlamaktadır. Gradyan tabanlı bir kestirici kullanılarak, yoğunluk fonksiyonu kestirilmekte ve holonomik olmayan kontrol kuralı ile robotlar bulunan optimal noktalara ulaştırılmaktadır. Böylelikle algoritma, dışarıdan aldığı sensör bilgisini bir konsensus protokolü ve kestirici ile hesaplayarak etmenlerin otonom bir şekilde koordinasyon kontrolü yapmasını sağlamaktadır. Yine Lyapunov tipi bir ıspat ile teorik analiz yapılmış ve simülasyonlar ile elde edilen sonuçlar doğrulanmıştır.

Dördüncü kısım, konum belirsizliği altında eyleyici performans değişimlerine göre uyarlamalı işbirliği yapan bir algoritmayı önermektedir. Bu algoritma, yoğunluk fonksiyonunun sabit olduğu durumda, otomatik olarak etmenlerin ağırlıklarını parametre kestirim vektörlerinden çıkararak, GPD ile bölgelendirme yapmakta ve performansı yüksek robotlara daha çok çalışma alanı, performansı düşük olanlara ise daha az çalışma alanı vermektedir. Bu şekilde enerji ve zaman açısından fayda sağlanabilmektedir. Lyapunov tipi bir ıspat ile uyarlamalı parametre kestirici, ağırlık kestirici ve holonomik olmayan kontrol kuralının global asimptotik kararlılığı ıspatlanmıştır. Elde edilen sonuçlar, simülasyon ile doğrulanmıştır.

Beşinci kısımda ise, parametre kestiricisi olarak HNN kullanılmış ve temel alınan çalışmada olduğu gibi kestirim modeli olarak holonomik bir nokta dinamiği modeli seçilmiştir. Kontrol kuralı da holonomik bir kural olup, holonomik olmayan robotlar için özel bir dönüşüm kullanmaktadır. Bu şekilde holonomik olmayan robotların kontrol işaretleri elde edilmektedir. Lyapunov tipi bir ıspat ile HNN kestiricisi, ağırlık kestiricisi ve kontrol kuralının global asimptotik kararlı olduğu ıspatlanmış ve sonuçlar simülasyonlar ile doğrulanmıştır. Ayrıca, etmenlerden bir ya da birkaçının eyleyicilerinin bozulması ve tamamen hareketsiz kalması durumu için özel bir algoritma önerilmiştir. Bu algoritma, hareketsiz robotları tespit ederek bu robotlar ilişkin ağırlıkları otomatik olarak sıfıra getirmektedir.

Altıncı kısımda enerji tüketimi ve kapsama zamanı arasında bir ayarlanabilir denge sağlayan, Hopfield ağlarına dayalı bir kestirici kullanan bir işbirlikçi kapsama kontrolü algoritması önerilmiştir. Lyapunov tipi bir ıspat ile kararlılık incelenmiştir ve global asimptotik kararlılık kanıtlanmıştır.

Yedinci kısımda üçüncü, dördüncü, beşinci ve altıncı bölümlere ilişkin simülasyon sonuçları verilmiştir. Üçüncü bölümdeki algoritma için 5, 10 ve 15 robot ile simülasyon yapılmış, dördüncü bölüme ilişkin iki senaryo ile ilgili sonuçlar verilmiştir. Beşinci bölümde HNN ile yapılan 5, 10 ve 15 robotlu simülasyon sonuçları incelenmiş, Hopfield parametrelerinin değişimlerine göre sistemin tepkisi verilmiş ve son olarak hareketsiz etmenlere ilişkin simülasyon sonuçları sunulmuştur. Altıncı bölüme ilişkin simülasyon sonuçları 6 robot için verilmiştir.

Sekizinci kısım olan uygulama kısmı, ROS/Gazebo simülasyon ortamında alınan sonuçları ve robotlar ile elde edilen deneysel sonuçları göstermektedir. C++ ile yazılan ROS düğümleri ve haberleşme diyagramları verilmiştir. Elde edilen sayısal sonuçlar ve deney sonuçları sunulmuştur.

1. INTRODUCTION

Multi-agent coordination problems are challenging topics studied intensively in the past years. In many applications, the result is better and faster by using more than one agent. The multi-agent coverage control topic which is the focus of this work is used for solving coordination problem of the multiple agents. Particularly, in mobile sensor networks area the distributed coverage control topic has gained a lot of importance. By using locational optimization the sensors are positioned in an optimal way to improve the overall coverage performance.

In a Voronoi-based multi-agent coverage algorithm, the regions of the robots are calculated by using the positions of the agents. For the move-to-centroid law, the robots position themselves by using the centroid locations calculated by using a density function also known as information gain function. The density function is given to the algorithm as an input or taken from the sensory information or may be calculated by using an entropy function for the exploration problem using the probabilities of the cells which can be considered as high probability for the visited regions (Bhattacharya, Michael, & Kumar, 2013). In this work, the density function is taken as an input, for example, a constant or time-varying Gaussian function, or sensory information, for example, a chemical concentration over an area or intensity of a light source. The latter case is especially useful for search and rescue operations.

Adaptation to performance variations of the agents is an important issue in multi-agent systems. In most of the applications, the adaptation of the agents with respect to their performances is not considered. In this approach, to gain benefits from time and energy, robots with poor actuator performance will take smaller area while strong robots will take larger partitions of the workspace. Thus, the Voronoi cells are changed dynamically in response to both the agent positions and their performance variations (Pierson, Figueiredo, Pimenta, & Schwager, 2015).

The work also will take imprecise localization into account. The so-called Guaranteed Voronoi diagrams (GVD) are going to be used. GVDs enable consideration of the uncertainty circles for the position of each agent, instead of neglecting the localization uncertainty while drawing Voronoi tessellations. In practical applications of the multi-agent problems, these issues also have significant importance.

1.1 Purpose and Scope of Thesis

The purpose of this work is especially solving the problem of control of the multi-agent systems. The problem can be divided into four categories according to the algorithm types used; estimation based control, optimal control-based approaches, hybrid switching approaches, and Lyapunov-based approaches. In the estimation based control, the coverage or multi-agent coordination algorithm estimates a state variable, for example, a density function as in the adaptive coverage control algorithm. A consensus protocol may be used with communication among agents. In the second category, the optimal control-based approaches use gradient laws to find the control inputs according to a locational optimization function for the optimal coverage. In the hybrid switching approaches category, hybrid systems and optimal control are used for the solution of the coordination problem. For the last category, the Lyapunov-based approaches include deriving the coverage control laws. A model for the system is chosen and the coverage control laws are obtained by using a Lyapunov-type proof. This work can be considered in estimation based control and Lyapunov-based approaches category since the uncertainties of the actuators are going to be estimated and coverage control laws for the agents are going to be derived for the nonholonomic agents.

The thesis is focused on the adaptation to performance variations of the agents and imprecise localization. At the end of the work, a Voronoi-based multi-agent coverage control algorithm with an online estimator will be obtained and the regions of the agents will be assigned dynamically according to the robots' performance variations. Additionally, the localization uncertainty will be handled in the algorithm. Moreover, the resulting algorithm will be convertible to a real-time application which can be run in real robots. In brief, with the resulting algorithm, the robots will position themselves optimally according to the density function taking the uncertainties of the actuators and imprecise localization into account.

1.1.1 Unique aspect

In this work, estimation of unknown and uncertain actuator parameters of collaborative working multiple agents will be studied. For this purpose, the problem can be divided into three main parts:

- Dynamic Voronoi partitioning
- Coverage control laws
- Density function selection

The dynamic Voronoi partitioning can be done in several ways. Except for the standard Voronoi diagram method, one of the most important ways to accomplish partitioning is using a power Voronoi diagram with additive weights. This enables to change the partitions with respect to another variable such as battery, fuel level or the performances of the agents. The weights also can be given manually as done in some works (Arslan & Koditschek, 2016). Additionally, additively weighted and multiplicatively weighted Voronoi diagrams are another methods to partition the robot space. Other types of Voronoi tessellations exist such as Guaranteed Voronoi diagrams which are about localization uncertainty and used in this work. It involves uncertainty circles for the agent locations.

The coverage control law is one of the main important aspects of the solution of the problem. Some of the works use standard coverage control laws for the point dynamics while some use their own with a kinematic model and Lyapunov-type proof. For example, for the nonholonomic vehicles, the coverage control laws with both unicycle and point dynamic models can be found in the literature. Another kind of coverage algorithms that exist in the literature may use gradient laws minimizing locational optimization function.

The density function selection is dependent on the application. The function may be an input to the algorithm or sensory information or may be calculated by using an entropy function for the exploration problem (Bhattacharya et al., 2013). For example, a constant or time-varying Gaussian function can be used as an input. Moreover, for the sensory information, a chemical concentration over an area or intensity of a light source may be chosen as information gain function with the appropriate sensors mounted on the robots.

The contributions of this work are listed below:

- The first contribution is that each agent has different actuator characteristic and the strong agents will take larger regions while for the weak ones smaller regions from the environment are assigned. For this purpose, dynamic additively weighted power Voronoi diagrams are used to take performance variations of agents into account. The dynamic regions are determined by using an estimation algorithm. The results for the first contribution are summarized in the work (Pierson et al., 2015).
- The second contribution of this work is about imprecise robot localization. The Guaranteed Voronoi Diagram (GVD) concept is going to be used for this purpose. The assumption is the robot's location is not known precisely, but it is guaranteed to be within an uncertainty circle. It is planned to define the uncertainty circles as ellipses so that they can be used in probability theory more efficiently. In the resulting algorithm, the Guaranteed Power Voronoi Diagram (GPVD) (Mahboubi, Vaezi, & Labeau, 2014) will be used which can handle weighted regions with localization uncertainty. Besides, in the original work the radii of the vehicles are equal. They are going to be chosen different in order to take heterogenous vehicles into account. The results for the second contribution is given in the work (Papatheodorou, Stergiopoulos, & Tzes, 2016).
- The work will extend a multi-agent coverage control algorithm for non-holonomic robots. The control law drives the agents to the locations which are found by the locational optimization part of the coverage algorithm. The results for the last contribution is explained in (Luna, J. M., Fierro, R., Abdallah, C. T. and Wood, 2013).

- The last contribution is that an HNN based estimator is used in the collaboration algorithm to learn the unknown parameters of each agent online (Alonso, Mendonça, & Rocha, 2009; Atencia & Joya, 2015). Also, for the HNN based method, the point-offset controller is going to be used (Michael & Kumar, 2009; Pierson & Schwager, 2015) in order to drive the robots to the centroid locations.

At the end of the work, an online multi-agent coverage control algorithm for non-holonomic agents adjusting the regions of the agents according to their actuator performances will be obtained which can deal with the localization uncertainties of the robots. The main contribution will be a Voronoi based adaptive estimator algorithm which is robust to localization uncertainties and can adapt itself to different actuator capacities.

1.1.2 Impact

Nowadays, the valuable human resources are shifted to the tasks that are more important and have less risk than the tasks allocated for the robots. Thus, the need for the use of the multi-agent systems, especially robots, is increasing. Search and rescue, detection, surveillance and exploration tasks may be done with the multi-agent systems more efficiently and effective without harming the humans. They may prevent death and injuries. The tasks may be executed autonomously, without human invention.

1.2 Literature Review

In the literature, there are a lot of recent research papers about multi-agent systems topic. Some of them are listed below with their brief summaries.

In a graph theory-based framework, the latest works in distributed coordination control area are briefly explained in the literature survey (X. Wang, Zeng, & Cong, 2016). Different coordination problems are explained with their connection to the graph theory such as consensus, formation control, rendezvous, alignment, swarming, flocking, containment control and circumnavigation control. The graphs are used to describe the connections among agents. An overview of the coordination control problems, system modeling, control law designs and analysis and structure transformation is given. The practical applications and potential directions in the area are also mentioned.

The paper (Ren, Beard, & Atkins, 2005) gives a survey of consensus problems in multi-agent cooperative control area. Time invariant and dynamical information change topologies are mentioned. Applications of the consensus protocols are given and research directions and open problems are investigated.

In another survey paper (Choi & Ahn, 2010), learning in multi-agent systems for complex tasks are investigated. The agents compete or cooperate in multi-agent reinforcement learning to accomplish the task. In cooperative multi-agent reinforcement learning topic (CMRL) the agents work in a cooperative way with other agents to complete the task. The survey explains coordination problems in CMRL area and proposes a new approach to the area.

With a second-order multi-agent system subjected to unknown disturbances, the leader-following problem is investigated (Hu & Zheng, 2014). In order to represent the disturbances, linearly parametrized models are used. With the help of relative position and velocity measurements, an adaptive tracking control for each follower is designed. Then the stability of the tracking system and parameter convergence is investigated. The theoretical results are illustrated with a numerical example.

In another paper (Mahboubi, Sharifi, Aghdam, & Zhang, 2012), an operation cost concept is used to formulate the distributed deployment of the multi-agent systems. The cases of obstacle-free and fixed obstacle environments are investigated. For the first case, the center multiplicatively weighted Voronoi configuration is given and it is proven to be the optimal configuration. For the second case, the visibility-aware multiplicatively weighted Voronoi diagram is shown with a motion coordination strategy. Simulation results are given for the both cases.

Another work (Cassandras, Lin, & Ding, 2013) considers the problem of persistent monitoring. An optimal control framework is given for the persistent monitoring problems. The multiple cooperating agents are controlled such that an uncertainty metric is minimized in a given environment. In the 1-D environment, it is shown that the optimal solution is to move at a maximal speed between the switching points and waiting some time and then changing the direction. A hybrid system is introduced to get a solution through gradient-based algorithm. The solution is robust with respect to the uncertainties. The basis for a two-dimensional mission space is created. In the paper (Lin & Cassandras, 2015), the persistent monitoring problem in two-dimensional spaces is introduced. An uncertainty metric is minimized to control multiple cooperating agents. In contrast to one-dimensional space, the optimal control is no longer the solution which consists of moving at maximal speed, waiting some time and changing the direction. For each agent, an optimal linear trajectory is assigned. It is shown that the elliptical trajectories give better results. Using Infinitesimal Perturbation Analysis, the performance gradients are obtained. Because the solutions obtained are locally optimal, a stochastic comparison algorithm to derive globally optimal elliptical trajectories are investigated. Numerical examples are given and the results are compared.

The agents communicating with a cloud network is presented (Nowzari & Pappas, 2016). The purpose is to minimize the amount of required communication. In the work, to accomplish this task, a self-triggered algorithm is developed. It guarantees that the system asymptotically converges to the desired states. Simulations are given with comparison to the results.

For the linear multi-agent systems, an output regulation problem is introduced (Lei, Sun, & Yeow, 2016). The distributed controller design is based on composite nonlinear feedback control which includes dynamic state feedback and nonlinear control parts. The former one consists of an internal model to achieve asymptotic tracking and disturbance rejection. The latter one is used to get a better transient response. The algorithm is applied to leader-follower problem and simulation results are given.

The work (Abdessameud, Polushin, Tayebi, & others, 2017) investigates distributed coordination problem of multi-agent systems with discrete time information exchange and time-varying delays. There are multiple dynamic leaders and the dynamics are assumed to be double integrators and second order systems with locally Lipschitz nonlinear dynamics. The effectiveness of proposed control laws is shown and numerical simulation results are given.

In another work (Romvary & Annaswamy, 2016), the eps-coordination concept is introduced to determine how well the agents are being coordinated. The conclusion with an application to a 6 agent network is introduced.

Multi-agent coordination control problem (Sun & Anderson, 2016) is investigated with individual agents having kinematic constraints. The constraints include nonlinear terms with possible drift terms. The global formation task is defined by edge constraints. The dynamics of the agents may be different which corresponds to a heterogeneous system. Examples with coordination control of constant speed agents are given.

In the paper (Tsumura & Kawasaki, 2016), finding optimal control/observation points of large scaled multi-agent systems are investigated.

In the paper (Baker, Ramchurn, Teacy, & Jennings, 2016), the coordination of multiple UAVs to perform emergency respond actions is studied. By using the real-time data about the disasters, planning the flight paths of UAVs is possible. For this purpose, a factored Monte Carlo tree search algorithm is proposed and the benchmarks of the algorithm are given. The results show that the algorithm performs 10% faster than the standard approaches.

The paper (de Souza, Eduardo, Ponzoni Carvalho Chanel, & Givigi, 2016) mentions a multi-aerial robot coordination game theoretical approach providing a surveillance mission. The mission involves visiting a set of points while minimizing the idle time. The method used is based on N-player game framework. Simulation results are given. Different policies obtained are compared using the metric of the average idleness. The results show that the proposed method decrease the idleness of all points compared to random action selection.

The paper (Imlauer, Mühlbacher, Steinbauer, Gspandl, & Reip, 2016) proposes a new approach with a hierarchical navigation system for team of robots using zones with traffic constraints in the environment. The planning system is based on an enriched roadmap representation, a central zone reservation and a search heuristic. The paper gives implementation of the method on the top of existing industrial transport robots.

In the paper (Surynek, Felner, Stern, & Boyarski, 2016), a method to find non-conflicting paths for multi-agent systems is proposed. The method is based on SAT solver. The authors improve he ideas from ICTS (a search based solver) and an experimental evaluation is given. The proposed method gives better results compared to search based algorithms - ICTS and ICBS.

The academic researches about the coverage problem gained a lot of attention in the past years. Some of them are given below with their summaries.

In the paper (Arslan & Koditschek, 2016), the authors introduced a multi-robot coverage control method which makes use of heterogenous vehicles and is capable of handling collision avoidance. They used power Voronoi diagrams with pre-selected weights and explained the problems encountered with the power diagrams under the topic occupancy defects. The method includes constrained optimization in coverage sense with a vector field planner and a heuristic congestion manager.

The paper (Palacios-Gasós, Montijano, Sagues, & Llorente, 2016) investigates the persistent coverage problem in which desired coverage level is maintained in the environment which decays over time. The authors converted the problem into a discrete optimization problem and solved it with the branch and bound method. The simulation results are given.

In the work (Bochkarev & Smith, 2016), the sweep coverage path planning topic is investigated. For the cleaning and monitoring applications, a minimal number of turns are needed for the path planning. The method proposed calculates the minimum height of non-convex polygonal partition taking the number of turns. For non-convex polygon, the polygon is cut into two pieces and the sum of heights are minimized. The algorithm iteratively reoptimizes the results and deletes the cuts. The results are given in the environment with obstacles.

The authors of the paper (Loizou & Constantinou, 2016) proposed a location dependent communication topology for the multi-agent coverage problem. The motion of the robots is constrained by the other robots. The algorithm gives the complete network coverage by minimum number of the robots. The maintenance of the communication constraints and the collision avoidance are provided with the algorithm.

The paper (J. Wang, Chen, Cheng, & Xie, 2016) is about the coverage path planning for the robot mowers. The authors proposed a hierarchical regioning strategy with two heuristic optimization algorithms: Ant Colony Optimization for the global path planning and Tabu Search for the local coverage planning in the lower layer. The simulation results are given.

The work (Papatheodorou et al., 2016) is about Voronoi-based multi-agent coverage control with uncertainty in localization. The sensing capabilities of the agents are uniform and radial. Also, the locations of the agents are not known precisely. The

Voronoi partitioning in the algorithm is done by Guaranteed Voronoi principle which takes positioning uncertainty into account. A distributed control law is proposed and each robot's region of responsibility corresponds to its GV-cell. The algorithm includes inherent collision avoidance and positioning information is shared with the neighbors of the robot.

In the paper (Shnaps & Rimon, 2015), a battery powered coverage algorithm is proposed for a single robot. There is limited energy capacity battery whose capacity is represented by the path length which can be taken with full battery charge. Unknown obstacles in the environment are considered.

The paper (Zhang, Zhou, & Ma, 2016) is about coverage control of multiple heterogenous non-holonomic mobile robots. There are no slip constraints and the move-to-centroid laws are used. The authors proposed a parameter adaptive law. The applications to the differential drive and the car-like robots are given.

The academic work (Perez-Imaz, Rezeck, Macharet, & Campos, 2016) investigates the coverage path planning problem for the aerial robots. The lawnmower path is created for the three-dimensional coverage with small UAVs. A customized cell decomposition algorithm is used using regular hexagons.

A distributed coverage control algorithm for convex and non-convex environments are proposed in the paper (Hung, Vinh, & Ngo, 2016). The algorithm is based on the interconnected hexagon-lattices based coverage and the hierarchical distributed control of networked multi-robot systems. The demonstration of the algorithm can handle up-to 14 real robots in convex and non-convex real-world environments.

The paper (Pierson et al., 2015) gives a new Voronoi-based multi-agent coverage method with the on-line adaptation to actuator performance variations among the robots. Some of the robots have poor actuation performance while some of them have strong actuation performance. The algorithm gives the weak robots a small region from the environment and the strong robots a large portion of the environment. The authors give a Lyapunov-type proof and simulation results. Experimental results are also given.

Besides the coverage problem, in the paper (Jin, Kim, Wee, & Gans, 2015), a decentralized control method for collision avoidance of the multi-robot systems is proposed. The authors designed a control law such that each robot avoids the closest obstacle in case of any collision. They propose two control laws and give the necessary proofs. Monte Carlo simulations and experimental results are also given.

In the papers (Kwok & Martinez, 2007) and (Kwok & Martinez, 2010) the authors give multi-agent coverage control algorithms under limited energy resources. They use power-weighted Voronoi diagrams and gradient-based control laws. Simulation results are also given.

The paper (Li, Song, Wang, Wang, & Li, 2016) gives a method for complete coverage path planning based on a bounded strategy of integration of the Lorenz dynamic system and robot kinematics equations. According to the simulation results they achieve a higher coverage rate compared to other similar works.

The authors in the paper (Lee, Fekete, & McLurkin, 2016) present a framework for exploration, coverage and surveillance of a swarm of robots with limited capabilities. The area is triangulated so that a single robot handles a small area. The whole environment is approximated by the triangles. A physical data structure is created which is a compact representation of the workspace. Auxiliary data can be written and read by using the data structure. This process provides solving complex problems for the simple robots. Theoretical results are validated with simulations and experiments.

In the paper (Bircher et al., 2016), the authors investigate a three-dimensional coverage path planning method for aerial robots. The authors propose an approach which can calculate short inspection paths with an optimization algorithm with full coverage capability. It can handle both fixed wing and rotorcraft aerial configurations since the motion constraints are taken into account in the optimization step. The algorithm can operate on both mesh and occupancy map based environmental representations. The evaluations are done with simulations and real-life experiments.

In the paper (Janani, Alboul, & Penders, 2016), a cooperative method for a team of robots spraying to a large field is proposed. Using only the robots' local information, task allocation and coordination is achieved.

The paper (Hungerford, Dasgupta, & Guruprasad, 2016) gives a new coverage path planning method using multiple robots in an unknown or partially known environment. They investigated the problem in Voronoi based coverage in presence of unknown obstacles. They proposed a new method that can repartition the initial Voronoi cells and achieve complete coverage. The authors give a proof for the algorithm and according to the proof it guarantees a complete non-overlapping coverage. The simulation results are given in Webots simulator.

In another paper, multi-robot coverage and collision avoidance problems are presented (Breitenmoser & Martinoli, 2016). The authors give a taxonomy of collision scenarious in multi-robot coverage problems.

In the paper (Yehoshua & Agmon, 2016), the authors consider the problem in which one or more robot is required to visit every point of an area involving threats stopping the robots. They propose a multi-robot coverage algorithm for adversarial environments. They give a proof for the algorithm which shows that the algorithm is robust. The effectiveness of the algorithm is evaluated in different environments.

The paper (Jia, Wermelinger, Diethelm, Krüsi, & Hutter, 2016) presents a coverage path planning method for autonomous exploration of legged robots. The environment is divided into cells and each cell is covered by a zig-zag motion pattern. The authors give two simulations of the proposed algorithm in Gazebo.

The paper (Pierson & Schwager, 2016) gives a method that can adapt trust online in response to the relative performance of the robots. The decentralized algorithm controls the positions of the robots. Robots with larger quality sensors have larger regions. They give a Lyapunov-type proof of the method and the results are validated with MATLAB simulations.

In the paper (Olaonipekun & Vaughan, 2015), a complete coverage algorithm for a flexible parent child unit robot is proposed. The complete coverage path is planned by using a combination of input shaping algorithm and the wavefront algorithm. The results show that the coverage process is improved compared to unshaped coverage case.

In another paper (Dogru & Marques, 2015), the coverage path planning problem is studied. The authors take the terrain relief into account and they optimize the solution with Genetic algorithm in terms of energy consumption. Simulation results are given.

In the paper (Mitchell, Corah, Chakraborty, Sycara, & Michael, 2015), the authors solve the Multi-Robot Persistent Coverage Problem. A schedule which enables a team of agents to visit all targets given is generated by the algorithm. The frequency of visitation is maximized and the algorithm maintains a enough fuel capacity. According to the results given, the algorithm performs coverage in response to a given set of targets or tasks. The efficiency of the algorithm is shown.

The authors of the paper (Ramaithitima, Whitzer, Bhattacharya, & Kumar, 2015) propose a solution to the problem of deploying a team of mobile robots where the map is not known. The goal is complete sensor coverage of the environment. The sensing capabilities of the robots are limited and no global information is available. The proposed algorithm is complete and robust to sensor failures. Simulation results are given and show that the algorithm covers and explores unknown indoor environments.

A Stochastic Multi-Robot Persistent Coverage Problem (Mitchell, Chakraborty, Sycara, & Michael, 2015) is solved and an optimal schedule for unmanned aerial vehicles having energy constraints is calculated. The vehicles perform a set of tasks repeatedly while maximizing the frequency of task completion under energy constraints. Simulation and experimental results show that the proposed method gives efficient results.

The authors (Muddu, Wu, & Wu, 2015) propose a multi-robot approach for coverage where the map is not known. The method is based on frontiers. The robots explore and map the environment at the same time. The algorithm enables the robots to go toward frontiers. For implementation, results with Robot Operating System (ROS) give the efficiency of the approach.

The study (Viet, Dang, Choi, & Chung, 2015) gives a method to perform online complete coverage through boustrophedon and backtracing approaches. The robots use only local interactions. To enable complete coverage, the algorithm includes a greedy A* search to move to the closest unvisited region. The efficiency of the algorithm are shown with simulation results.

In another work, a control system for a team of wheeled mobile robots to cover of a ground area is proposed (Rabbath & Léchevin, 2015). The algorithm uses a cost balancing approach. The system is decentralized and robust to single failures. The experimental results are given.

1.3 Hypothesis

The formal statement of the hypothesis is as follows. Consider a team of a number of non-holonomic wheeled mobile agents. The aim is to maximize coverage metric according to locational optimization function by performing collaborative coverage control among the agents. The agents will learn their unknown performance parameters by using the proposed estimators and partition the workspace by using GPDs taking the uncertainty in the localization into account. After, the agents deploy themselves autonomously to the optimal locations and complete the collaborative coverage task by giving the robots with strong actuators greater amount of the regions of the workspace and assigning smaller regions to the weak ones.

2. VORONOI BASED COVERAGE CONTROL FOR MULTIPLE AGENTS

2.1 Preliminaries

A convex region in *N* dimensional space $S \subseteq \mathbb{R}^N$ is defined. A team of n non-holonomic mobile agents are considered. It is assumed that the exact positions of the robots are not precisely known but their locations are within an uncertainty regions. The uncertain region can be defined as a circle or an ellipse as done in this work.

We can represent the region with center p_i and radius r_u is as follows for each agent *i* for the uncertainty circle case (Papatheodorou et al., 2016):

$$C_i^u(p_i, r_u) = \{p \in S | \|p - p_i\| \leq r_u\} \tag{2.1}$$

The definition of the uncertainty ellipse region is given as follows.

The eigenvectors of the covariance matrix form the transformation matrix between the world frame and the coordinate frame T_0^b. The R_0^c is the rotation part of the T_0^b. The eigenvectors of the covariance matrices are orthogonal. Also, the position of the *ith* robot is p_i and the position vectors in world and body frames are given as p^0 and p^b, respectively

$$T_0^b = \begin{bmatrix} R_0^c & p_i \\ 0 & 1 \end{bmatrix} \tag{2.2}$$

$$p^b = (T_0^b)^{-1} p^0 \tag{2.3}$$

$$E_i^u(p_i, \sigma) = \left\{ p^0 \epsilon S \mid \sum_{j=1}^{N} \frac{\left(p_j^b\right)^2}{\sigma_j^2} \leq 1, i = 1,2, \dots n \right\} \tag{2.4}$$

In this part, the Euclidean norm operator is defined as $\|.\|$.

The uniform radial sensing footprints are defined for each robot *i* as below:

$$C_i^s(p_i, r_s) = \{p \in S | \|p - p_i\| \leq r_s\} \tag{2.5}$$

The intersection of the uncertainty circle/ellipse and sensing footprint as the guaranteed sensing region is denoted as given below:

$$C_i^{gs}(C_i^u, C_i^s) = \left\{ \cap_{p_i} C_i^s(p_i, r_s), \forall p_i \in C_i^u \right\} \tag{2.6}$$

$$C_i^{gs}(E_i^u, C_i^s) = \left\{\cap_{p_i} C_i^s(p_i, r_s), \forall p_i \in E_i^u\right\} \tag{2.7}$$

The sensing region in the guaranteed sense for the uncertainty circle and ellipse are given in the equations (2.6) and (2.7), respectively.

2.2 Voronoi Tessellations

The union $\cup_i \ V_i$ defines the Voronoi tessellation of open set S, if $V_i \cap V_j = \emptyset$ for $i \neq j$ and $\cup_i \ V_j = S$ (Luna, J. M., Fierro, R., Abdallah, C. T. and Wood, 2013). The Voronoi region V_i is:

$$V_i = \{q \in S \mid \|q - p_i\| \leq \|q - p_j\|, i \neq j\} \tag{2.8}$$

The points p_i are denoted by the generator or site points. Also, the $\|.\|$ operator is Euclidean norm in $\mathbb{R}^N$ space.

For calculating the classical Voronoi tessellations, the Fortune's Sweepline algorithm can be utilized.

2.3 Power Voronoi Diagrams

If the distance function is selected as the additively weighted power distance or the power distance (Okabe, 1992), then the diagram is called as Power Voronoi Diagram. The Voronoi region V_i corresponding to the Power Voronoi Diagram then can be given as:

$$V_i = \left\{q \in S \mid \|q - p_i\|^2 - w_i \leq \left\|q - p_j\right\|^2 - w_j, i \neq j\right\} \tag{2.9}$$

The bisector of the power diagram is a line. The Voro++ library (Rycroft, n.d.) can be used for drawing the power diagrams.

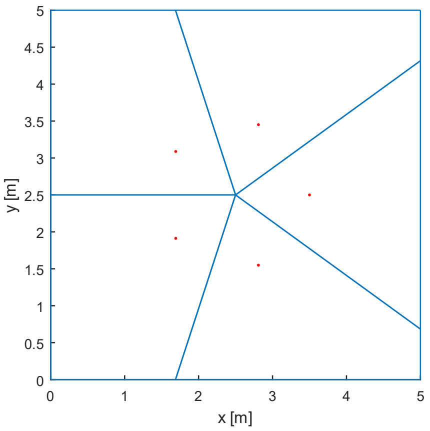

Figure 2.1 : Power Voronoi Diagram (all weights are zero).

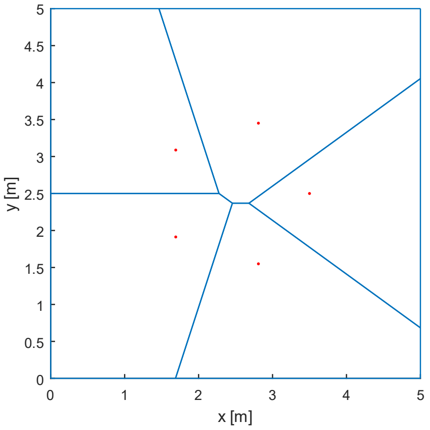

Figure 2.2 : Power Voronoi Diagram ($\omega_1 = 0.5$).

In Figure 2.1 and Figure 2.2, two example power diagrams are given. As the weight parameter is changed, the areas of the regions of the generator points also change with respect to the given weights.

2.4 Guaranteed Voronoi Diagrams

The Guaranteed Voronoi diagram (GVD) can be defined as the Voronoi tessellation of the generator points where the positions of the site points are within a particular region only, for example, uncertainty circles. The set $U = \{U_1, U_2, \ldots U_n\}$ represents the uncertain regions. For GVD, the points in the cells are the closest points to the corresponding site points in a guaranteed sense (Evans & Sember, 2008).

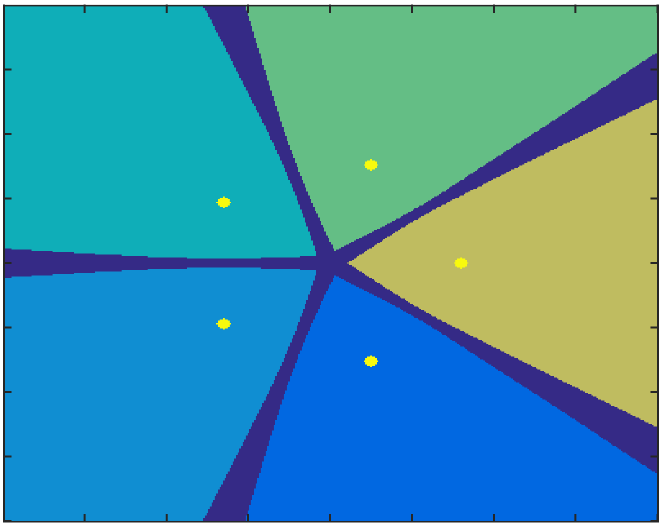

Figure 2.3 : Guaranteed Voronoi Diagram of Disks.

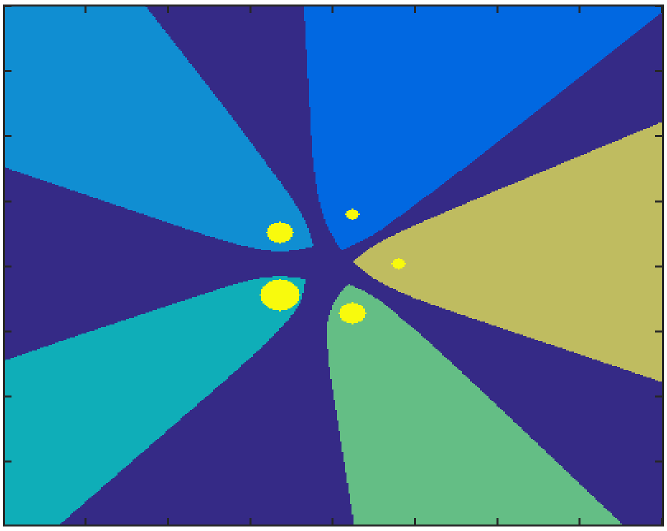

Figure 2.4 : Guaranteed Voronoi Diagram of Disks with Different Radii.

The set of points that are at least as close to the region U_i as to the region U_j in a guaranteed sense can be represented as follows:

$$H_{ij}^{g} = \left\{ \begin{matrix} p \in S \mid \|p - p_i\| \leq \|p - p_j\|, \\ \forall p_i \in U_i, \forall p_j \in U_j \end{matrix} \right\} \tag{2.10}$$

$$V_i^{g} = \bigcap_{j \neq i} H_{ij}^{g}$$

$$V_i^{g} = \left\{ \begin{matrix} p \in S \mid \max\|p - p_i\| \leq min\|p - p_j\|, \\ i \neq j, j = 1,2, \dots n, \forall p_i \in U_i, \forall p_j \in U_j \end{matrix} \right\} \tag{2.11}$$

The equation (2.11) gives the guaranteed Voronoi cell V_i^g. Two branches of a hyperbola define the bisector curve of the two guaranteed Voronoi cells as shown in (Evans & Sember, 2008).

Figure 2.3 and Figure 2.4 give the two examples of the Guaranteed Voronoi Diagrams of disks. In the first diagram, the radii of disks are equal while the second diagram represents the GVD of uncertainty disks with different radii. By increasing the radius of the disks, the effect is increase of the uncertainty. The effect can be investigated from the amount of the areas of the neutral regions.

2.5 Guaranteed Voronoi Diagrams of Ellipses

If the uncertain regions are defined as ellipses the bisector curves should be calculated in order to draw the diagram. Here, the coordinate frame of the ellipse is coincident with the coordinate frame formed by the eigenvectors of the covariance matrix as given in equation (2.4),

The Guaranteed Voronoi Diagram has its bisectors in the form:

$$\langle i,j\rangle = \left\{p \in S \mid \max_{p_i} d(p,p_i) = \min_{p_j} d(p,p_j)\,, \forall p_i \in U_i, \forall p_j \in U_j\right\} \tag{2.12}$$

The region U_i is defined as an ellipse. Thus, the closest and farthest points of the point on the bisector to the ellipse should be found.

The parametric ellipse equation is given as below (S. Chen, Xin, He, & Wang, 2012):

$$\begin{bmatrix} x(\theta) \\ y(\theta) \end{bmatrix} = \cos(\theta) \begin{bmatrix} a_x \\ a_y \end{bmatrix} + \sin(\theta) \begin{bmatrix} b_x \\ b_y \end{bmatrix} + \begin{bmatrix} c_x \\ c_y \end{bmatrix} \tag{2.13}$$

Here, $[a_x \quad a_y]^T$ represents the major axis, $[b_x \quad b_y]^T$ is given as the minor axis and $[c_x \quad c_y]^T$ is the center point of the ellipse.

The definition of the parameters *A*, *B*, *C* and *D* are given in (S. Chen et al., 2012). The coefficients *a*, *b*, ϕ and t can be denoted as:

$$\phi = \frac{1}{2} atan2(B,A) \tag{2.14}$$

$$t = \theta - \phi \tag{2.15}$$

$$a = \frac{C\, cos\phi - D\, sin\phi}{\sqrt{A^2+B^2}} \tag{2.16}$$

$$b = \frac{C\, sin\phi + D\, cos\phi}{\sqrt{A^2+B^2}} \tag{2.17}$$

If the variables $u = \cos(t)$ and $v = \sin(t)$ are defined, one can get the quartic equation giving the closest and farthest point solutions:

$$4u^4 - 4bu^3 + (a^2 + b^2 - 4)u^2 + 2bu + 1 - a^2 = 0 \tag{2.18}$$

$$v = \frac{2u^2 - bu - 1}{a} \tag{2.19}$$

After the solution of the quartic equation is found by using (2.18) and (2.19), the parameter θ is obtained. So, by using the closest and farthest points, an algorithm is proposed for checking the bisector condition for each grid point.

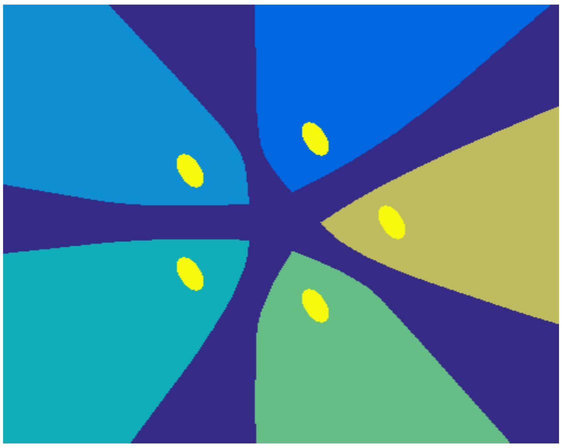

Figure 2.5 : Example Guaranteed Voronoi Diagram of Ellipses.

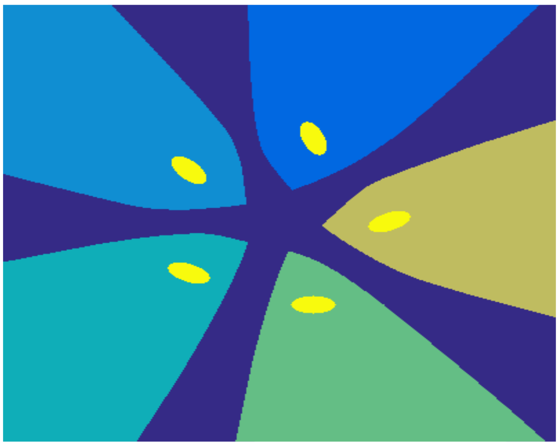

Figure 2.6 : Example Guaranteed Voronoi Diagram of Ellipses.

Figure 2.5 and Figure 2.6 give two examples of Guaranteed Voronoi Diagram of Ellipses. The ellipse parameters cause the bisector curves to change. The major and the minor axes of the ellipse determine the size of the uncertain region. The axes of ellipse form the base vectors of R_0^c together with the uncertainty radii σ_i. Also, they affect the size of the neutral area.

```
For each site point $i$
 For each neighbor $j$ of the site point $i$
  For each point p in the grid
   Calculate $\max_{p_i} d(p, p_i)$ using the quartic equation
   Calculate $\min_{p_j} d(p, p_j)$ using the quartic equation
   If the both distances are equal
    Add point to the bisector curve of $i$ and $j$
   End
  End
 End
End
```

Figure 2.7 : Algorithm for GVDs of Ellipses.

A bisector curve finding algorithm is given in Figure 2.7.

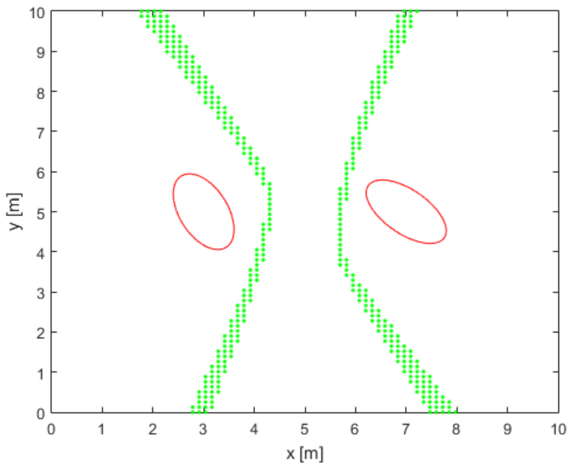

Figure 2.8 : Bisectors of a GVD of Ellipses.

In Figure 2.8, an example algorithm output for the bisectors of the two ellipses is given. The bisectors are asymmetric and different from the uncertainty circle case because of the regions are two ellipses with different orientations.

2.6 Guaranteed Power Voronoi Diagrams

The weighted Voronoi diagrams are the types of Voronoi tessellations in which the weights are used in distance function. The areas of the cells can be changed with respect to the weights. An additive term added to the squared Euclidean distance function in the GVD case results in the diagram that is called as the Guaranteed Power Diagram (GPD or GPVD) (Mahboubi et al., 2014). The r_i is the radius of the uncertainty. Also, a GPD-cell can be defined as below:

$$W_i^g = \begin{Bmatrix} p \in S \mid (\|p - p_i\| + r_i)^2 - \omega_i \leq \left(\|p - p_j\| - r_j\right)^2 - \omega_j, \\ i \neq j, j = 1,2, \dots n \end{Bmatrix} \tag{2.20}$$

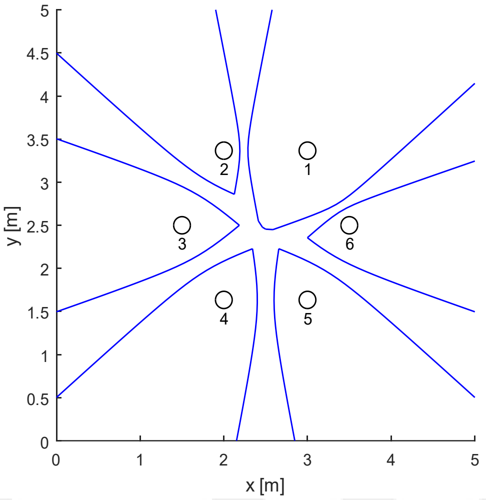

Figure 2.9 : Example GPVD with $\omega_1 \neq 0$.

Figure 2.9 gives an example GPVD. It can be seen that the region of the cell of first agent increased with respect to its increased weight parameter.

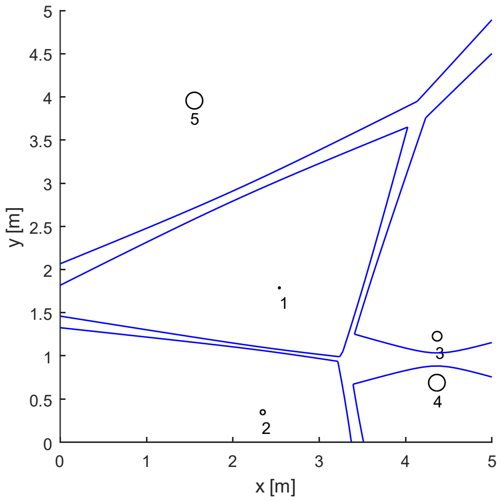

Figure 2.10 : Example GPVD with different radii.

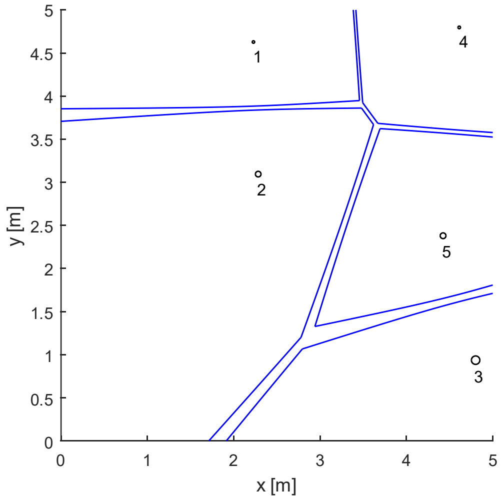

Figure 2.11 : Example GPVD with different radii.

Figure 2.10 and Figure 2.11 show two example GPVDs with different radii. By changing the radius parameter, the bisector curves and the uncertain regions change.

A previous work using GPVDs and adaptive coverage control is given in the paper (Turanli & Temeltas, 2018a).

2.7 Optimal Coverage Formulation with Locational Optimization

If $S \subseteq R^N$ is defined as a bounded environment, $\phi: R^N \to R^+$ as a density function, and $\ell: R^+ \to R$ as a non-decreasing performance function, then, the locational optimization function $\mathfrak{I}$ can be denoted as follows:

$$\mathfrak{I}(p_1, p_2, \dots p_m) = \sum_{i=1}^{m} \int_{V_i} (\ell(\|q - p_i\|)\phi(q) - w_i)dq \tag{2.21}$$

Here, the V_i is defined as the Voronoi region *i*, *m* is the number of the generator points, w_i is the weight of the *i*th cell and the generator point of the Voronoi cell can be represented as p_i.

The centroid C_{V_i} and mass M_{V_i} of a Voronoi region are defined as below (Luna, J. M., Fierro, R., Abdallah, C. T. and Wood, 2013):

$$C_{V_i} = \frac{1}{M_{V_i}} \int_{V_i} q\phi(q)dq \tag{2.22}$$

$$M_{V_i} = \int_{V_i} \phi(q)dq \tag{2.23}$$

The performance function is chosen as the squared distance $\ell(\|q - p_i\|) = \|q - p_i\|^2$. The locational optimization function $\Im$ then becomes:

$$\Im(p_1, p_2, \dots p_m) = \sum_{i=1}^{m} \int_{\mathrm{V}_i} (\|q - p_i\|^2 \phi(q) - w_i) dq \tag{2.24}$$

Now, to solve the optimization, the closed form solution of the optimal coverage should be found by using the locational optimization function. For this purpose, the partial derivatives of the function $\Im$ with respect to p_i are taken. In the resulting case, the centroid positions given in (2.22) are local minimizers of the locational optimization function given in (2.24). So, the optimal solution is the centroid positions.

For the simplest case which is the point dynamics, we can get the control law for *ith* agent as follows:

$$v_i = K_p(C_{\mathrm{V}_i} - p_i) \tag{2.25}$$

Here, p_i represents the position of the *ith* robot and K_p denotes a positive-definite coefficient matrix.

2.8 Discussion

In this section, the Voronoi-based coverage control method is explained with preliminary information, definitions of the Voronoi diagram, Power diagram, guaranteed Voronoi diagram and guaranteed Power diagrams. Also, the guaranteed Voronoi diagram of ellipses is introduced. Then, the optimal coverage formulation with locational optimization is summarized with the definition of a simple control law in a point dynamics case.

3. ADAPTIVE COVERAGE CONTROL WITH IMPRECISE LOCALIZATION

In this section, an adaptive coverage control algorithm is proposed taking the imprecise localization into account by using GPVDs (Turanli & Temeltas, 2018a).

The problem statement is as follows. Given a number of non-holonomic wheeled mobile robots, the task is to achieve the optimal coverage control by using the proposed adaptive coordination algorithm. The agents are driven to optimal configurations by using locational optimization function and adaptive estimator with a linear consensus protocol. They estimate the density function by sharing the estimation vectors among each other and are driven to the optimal locations by using the non-holonomic control law. Also, the GPVDs enable to take the localization uncertainty of the agents into account.

In the following subsections, the coverage control algorithm, the non-holonomic control law and adaptation law are explained. Then, the Lyapunov-type stability proof of the control and adaptation law is given.

3.1 Coverage Control with GPVDs

In this section, an adaptive control law which is move-to-centroid type with linear consensus protocol for a unicycle mobile robot model is presented.

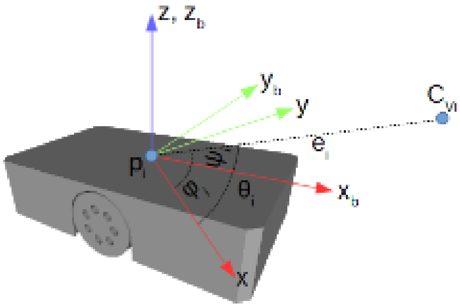

Figure 3.1 : Position of the agent and model parameters.

3.2 Control Law for Non-holonomic Agents

The proposed algorithm consists of the control law presented in the work (Luna, J. M., Fierro, R., Abdallah, C. T. and Wood, 2013) and it uses the unicycle model used in (Aicardi, Casalino, Bicchi, & Balestrino, 1995). The agents achieve the optimal centroid locations by using the aforementioned control laws. Also, the unicycle model is given below:

$$\begin{pmatrix} \dot{e}_i \\ \dot{\psi}_i \\ \dot{\theta}_i \end{pmatrix} = \begin{pmatrix} -u_i \, cos\psi_i \\ -\omega_i + u_i \frac{sin\psi_i}{e_i} \\ u_i \, \frac{sin\psi_i}{e_i} \end{pmatrix} \tag{3.1}$$

The non-holonomic control law is shown as follows:

$$\begin{pmatrix} u_i \\ \omega_i \end{pmatrix} = \begin{pmatrix} (\gamma cos\psi_i)e_i \\ 2\gamma sin\psi_i cos\psi_i + \lambda(\psi_i + \theta_i) \end{pmatrix} \tag{3.2}$$

The p_i and C_{V_i} represent the location of the agent and the centroid location for the agent i, respectively, as shown in Figure 3.1. Also, the linear and angular control signals are denoted as u_i and ω_i. The heading angle is represented by φ_i, the distance between the robot and the corresponding centroid is given as e_i. Finally, ψ_i gives the angle between the robot and the centroid.

Here, the control parameters are $\gamma > 0$ and $\lambda > 0$. The optimal configuration is achieved by driving the agents to the centroid locations C_{V_i} by using the proposed control laws.

3.3 Non-holonomic Adaptive Coverage Control

In the adaptive coverage control algorithm, The vector function $Z \in S \to \mathbb{R}_+^m$ together with the parameter vector $\varsigma \in S \to \mathbb{R}_+^m$ are used as given in (Luna, J. M., Fierro, R., Abdallah, C. T. and Wood, 2013):

$$\phi(q) = \mathrm{Z}^T(q)\varsigma \tag{3.3}$$

The lower bound condition for the estimation vector should be satisfied by the algorithm for each element of vector ς:

$$\varsigma(j) \geq \alpha \, , for \; j = 1, 2, \ldots m \tag{3.4}$$

$$\hat{\phi}_i(q) = \mathrm{Z}^T(q)\hat{\varsigma}_i \tag{3.5}$$

$$\tilde{\varsigma}_i = \hat{\varsigma}_i - \varsigma_i \tag{3.6}$$

The distributed density function is estimated by the adaptation law given below (Luna, J. M., Fierro, R., Abdallah, C. T. and Wood, 2013):

$$\dot{\hat{\varsigma}}_i = \mathrm{K_2}^{-1}(\mathrm{I} - \Xi_{\mathrm{pr}})\dot{\hat{\varsigma}}_{p,i} \tag{3.7}$$

$$\dot{\hat{\varsigma}}_{p,i} = -\xi\mathrm{ZZ^T}\tilde{\varsigma}_i - \eta\sum\nolimits_{j\in\mathcal{L}_i}(\hat{\varsigma}_i - \hat{\varsigma}_j) \tag{3.8}$$

Here, the adaptation gains are represented as $\xi > 0$ and $\eta > 0$. I gives the identity matrix, $\mathrm{K_2}^{-1}$ denotes a positive definite matrix, Z gives the vector function in (3.3). The $\mathcal{L}_i$ is the set of the neighbors of the *ith* agent. Also, a projection law Ξ_{pr} is defined as shown below:

$$\Xi_{pr,i}(j) = \begin{cases} 0, & for\ \hat{\varsigma}_i(j) > \alpha \\ 0, & for\ \hat{\varsigma}_i(j) = \alpha\ and\ \dot{\hat{\varsigma}}_{p,i} \geq 0 \\ 1, & otherwise \end{cases} \tag{3.9}$$

A diagonal entry of the $\Xi_{pr,i}$ is denoted by $\Xi_{pr,i}(i)$. The gradient estimator is given as the first term in (3.8) as defined in (Slotine, J.-J. E. and Li, 1991) and the term from the consensus protocol is given in the second term (Luna, J. M., Fierro, R., Abdallah, C. T. and Wood, 2013). The next section presents the stability analysis of the control and adaptation laws.

A linear consensus protocol is utilized for estimation of the parameter vector since it speeds up the convergence of the estimator algorithm and enables to share the estimated information among the agents.

3.4 Stability Analysis

The Lyapunov stability analysis of the proposed controller and estimator is done in this section. The dynamics in (3.1) and the control laws (3.2)-(3.8) are considered for each agent. To start the proof, first, the definition of the m_i is given as below:

$$m_i = p_i - \hat{C}_{V_i} = -\begin{pmatrix} e_i\cos(\psi_i + \varphi_i) \\ e_i\sin(\psi_i + \varphi_i) \end{pmatrix} \tag{3.10}$$

A Lyapunov function candidate can be defined as below:

$$V = \frac{1}{2}\sum_i \begin{pmatrix} {m_i}^T \mathrm{K}_1 m_i + {\tilde{\varsigma}_i}^T \mathrm{K}_2 \tilde{\varsigma}_i + \\ \kappa_3 (\psi_i + \theta_i)^2 \end{pmatrix} \tag{3.11}$$

Here, two positive definite matrices $\mathrm{K}_1 = diag(k_1) \in \mathbb{R}^{2x2}$ and $\mathrm{K}_2 \in \mathbb{R}^{mxm}$ are given and κ_3 is defined as a positive constant. If we take the derivative of the Lyapunov candidate, the equations yield:

$$\dot{V} = \sum_i \begin{pmatrix} {m_i}^T \mathrm{K}_1 \dot{m}_i + {\tilde{\varsigma}_i}^T \mathrm{K}_2 \dot{\tilde{\varsigma}}_i \\ +\kappa_3 \begin{pmatrix} \psi_i \dot{\psi}_i + \psi_i \dot{\theta}_i \\ +\dot{\psi}_i \theta_i + \theta_i \dot{\theta}_i \end{pmatrix} \end{pmatrix} \tag{3.12}$$

After substituting the equation (3.7) into (3.12), the derivative of the Lyapunov function is becomes:

$$\dot{V} = \sum_i \begin{pmatrix} {m_i}^T \mathrm{K}_1 \dot{m}_i - {\tilde{\varsigma}_i}^T \Xi_{pr,i} \dot{\hat{\varsigma}}_{p,i} \\ +\kappa_3 \begin{pmatrix} \psi_i \dot{\psi}_i + \psi_i \dot{\theta}_i \\ +\dot{\psi}_i \theta_i + \theta_i \dot{\theta}_i \end{pmatrix} \\ -\xi {\tilde{\varsigma}_i}^T \mathrm{Z}\mathrm{Z}^{\mathrm{T}} \tilde{\varsigma}_i \\ -\eta {\tilde{\varsigma}_i}^T \sum_{j \in \mathcal{L}_i} (\hat{\varsigma}_i - \hat{\varsigma}_j) \end{pmatrix} \tag{3.13}$$

The first term is negative semi-definite as given in the proof in (Luna, J. M., Fierro, R., Abdallah, C. T. and Wood, 2013).

$$\dot{m}_i = -\begin{pmatrix} \gamma cos\psi_i e_i \cos{(\psi_i + \varphi_i)} \\ \gamma cos\psi_i e_i \sin{(\psi_i + \varphi_i)} \end{pmatrix} \tag{3.14}$$

$$\sum_i {m_i}^T \mathrm{K}_1 \dot{m}_i = -\sum_i \gamma k_1 {e_i}^2 cos^2 \psi_i \leq 0 \tag{3.15}$$

The second term gives a negative semi-definite result as below:

$$-\sum_{i=1}^{n} {\tilde{\varsigma}_i}^T \Xi_{pr,i} \dot{\hat{\varsigma}}_{p,i} \ \leq 0 \tag{3.16}$$

The third term becomes also negative semi-definite:

$$\sum_i \kappa_3 \begin{pmatrix} \psi_i \dot{\psi}_i + \psi_i \dot{\theta}_i \\ +\dot{\psi}_i \theta_i + \theta_i \dot{\theta}_i \end{pmatrix}$$

$$= \sum_i \kappa_3 (\psi_i + \theta_i) \left(2u_i \frac{sin\psi_i}{e_i} - \omega_i \right)$$

$$= -\sum_i \kappa_3 \lambda (\psi_i + \theta_i)^2 \leq 0 \tag{3.17}$$

Similiarly, the fourth term yields a negative semi-definite result:

$$-\xi {\tilde{\varsigma}_i}^T \mathrm{Z}\mathrm{Z}^{\mathrm{T}} \tilde{\varsigma}_i \leq 0 \tag{3.18}$$

For the fifth term, the proof for the linear consensus protocol is derived from the detailed proof for the binary consensus protocol in (G. Chen & Lewis, 2011). Here, the cell (i, j) of the adjacency matrix is denoted by A_{ij}.

$$-\eta \sum_{i} \tilde{\varsigma}_i^{\,T} \sum_{j\in\mathcal{L}_i} (\hat{\varsigma}_i - \hat{\varsigma}_j)$$

$$= -\frac{1}{2}\eta \sum_{i} \sum_{j\in\mathcal{L}_i} A_{ij} \left(\hat{\varsigma}_i - \hat{\varsigma}_j\right)^T \left(\hat{\varsigma}_i - \hat{\varsigma}_j\right)$$

$$\leq -\tfrac{1}{2}\eta \textstyle\sum_{i} \sum_{j\in\mathcal{L}_i} \left\|\hat{\varsigma}_i - \hat{\varsigma}_j\right\|^2 \tag{3.19}$$

All in all, $\dot{V} \leq 0$ is negative semi-definite and $\ddot{V}$ is bounded because V is positive definite and lower bounded. As a result, we can apply the the Barbalat's Lemma and see that the tracking errors e_i, $|\psi_i|$, $|\theta_i|$ together with the estimation errors $\tilde{\varsigma}_i$ and $\left\|\hat{\varsigma}_i - \hat{\varsigma}_j\right\|$ go to zero as $t \to \infty$. Thus, the result is that the system is asymptotically stable.

3.5 Discussion

In this section, an adaptive coverage control with localization uncertainty is introduced. The algorithm includes a non-holonomic move-to-centroid control law for unicycle vehicles, a parameter estimator, guaranteed power diagram for handling uncertain positions of the agents and a consensus protocol. The Lyapunov-type stability proof is given.

4. ADAPTATION TO PERFORMANCE VARIATIONS WITH GPVDs

In this section, the control and adaptation laws for a non-holonomic agent model (Turanli & Temeltas, 2018b) are given within the multi-agent collaboration algorithm.

The problem statement is as follows. The problem is to achieve the optimal coverage control by using the proposed adaptive coordination algorithm with a number of non-holonomic wheeled mobile agents. The agents perform self-deployment according to the locational optimization function by using the adaptive collaboration algorithm. They learn their performance parameters online which are not known beforehand. The collaboration algorithm gives greater regions to the agents with stronger actuation performances while giving smaller regions to the ones with poor actuators according to the performance parameters in a decentralized way. Here, the good actuation performances can be defined as favorable terrain and powerful motors while wheel slip and weak motors can be counted as poor performances. Also, the GPVDs enable to take the localization uncertainty of the agents into account.

In the following subsections, the coverage control algorithm, the non-holonomic control law and adaptation law are explained. Lastly, the Lyapunov-type stability proof of the control and adaptation law is given.

4.1 Control Laws

The robots move to the centroid locations by using the control law in (Luna, J. M., Fierro, R., Abdallah, C. T. and Wood, 2013). For the control law, a unicycle model given in equation (3.1) is used.

The proposed control law is shown as below:

$$\begin{pmatrix} u_i \\ \omega_i \end{pmatrix} = \begin{pmatrix} (\gamma\, cos\psi_i)e_i \\ \gamma\, sin\psi_i cos\psi_i + \lambda\, \psi_i \end{pmatrix} \tag{4.1}$$

Figure 3.1 gives the agent position p_i and the position of the centroid for the agent i C_{V_i}, respectively. φ_i represents the yaw angle, e_i is the distance between the agent and the centroid, ψ_i gives the angle between the agent and the centroid. Also, u_i and ω_i are the linear and angular control inputs, respectively.

Here, $\gamma > 0$ and $\lambda > 0$ are the control parameters. The agents position themselves to the centroid C_{V_i} locations by using the proposed control law.

4.2 Adaptation Laws

The collaboration according to the actuation performance variations is achieved by using the base collaboration method given in (Pierson et al., 2015). Different from the base method, the estimation model and the control laws are non-holonomic.

We can start defining the positive matrix *k* as follows:

$$k = (\gamma \quad \lambda)^T \tag{4.2}$$

The uncertainty matrix of actuation is represented as shown below:

$$k_\Delta = (k_{\Delta_1} \quad k_{\Delta_2})^T \tag{4.3}$$

The parameter to be estimated can be given as follows:

$$K_i = k + k_\Delta \tag{4.4}$$

The weight adaptation law for weight w_i of the agent *i* is:

$$\dot{w}_i = -\frac{k_\omega}{\mathrm{M}_{\mathrm{V}_i}} \sum_{j \in N_i} \left(\left(w_i - f(\widehat{K}_i) \right) - \left(w_j - f(\widehat{K}_j) \right) \right) \tag{4.5}$$

where the set of the neighbors of the agent *i* is denoted by N_i.

In order to estimate the unknown actuator parameters, the non-holonomic system dynamics should be considered. We can write the nominal closed loop system as follows:

$$\dot{x}_i = Yk \tag{4.6}$$

The definition of the actuator uncertainty is as below:

$$\dot{x}_i = Y(k + k_\Delta) = Yk + \Delta_i \tag{4.7}$$

$$K_i = k + k_\Delta \tag{4.8}$$

From the differential drive kinematic model and the non-holonomic control law, the closed loop system matrix is derived as *W*:

$$Y = \begin{pmatrix} \cos(\psi_i)\, e_i \cos(\varphi_i) & 0 \\ \cos(\psi_i)\, e_i \sin(\varphi_i) & 0 \\ \sin(\psi_i) \cos(\psi_i) & \psi_i \end{pmatrix} \tag{4.9}$$

The estimator equations can be written as follows:

$$\dot{\hat{K}}_i = \lambda_i - \Lambda_i \hat{K}_i \tag{4.10}$$

$$\dot{\lambda}_i = Y^T \dot{x}_i \tag{4.11}$$

$$\dot{\Lambda}_i = Y^T Y \tag{4.12}$$

$$\dot{\hat{K}}_i = (\int Y^T Y dt) K_i - (\int Y^T Y dt) \hat{K}_i \tag{4.13}$$

$$\dot{\hat{K}}_i = (\int Y^T Y dt)(K_i - \hat{K}_i) = -\Lambda_i \tilde{K}_i \tag{4.14}$$

The simplification of the equations gives the equation (4.14). However, for online calculations, the equations (4.10) - (4.12) should be used.

4.3 Stability Analysis

For the stability proof of the closed loop system, first a Lyapunov function candidate should be proposed. Then, we show that for the position and estimation laws, the following results are obtained by using the LaSalle's Invariance Principle:

$$e_i \to 0, \psi_i \to 0 \; as \; t \to \infty \tag{4.15}$$

$$\tilde{K}_i \Lambda_i \to 0 \; as \; t \to \infty \tag{4.16}$$

Let us define κ_1 and κ_2 as positive constants. The Lyapunov function candidate can be written as below:

$$m_i = p_i - C_{V_i} = -\begin{pmatrix} e_i \cos(\psi_i + \varphi_i) \\ e_i \sin(\psi_i + \varphi_i) \end{pmatrix} \tag{4.17}$$

$$V = \frac{1}{2} \sum_i \begin{pmatrix} {m_i}^T \kappa_1 m_i + \kappa_2 {\psi_i}^2 + \\ w_i + Tr\left[\tilde{K}_i {\tilde{K}_i}^T\right] \end{pmatrix} \tag{4.18}$$

After the derivative of the Lyapunov function is taken, the equation yields:

$$\dot{V} = \sum_i \begin{pmatrix} {m_i}^T \kappa_1 \dot{m}_i + \kappa_2 \psi_i \dot{\psi}_i + \\ \dot{w}_i + Tr\left[-\tilde{K}_i \Lambda_i {\tilde{K}_i}^T\right] \end{pmatrix} \tag{4.19}$$

The first term in the time-derivative of the Lyapunov function gives a negative semi-definite result:

$$\sum_i \; m_i{}^T \kappa_1 \dot{m}_i = -\sum_i \gamma \kappa_1 e_i{}^2 cos^2\, \psi_i \leq 0 \tag{4.20}$$

The second term of the function is also negative semi-definite as follows:

$$\sum_i \kappa_2 \psi_i \dot{\psi}_i = \sum_i \kappa_2 \psi_i (-(\gamma\; sin\psi_i cos\psi_i + \lambda\, \psi_i) + \gamma\; cos\psi_i\; sin\psi_i)$$

$$= -\sum_i \kappa_2\, \lambda\, \psi_i{}^2 \leq 0 \tag{4.21}$$

The third term is zero as given below:

$$\sum_i \; \dot{w}_i = \sum_i \left[-\frac{k_\omega}{\mathrm{Mv}_i} \sum_{j \in N_i} \begin{pmatrix} \left(w_i - f(\widehat{K}_i)\right) - \\ \left(w_j - f(\widehat{K}_j)\right) \end{pmatrix} \right] = 0 \tag{4.22}$$

It can be seen that the last term is negative semi-definite, as well:

$$\sum_i \; Tr\left[-\widetilde{K}_i \Lambda_i \widetilde{K}_i{}^T \right] \leq 0 \tag{4.23}$$

In brief, the derivative of the Lyapunov candidate is shown to be negative semi-definite ($\dot{V} \leq 0$).

Here, we conclude that the trajectories are bounded, e_i, ψ_i and $\widetilde{K}_i$ are also bounded. Thus, the estimation vector $\widehat{K}_i$ is bounded.

Also, the boundedness of the weights can be proven. The main proof is shown in the work (Pierson et al., 2015). Additionally, in steady state, the convergence of the weights to constants can be proven from the stable filter theory. Assuming that Λ_i becomes full rank for all agents and for all t > 0, the following result is obtained:

$$\lim_{t \to \infty} \left(\widehat{K}_i\right) = K_i \tag{4.24}$$

$$\lim_{t \to \infty} \left(w_i - w_j\right) = f(K_i) - f\left(K_j\right) \tag{4.25}$$

In the steady state, $w_i - f(K_i)$ converge to a common value among all agents.

To complete the proof, we should find the largest invariant in $\dot{V} = 0$. Here, $\dot{V} = 0$ occurs only when $e_i = 0$, $\psi_i = 0$ and $\widetilde{K}_i \Lambda_i = 0$. From the control and estimation laws the result is that this set is an invariant set.

We should also investigate the internal dynamic φ_i. If we consider the equilibrium point of the system:

$$\psi_i = 0 \tag{4.26}$$

By using the small angle approximation;

$$\dot{\varphi}_i = \gamma \sin(\theta_i - \varphi_i) \cos(\theta_i - \varphi_i) \tag{4.41}$$

$$\dot{\varphi}_i = \gamma(\theta_i - \varphi_i) \tag{4.42}$$

$$\dot{\varphi}_i + \gamma\varphi_i = \gamma\theta_i \tag{4.43}$$

The θ_i is shown to be bounded because $\theta_i = \angle\left(C_{V_i} - p_i\right)$ is the angular distance between the agent and the centroid. Thus, the boundedness and stability of φ_i is shown.

So, it can be seen that from the LaSalle's Invariance Principle, the asymptotical convergence of the errors e_i and ψ_i along with the estimation error $\widetilde{K}_i$ is achieved as $t \to \infty$. Moreover, the internal dynamics φ_i and θ_i are stable.

4.4 Discussion

In this section, a coverage-based multi-agent collaboration algorithm which adapts to actuation uncertainties is introduced. The algorithm consists of a non-holonomic move-to-centroid control law, an online parameter estimator for learning the performances of individual agents, and a weight estimator for adjusting guaranteed power regions. By utilization of the algorithm, the localization uncertainty is taken into account by using the guaranteed power diagrams and meanwhile, the regions of the agents can be assigned according to their acuation performances. Thus, the collaboration among the agents is achieved. At last, the Lyapunov-type stability proof is given.

5. ADAPTATION TO PERFORMANCE VARIATIONS WITH HNNs UNDER LOCALIZATION UNCERTAINTY

In this section, the move-to-centroid control and adaptation laws for the method with the multi-agent collaboration method with Hopfield Neural Networks (HNN) are given (Turanli & Temeltas, 2019, 2020).

The problem statement is as follows. The problem is to deploy the non-holonomic wheeled mobile agents autonomously by using the optimal coverage control and the proposed adaptive coordination algorithm with HNNs. The locational optimization function and the adaptive collaboration algorithm enable the robots to estimate their performance parameters online which are not known beforehand and to compansate the performance differences by giving larger regions to the agents with better actuation performances while giving smaller regions to the ones with poor actuators in a decentralized way. In addition, the GPVDs allow to take the localization uncertainty of the agents into account.

The following subsections explain the coverage control algorithm, the control law and adaptation law. Lastly, the Lyapunov-type stability proof of the control and adaptation law is given.

5.1 Hopfield Neural Networks

An online parameter estimation method is utilized to calculate the performance parameters of the agents. The Hopfield Neural Network is the prefered method to carry out the online parameter estimation as proposed in (Alonso et al., 2009) because accuracy limitations may occur in other approaches like classification.

In a network of M neurons, we define z_i as the total input to neuron i, s_j can be represented as the output of the neuron j, the weight from neuron j to neuron i denoted by W_{ij} and B_i is given as the bias of the neuron i. Then, we can write the state equations of the HNN as follows:

$$\frac{dz_i}{dt}(t) = -\left(\sum_{j=1}^{M} W_{ij}(t)s_j(t) + B_i(t)\right) \tag{5.1}$$

$$s_i(t) = \alpha \tanh\left(\frac{z_i(t)}{\beta}\right) \tag{5.2}$$

The equation (5.2) gives the total input-state equation where $\alpha, \beta > 0$ and $i \in M$.

$$\frac{dz}{dt}(t) = -(W(t)s(t) + B(t)) \tag{5.3}$$

$$s(t) = \alpha \tanh\left(\frac{z(t)}{\beta}\right) \tag{5.4}$$

The equations (5.3) and (5.4) show the matrix representation of the HNN.

5.2 Point-Offset Control Law

The controller for the non-holonomic agents consists of a reference point *P* and an distance *l* from the center of the agent (Michael & Kumar, 2009; Pierson & Schwager, 2015). The linear v and angular ω velocities are obtained from the velocity of the reference point *P* by using the matrix transformation shown below. Here, θ denotes the heading angle of the robot.

$$\begin{pmatrix} v \\ \omega \end{pmatrix} = \begin{pmatrix} cos\theta & sin\theta \\ -\frac{sin\theta}{l} & \frac{cos\theta}{l} \end{pmatrix} \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} \tag{5.5}$$

The collision check should be made by using the center *P* with the radius $\rho = l + r_{robot}$, where r_{robot} is the radius of the robot to provide collision-free motion.

5.3 HNN Parameter Estimator

We should transform the system given in (5.6) with the control law in (2.25) into linear in parameters form (LIP) in order to estimate its parameters:

$$\dot{p}_i = K_p(C_{\mathrm{V}_i} - p_i) \tag{5.6}$$

$$\dot{p}_i = K_p C_{\mathrm{V}_i} - K_p p_i \tag{5.7}$$

$$p_i = -K_p{}^{-1}\dot{p}_i + C_{\mathrm{V}_i} \tag{5.8}$$

$$y = p_i - C_{\mathrm{V}_i} = -K_p{}^{-1}\dot{p}_i \tag{5.9}$$

$$y = -\begin{pmatrix} 1/K_{p,1} & 0 \\ 0 & 1/K_{p,2} \end{pmatrix} \dot{p}_i \tag{5.10}$$

$$y = \begin{pmatrix} -\dot{p}_{i,x} & 0 \\ 0 & -\dot{p}_{i,y} \end{pmatrix} \begin{pmatrix} 1/K_{p,1} \\ 1/K_{p,2} \end{pmatrix} \tag{5.11}$$

Then, the parameter estimation vector can be defined as below:

$$\theta_{est} = \begin{pmatrix} 1/K_{p,1} \\ 1/K_{p,2} \end{pmatrix} \tag{5.12}$$

The LIP form of the system yields:

$$A = \begin{pmatrix} -\dot{p}_{i,x} & 0 \\ 0 & -\dot{p}_{i,y} \end{pmatrix} \tag{5.13}$$

$$y = p_i - C_{\mathrm{V}_i}. \tag{5.14}$$

$$y = A\,\theta_{est} \tag{5.15}$$

We can obtain the weight matrix W and bias B in equation (5.3) as done in the following equations:

$$W = A^T A \tag{5.16}$$

$$B = -A^T y \tag{5.17}$$

$$\hat{\theta}_{est} = \begin{pmatrix} s_1(t) \\ s_2(t) \end{pmatrix} \tag{5.18}$$

$$\widehat{K}_i = \begin{pmatrix} 1/\hat{\theta}_{est,1} \\ 1/\hat{\theta}_{est,2} \end{pmatrix} \tag{5.19}$$

The online parameter estimation of the closed-loop system (5.6) can be accomplished by using (5.3), (5.4) together with (5.13), (5.14), (5.16) and (5.17). We can then obtain the estimated parameters by using the equations (5.18) and (5.19). Numerical integrators can be used to implement the equation (5.3).

5.4 Weight Adaptation Law

The method described in this subsection is based on the work (Pierson et al., 2015). Here, we assume that $\widehat{K}_i$ is calculated by using (5.18) and (5.19) after the estimation. So, the vector $\widehat{K}_i$ can be used in the weight estimator. Then, the GPD workspace partitioning algorithm takes the calculated weights as input.

The adaptation law is written for the weight w_i of the agent *i* as below:

$$\delta_i = w_i - f(\widehat{K}_i) \tag{5.20}$$

$$\dot{w}_i = -k_\omega \sum\nolimits_{j \in N_i} (\delta_i - \delta_j) \tag{5.21}$$

where k_ω denotes a positive parameter and N_i defined the neighbor set of the corresponding agent. The performance function $f(\widehat{K}_i)$ is gives the relationship between the desired performance and the estimation vector and can be selected as $f(\widehat{K}_i) = \|\widehat{K}_i\|$.

5.5 Stability Analysis

In this subsection, the stability proof of the control and estimation laws is given. First, we start defining a Lyapunov function candidate as given below for $V_3 \colon \theta_{est} + (-c, c)^M \to \mathbb{R}$:

$$V = V_1 + V_2 + V_3 \tag{5.22}$$

$$V_1 = \sum\nolimits_i \frac{1}{2} \left\| C_{\mathrm{V}_i} - p_i \right\|^2 \tag{5.23}$$

$$V_2 = \sum\nolimits_i w_i \tag{5.24}$$

$$V_3 = \sum\nolimits_i -\frac{1}{2c} \sum\nolimits_{j=1}^{M} ln \left(\frac{\left(1 + \frac{\tilde{\theta}_{est(i,j)}}{c - \theta_{est(i,j)}}\right)^{c - \theta_{est(i,j)}}}{\left(1 - \frac{\tilde{\theta}_{est(i,j)}}{c + \theta_{est(i,j)}}\right)^{c + \theta_{est(i,j)}}} \right) \tag{5.25}$$

where $\tilde{\theta}_{est(i,j)} = \theta_{est(i,j)} - \hat{\theta}_{est(i,j)}$ and $\hat{\theta}_{est(i,j)}$ denotes the output of the j^{th} neuron of the parameter estimator of the i^{th} agent. Also, c represents a positive coefficient (Alonso et al., 2009).

If the state vector $x_i = \begin{pmatrix} C_{\mathrm{V}_i} - p_i & w_i & \tilde{\theta}_{est(i)} \end{pmatrix}^T$ is considered, it can be shown that $V(x = 0) = 0$. In addition, $V_3(x) > 0$ is shown in the proof in (Alonso et al., 2009). So, we can conclude that $V(x) > 0$.

The first time derivative of the Lyapunov function is obtained and the negative definiteness of its parts is investigated as below.

$$\dot{V}_1 = \sum\nolimits_i -\left(C_{\mathrm{V}_i} - p_i\right)^T \dot{p}_i \tag{5.26}$$

$$\dot{V}_1 = \sum\nolimits_i -\left(C_{\mathrm{V}_i} - p_i\right)^T K_p \left(C_{\mathrm{V}_i} - p_i\right) \leq 0 \tag{5.27}$$

The negative semi-definite result is obtained for the first term.

$$\dot{V}_2 = \sum_i -k_\omega \sum_{j\in N_i} \begin{pmatrix} \left(w_i - f(\widehat{K}_i)\right) - \\ \left(w_j - f(\widehat{K}_j)\right) \end{pmatrix} = 0 \tag{5.28}$$

Moreover, the second term is zero. The derivative of the third term yields:

$$\dot{V}_3 = \sum_i -\frac{1}{c\,\beta} \tilde{\theta}_{est(i)}^T\, W_i\, \tilde{\theta}_{est(i)} \leq 0 \tag{5.29}$$

where W_i represents weight matrix for i^{th} agent and is clearly positive semi-definite. Thus, for $\beta > 0, c > 0$, the third term gives a negative semi-definite result.

Because the $\dot{V} \leq 0$, the trajectories and estimation errors remain bounded. The weights are also shown to be bounded because the $\sum_i \dot{w}_i$ can be converted into the stable filter form as shown in (Pierson et al., 2015). The following theorem is introduced from (Alonso et al., 2009) to complete the proof:

Theorem 5.1: The equilibrium point $\tilde{\theta}_{est(i)}^* = 0$ is globally asymptotically stable if $I \subset [t_0, \infty)$ and $\bigcap_{t\in I} \ker\left(A(t)\right) = \{0\}$.

From the *Theorem 5.1* and the proof in (Alonso et al., 2009), the conclusion is that for a non-degenerate interval $t \in I$, the equilibrium point $\tilde{\theta}_{est(i)}^* = 0$ is globally uniformly asymptotically stable and the equilibrium point is unique.

From LaSalle's Invariance Principle, we should find the largest invariant set defined by $\dot{V} = 0$. $\dot{V} = 0$ occurs only when $C_{V_i} = p_i$ and $\tilde{\theta}_{est(i)} = 0$. This gives the case that the tracking and estimation errors are zero. We can conclude that the set is an invariant set by using the control and estimation laws. So, the system is shown to be globally asymptotically stable.

Corollary 5.1: In the steady state, the estimation vector converges to its real value (Pierson et al., 2015):

$$\lim_{t\to\infty}\left(\widehat{K}_i\right) = K_i \tag{5.30}$$

$$\lim_{t\to\infty}\left(w_i - w_j\right) = f(K_i) - f\left(K_j\right) \tag{5.31}$$

From the stable filter theory, in the steady state, the weights converge to constant values. Also, from (5.31) the value of $w_i - f(K_i)$ converges to a common value for all agents.

5.6 Agent Disorder

The failure of the agent i which corresponds to the case of $\widehat{K}_i = 0$ should be seperately considered. The reason is that from the equation (5.19) we can see that the parameter vector $\hat{\theta}_{est}$ should take infinite value in order the $\widehat{K}_i$ to be zero. So, we present the following algorithm:

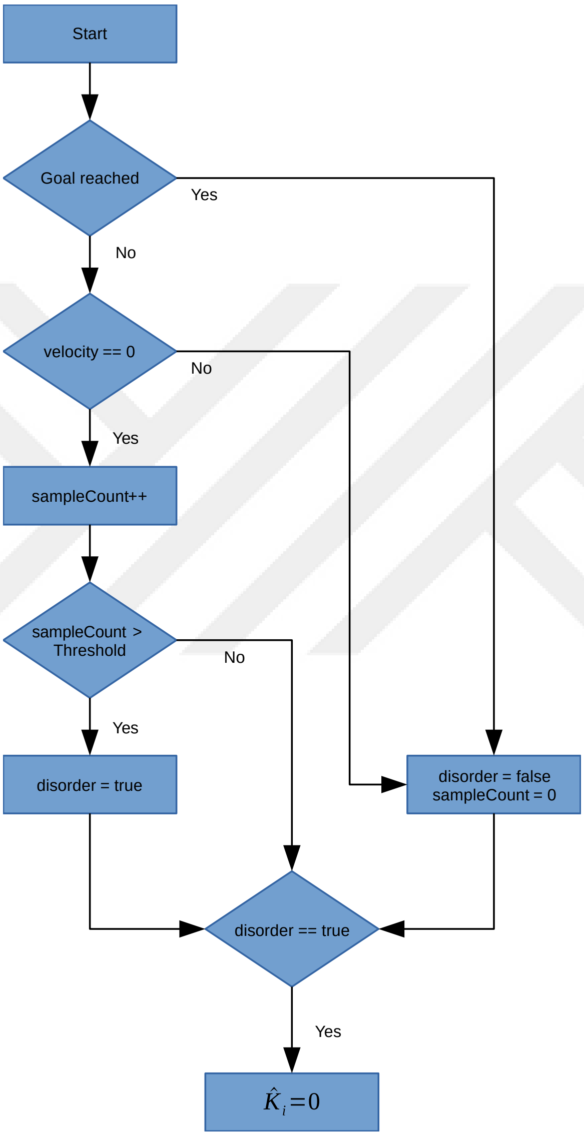

Figure 5.1 : Agent disorder algorithm.

The algorithm in Figure 5.1 detects the immobilized agent and sets the parameter vector of the corresponding agent to zero to ensure that the immobilized agent can not take any region from the workspace.

Since the corresponding estimation vector is set to zero, it does not affect the overall stability as given in (5.28).

5.7 Discussion

In this section, an Hopfield network based coverage collaboration algorithm is presented which includes guaranteed power voronoi workspace partitioning. The adaptation to performance variations is achieved by utilization of GPVDs, HNNs, weight estimator and a point-offset control law. Also, an agent disorder algorithm is proposed which handles the immobilized agent case. The Lyapunov-type stability proof shows that the system consisting of control and adaptation laws is globally asymptotically stable.

6. ENERGY-EFFICIENT COVERAGE CONTROL WITH HOPFIELD NETWORKS

A new multi-agent coverage control approach is presented. The agents deploy themselves autonomously in the map by doing collaboration among themselves. They achieve this goal by learning their own actuation performances. Meanwhile, they optimize the energy-efficient optimal coverage cost function which makes a trade-off between the coverage time and the energy consumption.

6.1 Problem Formulation

In the problem, *m* non-holonomic mobile robots are considered. The goal is to allocate the regions from the environment to the agents according to their different actuation performances and at the same time to move the agents the optimal coverage configuration. The workspace allocation is performed by estimating the performance parameters online by utilizing HNNs and then calculating the PD weights of each agent. The PD weights together with the agent locations are used to draw the Power Diagram within the algorithm to find the centroid locations. Then, the agents are driven to the calculated centroid locations. The algorithm iteratively converges to the optimal configuration. The whole algorithm runs online and in a decentralized manner for each agent. The communication is performed among the agents.

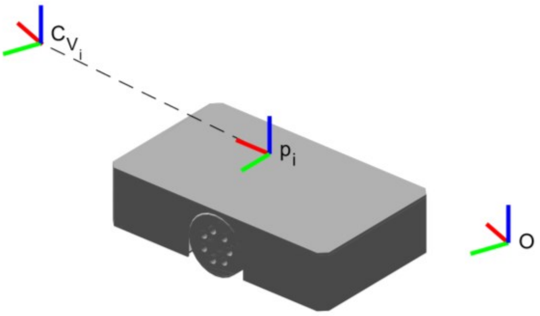

Figure 6.1 : The robot position $\mathbf{p_i}$ and centroid location $\mathbf{C_{V_i}}$.

In the next sections, the definitions of the Power Voronoi Diagrams, Energy-Efficient Coverage Optimal Control and Hopfield Neural Networks will be given.

6.2 Energy-Efficient Coverage Optimal Control

The energy-efficient coverage optimal control problem can be defined by giving the the cost function and control law (Di Perna & Rodrigues, 2017) as in the following sections.

$$J = \int_0^{\infty} \sum_{i=1}^{m} \left(s_i \left\| \int_{W_i} (p_i - q)\phi(q)dq \right\|^2 + r_i {u_i}^T u_i \right) d\tau \qquad (6.32)$$

Here, m is the number of the agents, s_i and r_i give the weighting coefficients for coverage and energy terms, respectively. Also, p_i represents the location of the i^th^ agent.

The mass M_{W_i} and centroid C_{W_i} of the Voronoi region *i* are defined as follows (Luna, J. M., Fierro, R., Abdallah, C. T. and Wood, 2013):

$$\mathrm{F}_i(h, q) = \int_{\mathrm{W}_i} h\, dq \qquad (6.33)$$

$$C_{\mathrm{W}_i} = \frac{1}{M_{W_i}} \mathrm{F}_i(q\, \phi(q), q) \qquad (6.34)$$

$$M_{W_i} = \mathrm{F}_i(\phi(q), q) \qquad (6.35)$$

$$\dot{p}_i = u_i \qquad (6.36)$$

$$u_i = \sqrt{\frac{s_i}{r_i}} M_{W_i}\left(C_{W_i} - p_i\right) \qquad (6.37)$$

The system model is shown in equation (6.36). The equation (6.37) represents the control law which minimizes the cost function in (6.32) for the agent *i*.

6.3 Multi-Agent Coverage Control with Different Actuation Capabilities

In this section, the parameter estimation with HNNs and estimation of the PD weights will be shown with the necessary definitions.

For this purpose, the definition of the actuation performances is given. The unequal capabilities of the actuators of the individual agents are considered as the different actuation performances. The examples can be given for strong actuation performances as strong motors, favorable terrain and good tire contacts. Weak motors, slippery terrain, wheel slip can be considered as weak actuation performances.

In this scope, in order to drive the agents to the centroid locations, the point offset control law (Michael & Kumar, 2009; Pierson & Schwager, 2015) is utilized in conjunction with the energy-efficient coverage optimal control law. The velocity transformation used for conversion of the velocities in the world frame coming from the energy-efficient coverage optimal control law to the linear v and angular velocities ω are given as below where θ represents the yaw angle of the non-holonomic robot:

$$\begin{pmatrix} v \\ \omega \end{pmatrix} = \begin{pmatrix} cos\theta & sin\theta \\ -\frac{sin\theta}{l} & \frac{cos\theta}{l} \end{pmatrix} \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} \tag{6.38}$$

Thus, the control inputs coming from the holonomic control law can be transformed to the non-holonomic control inputs by using the distance l which is defined as the distance from the reference point P to the geometric center of the non-holonomic robot.

The collision check for the agent is performed by checking the center P with a radius $\rho = l + r_{robot}$, where r_{robot} is the radius of the agent.

6.4 Estimating the Performance Parameters with HNNs

The system in (6.36) is utilized together with the control law in (6.37) to estimate the performance parameters of the agent i. For this pupose, the closed-loop system will be rewritten in linear in parameters (LIP) form. The closed-loop system equation can be obtained by using the control law as below:

$$\dot{p}_i = \sqrt{\frac{s_i}{r_i}} M_{W_i}\left(C_{W_i} - p_i\right) = K_i M_{W_i}\left(C_{W_i} - p_i\right) \tag{6.39}$$

Here, the velocity vector $\dot{p}_i$ is transformed into the body coordinates of the agents by using the conversion below where $R_{z,\theta}$ is the rotation matrix about z and $\dot{p}_{i,b}$ is the body velocities of the agent i:

$$\dot{p}_{i,b} = \begin{bmatrix} \dot{x}_{i,b} \\ \dot{y}_{i,b} \end{bmatrix} = R_{z,\theta}^{T} \sqrt{\frac{s_i}{r_i}} M_{W_i} \left(C_{W_i} - p_i \right) \tag{6.40}$$

Since the lateral body velocity $\dot{y}_{i,b}$ is equal to zero, the second row can be omitted:

$$-\dot{x}_{i,b} \sqrt{\frac{r_i}{s_i}} = [cos\theta \quad sin\theta] \begin{bmatrix} p_{i,x} - C_{W_{i,x}} \\ p_{i,y} - C_{W_{i,y}} \end{bmatrix} M_{W_i} \tag{6.41}$$

By using the first row, the LIP form can be given as below:

$$y = [cos\theta \quad sin\theta] \begin{bmatrix} p_{i,x} - C_{W_{i,x}} \\ p_{i,y} - C_{W_{i,y}} \end{bmatrix} M_{W_i} \tag{6.42}$$

$$A = -\dot{x}_{i,b} \tag{6.43}$$

$$\theta_{est} = \sqrt{\frac{r_i}{s_i}} \tag{6.44}$$

$$y = A\, \theta_{est} \tag{6.45}$$

The weight matrix and bias terms of the HNN can be calculated as below:

$$W = A^{T} A \tag{6.46}$$

$$I = -A^{T} y \tag{6.47}$$

Then, the parameter vector $\widehat{K}_i$ to be estimated is calculated by utilizing the estimated parameter $\widehat{\theta}_{est}$ as shown below:

$$\widehat{\theta}_{est} = s(t) \tag{6.48}$$

$$\widehat{K}_i = {}^{1}\!/_{\widehat{\theta}_{est}} \tag{6.49}$$

By utilizing (5.3), (5.4) with (5.13), (5.14), (5.16) and (5.17) the parameters of the closed-loop system (6.39) is calculated online by using HNN estimator. Then, by using the equations (6.48) and (6.49), the resultant parameters are obtained.

The implementation of the estimator is accomplished by using numerical integrators.

6.5 Estimating the PD Weights of the Agents

The weight estimator algorithm in this section is based on the algorithm in work (Pierson, Figueiredo, Pimenta, & Schwager, 2017). After calculating $\widehat{K}_i$ by using (6.48) and (6.49), the parameter vector $\widehat{K}_i$ is obtained and passed into the weight estimator. Then, the output of the weight estimator given in this section is passed to the PD algorithm in order to adjust the regions of the individual agents.

The estimation law for the weight w_i of the agent *i* is presented as below:

$$\kappa_i = w_i - f_{perf}(\widehat{K}_i) \tag{6.50}$$

$$\dot{w}_i = -k_\omega \sum\nolimits_{j \in N_i} (\kappa_i - \kappa_j) \tag{6.51}$$

where k_ω gives a positive coefficient and N_i denotes the neighbors of the agent i. The function $f_{perf}(\widehat{K}_i)$ corresponds to the desired performance function which is selected as $f_{perf}(\widehat{K}_i) = \|\widehat{K}_i\|$ in the simulations and experimental implementations.

After the estimation of the $\widehat{K}_i$ for the agent i, the PD weights are calculated by utilizing the estimation law in (6.50) and (6.51). For the implementation, the resultant estimation vector is calculated by using numerical integrators.

6.6 Stability Analysis

This section presents the Lyapunov-type stability proof for the control and estimation laws. First, a Lyapunov function candidate should be selected as given below for V_3: $\theta_{est} + (-c, c)^M \rightarrow \mathbb{R}$:

$$V = \sum_{i=1}^{3} V_i \tag{6.52}$$

$$V_1 = \sum\nolimits_i \frac{1}{2} \left\| C_{\mathrm{W}_i} - p_i \right\|^2 \tag{6.53}$$

$$V_2 = \sum_i w_i \tag{6.54}$$

$$f_1 = \left(1 + \frac{\widetilde{\theta}_{est(i,j)}}{c - \theta_{est(i,j)}}\right)^{c - \theta_{est(i,j)}} \tag{6.55}$$

$$f_2 = \left(1 - \frac{\widetilde{\theta}_{est(i,j)}}{c + \theta_{est(i,j)}}\right)^{c + \theta_{est(i,j)}} \tag{6.56}$$

$$V_3 = \sum_i -\frac{1}{2c} \sum_{j=1}^{M} ln(f_1\, f_2) \tag{6.57}$$

where $\widetilde{\theta}_{est(i,j)} = \theta_{est(i,j)} - \widehat{\theta}_{est(i,j)}$. $\widehat{\theta}_{est(i,j)}$ denotes the output of the *j*th neuron of the parameter estimator of the *i*th agent. Here, *c* is a positive constant (Alonso et al., 2009).

Next, the state vector $x_i = \left(C_{\mathrm{W}_i} - p_i \quad w_i \quad \widetilde{\theta}_{est(i)}\right)^T$ is defined. It can be shown that $V(x = 0) = 0$. The positive definiteness of the $V_3(x)$ is clear as shown in proof in (Alonso et al., 2009). So, the conclusion is that the Lyapunov function candidate is positive definite, $V(x) > 0$.

Then, the time derivative of the Lyapunov function should be investigated. The first time derivative of the Lyapunov function is as given below:

$$\dot{V}_1 = \sum_i -\left(C_{\mathrm{W}_i} - p_i\right)^T \sqrt{\frac{s_i}{r_i}} M_{W_i}\left(C_{\mathrm{W}_i} - p_i\right) \leq 0 \tag{6.58}$$

Here, $\dot{V}_1$ is negative semi-definite since $\sqrt{\frac{s_i}{r_i}}$ and M_{W_i} are positive-definite.

$$\dot{V}_2 = \sum_i -k_\omega \sum_{j \in N_i} \left(\left(w_i - f_{perf}(\widehat{K}_i)\right) - \left(w_j - f_{perf}(\widehat{K}_j)\right)\right) = 0 \tag{6.59}$$

Also, the second term is proven to be zero. The derivative of the third term gives:

$$\dot{V}_3 = \sum_i -\frac{1}{c\,\beta} \tilde{\theta}^T_{est(i)}\, W_i\, \tilde{\theta}_{est(i)} \leq 0 \tag{6.60}$$

where W_i is the weight matrix for *i*th agent and clearly positive semi-definite. For $\beta > 0, c > 0$, the third term $\dot{V}_3$ is shown to be negative semi-definite.

Since the $\dot{V} \leq 0$, the trajectories and estimation errors are bounded. Also, The weights are shown to be bounded as given the $\sum_i \dot{w}_i$ in the proof in (Pierson et al., 2015).

The following theorem should be given to guarantee that the equilibrium point $\widetilde{\theta}^{*}_{est(i)} = 0$ is globally asymptotically stable (Alonso et al., 2009):

Theorem 6.1: The equilibrium point $\widetilde{\theta}^{*}_{est(i)} = 0$ is globally asymptotically stable if $I \subset [t_0, \infty)$ and $\cap_{t \in I} \ker(A(t)) = \{0\}$.

From the *Theorem 6.1* and the Lemma 3 from (Alonso et al., 2009), for a non-degenerate interval $t \in I$, the equilibrium point $\widetilde{\theta}^{*}_{est(i)} = 0$ is globally uniformly asymptotically stable and is unique.

To show that the system is globally asymptotically stable, the LaSalle's Invariance Principle is used. The largest invariant set defined by $\dot{V} = 0$ should be found. It can be shown that $\dot{V} = 0$ only if $C_{\mathrm{W}_i} = p_i$ and $\widetilde{\theta}_{est(i)} = 0$. Since the equilibrium points are unique, the result is that the tracking and estimation errors become zero. From the control and estimation laws, the conclusion is that the set is an invariant set. Thus, the system is globally asymptotically stable.

From *Corollary 5.1*, in the steady state, the weights converge to their final values, as stated in the reference paper. From (5.31), the value of $w_i - f_{perf}(K_i)$ achieves a common value for all agents in the steady state.

6.7 Discussion

In this section, an Hopfield network based energy-efficient coverage collaboration algorithm is presented. The algorithm provides a trade-off between energy consumption and coverage time, as well as, workspace allocation according to the actuation performances of the agents by means of power diagrams. According to the Lyapunov-type stability proof, the control and estimation laws are proven to be globally asymptotically stable.

7. SIMULATION RESULTS

In this section, the simulation results of the proposed methods are explained.

7.1 Adaptive Coverage Control with Imprecise Localization

Simulations were done in MATLAB with 5, 10 and 15 agents in 5 by 5 meters environment. The parameters utilized in simulations are $\gamma = 0.3, \lambda = 0.4,\ \xi = 3,\ \eta = 2$ and $\alpha = 0.01$.

An expanded circle was selected for the distributed density function $\phi(q)$ which was triggered at predefined times in the simulation. The estimated density function $\hat{\phi}_k(q)$ consists of 64 cells. We can give the vector function Z as below:

$$\mathrm{Z}_k(q) = e^{\frac{-(q-\mu_k)^2}{2{\sigma_k}^2}} > 0 \tag{7.1}$$

where ${\sigma_k}^2 = 0.05$. Z_k represents k^{th} term of the vector function and μ_k denotes the mean of the Gaussian density function.

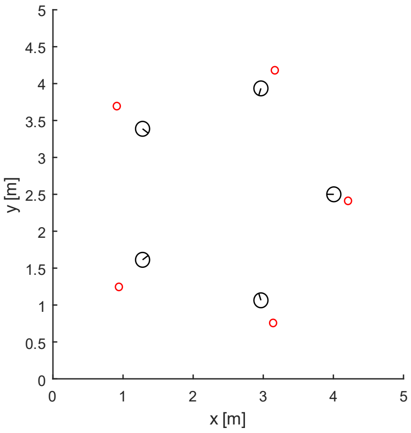

Figure 7.1 : Robot poses in a simulation frame (red circles show the centroid positions).

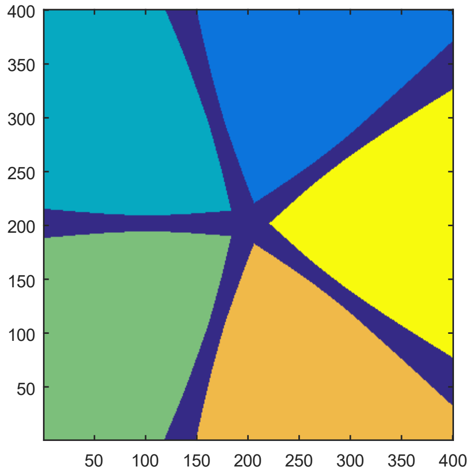

Figure 7.2 : A simulation frame with GPVD.

The robot locations and GPVD in an example simulation frame are shown in Figure 7.1 and Figure 7.2, respectively.

In the following subsections, the stability results of the algorithm and the results of the proposed algorithm with GPVD are given.

7.1.1 Stability results

The simulation was done with 5 agents. The trigger times were 0, 60 and 120 seconds. The sum of absolute values of the agent positions and the angular errors are shown in Figure 7.3. As given in the figures, the errors take large values in predefined trigger times and then decrease to zero asymptotically.

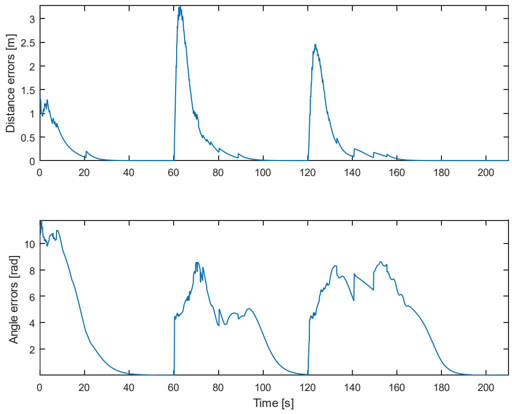

Figure 7.3 : Distance and Angle Errors of 5 Agents.

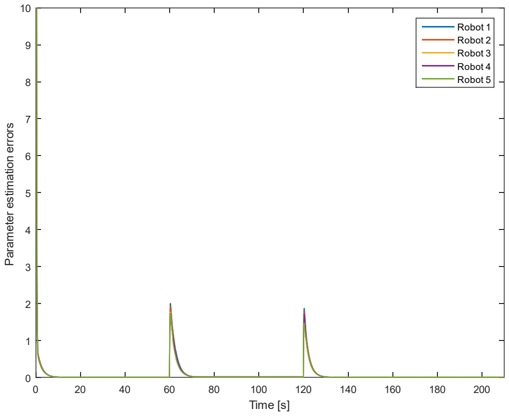

Figure 7.4 : Parameter Estimation Errors.

In Figure 7.4, with the start of the simulation and with the trigger times, the parameter estimation errors increased and then converged to zero asymptotically.

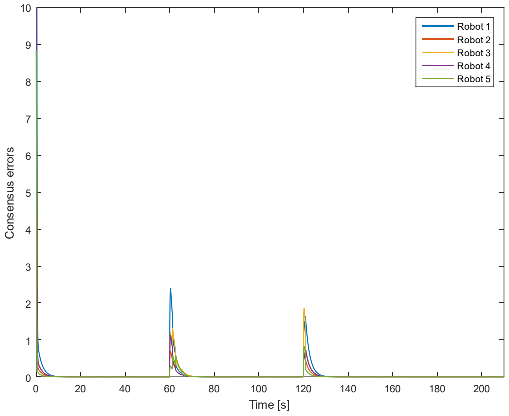

Figure 7.5 : Consensus Errors.

In Figure 7.5, the consensus errors of the agents were driven to zero by the linear consensus algorithm. The errors increased at the trigger times and then converged to zero asymptotically.

7.1.2 Results of the coordination algorithm with GPVD

In this section, the results regarding the proposed coordination algorithm with the GPVD are shown. The information gain function $\phi(q)$ was selected as $\phi(q) = 1$. We took the positioning uncertainty as 0.1 meters for each agent. Positive values were assigned to the weights of the regions in GPVD so that as in the power diagram, the agents with positive weights took larger portions from the environment. Also, the definition of the region ratio is the ratio of the region of the corresponding agent with respect to the total region of the workspace at the end configurations. In simulations, the Voronoi partitioning was accomplished by the GPVD in the coordination algorithm.

Table 7.1 : Region ratios of the agents at the end configuration (with 5 agents).

Agent	Region Ratio ($\omega_1 = 0$)	Region Ratio ($\omega_1 = 0.5$)
1	0.1641	0.2281
2	0.2130	0.1816
3	0.2120	0.2083
4	0.1421	0.1537
5	0.1708	0.1176

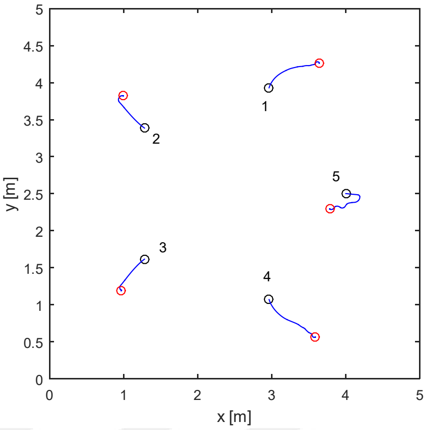

Figure 7.6 : The trajectories of the 5 agents for the case $\omega_1 = 0$ (red circles are the end positions, black circles give the start locations).

In Table 7.1, the results with the 5 robots are given. The weight of the robot 1 was changed between $\omega_1 = 0$ and $\omega_1 = 0.5$, in two simulations. The weights of the other agents were kept at zero. If we compare the two cases, it can be shown that the robot 1 took a greater region with $\omega_1 = 0.5$ than the case with $\omega_1 = 0$. Also, in the second case, the first agent took greater area than the other agents.

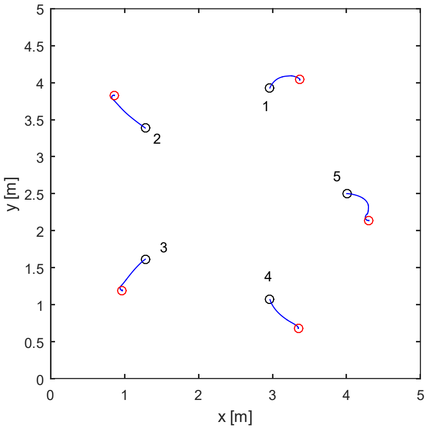

Figure 7.7 : The trajectories of 5 agents with $\omega_1 = 0.5$.

The trajectories of 5 agents are given in Figure 7.6 and Figure 7.7.

Table 7.2 : Region ratios of the agents at the end configuration (with 10 agents).

Agent	Region Ratio ($\omega_1 = 0$)	Region Ratio ($\omega_1 = 0.5$)
1	0.0800	0.1469
2	0.0851	0.0728
3	0.0801	0.1082
4	0.0587	0.0895
5	0.1226	0.0878
6	0.0536	0.0425
7	0.0731	0.0584
8	0.0786	0.0868
9	0.0730	0.0614
10	0.1303	0.0916

In Table 7.2, the results with the 10 robots are shown. The weight of the robot 1 was taken as $\omega_1 = 0$ and $\omega_1 = 0.5$ in two simulations, respectively. A similar result to the previous one was obtained. The first agent took a greater area than the other robots in the second case.

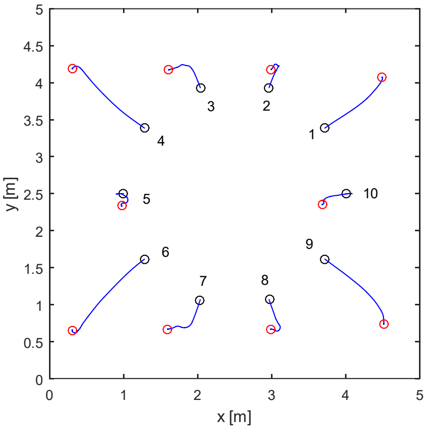

Figure 7.8 : The trajectories of 10 agents with $\omega_1 = 0$.

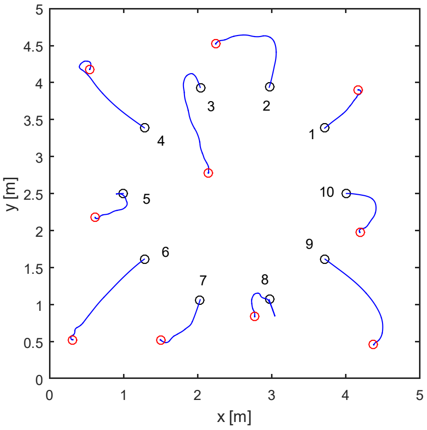

Figure 7.9 : The trajectories of 10 agents with $\omega_1 = 0.5$.

The trajectories of the 10 agents are shown in Figure 7.8 and Figure 7.9.

In Table 7.3, the results with the 15 robots are given. The weight of the robot 1 was chosen as $\omega_1 = 0$ and $\omega_1 = 0.5$, respectively. A similar result was obtained as in the previous cases with 5 and 10 agents. The area of the first agent with $\omega_1 = 0.5$ was greater than in the case with $\omega_1 = 0$.

Table 7.3 : Region ratios of the agents at the end configuration (with 15 agents).

Agent	Region Ratio ($\omega_1 = 0$)	Region Ratio ($\omega_1 = 0.5$)
1	0.0689	0.1082
2	0.0604	0.0713
3	0.0587	0.0347
4	0.0602	0.0591
5	0.0373	0.0467
6	0.0410	0.0325
7	0.0575	0.0588
8	0.0502	0.0604
9	0.0279	0.0338
10	0.0438	0.0368
11	0.0727	0.0563
12	0.0546	0.0535
13	0.0462	0.0380
14	0.0601	0.0485
15	0.0550	0.0557

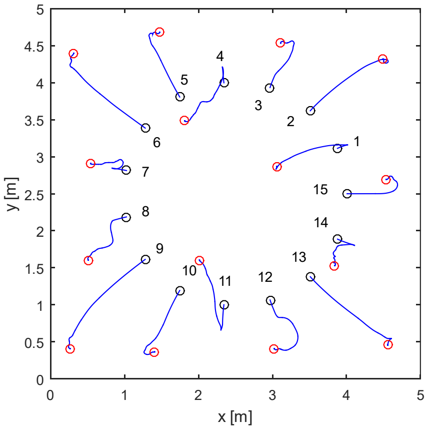

Figure 7.10 : The trajectories of 15 agents with $\omega_1 = 0$.

In Figure 7.10 and Figure 7.11, the trajectories of 15 agents are shown.

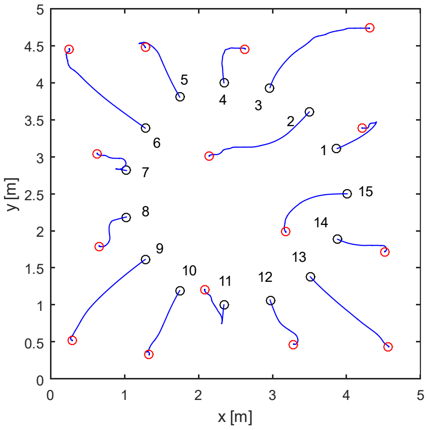

Figure 7.11 : The trajectories of 15 agents with $\omega_1 = 0.5$.

The move-to-centroid control laws drives the agents to the end configurations. A coordinated motion is provided by using the dynamic Voronoi partitioning. At the end configuration, the optimal coverage was achieved. The areas in the GPVD change with respect to the weights assigned to the mobile agents.

7.2 Adaptation with non-holonomic estimation model

In this section, the MATLAB simulation results are shown with five robots. The size of the map were chosen as 5 by 5 meters. The parameters in the simulation were selected as $k_\omega = 25$, $\gamma = 0.075$, $\lambda = 0.05$ and the radius of the uncertainty circle of the robot i was given as $r_i = 0.1m$.

The rate of convergence of the weight estimator can be adjusted by changing the parameter k_ω in (5.21). The controller parameters are represented by γ and λ as shown in (2.18). The distance and angular error dynamics between the position of the agent and the centroid were changed by using the controller coefficients. The standard deviation of the localization error for each agent is determined by the uncertainty circle radius r_i. The location of the agent is not known perfectly but the position is within an uncertainty circle.

In the stability results section, the verification results of the overall system stability are given. Also, in the next sections, two case studies will be discussed.

7.2.1 Stability results

Five agents were used in the simulation performed. In the simulations the distributed density function was chosen as $\phi(q) = 1$. The self-deployment of the non-holonomic agents is accomplished by using the density function. The online estimator algorithm calculates the weights of GPVD assigned to the agents by learning the actuation performances online from the motion of the agents. The portions from the environment given to the agents with stronger actuators are larger than the regions assigned to the robots with weaker actuators.

The actuation performance of the first robot is degraded by 10 percent while the performances of the other agents remain unchanged.

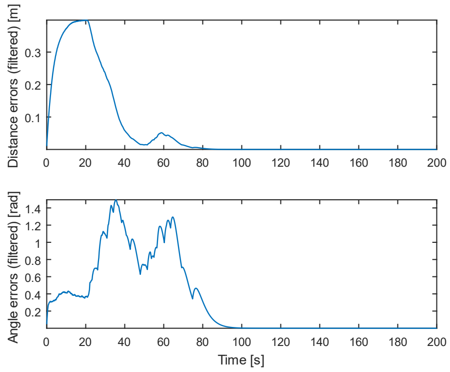

Figure 7.12 : Position and Angle Errors.

As it turns out from Figure 7.12, the position and angle errors converge to zero asymptotically, as expected from the stability proof. The coverage cost function in (2.7) reaches a minimum value as the time goes to infinity as depicted in Figure 7.13 showing that the optimal coverage task is performed and complete. Because the weight w_i in (2.7) is positive the coverage cost function settles to a negative value.

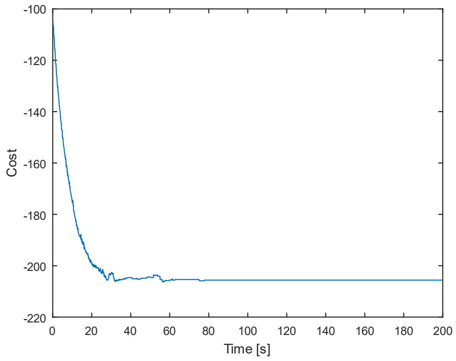

Figure 7.13 : Coverage Cost Function.

Table 7.4 : Region ratios of the agents at the end configuration (stability results).

Agent	Region Ratio
1	0.0970
2	0.2303
3	0.2150
4	0.1540
5	0.1976

Figure 7.14 illustrates the change of the parameter estimation errors of the robots with respect to time. The errors asymptotically converge to zero as explained in the stability proof. Also, the initial value of the error of the first agent is a non-zero value since its actuation performance is weaker than the other agents and the initial parameter estimation vectors are chosen as the nominal values. As shown in Figure 7.15, the

value of the function $w_i - f(\widehat{K}_i)$ goes to a common value among the agents as the time goes to infinity.

The weight of the first robot converges to a smaller value than the weights of the other agents as shown in Figure 7.16 since the actuation performance of the agent 1 was chosen worse than the other robots.

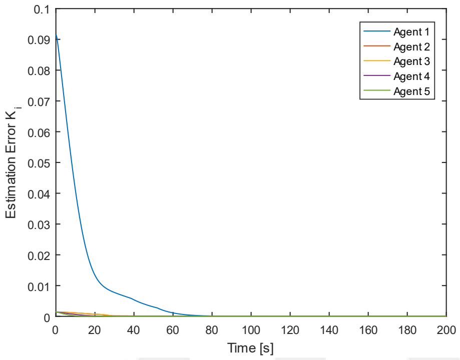

Figure 7.14 : Parameter Estimation Error.

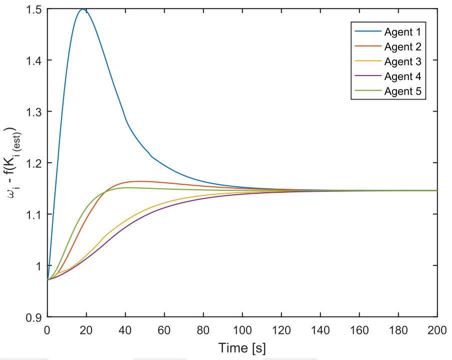

Figure 7.15 : The value of $\boldsymbol{w_i - f(\widehat{K}_i)}$.

As presented in the Table 7.4, the algorithm assigns larger regions to the robots having better actuators while it gives smaller regions to the agents that have worse actuators. As a result, less region ratio for the first robot was achieved comparing to the other robots.

7.2.2 Case study 1

In the case study, the initial positions of the robots were set in a circle with random deviations and yaw angles. They were released from a ground vehicle and performed a given coverage task in a collaborative manner.

The actuation performance of the first robot was set to a value that is 10 percent worse than the other robots.

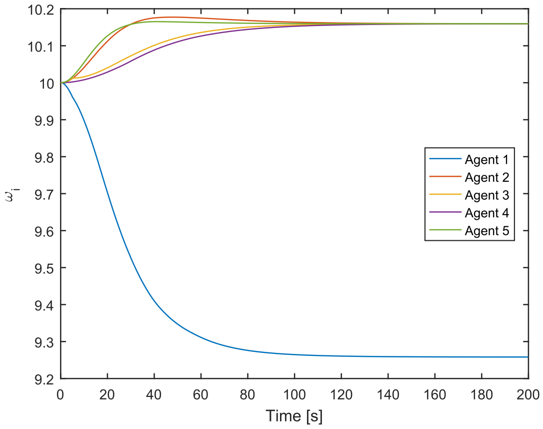

Figure 7.16 : The weights $\boldsymbol{w_i}$ of the agents.

In Figure 7.17, the value of the coverage cost function is illustrated. As the time progresses, the coverage cost settles to a minimum value showing that the robots reached the optimal positions.

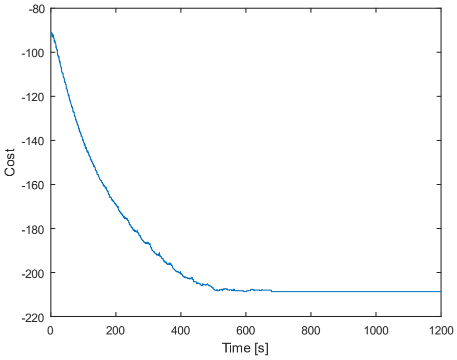

Figure 7.17 : Coverage Cost Function (Case Study 1).

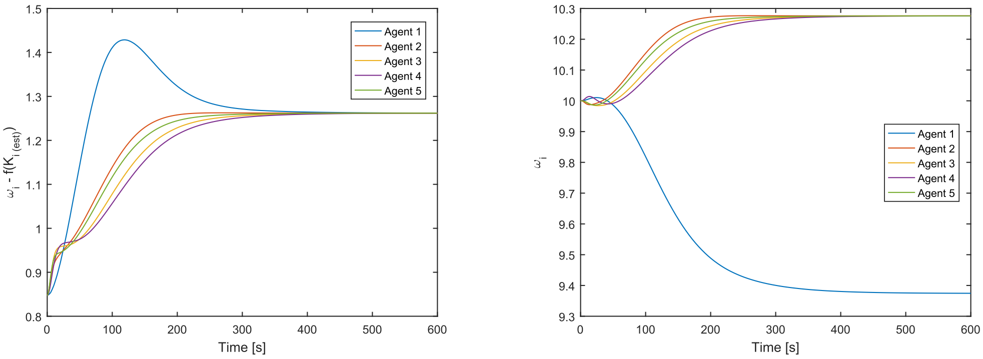

Figure 7.18 : The value of $\boldsymbol{w_i} - \boldsymbol{f}(\widehat{\boldsymbol{K}}_{\boldsymbol{i}})$.

Figure 7.19 : The weights $\boldsymbol{w_i}$ of the agents.

Figure 7.18 depicts the value of the function $\boldsymbol{w_i} - \boldsymbol{f}(\widehat{\boldsymbol{K}}_{\boldsymbol{i}})$. The function settles to a common value among the agents. Also, in Figure 7.19 the weights of the agents are illustrated where the robot 1 having weaker actuators has a smaller weight estimate value than the other robots.

Table 7.5 : Region ratios of the agents at the end configuration (case study 1).

Agent	Region Ratio
1	0.0860
2	0.2277
3	0.2232
4	0.1633
5	0.1936

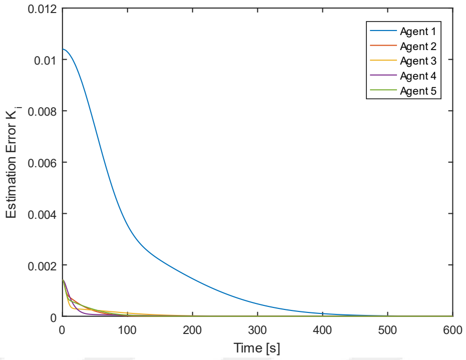

Figure 7.20 : Parameter estimation error.

According to the Table 7.5, the agents with better actuators took larger portions from the workspace while the robot 1 having worse actuators took smaller regions.

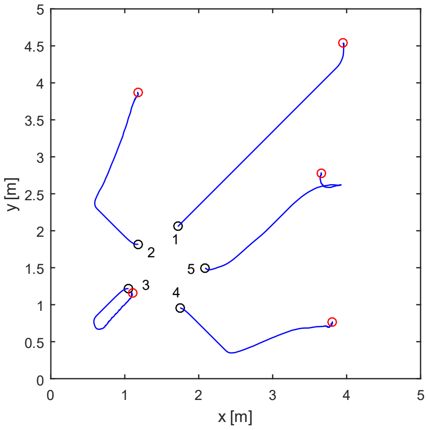

Figure 7.21 : The trajectories of the agents for case study 1 (black circles: start configuration, red circles: end configuration).

The evaluation of the simulation results shows that the change in the actuator performances results in different amount of areas assigned from the workspace by the algorithm to the robots according to their performances. The trajectories of the agents are illustrated in Figure 7.21.

7.2.3 Case study 2

In the case study, the initial configurations of the agents were given along a line with random deviations and yaw angles. An aerial vehicle released the agents and they achieved the optimal coverage in a collaborative way. The GPV-regions were adjusted by learning the different performances capabilities of the agents online and assigning the calculated weights to the agents.

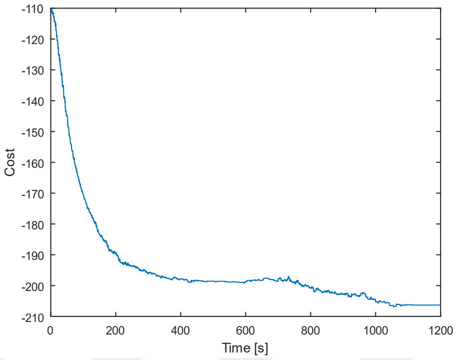

Figure 7.22 : Coverage Cost Function (Case Study 2).

The actuation performance of the first agent was set to a 10 percent lower value than the performances of the other agents.

Figure 7.22 depicts the coverage cost function value as the time goes to infinity. The function reaches an optimal value which satisfies the optimal coverage objective at the end of the simulation.

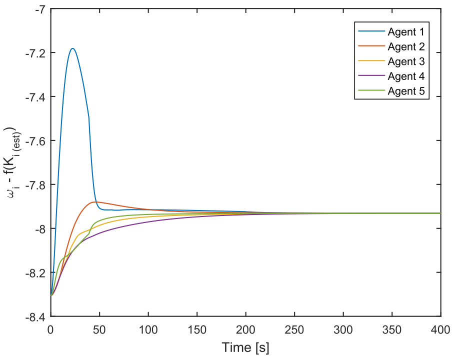

Figure 7.23 : The value of $\boldsymbol{w_i - f(\hat{K}_i)}$.

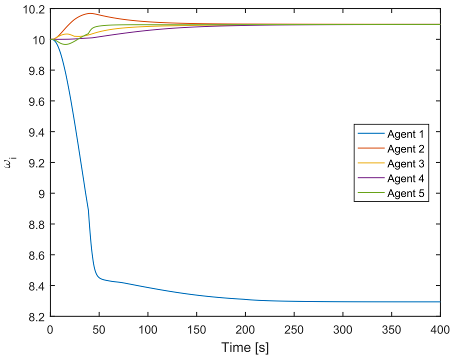

Figure 7.24 : The weights $\boldsymbol{w_i}$ of the agents.

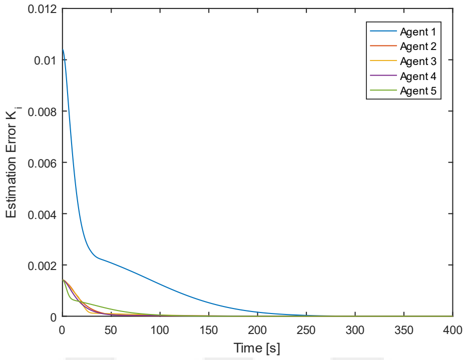

Figure 7.25 : Parameter estimation error.

In Figure 7.23, the value of the function $w_i - f(\hat{K}_i)$ settles to a common value across all agents. Moreover, the evaluation of the Figure 7.24 gives the result that the weights of other agents were larger than the weight of the first agent since the performance of the first agent was worse than the others.

As shown in the Table 7.6, larger regions were assigned to the agents having better actuators while the agent 1 took smaller region from the workspace since it has actuators with worse performances than the other robots.

Table 7.6 : Region ratios of the agents at the end configuration (case study 2).

Agent	Region Ratio
1	0.0065
2	0.2355
3	0.2170
4	0.2283
5	0.2173

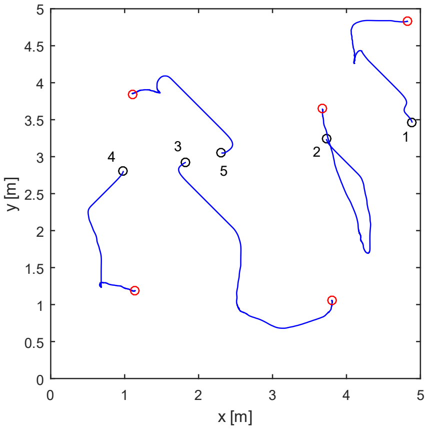

Figure 7.26 : The trajectories of the agents for the case study 2 (black circles: start configuration, red circles: end configuration).

According to the simulation results, it is concluded that the agents take greater or smaller regions from the environment relative to each other according to their different actuation performance capabilities. The trajectories of the agents are illustrated in Figure 7.26.

7.3 Adaptation with Hopfield Neural Networks

The simulations were done in MATLAB environment with the map dimensions of 5x5 and 10x10 meters. The parameters for the simulation were chosen as $k_\omega = 0.2$, $K_p = diag([1 \quad 1])$, $r_{robot} = 0.11$, $l = 0.1$, $\alpha = 3000$ and $\beta = 1$. Also, the uncertainty radius was selected as $r_i = 0.1$ meters.

For the GPD plots in simulations, a grid for discritization of the space was utilized. The points in discrete space were converted into the metric space for the selected resolution and vice versa.

The parameter k_ω affects the convergence rate of the weight estimator. Moreover, K_p shows the matrix representation for the holonomic coverage controller. Also, by changing the α and β the convergence speed of the Hopfield Network estimator is adjusted.

First, the simulation results for 5, 10 and 15 agents are shown. The first agent had a performance degradation of 10 percent. Second, the effect of α on the estimation error is explained. In the third section, the change of regions with respect to performances is given. In three simulation runs, the performance of the first robot was degraded by

10, 20 and 30 percent, respectively. In the last part, the simulation of the agent disorder algorithm is shown.

7.3.1 Results with 5, 10 and 15 agents

The first simulation was performed with 5 agents. The initial configuration of the agents was a circular configuration and they accomplished the collaborative coverage task by minimizing locational optimization function as depicted in Figure 7.28.

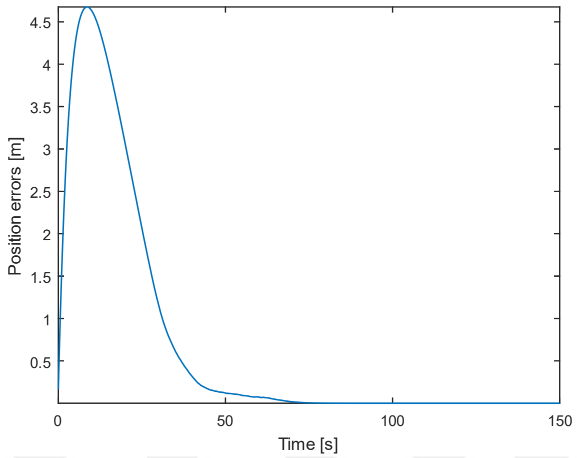

Figure 7.27 : Position errors.

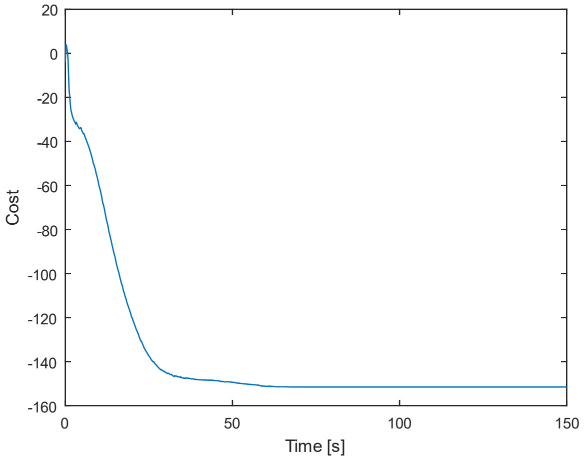

Figure 7.28 : Coverage cost.

Figure 7.27 illustrates the total position errors of the agents which asymptotically reach zero as given in the proof section.

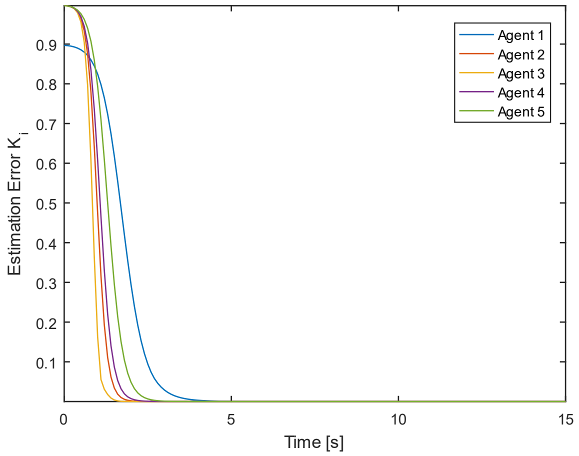

Figure 7.29 : Parameter estimation errors.

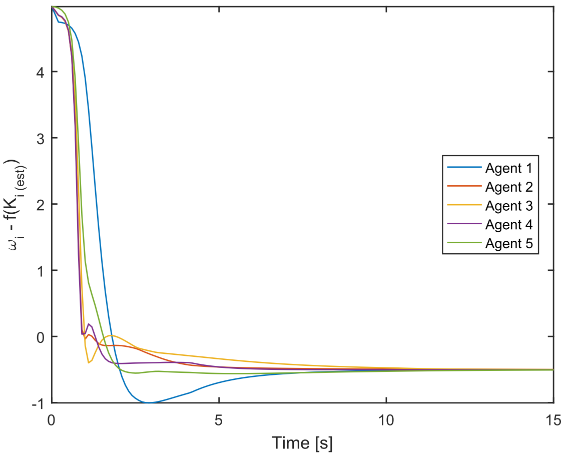

Figure 7.30 : The value of $\boldsymbol{w_i - f(\widehat{K}_i)}$.

Figure 7.29 depicts the parameter estimation error of the agents. As it can be seen from the figure, the estimation error of the HNN algorithm settles to zero asymptotically as given in *Theorem 5.1*. In Figure 7.30, the value of $w_i - f(\widehat{K}_i)$ achieves a common value among the agents as given in *Corollary 5.1*.

In Figure 7.31 and Figure 7.32, the weights w_i and the trajectories of the agents are shown, respectively. The weight of the first robot converges to a value less than the weights of the other robots since its actuator performance was degraded by 10 percent. Here, in the trajectory plot, black circles represent the start locations while the red circles illustrate the goal positions.

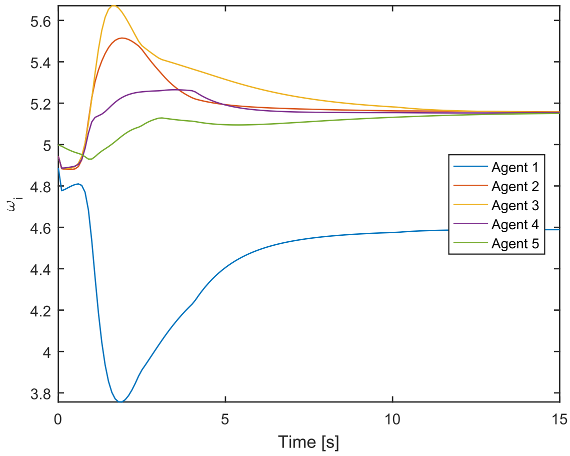

Figure 7.31 : The weights $\boldsymbol{w_i}$ of the agents.

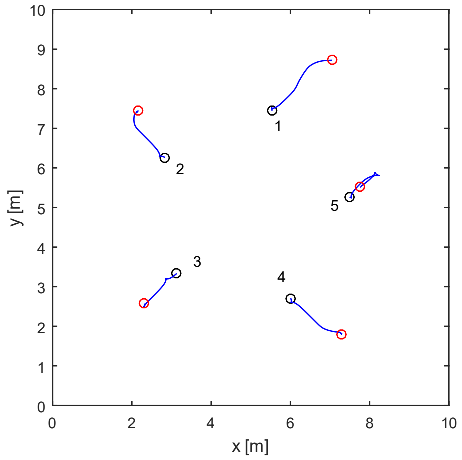

Figure 7.32 : The trajectories of the agents.

The second simulation is accomplished with 10 agents. Figure 7.33 and Figure 7.34 illustrate the position errors and coverage cost, respectively. In Figure 7.33, the total tracking error converges to zero asymptotically which results in the convergence of the positions to the centroid locations. In Figure 7.34, the minimization of the coverage cost causes the function to take a constant negative value since the weights of the agents are enforced to take positive values as given in (2.24).

Table 7.7 : Region ratios of the agents at the end configuration (HNN with 5 agents).

Agent	Region Ratio
1	0.1399
2	0.2218
3	0.2296
4	0.1827
5	0.1687

As shown in Table 7.7, the first agent takes smaller region than other agents due to its degraded performance.

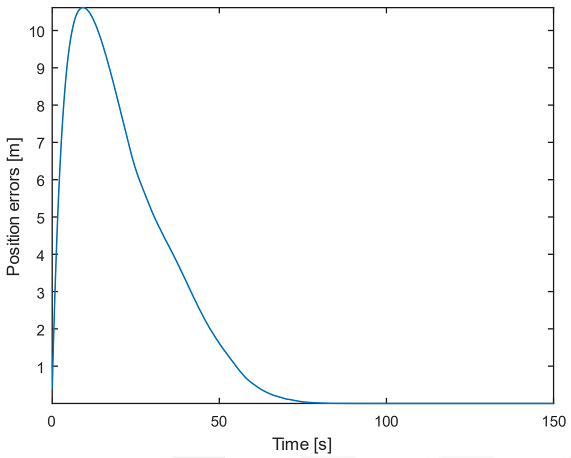

Figure 7.33 : Position errors of the agents.

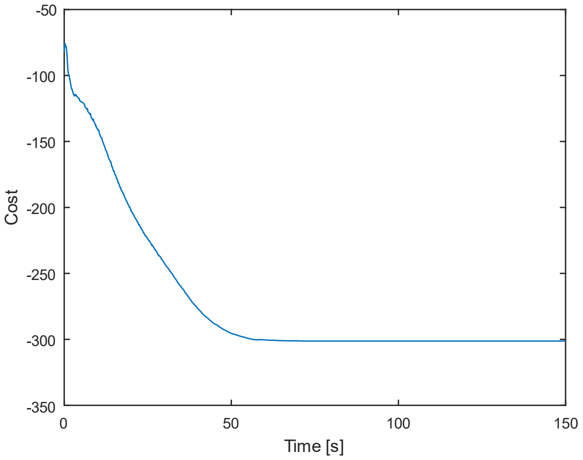

Figure 7.34 : Coverage cost.

Figure 7.35 and Figure 7.36 depict the parameter estimation errors of HNN and the value of $w_i - f(\widehat{K}_i)$. Again, the estimation errors asymptotically go to zero which enforces the parameters to converge to their real values. Also, the value of $w_i - f(\widehat{K}_i)$ has a common value among the agents.

Table 7.**8** : Region ratios of the agents at the end configuration (HNN with 10 agents).

Agent	Region Ratio
1	0.0707
2	0.0889
3	0.0809
4	0.0780
5	0.1247
6	0.0908
7	0.0938
8	0.0731
9	0.0724
10	0.1253

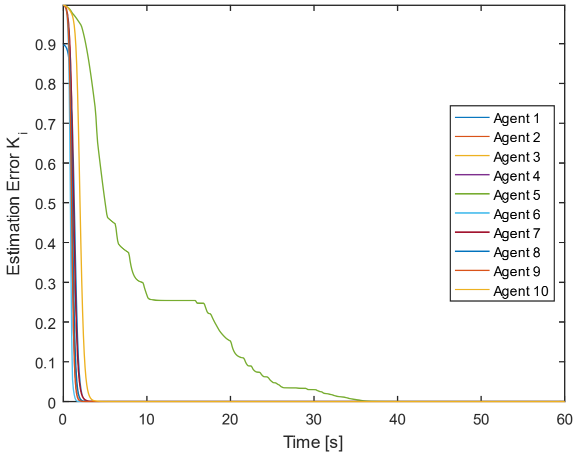

Figure 7.35 : Parameter estimation errors.

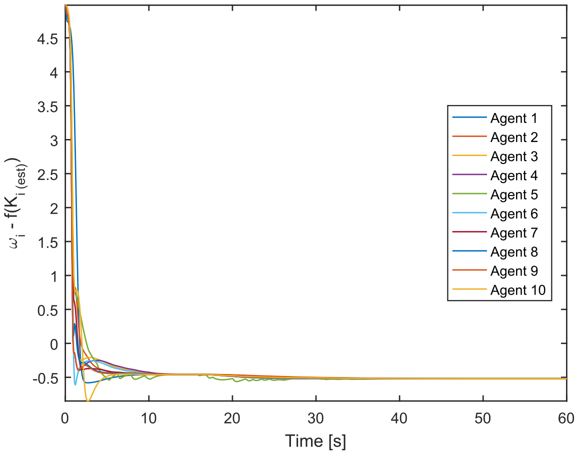

Figure 7.36 : The value of $\boldsymbol{w_i - f(\widehat{K}_i)}$.

Figure 7.35 and Figure 7.36 illustrate the parameter estimation errors of HNN and the value of $w_i - f(\widehat{K}_i)$.

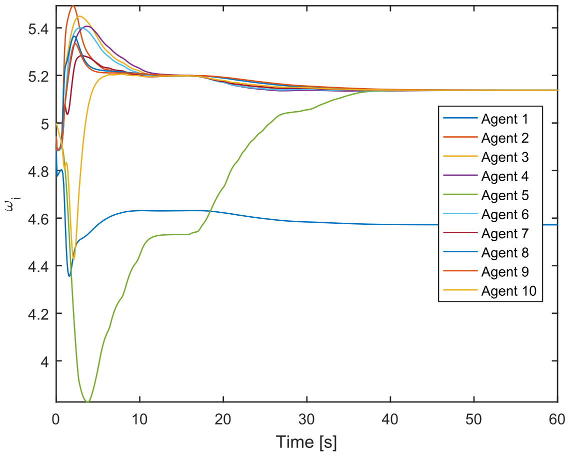

Figure 7.37 : The weights of the agents.

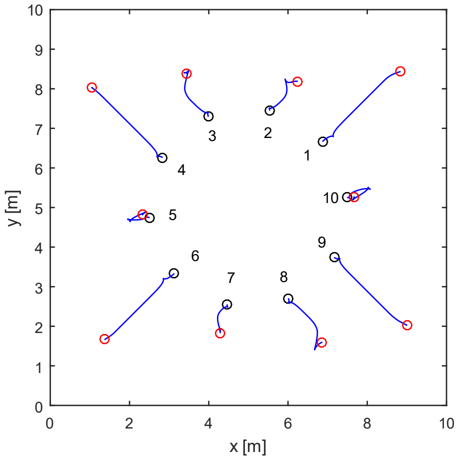

Figure 7.38 : The trajectories of the agents.

Again, the estimation errors converge to zero asymptotically showing that the parameters converge to their real values. Also, the $w_i - f(\widehat{K}_i)$ settles to a common value among the agents.

In Figure 7.37 and Figure 7.38, the calculated weights of the agents together with the trajectories are given, respectively. The weight of the first robot takes a less value than the weights of the other robots since its actuation performance was degraded by 10 percent.

According to the Table 7.8, from the region ratios, the first agent takes smaller region than the other agents because it has weaker actuators.

The third simulation is performed with 15 agents. In Figure 7.39 and Figure 7.40, the position errors and the coverage cost are illustrated, respectively. The position errors of the robots asymptotically reached zero and the coverage cost converged to a minimum value after the coverage task was accomplished.

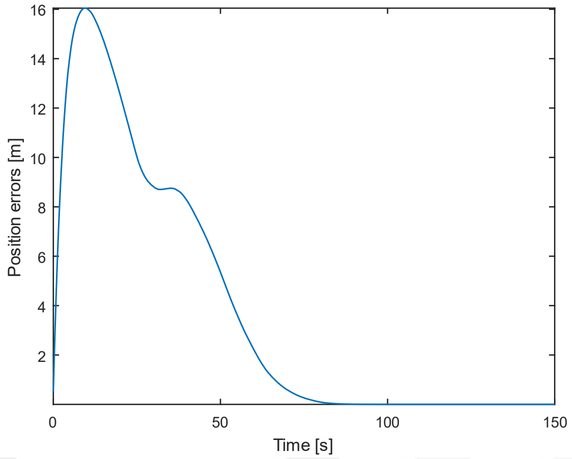

Figure 7.39 : Position errors.

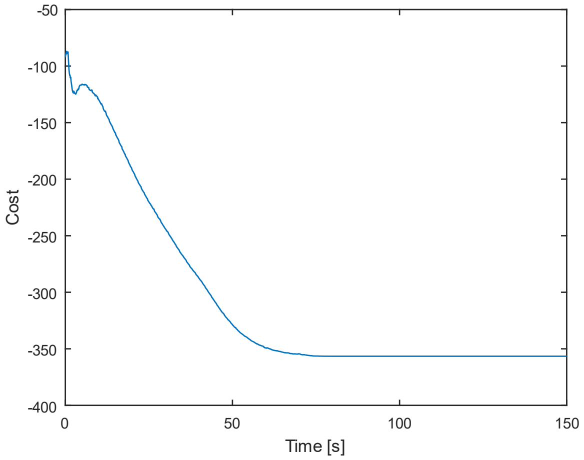

Figure 7.40 : Coverage cost.

The asymptotical convergence of the parameter estimation errors of the Hopfield Network is given in Figure 7.41, obeying the *Theorem 5.1*.

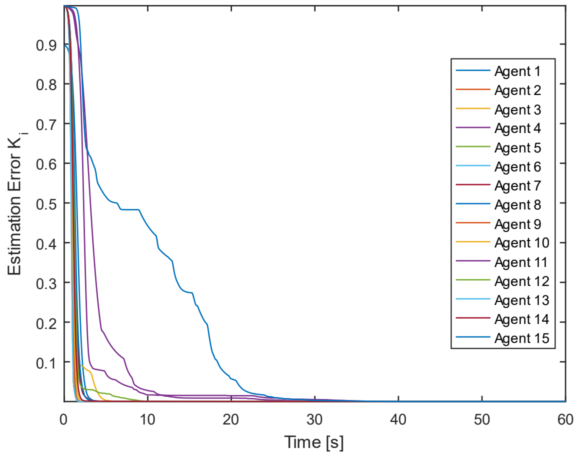

Figure 7.41 : Parameter estimation errors.

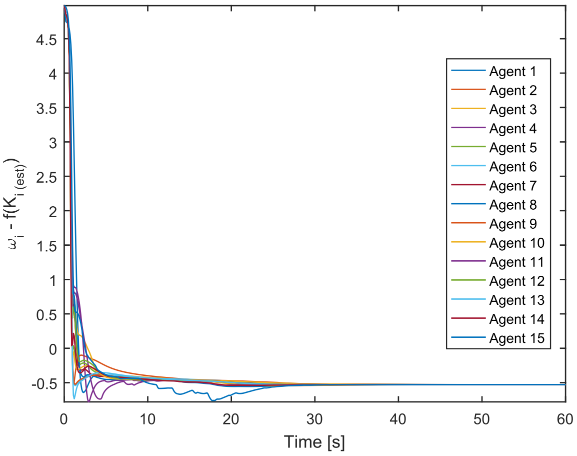

Figure 7.42 : The value of $\boldsymbol{w_i - f(\widehat{K}_i)}$.

Figure 7.42 illustrates that the value of $w_i - f(\widehat{K}_i)$ reaches a common value among the agents as given in the *Corollary 5.1*.

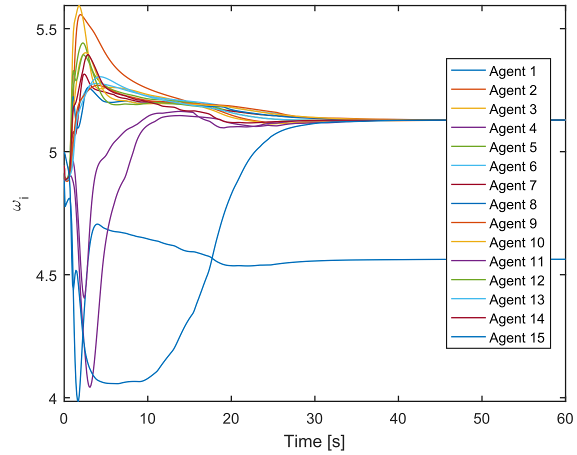

Figure 7.43 : The weights of the agents.

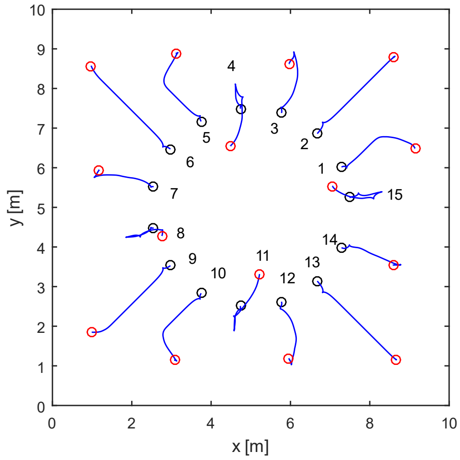

Figure 7.44 : The trajectories of the agents.

Figure 7.43 and Figure 7.44 give the weight of agents and the trajectories, respectively. A lower value for the weight of the first agent than the other agents was achieved since it has a degraded performance. The trajectories of the agents show the convergence to the optimal coverage positions of the agents, as illustrated in Figure 7.44.

Table 7.9 : Region ratios of the agents at the end configuration (HNN with 15 agents).

Agent	Region Ratio
1	0.0393
2	0.0618
3	0.0628
4	0.0620
5	0.0538
6	0.0513
7	0.0642
8	0.0599
9	0.0674
10	0.0579
11	0.0600
12	0.0573
13	0.0576
14	0.0635
15	0.0598

The region ratios of the agents at the end of the simulation are shown in Table 7.9. As it turns out from the region ratios, the first agent has less region than the other agents.

7.3.2 Effect of Hopfield Network parameter

The simulation was performed with 5 agents with a workspace chosen as 5x5 meters. By changing the parameter α of the Hopfield Network, the influence on the convergence speed is examined.

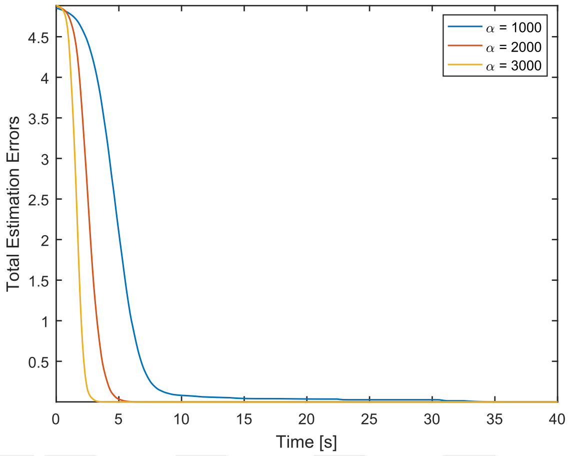

Figure 7.45 : The effect of the parameter $\boldsymbol{\alpha}$ of the HNN on the convergence speed.

Figure 7.45 illustrates that if we increase the α parameter, the convergence rate is improved. It is worth to note that large values of the convergence parameter may lead to divergence of the estimator.

7.3.3 Change of regions with respect to performances

For observing the effect of change of regions with respect to performances, three simulation runs were performed in which we degraded the performance of the first agent by 10, 20 and 30 percent, respectively. In Figure 7.46, Figure 7.47 and Figure 7.48, the corresponding GPDs to the simulation runs at the end configurations are illustrated.

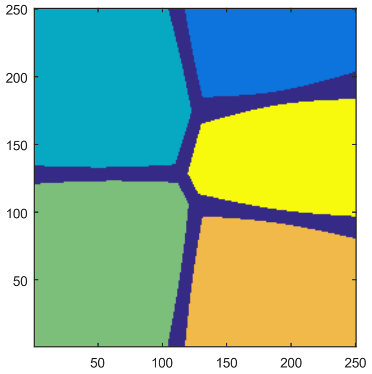

Figure 7.46 : Blue region: first agent (degraded by 10 percent).

In figures, the area of the first agent are shown with blue color.

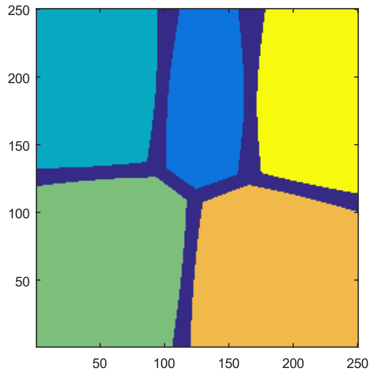

Figure 7.47 : Blue region: first agent (degraded by 20 percent).

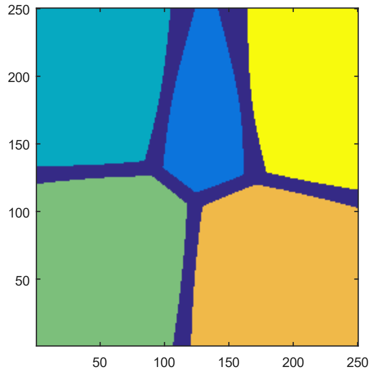

Figure 7.48 : Blue region: first agent (degraded by 30 percent).

The increase in the amount of the degradation of the actuation performance lead the region of the corresponding agent to become smaller. Thus, if the actuation performance is decreased, a smaller region is taken from the environment for the corresponding agent.

Table 7.10 : Region ratios of the agents at the end configuration (HNN with 10% degradation).

Agent	Region Ratio
1	0.1224
2	0.2171
3	0.2204
4	0.1802
5	0.1465

Table 7.**11** : Region ratios of the agents at the end configuration (HNN with 20% degradation).

Agent	Region Ratio
1	0.1132
2	0.1739
3	0.2200
4	0.2217
5	0.1576

The results with degradations of 10%, 20% and 30% are shown in Table 7.10,

Table 7.11 and Table 7.12, respectively.

Table 7.**12** : Region ratios of the agents at the end configuration (HNN with 30% degradation).

Agent	Region Ratio
1	0.0965
2	0.1813
3	0.2206
4	0.2208
5	0.1660

If the amount of the performance degradation rises, the region ratio of the first increases.

7.3.4 Agent disorder

The simulation showing the failure of the agent was performed with 5 agents in 5x5 meter environment. The demonstration corresponds to the case where the parameter vector K_i becomes zero. The proposed algorithm identifies the immobilized agent and sets its parameter vector to zero in order to degrade its workspace utilization to zero. In the simulation run, the immobilized agent was chosen as the first agent.

The position errors and coverage cost are depicted in Figure 7.49 and Figure 7.50, respectively. The total position error do not converge to zero since the first agent is immobilized.

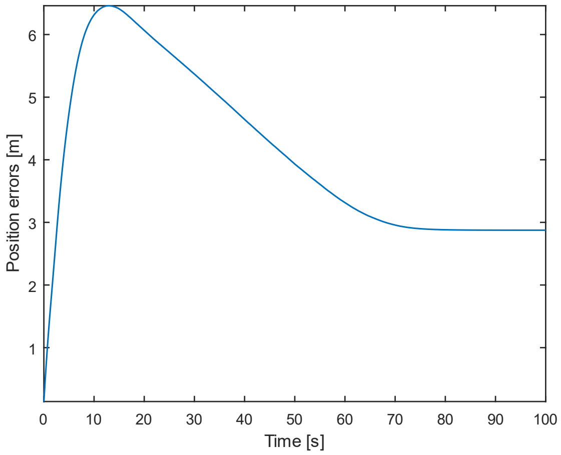

Figure 7.49 : Position errors.

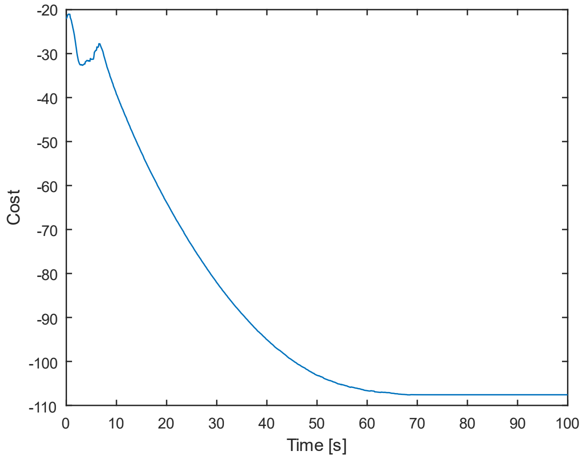

Figure 7.50 : Coverage cost.

The control law enforces the agents to go to the optimal centroid locations except for the first agent. Also, a minimum value is reached for the coverage cost at the end of the simulation since the coverage task is finished.

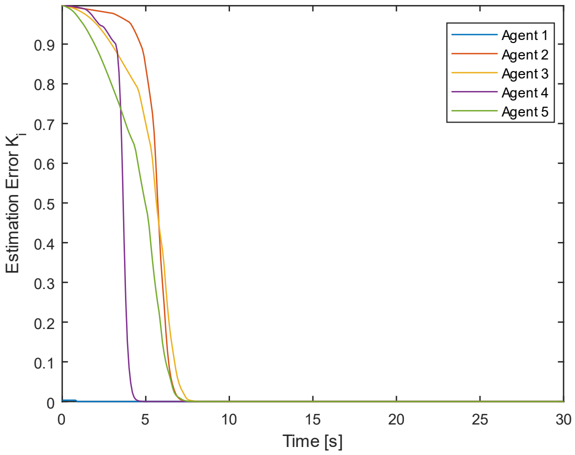

Figure 7.51 : Parameter estimation errors.

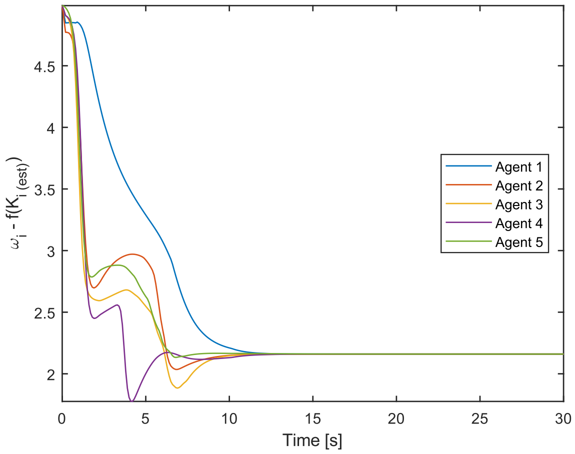

Figure 7.52 : The value of $\boldsymbol{w_i - f(\widehat{K}_i)}$.

Figure 7.51 and Figure 7.52 illustrate the parameter estimation errors of the HNN and the value of the $w_i - f(\widehat{K}_i)$, respectively.

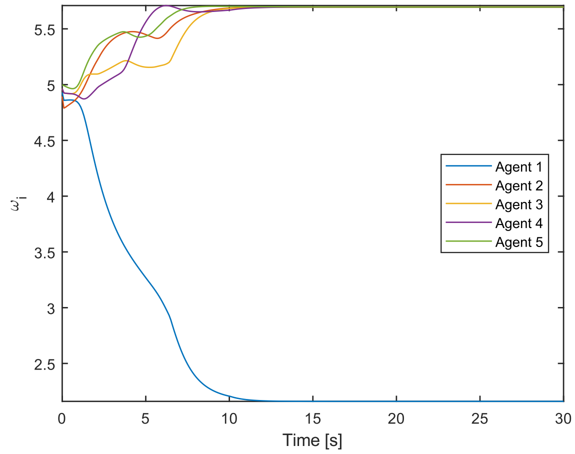

Figure 7.53 : The weights of the agents.

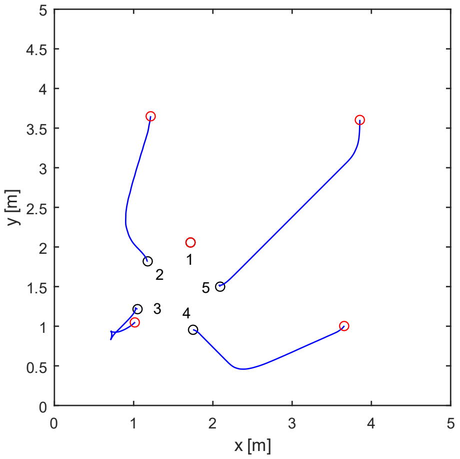

Figure 7.54 : The trajectories of the agents.

In Figure 7.51, asymptotical convergence is achieved for the estimation error and in Figure 7.52, a common value is reached for the value of $w_i - f(\widehat{K}_i)$ among the agents.

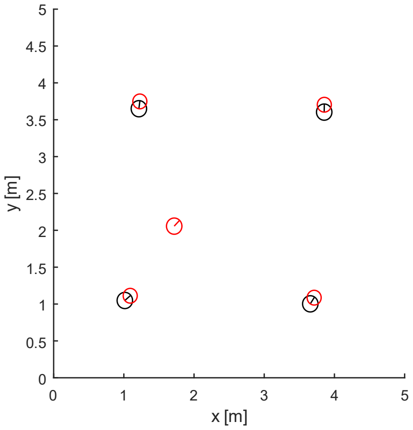

Figure 7.55 : The end configurations of the agents.

In Figure 7.53, the weights settle to constant values and the weight of the immobilized agent reaches zero. From Figure 7.54, it is concluded that the first agent cannot achieve the optimal centroid location. The other agents accomplish the optimal coverage.

Table 7.13 : Region ratios of the agents at the end configuration (HNN with agent disorder).

Agent	Region Ratio
1	0
2	0.2479
3	0.2007
4	0.2231
5	0.2412

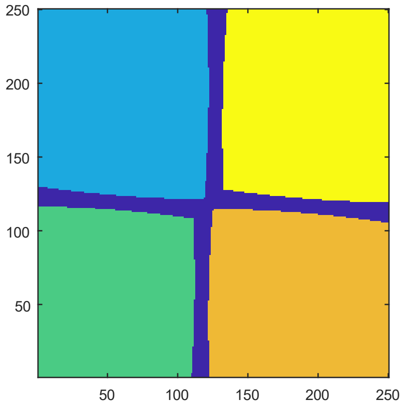

Figure 7.56 : GPD at the end configuration of the agents.

In Figure 7.55, the immobilized agent is represented with red color. The rest of the agents are able to finish the coverage task by enforcing the position errors to go to zero. Also, the red circles give centroid locations of each agent.

The reference point *P* of the point-offset controller has an axle distance *l* to the center of the robot, as given in (5.5). Thus, the center point of the robot achieves a goal point in a ball with radius *l* centered at the centroid location when the reference point *P* go to centroid position which is illustrated in Figure 7.55.

Figure 7.56 depicts the GPD at the end configurations for the simulation run. It is concluded that the first agent has no region at all since its estimation vector was zero which was set by the agent disorder algorithm.

Table 7.13 shows the region ratios of the agents. The region ratio for the immobilized agent is zero.

7.4 Energy-Efficient adaptation with Hopfield Neural Networks

In this section, the results of the MATLAB simulations were presented within a map of 5x5 meters. The simulation parameters were $k_\omega = 0.2$, $r_{robot} = 0.11$, $l = 0.1$, $\alpha = 6000$ and $\beta = 1$. Additionally, the vertices of the map were $(0.625, 0)$, $(3.125, 0)$, $(5.0, 4.375)$, $(3.75, 5.0)$ and $(0, 4.375)$.

The convergence speed of the weight estimator and HNN can be adjusted by the parameters k_ω, α and β.

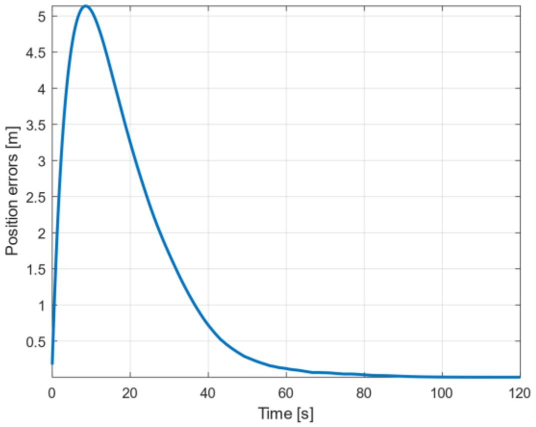

Figure 7.57 : Position errors.

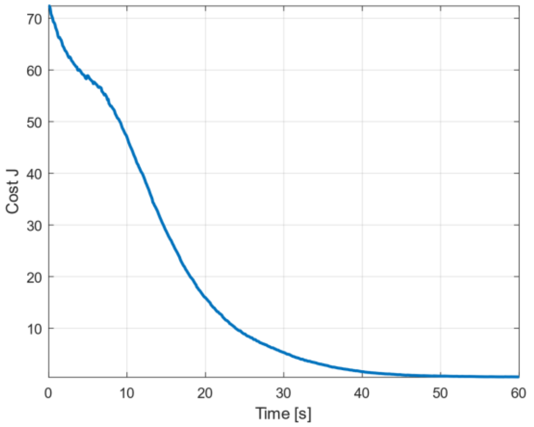

Figure 7.58 : Coverage cost.

In the first simulation, the actuation performance of the first agent was 90% where the performances of the other five agents were kept as 100%. Figure 7.49 and Figure 7.50 depict the position errors and the coverage cost, respectively. The results show that the position errors reach zero asymptotically. Also, the convergence of the coverage cost to the local minimum can be seen at the end of the simulation.

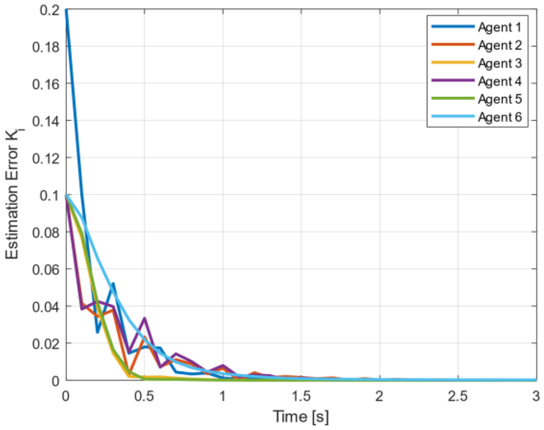

Figure 7.59 : Parameter estimation errors.

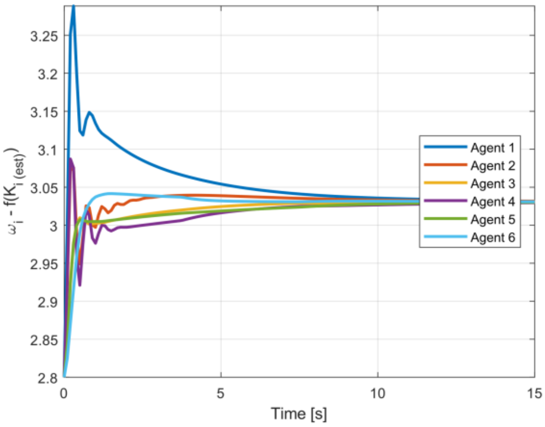

Figure 7.60 : The value of $w_i - f_{perf}(\widehat{K}_i)$.

The HNN parameter estimation errors are given in Figure 7.59. They converge to zero asymptotically as shown in the stability proof. In Figure 7.60, the value of $w_i - f_{perf}(\widehat{K}_i)$ converge to the same constant among the agents as given in *Corollary 5.1*.

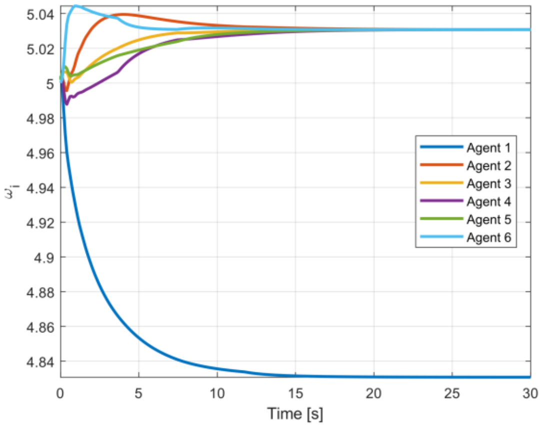

Figure 7.61 : The weights of the agents.

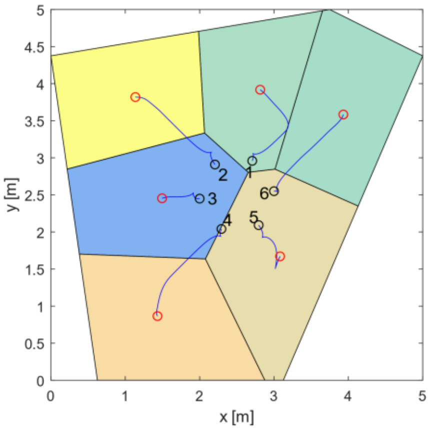

Figure 7.62 : The trajectories of the agents.

Figure 7.61 and Figure 7.62 illustrate the calculated PD weights and the trajectories of the agents with the Power Diagram regions at the end of the simulation, respectively. The convergence of the weight of the first robot to a lower value than the other robots is depicted which is caused by the weak actuators of the first agent. The results show that the algorithm automatically assign the regions according to the actuation performances. In Figure 7.62, the end configurations are shown with red circles while the initial configurations are illustrated by black circles.

In Table 7.14, the resulting region ratios at the end of the simulations are shown. The first agent has a lower region ratio than the other ones since it had a performance degradation. It should be noted that the convex shape of the map caused the empty regions in the workspace as illustrated in Figure 7.44.

Table 7.14 : Region ratios of the agents at the end configurations.

Agent #	Region Ratio	Agent #	Region Ratio
1	0.0935	4	0.1280
2	0.1113	5	0.1313
3	0.1175	6	0.1181

Table 7.15 summarizes three simulation results where the parameter r_i was changed and the actuation performances were kept constant at 100% (1.0). With the rise in the parameter r_i, the energy consumption of the agents decreases, and it results in an increase in the coverage time.

Table 7.15 : Change of energy consumption with respect to r_i.

Simulation #	s_i / r_i	Actuation Performances of the Agents	Coverage Time [s]	Energy Consumption
1	1.0 / 1.0	All are the same (1.0)	93.6	0.2766
2	1.0 / 2.0	All are the same (1.0)	123.5	0.1954
3	1.0 / 4.0	All are the same (1.0)	168	0.1378

Table 7.16 : Change of coverage time with respect to s_i.

Simulation #	s_i / r_i	Actuation Performances of the Agents	Coverage Time [s]	Energy Consumption
1	1.0 / 1.0	All are the same (1.0)	93.6	0.2766
4	0.75 / 1.0	All are the same (1.0)	105.2	0.2392
5	0.125 / 1.0	All are the same (1.0)	266.9	0.0974

Table 7.16 illustrates the first, fourth and fifth simulation results. The s_i parameter was increased and the actuation performances were kept constant at 1.0. With the decrease of the s_i parameter, the energy consumption is decreased, and the coverage time is increased.

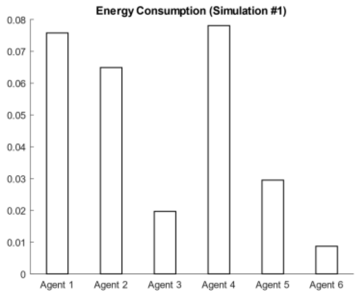

Figure 7.63 : Energy consumptions of the agents (Simulation #1).

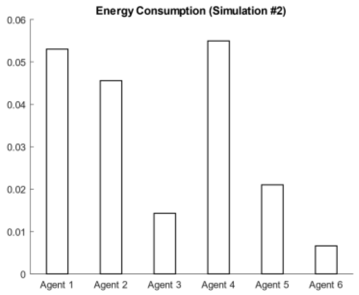

Figure 7.64 : Energy consumptions of the agents (Simulation #2).

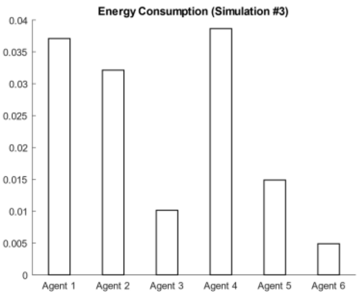

Figure 7.65 : Energy consumptions of the agents (Simulation #3).

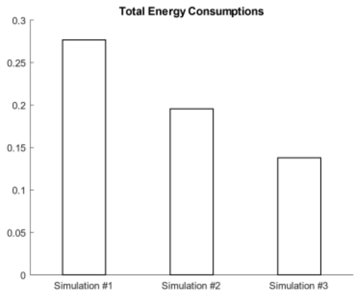

Figure 7.66 : Total energy consumptions.

In Figure 7.63, Figure 7.64 and Figure 7.65, the energy consumptions of the agents in first, second and third simulations are shown. Figure 7.66 depicts the overall energy consumption. The energy consumption decreases with the increase in the parameter r_i.

In Table 7.15 and Table 7.16, the resultant region ratios in the simulations 1, 6 and 7 are illustrated. In the first simulation, the actuation performances are taken as 1.0 corresponding to 100% actuation capability. In the simulation 6, the performance of the first agent was 90% and in the simulation 7, the performance of the second agent was taken as 80%. As it can be seen from the results, the performance degradations in actuators results in a decreased region ratio for the corresponding agent.

The results obtained with the MATLAB simulations were satisfactory which show that the algorithm is capable of automatically assigning the areas from the workspace to the agents with respect to their actuation performances. Also, the stability of the control and estimation laws were verified with simulations.

7.5 Discussion

In this section, the simulation results regarding the methods given in the previous sections are presented. First, the MATLAB simulation results of the adaptive coverage control algorithm with GPDs are shown. Second, the simulation results of the adaptation to performance variations with GPVDs and non-holonomic control law are given. Next, the MATLAB results regarding the collaborative coverage algorithm with HNNs and GPVDs are shown with the case of immobilized agent. Lastly, the results of energy-efficient collaborative coverage control algorithm with HNNs are presented. According to the simulation results, the theoretical Lyapunov stability proofs are verified. All of the systems in the methods show global asymptotical convergence as proven in the related sections.

8. EXPERIMENTS

In this section, the ROS implementation and experimental studies are presented.

8.1 ROS Implementation

The ROS/Gazebo simulation was carried out with three identical Turtlebot 3 robots in a 5x5 meter environment. The ROS nodes were written in C++. The uncertainty radius for the position was taken as 0.1 meters. The actuator performance of the first agent was 10% lower than the other two agents in simulations.

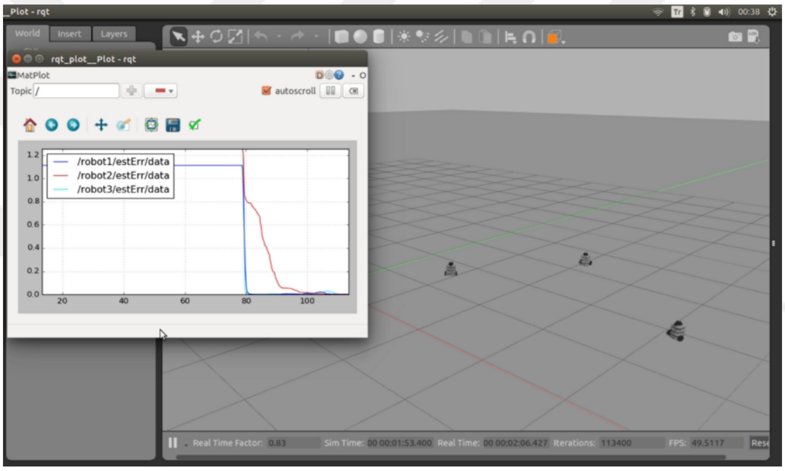

Figure 8.1 : A simulation frame from ROS/Gazebo with Turtlebot 3 robots. The plot window shows the parameter estimation errors.

The simulation with adaptation with non-holonomic estimator starts with a ROS node named as Coverage node running for each agent separately in a decentralized way. Each node estimates $\widehat{K}_i$ vector from its own motion by using non-holonomic adaptive estimator by using (4.10) - (4.12) and then calculates the weight of the agent by using (4.5). The node then passes the weights to the GPD algorithm.

After the centroid locations are found by using the GPV-cells and the non-holonomic control law is executed. The obtained velocities are sent to the corresponding topic of the agent. The corresponding nodes and topics are shown in Figure A.1. Here, cvgNode1, cvgNode2 and cvgNode3 are the coverage nodes which run the main decentralized algorithm for each agent. posePub is the topic which shares the position and weight information with the other agents. The ground_truth / state topics gives the position and velocity of the agent. The linear and angular velocity signals are sent over the topic cmd_vel for each agent.

Table 8.17 : Region ratios of the agents at the end configuration (with non-holonomic estimator).

Agent	Region Ratio
1	0.248
2	0.3194
3	0.3554

Table 8.18 : Weights of the agents at the end configuration (with non-holonomic estimator).

Agent	Weight
1	9.7327
2	10.1271
3	10.1224

Table 8.17 gives the resulting region ratios at the end configuration for the simulation. The first agent took a smaller partion from the workspace than the other agents since its actuation performance was degraded 10%. Similiarly, in Table 8.18 the weight of the first agent is less than the weights of the other agents.

The simulation with adaptation with Hopfield estimator starts with a ROS node named as Coverage node running for each agent separately in a decentralized way. Each node estimates $\widehat{K_i}$ vector from its own motion by using HNNs by using (5.3) and (5.4) and then calculates the weight of the agent by using (5.20) and (5.21). The node then passes the weights to the GPD algorithm.

After the centroid locations are found by using the GPV-cells and the non-holonomic control law is executed. The obtained velocities are sent to the corresponding topic of the agent. The corresponding nodes and topics are shown in Figure A.2. In the figure, cvgNode1, cvgNode2 and cvgNode3, posePub and ground_truth / state topics have the same functions as in the non-holonomic estimator algorithm. Also, the linear and angular velocity signals are sent over the wheel_right_joint_velocity_controller / command and wheel_left_joint_velocity_controller / command topics for each agent. The wheel_right_joint_velocity_controller and wheel_left_joint_velocity_controller topics come from the joint velocity controllers defined for the wheels.

Table 8.19 : Region ratios of the agents at the end configuration (with Hopfield estimator).

Agent	Region Ratio
1	0.1947
2	0.3401
3	0.3928

Table 8.20 : Weights of the agents at the end configuration (with Hopfield estimator).

Agent	Weight
1	4.32957
2	5.72433
3	5.64324

In Table 8.19, the region ratios at the end configuration are shown for the Hopfield estimator case. The first agent has a smaller region than the other agents because its actuation performance was 10% worse than the other ones. Similiar case can be seen from the Table 8.20 where the weights of the other agents are greater than the weights of the first agent.

8.2 Experiments: Adaptation to Performance Variations with HNNs Under Localization Uncertainty

Two Turtlebot 2 robots were utilized in the ground experiments conducted in 2x2 meter environment. The actuation performance of the first robot was degraded by 10%. The robots start from the initial configurations and collaboratively achieve the optimal coverage configuration.

8.2.1 Experimental Setup

The experiments were performed in the ITU Robotics Laboratory. The computers on the robots were Acer Aspire E11 with Intel Celeron N2940 processors and 4 GB of memory running Ubuntu 14 and ROS Indigo. The agents were differential drive robots with a maximum velocity of 0.65 m/s and a maximum payload of 5 kg. The weight of each agent was 6.3 kg.

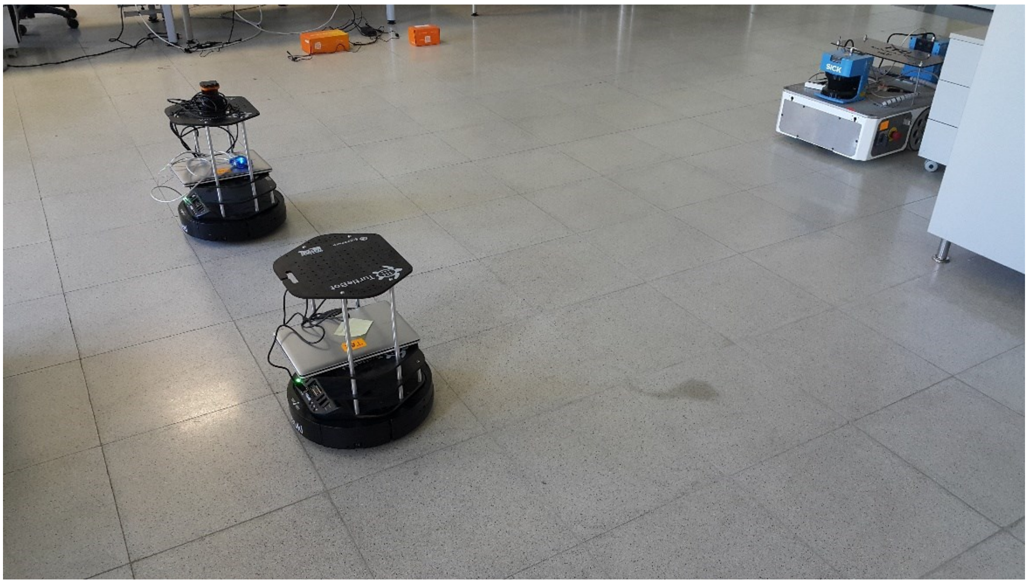

Figure 8.2 : The initial configurations of the robots (ITU Robotics Laboratory).

Figure 8.2 illustrates the experimental setup in the ITU Robotics Laboratory. For the localization of the agents, the wheel odometry was used. The ROS driver "kobuki_node" performed the low level tasks for the algorithm on each robot. The communication between the agents was done by using publisher/subscriber architecture making use of ROS topics. For the Wi-Fi communication, a 450Mbps Wi-Fi N access point was utilized.

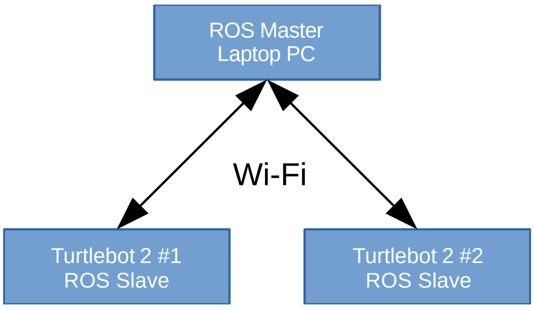

Figure 8.3 : Communication architecture used in the experimental setup.

The ROS master computer was a laptop PC running Ubuntu and ROS Kinetic. The connection of the computers on the agents to the ROS master node was done over the Wi-Fi network as shown in the communication diagram in Figure 8.3.

8.2.2 Experimental Results

The coverage node is the node which runs the main coverage algorithm and is running on the controller PC of the agent on ROS in a distributed manner. The agents estimate their own performance parameters, the weights and calculate the GPD regions by communicating with each other. Then, they run the control law after calculating the centroid locations.

The ground experiment is performed three times and the results are shown in the table of region ratios.

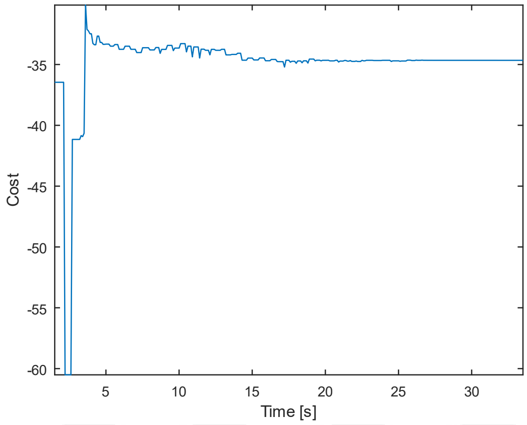

Figure 8.4 : Coverage cost.

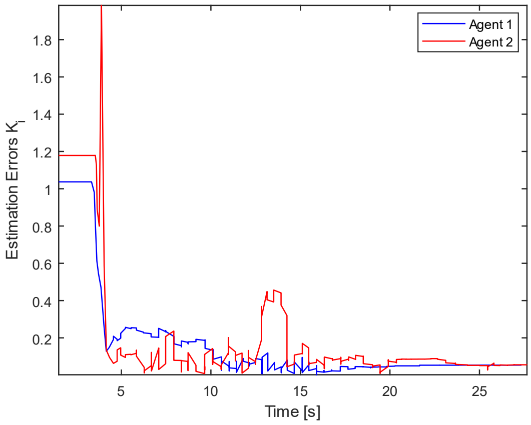

Figure 8.5 : Estimation errors.

For the first experiment, Figure 8.4 illustrates the total coverage cost of the agents. The cost achieves its local minima as the coverage task is complete. In addition, Figure 8.5 depicts the estimation errors of each agent converging to zero.

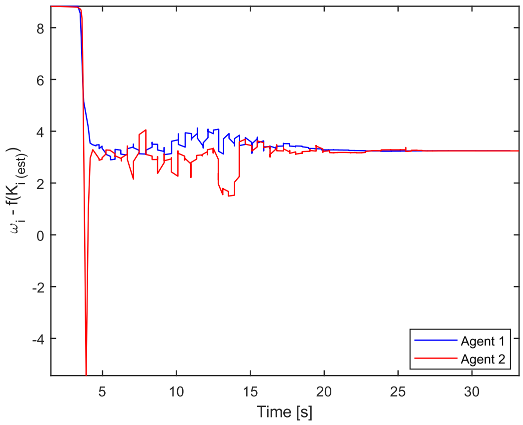

Figure 8.6 : The value of $\boldsymbol{w_i - f(\widehat{K}_i)}$.

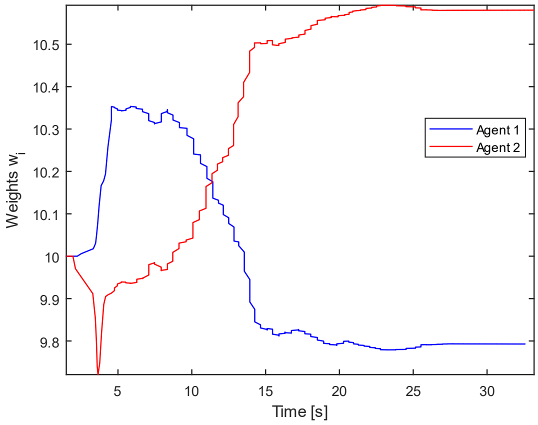

Figure 8.7 : The weights of the agents.

Figure 8.6 shows the value of $w_i - f(\widehat{K}_i)$. As given in the *Corollary 5.1*, the value goes to the same value among all agents. Also, Figure 8.7 illustrates the weight values which are the outputs of the the online estimator. As expected, the first agent has a weight value which is less than the other agent since it had a actuation performance degradation.

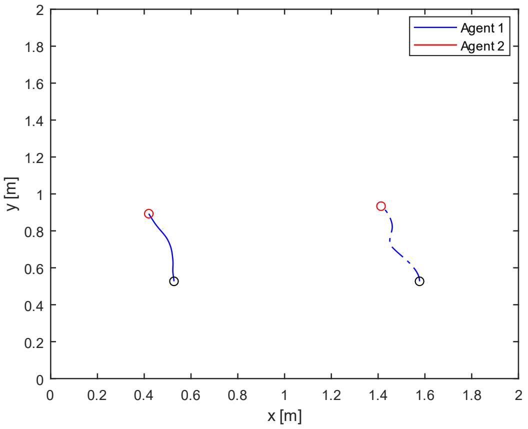

Figure 8.8 : The trajectories of the agents.

In Figure 8.8, the trajectories of the two agents are given. The black circles give the initial positions while the red circles denote the final locations.

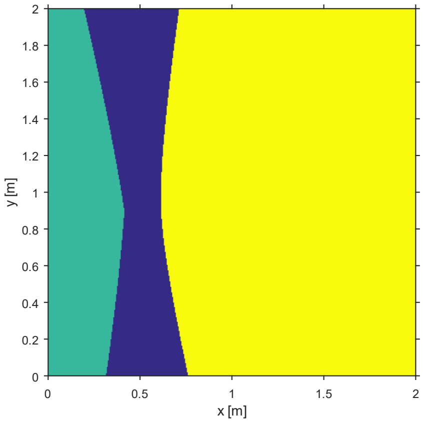

Figure 8.9 : The GPD plot obtained from the first experiment.

At the end configuration, the obtained regions are shown in Figure 8.9 for the first experiment. The first agent has a less region ratio compared to the second one. In the first experiment, the α parameter is selected as $\alpha = 6000$.

Table 8.21 : Region ratios of the agents at the end configuration with HNN of three experiments.

Agent	Region Ratio Experiment 1	Region Ratio Experiment 2	Region Ratio Experiment 3
1	0.1950	0.1917	0.2120
2	0.6484	0.6534	0.6342

At the end configuration, the obtained region ratios are shown in Table 8.21 for the second, third and fourth experiments. Here, in all of the three experiments, the parameter of the Hopfield Network α was selected as $\alpha = 3000$. The results show that the first agent had a region ratio less than the other agent in each experiment. The three experiments gave similar results. The weights of robots were assigned according to their actuator performances by the distributed coverage algorithm automatically.

The experiments show that similar results were obtained compared to the Gazebo and MATLAB simulations. The agents which have better actuation performances took larger regions than the weaker ones from the environment.

8.3 Experiments: Energy-Efficient Coverage Control with Hopfield Networks

The ground experiments were performed in ITU Robotics Laboratory with two Turtlebot 2 robots. The ROS/Gazebo multi-robot simulator was utilized for simulations and the same ROS node was used in experiments. The ROS implementation package includes a coverage ROS node written in C++.

In the ROS/Gazebo simulation the three identical Turtlebot 2 robots were simulated in a 5x5 meters environment as shown in Figure 8.10. The same coverage node run in distributed and decentralized way. The communication of the robots are performed by utilization of the Power Diagram neighbors.

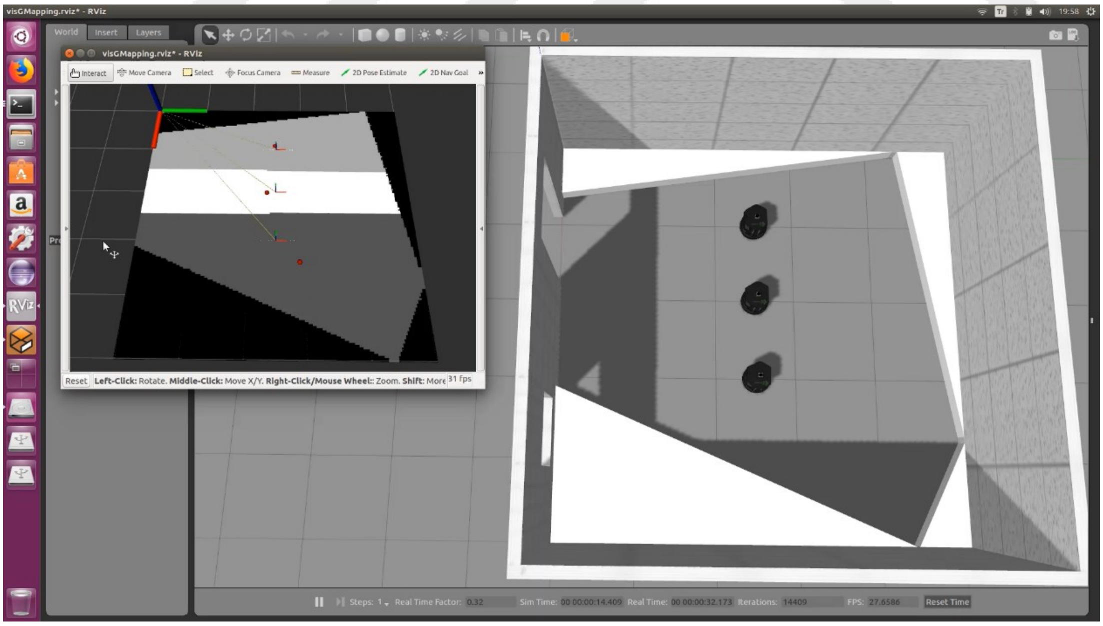

Figure 8.10 : The ROS/Gazebo Simulation Environment.

The algorithm in the coverage ROS node starts by calculating Power Voronoi regions by using initial positions and weight values. Then, the parameter vector of each agent are estimated by the HNN algorithm. The Power Voronoi weights are obtained by the weight estimator and the desired performance function. After, by using the Power

Diagrams and the calculated PD weights the regions are assigned to the agents. The algorithm finds the centroid locations from the PD regions, and the control inputs of each robot are calculated by using the control law. Thus, the optimal coverage configuration is achieved by running the algorithm online and in a decentralized way. At the end configuration, the assignment of the regions to the agents with respect to their different actuation performances is completed.

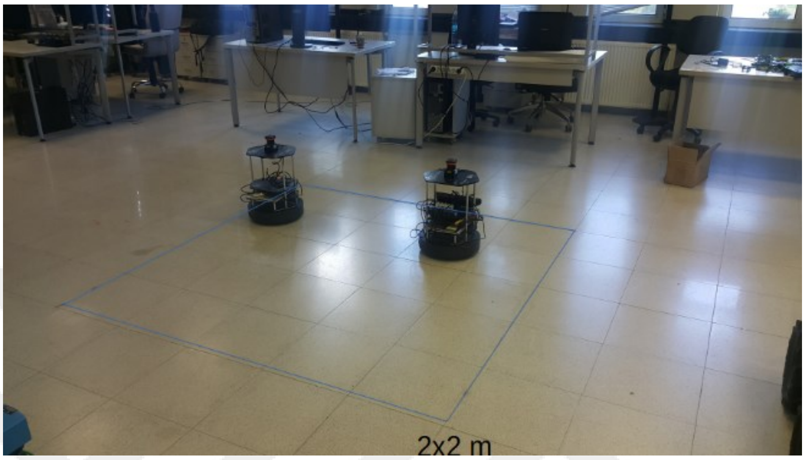

Figure 8.11 : The environment in which the experiments were performed.

In the experiments, the two Turtlebot 2 robots run the Coverage node in a decentralized manner and perform energy-efficient collaboration in a 2x2 meters environment. For localization, the SLAM gmapping ROS node (Gerkey, 2020) was utilized in conjunction with Hokuyo UTM-30LX laser scanners and wheel odometry.

The communication among the agents was performed over a Wi-Fi network. Two ROS slave Turtlebot 2 robots were connected to the same ROS master laptop over the communication link. The robots were equipped with laptop computers running Ubuntu, ROS Indigo and the coverage ROS node in real-time.

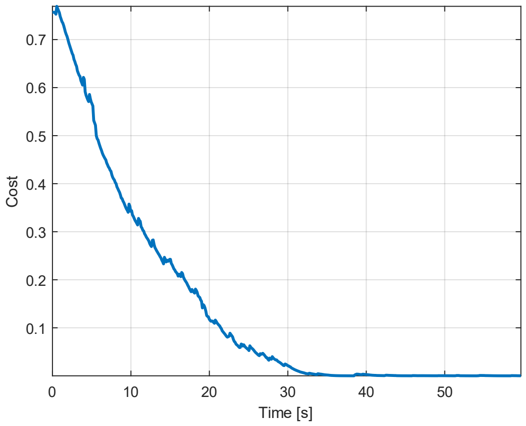

Figure 8.12 : The value of cost function J.

In the following sections, the results regarding to the ROS experiments are given. In the first result set, the performance of the first agent was lower than the performance of the other agents and the resultant plots regarding the algorithm are illustrated. Then, the coverage time along with the energy consumptions were investigated with respect to the change of the energy-efficient optimal controller parameters. Finally, the results regarding the change of the actuation performances of the first and second agent are shown.

In Figure 8.12, the energy-efficient coverage cost function is converged to the local minimum. Figure 8.13 depicts the parameter estimation errors which go to zero asymptotically and Figure 8.14 gives the value of $\boldsymbol{w_i} - \boldsymbol{f}(\widehat{\boldsymbol{K}}_i)$ converging to the same value among the two agents as given in *Corollary 5.1*.

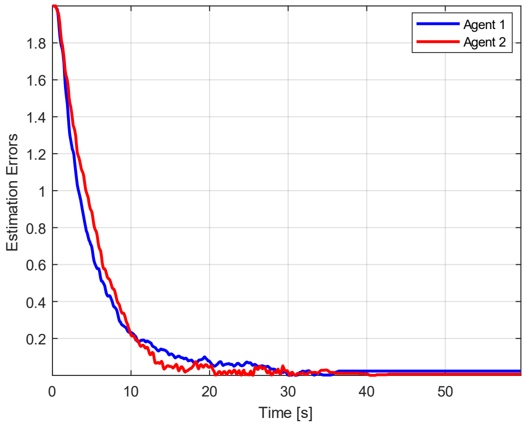

Figure 8.13 : The parameter estimation errors.

Figure 8.15 shows the weights of the agents with respect to time where the weight of the first agent converges to a lower value than the other one since the corresponding agent had an actuation performance degradation of 10%. In summary, the performance degradation is automatically compensated by the algorithm relatively by assigning a lower weight value and region from the workspace to the corresponding agent.

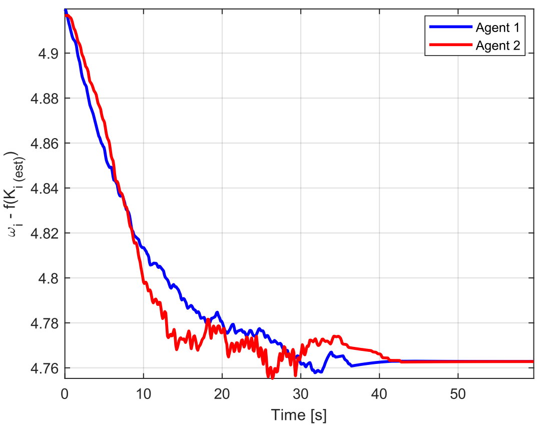

Figure 8.14 : The value of $\boldsymbol{w_i - f(\hat{K}_i)}$.

Figure 8.16, the trajectories of the agents obtained in the experiment are illustrated. The centroid location is calculated for each agent from its Power Voronoi region and the point-offset control law is executed.

Table 8.22 shows the increase in the coverage time of the agents as the parameter r_i increases. At the same time, the decrease in energy consumption is observed since the parameter r_i affects the energy consumption of the agents in the coverage optimization formulation.

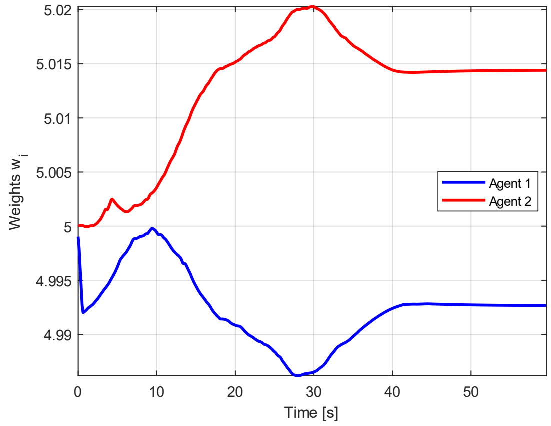

Figure 8.15 : The weights of the agents.

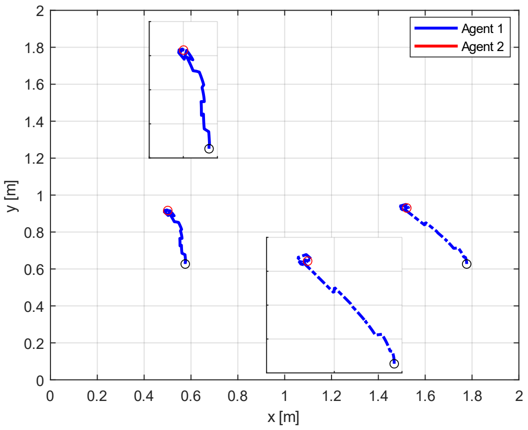

Figure 8.16 : **The trajectories of the agents.**Table 8.23, the increase in the coverage time with the decrease in the parameter s_i is illustrated. At the same time, the decrease in energy consumption occurs.

Table 8.22 : Change of energy consumption with respect to r_i.

Experiment #	s_i / r_i	Actuation Performances	Coverage Time [s]	Energy Consumption
1	1.0 / 1.0	0.9, 1.0	37.76	0.089034
2	1.0 / 2.0	0.9, 1.0	47.66	0.054833
3	1.0 / 4.0	0.9, 1.0	70.63	0.043552

Table 8.23 : Change of energy consumption with respect to s_i.

Experiment #	s_i / r_i	Actuation Performances	Coverage Time [s]	Energy Consumption
1	1.0 / 1.0	0.9, 1.0	37.76	0.089034
4	0.75 / 1.0	0.9, 1.0	40.25	0.072635
5	0.5 / 1.0	0.9, 1.0	47.37	0.058017

Table 8.24 shows the resultant region ratios with respect to the changes in the actuation performances of the agents. With the decrease in the actuation performance of the corresponding agent, the resultant region ratio decreases.

Table 8.24 : Change of region ratios with respect to actuation performances of the agents.

Experiment #	s_i / r_i	Actuation Performances	Region Ratios
6	1.0 / 1.0	0.9, 1.0	0.4, 0.6
7	1.0 / 1.0	0.8, 1.0	0.35, 0.65

Two videos of the experiments can be viewed at:

- https://web.itu.edu.tr/turanlim/video/exp-20200123.mp4
- https://web.itu.edu.tr/turanlim/video/exp-20200124.mp4

It can be concluded from the experiments that the algorithm has the capability of achieving the energy-efficient optimal coverage configuration and the collaboration among the agents. By making a trade-off between the energy consumption and the coverage time, reducing the energy consumption of the agents or decreasing the coverage time is possible. At the same time, the regions of the robots can be assigned by the algorithm which is done by learning their own actuation performances. Then, larger regions are given to the stronger robots, and smaller regions are assigned to the weaker ones. A fast and robust estimation is performed by the HNN which can be adjusted with the convergence parameter of the estimator.

8.4 Discussion

In this section, the experimental results of the previously presented coverage control algorithms are given with ROS/Gazebo results. The first set of experimental results is regarding the collaborative coverage control with GPVDs. Moreover, the set of second experimental results is related to the energy-efficient coverage control with HNNs and collaboration among the agents.

According to the experimental results, the systems of the corresponding methods show global asymptotical convergence, as shown in proofs and MATLAB simulation results. The videos regarding the experiments are given. The methods given are able to perform collaboration among the agents according to their unequal actuation capabilities.

9. CONCLUSION

In this thesis, an adaptive coverage coordination algorithm and an adaptive coverage collaboration algorithm for a multi-agent system are introduced. The agents are considered as wheeled mobile robots. Their positions are known to be in uncertainty circles.

For the coordination method, an adaptive coordination algorithm is presented which estimates the density function and drives the robots to the optimal configuration. The robots communicate with each other by using a consensus protocol and take imprecise localization into account by making use of GPD algorithm.

For the proposed collaboration algorithms, it is assumed that the actuation performance capabilities of the agents differ from each other. The workspace partitioning is performed by using Guaranteed Power Diagrams (GPD). Since the actuation performances are not known beforehand, the robots learn the performances by making use of the online parameter estimator.

Also, three collaboration algorithms are proposed. In the first one, an adaptive estimator using non-holonomic model is utilized with a non-holonomic control law. In the second collaboration algorithm, the HNN estimator is used with a point-offset control law. The collaboration algorithm estimates the actuation parameters of the agents and assigns portions of the workspace to the robots according to their relative performance variations. Thus, a collaborative coverage task is performed which enable the agents to position themselves to the optimal locations by giving more region to the stronger robots and less region to the weaker ones. In the third algorithm, the collaboration is achieved similar to the second method, however Power Diagrams and an energy-efficient coverage optimal control law are utilized. In the energy-efficient coverage optimal control, both the control and estimation laws are changed. By using the algorithm, a trade-off between coverage time and energy consumption can be made. Meanwhile, the collaboration is achieved by using HNN and Voronoi Diagrams together with the weight estimation and point-offset control laws.

For the both coordination and collaboration methods, the Lyapunov-type stability proofs are given and the theoretical results are verified with MATLAB and ROS/Gazebo simulations and ground experiments. At the end section, the results of the ROS implementations for the two methods are presented. Simulation results with the ROS implementation and experimental results show the efficiency of the algorithm.

REFERENCES

Abdessameud, A., Polushin, I. G., Tayebi, A., & others. (2017). Distributed Coordination of Dynamical Multi-Agent Systems Under Directed Graphs and Constrained Information Exchange. *IEEE Trans. Automat. Contr.*, *62*(4), 1668–1683.

Aicardi, M., Casalino, G., Bicchi, A., & Balestrino, A. (1995). Closed loop steering of unicycle like vehicles via Lyapunov techniques. *IEEE Robotics & Automation Magazine*, *2*(1), 27–35.

Alonso, H., Mendonça, T., & Rocha, P. (2009). Hopfield neural networks for on-line parameter estimation. *Neural Networks*, *22*(4), 450–462.

Arslan, O., & Koditschek, D. E. (2016). Voronoi-based coverage control of heterogeneous disk-shaped robots. *Robotics and Automation (ICRA), 2016 IEEE International Conference On*, 4259–4266.

Atencia, M., & Joya, G. (2015). Hopfield networks: from optimization to adaptive control. *Neural Networks (IJCNN), 2015 International Joint Conference On*, 1–8.

Baker, C., Ramchurn, G., Teacy, L., & Jennings, N. (2016). *Factored Monte-Carlo tree search for coordinating UAVs in disaster response*.

Bhattacharya, S., Michael, N., & Kumar, V. (2013). Distributed coverage and exploration in unknown non-convex environments. In *Distributed autonomous robotic systems* (pp. 61–75). Springer.

Bircher, A., Kamel, M., Alexis, K., Burri, M., Oettershagen, P., Omari, S., … Siegwart, R. (2016). Three-dimensional coverage path planning via viewpoint resampling and tour optimization for aerial robots. *Autonomous Robots*, *40*(6), 1059–1078.

Bochkarev, S., & Smith, S. L. (2016). On minimizing turns in robot coverage path planning. *Automation Science and Engineering (CASE), 2016 IEEE International Conference On*, 1237–1242.

Breitenmoser, A., & Martinoli, A. (2016). On combining multi-robot coverage and reciprocal collision avoidance. In *Distributed Autonomous Robotic Systems* (pp. 49–64). Springer.

Cassandras, C. G., Lin, X., & Ding, X. (2013). An optimal control approach to the multi-agent persistent monitoring problem. *IEEE Transactions on Automatic Control*, *58*(4), 947–961.

Chen, G., & Lewis, F. L. (2011). Synchronizing networked lagrangian systems via binary control protocols. *IFAC Proceedings Volumes*, *44*(1), 1225–1230.

Chen, S., Xin, S., He, Y., & Wang, G. (2012). The closest and farthest points to an affine ellipse or ellipsoid. *Tsinghua Science and Technology*, *17*(4), 481–484.

Choi, Y.-C., & Ahn, H.-S. (2010). A survey on multi-agent reinforcement learning: Coordination problems. *Mechatronics and Embedded Systems and Applications (MESA), 2010 IEEE/ASME International Conference On*, 81–86.

de Souza, U., Eduardo, P., Ponzoni Carvalho Chanel, C., & Givigi, S. (2016). *A game theoretical formulation of a decentralized cooperative multi-agent surveillance mission*.

Di Perna, M., & Rodrigues, L. (2017). Distributed backstepping coverage control for multi-agents moving on a plane. *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, 1590–1595.

Dogru, S., & Marques, L. (2015). Energy efficient coverage path planning for autonomous mobile robots on 3D terrain. *Autonomous Robot Systems and Competitions (ICARSC), 2015 IEEE International Conference On*, 118–123.

Evans, W., & Sember, J. (2008). Guaranteed voronoi diagrams of uncertain sites. *20th Canadian Conference on Computational Geometry*, 207–210.

Gerkey, B. (2020). gmapping - ROS Wiki. Retrieved from http://wiki.ros.org/gmapping

Hu, J., & Zheng, W. X. (2014). Adaptive leader-following control of second-order multi-agent systems. *IFAC Proceedings Volumes*, *47*(3), 11691–11696.

Hung, P. D., Vinh, T. Q., & Ngo, T. D. (2016). Distributed coverage control for networked multi-robot systems in any environments. *Advanced Intelligent Mechatronics (AIM), 2016 IEEE International Conference On*, 1067–1072.

Hungerford, K., Dasgupta, P., & Guruprasad, K. R. (2016). A repartitioning algorithm to guarantee complete, non-overlapping planar coverage with multiple robots. In *Distributed Autonomous Robotic Systems* (pp. 33–48). Springer.

Imlauer, S., Mühlbacher, C., Steinbauer, G., Gspandl, S., & Reip, M. (2016). Hierarchical planning with traffic zones for a team of industrial transport robots. *4th Workshop on Distributed and Multi-Agent Planning (DMAP)*.

Janani, A., Alboul, L., & Penders, J. (2016). Multi robot cooperative area coverage, case study: spraying. *Conference Towards Autonomous Robotic Systems*, 165–176.

Jia, D., Wermelinger, M., Diethelm, R., Krüsi, P., & Hutter, M. (2016). Coverage path planning for legged robots in unknown environments. *Safety, Security, and Rescue Robotics (SSRR), 2016 IEEE International Symposium On*, 68–73.

Jin, J., Kim, Y.-G., Wee, S.-G., & Gans, N. (2015). Decentralized cooperative mean approach to collision avoidance for nonholonomic mobile robots. *Robotics and Automation (ICRA), 2015 IEEE International Conference On*, 35–41.

Kwok, A., & Martinez, S. (2007). Energy-balancing cooperative strategies for sensor deployment. *Decision and Control, 2007 46th IEEE Conference On*, 6136–6141.

Kwok, A., & Martinez, S. (2010). Deployment algorithms for a power-constrained mobile sensor network. *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, *20*(7), 745–763.

Lee, S. K., Fekete, S. P., & McLurkin, J. (2016). Structured triangulation in multi-robot systems: Coverage, patrolling, Voronoi partitions, and geodesic centers. *The International Journal of Robotics Research*, *35*(10), 1234–1260.

Lei, C., Sun, W., & Yeow, J. T. W. (2016). A distributed output regulation problem for multi-agent linear systems with application to leader-follower robot's formation control. *Control Conference (CCC), 2016 35th Chinese*, 614–619.

Li, C., Song, Y., Wang, F., Wang, Z., & Li, Y. (2016). A bounded strategy of the mobile robot coverage path planning based on lorenz chaotic system. *International Journal of Advanced Robotic Systems*, *13*(3), 107.

Lin, X., & Cassandras, C. G. (2015). An optimal control approach to the multi-agent persistent monitoring problem in two-dimensional spaces. *IEEE Transactions on Automatic Control*, *60*(6), 1659–1664.

Loizou, S. G., & Constantinou, C. C. (2016). Multi-robot coverage on dendritic topologies under communication constraints. *Decision and Control (CDC), 2016 IEEE 55th Conference On*, 43–48.

Luna, J. M., Fierro, R., Abdallah, C. T. and Wood, J. (2013). An Adaptive Coverage Control for Deployment of Nonholonomic Mobile Sensor Networks over Time-Varying Sensory Functions. *Asian Journal of Control*, *15*(4), 988–1000.

Mahboubi, H., Sharifi, F., Aghdam, A. G., & Zhang, Y. (2012). Distributed coordination of multi-agent systems for coverage problem in presence of obstacles. *American Control Conference (ACC), 2012*, 5252–5257.

Mahboubi, H., Vaezi, M., & Labeau, F. (2014). Mobile sensors deployment subject to measurement error. *Vehicular Technology Conference (VTC Fall), 2014 IEEE 80th*, 1–6.

Michael, N., & Kumar, V. (2009). Planning and control of ensembles of robots with non-holonomic constraints. *The International Journal of Robotics Research*, *28*(8), 962–975.

Mitchell, D., Chakraborty, N., Sycara, K., & Michael, N. (2015). Multi-robot persistent coverage with stochastic task costs. *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference On*, 3401–3406.

Mitchell, D., Corah, M., Chakraborty, N., Sycara, K., & Michael, N. (2015). Multi-robot long-term persistent coverage with fuel constrained robots. *Robotics and Automation (ICRA), 2015 IEEE International Conference On*, 1093–1099.

Muddu, R. S. D., Wu, D., & Wu, L. (2015). A frontier based multi-robot approach for coverage of unknown environments. *Robotics and Biomimetics (ROBIO), 2015 IEEE International Conference On*, 72–77.

Nowzari, C., & Pappas, G. J. (2016). Multi-agent coordination with asynchronous cloud access. *American Control Conference (ACC), 2016*, 4649–4654.

Okabe, A. (1992). *Spatial tessellations*. Wiley Online Library.

Olaonipekun, D., & Vaughan, J. (2015). Complete Coverage Path Planning for Flexible Parent-Child Unit Robots. *ASME 2015 Dynamic Systems and Control Conference*, V003T40A004--V003T40A004.

Palacios-Gasós, J. M., Montijano, E., Sagues, C., & Llorente, S. (2016). Multi-robot persistent coverage using branch and bound. *American Control Conference (ACC), 2016*, 5697–5702.

Papatheodorou, S., Stergiopoulos, Y., & Tzes, A. (2016). Distributed area coverage control with imprecise robot localization. *Control and Automation (MED), 2016 24th Mediterranean Conference On*, 214–219.

Perez-Imaz, H. I. A., Rezeck, P. A. F., Macharet, D. G., & Campos, M. F. M. (2016). Multi-robot 3D coverage path planning for First Responders teams. *Automation Science and Engineering (CASE), 2016 IEEE International Conference On*, 1374–1379.

Pierson, A., Figueiredo, L. C., Pimenta, L. C. A., & Schwager, M. (2015). Adapting to performance variations in multi-robot coverage. *Robotics and Automation (ICRA), 2015 IEEE International Conference On*, 415–420.

Pierson, A., Figueiredo, L. C., Pimenta, L. C. A., & Schwager, M. (2017). Adapting to sensing and actuation variations in multi-robot coverage. *The International Journal of Robotics Research*, *36*(3), 337–354.

Pierson, A., & Schwager, M. (2015). Bio-inspired non-cooperative multi-robot herding. *Robotics and Automation (ICRA), 2015 IEEE International Conference On*, 1843–1849.

Pierson, A., & Schwager, M. (2016). Adaptive inter-robot trust for robust multi-robot sensor coverage. In *Robotics Research* (pp. 167–183). Springer.

Rabbath, C. A., & Léchevin, N. (2015). Coverage with a Team of Wheeled Mobile Robots. *Journal of Intelligent & Robotic Systems*, *78*(3–4), 553–575.

Ramaithitima, R., Whitzer, M., Bhattacharya, S., & Kumar, V. (2015). Sensor coverage robot swarms using local sensing without metric information. *Robotics and Automation (ICRA), 2015 IEEE International Conference On*, 3408–3415.

Ren, W., Beard, R. W., & Atkins, E. M. (2005). A survey of consensus problems in multi-agent coordination. *American Control Conference, 2005. Proceedings of the 2005*, 1859–1864.

Romvary, J. J., & Annaswamy, A. M. (2016). Multi-agent coordination in dynamic networks. *Decision and Control (CDC), 2016 IEEE 55th Conference On*, 2802–2807.

Rycroft, C. (n.d.). Voro++ - A 3D Voronoi cell software library. Retrieved August 30, 2017, from http://math.lbl.gov/voro++/

Shnaps, I., & Rimon, E. (2015). On-line coverage of planar environments by a battery powered autonomous mobile robot. In *Algorithmic Foundations of Robotics XI* (pp. 571–589). Springer.

Slotine, J.-J. E. and Li, W. (1991). *Applied Nonlinear Control*. Prentice Hall, New Jersey.

Sun, Z., & Anderson, B. D. O. (2016). Formation feasibility on coordination control of networked heterogeneous systems with drift terms. *Decision and Control (CDC), 2016 IEEE 55th Conference On*, 3462–3467.

Surynek, P., Felner, A., Stern, R., & Boyarski, E. (2016). Efficient SAT Approach to Multi-Agent Path Finding Under the Sum of Costs Objective. *ECAI*, 810–818.

Tsumura, K., & Kawasaki, I. (2016). Optimal control/observation points problem and separation principle of weakly controlled large-scaled multi-agent systems. *Decision and Control (CDC), 2016 IEEE 55th Conference On*, 5110–5115.

Turanli, M., & Temeltas, H. (2018a). Adaptive coverage control with Guaranteed Power Voronoi Diagrams. *2017 4th International Conference on Systems and Informatics, ICSAI 2017*, *2018-Janua*, 7–13. https://doi.org/10.1109/ICSAI.2017.8248255

Turanli, M., & Temeltas, H. (2018b). Workspace Allocation for Team of Robots with Different Actuation Capabilities. *ICCR 2018*.

Turanli, M., & Temeltas, H. (2019). Multi-Robot Collaborative Coverage Under Localization Uncertainty. *2019 IEEE International Conference on Mechatronics and Automation (ICMA)*, 1999–2005.

Turanli, M., & Temeltas, H. (2020). Multi-Robot Workspace Allocation with Hopfield Networks and Imprecise Localization. *Acta Polytechnica Hungarica*, *17*(5).

Viet, H. H., Dang, V.-H., Choi, S., & Chung, T. C. (2015). BoB: an online coverage approach for multi-robot systems. *Applied Intelligence*, *42*(2), 157–173.

Wang, J., Chen, J., Cheng, S., & Xie, Y. (2016). Double Heuristic Optimization Based on Hierarchical Partitioning for Coverage Path Planning of Robot Mowers. *Computational Intelligence and Security (CIS), 2016 12th International Conference On*, 186–189.

Wang, X., Zeng, Z., & Cong, Y. (2016). Multi-agent distributed coordination control: developments and directions via graph viewpoint. *Neurocomputing*, *199*, 204–218.

Yehoshua, R., & Agmon, N. (2016). Multi-Robot Adversarial Coverage. *ECAI*, 1493–1501.

Zhang, J., Zhou, P., & Ma, L. (2016). Coverage control of multiple heterogeneous mobile robots with nonholonomic constraints. *Control Conference (CCC), 2016 35th Chinese*, 6272–6277.

APPENDICES

APPENDIX A: Communication Diagrams of ROS Nodes

APPENDIX A : Communication Diagrams of ROS Nodes

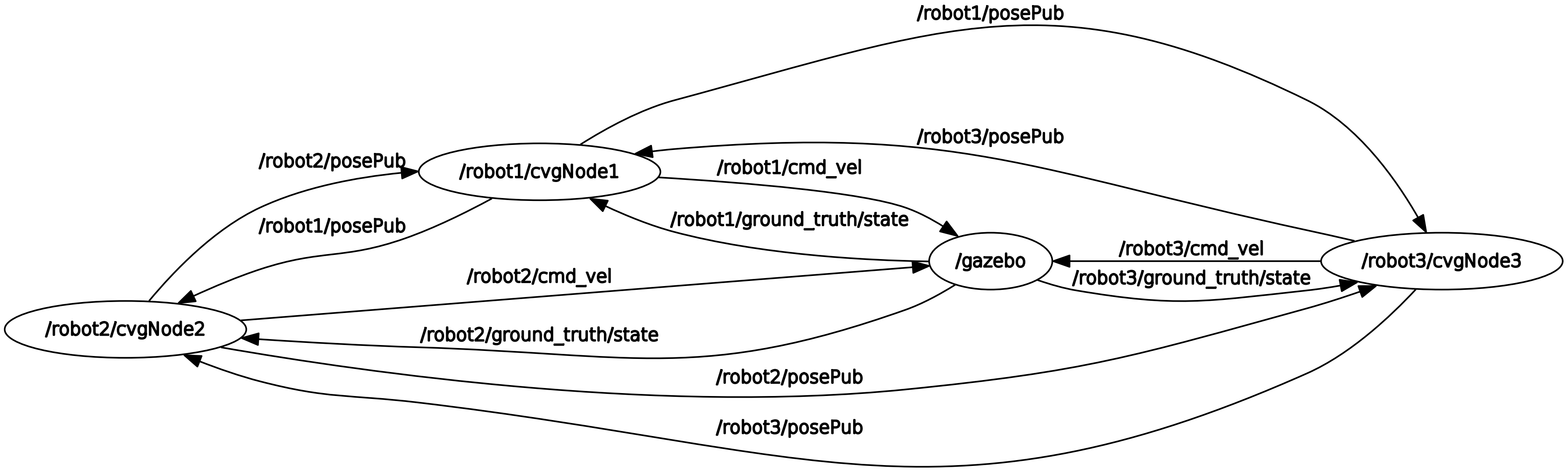

Figure A.1 : ROS Nodes and Topics for Coverage Control with Non-holonomic Estimator.

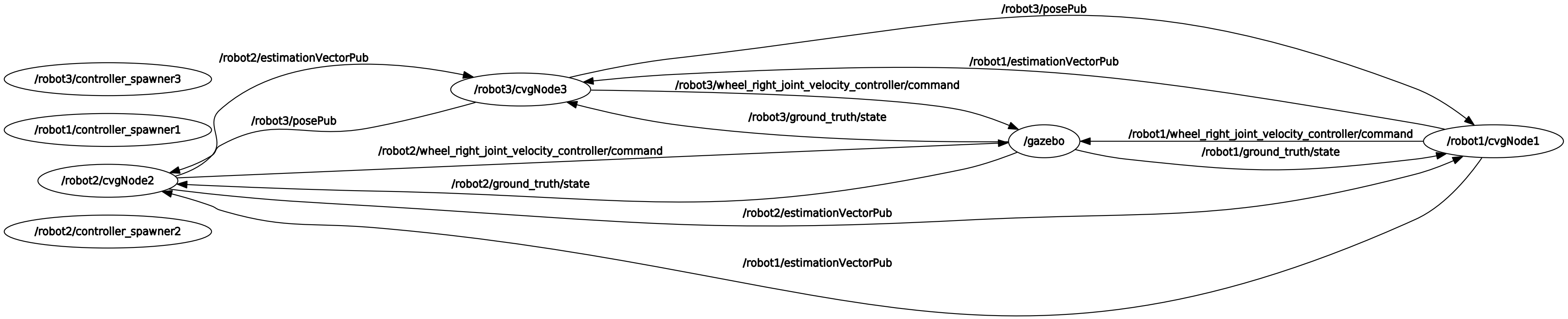
/robot3/posePub
/robot2/estimationVectorPub
/robot1/estimationVectorPub
/robot3/controller_spawner3
/robot3/cvgNode3
/robot3/wheel_right_joint_velocity_controller/command
/robot3/posePub
/robot3/ground_truth/state
/robot1/controller_spawner1
/gazebo
/robot1/wheel_right_joint_velocity_controller/command
/robot1/cvgNode1
/robot2/wheel_right_joint_velocity_controller/command
/robot1/ground_truth/state
/robot2/cvgNode2
/robot2/ground_truth/state
/robot2/estimationVectorPub
/robot2/controller_spawner2
/robot1/estimationVectorPub

Figure A.2 : ROS Nodes and Topics for Coverage Control with Hopfield Estimator.

CURRICULUM VITAE

Name Surname : Mert TURANLI

Place and Date of Birth : Istanbul, 20.01.1986

E-Mail : mertturanli@gmail.com

EDUCATION :

- **B.Sc.** : 2008, Istanbul Technical University, Faculty of Electrical and Electronics, Control Engineering Department
- **M.Sc.** : 2011, Istanbul Technical University, Institute of Science and Technology, Control and Automation Engineering Department

PROFESSIONAL EXPERIENCE AND REWARDS:

- 2008 Control and Avionics Laboratory, Faculty of Aeronautics and Astronautics, Istanbul Technical University
- 2010 – 2011 Atlantis Unmanned Vehicle Solutions Ltd.
- 2011 Completed Master of Science at Istanbul Technical University

PUBLICATIONS, PRESENTATIONS AND PATENTS ON THE THESIS:

- **Turanli, M.**, & Temeltas, H. (2020). Multi-Robot Energy-Efficient Coverage Control with Hopfield Networks. In *Studies in Informatics and Control*. Volume 29, 2020.
- **Turanli, M.**, & Temeltas, H. (2020). Multi-Robot Workspace Allocation with Hopfield Networks and Imprecise Localization. In *Acta Polytechnica Hungarica*. Volume 17, 2020.
- **Turanli, M.**, & Temeltas, H. (2019). Multi-Robot Collaborative Coverage Under Localization Uncertainty. In *IEEE International Conference on Mechatronics and Automation (ICMA), 2019*.

- **Turanli, M.**, & Temeltas, H. (2018). Workspace Allocation for Team of Robots with Different Actuation Capabilities. In *International Conference on Control and Robots (ICCR), 2018,* (pp. 11-18).
- **Turanli, M.**, & Temeltas, H. (2017). Adaptive coverage control with Guaranteed Power Voronoi Diagrams. In *Systems and Informatics (ICSAI), 2017 4th International Conference on* (pp. 7–13).

OTHER PUBLICATIONS, PRESENTATIONS AND PATENTS:

- **Turanli, M.**, & Temeltas, H. (2016). Adaptive Coverage Control with Power-Aware Control Laws and Exponential Forgetting. In *International Journal of Applied Mathematics and Informatics*. Volume 10, 2016.
- **Turanli, M.**, & Temeltas, H. (2015). Power-Aware Adaptive Coverage Control with Consensus Protocol. In *Proceedings of the 2015 International Conference on Mechanical Manufacturing and Control.*
- **Turanli, M.**, & Temeltas, H. (2011). Location Dependent Variable Information Gain Approach to Multi Robot Coverage Problem. In *Proceedings of 2011 International Conference on Mechatronics and Automation.*
- Sencan, O., **Turanli, M**., Sariyanidi, E., Temeltas, H., Kurnaz, S., Bogosyan, S. (2010). A Real-time SLAM Algorithm with Optical Flow based Motion Extraction for Autonomous Robot Navigation. In *International Unmanned Vehicles Workshop 2010.*
- **Turanli, M.**, Temeltas, H. (2009). Design of a Mobile Robot Trajectory Controller. In *TOK 2009 Turkish Automatic Control Conference*.