

# Multi-Robot Energy-Efficient Coverage Control with Hopfield Networks

Mert TURANLI\*, Hakan TEMELTAS

Institute of Science and Technology, Istanbul Technical University, Maslak, Istanbul, 34469, Turkey  
turanlim@itu.edu.tr (\*Corresponding author), hakan.temeltas@itu.edu.tr

**Abstract:** The control problem of the multi-robots with different actuation capabilities has caught the attention of the robotics researchers over the last years. The algorithm proposed in the paper not only makes use of the energy-efficient coverage optimal control scheme but also utilizes Hopfield Neural Networks (HNN) in order to perform collaboration among the agents according to their different actuation performances without knowing them beforehand. The agents estimate their own performances by using HNN estimator and calculate the Power Voronoi Diagram (PVD or PD) weights for each agent. The robots achieve the optimal coverage configuration autonomously. The algorithm can assign the regions from the workspace and in resultant configuration, the robots with strong actuators obtain wider areas allocated from the workspace while the weaker robots obtain smaller areas. Also, by making use of the energy-efficient optimal coverage control scheme, a trade-off between the coverage time and energy consumption of the agents can be made. The MATLAB simulation and experimental results regarding the proposed online and distributed algorithm are given. The novelty of this work lies in the using of HNN-based multi-robot coverage collaboration according to the different actuation capabilities and energy-efficient coverage control laws at the same time. Significant improvements compared with the method utilized in (Pierson et al., 2017) have been observed.

**Keywords:** Energy efficient, Coverage control, Power Voronoi diagrams, Hopfield neural networks.

## 1. Introduction

Multi-robot coordination and collaboration problems are challenging topics which have been studied intensively over the last decades. Many problems such as search and rescue, patrolling, humanitarian demining and surveillance can be efficiently solved when employing more agents.

In the specialized literature there are many recent papers related to the coverage topic. In the research paper (Tardós et al., 2018), a multi-robot team consisting of ground and aerial vehicles performs area monitoring task. The ground robots follow aerial robots which are the targets. A Lloyd-based strategy is used. Simulation and experimental results are given. The work (Pierson et al., 2017) proposes an algorithm which considers actuation and sensing performance parameters in a coverage collaboration algorithm. The algorithm automatically compensates the variations among the robots by using the online method and by learning the performance variations without prior knowledge. Experimental results are given.

The Hopfield Neural Networks are also widely used for the robotic systems. In the paper (Atencia, Joya & Sandoval, 2004), an online identification method for non-linear systems with Hopfield networks is proposed. In the case of constant parameters and time-varying parameters to a bounded neighbourhood, the convergence is successful. The proposed method is used in the identification of a robotic system with a flexible link. Simulation results are given. In

(Alonso, Mendonça & Rocha, 2009), the online parameter estimation with Hopfield networks is proposed for a robotic system. The stability and robustness analysis are given, and the results are experimentally verified. In order to mention about the control and estimation by using neural networks, the work (Jin et al., 2018) gives a review about the utilization of neural network controllers in the literature. In the work (Chen et al., 2020b), a method for simultaneous identification, tracking control and disturbance rejection by using Zeroing Neural Networks is proposed with applications to robot manipulators.

In the paper (Das & Ghose, 2015), a new approach for multi-agent consensus problem is introduced. The authors propose a linear programming based efficient method including an experimental verification with three robots. The work (Su & Lin, 2015) proposes a new distributed consensus control approach for general higher order linear systems. In the work (Liu et al., 2015), a synchronization analysis of a class of multi-agent systems is presented by giving conditions for synchronization. The work (Dong et al., 2016) gives a study of time-varying formation tracking of second order multi-agent systems. They show numerical simulation and experimental results taken with one leader and three follower quadrotors. The paper (Zhao et al., 2017) proposes an general approach to construct coordination control laws for multi-agent systems under motion constraints.

In the work (Alonso-Mora, Beardsley & Siegwart, 2018), a dynamical collision avoidance approach among the agents is proposed. The distributed and centralized variants of the method are also introduced. The work (Zhang et al., 2018), a cooperative localization method among air and ground robots is presented. The aerial vehicles map the environment and share the map with the ground vehicles. The ground vehicles estimate their pose with respect to aerial vehicles. The work (Sartoretti et al., 2019) introduces a method that uses the asynchronous advantage of the actor-critic (A3C) algorithm for learning decentralized control policies. The method is utilized for single articulated robots. The paper (Lin & Ling, 2020) gives the results of a proposed method about a model based event triggering approach with dynamical quantization for multi-agent systems.

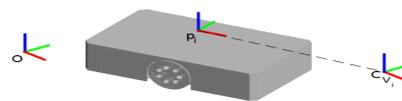
In the paper (Chen et al., 2020a), the authors propose a new method for finding unexpected faults on communication channels of multi-agents systems to improve the synchronization between the agents. A novel broadcast control law for multi-agent coordination problem is developed in (Ito et al., 2020) where a low communication volume is utilized for global coordination. The paper (Lin et al., 2017) presents a new method for multi-autonomous underwater vehicles utilized for tracking the targets in the sea autonomously and in coordination. The paper (Castaño et al., 2019) presents a cooperative method for multiple unmanned aerial vehicles as a part of a competition challenge where the vehicles search, pick up and drop the objects in the environment.

The contribution of the paper is that the collaboration is performed among the agents according to their actuation performances (Pierson et al., 2017) in which the Hopfield Networks (Alonso, Mendonça & Rocha, 2009) are utilized as parameter estimator and energy-efficient optimal coverage control (Di Perna & Rodrigues, 2017) laws which enable the algorithm to make a trade-off between the energy consumption of the agents and the coverage time. This is the first work in the research literature that uses multi-robot coverage collaboration according to the different actuation capabilities and energy-efficient coverage control laws at the same time. Also, the utilization of Hopfield Networks provides a better convergence rate and disturbance rejection capability in the collaboration algorithm compared to the method utilized in (Pierson et al., 2017).

The paper is structured as follows. The first section gives the introduction. The second part presents the preliminary information about the problem. The next section presents the coverage collaboration algorithm. In the fourth section, Lyapunov-type stability analysis is presented, and the necessary proofs are provided. Section five presents the simulation results obtained by employing MATLAB. In the next section, the experimental results obtained by using ROS are given. The last section summarizes the conclusions of the paper.

## 2. Problem Formulation

Let us consider  $m$  non-holonomic mobile robots spread over an environment. The aim is to drive the agents to the optimal coverage configuration by compensating their different actuation capabilities. The objective is to perform collaboration by assigning wider areas to the robots with stronger actuation capabilities and smaller areas to the ones with weaker actuator performances. This goal is achieved by estimating the performance parameters online with HNNs and then calculating the weights for the Power Diagram. The calculated weights are passed into the PD so that the regions are assigned to the agents according to the performances. After calculating the dynamic Power Voronoi regions and the centroids of the agents, the non-holonomic control law drives the robots to the optimal configurations.



**Figure 1.** The robot position  $p_i$  and centroid  $C_{v_i}$  courtesy of (Turanli & Temeltas, 2019)

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

### 2.1 Power Voronoi diagrams

Let the space be defined as  $S \subseteq \mathbb{R}^N$  which is a bounded environment. Here,  $N$  is the number of dimensions. A Power Voronoi region  $W_i$  can be given as below (Okabe, 1992):

$$W_i = \{p \in S \mid \|p - p_i\|^2 - w_i \leq \|p - p_j\|^2 - w_j, i \neq j, j = 1, 2, \dots, m\} \quad (1)$$

Here,  $w_i$  and  $w_j$  are Power Diagram weights used for changing the area of the corresponding regions and  $p_i$  is the position of the  $i^{th}$  agent. In this work, Voro++ library (Rycroft, n.d.) was used for drawing the power diagrams.

## 2.2 Energy-efficient Coverage Optimal Control

The energy-efficient coverage optimal control provides a trade-off between the energy consumption of the agents and the coverage time by adjusting two parameters in the control law.

The optimal coverage control problem can be described by the cost function minimized by using an energy-efficient optimal control law derived analytically (Di Perna & Rodrigues, 2017). The cost function can be given as below:

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

Here,  $m$  is the number of the agents,  $s_i$  and  $r_i$  denote the weighting coefficients for coverage and energy terms, respectively. Also,  $p_i$  gives the position of the  $i^{th}$  agent and  $u_i$  is the control inputs of the  $i^{th}$  agent according to the holonomic model given in (6).

As the robot and the centroid positions illustrated in Figure 1, the mass  $M_{w_i}$  and centroid  $C_{w_i}$  of the Voronoi region  $i$  can be shown as follows (Luna et al., 2013):

$$F_i(h, q) = \int_{w_i} h dq \quad (3)$$

$$M_{w_i} = F_i(\phi(q), q) \quad (4)$$

$$C_{w_i} = \frac{1}{M_{w_i}} F_i(q\phi(q), q) \quad (5)$$

The system model and the energy-efficient optimal coverage control law are defined as shown below where  $p_i$  is the position vector of the agent  $i$ :

$$\dot{p}_i = u_i \quad (6)$$

$$u_i = \sqrt{\frac{s_i}{r_i}} M_{w_i} (C_{w_i} - p_i) \quad (7)$$

Equation (7) presents the control law obtained by minimizing the cost function in (2) for the agent  $i$ .

## 2.3 Hopfield Neural Networks

The Hopfield Neural Network is an online parameter estimation technique utilized to estimate the actuator performance parameters as proposed in (Alonso, Mendonça & Rocha, 2009). The HNN is a non-autonomous non-linear dynamical system which provides the time-evolving estimation of the system in continuous time. The parameterization of the system is given. The approach is chosen since limitations in accuracy can occur in other neural networks based approaches such as classification-based methods.

Consider a Hopfield-network consisting of  $M$  neurons. Here,  $x_i$  denotes the total input of the neuron  $i$ ,  $s_j$  gives the output of the neuron  $j$ ,  $W_{ij}$  shows the weight corresponding to the connection from neuron  $j$  to neuron  $i$  and  $I_i$  represents the bias of the neuron  $i$ .

$$\frac{dx}{dt} = -(W(t)s(t) + I(t)) \quad (8)$$

$$s(t) = \alpha \tanh\left(\frac{x(t)}{\beta}\right) \quad (9)$$

The matrix representation of the HNN is given in (8) and (9) where  $\alpha, \beta > 0$ .

## 3. Multi-Agent Coverage Control with Different Actuation Capabilities

This part gives the parameter estimation with HNNs and estimation of the PD weights.

Here, the variations in the actuator performances of the individual agents are defined. The presence of strong motors and good tire contacts can be considered strong actuation performances. On the other hand, weak motors and slippery terrain can result in weak actuation performances.

In this work, the point offset controller from (Michael & Kumar, 2009), (Pierson et al., 2017) is used to transform the holonomic control inputs into the non-holonomic linear ( $v$ ) and angular ( $\omega$ ) velocities with respect to  $P$ . Here, the variable  $\theta$  gives the yaw angle and  $P$  is the reference point with a distance  $l$  to the center of the robot.

$$\begin{pmatrix} v \\ \omega \end{pmatrix} = \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix} \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} \quad (10)$$

The transformation in (10) with the control law given in (7) enables the agents to perform the autonomous operation. So, the agents can be driven to the centroid locations in order to complete the energy-efficient collaborative coverage task.

### 3.1 Estimating the Performance Parameters with HNNs

Each agent learns its own actuation performance vector which is not known beforehand in order to perform the collaboration. For this purpose, the system parametrization for the HNN algorithm will be derived.

The system in (6) is used together with the control law in (7) to estimate the parameters of the agent  $i$ . For this purpose, the system is written in linear in the parameters (LIPs) form. The closed-loop system equation is:

$$\dot{p}_i = \sqrt{\frac{S_i}{r_i}} M_{w_i} (C_{w_i} - p_i) = K_i M_{w_i} (C_{w_i} - p_i) \quad (11)$$

The velocity vector  $\dot{p}_i$  can be converted into body coordinates by using the transformation where  $R_{z,\theta}$  is the rotation about  $z$  axis and  $\dot{p}_{i,b}$  is the body velocities of the agent  $i$ :

$$\dot{p}_{i,b} = \begin{pmatrix} \dot{x}_{i,b} \\ \dot{y}_{i,b} \end{pmatrix} = R_{z,\theta}^T \sqrt{\frac{S_i}{r_i}} M_{w_i} (C_{w_i} - p_i) \quad (12)$$

Taking the first row yields the equation below since the lateral body velocity  $\dot{y}_{i,b}$  is equal to zero:

$$-\dot{x}_{i,b} \sqrt{\frac{r_i}{S_i}} = (\cos(\theta) \quad \sin(\theta)) \begin{pmatrix} p_{i,x} - C_{w_i,x} \\ p_{i,y} - C_{w_i,y} \end{pmatrix} M_{w_i} \quad (13)$$

The LIP form can be written as below:

$$y = (\cos(\theta) \quad \sin(\theta)) \begin{pmatrix} p_{i,x} - C_{w_i,x} \\ p_{i,y} - C_{w_i,y} \end{pmatrix} M_{w_i} \quad (14)$$

$$A = -\dot{x}_{i,b} \quad (15)$$

$$\theta_{est} = \sqrt{\frac{r_i}{S_i}} \quad (16)$$

$$y = A \theta_{est} \quad (17)$$

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

$$W = A^T A \quad (18)$$

$$I = -A^T y \quad (19)$$

Then, the estimated parameter vector  $\hat{K}_i$  is obtained by using the estimated parameter  $\hat{\theta}_{est}$  as given in the following equations:

$$\hat{\theta}_{est} = s(t) \quad (20)$$

$$\hat{K}_i = \frac{1}{\hat{\theta}_{est}} \quad (21)$$

By using (8), (9) along with (14), (15), (18) and (19), the HNN estimates the parameters of the agent online whose closed-loop dynamics are given in (11). Then, the resulting parameters can be obtained by using equations (20) and (21).

The implementation of equation (8) can be done by using numerical integrators.

### 3.2 Estimating the PD Weights of the Agents

The Power diagram is utilized to partition the workspace and adjust the regions of individual agents with respect to their actuation performance vectors. This section will present the estimator equations based on (Pierson et al., 2017) which are used to calculate the PD weights from the actuation performance estimation vector  $\hat{K}_i$ .

After  $\hat{K}_i$  is calculated by using (20) and (21), the values of the parameter vector  $\hat{K}_i$  are passed into the weight estimator. The output of the weight estimator is given to the PD algorithm.

The adaptation law (weight estimator) is given as below:

$$\kappa_i = w_i - f_{perf}(\hat{K}_i) \quad (22)$$

$$\dot{w}_i = -k_\omega \sum_{j \in N_i} (\kappa_i - \kappa_j) \quad (23)$$

where  $k_\omega$  denotes a positive coefficient,  $w_i$  gives the weight of the agent  $i$  and the neighbours are represented by the variable  $N_i$ . The desired performance function  $f_{perf}(\hat{K}_i)$  is taken as  $f_{perf}(\hat{K}_i) = \|\hat{K}_i\|$  in the simulations and experiments.

The communication among the robots is done only with the PD neighbours.

After  $\hat{K}_i$  is calculated for the agent  $i$ , the PD weights are estimated using equations (22) and (23). In the implementation, the integrations are converted into numerical trapezoidal integrations.

## 4. Stability Analysis

This part of the paper provides the Lyapunov-type proof for the control and estimation laws. First, the definition of a Lyapunov function candidate is given as shown below for  $V_3 : \theta_{est} + (-c, c)^M \rightarrow \mathbb{R}$  where  $c$  is a positive constant:

$$V = \sum_{i=1}^3 V_i \quad (24)$$

$$V_1 = \sum_{i=1}^m \frac{1}{2} \|C_{w_i} - p_i\|^2 \quad (25)$$

$$V_2 = \sum_{i=1}^m w_i \quad (26)$$

$$f_{1,j} = (1 + \frac{\tilde{\theta}_{est,i,j}}{c - \theta_{est,i,j}})^{(c - \theta_{est,i,j})} \quad (27)$$

$$f_{2,j} = (1 - \frac{\tilde{\theta}_{est,i,j}}{c + \theta_{est,i,j}})^{(c + \theta_{est,i,j})} \quad (28)$$

$$V_3 = \sum_{i=1}^m (-\frac{1}{2c} \sum_{j=1}^M \ln(f_{1,j} f_{2,j})) \quad (29)$$

where  $\tilde{\theta}_{est,i,j} = \theta_{est,i,j} - \hat{\theta}_{est,i,j}$ . For the parameter estimator of the  $i^{th}$  agent, the variable  $\hat{\theta}_{est,i,j}$  gives the output of the  $j^{th}$  neuron.

Next, the state vector is defined as  $x_i = (C_{w_i} - p_i \quad w_i \quad \tilde{\theta}_{est,i})^T$ . It can be easily noticed that  $V(x=0) = 0$ . In order to investigate the positive definiteness of  $V(x)$ , the positive definiteness of the  $V_1(x)$ ,  $V_2(x)$  and  $V_3(x)$  should be proven. The  $V_1(x)$  is clearly positive definite. The  $V_2(x)$  is positive definite since the weights of the agents are positive. Also, the  $V_3(x)$  is guaranteed to be positive definite as proven in (Alonso, Mendonça & Rocha, 2009). So, it can be concluded that  $V(x) > 0$ .

The time derivative of the Lyapunov function yields the first term:

$$\dot{V}_1 = \sum_{i=1}^m -(C_{w_i} - p_i)^T \sqrt{\frac{s_i}{r_i}} M_{w_i} (C_{w_i} - p_i) \leq 0 \quad (30)$$

The first term  $\dot{V}_1$  is negative semi-definite since  $\sqrt{\frac{s_i}{r_i}}$  and  $M_{w_i}$  are positive.

Also, the second and third terms are shown to be negative semi-definite as demonstrated in (Turanli & Temeltas, 2019). The details are omitted here. Theorem 1 from (Turanli & Temeltas, 2019) shows that for an interval  $t \in I$  which is non-degenerate, the  $\tilde{\theta}_{est,i}^* = 0$  is a unique equilibrium point.

*Corollary 1:*  $w_i - f_{perf}(K_i)$  has the same value for all agents in the steady-state (Pierson et al., 2017) as shown in (32):

$$\lim_{t \rightarrow \infty} (\hat{K}_i) = K_i \quad (31)$$

$$\lim_{t \rightarrow \infty} (w_i - w_j) = f_{perf}(K_i) - f_{perf}(K_j) \quad (32)$$

The LaSalle's Invariance Principle will be used for the proof. It can be concluded that  $\dot{V} = 0$  occurs only when  $C_{w_i} = p_i$  and  $\tilde{\theta}_{est,i} = 0$ . The equilibrium points were shown to be unique. If the system reaches the given equilibrium points the tracking and estimation errors converge to zero. The result is that this set is an invariant set when the control and estimation laws are considered. Thus, the system is shown to be globally asymptotically stable.

## 5. Simulations

In this section, MATLAB simulations are given with a map of 5x5 meters. The parameters used in the simulations are given as  $k_\omega = 0.2$ ,  $r_{robot} = 0.11$ ,  $l = 0.1$ ,  $\alpha = 6000$  and  $\beta = 1$ . Also, the vertices of the map are given as (0.625,0), (3.125,0), (5.0,4.375), (3.75,5.0) and (0,4.375).

The parameter  $k_\omega$  gives the convergence rate of the weight estimator. Also,  $\alpha$  and  $\beta$  determine the convergence rate of the Hopfield Network estimator.

In the first simulation performed with 6 agents, the actuation performance of the first agent has a performance degradation of 10 percent. In Figure 2, the coverage cost is illustrated. The coverage cost settles to its local minima in the end configuration. Figure 3 depicts the HNN parameter estimation errors. Asymptotical convergence to zero is achieved for all agents as given in the stability proof. In Figure 4 the value of  $w_i - f_{perf}(\hat{K}_i)$  is the same value among all the agents as given in Corollary 1.

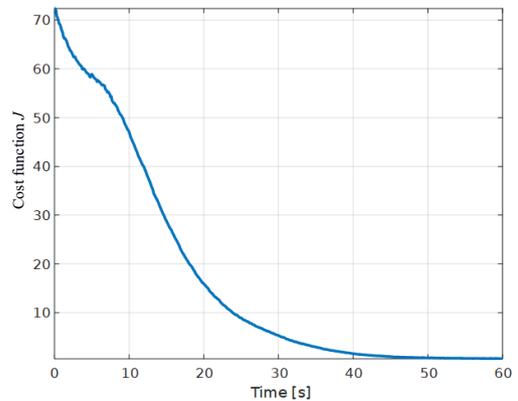
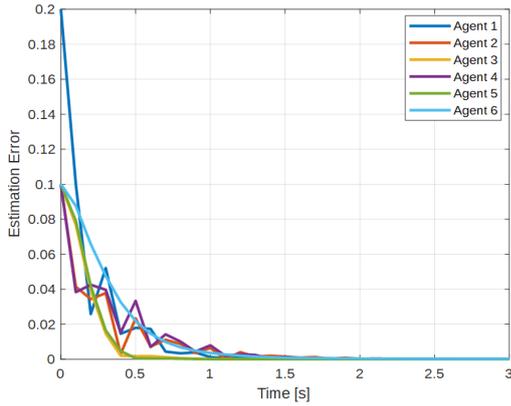


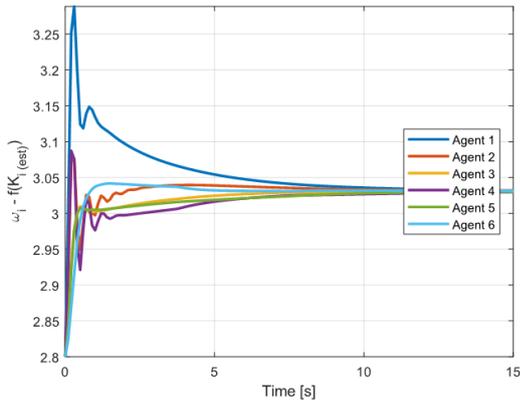
Figure 2. The value of cost function  $J$



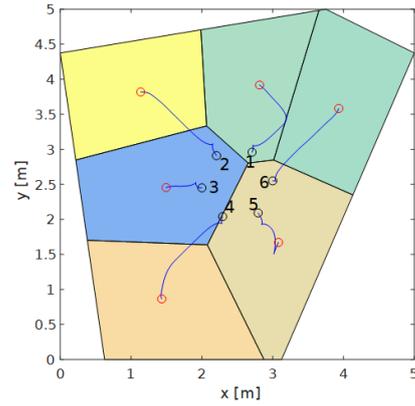
**Figure 3.** The parameter estimation errors

In Figure 5, the trajectories of the agents with the PD regions at the end configurations are illustrated. The red circles show the end configurations while the initial configurations are represented by the black markers.

In Table 1, three simulation results are summarized. In the first simulation, the value of parameters  $s_i / r_i$  has been taken as 1.0 and actuation performances have been kept at 1.0 (100%). As the  $r_i$  parameter increases in second and third simulations, the energy consumption is reduced accordingly, and the coverage time increases.



**Figure 4.** The value of  $w_i - f_{perf}(\hat{K}_i)$



**Figure 5.** The trajectories of the robots with Power Voronoi regions

In Table 2, the first, fourth and fifth simulation results are given. In the first simulation, the value of the parameters  $s_i / r_i$  has been taken as 1.0 and actuation performances have been kept at 1.0, similarly. As  $s_i$  parameter decreases in fourth and fifth simulations, the energy consumption is reduced accordingly, and the coverage time increases.

Table 3 shows the resulting region ratios in simulations 1, 6 and 7. In the first simulation, all the values of the performance parameters have been taken as 1.0 which corresponds to 100% actuation capability. In the second simulation, the first agent has had a performance degradation of 10% and in the third simulation, the performance of the second agent has been decreased by 20%. As seen in Table 3, in the sixth simulation the first agent has a region ratio lower than that of all the other agents. Similarly, in the seventh simulation the second agent has a region ratio which is lower than all the other agents. The region ratios are represented so that the value 1.0 corresponds to an area of 100% from the workspace. Note that as seen in Figure 5, the convex shape of the environment leads to empty regions in the workspace.

Successful results have been obtained with the MATLAB simulations obeying the stability analysis.

**Table 1.** Change of energy consumption with respect to  $r_i$

Simulation #	$s_i / r_i$	Actuation Performances	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	1.0 / 2.0	123.5	0.1954
3	1.0 / 4.0	All are the same (1.0)	168	0.1378

**Table 2.** Change of energy consumption with respect to  $s_i$ 

Simulation #	$s_i / r_i$	Actuation Performances	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 3.** Change of region ratios with respect to actuation performances of the agents

Simulation #	$s_i / r_i$	Actuation Performances	Region Ratios
6	1.0 / 1.0	0.9, 1.0, 1.0, 1.0, 1.0, 1.0	0.0935, 0.1113, 0.1175, 0.1280, 0.1313, 0.1181
7	1.0 / 1.0	1.0, 0.8, 1.0, 1.0, 1.0, 1.0	0.1512, 0.0585, 0.1570, 0.1358, 0.1021, 0.0970

## 6. Experiments

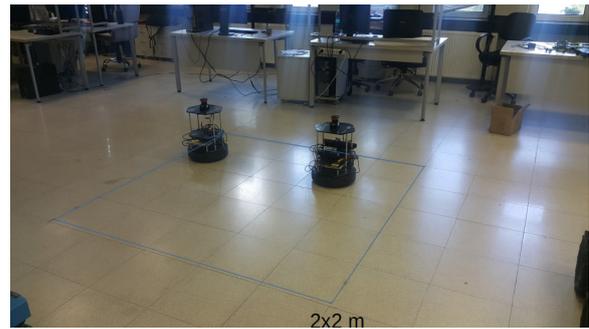
The ground experiments have been conducted in ITU Robotics Laboratory with two Turtlebot 2 robots. First, a simulation has been performed in the ROS/Gazebo multi-robot simulator and the same ROS node has been used in experiments. Second, seven experiments have been performed in the laboratory environment. The robots have been placed at known initial locations and the ROS nodes have been started remotely with a start service at the same time. The data have been collected remotely by using rosbag and the bag files have been processed by using a MATLAB script.

The ROS implementation package consists of a coverage ROS node written in C++. The agents communicate with each other by using the Power Diagram neighbours.

The algorithm in the coverage ROS node starts by calculating Power Voronoi regions by using initial positions and weight values. Then, the HNN algorithm estimates the parameter vector of each agent. The weight estimator calculates the Power Voronoi weights by using the parameter vector and the desired performance function. After, the calculated weights are utilized in the Power Diagram to assign the regions to the agents. By using the Power Diagram, the centroid locations are found, and each robot calculates its own control input by using the control law. So, the robots achieve the optimal coverage configuration by running the algorithm online and in a decentralized way. In the end, the regions are assigned to the agents according to their actuation performances.

In the experiments, the two Turtlebot 2 robots as shown in Figure 6 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 has been used in conjunction with Hokuyo UTM-30LX laser scanners and wheel odometry.

**Figure 6.** The environment in which the experiments were performed

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

In the following paragraphs, the results regarding the ROS experiments are shown. The first result set corresponds to the first experiment in which the performance of the first agent has been degraded and the resultant plots regarding the algorithm are depicted. Then, the energy-efficient optimal controller parameters have been changed and the coverage time along with the energy consumptions have been investigated. Finally, the actuation performances of the first and second agents have been degraded by a certain amount and the results are shown.

The energy-efficient coverage cost function is minimized as shown in Figure 7. In Figure 8, asymptotical convergence for the parameter estimation errors is achieved. Figure 9 illustrates the value of  $w_i - f_{perf}$  which has the same value among the two agents as given in Corollary 1.

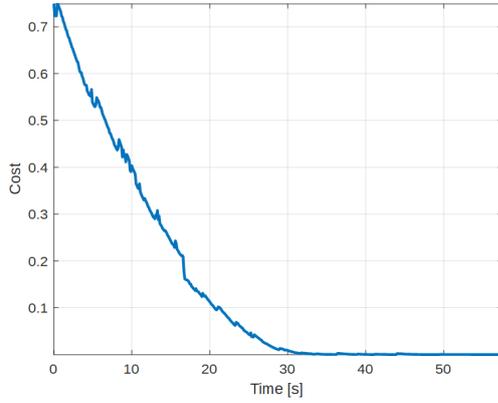


Figure 7. The value of cost function  $J$

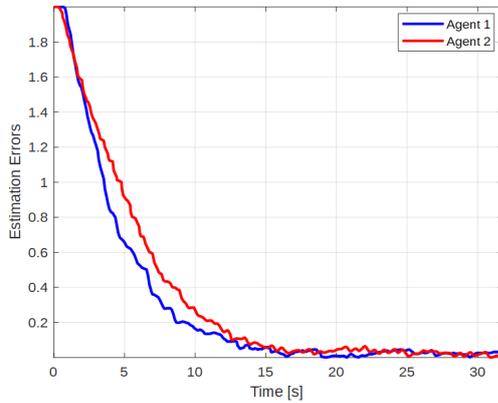


Figure 8. The parameter estimation errors

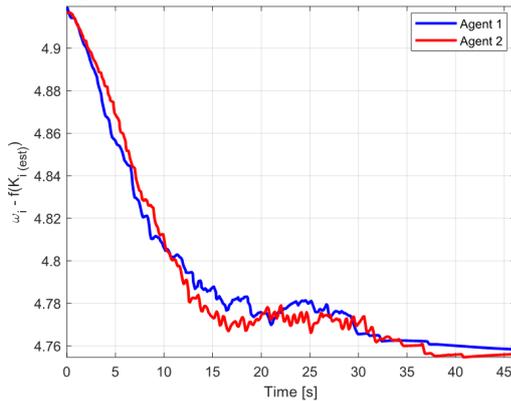


Figure 9. The value of  $w_i - f_{perf}(\hat{K}_i)$

The weights of the agents with respect to time are shown in Figure 10. A lower weight value for the first agent is obtained since it has an actuation performance degradation of 10%. The algorithm has automatically compensated the performance degradation of the first agent relatively by assigning a lower weight value and area from the workspace to the agent one.

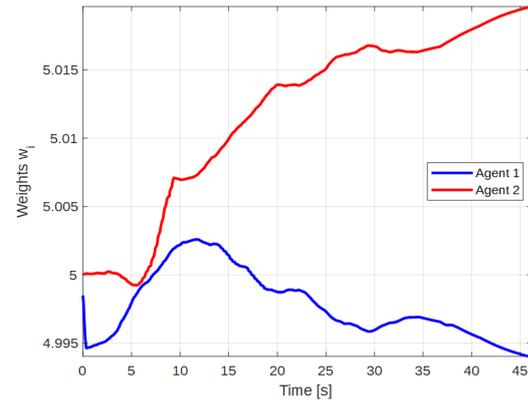


Figure 10. The weights of the agents

In Figure 11, the trajectories of the agents in the experiment are shown. Each agent calculates its centroid location from its Power Voronoi region and executes the point-offset control law.

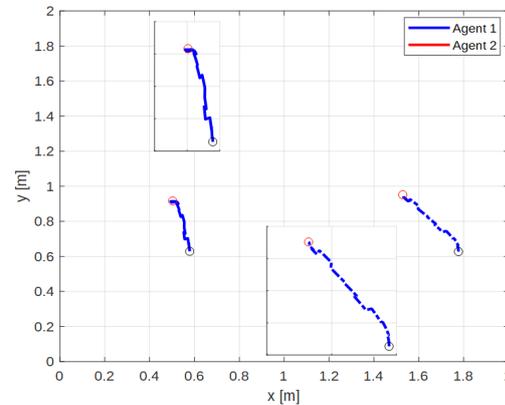


Figure 11. The trajectories of the agents

In Table 4, the coverage time of the agents increases as the parameter  $r_i$  increases. Meanwhile, the energy consumption decreases since the parameter  $r_i$  affects the energy consumption in the coverage optimization formulation.

In Table 5, the coverage time increases with the decrease of the parameter  $s_i$ . At the same time, the energy consumption decreases. In Table 6, as the actuation performance degrades, the resultant region ratio corresponding to agent 1 decreases. The reason is that the performance of the first agent has been degraded in experiments 6 and 7 by 10% and 20%, respectively.

ROS/Gazebo simulation video can be seen at:

- <https://web.itu.edu.tr/turanlim/video/sim-20200221.mp4>

Two videos of the experiments can be seen at:

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

**Table 4.** 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 5.** 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 6.** 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

The experiments show that the algorithm is capable of performing energy-efficient optimal coverage and collaboration among the agents. The energy consumption of the agents can be adjusted by making a trade-off between energy consumption and the coverage time. At the same time, the algorithm can assign the regions of the robots by learning their own actuation performances by giving larger regions to the stronger robots, and smaller regions to the weaker ones. The HNN algorithm can perform a fast and robust estimation which can be adjusted with its convergence parameter.

When compared to the method utilized in (Pierson et al., 2017), the proposed method has a faster convergence rate which is about 20 seconds at best while the compared method has converged in more than 100 seconds in the experiments. The convergence rate depends only on the velocities of the robots and convergence rate parameter. In the method utilized in the compared paper, it is not possible to adjust the convergence rate by using parameters in the proposed method.

## REFERENCES

- Alonso-Mora, J., Beardsley, P. & Siegwart, R. (2018). Cooperative collision avoidance for nonholonomic robots, *IEEE Transactions on Robotics*, 34(2), 404–420.
- Alonso, H., Mendonça, T. & Rocha, P. (2009). Hopfield neural networks for on-line parameter estimation, *Neural Networks*, 22(4), 450–462.
- Atencia, M., Joya, G. & Sandoval, F. (2004). Parametric identification of robotic systems with stable time-varying Hopfield networks, *Neural Computing & Applications*, 13(4), 270–280.
- Castaño, Á. R., Real, F., Ramón-Soria, P., Capitán, J., Vega, V., Arrue, B. C., Torres-González, A. & Ollero, A. (2019). Al-Robotics team: A cooperative multi-

## 7. Conclusion

The paper presents the results of an energy-efficient multi-agent collaboration method. The proposed algorithm automatically compensates different actuation capabilities of the agents by changing their regions. The performances of the agents are not known beforehand.

The simulations were conducted in MATLAB and ROS/Gazebo environments. Also, the experiments were performed in ITU Robotics Laboratory with two Turtlebot 2 agents equipped with laser scanners. The localization information was obtained from the ROS gmapping algorithm. The ROS nodes run real-time on the laptop computers of the robots.

The results of the energy-efficient coverage control are shown along with the results of the main collaboration algorithm. Simulation and experimental results show the efficiency of the proposed method. The HNN proves better performance when compared to the method utilized in (Pierson et al., 2017) with an adjustable and better convergence rate.

- unmanned aerial vehicle approach for the Mohamed Bin Zayed International Robotic Challenge, *Journal of Field Robotics*, 36(1), 104–124.
- Chen, C., Xie, K., Lewis, F. L., Xie, S. & Fierro, R. (2020a). Adaptive synchronization of multi-agent systems with resilience to communication link faults, *Automatica*, 111, 108636. DOI: 10.1016/j.automatica.2019.108636.
- Chen, D., Li, S., Wu, Q. & Liao, L. (2020b). Simultaneous identification, tracking control and disturbance rejection of uncertain nonlinear dynamics systems: A unified neural approach, *Neurocomputing*, 381, 282–297.
- Das, K. & Ghose, D. (2015). Broadcast control mechanism for positional consensus in multiagent systems, *IEEE Transactions on Control Systems Technology*, 23(5), 1807–1826.
- Dong, X., Zhou, Y., Ren, Z. & Zhong, Y. (2016). Time-varying formation tracking for second-order multi-agent systems subjected to switching topologies with application to quadrotor formation flying, *IEEE Transactions on Industrial Electronics*, 64(6), 5014–5024.
- Ito, Y., Kamal, M. A. S., Yoshimura, T. & Azuma, S. (2020). Pseudo-perturbation-based broadcast control of multi-agent systems, *Automatica*, 113, 108769.
- Jin, L., Li, S., Yu, J. & He, J. (2018). Robot manipulator control using neural networks: A survey, *Neurocomputing*, 285, 23–34.
- Lin, N. & Ling, Q. (2020). Bit-Rate Conditions for the Consensus of Quantized Multiagent Systems Based on Event Triggering, *IEEE transactions on cybernetics*. DOI: 10.1109/TCYB.2020.2977969
- Lin, Y., Hsiung, J., Piersall, R., White, C., Lowe, C. G. & Clark, C. M. (2017). A Multi-Autonomous Underwater Vehicle System for Autonomous Tracking of Marine Life, *Journal of Field Robotics*, 34(4), 757–774.
- Liu, Z., Li, D., Wang, L. & Dong, D. (2015). Synchronization of a group of mobile agents with variable speeds over proximity nets, *IEEE transactions on cybernetics*, 46(7), 1579–1590.
- Luna, J. M., Fierro, R., Abdallah, C. T. & 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.
- 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.
- Okabe, A. (1992). *Spatial tessellations*. Wiley Online Library.
- Di Perna, M. & Rodrigues, L. (2017). Distributed backstepping coverage control for multi-agents moving on a plane. In *2017 IEEE 56<sup>th</sup> Annual Conference on Decision and Control (CDC)*, (pp. 1590–1595).
- 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.
- Rycroft, C. (n.d.) *Voro++ - A 3D Voronoi cell software library* [Online]. Available at: <<http://math.lbl.gov/voro++/>>. Last accessed: 30 August 2017.
- Sartoretti, G., Paivine, W., Shi, Y., Wu, Y. & Choset, H. (2019). Distributed Learning of Decentralized Control Policies for Articulated Mobile Robots, *IEEE Transactions on Robotics*, 35(5), 1109–1122.
- Su, S. & Lin, Z. (2015). Distributed consensus control of multi-agent systems with higher order agent dynamics and dynamically changing directed interaction topologies, *IEEE Transactions on Automatic Control*, 61(2), 515–519.
- Tardós, J., Aragues, R., Sagüés, C. & Rubio, C. (2018). Simultaneous deployment and tracking multi-robot strategies with connectivity maintenance, *Sensors*, 18(3), 927.
- Turanli, M. & Temeltas, H. (2019). Multi-Robot Collaborative Coverage Under Localization Uncertainty. In *2019 IEEE International Conference on Mechatronics and Automation (ICMA)*, (pp. 1999–2005).
- Zhang, J., Liu, R., Yin, K., Wang, Z., Gui, M. & Chen, S. (2018). Intelligent collaborative localization among air-ground robots for industrial environment perception, *IEEE Transactions on Industrial Electronics*, 66(12), 9673–9681.
- Zhao, S., Dimarogonas, D. V., Sun, Z. & Bauso, D. (2017). A general approach to coordination control of mobile agents with motion constraints, *IEEE Transactions on Automatic Control*, 63(5), 1509–1516.