electronics

Article

Intelligent Multi-Robot System for Collaborative Object Transportation Tasks in Rough Terrains

Daegyun Choi and Donghoon Kim *

Department of Aerospace Engineering & Engineering Mechanics, University of Cincinnati, Cincinnati, OH 45221, USA; choidg@mail.uc.edu
* Correspondence: Donghoon.Kim@uc.edu

Abstract: Human missions on other planets require constructing outposts and infrastructures, and one may need to consider relocating such large objects according to changes in mission spots. A multi-robot system would be a good option for such a transportation task because it can carry massive objects and provide better system reliability and redundancy when compared to a single robot system. This paper proposes an intelligent and decentralized approach for the multi-robot system using a genetic fuzzy system to perform an object transportation mission that not only minimizes the total travel distance of the multi-robot system but also guarantees the stability of the whole system in a rough terrain environment. The proposed fuzzy inference system determines the multi-robot system's input for transporting an object to a target position and is tuned in the training process by a genetic algorithm with an artificially generated structured environment employing multiple scenarios. It validates the optimality of the proposed approach by comparing the training results with the results obtained by solving the formulated optimal control problem subject to path inequality constraints. It highlights the performance of the proposed approach by applying the trained fuzzy inference systems to operate the multi-robot system in unstructured environments.

Keywords: multi-robot system; object transportation; collaborative task; intelligent system

check for updates

Citation: Choi, D.; Kim, D. Intelligent Multi-Robot System for Collaborative Object Transportation Tasks in Rough Terrains. *Electronics* **2021**, *10*, 1499. https://doi.org/10.3390/electronics10121499

Academic Editor: Hamid Reza Karimi

Received: 18 May 2021
Accepted: 18 June 2021
Published: 21 June 2021

1. Introduction

As the interests in space and other planets increase, several space exploration missions including planetary missions are scheduled [1]. Among several planets within our solar system, Mars is the most attractive planet for human exploration and missions because of the characteristics of Mars that provide moderate temperatures, atmosphere, and the nearly identical day length, etc. [2]. Currently, robotic explorations for acquiring the geological and biological information of Mars and for preparing human missions are planned [3]. In particular, human missions on Mars require construction of outposts and infrastructures and/or (re)locate such large structures or experimental devices to support a long-duration scientific expedition to extreme environments. This leads to the necessity of means to support those activities, and the use of exploration-assisting robots would be one of the good approaches.

Up until now, a single robot system such as a rover with high-power capacity and various functions has been used to conduct several space explorations missions [4]. This required a complex mechanism that includes multiple sensors and experimental devices, and thus, it yielded high costs for building and management and provoked high risks of mission failures [5]. Such disadvantages of operating the single robot system can possibly be resolved by utilizing a multi-robot system (MRS) that operates multiple and small robotic platforms cooperatively. The combination of multiple single-functioning robots, when compared to the multi-functioning single robot system, can bring huge advantages that include, but are not limited to lower cost, better system reliability, greater redundancy, and larger flexibility. In light of this, the operation of the MRS would be beneficial for planetary missions, especially supporting the transportation task.

By virtue of such advantages, the MRS-based object transportation problem has been studied in recent decades. Wang and Schwager [6] proposed a multi-robot manipulation algorithm, which allows the MRS to move an object along the desired trajectory to a goal location. The robots coordinate their actions by sensing the motion of the object without an explicit communication network among themselves, and a force consensus technique using the sensing information is applied to achieve the mission. Chand et al. [7] solved a deformable object transportation problem using a leader-follower formation control algorithm. A path planning algorithm was used to avoid static obstacles for the virtual leader, and a constrained optimization method for the multi-robot formation control was proposed. In addition, in the research investigated by Alonso-Mora et al. [8], a local planner calculates a large obstacle-free convex region around robots, and the parameters of the formation were optimized by sequential convex programming. Then, global path planning is performed via the constrained optimization. Bujarbaruah et al. [9] proposed a leader-follower hierarchical strategy for collaborative object transportation using two robots. Here, the leader solved a model predictive control problem at any given time with the known obstacles to planning a trajectory, and the follower assisted the leader while reacting to additional obstacles. Eoh et al. [10] proposed a cooperative object transportation technique that creates a corridor around objects by lining up two rows using robots. By following a unified field that is composed of a virtual electric dipole field and potential fields, the robots generated an extended corridor to a goal.

Artificial intelligence (AI) is generally divided into (i) deterministic AI that uses the physics of the underlying problem or system and (ii) stochastic AI that has no knowledge of the problem or system [11]. The use of deterministic AI allows an agent to respond to uncertainties or even damages, but re-parameterization of the underlying problem with complexity may be a challenge. For this reason, several studies based on stochastic AI have been actively investigated for an object transportation problem using multiple robots. Jhang et al. [12] proposed an interval type-2 fuzzy neural controller based on a dynamic group differential evolution, which combines a group concept with improved differential evolution, to implement the carrying control and wall-following control for multiple mobile robots. In addition, the authors adopted a reinforcement learning technique to develop an adaptive wall-following control. Dai et al. [13] utilized a fuzzy sliding mode control technique for tracking control of robots while transporting an object, and an artificial potential field approach is additionally applied to avoid obstacles. In addition to this, some scholars considered decentralized approaches to perform a collaborative task using an MRS. Sirineni et al. [14] proposed a decentralized collision avoidance (CA) and motion planning approach for a multi-robot deformable payload transport system. This work solved a convex optimization problem by considering a multi-robot CA algorithm and multi-scale repulsive potential fields as constraints. Zhang et al. [15] proposed a decentralized control scheme on an MRS. Each robot utilized a deep Q-network controller to perform an object transportation task. Since it used a deep reinforcement learning technique, the robots can learn appropriate control strategies through trial-error style interactions within the task environment without the knowledge of the dynamics for the environment.

Plenty of existing collaborative control strategies generally used a centralized controller. Some used a decentralized technique, but the environments considered were flat surfaces in general. Such control techniques were developed assuming the environment would only be flat, so they cannot be applicable for exploration missions on the surface of planets (e.g., the Moon and Mars) with rugged and rough terrain. Based on the authors' preliminary work [16], this work proposes a decentralized approach for an MRS using fuzzy inference systems (FISs) trained by a genetic algorithm (GA) in order to perform a collaborative task with a near-optimal navigation solution in an unstructured environment. By using only the information of the target and the nearest obstacle without knowledge of the states of the object and other robots, the robots are operated in a decentralized manner. In addition, the optimality of the trained FIS model is evaluated by comparing the results obtained by the trained FIS model with one obtained from solving a path optimization problem. Then, the trained model is validated through various testing scenarios in unstructured environments that mimic rough terrain.

The remainder of this paper contains the following contents. Section 2 introduces preliminary knowledge related to the proposed system, and Section 3 explains the environment model considered and problem formulations for the system. Section 4 presents the proposed genetic fuzzy system model including the FISs to determine the inputs for the MRS. In Section 5, a formulation for the path optimization problem that is used as a metric for the optimality evaluation of the proposed model is introduced. Section 6 describes the training and testing processes and discusses the simulation results. Then, the last section summarizes this work.

2. Preliminary: Genetic Fuzzy System

The FIS can be utilized as an intelligent control technique that provides several benefits in the aspect of the design flexibility, its capability as a universal approximator, and the ability to combine with optimization techniques [17]. It has three processes to make decisions, such as fuzzification, inference, and defuzzification, and the grey region in Figure 1 represents the FIS. Through the fuzzification process, a crisp input is converted into a value of the input fuzzy set, and the value of the output fuzzy set is determined by the inference engine that contains the relationship between the input and output. Then, the obtained fuzzy output is transformed into a crisp value as an output through the defuzzification process. Here, the fuzzification and defuzzification steps are composed of the multiple numbers of membership functions that convert the crisp input into the value in the fuzzy set, and the rule-base in the inference engine consists of multiple rules. The main challenge to build FISs is to appropriately select the membership functions and rules of each FIS because the parameters of the membership functions and rules are generally determined by the expert's knowledge. Therefore, it is difficult to anticipate that the FISs with the given expert's knowledge provide the optimal solution.

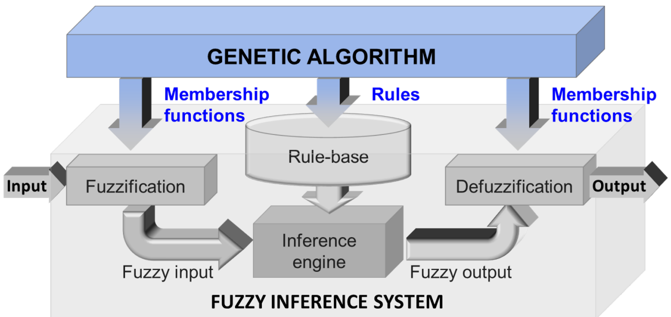


Figure 1. Concept of the genetic fuzzy system.

When the FISs have many inputs and outputs whose relationships are not straightforward, the learning or tuning capability, which is surely useful to build FISs, can be given by using different optimization algorithms, such as an artificial neural network and the GA. While the adaptive network-based FIS (ANFIS) [18] automatically creates sufficient rules considering input and output data and uses the benefit of the learning capability of neural networks, the genetic fuzzy system (GFS) [19] automates the selection of all the parameters of the FISs by using the optimization algorithm, which is the GA, and provides a near-optimal set of FISs' parameters (membership functions and rules) to minimize the pre-defined cost function. Figure 1 illustrates the concept of the GFS. One of the main benefits of the GFS is to utilize the FIS that can provide explainability in terms of the determination of the output that is expressed linguistically. In addition, the GA does not destroy the characteristics of the FIS and ensures a near-optimal solution using its aggressive search capability. Therefore, this work takes advantage of the GFS for the collaborative task of the MRS.

3. Environment Model and Problem Formulations

3.1. Environment Model

Terrain information in other planets is generally provided in the form of the digital elevation map (DEM) that is acquired by reconnaissance orbiters, such as the Lunar Reconnaissance Orbiter or the Mars Reconnaissance Orbiter. The DEM provides the latitude and longitude along with a map scale as a distance per pixel including the elevation information. For path planning and navigation purposes on the ground, in this work, one performs a terrain traversability analysis (TTA) with respect to the slope at each grid.

A square shaped terrain patch centered at x_i and y_j is defined as $P = \{z_{kl} | k = i-L, \ldots, i+L;\ l = j-L, \ldots, j+L\}$ over all information, where L is a positive integer. The number of data points in P is $N = (2L+1) \times (2L+1)$, and the data in P fit to a plane via a least-square approach. To find the plane, the matrix Q is constructed using the data in P as follows [20]:

$$Q = \begin{bmatrix} x_1 - \bar{x} & x_2 - \bar{x} & \cdots & x_N - \bar{x} \\ y_1 - \bar{y} & y_2 - \bar{y} & \cdots & y_N - \bar{y} \\ z_1 - \bar{z} & z_2 - \bar{z} & \cdots & z_N - \bar{z} \end{bmatrix} \begin{bmatrix} x_1 - \bar{x} & y_1 - \bar{y} & z_1 - \bar{z} \\ x_2 - \bar{x} & y_2 - \bar{y} & z_2 - \bar{z} \\ \vdots & \vdots & \vdots \\ x_N - \bar{x} & y_N - \bar{z} & z_N - \bar{z} \end{bmatrix}, \tag{1}$$

where z_m is the elevation value of m-th data with coordinates x_m and y_m for $m = 1, 2, \ldots, N$, and $\bar{x}$, $\bar{y}$, and $\bar{z}$ are defined as

$$\begin{aligned} \bar{x} &= \frac{1}{N} \sum_{m=1}^{N} x_m, \\ \bar{y} &= \frac{1}{N} \sum_{m=1}^{N} y_m, \\ \bar{z} &= \frac{1}{N} \sum_{m=1}^{N} z_m. \end{aligned} \tag{2}$$

Then, an eigenvalue decomposition is performed to obtain the eigenvalues and the corresponding eigenvectors of Q, and the normal vector to the least-square plane is obtained as the eigenvector corresponding to the minimum eigenvalue of Q. Thus, the slope of the plain obtained by P is computed as follows:

$$\gamma_{ij} = \cos^{-1}(\mathbf{h}_{ij} \cdot \hat{\mathbf{n}}_3), \tag{3}$$

where $\mathbf{h}_{ij}$ is the normal vector to the least-square plane and $\hat{\mathbf{n}}_3$ is the unit vector that is normal to the horizontal plane.

The traversable and non-traversable areas are determined by the robot's capability that maintains an object with a stable attitude. For example, if all robots have a manipulator that can hold an object, the robots can keep the object's attitude stable by adjusting the manipulator, and thus, such robots can navigate surfaces with large slopes. If not, accessible areas for navigation are limited. That is, depending on the robots' capability and the MRS's composition, the threshold slope that differentiates the traversable and non-traversable areas is selected. For example, if the slope at the grid is less than the threshold, one assumes the grid as a traversable area. This means that the robots can transport the object while keeping the object's attitude stable. On the other hand, when the computed slope is greater than the threshold, that grid is set to be a non-traversable area. Through this process, the elevation map is transformed into the traversability map, which is composed of traversable and non-traversable areas. Note that non-traversable areas in the traversability map are assumed as obstacles.

3.2. Problem Formulation

To describe motions of robots and an object, frames and vectors are defined as shown in Figure 2. The inertial frame is defined using unit vectors $\hat{\mathbf{n}}_k$ for k = 1, 2, and 3, and the body frame that is fixed on the object's center-of-mass is defined by $\hat{\mathbf{b}}_k$. The position and velocity vectors of i-th robot with respect to the inertial frame are defined as $\mathbf{r}_i = [r_{i,x}, r_{i,y}, r_{i,z}]^T \in \mathbb{R}^3$ and $\dot{\mathbf{r}}_i = [\dot{r}_{i,x}, \dot{r}_{i,y}, \dot{r}_{i,z}]^T \in \mathbb{R}^3$, and the position vectors of each robot with respect to the body frame are defined as $\mathbf{p}_i = [p_{i,x}, p_{i,y}, p_{i,z}]^T \in \mathbb{R}^3$. In addition, the object's position and velocity vectors with respect to the inertial frame are defined as $\mathbf{r}_c = [r_{c,x}, r_{c,y}, r_{c,z}]^T \in \mathbb{R}^3$ and $\dot{\mathbf{r}}_c = [\dot{r}_{c,x}, \dot{r}_{c,y}, \dot{r}_{c,z}]^T \in \mathbb{R}^3$. The attitude (yaw, pitch, and roll angles) of the object with respect to the inertial frame are defined as ψ, θ, and ϕ with the 3-2-1 set of Euler angles, and its angular velocity is defined as $\boldsymbol{\omega} = [\omega_x, \omega_y, \omega_z]^T \in \mathbb{R}^3$.

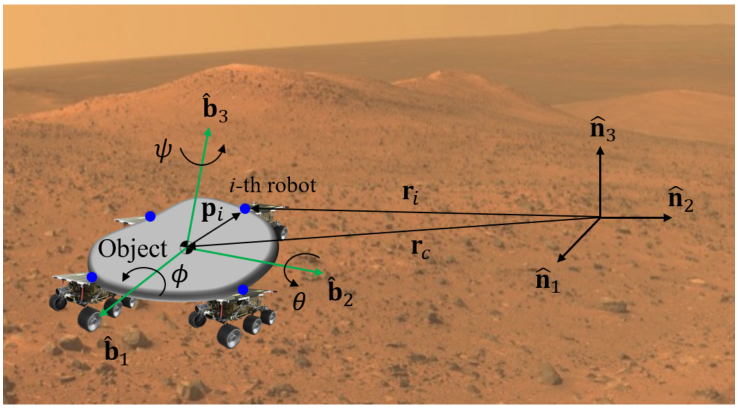


Figure 2. Definitions for frames and vectors.

Then, the kinematic equation between the i-th robot and the object is given by

$$\dot{\mathbf{r}}_i = \dot{\mathbf{r}}_c + \boldsymbol{\omega} \times \mathbf{p}_i, \tag{4}$$

where $i = 1, 2, \ldots, n$, and n is the number of robots. Once the traversability map is given, the motions of the robots with an object for navigation are modeled as planar motion on the ground. With two components for the position and velocity vectors in the x-y plane and one component for the attitude and angular velocity about the z-axis made by $\hat{\mathbf{n}}_3$, the changes of the velocity for each robot form the translational and rotational motion of the object on the ground. Then, Equation (4) is rearranged in terms of the unknown states for the object, $\dot{r}_{c,x}$, $\dot{r}_{c,y}$, and ω_z, as follows:

$$\begin{bmatrix} 1 & 0 & -p_{1,y} \\ 0 & 1 & p_{1,x} \\ & \vdots & \\ 1 & 0 & -p_{n,y} \\ 0 & 1 & p_{n,x} \end{bmatrix} \begin{bmatrix} \dot{r}_{c,x} \\ \dot{r}_{c,y} \\ \omega_z \end{bmatrix} = \begin{bmatrix} \dot{r}_{1,x} \\ \dot{r}_{1,y} \\ \vdots \\ \dot{r}_{n,x} \\ \dot{r}_{n,y} \end{bmatrix}. \tag{5}$$

The object's unknown states are obtained by the least square approach once the robots' velocity vectors and their position vectors are given. Then, the position vectors of the object and the robots are computed using the Euler method. Note that the robots' locations with respect to the center-of-mass of the object are specified in advance, and the robots' velocity inputs are determined by the proposed FIS models that will be discussed in Section 4.

4. Proposed Genetic Fuzzy System Model

4.1. Input and Output Variables of Fuzzy Inference Systems

At every time step, the robots' positions are determined by their velocities obtained by the FISs. Hence, this work utilizes two FISs to obtain the robots' velocity input. One of the FISs determines the components related to the i-th robot's velocity vector toward the target position, which is defined as $\mathbf{v}_{i,t}$, and the other determines the components associated with i-th robot's velocity vector avoiding obstacles, which is defined as $\mathbf{v}_{i,o}$. That is, FIS1 determines a magnitude ($v_{i,t}$) and a correction angle ($\beta_{i,t}$) for $\mathbf{v}_{i,t}$, and FIS2 determines a magnitude ($v_{i,o}$) and a correction angle ($\beta_{i,o}$) for $\mathbf{v}_{i,o}$, respectively.

FIS1 requires (i) the relative distance $d_{i,t}$ between the i-th robot and the target position and (ii) the angle difference $\alpha_{i,t}$ between i-th robot's velocity vector and the relative position vector $\mathbf{r}_{t/i}$ from the i-th robot's position to the target position as inputs. Figure 3 shows the vector definitions that are used to compute the input variables of the FISs. The inputs for FIS1 are computed as

$$d_{i,t} = ||\mathbf{r}_{t/i}||, \tag{6}$$

$$\alpha_{i,t} = \cos^{-1}\left(\frac{\dot{\mathbf{r}}_i^T \mathbf{r}_{t/i}}{||\dot{\mathbf{r}}_i||\ ||\mathbf{r}_{t/i}||}\right). \tag{7}$$

In addition, the inputs of FIS2 are defined as (i) the relative distance $d_{i,o}$ between the i-th robot's position and the nearest obstacle's position and (ii) the angle difference $\alpha_{i,o}$ between the i-th robot's velocity vector and the relative position vector $\mathbf{r}_{o/i}$ from the i-th robot's position to the nearest obstacle's position. Similarly, the inputs for FIS2 are found as

$$d_{i,o} = ||\mathbf{r}_{o/i}||, \tag{8}$$

$$\alpha_{i,o} = \cos^{-1}\left(\frac{\dot{\mathbf{r}}_i^T \mathbf{r}_{o/i}}{||\dot{\mathbf{r}}_i||\ ||\mathbf{r}_{o/i}||}\right). \tag{9}$$

With the inputs from Equations (6)–(9), two FISs respectively provide two outputs, and two velocity vectors for the i-th robot are calculated by using the following equations:

$$\mathbf{v}_{i,t} = [v_{i,t}\cos(\beta_{i,t} + \rho_i),\ v_{i,t}\sin(\beta_{i,t} + \rho_i)]^T, \tag{10}$$

$$\mathbf{v}_{i,o} = [v_{i,o}\cos(\beta_{i,o} + \rho_i),\ v_{i,o}\sin(\beta_{i,o} + \rho_i)]^T. \tag{11}$$

where ρ_i is the i-th robot's heading angle that represents the direction of the velocity vector of the robot as shown in Figure 3. Therefore, the velocity input for each robot is finally obtained by summing the two vectors as $\mathbf{v}_{i,t} + \mathbf{v}_{i,o}$. In fact, each robot only utilizes the information of the robot itself to determine the velocity input as a decentralized MRS.

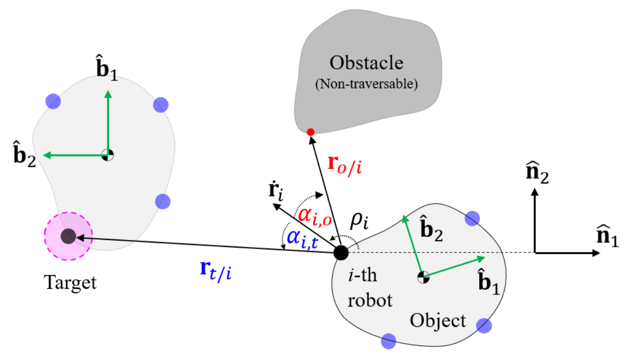


Figure 3. Definitions of FIS-related vectors.

4.2. Components of Fuzzy Inference Systems and Its Training Process

The outputs of the FISs defined in Section 4.1 are obtained through the fuzzification, inference, and defuzzification process using the inputs. The fuzzification and defuzzification processes contain several membership functions, and the inference engine has multiple rules in the rule-base. For the two FISs, it considers the triangle shape of membership functions to minimize the number of parameters. In addition, the edges and the center point of the membership functions and the rules in the rule-base are set to be the unknown parameters that would be optimized by the GA. There are 20 parameters to be optimized for the membership functions and 54 for the rules in the rule-base. For the defuzzification process to obtain the output value, the centroid method that is the most common method is utilized.

Figure 4 describes the training process of the GFS. At the beginning of the training process, it requires the initialization of the population and the GA parameters that include the number of generations, population size, and stall number of generations. Then, sets of individuals, which represent FIS parameters to be optimized, are randomly generated up to the population size and updated as the population. After that, the selection, crossover, and mutation process, which are the main contributions of the GA, are applied to sets of individuals. Once this process is completed, one performs the evaluation of the fitness function using the FIS models. Note that each set of individuals is substituted into the FIS parameters for the evaluation. Among several sets of individuals, the set of individuals that has the minimum fitness value is chosen as the best fit solution at the current generation, and it checks the convergence criteria that are defined as the stall number of generations and the maximum number of generations. If the fitness value is not changed during the stall number of generations or the maximum number of generations is achieved, the training process terminates. This way, the optimized FISs using the best fit solution that has the minimum fitness value is obtained. Note that the evaluation function for the optimization, which is known as the fitness function for the GA, is defined as

$$f_{\text{fit}} = \sum_{i=1}^{n} d_i + \zeta, \tag{12}$$

where d_i is the total travel distance of the i-th robot to reach the target position, and ζ is a penalty that is applied for any undesirable path. Here, the penalty is defined as a collision between each robot and the obstacles and a situation where any of the robots' positions is out of the tolerance range at the end of the simulation.

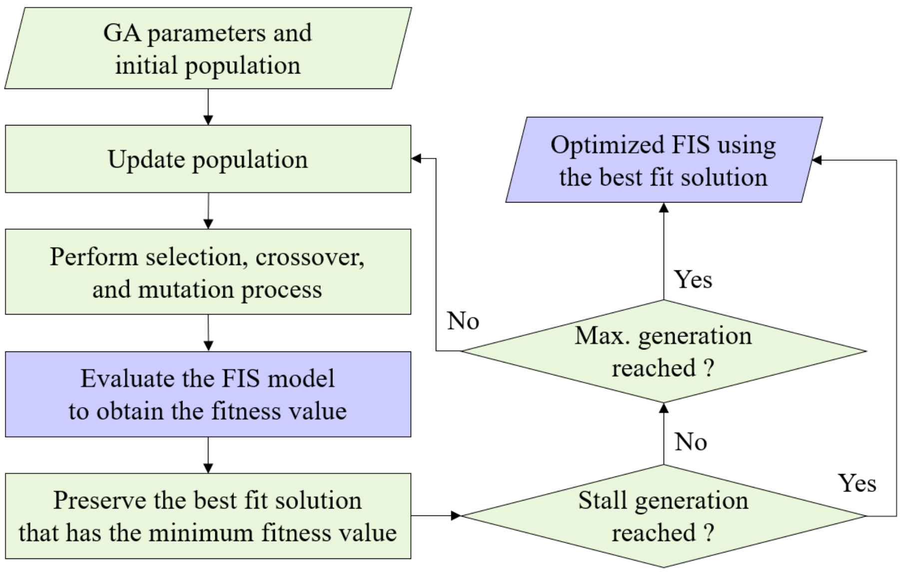


Figure 4. Flowchart for the training process.

5. Path Optimization to Evaluate the Proposed System

The proposed FIS models are optimized by the GA in the training process. To evaluate the optimality of the trained model, a path optimization problem within a structured environment is formulated, and the resulting solution is used as a metric. The performance index for the discretized path optimization is defined by

$$J = \sum_{i=1}^{n} \sum_{k=1}^{K-1} \sqrt{(\mathbf{r}_i[k+1] - \mathbf{r}_i[k])^T (\mathbf{r}_i[k+1] - \mathbf{r}_i[k])}, \tag{13}$$

subject to

$$\mathbf{r}_c[k+1] = \mathbf{r}_c[k] + \Delta t \mathbf{v}_c[k], \tag{14}$$

$$\psi[k+1] = \psi[k] + \Delta t \omega_z[k], \tag{15}$$

$$\mathbf{r}_i[k] = \mathbf{r}_c[k] + C(\psi[k])\mathbf{p}_i, \tag{16}$$

where n is the number of robots, K is the number of data, $\mathbf{r}_i[k]$ is the position of the i-th robot at the k-th step, $\mathbf{r}_c[k]$ and $\mathbf{v}_c[k]$ are the object's position and velocity at the k-th step, $\psi[k]$ is the orientation of the object at the k-th step, Δt is the time step between k and $k+1$ steps, and $C(\psi[k])$ is given by

$$C(\psi[k]) = \begin{bmatrix} \cos\psi[k] & -\sin\psi[k] \\ \sin\psi[k] & \cos\psi[k] \end{bmatrix}. \tag{17}$$

The final states of the robots must satisfy the following constraints:

$$(\mathbf{r}_i[K] - \mathbf{r}_{iT})^T (\mathbf{r}_i[K] - \mathbf{r}_{iT}) - d_{\text{tol}}^2 \leq 0, \tag{18}$$

where $\mathbf{r}_{iT}$ is the target position of the i-th robot, and d_{tol} is each robot's allowable tolerance range. In addition, the state constraints for avoiding obstacles are defined as

$$(R_j + d_{\text{col}})^2 - \left(\mathbf{r}_i[k] - \mathbf{c}_j\right)^T \left(\mathbf{r}_i[k] - \mathbf{c}_j\right) \leq 0, \tag{19}$$

where $j = 1, \ldots, n_o$, n_o is the number of obstacles, R_j is the radius of the j-th obstacle, d_{col} is the collision threshold, and $\mathbf{c}_j$ is the position of the j-th obstacle. Since the states are bounded, one must consider the following constraints:

$$-2 \leq v_{c,x}[k] \leq 2 \text{ (m/s)}, \tag{20}$$

$$-2 \leq v_{c,y}[k] \leq 2 \text{ (m/s)}, \tag{21}$$

$$-0.5 \leq \omega_z[k] \leq 0.5 \text{ (rad/s)}. \tag{22}$$

6. Simulation Studies

6.1. Descriptions of Training and Testing Environments

While performing a collaborative object transportation task, the MRS should navigate to the target position avoiding collision with obstacles in order to ensure safe operation. It considered the two well-known situations that generally affect CA performance in environments with static obstacles as the training scenarios. One is local minima where an obstacle locates exactly between the agent and the target position, and there is a high possibility that the agent gets stuck in front of the obstacle when the proper direction determination strategy is not considered. The other is the situation where the agent cannot reach the target position when an obstacle locates near the target position. Since the FIS model of the velocity vector avoiding the obstacle utilizes the relative distance between the robot and the obstacle, this situation should be considered in the training process. These two situations are displayed in Figure 5 as Scenario 1. In addition, Scenario 2 considers a cluttered environment with multiple obstacles between the initial and target positions. Therefore, this research considers two scenarios that are artificially generated structured environments shown in Figure 5 in order to guarantee the CA capability of the MRS.

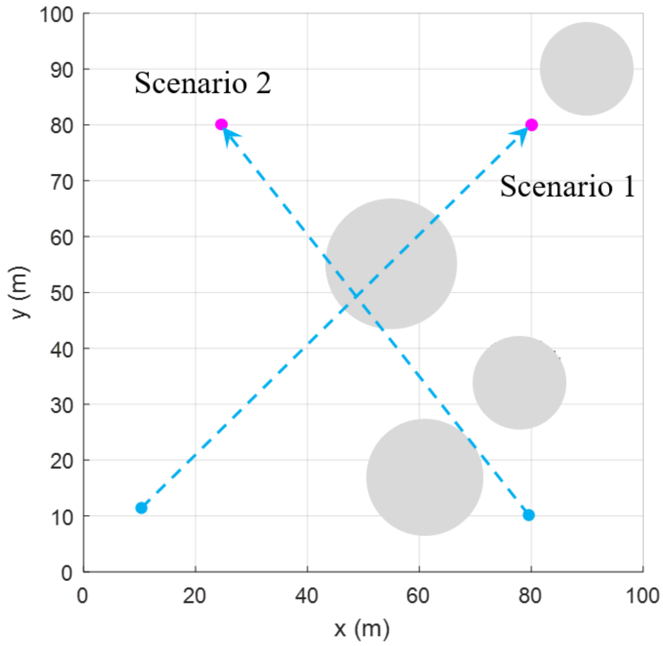


Figure 5. Training scenarios.

The GA parameters for optimization and simulation parameters for training are listed in Tables 1 and 2. Here, each robot is assumed to be allocated at each vertex around the square-shaped object with a width of 2 m. It considers a collision if the relative distance between the robot and the nearest obstacle is less than or equal to the collision threshold. In addition, if all robots are within the tolerance range from the target position, it regards the mission as completed.

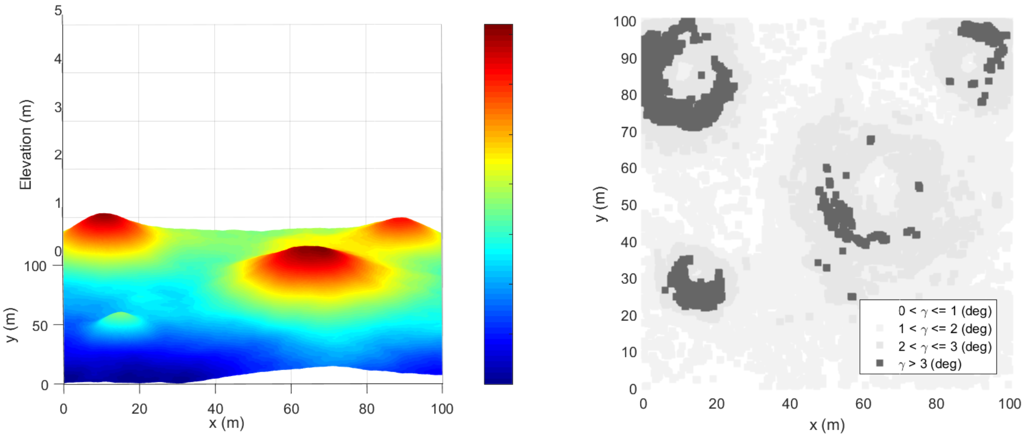
Table 1. GA parameters and algorithms for training.

Parameters	Values
Number of generations	100
Population size	64
Stall number of generations	20
Elitism ratio	0.2
Selection algorithm	Tournament selection
Crossover algorithm	Two-points crossover
Mutation algorithm	Adaptive feasible

Table 2. Simulation parameters for training.

Parameters			Values
Robots' initial & target position	Scenario 1	Initial	$[9, 11]^T$, $[11, 11]^T$, $[11, 9]^T$, & $[9, 9]^T$ (m)
		Target	$[81, 81]^T$, $[81, 79]^T$, $[79, 79]^T$, & $[79, 81]^T$ (m)
	Scenario 2	Initial	$[81, 11]^T$, $[81, 9]^T$, $[79, 9]^T$, & $[79, 11]^T$ (m)
		Target	$[25, 81.41]^T$, $[26.41, 80]^T$, $[25, 78.59]^T$, & $[23.59, 80]^T$ (m)
Target tolerance range			0.2 (m)
Collision threshold			1 (m)

For the testing environment, a Brownian surface [21], which is a factual surface generated via an elevation function, is considered. This is because it can model rough surfaces' complex shapes of nature. Based on the generated surfaces, one added multiple Gaussian bumps to mimic a hill and crater-like shape. Then, one performed the TTA to convert the elevation map into the traversability map discussed in Section 3.1 and set the threshold slope. This research assumed the MRS can transport the object with stable attitude for up to 3 deg slope changes. Thus, areas less than 3 deg slope changes are the traversable areas while the remaining areas became non-traversable areas. The elevation map generated and the converted terrain traversability map are displayed in Figure 6. The elevation information of the map is highlighted as a colored surface in Figure 6a, and Figure 6b shows the slope information at each position for the given elevation map. Here, the dark grey regions are non-traversable areas. With this traversability map, the trained FIS models are tested with the parameters listed in Table 3.



(**a**) Elevation map (**b**) Terrain traversability map

Figure 6. Testing environment.

Table 3. Simulation parameters for testing.

Parameters			Values
Robots' initial & target position	Case 1	Initial	$[16, 9]^T$, $[14, 9]^T$, $[14, 11]^T$, & $[16, 11]^T$ (m)
		Target	$[30.59, 77]^T$, $[32, 78.41]^T$, $[33.41, 77]^T$, & $[32, 75.59]^T$ (m)
	Case 2	Initial	$[71, 91]^T$, $[71, 89]^T$, $[69, 89]^T$, & $[69, 91]^T$ (m)
		Target	$[14, 6]^T$, $[16, 6]^T$, $[16, 4]^T$, & $[14, 4]^T$ (m)
	Case 3	Initial	$[11, 41]^T$, $[11, 39]^T$, $[9, 39]^T$, & $[9, 41]^T$ (m)
		Target	$[90, 31.41]^T$, $[91.41, 30]^T$, $[90, 28.59]^T$, & $[88.59, 30]^T$ (m)

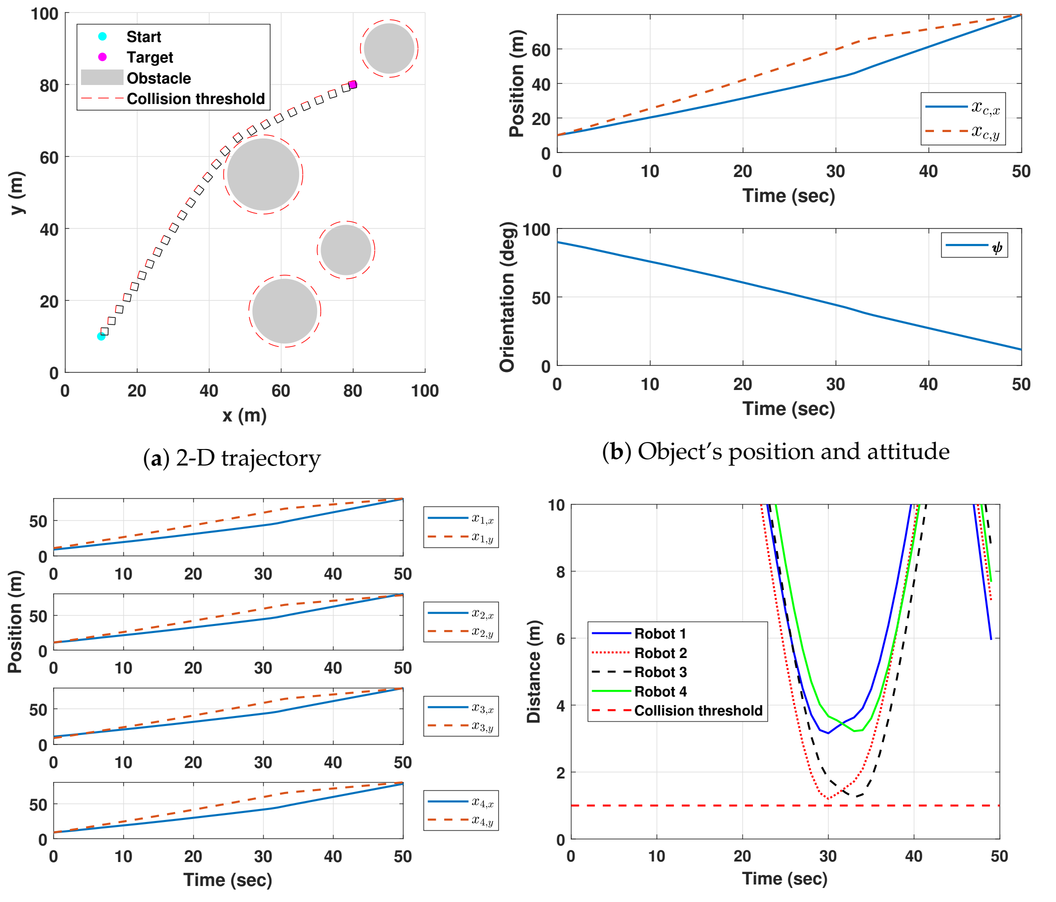
6.2. Path Optimization Results

The path optimization problem discussed in Section 5 is numerically solved for the training scenarios using the parameters listed in Table 2. Since the purpose for solving this problem is to evaluate the optimality of the trained model via the GA, this work mainly focuses on the aspect of the total path length traveled by all robots. That is, one assumes a certain long enough final time that can find the optimal solution. The optimization results are displayed in Figures 7 and 8 and Table 4.

Table 4. Path optimization results.

	Minimum Relative Distance (m)				Total Path Length (m)
	Robot 1	**Robot 2**	**Robot 3**	**Robot 4**	
Scenario 1	1.08	1.58	1.02	2.04	408.84
Scenario 2	3.16	1.20	1.24	3.22	357.89

The computed travel distances of all robots for each scenario are 408.84 m and 357.89 m, respectively, as shown in Table 4. Figure 7a depicts the 2-D trajectory of the object transported by the robots for Scenario 1. The robots go directly toward the obstacle's left side, and then move to the target position while adjusting the object's orientation. The position and orientation are displayed in Figure 7b, and it shows that the states are linearly changed until the robots meet the target conditions. Figure 7c shows the time history of the robots' positions, and the results are similar to the object's position shown in Figure 7b. Figure 7d shows the relative distance between the nearest obstacle and each robot during operation, and it is observed that each robot does not violate the collision threshold of 1 m that is highlighted with the red-dotted line. In addition, this is confirmed in the results for the minimum relative distance of each robot listed in Table 4: 1.02 m in Scenario 1 and 1.20 m in Scenario 2, which are greater than 1 m. That is, it can be said that the MRS successfully completes the object transportation task without collision. The 2-D trajectory of the object for Scenario 2 is displayed in Figure 8a, and it shows the similar result compared to Scenario 1 because the cost function is formulated to minimize the travel distance of the robots. Figure 8b shows the time histories of the object's states, and their trends over time are also linear. Figure 8c,d display the robots' positions and the relative distance between the nearest obstacle and each robot, respectively. The time histories of the robots' positions are similar to the object's position, and the robots stay away from the obstacles with the sufficient relative distance while avoiding obstacles as shown in Table 4.



(**a**) 2-D trajectory (**b**) Object's position and attitude

(**c**) Robots' position (**d**) Relative distance

Figure 7. Optimization results: Scenario 1.

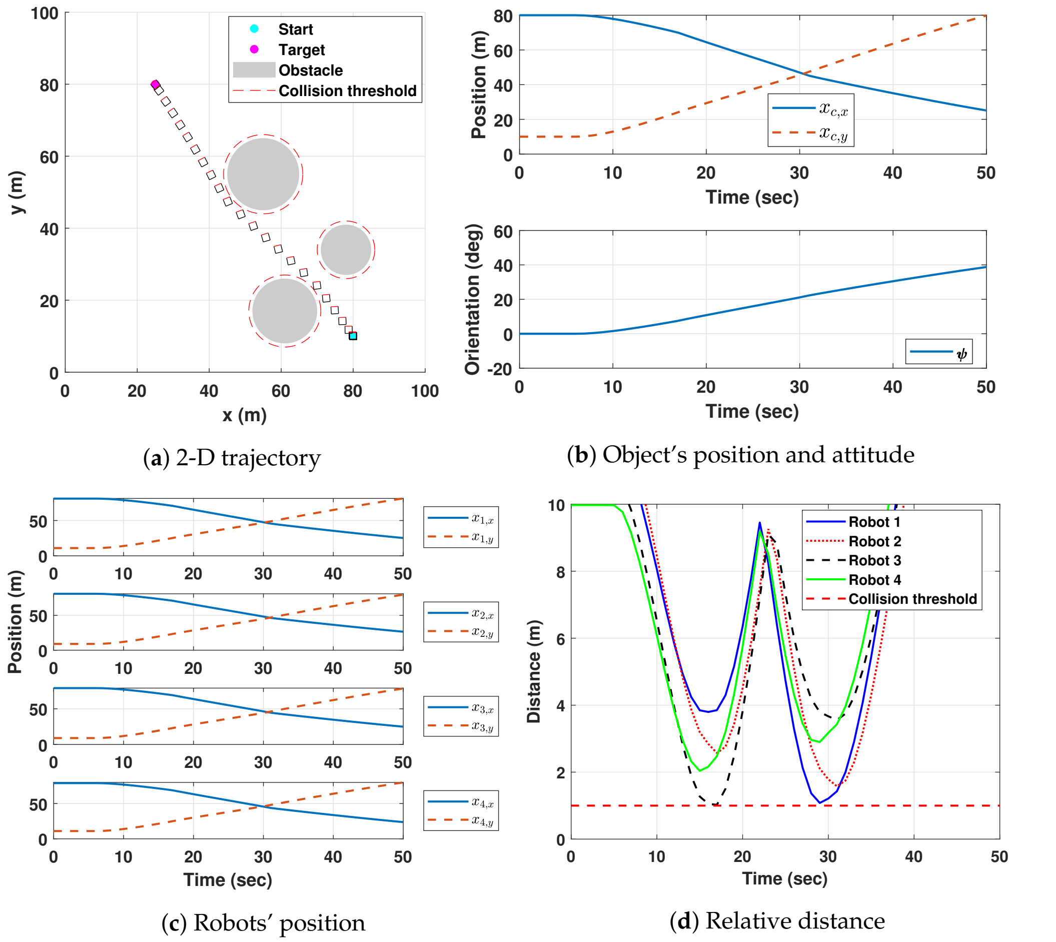


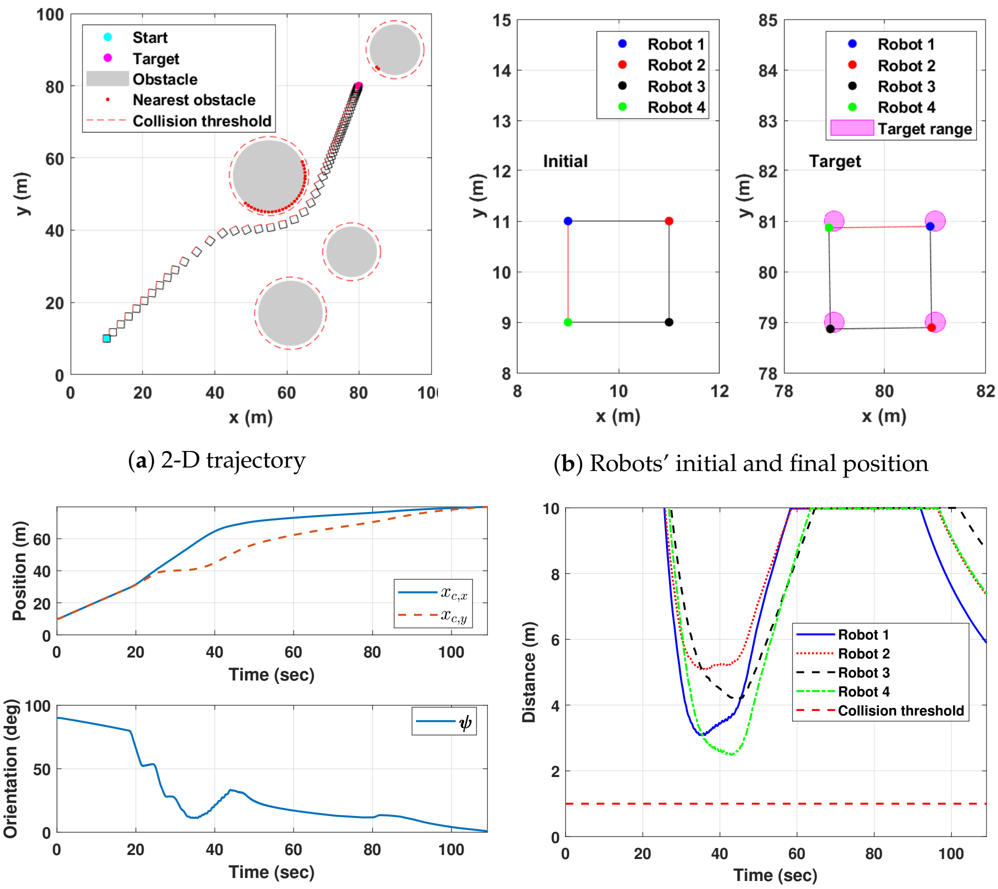
Figure 8. Optimization results: Scenario 2.

6.3. Training Results

With the parameters in Tables 1 and 2, the GA-based training process is applied to the MRS, and the training results are displayed in Figures 9 and 10 and Table 5. Figures 9a and 10a show the overall trajectory of the MRS for each scenario to transport the object to the designated target position. It is observed that the MRS successfully completes the collaborative mission without collision for the given scenarios while adjusting the object's orientation. The configurations of the robots and object and the orientation changes between the initial and final time are displayed in Figures 9b and 10b. The results show that the MRS meets the requirement that each robot locates within the target tolerance range. Here, the edge of the object between Robot 1 and 4 is highlighted as a red-colored line to clearly describe the orientation change of the object in the 2-D trajectories. Unlike the optimization results, the robots travel on the right side of the obstacle when avoiding it. In fact, whichever the direction the robots select provides the same results in terms of the travel distance of the robots, but the robots choose the right side of the object in this training result. The total travel distances of the MRS are respectively computed as 422.44 m and 370.52 m, listed in Table 5, and the MRS applying the trained FIS models provide 3.33% and 3.53% longer travel distance than the path optimization results, respectively. This might happen because of the unnecessary maneuver, which can be observed in the orientation changes, near the obstacle. However, since the differences of the total path length obtained by the path optimization and the trained FIS models are small enough (less than 5%), it can be said that the FIS models are well trained by the GA. In addition, Figures 9c and 10c display the time history of the object's position and orientation of the object. It is expected that the orientation of the object is linearly changed, as shown in the optimization result in Figure 7b, but it is observed that trend of the orientation changes is not exactly linear but mostly linear. In fact, there are some fluctuations when the MRS adjusts their velocity to avoid the obstacles, and the velocity changes of the robots are observed during the same time period, as shown in Figures 9f and 10f. Here, the oscillations of the velocity input happened during avoidance maneuvers because the changes of the angle inputs, $\alpha_{i,t}$ and $\alpha_{i,o}$. To enable realistic motions for the robots, the accelerations of each robot are restricted to less than 2 m/s^2. That is, if the current acceleration computed by the velocity difference (between the current and previous time step) divided by the time interval exceeds the acceleration limit, one recalculates the current velocity based on the acceleration limit and the previous velocity, instead of using the output of the trained model. Each robot's position over time shown in Figures 9e and 10e has the trajectories similar to the object's position as expected. Figures 9d and 10d show the relative distance between each robot and the nearest obstacle, and none of the relative distances violates the collision threshold, as shown in Table 5.

Table 5. Training results.

	Minimum Relative Distance (m)				Total Path Length (m)
	Robot 1	Robot 2	Robot 3	Robot 4	
Scenario 1	3.08	5.08	4.18	2.48	422.44 (3.33% longer)
Scenario 2	2.32	1.04	2.64	2.83	370.52 (3.53% longer)



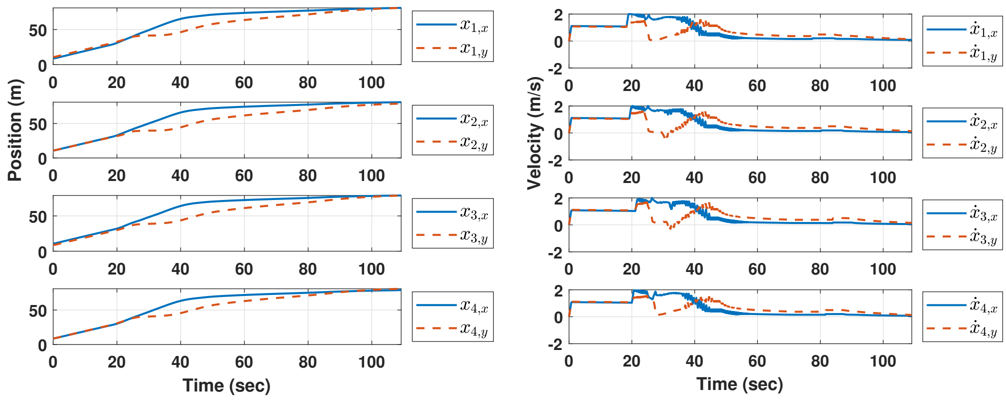
(**a**) 2-D trajectory

(**b**) Robots' initial and final position

(**c**) Object's position and attitude

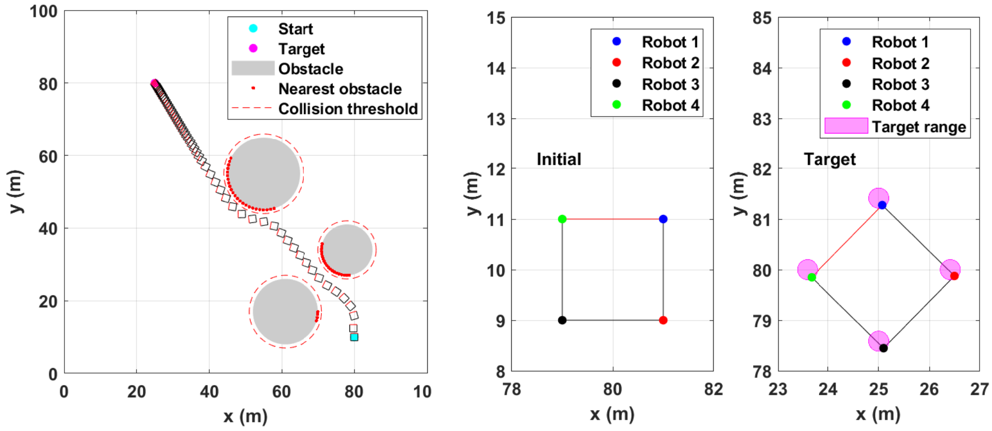
(**d**) Relative distance

Figure 9. *Cont.*

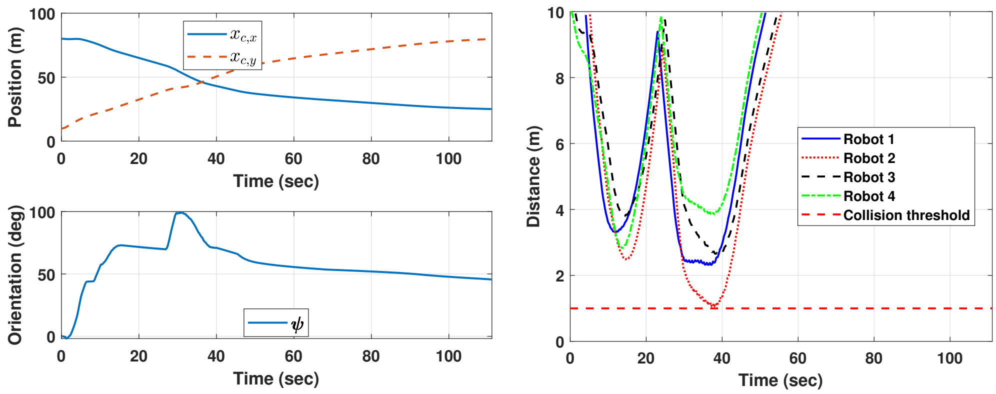


(**e**) Robots' position (**f**) Robots' velocity

Figure 9. Training results: Scenario 1.

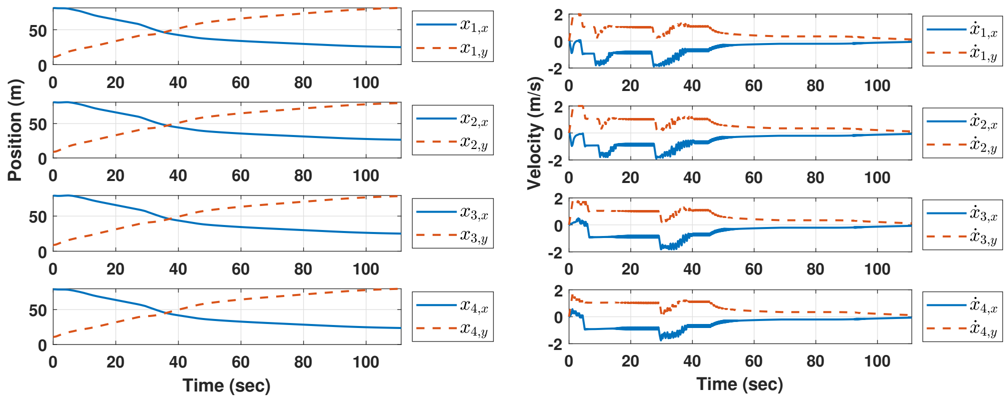


(**a**) 2-D trajectory (**b**) Robots' initial and final position



(**c**) Object's position attitude (**d**) Relative distance

Figure 10. *Cont.*

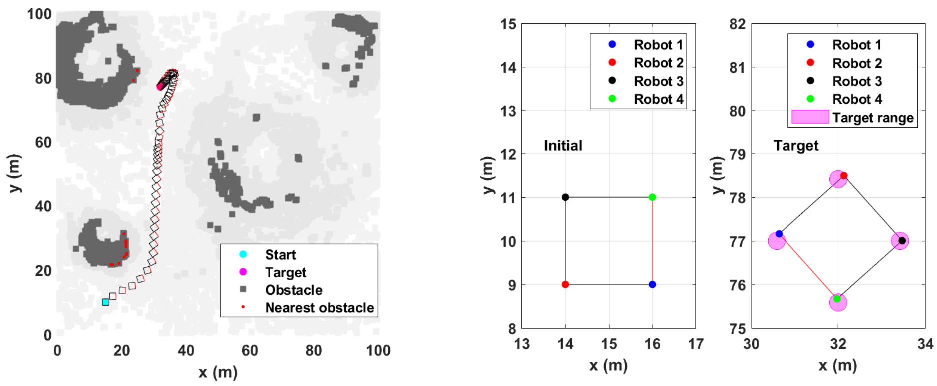


(**e**) Robots' position (**f**) Robots' velocity

Figure 10. Training results: Scenario 2.

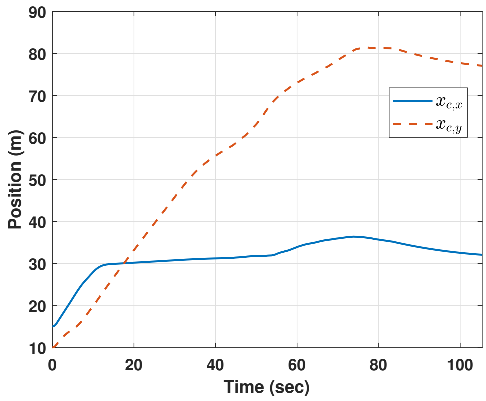
6.4. Testing Results

To validate the performance of the proposed models, the trained FIS models are applied to the testing environment shown in Figure 6b using the testing conditions listed in Table 3, and the results are displayed in Figures 11–13 and Table 6. The robots for all cases successfully transport the object to the designated target position without collision, as shown in Figures 11a, 12a and 13a, and the total travel distances of the robots in each case are obtained as 352.40 m, 420.82 m, and 325.98 m, respectively, as listed in Table 6. Figures 11b, 12b and 13b display that each robot is within the target tolerance range at the final time, which means that the object transportation mission is achieved. In addition, the mission achievement can be confirmed from Figures 11e, 12e and 13e that describe the robots' positions over time, and the object's position for each case shown in Figures 11c, 12c and 13c is observed similar to the time histories of the robots' positions. From the relative distance shown in Figures 11g, 12g and 13g, it is observed that the MRS for all cases never violates the collision threshold. The minimum relative distances of the robots in Table 6 support this statement. As shown in Figures 11d, 12d and 13d, the roll and pitch angles remain within 3 deg during the entire simulation time, and this represents the MRS stably transports the object to the target position. The trend of the yaw angle changes seems mostly linear for all cases with some fluctuations. These fluctuations are generated while the robots adjust their velocities to avoid obstacles, as shown in Figures 11f, 12f and 13f. In particular, for Case 1, the yaw angle is increased and then maintained during 40 s, as shown in Figure 11d. However, it suddenly decreases and dramatically increases after 40 s. This happened when the obstacles are close to the target position. That is, the robots' velocity vectors that play a role in avoiding obstacles are continuously activated, and this affects the yaw angle changes.

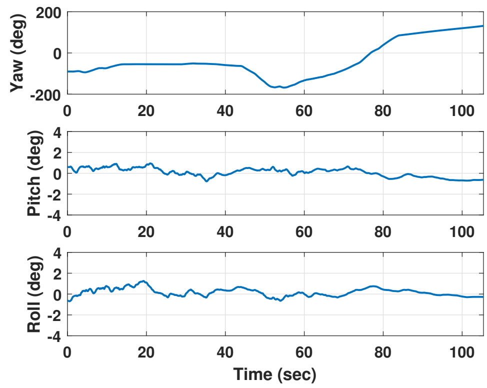


(**a**) 2-D trajectory (**b**) Robots' initial and final position

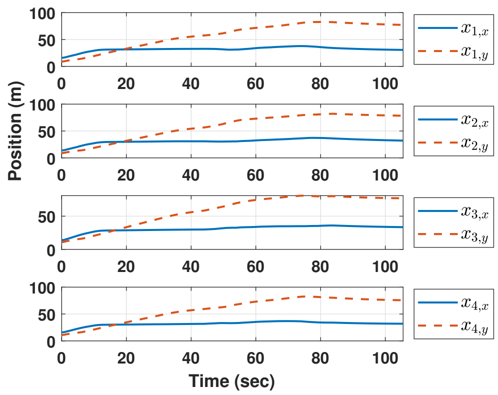
Figure 11. *Cont.*



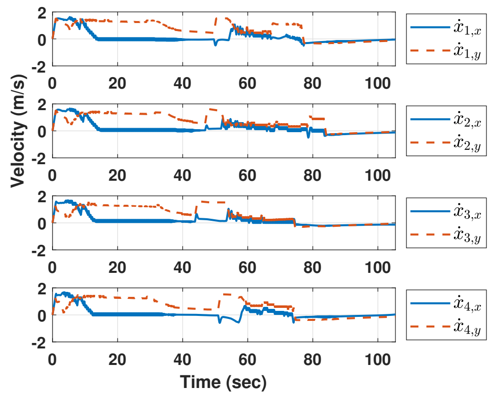
(**c**) Object's position



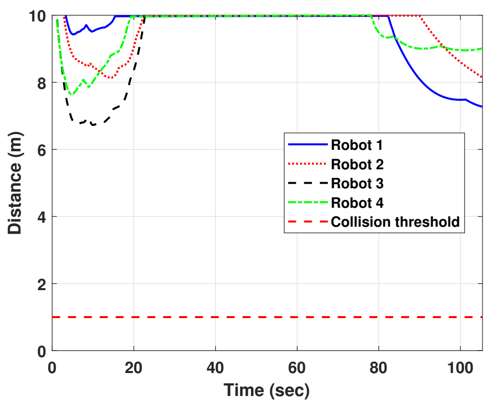
(**d**) Object's attitude



(**e**) Robots' position



(**f**) Robots' velocity

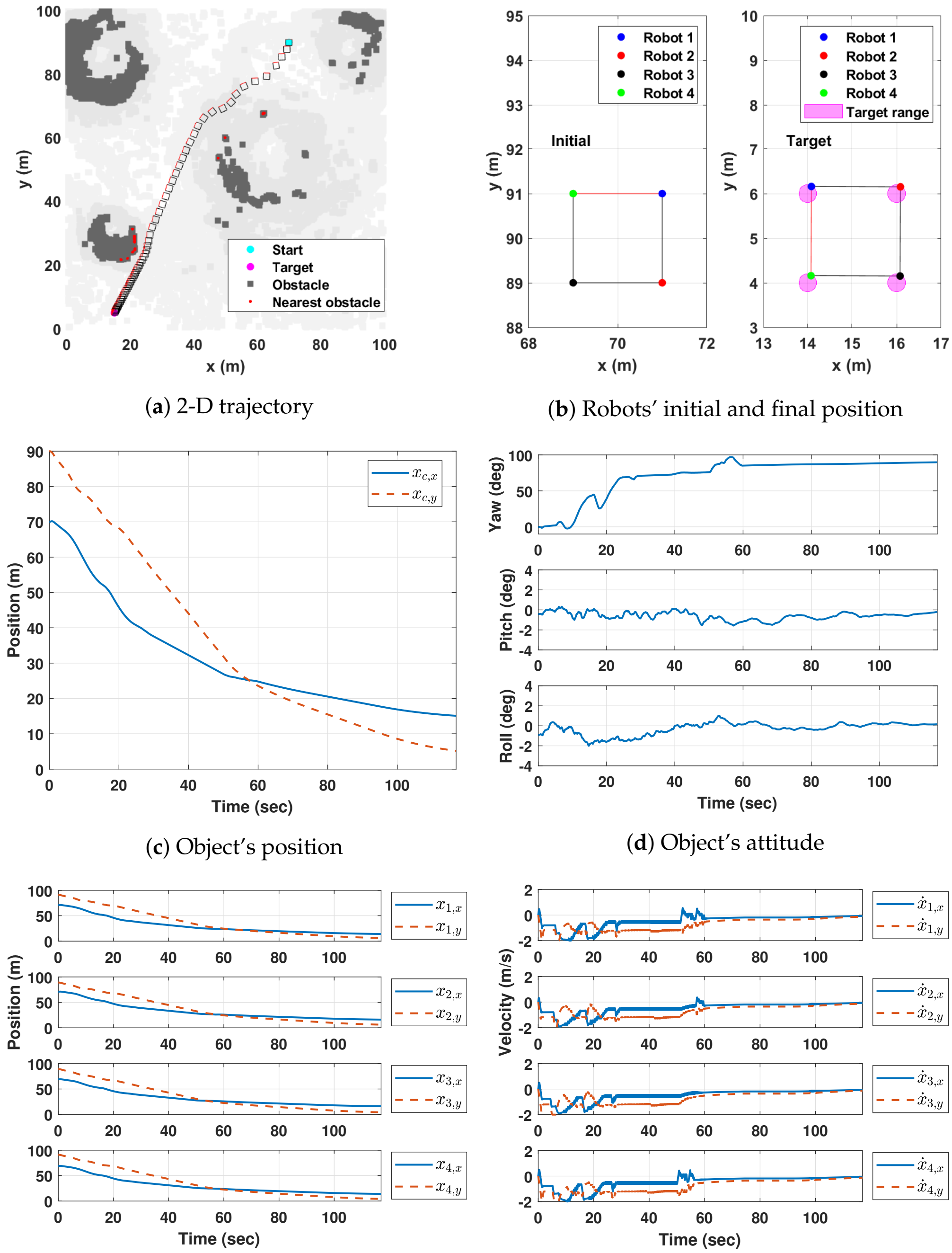


(**g**) Relative distance

Figure 11. Testing results: Case 1.

Table 6. Testing results.

	Minimum Relative Distance (m)				Total Path Length (m)
	Robot 1	**Robot 2**	**Robot 3**	**Robot 4**	
Case 1	7.27	8.13	6.73	7.65	352.40
Case 2	2.19	4.03	4.76	2.83	420.82
Case 3	1.12	2.59	1.87	2.44	325.98



(**a**) 2-D trajectory

(**b**) Robots' initial and final position

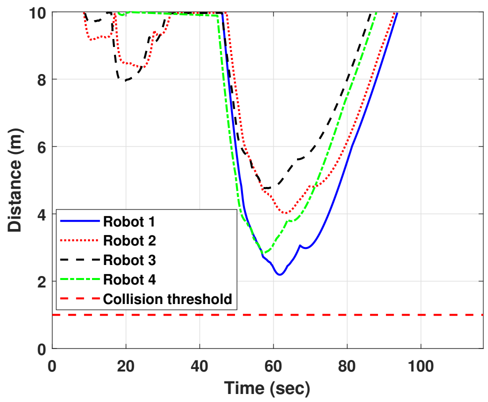
(**c**) Object's position

(**d**) Object's attitude

(**e**) Robots' position

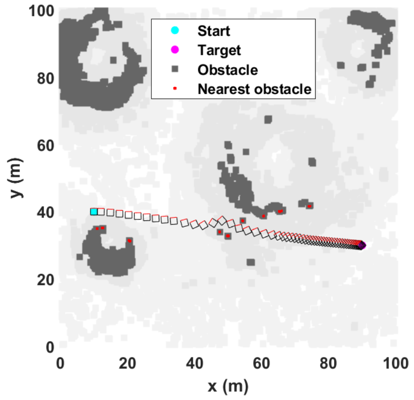
(**f**) Robots' velocity

Figure 12. *Cont.*

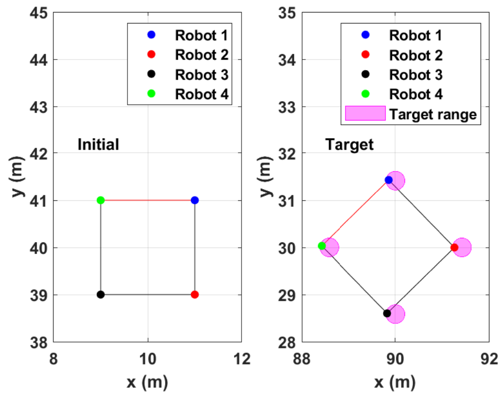


(**g**) Relative distance

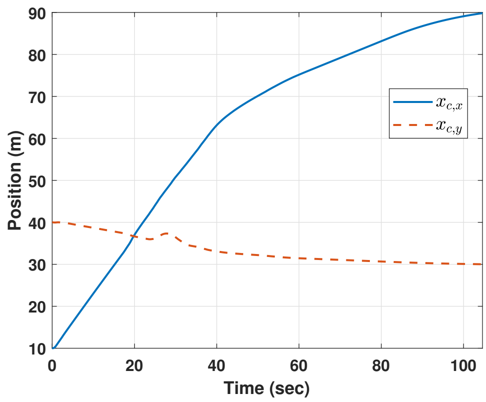
Figure 12. Testing results: Case 2.



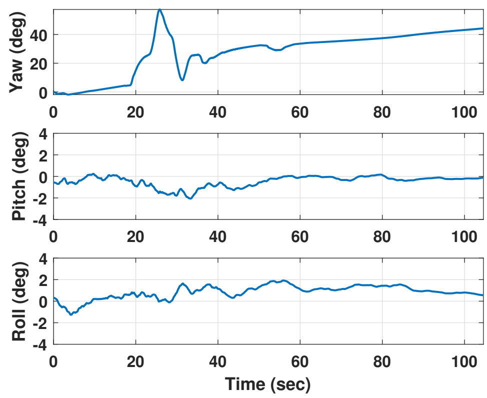
(**a**) 2-D trajectory



(**b**) Robots' initial and final position

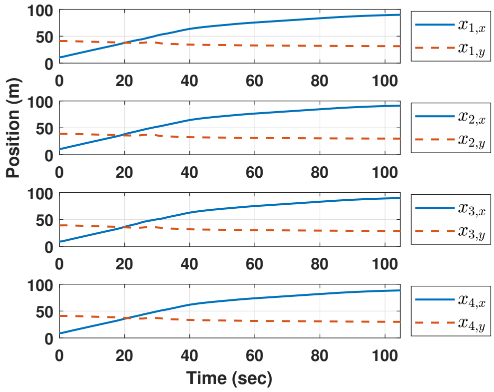


(**c**) Object's position

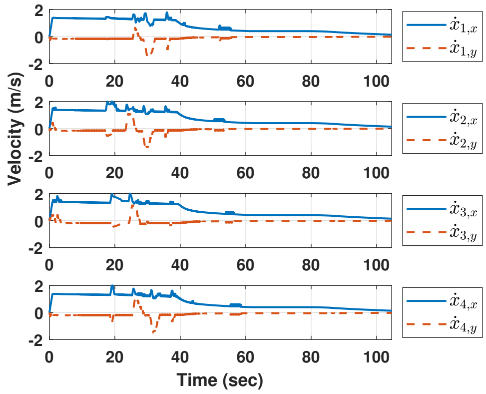


(**d**) Object's attitude

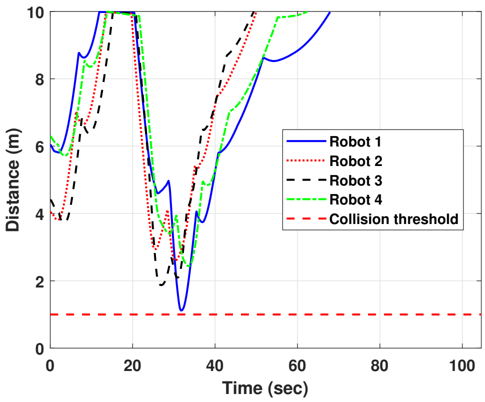
Figure 13. *Cont.*



(**e**) Robots' position



(**f**) Robots' velocity



(**g**) Relative distance

Figure 13. Testing results: Case 3.

7. Conclusions

A decentralized multi-robot system (MRS) operation technique is proposed to perform a collaborative object transportation mission with the shortest travel distance in a rough terrain environment stably. The proposed technique is developed based on the genetic fuzzy system that utilizes the genetic algorithm to optimize the fuzzy inference system (FIS). An artificially generated environment is used in the training process using multiple situations that include a local minima, inaccessible target, and a cluttered environment. In addition, the trained model is validated by comparing it with the path optimization results. It is shown that the differences between the trained model results and the optimization results are only 3.33% and 3.53% for scenarios 1 and 2, and this indicates that the models are well trained. Then, the trained FIS models are applied to the testing environment generated via the combination of a Brownian surface and Gaussian bumps that models the rough terrain. The terrain traversability analysis is performed to transform the given elevation map into a traversability map. The testing results show that the MRS using the well trained FIS models successfully transports the object to the target position with nearly minimum travel distance without collision while the small changes (less than 3 deg pre-defined) of the object's roll and pitch angles are achieved to maintain stable operation. This demonstrates that the proposed approach is beneficial for the decentralized MRS to achieve a common task. In future work, we will develop an MRS operation technique considering the additional practical factors, such as dynamics of the system, energy consumption, and mission execution time.

Author Contributions: Conceptualization, D.C. and D.K.; methodology, D.C.; software, D.C.; validation, D.C. and D.K.; formal analysis, D.C. and D.K.; investigation, D.C.; resources, D.C. and D.K.; data curation, D.C.; writing—original draft preparation, D.C.; writing—review and editing, D.K.; visualization, D.C.; supervision, D.K.; project administration, D.K.; funding acquisition, D.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. NASA. Artemis Program. 2020. Available online: https://www.nasa.gov/specials/artemis/ (accessed on 19 June 2021).
2. Halbach, E. Multi-Robot Hillside Excavation Strategies for Mars Settlement Construction. In Proceedings of the 47th International Conference of Environmental Systems, Charleston, SC, USA, 16 July 2017.
3. Sonsalla, R.; Cordes, F.; Christensen, L.; Roehr, T.M.; Stark, T.; Planthaber, S.; Maurus, M.; Mallwitz, M.; Kirchner, E.A. Field Testing of a Cooperative Multi-Robot Sample Return Mission in Mars Analogue Environment. In Proceedings of the 14th Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA 20017), ESA/ESTEC, ESA, Scheltema, The Netherlands, 20–22 June 2017.
4. Yan, Z.; Jouandeau, N.; Cherif, A.A. A survey and analysis of multi-robot coordination. *Int. J. Adv. Robot. Syst.* **2013**, *10*, 399. [CrossRef]
5. Huang, Y.; Wu, S.; Mu, Z.; Long, X.; Chu, S.; Zhao, G. A Multi-agent Reinforcement Learning Method for Swarm Robots in Space Collaborative Exploration. In Proceedings of the 2020 6th International Conference on Control, Automation and Robotics (ICCAR), Singapore, 20–23 April 2020; pp. 139–144. [CrossRef]
6. Wang, Z.; Schwager, M. Multi-robot manipulation without communication. In *Distributed Autonomous Robotic Systems*; Springer: Tokyo, Japan, 2016; Volume 112, pp. 135–149.
7. Chand, S.; Verma, P.; Tallamraju, R.; Karlapalem, K. Transportation of deformable payload through static and dynamic obstacles using loosely coupled nonholonomic robots. In Proceedings of the International Symposium on Multi-Robot and Multi-Agent Systems, MRS, New Brunswick, NJ, USA, 22–23 August 2019; pp. 228–230.
8. Alonso-Mora, J.; Baker, S.; Rus, D. Multi-robot formation control and object transport in dynamic environments via constrained optimization. *Int. J. Robot. Res.* **2017**, *36*, 1000–1021. [CrossRef]
9. Bujarbaruah, M.; Stürz, Y.R.; Holda, C.; Johansson, K.H.; Borrelli, F. Decentralized 2-Robot Transportation with Local and Indirect Sensing. *arXiv* **2021**, arXiv:cs.RO/2103.04460.
10. Eoh, G.; Jeon, J.D.; Lee, B.H. Cooperative object transportation using virtual electric dipole field. *Int. J. Mech. Eng. Robot. Res.* **2016**, *5*. [CrossRef]
11. Sands, T. Development of Deterministic Artificial Intelligence for Unmanned Underwater Vehicles (UUV). *J. Mar. Sci. Eng.* **2020**, *8*, 578. [CrossRef]
12. Jhang, J.Y.; Lin, C.J.; Young, K.Y. Cooperative Carrying Control for Multi-Evolutionary Mobile Robots in Unknown Environments. *Electronics* **2019**, *8*, 298. [CrossRef]
13. Dai, Y.; Kim, Y.; Wee, S.; Lee, D.; Lee, S. Symmetric caging formation for convex polygonal object transportation by multiple mobile robots based on fuzzy sliding mode control. *ISA Trans.* **2016**, *60*, 321–332. [CrossRef] [PubMed]
14. Sirineni, Y.; Tallamraju, R.; Rawat, A.; Karlapalem, K. Decentralized Collision Avoidance and Motion Planning for Multi-Robot Deformable Payload Transport Systems. In Proceedings of the 2020 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR), Abu Dhabi, United Arab Emirates, 4–6 November 2020; pp. 154–161. [CrossRef]
15. Zhang, L.; Sun, Y.; Barth, A.; Ma, O. Decentralized Control of Multi-Robot System in Cooperative Object Transportation Using Deep Reinforcement Learning. *IEEE Access* **2020**, *8*, 184109–184119. [CrossRef]
16. Choi, D.; Kim, D. Genetic Fuzzy System-Based Multi-Robot Coordination for Planetary Missions. In Proceedings of the 31st AAS/AIAA Space Flight Mechanics Meeting, Virtual, 1–4 February 2021.
17. Sathyan, A.; Ma, O. Collaborative Control of Multiple Robots Using Genetic Fuzzy Systems. *Robotica* **2019**, *37*, 1922–1936. [CrossRef]
18. Jang, J.S. ANFIS: Adaptive-network-based fuzzy inference system. *IEEE Trans. Syst. Man Cybern.* **1993**, *23*, 665–685. [CrossRef]
19. Cordon, O.; Herrera, F.; Hoffmann, F.; Magdalena, L. *Genetic Fuzzy Systems: Evolutionary Tuning and Learning of Fuzzy Knowledge Bases*; Advances in Fuzzy Systems-Applications and Theory; World Scientific Publishing Company: Singapore, 2001.
20. Gu, J.; Cao, Q.; Huang, Y. Rapid Traversability Assessment in 2.5D Grid-based Map on Rough Terrain. *Int. J. Adv. Robot. Syst.* **2008**, *5*, 40. [CrossRef]
21. Schmidt, V. (Ed.) *Stochastic Geometry, Spatial Statistics and Random Fields*; Springer International Publishing: Berlin/Heidelberg, Germany, 2015.