drones

Article

Genetic Fuzzy Methodology for Decentralized Cooperative UAVs to Transport a Shared Payload

Anoop Sathyan *, Ou Ma and Kelly Cohen

Department of Aerospace Engineering and Engineering Mechanics, University of Cincinnati, Cincinnati, OH 45231, USA

* Correspondence: sathyaap@ucmail.uc.edu

Abstract: In this work, we train controllers (models) using Genetic Fuzzy Methodology (GFM) for learning cooperative behavior in a team of decentralized UAVs to transport a shared slung payload. The training is done in a reinforcement learning fashion where the models learn strategies based on feedback received from the environment. The controllers in the UAVs are modeled as fuzzy systems. Genetic Algorithm is used to evolve the models to achieve the overall goal of bringing the payload to the desired locations while satisfying the physical and operational constraints. The UAVs do not explicitly communicate with one another, and each UAV makes its own decisions, thus making it a decentralized system. However, during the training, the cost function is defined such that it is a representation of the team's effectiveness in achieving the overall goal of bringing the shared payload to the target. By including a penalization term for any constraint violation during the training, the UAVs learn strategies that do not require explicit communication to achieve efficient transportation of payload while satisfying all constraints. We also present the performance metrics by testing the trained UAVs on new scenarios with different target locations and with different number of UAVs in the team.

Keywords: decentralized control; reinforcement learning; multi-agent; genetic fuzzy methodology; evolutionary algorithm; UAV team; air transportation

check for updates

Citation: Sathyan, A.; Ma, O.; Cohen, K. Genetic Fuzzy Methodology for Decentralized Cooperative UAVs to Transport a Shared Payload. *Drones* **2023**, *7*, 103. https://doi.org/10.3390/drones7020103

Academic Editors: Sergio Salazar and Filiberto Muñoz

Received: 25 December 2022
Revised: 20 January 2023
Accepted: 1 February 2023
Published: 3 February 2023

1. Introduction

We apply the Genetic Fuzzy Methodology (GFM) to the complex problem of controlling a team of Unmanned Aerial Vehicles (UAVs) to transport a shared payload from one location to another in 3D space. The team is considered to be decentralized in that the vehicles make their own decisions rather than being controlled by a central controller. This has advantages in being more fault tolerant and have lower bandwidth requirements since the data transferred between vehicles is much lower compared to centralized systems. The UAVs are trained using GFM in an evolutionary fashion to achieve the overall team goal of transporting the payload to the desired location. This is an important aspect of this work as the agents need to learn to forego individual rewards to achieve the overall goal.

The ability to collaborate allows us as humans to tackle many complex tasks such as moving large objects, manufacturing, driving, team sports, etc. For all these collaborative tasks, humans could be trained together for improved efficiency. However, each person in the team makes their own decision that collectively allow the team to achieve the overall goal. Our research takes motivation from the importance of using collaboration to achieve complex tasks and how training autonomous agents to learn collaboration or cooperative behavior has the potential for significantly improving efficiency. The research objective of this work is to use GFM in a reinforcement learning manner to allow a team of autonomous agents to work cooperatively to achieve the overall goal of the team. In this paper, we apply this method for the specific use case of developing controller for team of UAVs to transport a shared payload. We also discuss scalability to cases with larger teams.

Collaboration or cooperative behavior is a very important aspect of multi-agent systems where a team of agents work together to achieve a common goal. Numerous applications of cooperative robotics have been investigated. These include moving objects [1–4], truck and drone working together for last-mile delivery [5], formation flying [6], observing moving targets [7], helping human workers in collaborative tasks [8,9], coordinated search activities [10], etc. Just like any multi-robot system, a number of UAVs working as a team can accomplish tasks a single UAV cannot possibly do, such as jointly transporting a large and heavy payload. Multi-UAV transport is an important dynamics and control problem that is discussed much in literature. Different techniques for solving this problem have been presented. For example, Linear Quadratic Gaussian controllers have been used to control the UAVs to transport slung payload for different payload masses [11]. Another paper presents the use of cascaded control architecture to transport payload that is modeled as rigid body [12]. The different sub-controllers within the cascaded architecture perform different roles such as path following, attitude control, etc. Model Predictive controllers have also been used for such systems. Wehbeh et al. [13] in addition presented a comparison between the performance of centralized and decentralized controllers. They showed that the decentralized controllers functioned worse when operated at the same frequency as the centralized controller, but did marginally better when scaled to frequency that allowed for similar computational cost as the centralized controller.

Applying machine learning methods to train such systems is very recent and holds a lot of potential as machine learning methods can come up with more improved strategies for such complex problems. Neural networks are popularly used to model such controllers. One of the works presented radial basis neural networks for trajectory tracking and obstacle avoidance while transporting payload using multiple UAVs [14]. Most of the works treat this use case as a reinforcement learning problem. This allows for the models to learn based on feedback from the environment and the controllers may learn interesting strategies to achieve the operational goal. Recent developments in reinforcement learning have tremendously improved the applicability of learning based systems for various single agent and multi-agent dynamic systems. Li et al. used reinforcement learning to come up with optimal trajectory planning that minimized load swing [15]. Another work presents a multi-agent proximal policy optimization (MAPPO) for controlling team of drones in simulated air combat tasks [16].

Genetic Fuzzy Methodology has also been applied to different collaborative automation applications. GFM has been used for controlling automated vehicles for cooperative merge in high density traffic in cases of fully automated as well as mixed traffic involving both human and automated vehicles [17,18]. In another work, we had used GFM to control decentralized cooperative robots in 2D space to transport objects to desired target locations. We expanded on this work to scale it to teams with different number of robots without requiring any additional training [19–21]. In another work, GFM was used to carry an object using two robots holding two ends of the object and transport it between rooms through a narrow opening [22,23]. We have also used GFM to develop a set of controllers that controlled UAVs to transport a connected payload. This involved training several genetic fuzzy systems (GFSs) that were separately trained for its own objectives. The individual GFSs had its own targeted behaviors such as pitch control, altitude control, yaw control, etc. [24,25].

In this work, we use GFM to train a single GFS that acts as the decision making system for each agent in the team. The training is targeted towards achieving the team's goal of bringing the payload to the target.

2. Problem Statement

The dynamics problem considered in this work involves a team of UAVs with a connected payload. The objective is that the UAV team needs to learn to make its own decentralized decisions without any knowledge of the decisions made by its teammates to bring the payload to desired target location. The importance of decentralized decision making without relying on information about other UAV's decisions is that this highly reduces the communication overhead required for this application. This also reduces the need to solve communication related challenges such as latency. The schematic of a UAV team with slung payload is shown in Figure 1. Training such a team is a challenging task and this tests the applicability of the GFS based reinforcement learning framework to train a decentralized system of UAVs to work collaboratively toward the common goal of bringing the payload to the target.

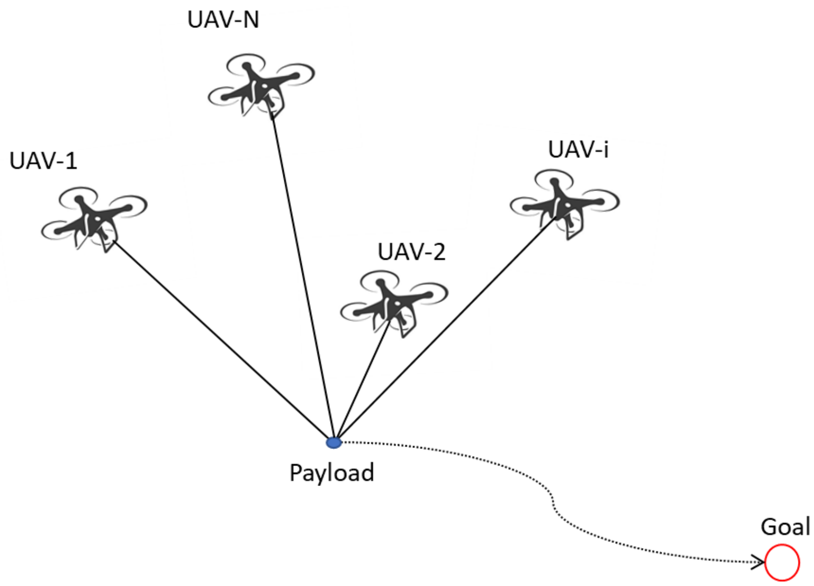


Figure 1. Schematic of the dynamics problem: Multiple UAVs need to cooperate to transport the shared payload connected through cables.

The objective of the GFM training is to come up with a strategy that allows the decentralized UAVs in the team to work collaboratively, even though it has no explicit knowledge of its teammates' future actions, to bring the payload to the target efficiently. Each UAV decides on the 3D thrust vector applied by it. It is assumed that the UAVs can apply thrust vector in any direction in the 3D space. Together, the team of UAVs have to apply appropriate thrusts to control the position of the payload. During the training process, the overall performance of the system will be used to evolve the team of UAVs. Thus, even though there is no communication between the UAVs, the UAVs can come up with strategies that allow implicit understanding of intent of its partners.

3. UAVs with Slung Payload

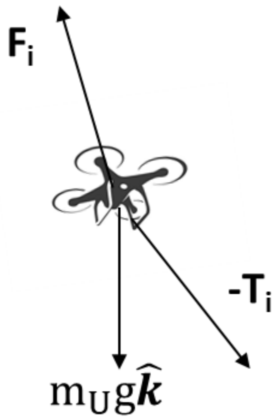
This section describes the physical dynamics of the UAV team with slung payload.

The UAVs and payloads are modeled as point objects with masses m_U and m_L, respectively. The forces acting on the UAV and the payload are represented in Figure 2. The UAVs can apply thrust $\mathbf{F_i}$ in any direction in the 3D space based on the recommendations provided by its own decentralized controller. A damping force, with damping factor μ, is considered to model the air resistance. Based on Figure 2a, we can write the force equation for the UAVs.

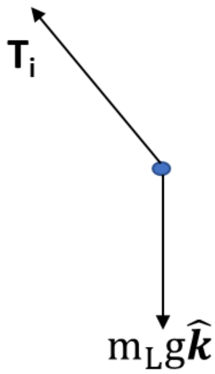
$$\boldsymbol{F_i} - \boldsymbol{T_i} - m_U g \hat{\boldsymbol{k}} - \mu \dot{\boldsymbol{r}}_{\boldsymbol{U}} = m_U \ddot{\boldsymbol{r}}_{\boldsymbol{U}} \tag{1}$$

$\boldsymbol{T_i}$ refers to the tension force vector in the cable connecting the UAV-i to payload. Since the cables are modeled as spring-dampers the following equation applies.

$$\boldsymbol{T_i} = k \boldsymbol{r_{LU}} + c \dot{\boldsymbol{r}}_{\boldsymbol{LU}} \tag{2}$$



(**a**) Forces on UAV-*i*



(**b**) Forces on the payload

Figure 2. Forces acting on the UAV and payload.

For the payload, the force equation can be written as follows based on forces shown in Figure 2b as well as the damping due to air resistance.

$$\sum_{i=1}^{N} \boldsymbol{T_i} - m_L g\hat{\boldsymbol{k}} - \mu\dot{\boldsymbol{r}}_{\boldsymbol{L}} = m_L\ddot{\boldsymbol{r}}_{\boldsymbol{L}} \tag{3}$$

4. Genetic Fuzzy Methodology

4.1. Fuzzy Logic Systems

Fuzzy logic uses degrees of truth, which are continuous values, unlike the the binary True or False values used by Boolean logic. Fuzzy Logic Systems are systems that use principles of fuzzy logic to define input-output relationships. FLSs are universal approximators that can be used to represent nonlinear functions to any arbitrary degree of accuracy. In an FLS, membership functions [21] are used to define each input and output variable. These membership functions provide a mechanism to convert the crisp variable values to fuzzy degrees of membership. If a variable has m membership functions, then a crisp 1-D value of the variable is converted to an m-D vector showing degrees of membership of the variable value to each membership function. Thus, the membership functions provide a 1-to-m mapping.

Triangular membership functions are commonly used and require the x-values of the left, center and right vertices to define it. The input-output relation is defined using a set of If-Then statements that connect the inputs to the output. As an example, for a 2-input-1-output system, a rule could be *If Input1 is mf1 and Input2 is mf3 Then Output is mf3*. To evaluate this rule, the membership values corresponding to mf1 for Input 1 and mf3 corresponding to Input 2 are evaluated. The fuzzy AND operation is used to combine the values to a single scalar value known as the rule firing value. This value is then multiplied (assuming product implication) with mf3 defined for the output.

A larger system with many more inputs would be have rules connecting all inputs to all the outputs. The same process can be used to evaluate each rule. The rule firing values are the combined the with the output membership functions from those rules. These are then aggregated to obtain an aggregated fuzzy output. This fuzzy output is then defuzzified to obtain a crisp real number as the output.

4.2. Training Process

In this work, we use GFS to model the decision making module of each UAV. The GFS module in each UAV uses the following 12 inputs to make decisions: (1–3) position vector of the payload relative to the UAV, (4–6) relative velocity of payload in 3D, (7–9) velocity of the UAV and (10–12) position vector of target relative to the UAV. The GFS needs to process these inputs, that are related to the states of the agent, payload and the target, and output a recommendation of the 3-D thrust vector in the agent's local reference frame attached to the UAV. The three axes of this local frame are along the vector from UAV to payload, along inertial z-axis and a vector perpendicular to these two axes. At each time step, the GFS module in each UAV is expected to recommend a 3D thrust vector that would efficiently help with transporting the payload to the target location. To achieve the necessary efficiency requirements, the GFS model needs to learn the appropriate relationship between the 12 inputs and near-optimal 3D thrust vector that need to be applied. This is achieved through the Genetic Algorithm (GA) based training process.

The schematic of the GFM training process is shown in Figure 3. Fuzzy Bolt© framework [17,26,27] provides an efficient training process for GA to tune the GFS model. Fuzzy Bolt© ensures that the training process for the GFS can efficiently handle cases with several inputs to the GFS. GA is used to tune the parameters related to the membership functions and the rulebase. The cost function is evaluated based on the performance of the model in the simulation environment. The cost function needs to be defined in such a way that it mathematically represents the requirements of the system. As seen from Figure 3, the GFM uses this cost value as feedback to try different strategies and improve the performance of the model.

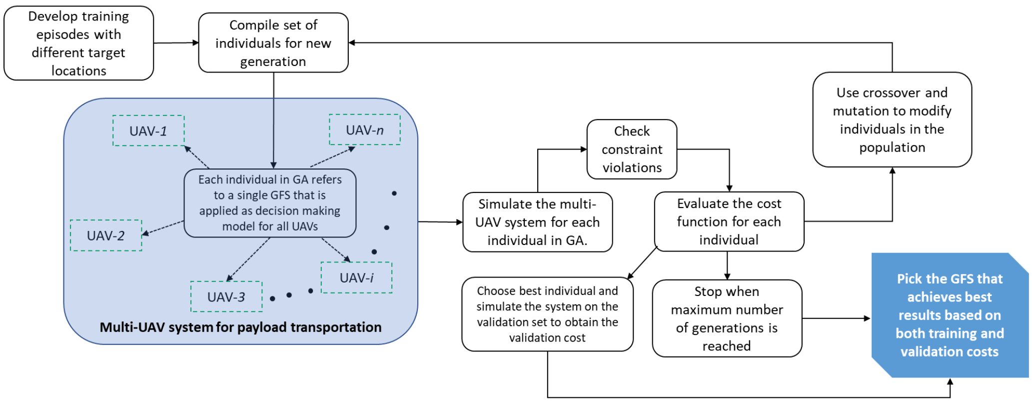


Figure 3. Schematic of the genetic fuzzy training process.

GA is a population-based approach. During each generation, the population in GA is a set of individuals. Each individual is a vector consisting of values of all paramters that need to be tuned in the GFS. This includes parameters related to shapes of input and output membership functions as well as parameters related to the rulebase. Thus, it is to be noted that an individual in GA does not refer to the individual agents and instead refers to a solution in the vector space of all parameters that are being tuned. Each individual in GA can be used to define a GFS. Since the same GFS is shared across all UAVs, each individual can be used to model the entire team of UAVs. For each GFS in the population, we can simulate the dynamics of the multi-UAV system using each individual in GA and evaluate the corresponding cost function (defined in Equation (4)) on the training scenarios. The cost values provide us information on the effectiveness of each individual (or corresponding GFS) in solving the multi-UAV problem.

For the GFS, we use triangular membership functions for the input and output variables. The x values of the boundaries of the membership functions are included in the set of parameters tuned by GA. All inputs and outputs are modeled using a maximum of five membership functions. The training process determines the actual number of membership functions used for each input and output variable. Additionally, the GFS uses product conjunction, product implication and summation to aggregate the output membership functions after evaluating the rulebase. Centroid defuzzification is used to determine the crisp output values.

As mentioned before, the cost function need to be defined such that it is representative of the ability of the system in achieving the goal. For this problem, since the objective is to bring the payload to the target while satisfying the constraints, the following cost function based on an integral of the distance between payload and target is used.

$$f = \int_0^{t_f} ||\mathbf{r}_{PT}||dt + \gamma C \tag{4}$$

where $t_f = \min(t_{fail}, t_{max})$. t_{max} is a pre-defined maximum time for an episode in the simulation environment and $\mathbf{r}_{PT}$ is the vector from the payload to the target. The simulation ends when any constraint is violated. t_{fail} refers to the first instance of constraint violation. In the second term, γ is a very high penalization factor and C is a variable that determines how early any constraint violations happen, if any. Thus, minimizing f would require minimizing the distance to the target without violation of any constraints.

At the beginning of the training process, the population is randomly generated. This means that the corresponding GFS produces random actions. This could even lead to constraint violations which could leave to high cost values for those individuals. In GA, the individuals with lower cost values have higher likelihood of being selected for crossover and mutation. Crossover and mutation are operations used by GA to modify the individuals in the population. During crossover, two chosen individuals are combined using random weight vector to generate a new individual. During a mutation, a small percent of the parameters in a chosen individual are randomly perturbed to generate a new individual. Mutation is mainly performed to reduce chances of becoming stuck in a local minima. In this work, we use uniform mutation. This means that, for any individual chosen for mutation, a certain number of parameters randomly chosen based on the mutation rate are changed to new values. Several such crossover and mutation operations are performed at the end of a generation to create new population for the next generation. The individuals with high cost values have higher chance of being eliminated from the population.

By giving more importance to individuals that perform better, GA produces populations that improve with each generation. This process of evolution continues for a predefined maximum number of generations. After each generation, the best GFS from the population is tested on a validation set to check for generalization to new scenarios. Once training is finished, the best individual that performs well on both the training and validation cost is chosen to design the multi-UAV system. This trained GFS will then be tested on a series of test scenarios with different target locations to ensure that the system works on a variety of scenarios. The hyperparamters of GA are listed in Table 1.

Table 1. The hyperparameters and the different operators used in our GA.

Hyperparameters	Values
Maximum number of generations	10,000
Population size	20
Crossover method	Intermediate crossover
Mutation method	Uniform mutation
Selection method	Stochastic universal sampling

5. Results

5.1. Training & Validation

Training scenarios were developed by defining target locations within 5 m from the starting location of the payload. The system starts from equilibrium where the UAVs are all positioned in a symmetric manner around the payload. This also means that the initial thrust vectors are defined such that they balance the weight of the payload. The GFS module that determines the 3D thrust vector was trained on teams with three UAVs on the training scenarios [28]. During the training, after each generation, the best individual or best GFS was applied to a smaller validation set and the validation cost was also evaluated. Tracking the validation cost along with the training allows us to make sure the GFS model is learning to generalize to new scenarios that are not part of the training set. Figure 4 shows the cost function on the training and validation scenarios for the best individual in each generation. It can be seen that the GFS model evolves and improves as it moves through the generations. The improvement in the validation score is an indication that the GFS model is able to generalize to new scenarios. Once the system was trained, it was tested on new scenarios to check its effectiveness and generalizing capability. Through the training process, the UAVs learn strategies that allow it to take actions that lead them towards the target without any information about the actions of the other UAVs in the team. Thus, the training process provides the team the capability to work together or cooperate to achieve the global goal of bringing the payload to the target location.

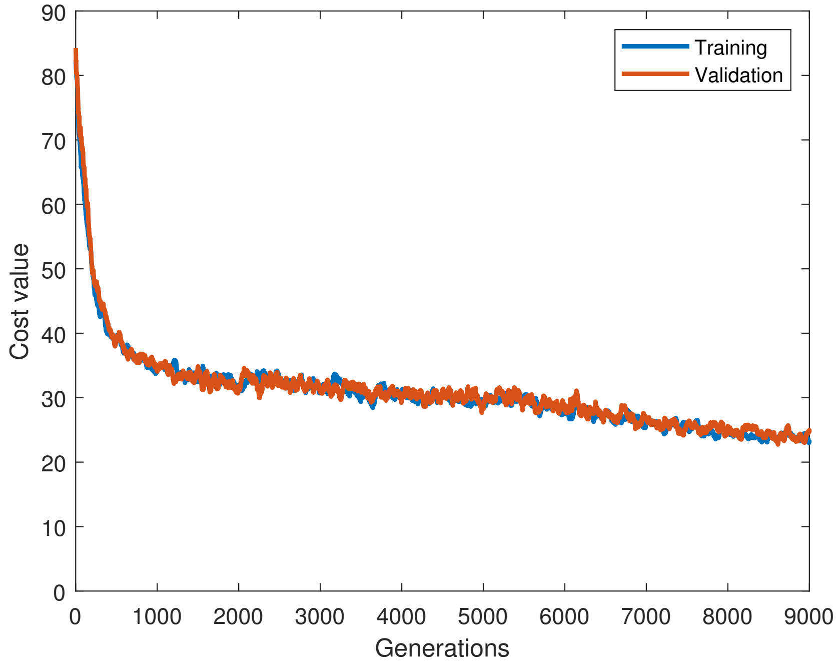
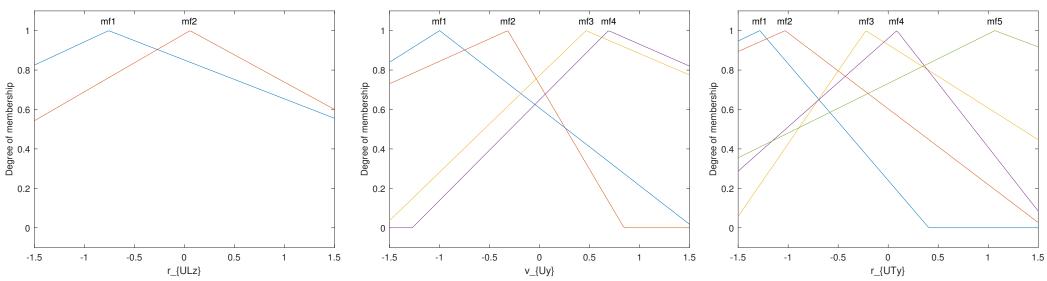


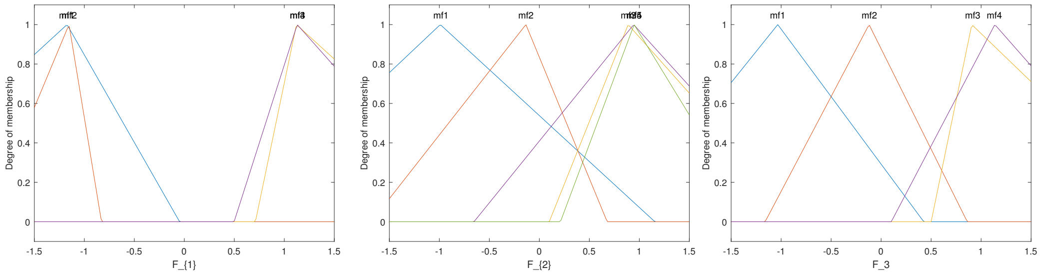
Figure 4. Training and validation costs during the training process [28].

The trained GFS model includes the membership functions for the different inputs and outputs. Figure 5 shows the input membership functions for three of the inputs. Similarly, the output membership functions related to the thrust vector in the UAV's local frame are shown in Figure 6. The variable values along the x axis are normalized values. It is to be noted that only the membership functions that are actually used by the GFS as defined in the rulebase are shown in the figures. For example, $r_U Lz$ (input 3) only uses $mf1$ and $mf2$ as seen from the rulebase shown in Table 2.



(**a**) r_{ULz} (**b**) v_{Uy} (**c**) r_{UTy}

Figure 5. Membership functions of the trained GFS model for three of the inputs.



(**a**) (**b**) (**c**)

Figure 6. Output membership functions of the trained GFS model. (**a**) F_1: Thrust component in the direction from UAV to payload. (**b**) F_2: Thrust component perpendicular to $F1$ and F_3. (**c**) F_3: Thrust component along the inertial z axis.

Table 2 shows the rulebase of the trained model. The trained model has 52 If-Then rules. The values in the table represent the indices of the membership function corresponding to the variable defined in each column. I1 through I12 are the 12 input variables for the GFS. F_1, F_2 and F_3 are the thrust vector components in the local frame of the UAV. As an example, the first rule can be read as: If input 1 is mf1 AND input 2 is mf2 AND input 3 is mf2 AND... AND input12 is mf5 THEN F_1 is mf1 , $F2$ is mf3, $F3$ is mf3. The output columns have zeros for some of the rules. This indicates that the output with rule consequent defined as zero can be ignored when evaluating that rule.At each instance, all 52 rules are evaluated and then aggregated based on the internal settings of the GFS described before to obtain the three output values for each UAV.

5.2. Testing the GFS Model

As mentioned in the previous subsection, the training was carried out on teams with three UAVs ($N = 3$). Additional test scenarios were generated by defining new target locations. 100 such test scenarios were used to test and evaluate the performance of the trained team of UAVs. The trained team of UAVs were able to bring the payload to the target for all cases. Figure 7 shows the payload path and the distance plot for two of the test scenarios with different target locations. The plots on the left show the trajectory of the payload as it is transported towards the target. The plots on the right show the decrease in distance between payload and target over time, which is the goal of the team. It can be seen from the plots that the team of UAVs can efficiently transport the payload even though each agent has no explicit information about the states or actions of other agents in the team. This testing further confirms the generalizability of the GFS model to new training scenarios defined using new target locations.

We used the same GFS model, with no additional training, to test and evaluate for teams with larger number of UAVs. It is to be noted that the payload mass was increased proportional to the increase in the number of UAVs. Figures 8–10 show payload paths and distance plots for two different target locations for different team sizes, $N = 5, 10$ and 25, respectively. The plots show that the UAVs can cooperate and efficiently bring the payload to the target location even for larger team sizes than the one considered during the training.

For each scenario, the UAV teams start from a symmetric topology where vehicles are distributed at equidistant angles around the payload. The UAVs break the symmetric pattern as they move towards the target location. The effectiveness of the multi-UAV team is evaluated using different figures of merit for different values of N on the 100 test scenarios. The settling time, the mean distance between payload and target after settling, and the minimum distance for each scenario were calculated. Table 3 summarizes the mean of these figures of merit across the 100 test scenarios, which clearly demonstrated the scalability of the trained control model.

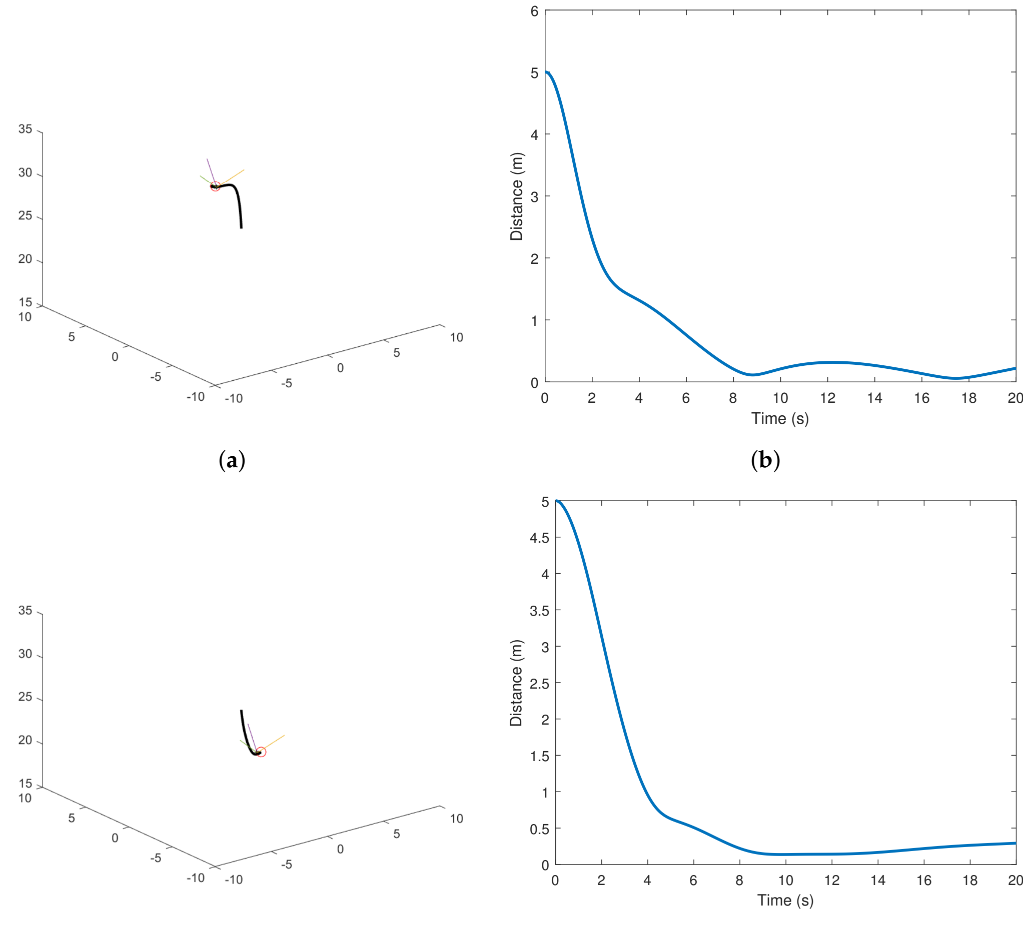
Table 2. Rulebase of the trained GFS model.

I1	I2	I3	I4	I5	I6	I7	I8	I9	I10	I11	I12	F_1	F_2	F_3
1	2	2	2	2	2	3	3	4	1	1	5	1	3	3
1	2	2	2	2	2	3	3	4	2	3	5	3	2	0
1	3	2	2	2	3	3	3	1	1	3	1	0	4	2
2	3	1	2	2	2	3	2	3	1	5	5	2	2	4
2	3	1	2	2	2	3	2	4	1	5	1	2	3	1
2	3	1	2	2	2	3	3	3	2	3	1	3	3	0
2	3	1	2	2	2	3	3	3	2	3	2	0	3	3
2	3	1	2	2	2	3	3	3	4	3	2	4	3	0
2	3	1	2	2	2	3	3	4	1	5	2	2	3	1
2	3	1	2	2	2	3	3	4	2	1	3	1	3	0
2	3	1	2	2	2	3	3	4	2	3	2	3	1	1
2	3	1	2	2	2	3	3	4	2	3	3	2	3	0
2	3	1	2	2	2	3	3	4	3	2	2	2	3	1
2	3	1	2	2	2	3	3	4	4	3	3	3	4	3
2	3	1	2	2	2	3	3	4	4	4	2	3	3	3
2	3	1	2	2	2	3	3	5	2	3	2	3	4	0
2	3	1	2	2	2	4	2	4	1	3	5	3	1	3
2	3	2	2	2	2	3	3	4	2	3	2	3	3	3
2	3	2	2	2	2	3	3	4	2	4	5	3	3	3
2	3	2	2	2	3	3	3	4	2	2	2	3	5	0
2	3	2	2	3	2	3	2	4	1	3	1	1	0	1
4	1	1	1	2	2	3	2	3	2	4	5	3	3	4
4	1	1	1	2	2	3	3	3	1	5	5	2	3	4
4	1	1	1	2	2	3	3	4	4	1	1	3	4	1
4	1	1	1	2	2	3	3	4	4	5	1	3	3	0
4	1	1	1	2	2	4	2	3	1	5	5	0	2	3
4	1	1	1	2	2	4	2	4	1	3	1	3	1	0
4	1	1	1	2	3	3	3	4	4	1	2	0	3	0
4	5	1	2	2	3	3	1	3	2	5	2	2	4	3
5	1	1	1	2	2	3	2	4	2	1	5	4	2	3
5	1	1	1	2	2	3	2	4	4	1	2	3	4	3
5	1	1	1	2	2	3	3	4	3	1	2	0	4	0
5	1	1	1	2	2	3	3	4	5	3	2	3	3	0
5	1	1	1	2	2	4	2	4	1	1	5	1	2	1
5	1	1	1	2	2	4	2	4	2	1	1	2	2	0
5	1	2	1	2	2	3	3	2	5	1	1	0	2	3
5	1	2	1	2	2	3	3	3	2	1	1	0	0	3
5	1	2	1	2	2	3	3	3	4	1	1	2	2	3
5	1	2	1	2	2	3	3	4	4	1	5	1	2	3
5	1	2	1	2	2	3	3	4	4	2	5	3	4	0
5	3	1	2	2	2	3	2	2	1	5	5	2	3	4
5	3	1	2	2	2	3	2	3	2	5	5	3	3	3
5	4	1	2	2	2	3	3	3	1	5	5	0	2	4
5	5	1	2	2	2	3	2	4	4	5	5	2	3	3
5	5	1	2	2	2	3	3	3	5	5	1	2	4	3
5	5	1	2	2	2	3	4	4	4	3	2	3	1	0
5	5	1	2	2	2	3	4	4	5	5	2	2	2	2
5	5	1	2	2	2	4	2	3	1	5	5	3	1	4
5	5	1	2	2	2	4	3	4	2	5	1	3	3	1
5	5	2	2	2	2	3	3	1	4	5	1	0	4	0
5	5	2	2	2	2	3	3	4	4	5	1	2	1	1
5	5	2	2	2	2	3	3	5	4	4	1	0	2	0

The GFS model is able to scale well for teams with more number of UAVs, provided the payload mass (m_L) is also proportionally increased. This is a very interesting result as no additional training was required to work on these cases with different team sizes. This allows us to utilize the same model and presents a good advantage of using decentralized control. Since the GFS models were using decentralized information related to the respective agents, it allows us to use the same model on the different UAVs even for cases where more UAVs were added to the team. It can be seen from Table 3 that the settling times and average distance after settling are very similar across the different team sizes. The minimum distance is noticed to be lower for $N = 3$ compared to larger team sizes. However, it is to be noticed that the cost function (Equation (4)), used for training, rewards reaching and staying close to the target rather than just minimizing the distance at a specific timestep. Since the original training was carried out on $N = 3$, it would be interesting to see how the model would perform if the training set included scenarios with different team sizes. Transfer learning, where only certain parameters of the GFS model are tuned, can also be used when training for different team sizes.

Table 3. Figures of merit on the test scenarios for verifying scalability.

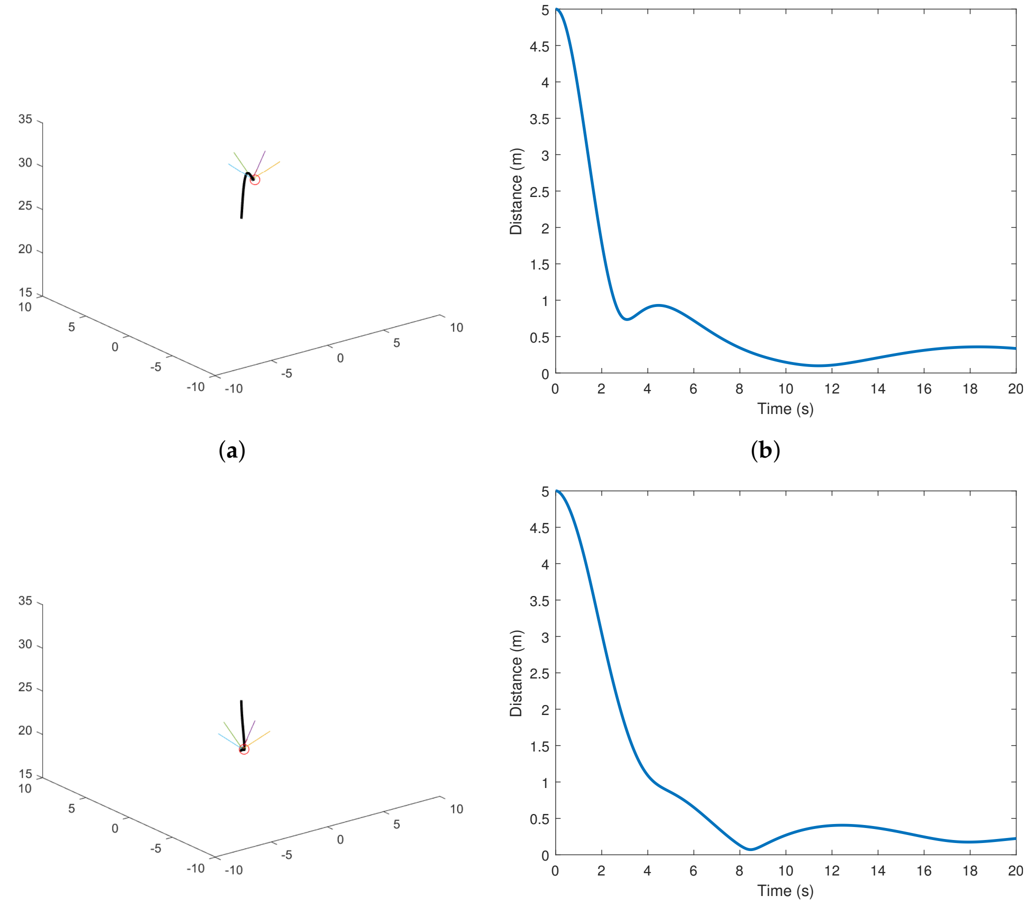
Team Size	Mean Settling Time (s)	Mean of the Average Distance after Settling (m)	Mean of the Minimum Distances (m)
$N = 3$	17.08	0.3522	0.1691
$N = 5$	17.82	0.3947	0.2463
$N = 10$	17.32	0.3898	0.2519
$N = 25$	17.39	0.3959	0.2564



(**a**) (**b**)

(**c**) (**d**)

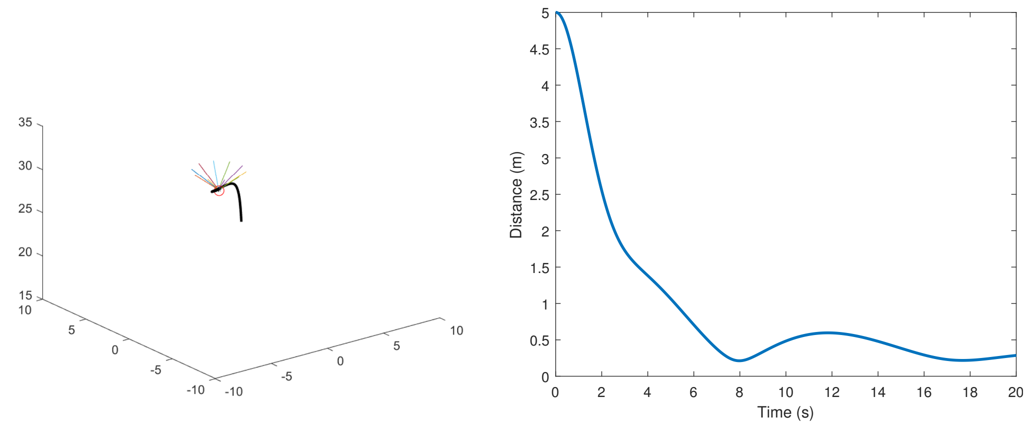
Figure 7. (**a**–**d**) Team with 3 UAVs: Payload path and distance plots for 2 test scenarios.



(**a**) (**b**)

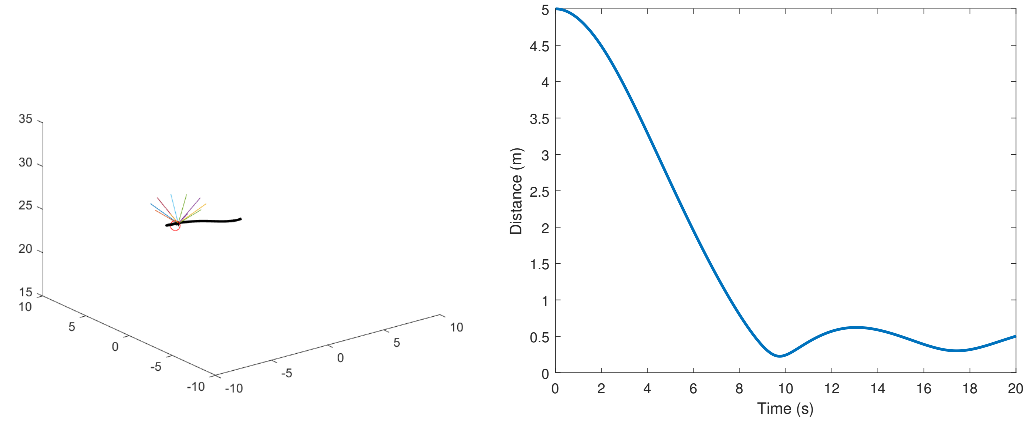
(**c**) (**d**)

Figure 8. (**a**–**d**) Team with 5 UAVs: Payload path and distance plots for 2 test scenarios.



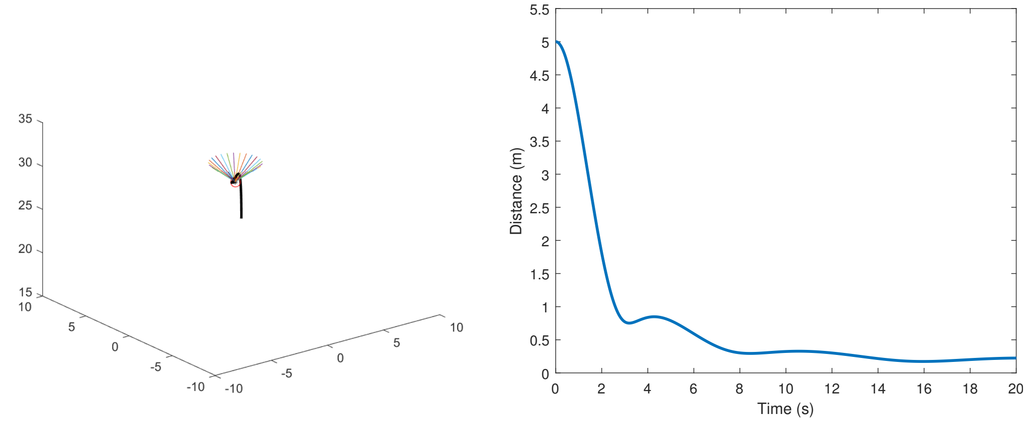
(**a**) (**b**)

Figure 9. *Cont.*

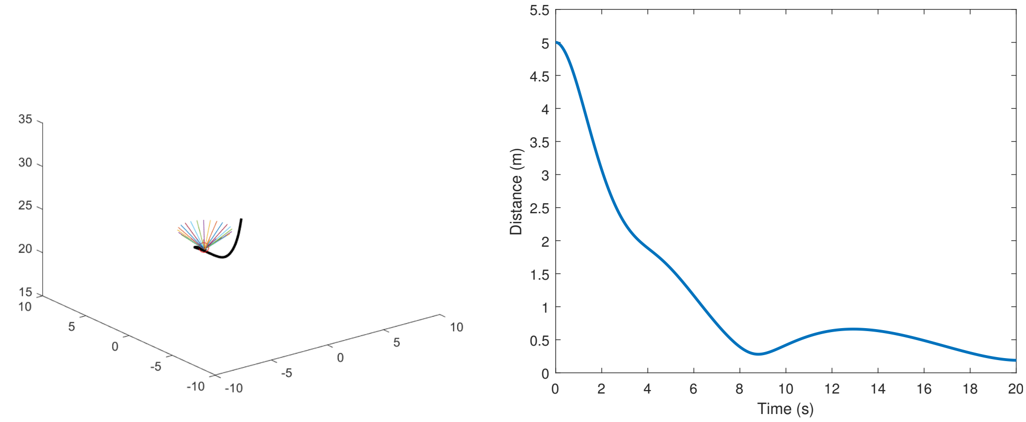


(**c**) (**d**)

Figure 9. (**a**–**d**) Team with 10 UAVs: Payload path and distance plots for 2 test scenarios.



(**a**) (**b**)



(**c**) (**d**)

Figure 10. (**a**–**d**) Team with 25 UAVs: Payload path and distance plots for 2 test scenarios.

The cost function in Equation (4) uses a global reward scheme as it uses the state of the payload rather than those of individual UAVs. During the training, even though the UAVs are making their own decisions, the global cost value rewards cases where the overall goal is important. This encourages learning cooperative behaviors in the system. As seen from the performance across 100 test scenarios and for different team sizes, it is to be understood that the trained UAVs have learned cooperative behaviors to transport the payload while also satisfying the different physical and operational constraints of the system.

6. Conclusions

In this work, we applied GFM to train a decentralized model that makes decisions for agents in a multi-UAV team to cooperatively transport a payload. GFS uses a set of input variables that are defined in the local frame of reference attached to each UAV. This allows the GFS to make decentralized decisions about the 3D thrust vector (output from GFS) that need to be applied at any instant. The team of UAVs is trained to bring the payload to different target locations. GA tunes the parameters of the GFS module which includes parameters related to the membership functions for the input and output variables as well as the parameters related to the If-Then rules in the rulebase that define the relationship between the inputs and outputs. During the training process, GA tunes the parameters to minimize a cost function designed to reward bringing the payload to the target. Physical and operational constraints such as maximum stretch of cables, safe distance between UAVs, no explicit communications between UAVs are enforced. Any constraint violation is penalized during the training process allowing for the UAV agents to learn cooperative strategies that would efficiently bring the payload to the target while also satisfying these constraints.

After training, the GFS module was evaluated on 100 new test scenarios with different target locations. The trained module was also tested on cases with different team sizes. It was noticed that the UAVs were able to successfully bring the payload to the target location even for cases with different team sizes. The team was also able to maintain it close to the target location within a distance of approximately 0.4 m. When scaling to larger teams, the payload mass (m_L) was also proportionally scaled. In the future, we plan to analyze the dynamics of the system to get a better understanding of how GFS models trained for a specific dynamics parameters such as UAV mass, payload mass, spring constant, etc work in a certain way. This will help us in scaling to different setups where the same trained model could be applied to various cases with different values of the dynamics parameters without having to explicitly train for those changes.

Future work will also include experimental validation using actual hardware. In addition, we plan to test the systems on cases with moving targets.

Author Contributions: Conceptualization, A.S. and O.M.; methodology, A.S., O.M. and K.C.; software, A.S. and K.C.; validation, A.S. and O.M.; investigation, A.S.; resources, A.S., O.M. and K.C.; writing, A.S.; review and editing, O.M. and K.C.; visualization, A.S.; supervision, O.M. and K.C.; project administration, O.M. and K.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data related to this work are available on request to the corresponding author.

Conflicts of Interest: Genexia LLC owns rights to the Fuzzy Bolt© algorithm. With respect to Fuzzy Bolt©, You accept and agree to be bound by the terms and conditions, which may be found at https://creativecommons.org/licenses/by-nc-nd/4.0/legalcode, accessed on 24 December 2022, of the Attribution-NonCommercial-NoDerivatives 4.0 International Public License (CCBY-NC-ND 4.0) ('Public License'). To the extent this Public License may be interpreted as a contract, You are granted the Licensed Rights in consideration of Your acceptance of the terms and conditions, and Genexia LLC grants You such rights in consideration of benefits Genexia LLC receives from making the Licensed Material available under the terms and conditions. Sathyan and Cohen are co-founders of Genexia, LLC. and hold equity ownership in the company.

Abbreviations

The following abbreviations are used in this manuscript:

UAV	Unmanned Aerial Vehicle
GA	Genetic Algorithm
GFS	Genetic Fuzzy System
GFM	Genetic Fuzzy Methodology

References

1. Tuci, E.; Alkilabi, M.H.; Akanyeti, O. Cooperative object transport in multi-robot systems: A review of the state-of-the-art. *Front. Robot. AI* **2018**, *5*, 59. [CrossRef] [PubMed]
2. Lee, T.; Sreenath, K.; Kumar, V. Geometric control of cooperating multiple quadrotor UAVs with a suspended payload. In Proceedings of the 52nd IEEE Conference on Decision and Control, Florence, Italy, 10–13 December 2013; pp. 5510–5515.
3. Gimenez, J.; Salinas, L.R.; Gandolfo, D.C.; Rosales, C.D.; Carelli, R. Control for cooperative transport of a bar-shaped payload with rotorcraft UAVs including a landing stage on mobile robots. *Int. J. Syst. Sci.* **2020**, *51*, 3378–3392. [CrossRef]
4. Gimenez, J.; Gandolfo, D.C.; Salinas, L.R.; Rosales, C.; Carelli, R. Multi-objective control for cooperative payload transport with rotorcraft UAVs. *ISA Trans.* **2018**, *80*, 491–502. [CrossRef] [PubMed]
5. Weng, Y.Y.; Wu, R.Y.; Zheng, Y.J. Cooperative Truck–Drone Delivery Path Optimization under Urban Traffic Restriction. *Drones* **2023**, *7*, 59. [CrossRef]
6. Zhang, J.; Sheng, H.; Chen, Q.; Zhou, H.; Yin, B.; Li, J.; Li, M. A Four-dimensional space-time automatic obstacle avoidance trajectory planning method for multi-UAV cooperative formation flight. *Drones* **2022**, *6*, 192. [CrossRef]
7. Khan, A.; Rinner, B.; Cavallaro, A. Cooperative robots to observe moving targets. *IEEE Trans. Cybern.* **2016**, *48*, 187–198. [CrossRef]
8. Realmuto, J.; Warrier, R.B.; Devasia, S. Data-inferred personalized human-robot models for iterative collaborative output tracking. *J. Intell. Robot. Syst.* **2018**, *91*, 137–153. [CrossRef]
9. Roveda, L.; Haghshenas, S.; Caimmi, M.; Pedrocchi, N.; Molinari Tosatti, L. Assisting operators in heavy industrial tasks: On the design of an optimized cooperative impedance fuzzy-controller with embedded safety rules. *Front. Robot. AI* **2019**, *6*, 75. [CrossRef]
10. Baranzadeh, A.; Savkin, A.V. A distributed control algorithm for area search by a multi-robot team. *Robotica* **2017**, *35*, 1452–1472. [CrossRef]
11. Lee, H.I.; Yoo, D.W.; Lee, B.Y.; Moon, G.H.; Lee, D.Y.; Tahk, M.J.; Shin, H.S. Parameter-robust linear quadratic Gaussian technique for multi-agent slung load transportation. *Aerosp. Sci. Technol.* **2017**, *71*, 119–127. [CrossRef]
12. Qian, L.; Liu, H.H. Path following control of multiple quadrotors carrying a rigid-body slung payload. In Proceedings of the AIAA Scitech 2019 Forum, San Diego, CA, USA, 7–11 January 2019; p. 1172.
13. Wehbeh, J.; Rahman, S.; Sharf, I. Distributed model predictive control for UAVs collaborative payload transport. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 24–29 October 2020; pp. 11666–11672.
14. Aliyu, A.; El Ferik, S. Control of Multiple-UAV Conveying Slung Load With Obstacle Avoidance. *IEEE Access* **2022**, *10*, 62247–62257. [CrossRef]
15. Li, X.; Zhang, J.; Han, J. Trajectory planning of load transportation with multi-quadrotors based on reinforcement learning algorithm. *Aerosp. Sci. Technol.* **2021**, *116*, 106887. [CrossRef]
16. Zhan, G.; Zhang, X.; Li, Z.; Xu, L.; Zhou, D.; Yang, Z. Multiple-UAV Reinforcement Learning Algorithm Based on Improved PPO in Ray Framework. *Drones* **2022**, *6*, 166. [CrossRef]
17. Sathyan, A.; Ma, J.; Cohen, K. Decentralized cooperative driving automation: A reinforcement learning framework using genetic fuzzy systems. *Transp. B Transp. Dyn.* **2021**, *9*, 775–797. [CrossRef]
18. Xu, R.; Guo, Y.; Han, X.; Xia, X.; Xiang, H.; Ma, J. OpenCDA: An open cooperative driving automation framework integrated with co-simulation. In Proceedings of the 2021 IEEE International Intelligent Transportation Systems Conference (ITSC), Indianapolis, IN, USA, 19–22 September 2021; pp. 1155–1162.
19. Sathyan, A.; Ma, O. Collaborative control of multiple robots using genetic fuzzy systems approach. In *Dynamic Systems and Control Conference*; American Society of Mechanical Engineers: New York, NY, USA, 2018; Volume 51890, p. V001T03A002.
20. Sathyan, A.; Ma, O. Collaborative control of multiple robots using genetic fuzzy systems. *Robotica* **2019**, *37*, 1922–1936. [CrossRef]
21. Sathyan, A.; Cohen, K.; Ma, O. Genetic fuzzy based scalable system of distributed robots for a collaborative task. *Front. Robot. AI* **2020**, *7*, 601243. [CrossRef] [PubMed]
22. Sun, Y.; Barth, A.; Ma, O. An intelligent approach for a two-robot team to perform a cooperative task. In Proceedings of the AIAA Scitech 2020 Forum, Orlando, FL, USA, 6–10 January 2020; p. 1116.
23. Barth, A.; Sun, Y.; Zhang, L.; Ma, O. Genetic fuzzy-based method for training two independent robots to perform a cooperative task. *Intell. Serv. Robot.* **2021**, *14*, 535–548. [CrossRef]
24. Bisig, C.; Montejo, J.B.; Verbryke, M.R.; Sathyan, A.; Ma, O. Genetic Fuzzy Systems for Decentralized, Multi-UAV Cargo Handling. In Proceedings of the AIAA Scitech 2020 Forum, Orlando, FL, USA, 6–10 January 2020; p. 1117.

25. Bisig, C.; Ma, O. Scalable and Robust Decentralized Cargo Transport via Genetic Fuzzy Systems. In Proceedings of the AIAA SCITECH 2022 Forum, San Diego, CA, USA, 3–7 January 2022; p. 1728.
26. Sathyan, A.; Harrison, H.S.; Kiefer, A.W.; Silva, P.L.; MacPherson, R.; Cohen, K. Genetic fuzzy system for anticipating athlete decision making in virtual reality. In *International Fuzzy Systems Association World Congress*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 578–588.
27. Mayadevi, B.; Martis, D.; Sathyan, A.; Cohen, K. Predictive Maintenance of Aircraft Engines Using Fuzzy Bolt©. In *North American Fuzzy Information Processing Society Annual Conference*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 121–128.
28. Sathyan, A.; Ma, O. Genetic Fuzzy Methodology for Decentralized Multi-UAV Cooperation with Physical Constraints. In Proceedings of the AIAA Scitech 2023 Forum, National Harbor, MD, USA, 23–27 January 2023.