Concurrent Morphology-Optimization and Behavior-Learning: Co-Designing Intelligent Quadcopters

Amir Behjat*, Chen Zeng†, Krushang Gabani‡, and Souma Chowdhury§
University at Buffalo, Buffalo, NY, 14260

In this paper, a novel co-design process is proposed to accomplish the coupled objectives of maximizing the flight range and learning capacity of a quadcopter unmanned aerial vehicle (UAV), with the end goal of producing a UAV that can learn to fly bottoms up (pun intended). The approaches presented here are motivated by the observed coherence in the evolution of morphology and behavior (or rather the capacity to learn behavior) in nature. The co-design process requires concurrent execution of the following two processes: i) optimization of the UAV morphology (e.g., geometry, component choices, etc.) and ii) learning of the flight controller to achieve take-off and preliminary hovering. In order to decrease the computational burden of applying the learning process nested under each morphology design, a novel metric called "Talent" is defined, which estimates the learning capacity of the system based on the UAV morphology and control theory. A concept of "Capacity Variables" is also introduced to help in reducing the dimension of the design variable space, and enable swift inverse design. Multi-objective optimization and neuroevolution processes are respectively used to perform the multi-level design and validate the learning capacity of the UAV designs produced thereof. The results of our co-design framework demonstrate the trade-offs accomplished in terms of UAV range performance and talent. More importantly, we show that through coherent morphology-talent optimization, it is possible to produce UAVs that can learn to fly (in this case, take off and hover briefly) with no prior guidance, which is in itself a major step forward in designing autonomous aerial robotic systems.

I. Nomenclature

C_1, C_2	=	tuning factors in talent function
f_t	=	talent objective function
f_r	=	flight range
k_i	=	reward coefficients; i= 1,2,3,4
$\|l\|$	=	distance from the initial position
$\|l\|_0$	=	distance from the reference position
N_r	=	number of reward terms
N_a	=	number of states
$R(t)$	=	reward function
R_i	=	direction of the controlling reward
s	=	state at the time step t
T	=	talent objective function
t	=	time step
t_f	=	final time step
$\|V\|$	=	UAV velocity
W_c	=	Controllability Gramian
x_i	=	capacity variables ; i=1,2,...4
z, z_{des}	=	height and the desired height

*Ph.D. Candidate, Department of Mechanical and Aerospace Engineering, AIAA student member.
†Ph.D. Student, Department of Mechanical and Aerospace Engineering, AIAA student member.
‡Ph.D. Student, Department of Mechanical and Aerospace Engineering, AIAA student member.
§Assistant Professor, Department of Mechanical and Aerospace Engineering, AIAA senior member. Corresponding author. Email: soumacho@buffalo.edu

z_i	=	design variables ; i=1,2,...5
σ_i	=	eigenvalue corresponding to that eigenvector
ζ_i	=	eigenvector of Gramian of controllability

II. Introduction

Design processes that involve the collaboration of different aspects and systems, or "Co-Design"[1], are widely adopted by various industries. Co-Design problems have been widely presented in control, fluids, biomechanics, and various fields, as well as growing usages in robotics. The principles of co-design emphasize on dealing with conflicting design interests or solving recursive constraints. Often, the design problem can be (or must be) presented as a PDE-constrained optimization problem, regardless of the field of application.

Various Co-design methods are proposed to tackle numerous types of design problems. From the perspective of workflow, there are two major categories of methods: 1) the nested approach[2] and 2) the collaborative approach[3]. The nested approach is robust and is confident that the global optimum can be reached[4, 5]. Therefore, it is widely used in structure-control co-design problems like FPGA development. The collaborative approach can provide feasible and "reasonably optimized" within limited resources and, therefore, best suited for large scale industry engineering problems with multiple aspects. New theories and methods aiming to represent co-design problems more efficiently in rigorous mathematical terms have been proposed recently[6–9].

Co-design procedures, especially with the nested approach, required a large amount of design function evaluations (experiments or simulations), making it very expensive and slow to solve the problem "as-is." This is due to the strong coupling among the design parameters and conflicts of the quantities of interest. However, we can evade the couplings of some (or all) design parameters and avoid fully nested procedures if we re-formulate the Co-design problem in the form of a PDE-constrained optimization problem.

Optimization of the morphology and flight intelligence based on proven UAV designs is a practical idea to improve performance and reliability. The optimization does not face solving complexities like designing unconventional configurations. This idea is adopted in this paper by designing a UAV with aerodynamic morphology and flight intelligence design while maintaining similarities to the mainstream quadcopters.

In this paper, we propose a non-nested method for PDE-constrained Co-design problems, featuring an intermediate-variable-based approach that reduces the couplings among the design parameters. A quadcopter morphology-learning Co-design problem is featured in this paper as an example problem. In the example problem, the continuous and categorical design variables of the morphology are represented as a series of intermediate variables, or "capacity variables." A two-step non-nested optimization process is applied around the capacity variables. An evaluation metric based on Controllability Gramian is introduced in the case study to reduce evaluation cost further.

A. The Co-Design Principles For UAV Design

In the previous research [10], we demonstrated an approach based on the nested principle that optimizes the physical performance and learning capacity of an aerodynamically shaped quadcopter. The co-design framework featured a Particle Swarm Optimization[11] algorithm that optimizes a neural network as the learning procedure, as well as an NSGA-II[12] dual-objective optimization that optimizes the morphology parameters and the learning hyper-parameters.

Some shortcomings emerged from this research, namely the design variable space and the learning efficiency. The design variable space is mixed with continuous and discrete parameters, limited not only the choice of alternative learning approaches but also restrained the room of feasible design candidates. In this paper, the discrete design parameters are represented with continuous intermediate parameters. A new learning approach can be applied to the revamped design problem.

We also studied a relatively high-level intelligence system[13]. In that work, we studied the collision avoidance based on TRACE framework[14]. While this was a relatively complex problem, in the current paper, we used a lower-level decision-making system. Instead of looking at decisions at a higher level, we study the UAV "Takeoff and Hover" problem and consider the motor Thrusts as the system's action list. These changes help us to increase the correlation between morphology and learning problems and also helps us to study our method more precisely.

In the case of complex continuous action spaces, Neuroevolution [15] can be utilized to solve both deterministic [16] and stochastic [17] problems. Also, Neuroevolution uses a gradient-free approach, which can increase the flexibility to design the rewards. The special variation of the neuroevolution method used in this paper is the AGENT [18] method, which is specialized to avoid premature convergence.

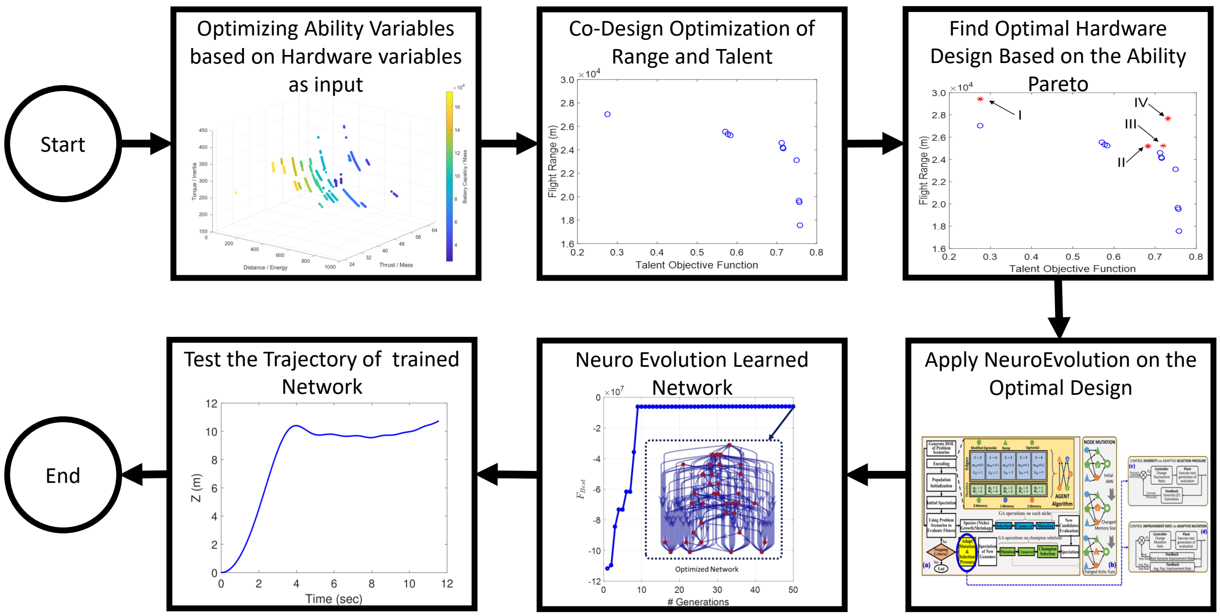


Fig. 1 Flowchart of the overall optimization framework

While using Neuroevolution has many advantages compared to gradient-based methods like PPO [19] for policy search, it is computationally more prohibitive. To tackle this problem, we used a novel method of bridging control theory and machine learning. In our approach, we use the controllability Gramian matrix to specify the best morphology design for learning in the feature. We argue that eigenvalues and eigenvectors of the controllability Gramian matrix must align with our desired reward control direction to provide the best learning performance in the feature. We define a parameter called "Talent" as our co-optimization objective.

The remaining portion of this paper is structured in the following manner: Section III introduces the overall framework and the procedure of quantifying morphology design with intermediate variables; Section IV brings a detailed explanation of the designed problem and all of the concepts utilized in it; Section VI presents the co-optimization problem formulation and explains the solving methods and process; Section VII lists the expected results. And finally section VIII concludes the paper.

III. The Example Problem with Morphology and Learning Co-Design

The proposed co-design framework comprises of two-level optimizations. The higher-level optimization uses the NSGA-II solver to optimize the capacity variables used to represent the morphology variables. These variables will be explained in detail in Section III.B. The lower level optimization is used to optimize the state-to-action model using the policy gradient reinforcement learning method with Proximal Policy Optimization. Figure 1 illustrates the workflow of the proposed co-design framework. The first step is the optimization of capacity variables. The second step is to apply the co-optimization. Then we use the Pareto frontier of capacity variables is used to find the best design sets. These sets are used as hardware setup for parallel Neuroevolution optimizations (step 4). Then the optimal network is chosen from the optimal answers of neuroevolution in step and is used in application as in the last step.

A. UAV morphology design

Introduced in our previous research [10], this quadcopter utilizes an H-shaped frame with an aerodynamic Blended-Wing-Body (BWB). The BWB effectively reduces the drag and creates some lift during the cruising flight. The proposed quadcopter design provides a practical lift during cruising flight without creating a significant drag, thus contributes to better endurance. Figure 2 illustrates the configuration of the aircraft.

Numerical models have been created to estimate the mass, aerodynamics, power-thrust correlations, and flight endurance of the quadcopter.

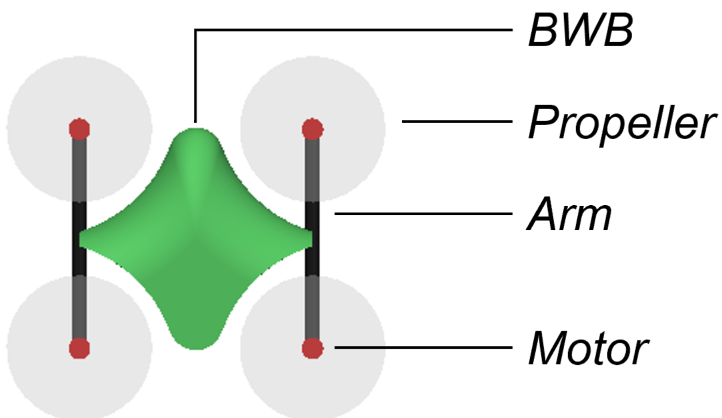


Fig. 2 The CAD model of the BWB-integrated quadcopter

B. Reduction of the Morphology Variables

The morphology variables include the morphology choices and geometry dimensions. The original morphology design variables include the dimensions of all the structures, plus the choice of motors, propellers, and battery. The cardinalities of these variables are extensively high, and some of them are discrete or categorical variables. The mixed variables will make the co-design process harder, especially with the learning process, as part of optimization. To avoid the mentioned issues, we use the "capacity variables," which are the functions of the morphology variables with lower cardinalities. Using the capacity variables, we can guarantee that the optimization problem becomes a continuous optimization problem with a reasonable size.

In the quadcopter design, the total thrust and the induced lift in the nominal flight state can be used as the capacity variables. The UAV arm length, type of the motors, and battery position are some of the morphology design variables. The complete list of these Variables is listed in table 2.

C. The Required Properties of The capacity variables

While using the capacity variables can help the optimization process, it is hard to guarantee that there is an actual morphology design to achieve these capacity variables. Therefore the problem of UAV design leads to an inverse design problem, which is not easy to solve. It is not always provable that such a design exists, especially in the presence of discrete and categorical variables. The typical approach in these conditions is using the most similar morphology design, which is common for turbine blade design[20]. Although this method is used in many applications, there is no guarantee of the amount of the error. Here we propose a novel idea to tackle the approximation problem and skip the inverse design problem in general. Our method ensures a design with the same properties of optimal capacity variables, or better than that.

The proposed approach is based on having the "Direct Non-Adverse Effect" on all of the desired tasks. For instance, The battery capacity increases the range of flight directly and has no direct impact on maneuverability. However, enlarging the battery capacity will increase the takeoff weight and may indirectly have an adverse effect on maneuverability. Our method can handle this issue because the desired mass is optimized as the function of morphology design variables.

The morphology design parameters in our example problem are a mixture of continuous and discrete variables, with extensive couplings among each other and concerning design objectives (long flight range Vs. quick learning). To reduce the couplings of the design parameters, we introduce a group of intermediate variables, as we call "Capacity Variables." The capacity variables are constructed following principles:

1) The capacity variables represent all vital physical properties corresponding to the design objectives and constraints.
2) The physical properties represented by the capacity variables should be **monotonically** correlated to the design objectives and/or constraints. Satisfying this condition allows us to largely ignore reverse design issues during co-optimization.
3) A sufficient combination of capacity variables can represent the contributions from all of the design variables.

4) The capacity variable space should not exceed the design variable space. That is, the range and constraints of the capacity variables should be limited so that no infeasible solution is present.
5) Once a set of capacity variables satisfy all the principles mentioned above, additional capacity variables are redundant and unnecessary. Having redundant capacity variables may introduce additional couplings to the design space.

D. Classification of Design Feasibility Using the Pareto Front

The capacity variables must have a non-adverse effect on each desired task (namely the flight endurance and maneuverability tasks). This condition helps define a simple way to find a feasible solution and avoid the inverse design problem. For this purpose, we firstly assume that we define each of these capacity variables as a function of actual morphology variables. Solving the multi-objective optimization on this function (with the morphology variables as design variables and the capacity variables as objectives) results in a Pareto front, which can be used as a classifier to find the inverse design. Any optimized capacity variables of the morphology can be substituted with a point in Pareto front, which has the better or equal capacity variables compared to the optimized capacity variables (from Co-Design problem).

E. Introducing capacity variables to morphology design

For the example problem, the design variables associated to the UAV morphology are:
- Frame Length
- Frame Width
- Motor Type
- Propeller Type
- Battery Type

The later 3 variables are categorical variables, which increase the difficulty for optimization without using the concept of capacity variables.

As emphasized previously, the capacity variables should have monotonic effects on both of the design objectives. It is not always possible to find the parameters directly. In this paper, we will solve this issue by substituting these variables with the ratio parameters of these variables. The capacity variables for both tasks are:

- $\frac{\texttt{Battery Capacity}}{\texttt{UAV Mass}}$: The battery can cause an indirect negative effect by increasing the mass, but the increase of the battery capacity brings a positive impact to the flight endurance. The variable of battery capacity to the UAV weight ratio is positively correlated both the flight endurance and collision evasion tasks.

- $\frac{\texttt{Maximum Total Thrust}}{\texttt{UAV Weight}}$: The ratio between the Total Thrust and Weight is useful to see the actual maneuverability. It is also important to note that the maximum thrust instead of the nominal thrust (during forward flight or hovering) is used for calculation here, which is associated with the capabilities of morphology design instead of its actual performance.

- $\frac{\texttt{Maximum Thrust-Induced Torque}}{\texttt{UAV Moment of Inertia}}$: This variable represents the amount of angular acceleration the UAV can generate.

- `Cruising Power Efficiency`: This variable is the distance traveled by the UAV at consuming 1 W$\cdot$ h at it's most efficient cruising flight state. This intermediate variable is monotonically correlated to the flight endurance and range performance.

IV. Learning to Take-off As The Learning Task

The learning Problem explained here is learning the low-level control of UAV to takeoff. In this problem, we consider the UAV to be at "rest" initially, and then it should learn to take off and take altitude. Besides taking height and preserving it till the end of the flight, we want the UAV to avoid any perturbation in X, Y directions, and we also want to minimize UAV speed, especially during the hovering portion. Subsection IV.A explains this reward

A. Reward

Here we explain the reward formulation for the specific formulation for our current problem.

Table 1 REWARD COEFFICIENTS

Parameter	k_1	k_2	k_3	k_4
Value for Co-Optimization	1	0.5	0.3	-
Values for Learning	100	2	3	100.000

For current problem, we can say we consider the whole trajectory instead of the last time step so that we can remove the last step part.

For the current problem we can define the reward based on our problem definition and its important terms as equation 1.

$$R(t) = -k_1 \frac{\min(z_{\mathtt{des}}, (z - z_{\mathtt{des}})^2}{z_{\mathtt{des}}^2} - k_2 \min(1, \frac{|l|}{|l|_0}) - k_3 \min(1, \frac{|V|}{|V|_0}) \tag{1}$$

where $z, z_{\mathtt{des}}$ are the height and the desired height; $|l|, |l|_0$ are the distance from the initial position $|l| = \sqrt{x^2 + y^2}$ and the reference value. Similar, $|V|$ to UAV speed and the reference value. The variable $z_{\mathtt{des}}$ is defined based on the design choice, but the variables $|l|_0, |V|_0$ are here to add flexibility to the system, by increasing these variables, we relax the perturbation from the initial point or speed to change more. The parameters k_1, k_2, k_3 ($k_1 + k_2 + k_3 = 1$) are used to change the weight of each weight respect to each other. By increasing k_1 we force the system to fly; by increasing k_2 we decrease the perturbation and finally by increasing k_3 we minimize speed.

In addition to the current terms, we observed that the system stays on the ground (to decrease the negative reward for velocity and perturbation). In order to avoid this situation, we added an extra parameter that has a small range of effect ($z \leq 0.1 \times z_{\mathtt{des}}$ but with a larger coefficient ($k4 \gg k_1, k_2, k_3$. Equation 2 explains this update.

$$R(t) = -k_1 \frac{\min(z_{\mathtt{des}}, (z - z_{\mathtt{des}})^2}{z_{\mathtt{des}}^2} - k_2 \min(1, \frac{|l|}{|l|_0}) - k_3 \min(1, \frac{|V|}{|V|_0}) - k_4 \frac{\max(0, 0.1 \times z_{\mathtt{des}}) - z}{0.1 \times z_{\mathtt{des}}} \tag{2}$$

The parameters associated with the reward during the co-optimization and learning are listed in table 1.

V. Learning Capacity metric based on Controllability Gramian

While the learning problem is well defined, computing the learning capacity is not an easy task. One approach, which we followed in [10] is capable of approximating the learning before convergence by using the expected learning capacity concept. While this formulation is promising, it is still computationally prohibitive for longer optimization problems, like the current one based on low-level control problems due to its nested optimization structure. In addition to the computational complexity, we must consider the difference between offline and online learning. For the problems that seek offline co-optimization of learning and morphology, we can increase the learning time, while for online learning problems, learning time must be limited. On the other hand, for offline learning problems, we are not limited to the shape of the model, and we can even change the network topology. Therefore for these problems we seek a way to estimate the learning capability instead of learning performance.

The main idea here is to find an approach to learning rate estimation. The most important criterion here is to find a metric to find the learning ability of the system. Here we assume that although for the case study we use a known learning problem, the learning problem may be unknown know or it may be due to a large set of uncertainties that may not be recognizable from the start. In addition to all these problems, we need a metric for learning that is preferably computable without undergoing the learning process, or at least it is less dependent on learning full process. Therefore, we propose an idea of defining the "learning ability" metric as a function of system ability to change its states. In other words, we define a parameter called "Talent" instead of learning. We hypothesis that there is a strong correlation between talent and learning capacity and to an extent learning rate.

A. Learning Objective Function

The main idea here is to find an approach to learning rate estimation. The most important criterion here is to find a metric to find the learning ability of the system. Here we assume that although for the case study we use a known learning problem, the learning problem may be unknown know or it may be due to a large set of uncertainties that may

not be recognizable from the start. In addition to all these problems, we need a metric for learning that is preferably computable without undergoing the learning process, or at least it is less dependent on learning full process. Therefore, we propose an idea of defining the "learning ability" metric as a function of system ability to change its states. In other words, we define a parameter called "Talent" instead of learning. We hypothesis that there is a strong correlation between talent and learning capacity and to an extent learning rate.

1. Controllability Gramian for UAV

The Gramian of Controllability is the solution to the Lyapunov equation [21]. Compared to the Controllability, which is a binary property of the system, Gramian has more information. The eigenvectors and eigenvalues of the Gramian matrix indicate the directions in state space and the energies required to move the system in the direction of eigenvectors [22]. Equation 3 shows the value of Controllability Gramian for a linear system.

$$W_C = \int_0^{\infty} e^{A\tau} B B^T e^{A^T \tau} d\tau \tag{3}$$

The quadrotor has a nonlinear state-space equation. There are approaches to calculate the nonlinear system Controllability [23, 24], but defining the Controllability Gramian is not a trivial task [25]. Therefore we used a simpler method and used the Contraolability Gramian of the "Linearized" system [26]. Because the UAV starts from the horizontal position the linearization is reasonable and the system does not perturb so much.

2. Talent Objective Function

As explained, although learning for a single problem seems easier, it may affect the robustness of the system in the presence of any uncertainty or control effort deficiency. Therefore we used the idea of more robustness for our control performance. Our Hypothesis is to study the correlation between "Controlabillity" Gramian and Learning. Here we use the eigenvectors of Control ability Gramian as the primary axis of control effect. We prefer to have the maximum correlation between the desired reward and the eigenvectors of Controllability Gramian. Equation 4 Explains the metric we defined:

$$T = \sum_{i=1}^{N_s} \sum_{j=1}^{N_r} |\zeta_i . \vec{R}_j| (C_1 - \frac{|k_j - \sigma_i|}{|k_j + \sigma_i|})^{C_2} \tag{4}$$

Where N_r and N_s are the number of reward terms and states, respectively. Here ζ_i is the eigenvector of Gramian of Controllability and shows the axis with maximum Controllability, σ_i is the eigenvalue corresponding to that eigenvector and shows the amount of Controllability in that direction. The parameters k_i, R_i are the weight and direction of the controlling reward. Note we magnified the values of k_i from the co-optimization for the final learning to improve the UAV flights (listed in table 1). Here We used the reward weights from equation 2 directly as K_i and for the directions we use the directions of controlled states: $\vec{e}_3$, $\frac{\vec{e}_1+\vec{e}_2}{\sqrt{2}}$, and $\frac{\vec{e}_4+\vec{e}_5+\vec{e}_6}{\sqrt{3}}$. These vectors correspond to z direction for the first reward term, r direction for the second reward term, and speeds as the third reward term. The term $|\zeta_i . \vec{R}_i|$ means the dot product of these vectors and enforces an alignment of these directions, and the term $(1 - \frac{|k_i - \sigma_i|}{|k_i + \sigma_i|})$ ensures the correlation between the reward weights and eigenvectors of the Controllability matrix. Finally, C_1, C_2 are factors to tune further the effect of weights and directions in overall performance. These variables are tuned during Co-Design optimization.

B. Learning the optimal Policy through Neroevolution

To find the optimal learning model, we use NeuroEvolution [15]. Neuroevolution is an evolutionary-based learning method. That can optimize both weights and the shape of the neural network. Compared to the classical optimization methods like PPO2, Neuroevolution gives a "fair" comparison, which is not diluted by the network optimization problem. The other advantage of Neuroevolution is its ability to work with skewed reward distribution. Here, we are using a special variation of Neuroevolution called AGENT [18], which is enhanced to avoid premature convergence. Figure 3 shows the framework of this method.

Compared to the Gradient-based Reinforcement learning approaches like PPO or Actor-Critic methods [27, 28], Neuroevolution is usually needed a sparse reward. Therefore here we define the integration of reward for the whole

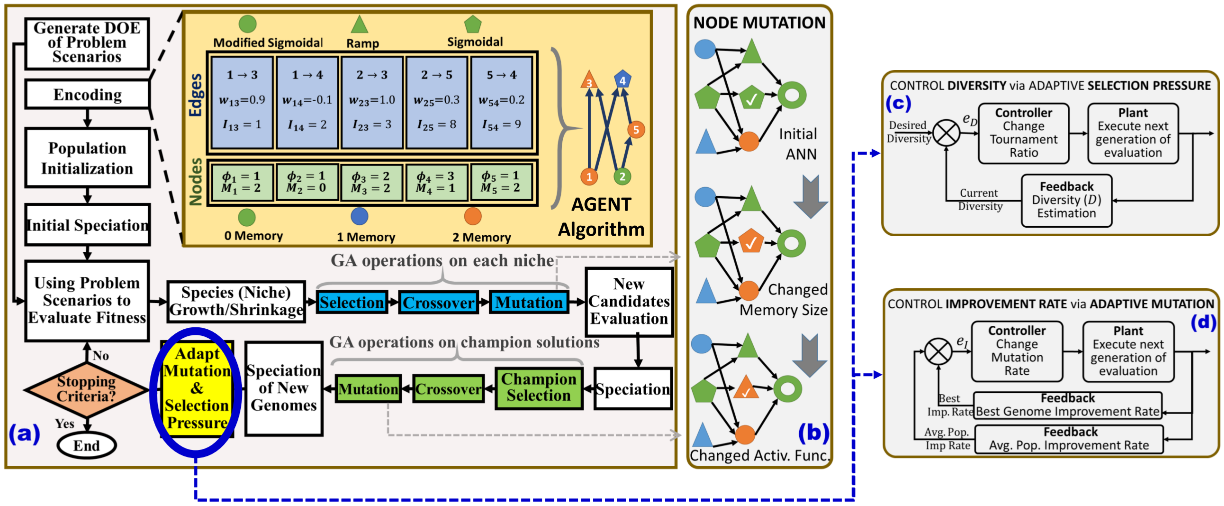


Fig. 3 AGENT framework [18]

Table 2 DESIGN VARIABLES TYPES AND BOUNDS

Variable	Property	Lower Bound	Upper Bound	Variable Type
x_1	$\frac{\texttt{Battery Capacity}}{\texttt{UAV Mass}}$	2 E4	20 E4	Continuous
x_2	$\frac{\texttt{Maximum Thrust}}{\texttt{UAV Mass}}$	10	80	Continuous
x_3	$\frac{\texttt{Maximum Torque}}{\texttt{UAV Moment of Inertia}}$	100	1000	Continuous
x_4	$\frac{\texttt{Distance Travelled}}{\texttt{Energy Consumed}}$	150	1000	Continuous
z_1	Arms length	0.2 m	0.5 m	Continuous
z_2	BWB width	0.2 m	0.5 m	Continuous
z_3	Motor type	1	8	Categorical
z_4	Propeller Type	1	8	Categorical
z_5	Battery Type	1	8	Categorical

episode. Equation 5 show the objective function which is the integration of reward in each time step from start of the simulation, $t = 0$, to the end of the simulation, $t = t_f$.

$$f_{\texttt{NeuroEvolution}} = \int_0^{t_f} R(t)dt \tag{5}$$

VI. The Flight Range/Learning Capacity Co-optimization

To have a more clear definition for the learning problem, we attempt to define the problem in a rigorous mathematical approach in this section.

The morphology design of the quadcopter is defined in terms of 5 variables, and the intelligence design defined in terms of 10 variables. Detailed information on the design variables is provided in Table 2.

The dual-objective optimization problem is defined in the following manner (This is the formulation without adding the optimization constraints):

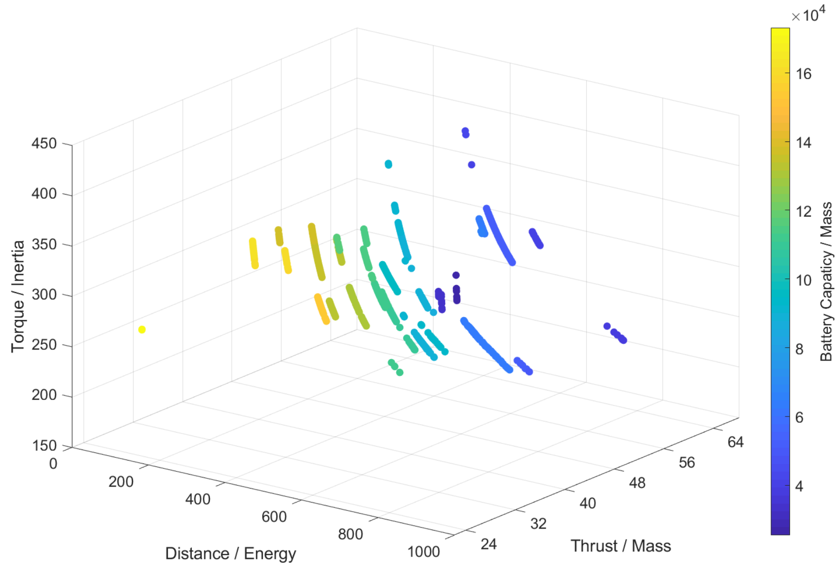


Fig. 4 The Pareto Frontier of the Capacity Variables

$$
\begin{aligned}
\text{Minimize:} \quad & f_t(X) \\
\text{Maximize:} \quad & f_r(X) \\
\text{Subject to:} \quad & [x_1, \cdots, x_4] \in \mathbb{R} \\
& X_L \leq [x_1, \cdots, x_4]^T \leq X_U
\end{aligned}
\tag{6}
$$

A. Optimization Setup and Coding

The capacity variables as explained are used as objective functions and actual morphology parameters as design variables. Equation 7 explains this problem. Here morphology variables $(z_1, z_2, \ldots z_5$ are design variables and capacity variables $(x_1, x_2, \ldots, x_8)$ are the objective functions. Similar to equation 6 the formulation here is without the constraints.

$$
\begin{aligned}
\text{Maximize:} \quad & (x_1(z_1, z_2, \cdots, z_5), x_2(z_1, z_2, \cdots, z_5), \ldots, x_4(z_1, z_2, \cdots, z_5)) \\
\text{Subject to:} \quad & [z_1, z_2] \in \mathbb{R} \\
& [z_3, \cdots, z_5] \in F \\
& Z_L \leq [z_1, \cdots, z_5]^T \leq Z_U
\end{aligned}
\tag{7}
$$

VII. Results

The results of Pareto front optimization for Capacity variables cannot be illustrated due to high dimensions of the answer. Figure 4 shows only 3 dimensions of the 4 capacity variables. This optimization generated 1800 points by combining all generations of Pareto front. The settings for this optimization is listed in table 4.

While it is possible to use capacity variables as design variables for Co-Optimization, here we used original morphology variables for this purpose. The main motivation here is the size of variables that are not changing so much. Using capacity variables is especially useful when the dimension of morphology variables is large, like optimizing the airfoil's shape. Figure 5 shows the Pareto front of the optimization between the Range of flight and Talent. The parameters used in this optimization is listed in table 4.

The optimized answers in figure 5 (blue circles) show a good coverage of the Pareto front. To use slightly smaller design sets, we compared these answers with our Pareto front answers from Capacity Variables optimization. For each point, we chose the most similar design set to the Pareto point. We evaluated these closest points with Flight Range and

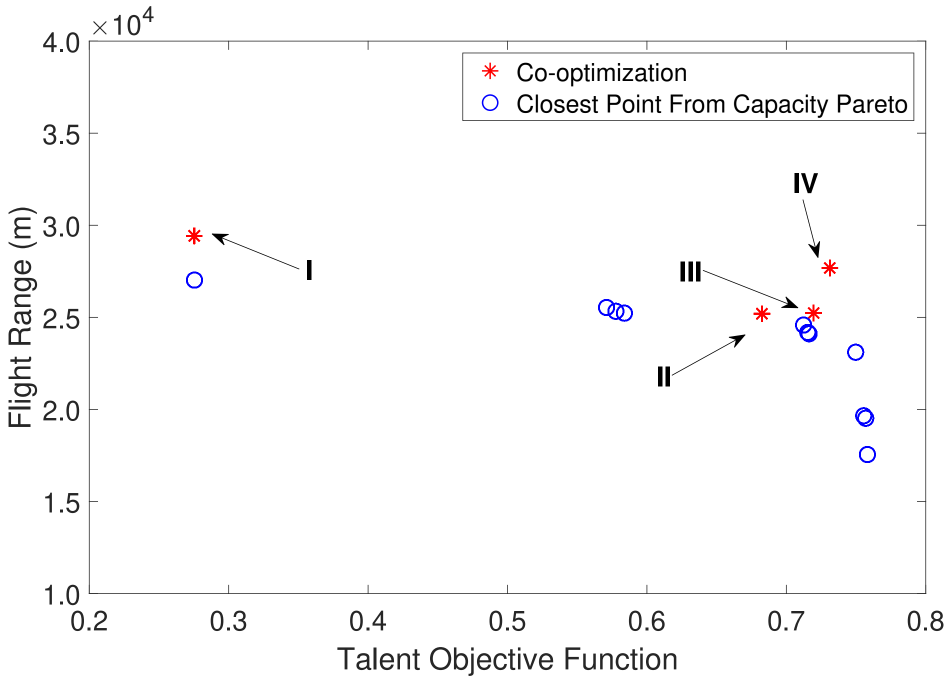


Fig. 5 The Pareto Frontier of the the Co-Design Optimization and the Points used for Optimization Results

Table 3 TALENT AND LEARNING COMPARISON

Design Point	Talent	Total Learning Reward
I	0.28	-5.7914E6
II	0.68	-5.7909E6
III	0.72	-5.788014E6
IV	0.73	**-6.0581E6**

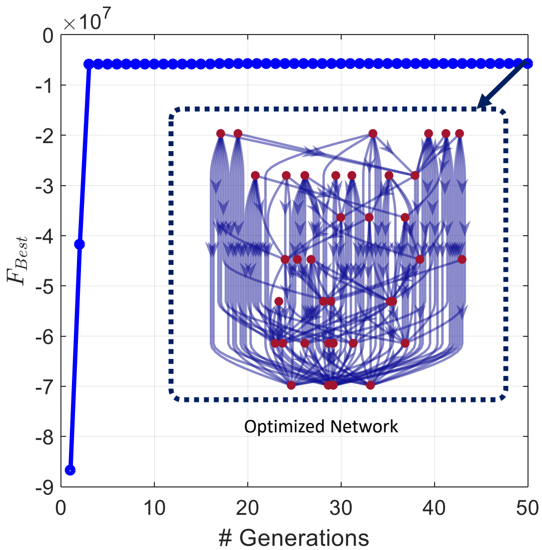
Talent objective functions and plotted them as red stars here (for simplicity, we call them I, II, III, IV). The results show that our concept of using capacity variables is valid, and our newer points all suppress the original Pareto Points. This observation indicates that the capacity variables can be useful for improving the optimization results and finding the inverse optimal answer.

Finally we used these morphology variables from co-optimization and applied learning on them. Figure 6a to 6d show the convergence history o these results. The settings applied for neuroevolution optimization is the same for all neuroevolution settings, and it is listed in table 4.

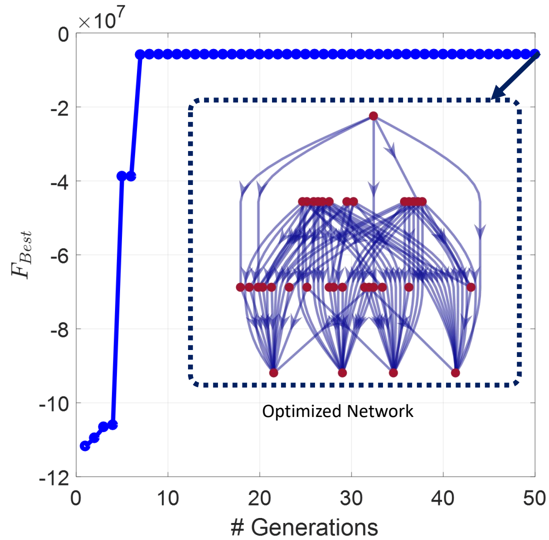
The results show that in all cases, there is improvement through Neuroevolution in system performance, and also, it is important to note that how Neuroevolution successfully works despite having small changes in total reward.

It is interesting to compare the talent objective function and actual learning results. Table 3 compares the talent, as the expected learning, and the learning for these 4 points. It is important to note that the reward does not change uniformly and especially the initial time steps are skewed to force UAV to takeoff.

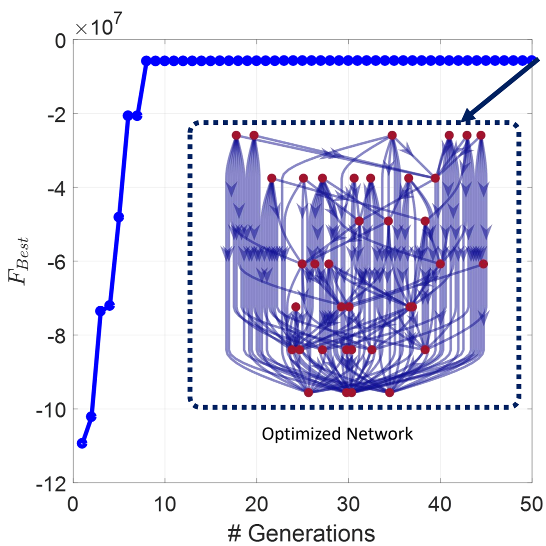
While the first 3 points follow a positive correlation, the last point follows a negative correlation. We further studied this anomaly and realized the difference between the learning and talent for this point is due to the difference between the weights of the co-optimization and reward. While we assumed some initial weights for the reward, it is not necessarily possible to learn base on these weights of rewards. Especially because the UAV can be stuck in a local minimum behavior, do not take off and have no negative reward for velocity and perturbation, we added a huge initial negative reward. To compensate this term, we increased the effect of the perturbation term, which lead to the special behavior of point *IV*.Figures 7a to 7d show the flight path of answers *I, IV* in 2D and 3D space. Compared to answer *I* answer *IV* has higher divergence in X-Y direction.



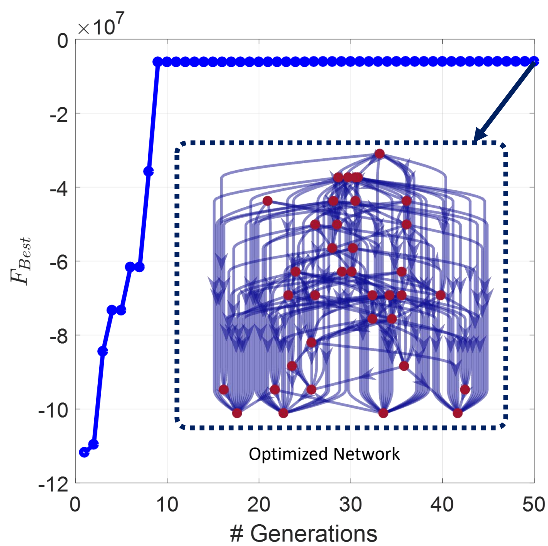
(a) Learning for the morphology *I*



(b) Learning for the morphology *II*



(c) Learning for the morphology *III*

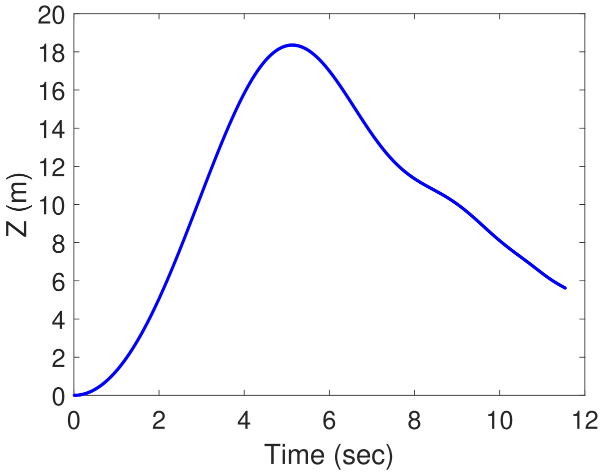


(d) Learning for the morphology *IV*

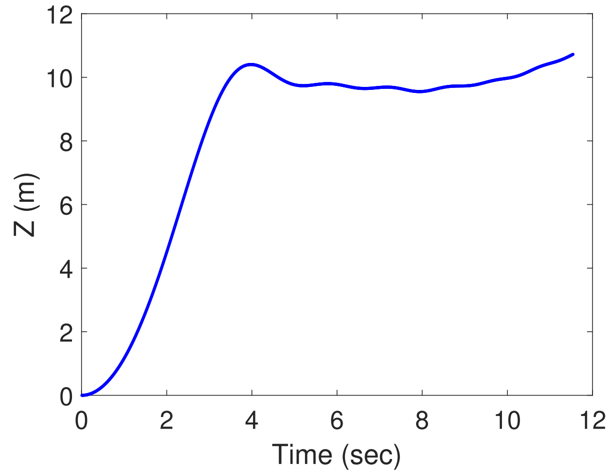
Fig. 6 The Convergence history and the optimal design for 4 different morphology settings using Neuroevolution

Table 4 OPTIMIZATION AND LEARNING SETTINGS

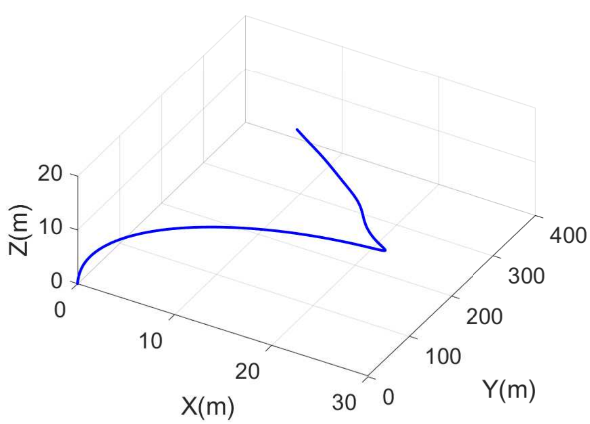
Optimization	**Variable**	**Value**
	Optimization Method	NSGAII
morphology → capacity variable Optimization	Population	1000
	Maximum Iteration	100
	Optimization Method	NSGAII
morphology → Talent Range of Flight	Population	2000
	Maximum Iteration	30
	Optimization Method	AGENT
NeuroEvolution	Population	360
	Maximum Iteration	50



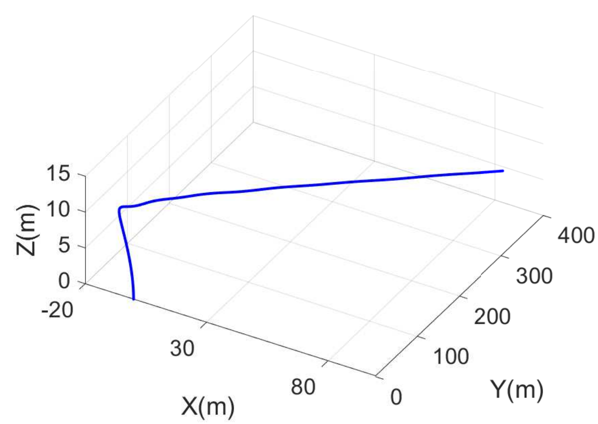
(a) Height Vs. Time for The optimal network based on morphology settings of *I*



(b) Height Vs. Time for The optimal network based on morphology settings of *IV*



(c) The Trajectory for The optimal network based on morphology settings of *I*



(d) The Trajectory for The optimal network based on morphology settings of *IV*

Fig. 7 Case *I* and *IV* trajectories

VIII. Conclusions

In this paper, a co-design framework is proposed to co-optimize the flight endurance and low-level control learning of a quadcopter UAV. The quadcopter features a unique aerodynamic morphology and an evolving neural network collision avoidance intelligence. Instead of directly applying the learning for each morphology design, which is computationally prohibitive, we introduced a metric called "Talent" that can be used to estimate the possible learning capacity of the system. Here we used the eigenvalues and eigenvectors of Controllability matrix Gramian as features to define this metric. Our results indicate that there is a strong correlation between this metric and system performance.

In addition to the "Talent" parameter we utilized an additional concept, called "capacity variables". These intermediate variables can change the design space from morphology variables to fewer continuous variables for co-design optimization. Also, if the design goal(s) are their monotonic functions, we can use them for inverse design.

This paper was the first work in adding the control theory as a metric of learning capacity. Therefore we used a deterministic problem for future studies. We plan to add uncertainty to the model and also consider the correlation between learning and the parameters associated with noise robustness in control theory.

References

[1] Pahl, G., and Beitz, W., *Engineering design: a systematic approach*, Springer Science & Business Media, 2013.

[2] Fathy, H. K., Reyer, J. A., Papalambros, P. Y., and Ulsov, A. G., "On the coupling between the plant and controller optimization problems," *Proceedings of the 2001 American Control Conference. (Cat. No.01CH37148)*, Vol. 3, 2001, pp. 1864–1869 vol.3. https://doi.org/10.1109/ACC.2001.946008.

[3] Ball, L. J., Scrivener, S. A. R., and Andreé, W., *Collaborative design: proceedings of CoDesigning 2000*, Springer, 2000.

[4] Allison, J. T., and Herber, D. R., "Special Section on Multidisciplinary Design Optimization: Multidisciplinary Design Optimization of Dynamic Engineering Systems," *AIAA Journal*, Vol. 52, No. 4, 2014, p. 691–710. https://doi.org/10.2514/1.j052182.

[5] Herber, D. R., and Allison, J. T., "Nested and simultaneous solution strategies for general combined plant and control design problems," *Journal of Mechanical Design*, Vol. 141, No. 1, 2019, p. 011402.

[6] Suh, H., *Axiomatic Design: advances and applications MIT-Pappalardo series in Mechanical Engineering*, Oxford University Press, USA Google Scholar, 2001.

[7] Lee, E. A., and Seshia, S. A., *Introduction to embedded systems: A cyber-physical systems approach*, Mit Press, 2016.

[8] Allison, J. T., Guo, T., and Han, Z., "Co-design of an active suspension using simultaneous dynamic optimization," *Journal of Mechanical Design*, Vol. 136, No. 8, 2014, p. 081003.

[9] Censi, A., "A Mathematical Theory of Co-Design," *arXiv preprint arXiv:1512.08055.*, 2015.

[10] Zeng, C., Behjat, A., Gabani, K. K., and Chowdhury, S., "Artificial Life-inspired Morphology/Learning CoDesign Framework: Towards Conceptual Design of Intelligent Hybrid Quadcopters," *AIAA Aviation 2019 Forum*, 2019, p. 3457.

[11] Chowdhury, S., Tong, W., Messac, A., and Zhang, J., "A mixed-discrete particle swarm optimization algorithm with explicit diversity-preservation," *Structural and Multidisciplinary Optimization*, Vol. 47, No. 3, 2013, pp. 367–388.

[12] Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T., "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 2, 2002, pp. 182–197. https://doi.org/10.1109/4235.996017.

[13] Behjat, A., Gabani, K. K., and Chowdhury, S., "Training Detection-Range-Frugal Cooperative Collision Avoidance Models for Quadcopters via Neuroevolution," *AIAA Aviation 2019 Forum*, 2019, p. 3312.

[14] Behjat, A., Paul, S., and Chowdhury, S., "Learning reciprocal actions for cooperative collision avoidance in quadrotor unmanned aerial vehicles," *Robotics and Autonomous Systems*, 2019, p. 103270. https://doi.org/10.1016/j.robot.2019.103270.

[15] Floreano, D., Dürr, P., and Mattiussi, C., "Neuroevolution: from architectures to learning," *Evolutionary intelligence*, Vol. 1, No. 1, 2008, pp. 47–62.

[16] Stanley, K. O., and Miikkulainen, R., "Evolving neural networks through augmenting topologies," *Evolutionary computation*, Vol. 10, No. 2, 2002, pp. 99–127.

[17] Prestwich, S., Tarim, S., Rossi, R., and Hnich, B., "A neuroevolutionary approach to stochastic inventory control in multi-echelon systems," *International journal of production research*, Vol. 50, No. 8, 2012, pp. 2150–2160.

[18] Behjat, A., Chidambaran, S., and Chowdhury, S., "Adaptive Genomic Evolution of Neural Network Topologies (AGENT) for State-to-Action Mapping in Autonomous Agents," *2019 International Conference on Robotics and Automation (ICRA)*, IEEE, 2019, pp. 9638–9644.

[19] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O., "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[20] Moghadassian, B., and Sharma, A., "Inverse design of single-and multi-rotor horizontal axis wind turbine blades using computational fluid dynamics," *Journal of Solar Energy Engineering*, Vol. 140, No. 2, 2018, p. 021003.

[21] Agarwal, R., Hristova, S., and O'Regan, D., "A survey of Lyapunov functions, stability and impulsive Caputo fractional differential equations," *Fractional Calculus and Applied Analysis*, Vol. 19, No. 2, 2016, pp. 290–318.

[22] Cortesi, F. L., Summers, T. H., and Lygeros, J., "Submodularity of energy related controllability metrics," *53rd IEEE Conference on Decision and Control*, IEEE, 2014, pp. 2883–2888.

[23] Sussmann, H. J., and Jurdjevic, V., "Controllability of nonlinear systems." 1972.

[24] Balachandran, K., and Dauer, J., "Controllability of nonlinear systems in Banach spaces: a survey," *Journal of Optimization Theory and Applications*, Vol. 115, No. 1, 2002, pp. 7–28.

[25] Kashima, K., "Noise response data reveal novel controllability Gramian for nonlinear network dynamics," *Scientific reports*, Vol. 6, 2016, p. 27300.

[26] Tahir, Z., Tahir, W., and Liaqat, S. A., "State Space System Modeling of a Quad Copter UAV," *arXiv preprint arXiv:1908.07401*, 2019.

[27] Konda, V. R., and Tsitsiklis, J. N., "Actor-critic algorithms," *Advances in neural information processing systems*, 2000, pp. 1008–1014.

[28] Grondman, I., Busoniu, L., Lopes, G. A., and Babuska, R., "A survey of actor-critic reinforcement learning: Standard and natural policy gradients," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, Vol. 42, No. 6, 2012, pp. 1291–1307.