AIAA SciTech Forum
January 3-7, 2022, San Diego, CA & Virtual
AIAA SCITECH 2022 Forum

10.2514/6.2022-1417

Robust Neurocontrol for Autonomous Dynamic Soaring

Eric J. Kim* and Ruben E. Perez†
Royal Military College of Canada, Kingston, Ontario, Canada

The application of dynamic soaring techniques on small unmanned aerial vehicles (SUAVs) aims to exploit naturally occurring wind gradients to increase flight endurance. However, considering the limited computational resources onboard SUAVs and the imperfect nature of real-world environments and physical systems, there is a practical need to design simple and robust control systems. As such, this paper presents a neuroevolutionary strategy for generating efficient neurocontrollers that exhibit generalized and robust behavior. The Neuroevolution of Augmenting Topologies (NEAT) algorithm is applied to evolve networks in a way that preserves simplicity while maximizing performance. Simulated flight tests in stochastic environments show that resulting controllers can perform dynamic soaring for a range of initial conditions and time-varying parameters. Flight trajectories and robustness metrics are presented and compared for a small autonomous aircraft operating in such environments.

Nomenclature

A	Wind profile shape parameter [-]	μ	Roll [deg]
C_D	Drag coefficient [-]	ψ	Heading [deg]
C_L	Lift coefficient [-]	ρ	Local air density [slug/ft^3]
C_{D0}	Zero-lift drag coefficient [-]	e_K	Kinetic energy [J]
D	Drag [N]	e_P	Potential energy [J]
E_{max}	Maximum lift-to-drag ratio [-]	e_T	Total energy [J]
K	Induced drag factor [-]	g	Standard gravity [ft/s^2]
L	Lift [N]	h	Height [ft]
S	Wing reference area [s^2]	h_{tr}	Wind profile transition height [ft]
V	Airspeed [ft/s]	m	Mass [slug]
W	Wind strength [ft/s]	n	Load factor [-]
β	Wind profile gradient [ft/s/ft]	x	X-axis position [ft]
γ	Pitch [deg]	y	Y-axis position [ft]

I. Introduction

The operational capabilities of modern small unmanned aerial vehicles (SUAVs) are primarily limited by their flight endurance. Their utility in various applications, such as scientific data gathering and security-related surveillance, would be greatly improved by increasing maximum flight duration, and by extension, maximum travel distance. Soaring techniques can achieve this objective by reducing the rate of energy consumption through thrustless flight. One such method known as dynamic soaring involves exploiting the differences in horizontal wind strengths across altitudes through the continual and cyclic exchange of kinetic and potential energies. The technique can be naturally observed in the behavior of seabirds such as the

*Master's Student, Department of Mechanical and Aerospace Engineering, Royal Military College of Canada, Kingston, ON, K7K 7B4, Canada
†Associate Professor and AIAA Senior Member, Department of Mechanical and Aerospace Engineering, Royal Military College of Canada, Kingston, ON, K7K 7B4, Canada

albatross, who commonly soar across transoceanic distances without flapping their wings [1]. Regardless, the task of implementing dynamic soaring on artificial systems can be divided into the problems of trajectory planning and aircraft control.

Initially, much of the research around planning well-defined flight paths to be tracked by aircraft has been inspired by numerical trajectory optimization methods. These transform the soaring problem into a nonlinear programming problem, which would then be solved through shooting or collocation methods by optimization algorithms. Zhao [2] used the collocation approach in studying the optimal dynamic soaring trajectories of different objectives, and the work of Sachs [3] extensively used numerical optimization to study the environmental requirements for dynamic soaring. However, while serving as analytical tools, such methods are impractical as implementable solutions due to their computationally intensive nature. The limited computing hardware on SUAV platforms and the need to rapidly generate trajectories for uninterrupted soaring require efficient planning solutions.

In consideration of this aspect, other research efforts have presented various simplification methods to bypass the computational constraints. Bird and Langelaan [4] proposed a trajectory planning approach based on a database of optimal loitering trajectories calculated for a range of aircraft and environmental conditions and loaded into onboard memory prior to flight. Akhtar *et al.* [5] used a polynomial parameterization of the flight trajectory, separating the spatial path from the speed profile to enable real time optimization, and similarly, Gao and Liu [6] presented a geometric parameterization method that used Dubins paths to reduce the problem's dimensionality. Such approaches provide implementable, real-time solutions to the problem of trajectory planning, but require separate control systems to steer the aircraft along the generated paths. When considering the stringent energy management required for sustained dynamic soaring, however, the correction of tracking errors, or deviations from the planned trajectory, requires the expenditure of additional energy to the point where there may no longer be sufficient energy to continue soaring, which is a disadvantage of explicit trajectory tracking.

For the task of aircraft control, numerous studies have examined the applicability of both linear and nonlinear control schemes. For instance, Lawrance and Sukkarieh [7] employed heuristics to cycle through piecewise segments of the overall trajectory using proportional-integral-derivative (PID) controllers. Deittert *et al.* [8] applied a linear quadratic regulator to a disturbance-injected flight model in an examination of robustness, and another study by Liu *et al.* [9] investigated a nonlinear model predictive control strategy in implementing more flexible controllers. More recently however, the pursuit of generalized and robust autonomous behavior, in addition to modern advancements in the accessibility and understanding of neural networks, have resulted in a recent interest in the field of neurocontrol. Particularly, much of the focus in using neural networks as components within the control system has been on the use of reinforcement learning algorithms and deep neural networks [10, 11]. Nevertheless, the complexity of deep neural network topologies and the difficulty in training robustness remain as obstacles in applying such methods to dynamic soaring. In consideration of these issues, Perez *et al.* [12] demonstrated the ability of the Neuroevolution of Augmenting Topologies (NEAT) algorithm [13] to produce optimal dynamic soaring controllers using simple neural networks. This work was extended by Kim *et al.* [14], who presented the advantages of the NEAT-based approach compared to numerical trajectory optimization.

This paper, further developing the concepts explored in the previous works [12, 14], presents a method of generating topologically simple and environmentally robust neurocontrollers for an integrated trajectory planning and aircraft control strategy. The neuroevolutionary approach introduced in this work produces sparsely-connected, minimal-size neural networks that are evolved to exhibit robustness against stochastic and time-varying conditions while requiring minimal training data. The paper is organized as follows: Section II describes the dynamic soaring problem and the NEAT-based neurocontroller-generation strategy. Section III presents the robustness results of testing the approach's resulting neural networks, and Section IV provides concluding remarks.

II. Methodology

A. Dynamic Soaring

A typical dynamic soaring cycle as first described by Rayleigh [15], consists of four primary segments: upwind climb, high altitude turn, downward sink, and low altitude turn. As shown in Fig. 1, an aircraft exchanges kinetic energy for potential energy as it gains altitude by turning into the wind field, and subsequently regains kinetic energy while losing altitude by descending away from the wind. Traveling dynamic soaring involves repeating this pattern along a specific direction to cover distance while also expending any additional energy gained over each cycle.

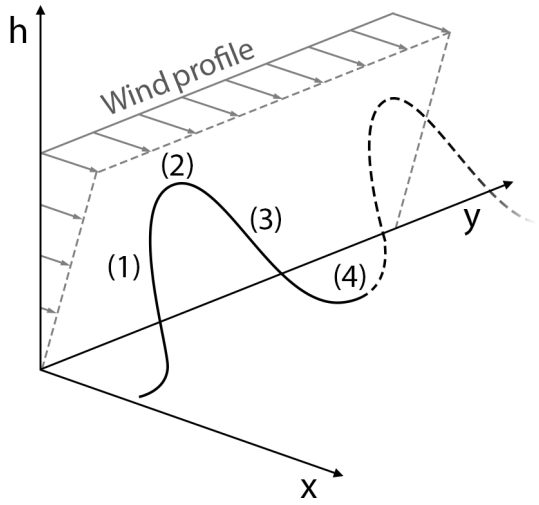


Figure 1. Typical dynamic soaring trajectory with a linear wind profile [14].

B. Aircraft Model

This work uses a three-degree-of-freedom (3DOF) point mass model of a typical commercial SUAV. The system's equations of motion [2] accommodating wind in both x and y directions are presented below, and Fig. 2 depicts a free body diagram of the model.

$$\dot{V} = -\frac{D}{m} - g\sin(\gamma) - \dot{W}_x\cos(\gamma)\sin(\psi) - \dot{W}_y\cos(\gamma)\cos(\psi) \tag{1}$$

$$\dot{\psi} = \frac{L\sin(\mu)}{Vm\cos(\gamma)} - \frac{\dot{W}_x\cos(\psi)}{V\cos(\gamma)} - \frac{\dot{W}_y\sin(\psi)}{V\cos(\gamma)} \tag{2}$$

$$\dot{\gamma} = \frac{1}{V}\left(\frac{L\cos(\mu)}{m} - g\cos(\gamma) + \dot{W}_x\sin(\gamma)\sin(\psi) + \dot{W}_y\sin(\gamma)\cos(\psi)\right) \tag{3}$$

$$\dot{x} = V\cos(\gamma)\sin(\psi) + W_x \tag{4}$$

$$\dot{y} = V\cos(\gamma)\cos(\psi) + W_y \tag{5}$$

$$\dot{h} = V\sin(\gamma) \tag{6}$$

Values for lift, drag, and drag coefficient, required for the equations of motion, are calculated with the following, where the drag coefficient is modeled as the sum of both parasitic and lift-induced drag:

$$L = \frac{1}{2}\rho V^2 S C_L \tag{7}$$

$$D = \frac{1}{2}\rho V^2 S C_D \tag{8}$$

$$C_D = C_{D0} + K C_L^2 \tag{9}$$

$$K = \frac{1}{4C_{D0}E_{max}^2} \tag{10}$$

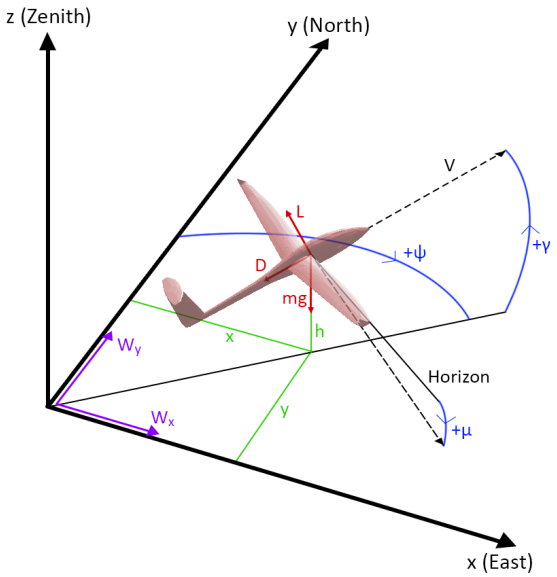


Figure 2. Free body diagram of SUAV model.

Additionally, the total energy of the aircraft is comprised of both the kinetic and potential energies:

$$e_K = \frac{1}{2}mV^2 \tag{11}$$

$$e_P = mgh \tag{12}$$

$$e_T = e_K + e_P \tag{13}$$

This mathematical model of the aircraft describes the dynamics of the simulation, which can be accessed by a control agent through the following state $\mathbf{x}$ and control $\mathbf{u}$ variables:

$$\mathbf{x} = \begin{bmatrix} V \\ \psi \\ \gamma \\ h \\ \dot{h} \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} C_L \\ \mu \end{bmatrix} \tag{14}$$

C. Wind Model

The wind profile initially formulated by Zhao [2] is modeled through a quadratic function that is dependent on the aircraft's height along with several shape parameters. As shown in Figure 3, the model takes a logarithmic shape when $0 \leq A < 1$, a linear profile when $A = 1$, and an exponential form when $1 < A \leq 2$. Furthermore, the wind strength and bearing are assumed to be uniform over the ground surface.

$$W_x = \begin{cases} \beta \left[Ah + \frac{1-A}{h_{tr}} h^2 \right] & h < h_{tr} \\ W_{max} & h \geq h_{tr} \end{cases} \tag{15}$$

$$\beta = \frac{W_{max}}{h_{tr}} \tag{16}$$

The wind model can be replicated for a profile in the y direction. However, the environment models used in this paper only consisted of a wind along the x axis. As a result, the rates of change of the wind profile

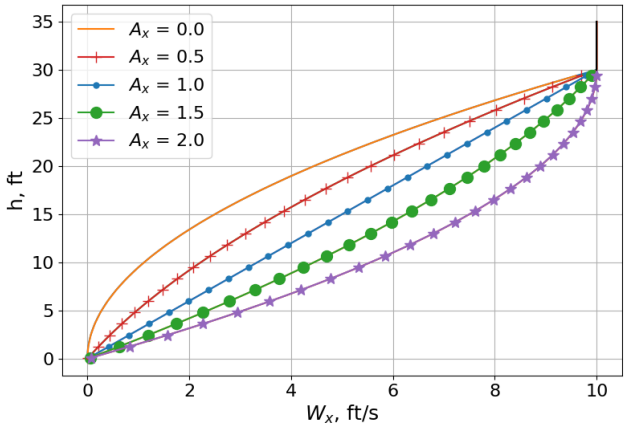


Figure 3. Wind profile shapes [14]

used in the aircraft model's equations of motion are defined as follows:

$$\dot{W}_x = \beta_x \left[A_x + 2\frac{1 - A_x}{h_{tr_x}} h \right] V \sin(\gamma) \tag{17}$$

$$\dot{W}_y = 0.0 \tag{18}$$

$$\dot{W}_z = 0.0 \tag{19}$$

D. Robust Neuroevolutionary Strategy

The neuroevolutionary strategy developed to evolve robust neural network controllers uses the NEAT algorithm. In a typical genetic algorithm, members of a population set are evaluated based on their performance at a certain task, which is measured by the fitness value. After evaluation, members are ranked according to their fitness, and either allowed to mutate and reproduce to propagate their genes to the next generation, or removed from the gene pool. In this way, the algorithm cultivates and develops the genomes of the highest-performing members. NEAT encodes the connections, weights, and biases of a neural network into a member's genome, allowing these aspects to procedurally evolve from an initially simple structure while only adding additional complexity when advantageous to the member's fitness. The result is that NEAT-based neural networks are extremely sparse, only requiring a minimal number of connections to exhibit specific behaviors.

In taking advantage of these characteristics, this work presents the robust neuroevolutionary strategy depicted by Fig. 4. The goal is to produce neural networks that can control the aircraft along a traveling dynamic soaring trajectory to maximize flight endurance, as measured by the distance covered over the trajectory. The outer loop illustrates the fitness calculation process that is required for the rest of the NEAT algorithm to continually evolve novel neurocontrollers while eliminating stagnant members. The inner loop takes a single member π_i of the neurocontroller population Π and evaluates its ability to perform dynamic soaring in a stochastic environment. The neural network receives the aircraft model's states, which are fed-forward to produce the output control values detailed in Eqn. 14. Furthermore, the simulation is advanced by a discrete time interval δ_t at every loop until a termination flag is raised when either the aircraft model crashes or the maximum simulation time t_{max} is reached. After termination, the state and control histories that were recorded during the simulation are used to produce a fitness value for the neural network undergoing evaluation. The fitness function uses predefined limits on each of the following aerodynamic and control variables to calculate a value that is proportional to the degree to which these limits were exceeded at every time step of the simulation, up to the final time t_f:

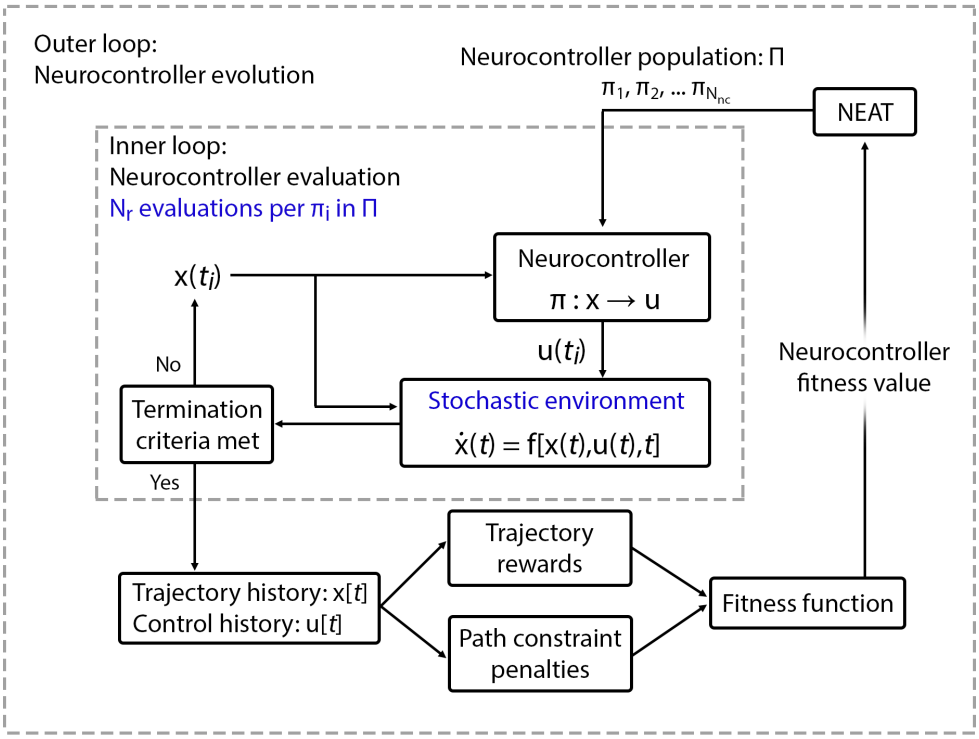


Figure 4. Robust neuroevolutionary strategy.

$$
\begin{aligned}
\text{Airspeed:} \quad & V_{min} \leq V \leq V_{max} \\
\text{Height:} \quad & h_{min} \leq h \leq h_{max} \\
\text{Pitch angle:} \quad & \gamma_{min} \leq \gamma \leq \gamma_{max} \\
\text{Load factor:} \quad & n \leq n_{max} \\
\\
\text{Pitch rate of change:} \quad & \dot{\gamma}_{min} \leq \dot{\gamma} \leq \dot{\gamma}_{max} \\
\text{Heading rate of change:} \quad & \dot{\psi}_{min} \leq \dot{\psi} \leq \dot{\psi}_{max} \\
\text{Lift coefficient rate of change:} \quad & \dot{C}_{Lmin} \leq \dot{C}_L \leq \dot{C}_{Lmax} \\
\text{Roll rate of change:} \quad & \dot{\mu}_{min} \leq \dot{\mu} \leq \dot{\mu}_{max}
\end{aligned}
$$

The penalty s_{pen} for exceeding the constraints of one of these variables s was computed as the sum of the differences between the variable's limits s_{min} and s_{max}, and its value s_i at a discrete time interval up to the end of the simulation:

$$s_{pen} = \sum_{t=0}^{t_f} [min(0, s_t - s_{min}) + max(0, s_t - s_{max})] \tag{20}$$

The penalties for each variable were then combined in a sum-squared-error formulation to produce the total penalty value p_π of the neurocontroller. To ensure that the evolutionary process, in early generations, favored longer trajectories with correspondingly greater penalties over shorter flights with smaller penalties, the total penalty was also normalized by the total flight time. Furthermore, a large constant penalty value was added in the event of the aircraft crashing.

$$p_\pi = \frac{(V_{pen}^2 + h_{pen}^2 + \gamma_{pen}^2 + n_{pen}^2 + \dot{\gamma}_{pen}^2 + \dot{\psi}_{pen}^2 + \dot{C}_{L\,pen}^{\;2} + \dot{\mu}_{pen}^2)}{t_f} \tag{21}$$

In addition to penalties, a reward mechanism that provided a value r_π proportional to the aircraft's total displacement at the end of the simulation was also designed to encourage traveling dynamic soaring with the goal of maximizing flight distance:

$$r_\pi = (x_f - x_i)^2 + (y_f - y_i)^2 \tag{22}$$

Therefore, the fitness value f_π of a single neurocontroller after evaluation was calculated as the sum of the distance reward and negated boundary penalty, with additional scaling constraints k that were experimentally tuned to prevent either value from dominating the other:

$$f_\pi = k_1 r_\pi - k_2 p_\pi \tag{23}$$

Since the simulation experienced by the neurocontroller is stochastic, initial conditions and environment parameters are randomized and therefore distinct for every instance of the simulation model. To evolve robustness in such varying environments, every neurocontroller undergoes N_r number of flight simulations and evaluations, and the final fitness value F_π assigned to the member is the sum of all the single-simulation fitnesses as described by Eqn. 23:

$$F_\pi = \sum_{n=0}^{N_r} f_{\pi_n} \tag{24}$$

This Monte Carlo method of sampling stochastic environments and testing neural networks allows all networks in the population to experience a variety of conditions, with the NEAT algorithm optimizing for the member that exhibits the best dynamic soaring performance across all of its stochastic runs. Additionally, the effectively random sampling of neurocontroller performances can be used to compute a metric for robustness, which is described in Section III.

E. Simulation Environment

This section describes the additional models used to represent variations and uncertainties in the different stochastic environments implemented for the inner loop of the robust neuroevolutionary strategy.

1. *Variable Initial States*

Neurocontrollers that are evolved to be robust against variable initial states are subjected, during the evaluation process, to environments in which the initial flight agent's states are randomly determined. In such a case, the airspeed and altitude are initialized according to uniform distributions that span a certain multiple $x \in [0, 1]$ of a nominal value. The heading angle is assigned a value within a 360 degree arc, and the pitch angle is set to be between 45 degrees above and below the horizontal plane. However, the altitude's rate of change is kept exempt from any stochastic mechanisms under the assumption that soaring maneuvers would typically commence from level flight. The overall scheme is depicted by Fig. 5.

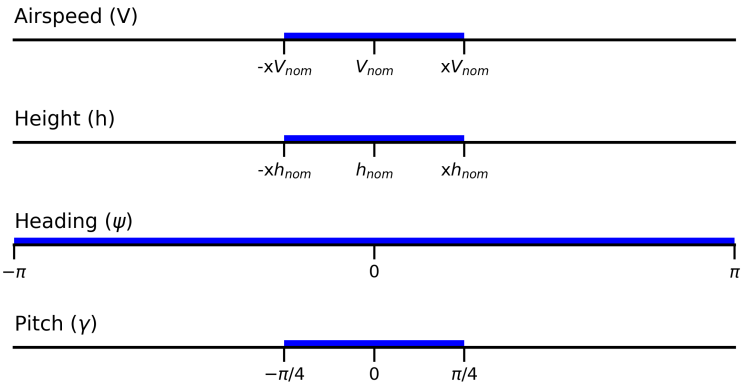


Figure 5. Uncertainty modeling: uniform distribution range for variable initial states.

2. *Variable Initial Wind Profile*

In the variable initial wind profile environment, the parameters characterizing the wind field are randomly initialized similar to the way in which the initial aircraft states are set. These variables include the shape parameter A, the transition height h_{tr}, and the maximum wind speed W_{max}.

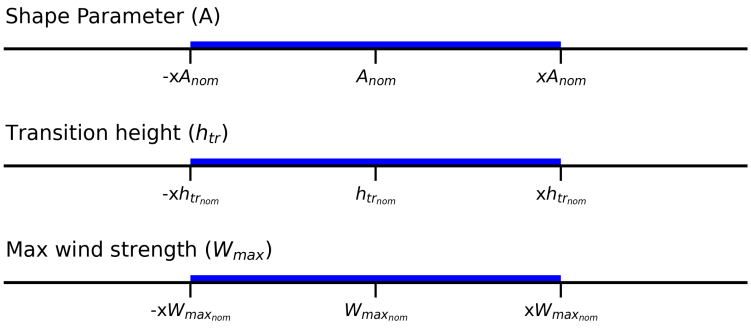


Figure 6. Uncertainty modeling: uniform distribution range for variable initial wind profiles.

3. *Dynamic Wind Profile*

Wind profiles can also be expected to change during flight. Therefore, in the dynamic wind profile environment, one of the three wind parameters (A, h_{tr}, W_{max}) are selected to be varied sinusoidally throughout the simulation, characterized by the frequency and amplitude of change, which are sampled from a uniform distribution when the environment is initialized. The effect is illustrated in Fig. 7, which shows a sinusoidal variance in the maximum wind strength.

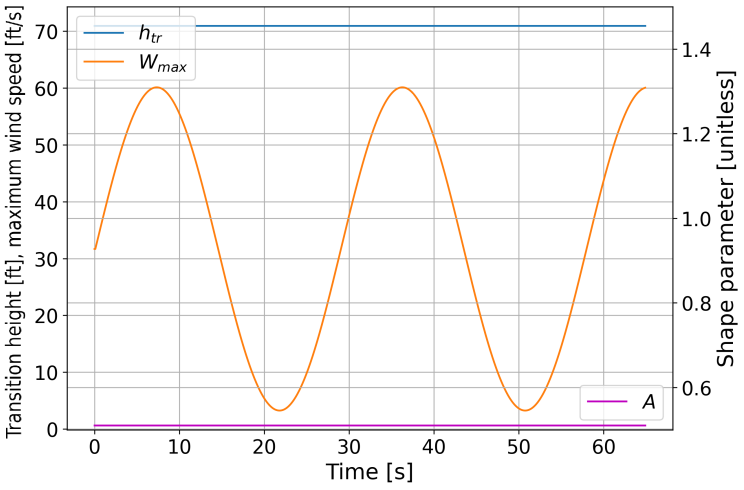


Figure 7. Uncertainty modeling: dynamically varying wind profiles.

These models describe the different stochastic environments that were used to evolve and test the robust neuroevolutionary strategy's resulting neural networks.

III. Results and Discussion

The robust neuroevolutionary method depicted in Fig. 4 and the uncertainty models described in Section II.E were used to evolve neurocontrollers robust to random initial aircraft states, random initial wind profiles, and dynamic wind profiles. This section presents the simulation results of evaluating these neurocontrollers, along with a baseline neurocontroller evolved in a deterministic environment. Additionally, the SUAV model and environmental parameters used for all simulations are described below in Table 1.

Table 1. Dynamic soaring SUAV neurocontroller simulation parameters

SUAV parameters	Value	Wind parameters	Value	Trajectory parameters	Value
g [ft/s^2]	32.174	A_x	1.0	V_0 [ft/s^2]	27.3
ρ [slug/ft^3]	0.0023769	h_{tr_x} [ft]	60.0	ψ_0 [deg]	0
m [slug]	0.295	W_{max_x} [ft/s^2]	30.0	γ_0 [deg]	0
S [ft^2]	10.7			h_0 [ft]	40.0
C_{D_0}	0.025	A_y	N/A	x_0 [ft]	0
E_{max}	20.0	h_{tr_y} [ft]	N/A	y_0 [ft]	0
n_{max}	15	W_{max_x} [ft/s^2]	N/A	h_{min} [ft]	0
$C_{L_{max}}$	1.5			$\dot{\gamma}_{max}$ [deg/s]	15
$C_{L_{min}}$	-0.2			$\dot{\psi}_{max}$ [deg/s]	50
μ_{max} [deg]	60			$\dot{C}_{Lmax}$	0.25
				$\dot{\mu}_{max}$ [deg/s]	90
				t_{max} [s]	600

A. Deterministic Neurocontroller

The initial aircraft states and wind profiles of the environment in which the neural network of Fig. 8 was trained remained constant both within and between flight simulations. The resulting network does not have any hidden layers, requiring only the heading and pitch angles, and height.

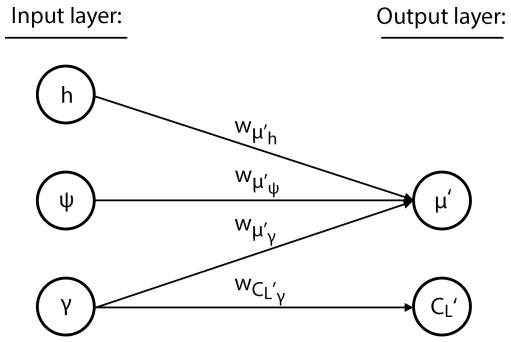


Figure 8. Topology of the deterministic dynamic soaring neurocontroller.

The bias nodes $b_{C'_L}$ and $b_{\mu'}$, and the connection weights w have the values:

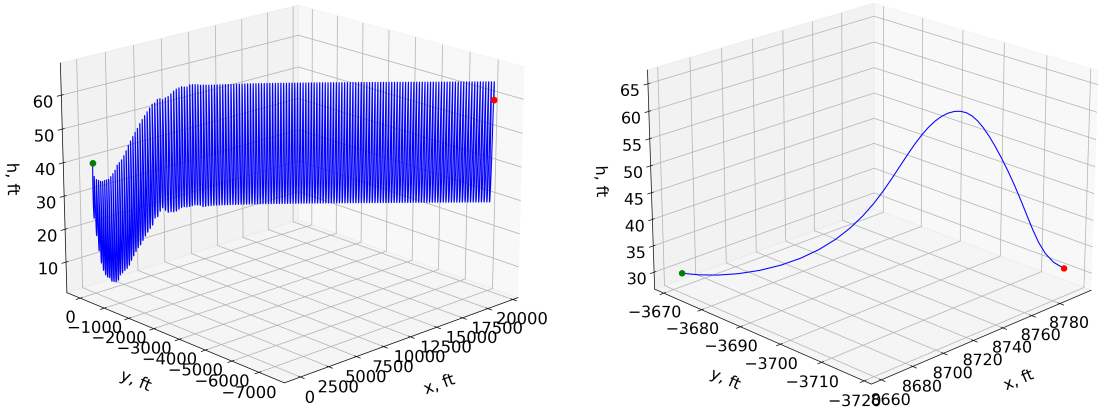
$$b_{C'_L} = 0.307 \qquad w_{\mu'_h} = -2.51$$
$$b_{\mu'} = -1.32 \qquad w_{\mu'_\psi} = -2.62$$
$$w_{\mu'_\gamma} = 0.547$$
$$w_{C'_{L_\gamma}} = -3.15$$

The signals from the input layer are multiplied by their respective connection weights before being aggregated through a summation operation at each output node. The feedforward process is described by Eqn. 28, where the normalized output values C'_L and μ' obtained from the sigmoid function $\sigma(x)$ are scaled to values within the aircraft control limits:

$$C'_L = \sigma(w_{C'_{L_\gamma}}\gamma + b_{C'_L}) \tag{25}$$
$$\mu' = \sigma(w_{\mu'_h}h + w_{\mu'_\psi}\psi + w_{\mu'_\gamma}\gamma + b_{\mu'}) \tag{26}$$
$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{27}$$

$$\begin{bmatrix} C_L \\ \mu \end{bmatrix} = \begin{bmatrix} C'_L & 0 \\ 0 & \mu' \end{bmatrix} \begin{bmatrix} C_{L_{max}} - C_{L_{min}} \\ 2\mu_{max} \end{bmatrix} + \begin{bmatrix} C_{Lmin} \\ -\mu_{max} \end{bmatrix} \tag{28}$$

Figure 9a shows the simulation results of a 600s flight, and Fig. 9b shows a single period at the midpoint of the same trajectory. In addition, the trace of a five-cycle segment of the flight is displayed in Fig. 10, indicating that after an initial transient period of energy gain, the neurocontroller produces sustained dynamic soaring trajectories as evidenced by the consistent maintaining of potential and kinetic energies.



(a) 151 cycles (600.0s) (b) 1 cycle (3.9s)

Figure 9. Simulated trajectories of the deterministic neurocontroller, with start marker (green), end marker (red), and path (blue).

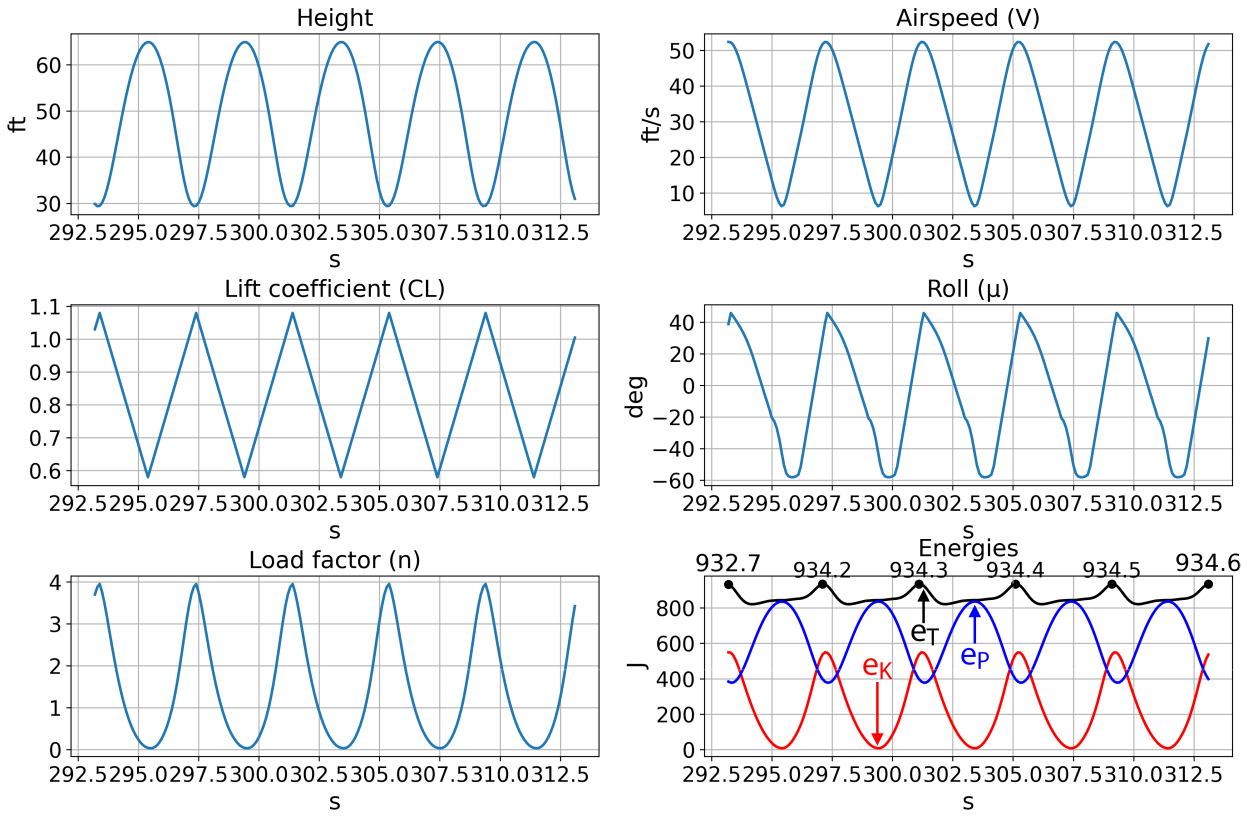
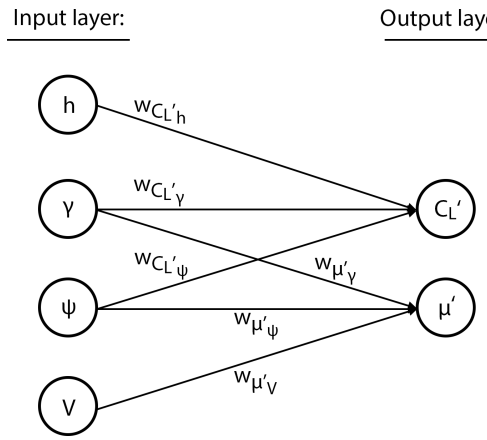


Figure 10. Trace of the deterministic neurocontroller.

B. Initial State-Robust Neurocontroller

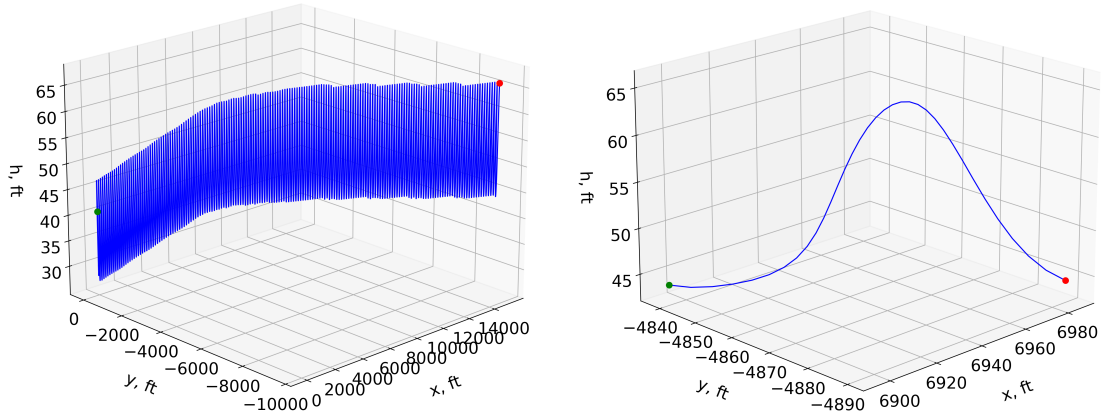
The deterministic neurocontroller would be ineffective in the more general and practical case where the initial aircraft states do not precisely match those seen during its evolution. Therefore, for the evolution of the neurocontroller shown in Fig. 11, the values of the state variables were all initialized from uniform distributions, which for the airspeed and height, spanned a range of ±25% of the nominal value. Despite these variations, the topology of the network evolved to have no hidden layers, only requiring airspeed, altitude, and pitch and heading angles as inputs.



$$b_{C'_L} = 1.43 \qquad w_{C'_{L\,h}} = 1.17$$
$$b_{\mu'} = -1.57 \qquad w_{C'_{L\,\gamma}} = -2.66$$
$$w_{C'_{L\,\psi}} = -0.530$$
$$w_{\mu'_\gamma} = 0.352$$
$$w_{\mu'_\psi} = -1.75$$
$$w_{\mu'_V} = 0.748$$

Figure 11. Topology of the initial-state-robust dynamic soaring neurocontroller.

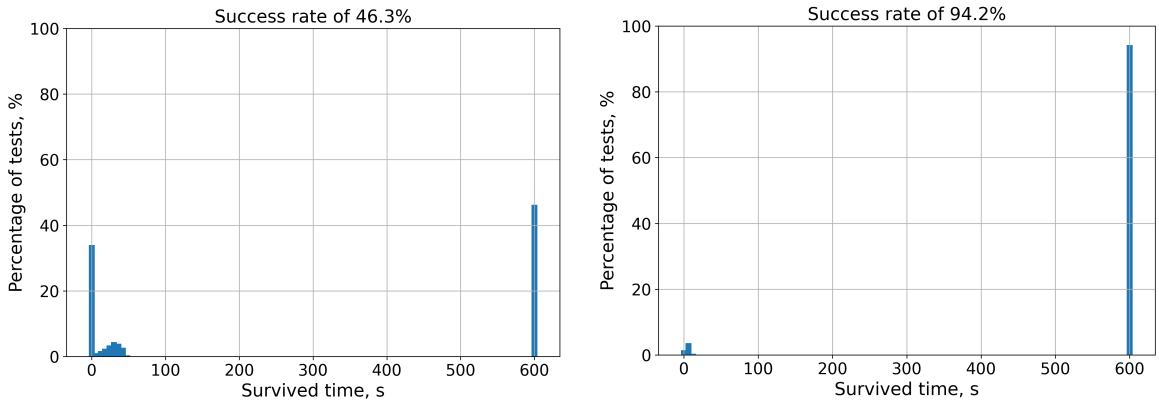
Initial velocity:	28.7 ft/s
Initial heading:	−18.5 deg
Initial pitch:	−32.3 deg
Initial height:	40.8 ft



(a) 172 cycles (600.0s) (b) 1 cycle (3.4s)

Figure 12. Simulated trajectories of the initial-state-robust neurocontroller.

The results of the Monte-Carlo simulations for the deterministic and initial-state-robust neurocontrollers tested in the variable initial state environment are plotted in Fig. 13. The height of each bar represents the percentage of the total number of tests where the model soared for the amount of time reflected in the horizontal axis.



(a) Deterministic neurocontroller (b) Initial state-robust neurocontroller

Figure 13. Monte Carlo robustness for deterministic and initial-state-robust neurocontrollers.

The success rate, defined as the percentage of the total number of tests for which the aircraft remained soaring for the maximum simulation time of 600.0s, shows that in the stochastic environment, the robust neurocontroller produces successful dynamic soaring trajectories twice as often compared to the deterministic network. The robust neurocontroller had been the member of the population that had consistently produced the highest fitness value, or equivalently had survived the greatest number of simulations to completion in the final generation of the robust evolutionary process. Therefore, it is able to exhibit sustained soaring for the range of parameters that were varied during evolution. The results of both controllers are also highly polarized towards immediate failure and continued soaring, since a controller would either fail to extract enough energy initially and soon crash, or quickly enter and maintain a sustainable dynamic soaring trajectory for the duration of the simulation.

Due to the stochastic nature of the environment, there will be certain combinations of initial aircraft states from which successful soaring is not possible. For instance, tests that start with the aircraft pointed downwards, or with a low initial height, are more likely to prematurely end in a crash, and as such, a certain probability of the failures in the Monte Carlo tests can be attributed to these unique cases. To better understand the individual sensitivities of the deterministic and robust neurocontrollers, Figures 14 and 15 show the survival times of the Monte Carlo simulations against each of the initial states that were varied in the stochastic test environment. Every data point represents the average time for which the aircraft remained soaring over all the simulations that were initialized with the discrete value of the initial parameter indicated on the horizontal axis. Comparison between the deterministic and robust controllers shows that the latter is able to more consistently perform dynamic soaring for the range of initial states that were evaluated. Furthermore, although there exists a variance in the sensitivities for each neurocontroller, the polynomial trendlines indicate that as expected, extreme pitch angles, low starting heights, and initial tailwinds are unfavorable for autonomous soaring. Mapping the space of initial parameter combinations into regions of feasible and infeasible soaring from the point of view of a hypothetical, optimal controller is not possible, as the existence of such a controller would solve the robust soaring problem. Therefore, the performance of any proposed robust neurocontroller must be measured and interpreted relative to that of other neural networks, which necessitates the deterministically-evolved neural network of the previous section.

C. Initial Wind-Robust Neurocontroller

Similar to the initial-state-robust neurocontroller, initial wind profiles were randomized during evolution to result in the initial-wind-robust neurocontroller of Fig. 16. The parameters of the wind profile A, h_{tr}, and W_{max}, were obtained from a uniform distribution spanning $\pm 50\%$ of the nominal values for each parameter.

Figures 17 and 18 show the sensitivities of the deterministic and initial-wind-robust neurocontrollers with respect to the ranges of randomly initialized wind profile parameters. Comparison of the two plots reveals

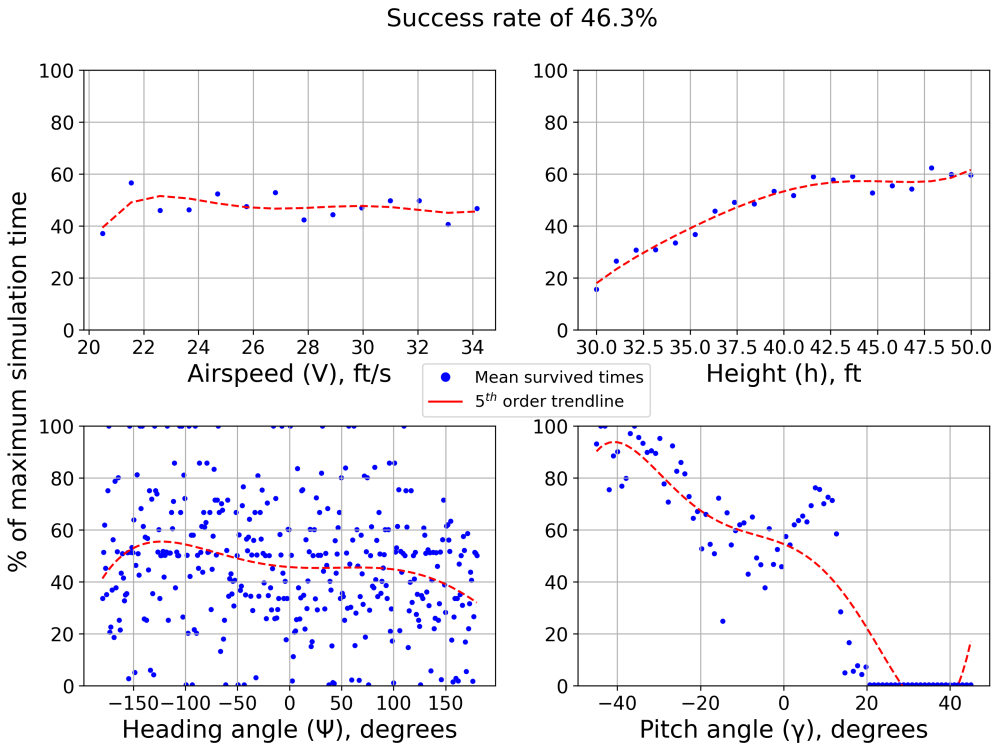


Figure 14. Initial state robustness for deterministic neurocontroller.

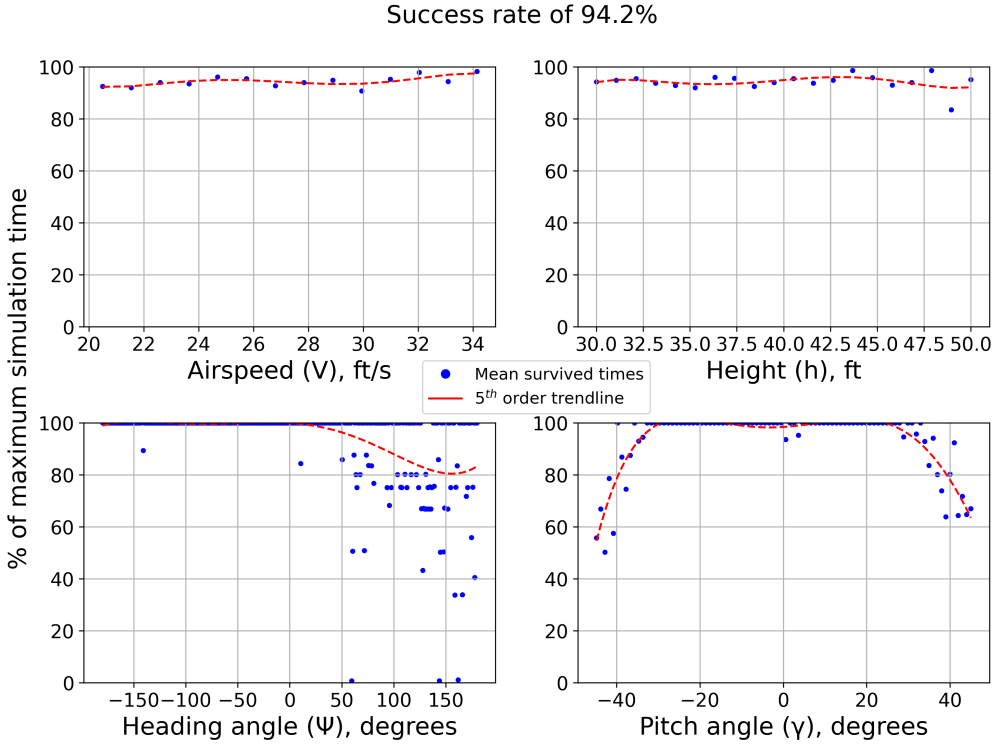


Figure 15. Initial state robustness for initial-state-robust neurocontroller.

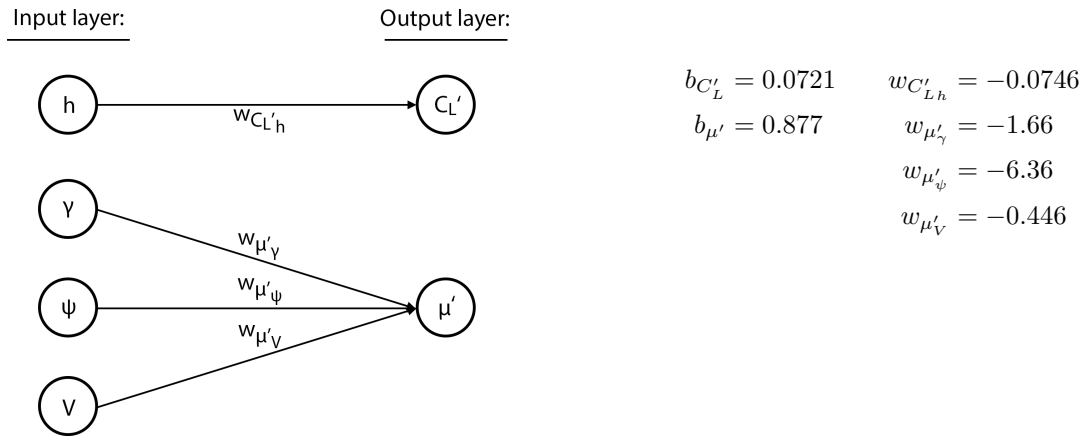


Figure 16. Topology of the initial-wind-robust neurocontroller.

that greater maximum wind strengths and lower transition heights are more favorable for soaring. The flight agent can extract greater amounts of energy in higher wind speeds that are more condensed across a narrower altitude range, which results in a stronger wind gradient that can be exploited for dynamic soaring. The tested range of the maximum wind strength did not exceed the value beyond which the aircraft model and its control surfaces would be overwhelmed and rendered ineffective. In addition, transition heights much larger than 40 ft and maximum wind strengths less than 40 ft/s are shown by the steep declines in performance to significantly interfere with variations in the other parameters, making the environment more hostile for autonomous soaring to the specific flight vehicle. Although the nominal wind parameters presented in Table 1 can lead to successful dynamic soaring trajectories, large deviations from any one of the values can severely impact the controller's success rate in stochastic environments. Furthermore, the Monte Carlo simulations can lead to an improved understanding of not only the neurocontroller, but also the environments that are best suited for certain flight techniques.

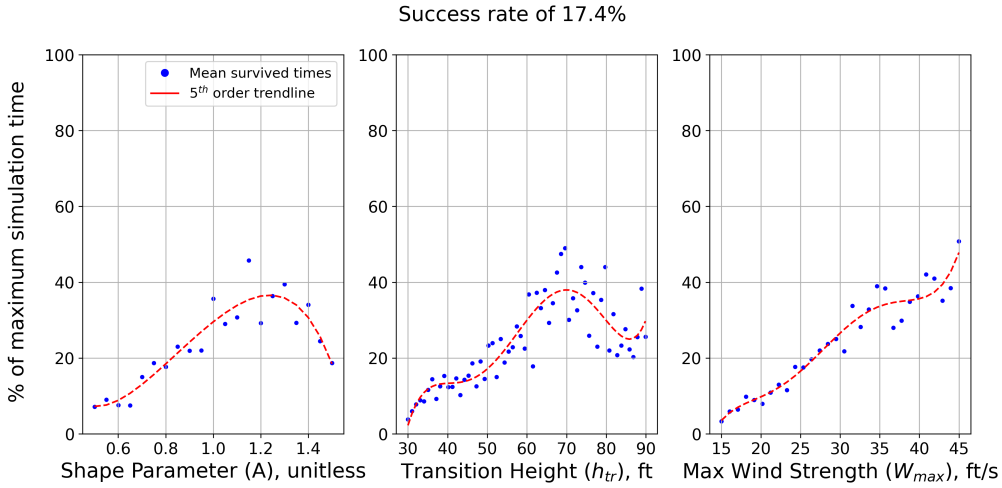


Figure 17. Initial wind robustness for deterministic neurocontroller.

Although the robust neurocontroller's success rate of 55.5% is less than ideal, the figure must be considered relative to those of the deterministic controller. It is difficult to determine what percentage of the wind profile parameter space used for robust evolution can support dynamic soaring maneuvers. Regardless, the threefold difference in the success rate compared to the deterministic network is indicative of the robust neuroevolutionary method's ability to target resilience to specific uncertainties.

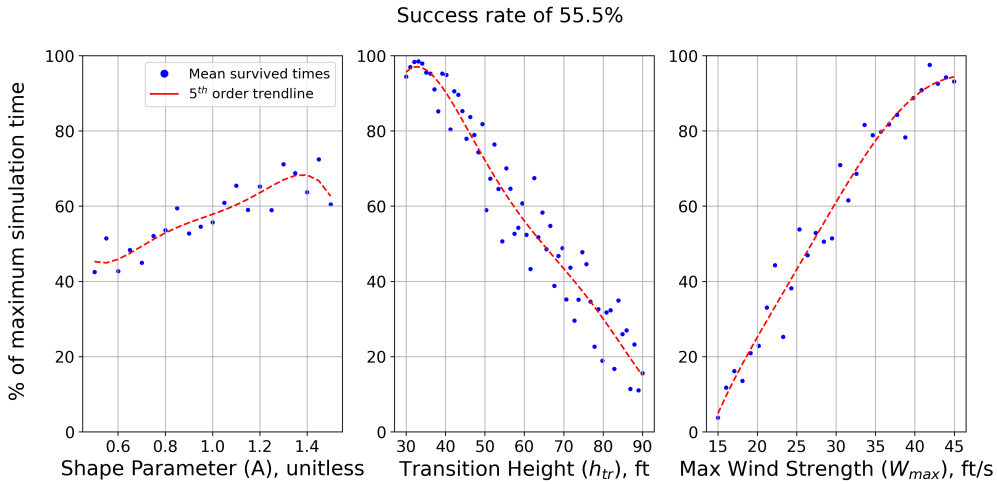
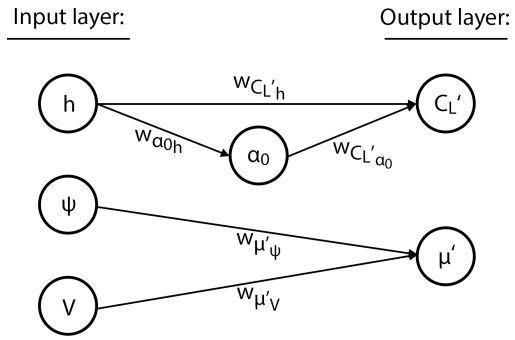


Figure 18. Initial wind robustness for initial-wind-robust neurocontroller.

D. Dynamic Wind-Robust Neurocontroller

Figure 19 presents one of the neurocontrollers evolved to be resilient to wind profiles that changed throughout the flight simulation. For a given run, the value of one of the three wind parameters was sinusoidally varied by a randomly sampled percentage of the nominal value, obtained from a uniform distribution from 0 to 100%. It was therefore possible to have the maximum wind strength or transition height fluctuate between 0 and twice the nominal value, or the shape parameter to vary between completely logarithmic and exponential profiles, should the sampled amplitude had been equal to 100%. Regardless of the parameter, however, the fluctuation of any of the three would be equivalent to a sinusoidal variation in the wind gradient's magnitude. Lastly, a random value was also selected for the oscillation frequency from a uniform distribution between 0 and 1 Hz.

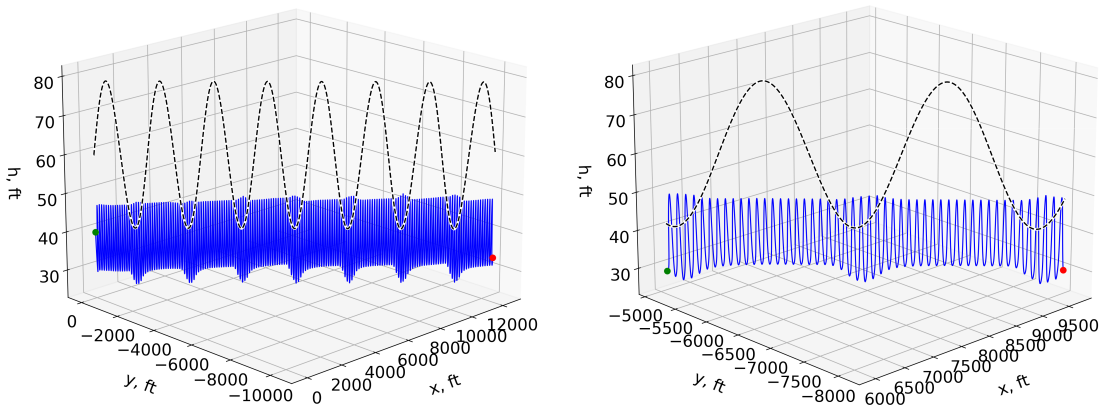


$$b_{C'_L} = 3.86 \qquad w_{C'_{L_h}} = -1.63$$
$$b_{\mu'} = -0.343 \qquad w_{\alpha_{0_h}} = -0.879$$
$$b_{\alpha_0} = -0.0125 \qquad w_{C'_{L_{\alpha_0}}} = 0.164$$
$$w_{\mu'_\psi} = -2.12$$
$$w_{\mu'_V} = 1.43$$

Figure 19. Topology of the dynamic wind-robust neurocontroller.

For this environment, the effects of the stochastic mechanism on the evolved controller can be visualized by the distinct and unique patterns of the resulting trajectories. Specifically, Fig. 20 depicts a simulation with a dynamically varying transition height. The transition height's value at every time step of the trajectory can be obtained from the height of the pure sinusoid (dashed, black). In the case of Fig. 20a, the transition height begins increasing from a value of 60 ft.

Dynamic parameter:	h_{tr}
Amplitude:	18.9 ft
Frequency:	0.0785 Hz



(a) 171 cycles (600.0s) (b) 50 cycles (176.1s)

Figure 20. Simulated trajectories of the dynamic wind-robust neurocontroller.

The oscillating nature of the aircraft's height is a direct result of the simulation's time-varying wind profile. Even though as the transition height increases and the vertical wind gradient becomes more diluted over a larger altitude range, the controller correspondingly gains altitude between cycles to reach regions of higher wind strength. When the transition height decreases and eventually surpasses the peak height of the soaring cycles, the aircraft experiences zero wind gradient for an increasingly larger fraction of its cycles. To avoid being in this saturation region of maximum wind strength, where dynamic soaring is not feasible, the controller sharply decreases its soaring altitude, before again pursuing the rising transition height. This indicates that the neural network had evolved a successful soaring strategy even in the presence of environmental changes.

To examine the relative sensitivities of the deterministic and dynamic wind-robust neurocontrollers to the amplitude and frequency of each of the three wind profile parameters, the Monte Carlo simulation results are plotted in Figures 21 and 22. The trendlines show that neither controller is particularly sensitive to specific oscillation amplitudes and frequencies. Nevertheless, the robust network exhibits a greater survival time across all the tested values, with the overall success rate of 94.4% indicating that most of the parameters within the dynamic parameter space are conducive to soaring.

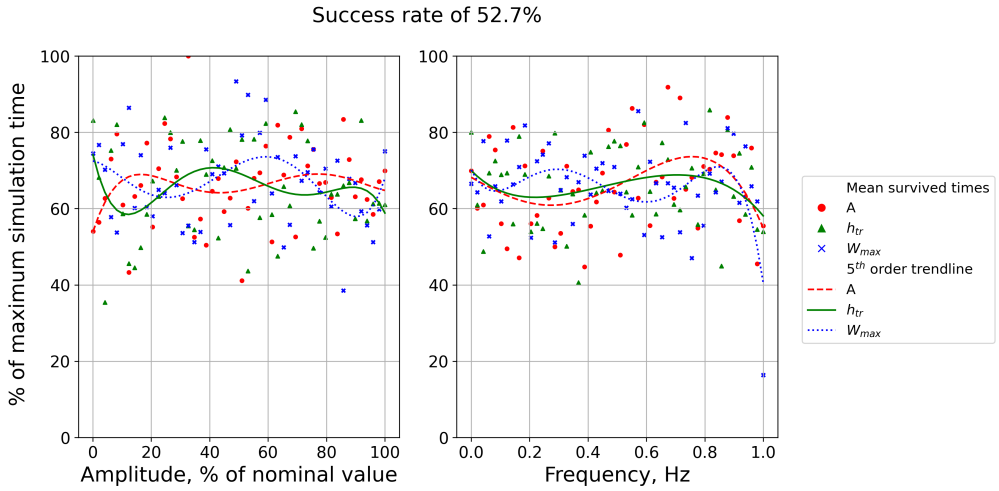


Figure 21. Initial wind robustness for deterministic neurocontroller.

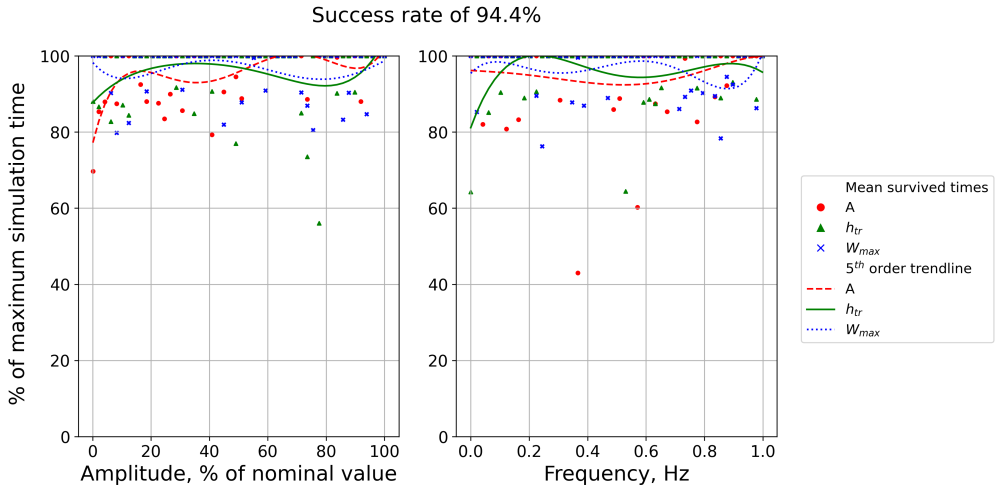


Figure 22. Initial wind robustness for dynamic wind-robust neurocontroller.

IV. Conclusions and Future Work

In this paper, the NEAT algorithm is applied to generate topologically sparse and robust neurocontrollers for autonomous dynamic soaring. This robust neuroevolutionary approach, based on the NEAT algorithm, was shown to create neural networks capable of controlling an SUAV model along sustained soaring trajectories for a variety of initial states, wind profiles, and changing parameters. Each of the robust neurocontrollers presented in this work showed a significant improvement in soaring success rate when compared to the performance of a deterministically-evolved network. In particular, the Monte Carlo simulations of the initial-state and initial-wind-robust networks exhibited results that would be expected from an aerodynamics perspective, and the trajectories of the dynamic-wind-robust controller provided insight into the way in which the neural network had evolved to adapt the necessary resilience.

The modeling of the stochastic environments was primarily motivated by the need for a simulation tool using which the evolutionary method could be tested, and less as a high-fidelity representation of real-world conditions. Therefore, although the extreme variation of certain environmental parameters in the time-varying simulations may not necessarily be observed in reality, the stochastic models provided a method of evaluating the resilience of the resulting controllers to persistent and regularly fluctuating conditions.

Beyond the environments tested in this paper, the presented neuroevolutionary control approach can be generalized to other uncertainty models and applied to different control tasks. The investigation of other autonomous soaring techniques in the presence of additional variations and disturbances will be explored in the future.

V. Acknowledgments

The authors gratefully acknowledge support by a Canadian Defence Academy Research Program Grant, No.757734, Enhancing UAV Persistent Surveillance through AI-Driven Optimal Atmospheric Energy Extraction.

References

[1]Denny, M., "Dynamic Soaring: Aerodynamics for Albatrosses," *European Journal of Physics*, Vol. 30, No. 1, Jan. 2009, pp. 75–84.

[2]Zhao, Y. J., "Optimal Patterns of Glider Dynamic Soaring," *Optimal Control Applications and Methods*, Vol. 25, No. 2, 2004, pp. 67–89.

[3]Sachs, G., "Minimum Shear Wind Strength Required for Dynamic Soaring of Albatrosses," *Ibis*, Vol. 147, No. 1, 2005, pp. 1–10.
[4]Bird, J. J., Langelaan, J. W., Montella, C., Spletzer, J., and Grenestedt, J. L., "Closing the Loop in Dynamic Soaring," *AIAA Guidance, Navigation, and Control Conference*, AIAA SciTech Forum, American Institute of Aeronautics and Astronautics, Jan. 2014.
[5]Akhtar, N., Whidborne, J. F., and Cooke, A. K., "Real-Time Optimal Techniques for Unmanned Air Vehicles Fuel Saving," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, Vol. 226, No. 10, Oct. 2012, pp. 1315–1328.
[6]Gao, C. and Liu, H. H. T., "Dubins Path-Based Dynamic Soaring Trajectory Planning and Tracking Control in a Gradient Wind Field," *Optimal Control Applications and Methods*, Vol. 38, No. 2, 2017, pp. 147–166.
[7]Lawrance, N. R. J. and Sukkarieh, S., "A Guidance and Control Strategy for Dynamic Soaring with a Gliding UAV," *2009 IEEE International Conference on Robotics and Automation*, Kobe, Japan, May 2009, pp. 3632–3637.
[8]Deittert, M., Richards, A. G., Toomer, C., and Pipe, A., "Dynamic Soaring Flight in Turbulence," *AIAA Guidance Navigation and Control Conference, Chicago*, Aug. 2009.
[9]Liu, Y., Longo, S., and Kerrigan, E. C., "Nonlinear Predictive Control of Autonomous Soaring UAVs Using 3DOF Models," *2013 European Control Conference (ECC)*, July 2013, pp. 1365–1370.
[10]Montella, C. and Spletzer, J. R., "Reinforcement Learning for Autonomous Dynamic Soaring in Shear Winds," *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Chicago, IL, USA, Sept. 2014, pp. 3423–3428.
[11]Kim, S.-h., Lee, J., Jung, S., Lee, H., and Kim, Y., "Deep Neural Network-Based Feedback Control for Dynamic Soaring of Unpowered Aircraft," *IFAC-PapersOnLine*, Vol. 52, No. 12, Jan. 2019, pp. 117–121.
[12]Perez, R. E., Arnal, J., and Jansen, P. W., "Neuro-Evolutionary Control for Optimal Dynamic Soaring," *AIAA Scitech 2020 Forum*, American Institute of Aeronautics and Astronautics, Jan. 2020.
[13]Stanley, K. O. and Miikkulainen, R., "Evolving Neural Networks through Augmenting Topologies," *Evolutionary Computation*, Vol. 10, No. 2, June 2002, pp. 99–127.
[14]Kim, E. J. and Perez, R. E., "Neuroevolutionary Control for Autonomous Soaring," *Aerospace*, Vol. 8, No. 9, Sept. 2021, pp. 267.
[15]Rayleigh, "The Soaring of Birds," *Nature*, Vol. 27, No. 701, April 1883, pp. 534–535.