Efficient and Resilient Micro Air Vehicle Flapping Wing Gait Evolution for Hover and Trajectory Control

Amor A. Menezes[a,*], Pierre T. Kabamba[b]

[a]*California Institute for Quantitative Biosciences, University of California, Berkeley, 2151 Berkeley Way, Berkeley, CA 94704-5230, USA*
[b]*Deceased September 20, 2014. Was with the Department of Aerospace Engineering, University of Michigan, 1320 Beal Avenue, Ann Arbor, MI 48109-2140, USA*

Abstract

This paper deploys a recently proposed, biologically inspired, on-line, search-based optimization technique called Selective Evolutionary Generation Systems (SEGS) for control purposes; here, to evolve Micro Air Vehicle (MAV) flapping wing gaits in changing flight conditions to maintain hovering flight and track trajectories in unsteady airflow. The SEGS technique has several advantages, including: (1) search-efficiency, by optimally trading off prior search space information for search effort savings as quickly as possible in dynamic environments; (2) model-independence, as in biology, avoiding biases induced by built-in models rendered incorrect by environment changes; and (3) resilience, through sufficiency for stochastic behavior that is itself sufficient for responsiveness to search-objective variations caused by environment fluctuations. This work presents the first approach that can simultaneously evolve optimal MAV flapping wing gaits efficiently and resiliently, adapt on-line, and, via model-independence, allow feedback from either experimental sensors or alternate external models (affording control versatility for hover or forward flight, unsteady or quasi-steady aerodynamics, and any dynamics or wing kinematics). Performance benchmarks are also provided. Because the (1+1)-Evolution Strategy (ES) and the Canonical Genetic Algorithm with Fitness Proportional Selection (CGAFPS) are two SEGS special extreme cases, an additional comparison showcases SEGS possession of both (1+1)-ES computational speed and CGAFPS resilience.

Highlights

1. A bioinspired, search-efficient, tunable optimization scheme is adapted for control.
2. Micro Air Vehicle (MAV) flapping wing gaits are evolved on-line, model-independently.
3. Scheme properties are benchmarked in a case study of evolution for MAV hover control.
4. Scheme speed and responsiveness compare favorably to related evolutionary methods.
5. A second study attains MAV trajectory control in unsteady flow with little computing.

Keywords: Micro Air Vehicles (MAVs), flapping wing gait evolution, selective evolutionary generation, hovering flight, trajectory tracking

1. Introduction

Flapping wing flight is a promising enabling technology for Micro Air Vehicles (MAVs), which are aircraft of maximal dimension less than 15 cm flying at airspeeds around 10 m/s (Shyy et al., 2008). Compared to their fixed wing and rotary wing brethren, flapping wing MAVs can accomplish reconnaissance, surveillance, engagement, or environmental monitoring tasks at lower airspeeds and with greater maneuverability and quietness. Examples of prototype flapping wing MAVs include Conn et al. (2006), Wood (2008), Yang et al. (2009), Fenelon (2009), De Croon et al. (2009), Keennon et al. (2012), and Arabagi et al. (2012).

Varying a flapping wing gait changes MAV lift and maneuverability, and mission phases may dictate the suitability, or fitness, of a particular flapping wing gait. For instance, an MAV may scout a target by favoring a hovering form of flapping flight, engage the target after increasing the fitness of descending flapping wing gaits, and then quickly escape after deeming ascending gaits to be the most fit. External environment variations within each mission phase, like wind gusts and direction fluctuations, can also affect MAV lift and thus constantly perturb flapping wing gait fitness. The small size of an MAV makes

*Corresponding author
Email address: amenezes@berkeley.edu (Amor A. Menezes)

it particularly susceptible to physical or environmental perturbations, necessitating continual fitness maximization through corresponding variations of the flapping wing gait to ensure altitude attainment and forward-flight trajectory tracking.

Current optimization of a flapping wing gait at low Reynolds number requires multiple iterations of computationally expensive three-dimensional flow simulations on multiple processors taking hours or days to complete (Trizila et al., 2008; Persson et al., 2010; Nielsen and Diskin, 2013; Vandenheede et al., 2014; Gogulapati et al., 2014) depending on the desired fidelity (Vandenheede et al., 2014). Moreover, these simulations are based on flow model physics for which open issues still persist (Shyy et al., 2010). Thus, there is a need for on-line, i.e., real-time, MAV control approaches to maximize the fitness of flapping wing gaits during flight. This paper focuses on such on-line control and optimization, and is based on preliminary results from the doctoral dissertation of Menezes (2010).

1.1. Related Work

Comprehensive reviews of the dynamics and control literature applicable to flapping wing MAVs are available in Orlowski and Girard (2012) and Taha et al. (2012). The seminal works of Ellington (1984) and Ellington et al. (1996) pointed out that unsteady aerodynamic mechanisms are responsible for insect flight, with implications for MAVs as documented in Ellington (1999). But it is quasi-steady/blade-element aerodynamic models that are often deployed in flapping wing MAV control studies, for reasons of computational efficiency (Orlowski and Girard, 2012). Examples include Khan and Agrawal (2005) that is utilized in Khan and Agrawal (2007) to construct a differential flatness based controller, Deng et al. (2006b) that is the basis for the LQR technique in Deng et al. (2006a) (although Orlowski and Girard (2012) notes that it is possible to implement other aerodynamic models with the presented dynamics), and the experimentally derived aerodynamic model in Sane and Dickinson (2001) that is used to drive the control simulation in Doman et al. (2010).

If unsteady aerodynamic theory is desired, then a relevant model is the rigid airfoil work of Theodorsen (1935) that examined lift and that was extended in Garrick (1936) to predict the horizontal thrust force generated by a flapping airfoil. An empirical state-space representation of Theodorsen (1935) was developed and utilized in Brunton and Rowley (2013) to robustly track a reference lift coefficient and reject gust disturbances. An alternative aerodynamic approach to that of unsteady or quasi-steady aerodynamics is that of surrogate modeling as described in Trizila et al. (2008).

Contemporary control-theoretic (Yan et al., 2001; Schenato, 2003; Lin et al., 2010; Ratti et al., 2011; Pérez-Arancibia et al., 2013; Duhamel et al., 2013; Caetano et al., 2013) and neurobiological (Weng et al., 2007; Chung and Dorothy, 2010) approaches to MAV flapping wing gait regulation exploit dynamical system models and experiments to improve performance. Hovering flight is the typical mode of flapping wing MAV operation in these cases, and there are few works that address the problem of forward-flight trajectory tracking. Menezes (2010) described the first high-fidelity simulation of forward-flight trajectory control of a flapping wing MAV in unsteady airflow, which is control that was achieved by indirect simultaneous lift and thrust modification using Theodorsen (1935) and Garrick (1936). More recently, Malhan et al. (2012) looked at experimental forward flight performance, Mahjoubi and Byl (2013) decoupled flapping wing MAV lift and thrust using a motion controller that modified mechanical impedance properties of the wings rather than their stroke characteristics, and Goppert et al. (2014) considered how to verify and validate a trajectory tracking controller that was subject to disturbances.

1.1.1. Evolutionary Approaches, On-line Adaptive Approaches, and Model-Independent Approaches

Early exploratory works applying evolutionary computation to flapping wing MAVs include Augustsson et al. (2002), Salles and Schiele (2004), Milano and Gharib (2005), Hunt et al. (2005), Van Breugel and Lipson (2005), Regan et al. (2006), Mouret et al. (2006), Shim and Kim (2006), de Margerie et al. (2007), Boddhu and Gallagher (2010), and Roberts et al. (2010). In the following years, Olhofer et al. (2011) suggested a way to combine evolutionary algorithms with physical measurements to facilitate the experimental evaluation of flapping wing optimization solutions, Doncieux and Hamdaoui (2011) used multi-objective evolutionary algorithms to generate a set of Pareto optimal controllers for further study, Gallagher and Oppenheimer (2012) put forward an 'Evolvable and Adaptive Hardware (EAH) oscillator that will replace a traditional oscillator inside a conventionally defined flight controller with the goal of restoring whole-system control efficacy in the face of damage to the vehicle' by providing the 'ability to correct for vehicle anomalies via flight control adjustment in-flight and while conducting normal missions,' and Gallagher (2013) posited an evolutionary algorithm to allow swarms of flapping wing MAVs to cooperatively and quickly find and correct common and uncommon MAV damages in the swarm.

The latter two references represent the first progress towards control schemes that are on-line, adaptive and capable of handling internal and externally-induced parameter variations. Gallagher et al. (2014) proceeds further along these lines by considering 'adaptive learning [that] is used to simultaneously produce fault-correcting solutions as well as testable hypotheses that can be used to identify areas of inconsistency between the adapted system and its model,' which can then be corrected.

Recently, Pérez-Arancibia et al. (2015) came out with a model-independent experimental method to find a control strategy for achieving stable hovering flight of a flapping wing MAV. But the present paper remains the only work that simultaneously is an evolutionary approach to MAV flapping wing gait optimization, is capable of on-line adaptation, is model-independent in a way that allows it to obtain feedback from experimental sensors or from other external models acting in the place of sensors (which thereby afford a versatile control capacity to the MAV that can deal with hover, forward-flight, aerodynamics that are

chosen to be unsteady or quasi-steady, and any dynamics or wing kinematics), and is a process that is optimally search-efficient and also resilient to environmental or internal-parameter fluctuations. The motivation for, and importance of, all of these concepts is explained next.

1.2. The Motivation for Model-Independence

Although the flapping wing gait regulation techniques in Yan et al. (2001), Schenato (2003), Deng et al. (2006a), Weng et al. (2007), Chung and Dorothy (2010), Doman et al. (2010), Lin et al. (2010), Ratti et al. (2011), Pérez-Arancibia et al. (2013), Duhamel et al. (2013), and Caetano et al. (2013) can control a flapping wing MAV, the approaches are model-dependent; that is, these strategies harness knowledge of MAV dynamics and are tailored to defined narrow ranges of MAV parameters to guarantee robustness and disturbance-rejection properties. However, because of the small size of an MAV, its dynamics and parameters can vary drastically with payload gain or loss, traversal in constricted or open spaces, and particulate or droplet deposition that increases weight or asymmetrically affects flapping. Therefore, an MAV cannot have its parameter ranges limited by a control strategy during each mission objective or scenario if it is to have sufficient freedom to respond to all possible physical or environmental perturbations. Any attempt to augment control-theoretic techniques and their predefined parameter ranges with adaptive observers and estimators to account for perturbations or to deal with unsteady airflow complications only serves to increase MAV computational load. Further, the traditional computationally-inexpensive on-line solution of a stored lookup table robs an MAV of flexibility in its response to unexpected changes in its internal or external conditions.

Thus, an alternative approach to flapping wing MAV control that is flexible and less dependent on a model may be preferable to the scheduled deployment of multiple model-dependent control-theoretic strategies that are designed for different conditions and situations. This paper employs such an approach, and the work is distinguishable from the current related literature on the basis of this philosophy. An example of an approach that is less model-dependent than control-theoretic methods is an on-line search-based strategy that continually maximizes the time-varying fitness of MAV flapping flight by seeking desirable gait parameters from among a bounded space of all possible gait parameters. In this example, the search process occurs according to a rationale that corresponds only partially (instead of fully) to some underlying flapping wing model dynamics, and these dynamics are typically inferred rather than imposed.

Such time-varying fitness constitutes feedback to the search process, but what determines fitness is not specified by the search process by virtue of its model-independence. That is, the search process can be connected to any experimental sensor input or to any model output that acts as a sensor input, and what is considered maximal fitness may be any of a number of different definitions, one of which is stipulated at the time of design. The search process simply works to maximize fitness in an on-line fashion in a dynamic environment regardless of how fitness is defined and of how complex the model or sensor is that produces the fitness values for the process.

1.3. The Motivation for Efficient and Resilient Search

Since the airflow past a flapping wing is unsteady, any function of lift for the fitness objective for the search process is highly nonlinear with local optima, which precludes the use of traditional gradient-based search optimization methods. Moreover, optimization of the flapping wing gait to meet lift requirements must be performed on-board the MAV by a controller that has other flight responsibilities in addition to flapping wing gait selection; consequently, its computing resources cannot be devoted solely to completing an optimization process prior to each gait's selection. In fact, any strategy that involves sequentially repeating a previously-completed optimization (even over multiple gait selection cycles, typically under the assumption that the environment is static for the duration of the optimization) can be ruled out. This is because such an approach to achieving resilience to changing fitnesses is computationally-expensive: the scheme essentially determines, repeatedly, a stationary probability mass function on the space of gait parameters that is a delta function at the location of a suitable flapping wing gait (i.e., repetitive off-line optimization). But a truly resilient scheme that is responsive to fitness perturbations is one that changes the distribution over the gait parameters at the same time as MAV performance and fitness changes. Intuitively, resilience implies that under-performing fitness gaits have a low, non-zero realization probability, so that they can be infrequently sampled and evaluated. Once fitness changes occur, and the low-probability but now high-performing gaits are realized, a resilient strategy changes the gait parameter distribution to reflect changes in performance and fitness.

Hence, the motivation for an evolutionary search-based method of flapping wing gait fitness optimization that efficiently explores the space of flapping wing gait parameters to produce a stationary probability distribution over the gait parameters that is a function of time-varying fitness. Efficiency here refers to a scheme that trades off prior information about the search space for search effort savings as quickly as possible (Jaynes, 1981). It has been shown (Jaynes, 1981; Menezes and Kabamba, 2014, 2016) that the most advantageous search-based strategy in a dynamic environment is one that is completely independent of underlying model dynamics, because any reliance on a surmised model biases and slows searches when changes in operating conditions invalidate that model. Thus, the biologically inspired optimization technique that is deployed in this paper lacks bias as a result of independence from modeled dynamics. Instead, a heavy dependence on external stimuli exists, mimicking flapping wing insect behavior. Additionally, the algorithm is responsive and resilient to fitness perturbations in their aftermath because it is not hindered by model inaccuracies caused by the perturbations. Further, the desired responsiveness, model-independence,

and global optimization properties are attained through only local, simplistic, comparative decisions, which render the scheme computationally-inexpensive. The theory also shows (Menezes and Kabamba, 2016) that the strategy is not gradient-based, but instead dwells longer in states that are more fit.

This paper's contributions are unique because, as will be demonstrated, resilient flapping wing gait optimization is achieved efficiently even in complicated settings without significant computation.

1.4. Goals

The objectives of this paper are to

1. summarize the recently proposed optimization algorithm called Selective Evolutionary Generation Systems (SEGS) to facilitate practical implementation for control, and to highlight the technique's information-theoretic and dynamic properties that result in efficiency and resilience, respectively (Section 2);
2. demonstrate the efficacy of the SEGS method in evolving MAV flapping wing gaits on-line to achieve hovering flight (Section 3.1);
3. benchmark the SEGS algorithm's properties on the flapping wing MAV hover problem (Section 3.2);
4. compare the SEGS technique's performance on the flapping wing MAV hover problem to that of known evolutionary search strategies that are special SEGS cases (Section 3.3); and
5. validate the SEGS scheme's evolution of MAV flapping wing gaits for trajectory control in complex conditions like unsteady airflow with coupled aerodynamic forces (Section 4).

1.5. Novelty

A preliminary version of a SEGS approach to MAV flapping wing gait estimation appears in the doctoral dissertation of Menezes (2010). The SEGS technique is introduced as a Markov chain Monte Carlo method in Menezes and Kabamba (2014), and its theory and its modeling of biology are fully documented in Menezes and Kabamba (2016). Sections 1 and 3–6 have never been published in the peer-reviewed literature. Section 2 is an adaptation of the theory in Menezes and Kabamba (2014, 2016).

1.6. Outline

The remainder of this paper is as follows. Section 2 recaps the theoretical underpinnings of the SEGS technique described in detail in Menezes and Kabamba (2014, 2016), and distills the core algorithmic procedure. Section 3 applies this procedure to estimate MAV flapping wing gaits on-line to achieve hovering flight, provides various technique metrics, and compares performance to two related evolutionary search strategies. Section 4 illustrates algorithm applicability and success in a realistic flapping wing MAV trajectory control scenario. Section 5 highlights limitations that can be observed in the two case studies of flapping wing gait evolution for hover and trajectory control, and suggests possible ways to address them. Section 6 presents concluding remarks.

2. A Review of Selective Evolutionary Generation Systems

This section outlines a model-independent fitness optimization method that searches a space to produce a stationary probability distribution over the elements of that space, with the produced stationary probability distribution being a function of the time-varying fitness of the elements. The method is a Markov chain Monte Carlo one that can be search-efficient (Menezes and Kabamba, 2014) and that also models evolution and responsiveness in dynamic environments (Menezes and Kabamba, 2016). For optimization, the stationary probability distribution that is produced by the Markov chain Monte Carlo method must reflect higher stationary probabilities for elements with higher fitness.

Accordingly, we first formulate the search problem, then postulate a method that uses local decisions to achieve the desirable global stationary probability distribution that maximizes fitness, and finally highlight the method's properties of search-efficiency, resilience to fitness perturbations and relationship to two common evolutionary search algorithms. More theoretical detail is available in Menezes and Kabamba (2016).

2.1. Problem Definition

Let X be a search space with elements x_i, $1 \leq i \leq n$. In the context of evolutionary computation, X is the set of genotypes; for instance, X is the set of permissible flapping wing gaits. The search problem seeks a probability mass function $\phi_X : X \to \mathbb{R}^+$ that accomplishes the specified objective below, and dynamic transition laws that cause X to be distributed according to ϕ_X. Let $z : X \to Z$ be an unknown, computable, and possibly changing function that we are interested in. The set Z is a metric space, the set of phenotypes. For example, z is the unknown mapping between flapping wing gaits and their resultant coefficients of

lift, the set of which is Z. Suppose that we are given a desired element z_{des} in the image of z, and we wish to find $x \in X$ such that $\|z(x) - z_{des}\|$ is small (i.e., $z(x) \approx z_{des}$). Formally, we want a ϕ_X that helps achieve a known expected value $Y \geq 0$, i.e.,

$$\mathrm{E}_{\phi_X}[\|z(x) - z_{des}\|] = Y. \tag{1}$$

In the above, Y is effectively a tolerance, i.e., it is the acceptable mean distance between candidates in the image of z compared to the desired image value. Let $y(x) = \|z(x) - z_{des}\|$. The scheme to find ϕ_X should be efficient in that it trades off prior information about X for search effort savings as quickly as possible.

Let $f : Z \to \mathbb{R}^+$. We allow the method to employ a function $F : X \to \mathbb{R}^+ : x \mapsto F(x) = (f \circ z)(x) = f(z(x))$, a real-valued, positive fitness function. We assume that for any $x_i \in X$, $\phi_X(x_i)$ is a differentiable function of the fitnesses $F(x_1), \ldots, F(x_n)$. We desire $\phi_X(x_i)$ to be responsive to perturbations, i.e., for all $x_j \in X$,

$$\frac{\partial \phi_X(x_i)}{\partial F(x_j)} \neq 0. \tag{2}$$

2.2. The Selective Evolutionary Generation Systems Algorithm

To search in X, a Markov chain Monte Carlo (Brémaud, 1999a) method is postulated that makes use of a *selective evolutionary generation system*, which is a quintuple $\Gamma = (X, R, P, G, F)$, where

- X is the set of genotypes, $X = \{x_1, x_2, \ldots, x_n\}$;
- R is a set of resources whose elements can be utilized to transition between genotypes, $R = \{r_1, r_2, \ldots, r_m\}$;
- $P : R \to (0, 1]$ is a probability mass function on R, given by $P(r_i) = \Pr[\mathcal{R} = r_i] = p_i$, $\sum_{k=1}^{m} p_k = 1$;
- $G : X \times R \to X$ is a mapping, called a generation function, from one genotype to another using a resource from R;
- $F : X \to \mathbb{R}^+$ is a positive function that evaluates genotype fitness;
- X is reachable (Cormen et al., 2009) through G and R; and
- the dynamics of the system are given by

$$\mathcal{X}(t+1) = \mathit{Select}(\mathcal{X}(t), G(\mathcal{X}(t), \mathcal{R}(t)), N), \tag{3}$$

where $\mathit{Select} : X \times X \times [0, \infty) \to X$ is a random function such that if $x_1 \in X$ and $x_2 \in X$ are any two genotypes, and $N \in [0, \infty)$ is the *level of selectivity*, then

$$\mathit{Select}(x_1, x_2, N) = \begin{cases} x_1 & \text{with probability } \frac{F(x_1)^N}{F(x_1)^N + F(x_2)^N}, \\ x_2 & \text{with probability } \frac{F(x_2)^N}{F(x_1)^N + F(x_2)^N}. \end{cases} \tag{4}$$

In (3), $\mathcal{X}(t)$ denotes the realization of a random genotype at time t, $\mathcal{R}(t)$ denotes the realization of a random resource at time t, $G(\mathcal{X}(t), \mathcal{R}(t))$ denotes the outcome genotype mapped from the realized genotype at time t utilizing the resource at time t, and $\mathcal{X}(0)$ has a known probability mass function. Also in (3), the probability of a genotype realization at some future time given the present genotype realization is conditionally independent of the past time history of genotype realizations. Thus, the dynamics of a selective evolutionary generation system form a discrete-time homogeneous Markov chain (Brémaud, 1999b).

The *Select* function has a number of interesting properties (Menezes and Kabamba, 2016), including that for all N,

$$\frac{\Pr[\mathit{Select}(x_1, x_2, N) = x_1]}{\Pr[\mathit{Select}(x_1, x_2, N) = x_2]} = \left(\frac{F(x_1)}{F(x_2)}\right)^N. \tag{5}$$

That is, the ratio of the probabilities of selecting any two genotypes is equal to the ratio of their respective fitnesses raised to the power N. This property is called *local rationality*, where "rational" refers to the ratio of the probabilities and is a historical term that does not imply any agency (recall that a rational number is a ratio of integers). For any $x_i, x_j \in X$ and $r_k \in R$ of the selective evolutionary generation system $\Gamma = (X, R, P, G, F)$, we can define the *descendancy tensor*, δ, with elements

$$\delta_{ijk} = \begin{cases} 1 & \text{if } x_j = G(x_i, r_k),\ 1 \leq i \leq n,\ 1 \leq j \leq n,\ 1 \leq k \leq m, \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

Hence, the descendancy tensor indicates whether it is possible to produce offspring x_j in one step from progenitor x_i via generation function G that employs a resource r_k. We can use this tensor to create a matrix that represents the conditional probability of transitioning to x_j from x_i, by utilizing the probability of selecting each available element in R and summing over all m elements. The matrix $\boldsymbol{\gamma}$, called the *unselective matrix of transition probabilities*, has elements

$$\gamma_{ij} = \Pr[\text{offspring is } x_j \mid \text{progenitor is } x_i] = \sum_{k=1}^{m} \delta_{ijk} p_k, \ 1 \leq i \leq n, \ 1 \leq j \leq n, \tag{7}$$

and is a stochastic matrix (Menezes, 2010). The *matrix of transition probabilities*, $\mathbf{P}$, has elements

$$P_{ij} = \Pr[\mathcal{X}(t+1) = x_j \mid \mathcal{X}(t) = x_i], \tag{8}$$

$$= \begin{cases} \Pr[Select(x_i, x_j, N) = x_j \mid \mathcal{X}(t) = x_i] \\ \times \Pr[\text{offspring is } x_j \mid \text{progenitor is } x_i], & \forall j \neq i, \\ \\ \Pr[Select(x_i, x_i, N) = x_i \mid \mathcal{X}(t) = x_i] \\ \times \Pr[\text{offspring is } x_i \mid \text{progenitor is } x_i] \\ + \sum\limits_{\substack{k=1 \\ k \neq i}}^{n} \Pr[Select(x_i, x_k, N) = x_i \mid \mathcal{X}(t) = x_i] \\ \times \Pr[\text{offspring is } x_k \mid \text{progenitor is } x_i], & \text{if } j = i, \end{cases} \tag{9}$$

$$= \begin{cases} \frac{1}{1+\left(\frac{F(x_i)}{F(x_j)}\right)^N} \gamma_{ij}, & \forall j \neq i, \\ \gamma_{ii} + \sum\limits_{\substack{j=1 \\ j \neq i}}^{n} \frac{1}{1+\left(\frac{F(x_j)}{F(x_i)}\right)^N} \gamma_{ij}, & \text{if } j = i, \end{cases} \tag{10}$$

and is also a stochastic matrix (Menezes, 2010).

The central idea behind the Selective Evolutionary Generation Systems (SEGS) algorithm (Algorithm 1) is to deploy an ergodic selective evolutionary generation system $\Gamma = (X, R, P, G, F)$ with symmetric $\boldsymbol{\gamma}$ (i.e., equiprobable forward and reverse transitions between any pair of genotypes prior to the selection process) so that the Markov chain that represents the resultant stochastic dynamics has a row vector of stationary probabilities , $\boldsymbol{\pi} = \begin{bmatrix} \pi_1 & \pi_2 & \dots & \pi_n \end{bmatrix}$, given by

$$\pi_i = \frac{F(x_i)^N}{\sum\limits_{k=1}^{n} F(x_k)^N}, \ 1 \leq i \leq n. \tag{11}$$

This stationary distribution represents a more general, probabilistic version of the optimization of an objective function. The Markov chain selects the state of maximum fitness with the highest stationary probability, and, in the limit as N approaches ∞, this probability is 1. That is, N tunes the concentration of the stationary probability distribution around the state of maximum fitness, and in the limit as N approaches ∞, the problem and solution then revert to one of standard, off-line optimization. In Menezes and Kabamba (2016), it is also proven that the Markov chain is time-reversible, that the SEGS algorithm is correct, and that increasing N reduces the mean hitting time to the fittest genotype.

Algorithm 1 The SEGS Algorithm

1: choose a level of selectivity
2: initialize the parent genotype
3: **repeat**
4: determine the fitness of the parent genotype
5: choose a resource
6: determine, using the generation function, the offspring genotype that is generated by the parent genotype with the chosen resource
7: determine the fitness of the offspring genotype
8: select one genotype from the parent and offspring genotypes using the *Select* function
9: set the selected genotype as the new parent genotype
10: **until** some stopping criterion applies

2.3. Information Theory and Efficiency

The Markov chain that represents the stochastic dynamics of a selective evolutionary generation system belongs (as proved in Menezes and Kabamba (2016)) to a class of time-homogeneous, irreducible, ergodic Markov chains that are said to *behave rationally* with respect to fitness F with level N. Such chains are characterized by stationary probability row vectors with elements that satisfy

$$\frac{\pi_i}{\pi_j} = \left(\frac{F(x_i)}{F\left(x_j\right)} \right)^N, \ 1 \leq i \leq n, \ 1 \leq j \leq n, \tag{12}$$

which is a definition of *global rationality*, where "rational" again refers to the ratio of the probabilities (again, as in rational numbers, without any implications of agency), and "global" refers to the stationarity of these probabilities.

Menezes and Kabamba (2014, 2016) discuss how Markov chain rational behavior minimizes a cross-entropy function to yield search entropy. The stationary distribution $\boldsymbol{\pi}$ of the ergodic Markov chain that behaves rationally with respect to fitness F with level N is shown to solve the optimization problem

$$\min_{\pi_1,\ldots,\pi_n} U(\boldsymbol{\pi}) = -\sum_{i=1}^{n} \varphi_i \ln(\pi_i), \tag{13}$$

subject to the constraints $\sum_{i=1}^{n} \pi_i = 1$, and $\pi_i > 0, \ \forall i$, utilizing the fitness distribution

$$\varphi_i = \frac{F(x_i)^N}{\sum_{k=1}^{n} F(x_k)^N}, \ 1 \leq i \leq n. \tag{14}$$

This result states that at the optimum, the stationary distribution agrees with the fitness distribution, i.e., $\boldsymbol{\pi} = \boldsymbol{\varphi}$. A corollary is that the time-homogeneous, irreducible, ergodic Markov chain behaves rationally with respect to fitness F with level N if and only if its stationary probability distribution minimizes the "fitness-expectation of information" (the right hand side of (13), with information as defined by Shannon (1948)). At the optimum, this fitness-expectation of information is the entropy of the fitness distribution:

$$U^* = H(\boldsymbol{\varphi}) = -\sum_{i=1}^{n} \varphi_i \ln(\varphi_i). \tag{15}$$

Thereafter, the maximization of this search entropy is investigated, based on results about efficient search from Jaynes (1957, 1981) that specify entropy maximization to eliminate search biases. Such search biases can be induced by, for example, a model that predisposes the optimization process; this predisposition causes inefficient search when the model itself is incorrect as a result of internal or external change.

Maximizing entropy has another interpretation. Equations (13) and (15) can be used to derive (see Menezes and Kabamba (2016)) measures of prior information for a search that were defined in Jaynes (1981). These include a measure of the amount of prior information utilized by the search up to time t, and a measure of the savings in search effort thereby achieved. The former measure can be used to prove correctness of the SEGS algorithm (Menezes and Kabamba, 2016). Both measures can be utilized to demonstrate that 'the optimal [search] policy is... the one that trades off initial information for reduced search effort, as quickly as possible' (Jaynes, 1981), and this policy is proved in Jaynes (1981) to be one of entropy maximization resulting in optimally efficient search.

During model-independent search-based optimization with time-varying objective function or time-varying state fitnesses, an exponential fitness function is proved to relate Markov chain rational behavior, search entropy and optimally efficient search (Menezes and Kabamba, 2014). That is, suppose that $y : X \to \mathbb{R}$ is an unknown function for which an expected value, $\mathrm{E}[y(x)]$, is a known number Y in accordance with Section 2.1. Then a scheme with underlying Markov chain dynamics that behave rationally and a fitness function that is exponential solves the search problem and also maximizes the search entropy while doing so. It is shown in Menezes and Kabamba (2014) that the normalized fitness

$$\varphi_i = \alpha e^{-\beta y(x_i)}, \ 1 \leq i \leq n, \tag{16}$$

(where α and β are constants) and the stationary distribution $\boldsymbol{\pi}$ of the time-homogeneous, irreducible, ergodic Markov chain that behaves rationally with respect to fitness F with level N solves the optimization problem

$$\max_{\varphi_1,\ldots,\varphi_n} \min_{\pi_1,\ldots,\pi_n} U(\boldsymbol{\varphi}, \boldsymbol{\pi}) = -\sum_{i=1}^{n} \varphi_i \ln(\pi_i), \tag{17}$$

subject to the constraint $\mathrm{E}[y(x)] = Y$. The implication is that a fitness function like

$$F(x_i) = e^{-\left((z(x_i)-z_{des})^2\right)} \tag{18}$$

together with a scheme that makes use of Markov chain rational behavior (for instance, the SEGS technique, see Section 2.2) guarantees efficient search-based optimization.

2.4. Markov Chain Rational Behavior and Resilience

Resilience of Markov chains that behave rationally is taken as the sensitivity of the stationary distribution to changes in fitness. Specifically, for any time-homogeneous, irreducible, ergodic Markov chain, the *extrinsic resilience* of state x_i to changes in the fitness of state x_j, $j \neq i$, is

$$\rho_{ij} = \frac{\partial \pi_i}{\partial F(x_j)}, \tag{19}$$

and the *intrinsic resilience* of state x_i to changes in its own fitness is

$$\rho_{ii} = \frac{\partial \pi_i}{\partial F(x_i)}. \tag{20}$$

The Markov chain is resilient if $\rho_{ij} \neq 0$ for all i and j. Since the stationary distribution $\boldsymbol{\pi}$ has the closed form expression (11) for the time-homogeneous, irreducible, ergodic Markov chain that behaves rationally with respect to fitness F with level N, the extrinsic and intrinsic resiliencies are

$$\rho_{ij} = \frac{\partial \pi_i}{\partial F(x_j)} = \frac{-N\pi_i\pi_j}{F(x_j)}, \ \forall j \neq i, \tag{21}$$

$$\rho_{ii} = \frac{\partial \pi_i}{\partial F(x_i)} = \frac{N\pi_i(1-\pi_i)}{F(x_i)}. \tag{22}$$

These equations for resilience are used in Menezes and Kabamba (2016) to prove that the level of selectivity has the following asymptotic effect:

$$\rho_{ij}\Big|_{\substack{N=0 \\ j\neq i}} = \rho_{ii}\Big|_{N=0} = 0, \tag{23}$$

$$\lim_{\substack{N\to\infty \\ j\neq i}} \rho_{ij} = \lim_{N\to\infty} \rho_{ii} = 0. \tag{24}$$

That is, standard, off-line optimization ($N \to \infty$) is non-resilient. Purely random optimization ($N = 0$) is also non-resilient. Because the expected hitting time of the genotype that optimizes fitness also decreases with an increasing level of selectivity N, a trade-off exists between this expected hitting time and resilience, with the trade-off controlled by N.

Menezes and Kabamba (2016) goes on to prove that Markov chain rational behavior is a sufficient condition for resilience, while ergodicity is a necessary condition for resilience. In addition, Menezes and Kabamba (2016) also provides four equations to analyze the effect of changes in genotype fitness on elements of the matrix of transition probabilities, $\mathbf{P}$. These equations demonstrate that, unlike gradient ascent optimization where the transition to another genotype would be directly proportional to the fitness value, optimization with Markov chain rational behavior is reminiscent of the retardation property in the original rational behavior (Meerkov, 1979); the stochastic process "slows down" transitions in more favorable fitness conditions to take advantage of the external environment.

2.5. Relationship to Other Evolutionary Search Algorithms

Of the many machine learning strategies that the SEGS algorithm is compared to in Menezes (2010), two techniques are found to be related: the (1+1)-Evolution Strategy (ES) (Beyer and Schwefel, 2002) and the Canonical Genetic Algorithm with Fitness Proportional Selection (CGAFPS) (Rudolph, 1994). These algorithms are shown (Menezes, 2010) to be particular cases of the SEGS algorithm by comparing (5) to each method's ratio of the probability of selecting a candidate genotype for objective function optimization to the probability of selecting the genotype's offspring. A summary of the results follows.

For the (1+1)-ES strategy, one genotype, x_1, produces one mutated offspring genotype, x_2, and the ratio of the probability of selecting x_1 to the probability of selecting x_2 is simply

$$\frac{\Pr[x_1 \text{ is selected}]}{\Pr[x_2 \text{ is selected}]} = \frac{\mathrm{ind}\left(F(x_1) > F(x_2)\right)}{\mathrm{ind}\left(F(x_2) \geq F(x_1)\right)}, \tag{25}$$

where ind denotes the indicator function, satisfying $\mathrm{ind}(True) = 1$ and $\mathrm{ind}(False) = 0$. The ratio in (25) is taken to be ∞ if the denominator is zero. This ratio equals (5) when the parameter N in (5) approaches ∞.

For the CGAFPS, we are interested in the probability that a genotype, x_1, of the population is chosen to be a member of the population for the next generation without experiencing crossover or mutation. We then compare this probability to the probability that an offspring of x_1 is a member of the population at the next generation. Per Menezes (2010), the ratio of the probability of selecting x_1 to the probability of selecting x_2 has form

$$\frac{\Pr[x_1 \text{ is selected}]}{\Pr[x_2 \text{ is selected}]} = K\frac{F(x_1)}{F(x_2)},\ K > 0. \tag{26}$$

Although the equation above is similar to (5), the CGAFPS ratio of selection probabilities is proportional to the fitness ratio. In (26), if $K = 1$ we obtain a particular case of (5) where $N = 1$.

3. MAV Flapping Wing Gait Estimation for Hover

The preceding theory indicates that the SEGS algorithm is suitable for on-line control whereby MAV flapping wing gaits are efficiently and resiliently evolved in changing flight conditions. Simulation results of flapping wing MAV hovering flight are provided in the following subsections to verify that the SEGS technique optimizes flapping wing gaits, is responsive, has benchmark performance that corroborates the theory, and generalizes the two well-known evolutionary search methods as previously described.

3.1. Applying the SEGS Algorithm

To implement Algorithm 1 to ensure flapping wing MAV hover, we first determine the selective evolutionary generation system, i.e., its range of genotypes, generation function, etc. We begin by identifying a means of evaluating flapping wing gait fitness for this selective evolutionary generation system such that the evaluation is external to the SEGS method and also accurately represents the airflows that are induced by the flapping wing gaits that are selected by the technique. An experimental implementation option for this fitness evaluation could involve the reading from an altitude sensor, but another on-line option that is suited to simulation tests of the SEGS algorithm is that of an aerodynamic surrogate model, which is an alternative to computationally expensive flow simulations for each flapping wing gait. In general, 'surrogate models replace costly objective function evaluations with inexpensive approximations of sufficient fidelity' (Trizila et al., 2008). Such models can be developed (Queipo et al., 2005; Forrester and Keane, 2009) using a variety of techniques. An aerodynamic surrogate model, when trained on the computational results of a suitable number of design flapping wing gaits, can quickly provide information about the aerodynamics of off-design gaits. Thus, a surrogate model of aerodynamic flow that is external to the SEGS optimization is employed here to simulate flapping wing MAV flight performance, acting as a "sensor" that provides phenotype feedback to the SEGS technique. Lim et al. (2010), Olhofer et al. (2011), and Jin (2011) validate such use of a surrogate model, and also the use in the original Menezes (2010) that preceded these works. The external surrogate model can be substituted in SEGS algorithm hardware implementations with information from flapping wing MAV sensors (e.g., from an altimeter).

Flapping wing aerodynamic surrogate models developed by Trizila et al. (2008) are readily available and have been analyzed extensively. In what follows, an ensemble surrogate model that is a weighted average of a surrogate model developed using kriging and another surrogate model developed using support vector regression with a linear spline kernel is employed. This ensemble surrogate model first determines an average coefficient of lift, C_L, for each flapping wing stroke profile that is produced by flapping wing gait parameters, and the computed C_L is then provided back to the SEGS algorithm. The reason for the focus on non-dimensional C_Ls instead of cycle-averaged dimensionalized lift force, L, values can be attributed to the lift expression

$$L = \frac{1}{2}C_L\rho U^2 S \tag{27}$$

(where ρ is the local air density, U is the forward flight velocity, and S is the wing planform area), which implies that maintenance of a fixed L that is chosen by supervisory control because of the current mission scenario (e.g., hover, ascent, or descent modes) requires that disturbances like wind gusts that affect U be countered by manipulating C_L through flapping wing gait changes.

The prescribed flapping motion (Fig. 1) that is utilized by the ensemble surrogate model is

$$h(t) = h_a(t)\sin(\omega t), \tag{28}$$

$$\alpha(t) = 90 - \alpha_a(t)\sin\left(\omega t + \phi_\alpha\left(t\right)\right), \tag{29}$$

where $h_a(t) \in [1, 2]$ and $\alpha_a(t) \in [45, 80]$ are the piecewise-constant amplitudes of flapping stroke height and pitch, respectively, ω is a frequency that depends on h_a and a constant Reynolds number of 100, and $\phi_\alpha(t) \in [60, 120]$ is the piecewise-constant phase shift angle for flapping pitch. Hence, the hovering flapping flight problem: given a time history of the target lift coefficient, $C_{L_{des}}(t)$, determine suitable time-varying flapping wing kinematic parameters that meet the target. (In the experimental version

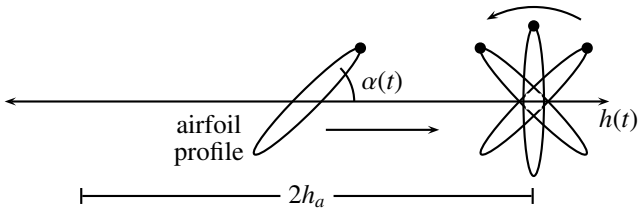


Figure 1: Schematic representation of a flapping wing forward half-stroke, where the dot represents the leading edge of a 15% elliptical airfoil. The flapping wing airfoil moves forward in an aircraft body-fixed reference frame through twice the stroke height h_a, and is pictured here at the midpoint of the forward half-stroke. Upon reaching its most forward position, the airfoil turns around for the back half-stroke.

of this problem, $C_{L_{des}}(t)$ and the computed C_L can be replaced, for instance, by time-varying desired altitude and the current altitude, respectively.)

We craft the following selective evolutionary generation system to solve this problem. The set of genotypes, X, is the set of ordered triples $(h_a(t), \alpha_a(t), \phi_\alpha(t))$, where

$$h_a(t) \in \{1, 1.1, 1.2, \ldots, 1.9, 2\}, \tag{30}$$

$$\alpha_a(t) \in \{45, 46, 47, \ldots, 79, 80\}, \tag{31}$$

$$\phi_\alpha(t) \in \{60, 61, 62, \ldots, 119, 120\}. \tag{32}$$

The set of resources, R, is the set $\{r_1, r_2, r_3, r_4, r_5, r_6\}$, with $r_i = \mathbf{e}_i,\ 1 \le i \le 6$ (where $\mathbf{e}_i$ are the standard basis vectors for $\mathbb{R}^6$). This set facilitates the perturbation of one of the three elements of a genotype in either a positive or negative direction when an offspring is generated with G. The probability mass function on R, P, is the discrete uniform distribution. This choice of probability mass function ensures that the matrix $\boldsymbol{\gamma}$ is symmetric. The generation function, G, when applied to $x \in X$ using resource $r \in R$, yields $G((h_a(t), \alpha_a(t), \phi_\alpha(t)), r_i)$, $1 \le i \le 6$, the triple given by

$$\begin{cases} \begin{bmatrix} 0.1 & -0.1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix} r_i + \begin{bmatrix} h_a(t) \\ \alpha_a(t) \\ \phi_\alpha(t) \end{bmatrix}, & \text{if } 1 < h_a(t) < 2,\ 45 < \alpha_a(t) < 80,\ 60 < \phi_\alpha(t) < 120, \\ (h_a(t), \alpha_a(t), \phi_\alpha(t)), & \text{otherwise.} \end{cases} \tag{33}$$

The above is a modified version of a random walk over a discretized search space, where the modification involves selection dynamics described by the *Select* function to produce a selective evolutionary generation system. Since the objective is for $C_L(t)$ to track $C_{L_{des}}(t)$, and we wish to do so efficiently, we choose to use the fitness function

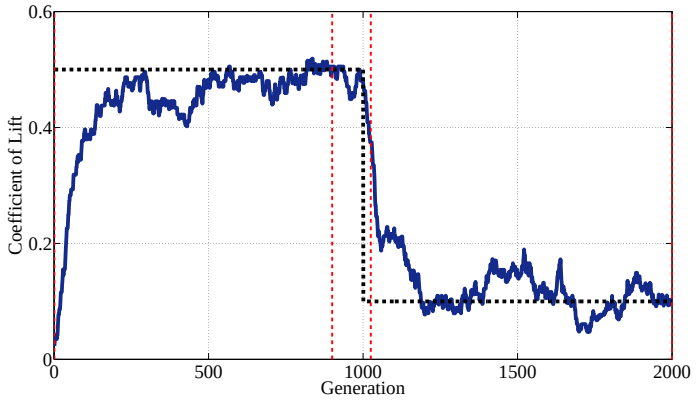
$$F(h_a(t), \alpha_a(t), \phi_\alpha(t)) = \exp\left(-\left(K_f\left(C_{L_{des}}(t) - C_L(t)\right)\right)^2\right), \tag{34}$$

where $K_f = 10$, and $C_L(t) = C_L(h_a(t), \alpha_a(t), \phi_\alpha(t))$ is the output of the surrogate model. Again, the fitness function can be modified to use sensor readings while implemented in hardware.

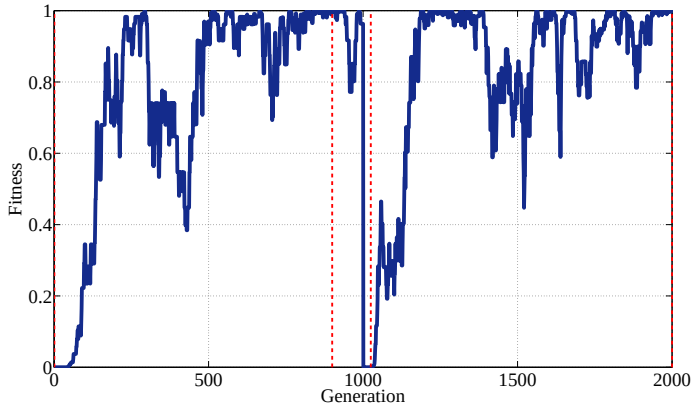
A sample run of the evolution scheme when $N = 5$ is depicted in Figs. 2 to 3. A genotype triple that achieves satisfactory performance is found within 1000 generations, and the scheme is resilient because it quickly finds a new triple that achieves an acceptable output when the target lift coefficient, and hence the fitness function, changes. In Fig. 2, the red vertical dashed lines indicate a generation for which the evolved flapping forward and backward motion is illustrated in Fig. 3.

For generations 1, 900, 1025, and 2000, the plots in Fig. 3 each display 10 snapshots of a 15% elliptical airfoil through a flapping half-stroke. The dot represents the leading edge of the airfoil, which moves in an aircraft body-fixed reference frame with neutral position at (0,0). The arrows on the forward half-stroke plots indicate that the airfoil travels from the most rearward position to the most forward position, whereas the opposite is true for a back half-stroke. Although the periods of the strokes vary at different generations because of the constant Reynolds number, the snapshots are taken at the same fractional period interval. Hence, a stroke with more spacing between snapshots has a faster motion than a stroke with snapshots that are closely spaced.

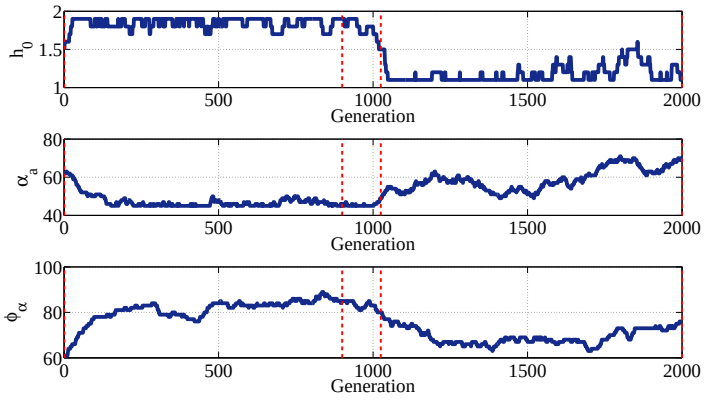
The scheme averages 1 min 18 s to compute the output of 1000 generations in Matlab on a 2.50 GHz dual-core processor laptop with 4.00 GB of RAM and the Windows Vista operating system. This time includes generational surrogate model computation time. If the generational surrogate model computation time is not included, the scheme averages 0.13 ms to complete a generation, measured over 2000 generations.



(a) Target (dashed) and actual (solid) lift coefficients that were found in a sample run of the evolution scheme. At each generation, the actual lift coefficient is the output of a surrogate model for an input of a flapping wing kinematic parameter triple that is determined by the scheme.

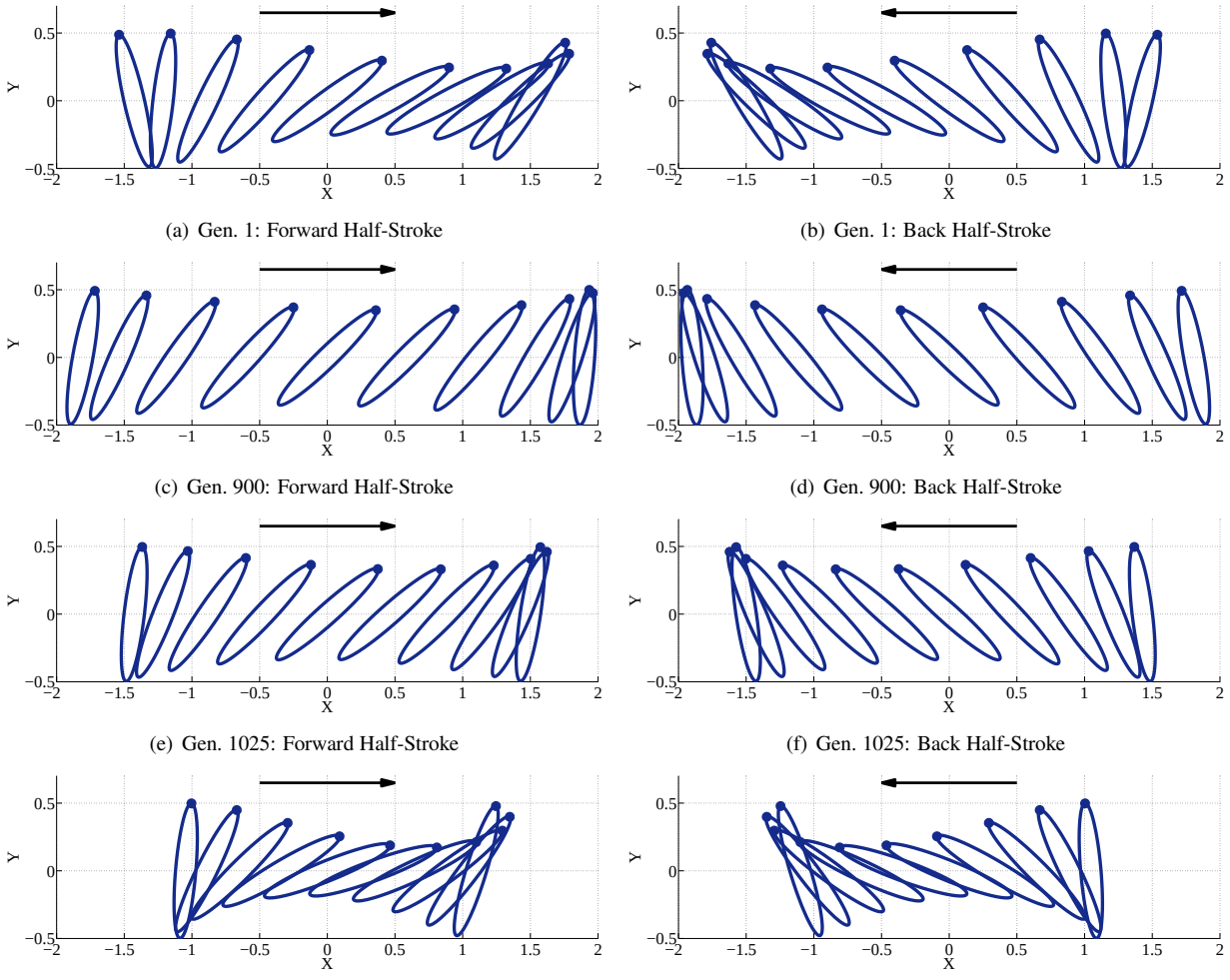


(b) Fitness values of the lift coefficients that correspond to the flapping wing kinematic parameter triples determined by the scheme. Since the fitness value is highest when the target and actual lift coefficients match, a change in the target lift coefficient at generation 1001 causes a significant reduction in a previously high fitness value.



(c) Flapping wing kinematic parameters evolved by the scheme. The converged solutions at generations 1000 and 2000 differ from each other, but both have high fitness and achieve the target lift coefficient at that generation.

Figure 2: Sample resilient selective evolutionary generation system results with a level of selectivity N value of 5. When the target lift coefficient is changed at generation 1001, which nullifies the fitness of the scheme's previously converged-upon flapping wing kinematic parameters, the scheme displays resilience by recovering to find a new triple of suitable kinematic parameters. Illustrations of the flapping wing forward and back half-strokes for the generations indicated with red vertical dashed lines are in Fig. 3.



(a) Gen. 1: Forward Half-Stroke

(b) Gen. 1: Back Half-Stroke

(c) Gen. 900: Forward Half-Stroke

(d) Gen. 900: Back Half-Stroke

(e) Gen. 1025: Forward Half-Stroke

(f) Gen. 1025: Back Half-Stroke

(g) Gen. 2000: Forward Half-Stroke

(h) Gen. 2000: Back Half-Stroke

Figure 3: Snapshots of the forward and back half-strokes of the flapping wing sampled at the 1st, 900th, 1025th, and 2000th generations of the run of the evolution scheme depicted in Fig. 2. The 1st and 1025th generations are at the start of the scheme's determinations of suitable flapping wing kinematic parameters that meet a target lift coefficient, while the 900th and 2000th generations are when the actual lift coefficient is satisfactory. Because the snapshots in this figure are taken at the same fractional period interval, the stroke at the 900th generation is faster than the stroke at the 2000th generation, since it has a greater spacing between snapshots. The stroke at the 900th generation is also comparatively longer. It is this rapid and expansive motion at the 900th generation that achieves the higher desired target lift coefficient of 0.5, in contrast to the slower and shorter stroke at the 2000th generation that achieves the lower desired target lift coefficient of 0.1.

3.2. Benchmark Results of the SEGS Algorithm

3.2.1. Level of Selectivity Effects

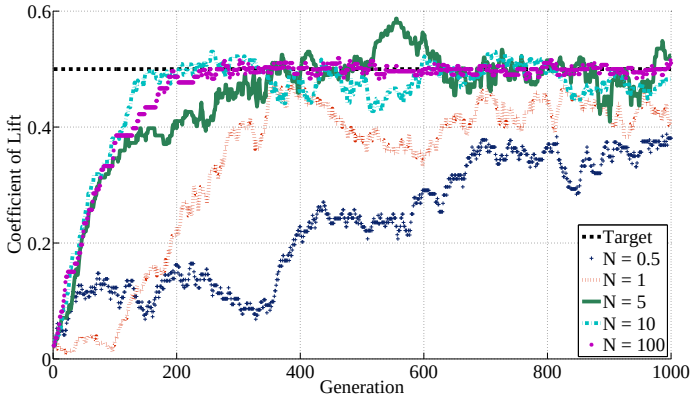
Sample runs of the selective evolutionary generation system for various levels of selectivity are plotted in Fig. 4. For this comparison, the algorithm was initialized to the triple $(h_a(0), \alpha_a(0), \phi_\alpha(0)) = (1.5, 62, 60)$, and the target lift coefficient was held constant at $C_L = 0.5$. These figures illustrate the rationale for choosing $N = 5$ in Section 3.1. At low levels of selectivity ($N = 0.5$ and $N = 1$), the algorithm wanders through the search space and does not reach the target lift coefficient within a user-specified limit of 1000 generations. Increases in the level of selectivity cause a corresponding improvement in target lift coefficient tracking. The $N = 5$ trajectory depicts excursions away from the desired lift coefficient; these excursions are minimized at the slightly higher level of selectivity, $N = 10$. The $N = 100$ trajectory achieves near perfect lift coefficient tracking with few excursions. A suitable choice of the level of selectivity that tolerates excursions is therefore either $N = 5$ or $N = 10$, since excursions are one indicator of resilience. Another indicator of resilience is the initial behavior of the $N = 5$ and $N = 10$ trajectories; however, these two trajectories are approximately equal during the first 50 generations. Since the $N = 5$ trajectory achieves tracking and greater resilience than the $N = 10$ trajectory, we choose the level of selectivity $N = 5$.

The trade-off between optimality and resilience is documented in Fig. 5. The figure shows simulations where the target lift coefficient varies frequently during 1000 generations, and the target includes a $C_L = 0.7$ value that is beyond the flapping wing capabilities that the surrogate model simulates. It is clear that the $N = 5$ trajectory displays a more immediate response to the change in target than the $N = 100$ trajectory. This response is also evident in the initial higher fitness values of the $N = 5$ trajectory in Fig. 5(b). However, the more selective $N = 100$ trajectory overtakes the $N = 5$ trajectory after a short period of time, in accordance with the shorter convergence times and optimality properties of high levels of selectivity. Both trajectories handle an unattainable target similarly.

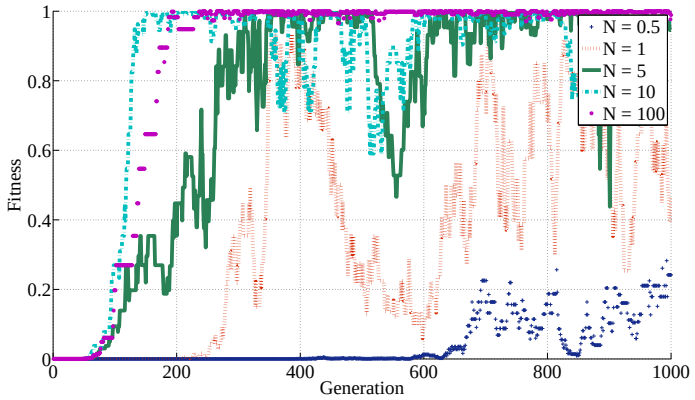
The effect of the level of selectivity on one possible stopping criterion is outlined in Table 1. The table lists the average number of generations required to find a flapping wing gait with a lift coefficient that is within ±3% of the target value. This tolerance corresponds to a fitness value that is at least 0.975 or greater. As expected, an increase in N decreases the number of generations to find a "good" solution.

Table 1: Level of Selectivity Effects on a Stopping Criterion

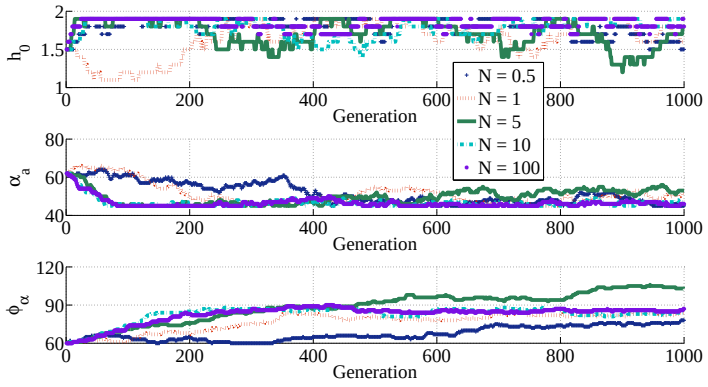
N	Initial Conditions	Target C_L	Avg. No. of Generations
1	(1.5,62,60)	0.5	1218
5	(1.5,62,60)	0.5	399
10	(1.5,62,60)	0.5	246
100	(1.5,62,60)	0.5	191



(a) Lift coefficients that were found in sample runs of the evolution scheme at different levels of selectivity, N. The scheme wanders through the search space when N is low and achieves near perfect target tracking with few excursions when N is high.

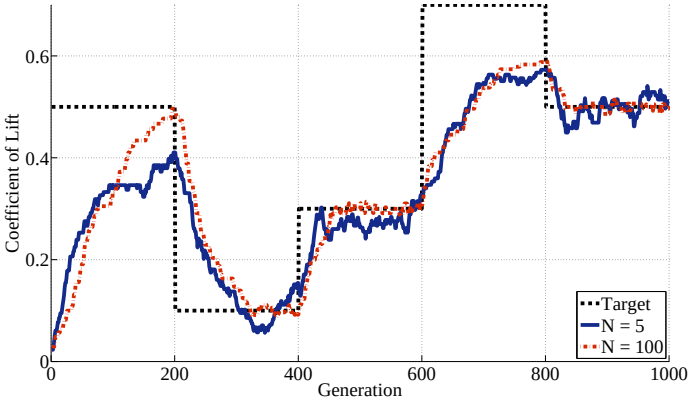


(b) Fitness values of the lift coefficients that correspond to the flapping wing kinematic parameter triples determined by the scheme during the sample runs at different levels of selectivity, N.

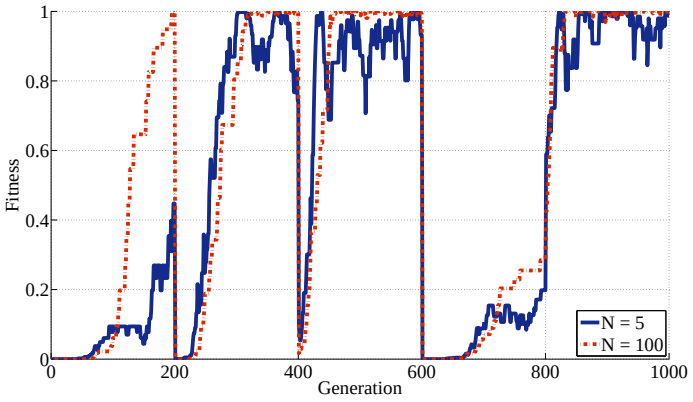


(c) Flapping wing kinematic parameters evolved by the scheme during the sample runs at different levels of selectivity, N. At high N (e.g., here $N = 10$ and $N = 100$), a similar evolutionary process and converged solution result.

Figure 4: Sample level of selectivity N effects in a selective evolutionary generation system. The $N = 5$ trajectory achieves tracking and also tolerates excursions from the target, which is an indicator of resilience.



(a) Lift coefficients that were found in sample runs of the evolution scheme at two different levels of selectivity, N, while the target setpoint varies (including at an unattainable value of 0.7). After every target change, the lower selectivity $N = 5$ trajectory is quicker to initially respond compared to the higher selectivity $N = 100$ trajectory. If the target stabilizes after a change, the $N = 100$ trajectory overtakes the $N = 5$ trajectory to more quickly attain and better track an optimal solution, in accordance with the theory.



(b) Fitness values of the lift coefficients that correspond to the flapping wing kinematic parameter triples determined by the scheme during the sample runs at two different levels of selectivity, N. The $N = 5$ trajectory clearly demonstrates greater resilience than the $N = 100$ trajectory with consistently higher fitness values in the immediate aftermath of a target change.

Figure 5: Sample illustrative trade-off between optimality and resilience in a selective evolutionary generation system. For highly fluctuating environments, a lower level of selectivity N permits a greater resilience to fluctuations, whereas in less frequently perturbed environments, a larger N ensures shorter times for attaining an optimal solution.

3.2.2. Initialization Effects

The initial conditions of a SEGS algorithm affect performance. Consider Table 2(a), which displays the effect of various initializations on the average number of generations required to find a flapping wing gait with a lift coefficient that is within ±3% of the target. As the table indicates, there is significant disparity in the average number of generations that is required before the stopping criterion is reached. Hence, initial conditions do play a role in the convergence of the SEGS algorithm. Moreover, the number of fit solutions in the search space also affect convergence. The surrogate model employed by the selective evolutionary generation system was trained on 24 samples of two dimensional computational fluid dynamics data, of which four samples had a lift coefficient of approximately 0.5 (the most number of samples for a given lift coefficient), and only one sample had a lift coefficient above 0.6 (Trizila et al., 2008). Accordingly, the effect of the initializations in Table 2(a) on the average number of generations required to find a flapping wing gait with lift coefficient within ±3% of 0.62 is tabulated in Table 2(b). It is clear that a significantly greater average number of generations is required when there are fewer fit solutions in the search space.

Table 2: Initialization Effects on a Stopping Criterion

(a) Many Fit Solutions

N	Initial Conditions	Target C_L	Avg. No. of Generations
5	(1.5,62,60)	0.5	399
5	(1.5,45,60)	0.5	370
5	(1.5,62,90)	0.5	79
5	(1,45,60)	0.5	368
5	(1,45,120)	0.5	1
5	(1,80,60)	0.5	466
5	(1,80,120)	0.5	120
5	(2,45,60)	0.5	390
5	(2,45,120)	0.5	94
5	(2,80,60)	0.5	328
5	(2,80,120)	0.5	120

(b) Few Fit Solutions

N	Initial Conditions	Target C_L	Avg. No. of Generations
5	(1.5,62,60)	0.62	1171
5	(1.5,45,60)	0.62	967
5	(1.5,62,90)	0.62	468
5	(1,45,60)	0.62	1065
5	(1,45,120)	0.62	80
5	(1,80,60)	0.62	1280
5	(1,80,120)	0.62	150
5	(2,45,60)	0.62	995
5	(2,45,120)	0.62	1
5	(2,80,60)	0.62	820
5	(2,80,120)	0.62	161

3.2.3. Discretization Effects

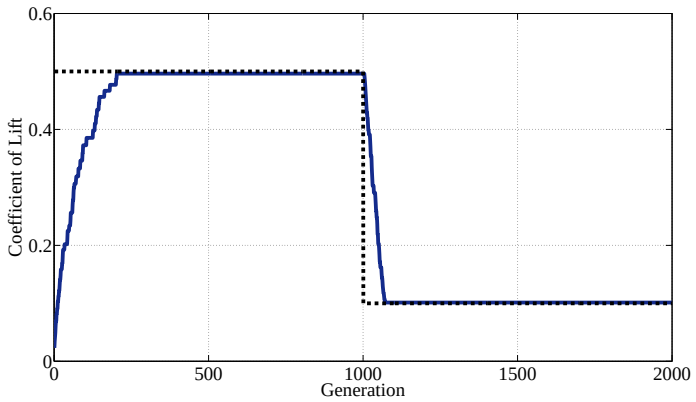
The selective evolutionary generation system in Section 3.1 discretizes the search space into h_a step sizes of 0.1, and into α_a and ϕ_α step sizes of 1 degree. The type of discretization employed by the SEGS algorithm affects the average number of generations required to find a flapping wing gait subject to the stopping criterion previously outlined. Table 3 provides the details for possible discretizations, with $N = 5$, initial conditions $(h_a(0), \alpha_a(0), \phi_\alpha(0)) = (1.5, 62, 60)$, and target lift coefficient 0.5. The table hints at the prospect of an optimal discretization of the search space that minimizes the average number of generations required to find a fit flapping wing gait. However, care must be taken to not use too coarse a discretization in the quest for reduced computation, since such a discretization may omit subtle features of the search space. It is expected that the optimal discretization be application dependent.

Table 3: Discretization Effects on a Stopping Criterion

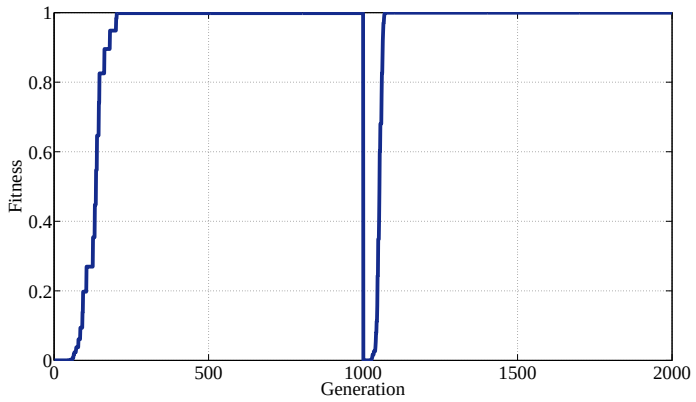
Discretization Type	h_a Step	α_a Step	ϕ_α Step	Avg. No. of Generations
Fine	0.05	0.5	0.5	918
Baseline	0.1	1	1	399
Coarse	0.2	5	5	49
Very Coarse	0.5	10	10	58

3.3. Comparison to Related Evolutionary Search Algorithms

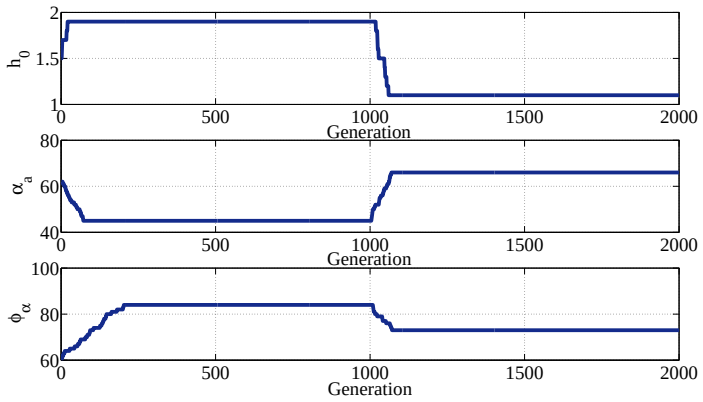
A sample run of the (1+1)-ES is depicted in Fig. 6. A genotype triple that achieves satisfactory performance is found within 1000 generations. Identical to the average computation time for the SEGS algorithm, the scheme averages 1 min 18 s to compute the output of 1000 generations in Matlab on a 2.50 GHz dual-core processor laptop with 4.00 GB of RAM and the Windows Vista operating system. Again, this time includes generational surrogate model computation time. However, the scheme is faster than the SEGS algorithm on a per generation basis when not including the generational surrogate model computation time, as it averages 0.073 ms measured over 2000 generations.



(a) Target (dashed) and actual (solid) lift coefficients that were found in a sample run of the (1+1)-ES.



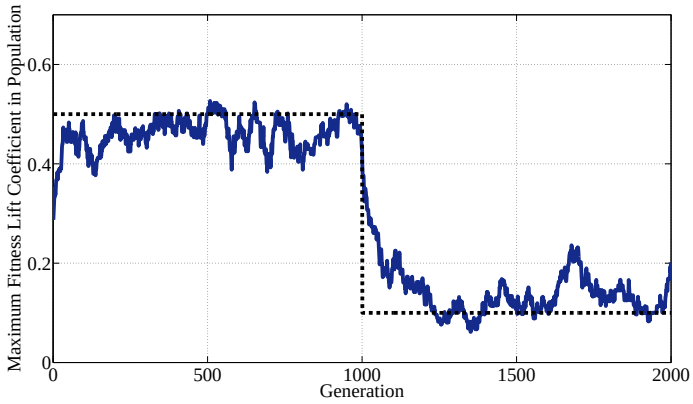
(b) Fitness values of the lift coefficients that correspond to the flapping wing kinematic parameter triples determined by the (1+1)-ES.



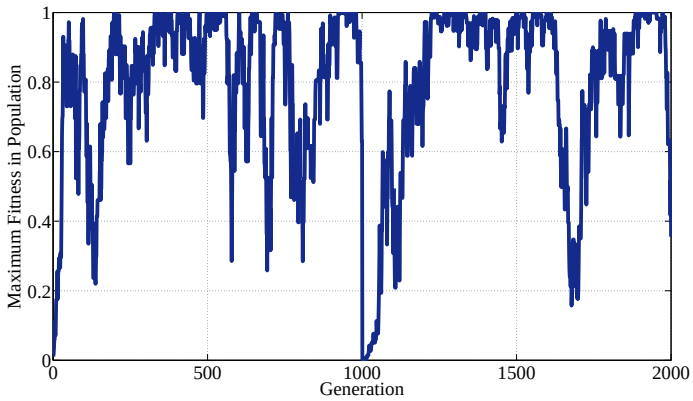
(c) Flapping wing kinematic parameters evolved by the (1+1)-ES.

Figure 6: Sample (1+1)-ES results. Characteristically, there are no excursions while finding or tracking the target.

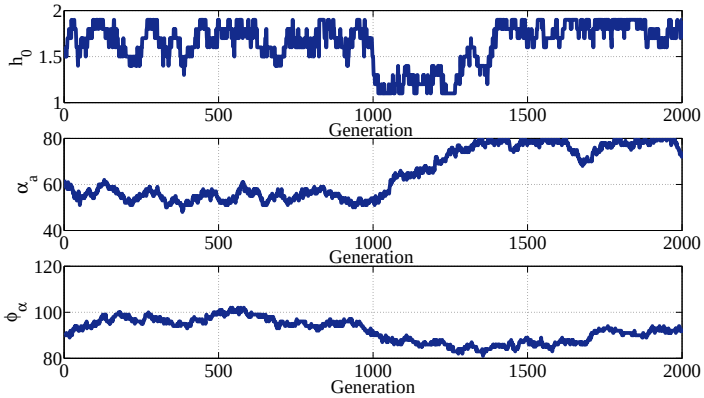
A sample run of the CGAFPS is depicted in Fig. 7. The algorithm was initialized with a population of four triples: $(1.5, 62, 60)$, $(1.0, 45, 60)$, $(2.0, 80, 120)$ and $(1.5, 62, 90)$. A genotype triple that achieves satisfactory performance is found within 1000 generations. The scheme averages 1 min 47 s to compute the output of 1000 generations in Matlab on a 2.50 GHz dual-core processor laptop with 4.00 GB of RAM and the Windows Vista operating system. This time includes generational surrogate model computation time, and is longer than the average computation time for the SEGS algorithm. On a per generation basis when not including the generational surrogate model computation time, this scheme is the slowest at an average of 0.21 ms measured over 2000 generations.



(a) Target (dashed) and actual (solid) lift coefficients of maximum fitness in a population that was initialized at four, as found in a sample run of the CGAFPS.



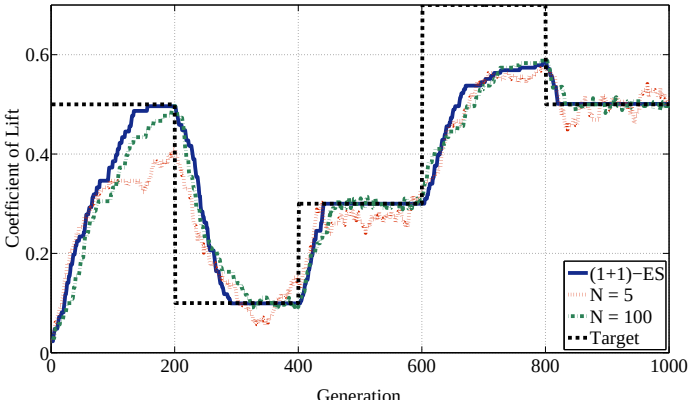
(b) Maximum fitness values of the lift coefficients in a population that was initialized at four, corresponding to the flapping wing kinematic parameter triples determined by the CGAFPS.



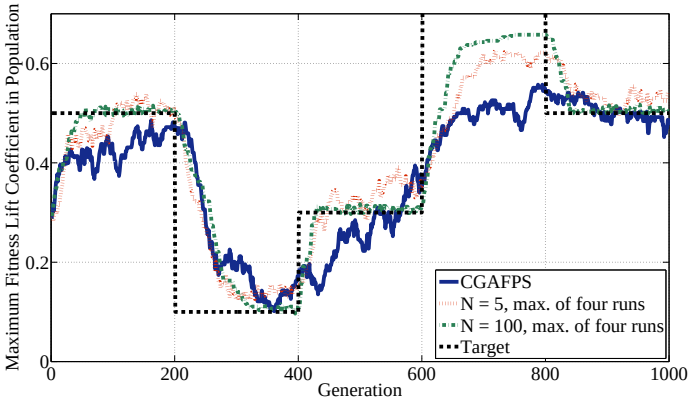
(c) Flapping wing kinematic parameters of the maximum-fitness lift coefficient in a population that was initialized at four, as evolved by the CGAFPS.

Figure 7: Sample CGAFPS results in the population. Unlike the (1+1)-ES behavior in Fig. 6, numerous excursions while finding and tracking the target occur.

Fig. 8 compares the resilience of these two algorithms to the SEGS technique. The (1+1)-ES behaves like a selective evolutionary generation system with an N that exceeds 100. The CGAFPS exhibits resilience, which is unsurprising since it is similar to a SEGS scheme with $N = 1$.



(a) Lift coefficients that were found in a sample run of the (1+1)-ES and in sample runs of the selective evolutionary generation scheme at two different levels of selectivity, N, while the target setpoint varies (including at an unattainable value of 0.7). After every target change, the (1+1)-ES trajectory is slower to respond than the $N = 5$ and $N = 100$ trajectories, but the former trajectory overtakes the latter two to more quickly attain and better track an optimal solution.



(b) Maximum-fitness lift coefficients that were found in a sample CGAFPS population that was initialized at four and that were found in four sample runs of the selective evolutionary generation scheme, where each was initialized to one member of the initial CGAFPS population. The latter was repeated for a second level of selectivity, N. All trajectories have a target setpoint that varies (including at an unattainable value of 0.7). After every target change, the CGAFPS trajectory is quicker to respond than the $N = 5$ and $N = 100$ trajectories, but the former trajectory has more excursions and poorer tracking of an optimal solution.

Figure 8: A visual comparison of the (1+1)-ES and the CGAFPS to the SEGS algorithm showing that the (1+1)-ES behaves like a selective evolutionary generation system with an N that exceeds 100, while the CGAFPS behaves like a selective evolutionary generation system with an N that is smaller than 5.

4. MAV Flapping Wing Gait Estimation for Trajectory Control

Model-independence of the SEGS technique implies that the external surrogate model of Section 3 can be easily replaced by an external high-fidelity model of the complex and realistic airflows caused by flapping wing gait genotypes, with the "sensed" fitness feedback now evaluated on a different phenotype, e.g., MAV forward-flight trajectory. As previously stated, it is also possible to leverage sensor readings as an alternate during hardware forward-flight. In this section, the produced trajectory solution is evolved in "real-time" as the MAV flies to demonstrate technique practicality.

The Theodorsen-Garrick model (Theodorsen, 1935; Garrick, 1936) predicts the lift and thrust forces on a flat plate under-

going a prescribed flapping motion with various input kinematic parameters. This flapping motion is described by

$$h(t) = h_a(t) \sin\left(\omega\left(t\right) t + \phi_h\left(t\right)\right), \tag{35}$$

$$\alpha(t) = \alpha_a(t) \sin\left(\omega\left(t\right) t + \phi_\alpha\left(t\right)\right), \tag{36}$$

where $h_a(t) \in (0, 1]$ and $\alpha_a(t) \in [-0.5, 0.5]$ are the piecewise-constant amplitudes of flapping stroke height and angle of attack respectively, $\omega(t) \in (0, 1]$ is a piecewise-constant frequency, and $\phi_h(t) \in [-0.5, 0.5]$ and $\phi_\alpha(t) \in [-0.5, 0.5]$ are the piecewise-constant phase shift angles for flapping stroke height and angle of attack, respectively. The flapping motion described in (35)–(36) leads to the computation of lift and thrust forces through the equations stated in Garrick (1936). These forces determine the trajectory followed by the flapping wing; hence, the flapping flight motion problem: given a target trajectory (e.g., a constant altitude forward motion trajectory), find suitable flapping wing kinematic parameters that meet the target.

We utilize the following selective evolutionary generation system. The set of genotypes, X, is the set of ordered pentuples $\left(h_a\left(t\right), \omega\left(t\right), \phi_h\left(t\right), \alpha_a\left(t\right), \phi_\alpha\left(t\right)\right)$, where

$$h_a\left(t\right) \in \{0.1, 0.2, 0.3, \ldots, 0.9, 1\}, \tag{37}$$

$$\omega\left(t\right) \in \{0.05, 0.1, 0.15, \ldots, 0.95, 1\}, \tag{38}$$

$$\phi_h\left(t\right) \in \{-0.5, -0.45, -0.4, \ldots, 0.45, 0.5\}, \tag{39}$$

$$\alpha_a\left(t\right) \in \{-0.5, -0.45, -0.4, \ldots, 0.45, 0.5\}, \tag{40}$$

$$\phi_\alpha\left(t\right) \in \{-0.5, -0.45, -0.4, \ldots, 0.45, 0.5\}. \tag{41}$$

The set of resources, R, is the set $\{r_1, r_2, r_3, r_4, r_5, r_6, r_7, r_8, r_9, r_{10}\}$, with $r_i = \mathbf{e}_i$, $1 \leq i \leq 10$ (where $\mathbf{e}_i$ are the standard basis vectors for $\mathbb{R}^{10}$). This choice of resources facilitates the perturbation of one of the elements of a genotype in either a positive or negative direction when an offspring is generated. Again, the probability mass function on R, P, is the discrete uniform distribution. As before, this choice of probability mass function ensures that the matrix $\boldsymbol{\gamma}$ is symmetric. The generation function, G, applied to X as $G\left(\left(h_a\left(t\right), \omega\left(t\right), \phi_h\left(t\right), \alpha_a\left(t\right), \phi_\alpha\left(t\right)\right), r_i\right)$, $1 \leq i \leq 10$, is the pentuple given by

$$\begin{cases} 1 \times 10^{-2} A r_i + \begin{bmatrix} h_a\left(t\right) \\ \omega\left(t\right) \\ \phi_h\left(t\right) \\ \alpha_a\left(t\right) \\ \phi_\alpha\left(t\right) \end{bmatrix}, \text{ where } A = \begin{bmatrix} 10 & -10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 5 & -5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 5 & -5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 5 & -5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 5 & -5 \end{bmatrix}, \\ \text{if } 0.1 < h_a\left(t\right) < 1,\ 0.05 < \omega\left(t\right) < 1,\ -0.5 < \phi_h\left(t\right) < 0.5,\ -0.5 < \alpha_a\left(t\right) < 0.5,\ -0.5 < \phi_\alpha\left(t\right) < 0.5, \\ \left(h_a\left(t\right), \omega\left(t\right), \phi_h\left(t\right), \alpha_a\left(t\right), \phi_\alpha\left(t\right)\right), \text{ otherwise.} \end{cases} \tag{42}$$

The above is again a modified version of a random walk over a discretized search space, where the modification involves selection dynamics described by the *Select* function to produce a selective evolutionary generation system. The flapping wing parameters evolved by the SEGS algorithm are inputs for the Theodorsen-Garrick model, which outputs lift $L(\tau)$ and time-averaged-thrust $T(\tau)$ over time τ. These forces are in turn inputs for the following double-integrator, unit-mass wing trajectory dynamics,

$$\begin{bmatrix} \dot{x}(\tau) \\ \dot{y}(\tau) \\ \dot{v}_x(\tau) \\ \dot{v}_y(\tau) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x(\tau) \\ y(\tau) \\ v_x(\tau) \\ v_y(\tau) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} L(\tau) \\ T(\tau) \end{bmatrix}, \tag{43}$$

where $\left(x\left(\tau\right), y\left(\tau\right)\right)$ is the trajectory of the center of mass of the flapping wing. This trajectory is sampled ν times, yielding $\left(x\left(k\right), y\left(k\right)\right)$, $1 \leq k \leq \nu$. For each $x(k)$, the target $y_{des}(k)$ is computed. Let

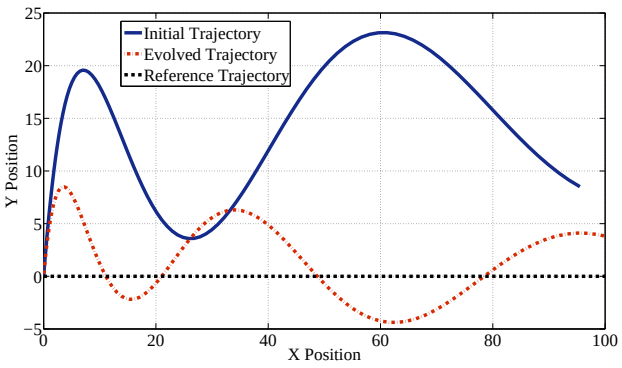
$$AvgDistance(t) = \frac{\sum_{k=1}^{\nu} \left| y_{des}(k) - y(k) \right|}{\nu} \tag{44}$$

be the mean difference between the target and current trajectories. Since the objective is to track the target, we use the following fitness function for the SEGS algorithm,

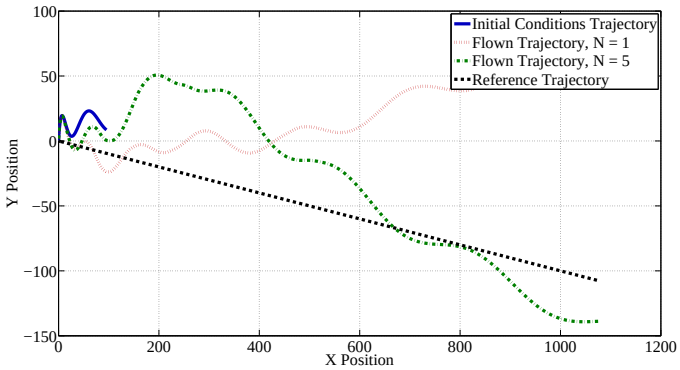
$$F\left(h_a\left(t\right), \omega\left(t\right), \phi_h\left(t\right), \alpha_a\left(t\right), \phi_\alpha\left(t\right)\right) = \exp\left(-\left(0.1 AvgDistance(t)\right)^2\right). \tag{45}$$

To check SEGS controller performance, a sample initial trajectory and an evolved trajectory after 200 generations with $N = 5$ are plotted in Fig. 9(a), where the trajectories are depicted over the same period of time and the vehicle state (position and

velocity) is not updated with the passage of each generation. Thus, these trajectories are "precomputed," i.e., the whole MAV trajectory at the initial and 200th-generation is shown. This plot shows that the evolved kinematic parameters reduce altitude excursions away from the target trajectory by a factor of four while utilizing roughly the same amount of time-averaged-thrust that was specified by the initial set of kinematic parameters. Moreover, the evolved trajectory trend tracks the constant altitude desired trajectory, while the initial trajectory trend does not (it ascends).



(a) Target trajectory (dashed), initial precomputed trajectory (solid), and the 200th (dashed-dotted) precomputed trajectory evolved by the SEGS algorithm.



(b) Target trajectory (dashed), trajectory from initial conditions without any SEGS evolution (solid), and evolved sample trajectories after 20 flapping wing strokes when $N = 1$ and $N = 5$ (dotted and dashed-dotted, respectively).

Figure 9: Successful flapping wing MAV forward-flight trajectory control via selective evolutionary generation for both "precomputed" and "real-time" trajectories. These trajectory types are explained in the text.

The scheme averages 2 min 34 s to compute the output of 200 generations in Matlab on a 2.50 GHz dual-core processor laptop with 4.00 GB of RAM and the Windows Vista operating system. This time includes generational Theodorsen-Garrick model computation time. Without it, the scheme averages 0.092 ms per generation, measured over 200 generations.

A true MAV trajectory consists of piecewise trajectories where each sub-trajectory piece is produced during the time taken to complete a flapping wing stroke, and vehicle position and velocity at the end of a stroke represent the initial state for the next stroke. In this practical scenario, a SEGS-based controller only updates a flapping wing gait at the start of every stroke, and is free to devote the remaining time during stroke completion to other tasks. Each stroke elucidates fitness for comparison with either a succeeding or preceding flapping wing gait in accordance with Algorithm 1. As shown by the samples in Fig. 9(b), when starting from the same initial conditions of Fig. 9(a) and with the new goal of tracking a slight descending trajectory (which was chosen to illustrate SEGS control versatility, i.e., a different task from Fig. 9(a) and in the opposite direction from the trajectory produced by the first flapping wing gait) the depicted motion during 20 flapping wing strokes now represents the "real-time" evolution of MAV flight. Fig. 9(b) presents simulated results for $N = 1$ and $N = 5$. Corroborating Fig. 5 and the described theory, SEGS-based flapping wing gait evolution with a lower level of selectivity displays a more immediate trajectory tracking response at the start of the simulation and then strays, while SEGS-based flapping wing gait evolution with a higher level of selectivity has a slower response but is more adept at target following.

The $N = 5$ trajectory of Fig. 9(b) showcases the last goal of this paper: on-line MAV flapping wing gait trajectory control with minimal computation (here, one updated parameter per stroke). By construction (through use of an exponential fitness function), the control process is search-efficient. The control process also inherits the responsiveness properties that were previously illustrated for hover.

5. Limitations and Possible Mitigation Strategies

The two case studies in this paper demonstrate the feasibility of evolving MAV flapping wing gait in real-time by showing that computation can be performed quickly without the burden of an aerodynamic model. However, two limitations are also apparent from these case studies.

First, the evolutionary computation approach that is deployed in this work emulates nature, which provides the requisite aerodynamic model through fitness feedback. But the associated downside of this process is that natural phenomena take time to occur, and thus the described controller has to wait for feedback to adjust a gait. It follows that many wing beats must be observed to evolve a gait that produces a desired mean aerodynamic force vector. This requirement introduces time delays into the control loop that can lead to low bandwidth performance. During these time delays and the ensuing evolution transient, the MAV may pitch, roll or yaw in an unmodeled way, resulting in feedback to the controller that is different from the depiction in the above studies. In sum, computational burden has been traded off for convergence time.

The second limitation relates to issues of MAV gait parameter coupling and of the reachability of the entire search space in practice. Because a flapping wing MAV is a coupled electromechanical aeroelastic system, the prime movers that actuate the wings (such as drive motors or piezoelectric actuators) are affected by the aerodynamic forces and moments that are produced by the wings. Therefore, when a voltage command is specified to the prime mover for a particular motion profile, the structural dynamics and aerodynamic loads at that time may either prevent or augment prime mover motion, which implies that any disturbance affecting the MAV changes simultaneously with gait parameters. Since gait parameters vary with prime mover voltage instead of via direct command, it is possible that more than one gait parameter may change together. Such parameter coupling may lead to different portions of the gait parameter space being unreachable for different flight conditions and times.

In addition to this variable search space reachability, the controller search process may also be complicated by sensor measurement noise, which adversely affects fitness feedback. Nevertheless, in the face of all these complications, the theory proves that the described controller can still evolve feasible parameters that achieve maximal fitness. As evidenced by the benchmark results, where convergence to a suitable but not necessarily unique gait phenotype occurs, the fact that this convergence is affected by initial conditions and the number of fit solutions in the search space means that variable parameter coupling and imperfect sensors will also affect the convergence process, with different fit gait parameters attainable at different times. Corresponding time delays associated with traversing a less-connected gait parameter space and also with the need to increase confidence in sensor readings then tie this limitation to the first one above.

Despite these two limitations, this paper has shown the viability of simultaneously evolving optimal MAV flapping wing gaits efficiently and resiliently, adapting on-line, and, through model-independence, allowing feedback from either experimental sensors or alternate external models (affording control versatility for hover or forward flight, unsteady or quasi-steady aerodynamics, and any dynamics or wing kinematics). One obvious way to mitigate the first limitation above, while keeping the one wing beat per generation constraint, is to increase the wing beat frequency. A second way that mitigates both limitations is to combine the control process in this work with a model-based feedback controller (i.e., to use an inner-loop, outer-loop approach), where the model-based controller also adjusts gait parameters once per wing beat using an *a priori* model that is an imperfect map of gait parameters to cycle-averaged forces and moments. Thus, the model-based controller stabilizes and achieves large-tolerance tracking performance, while the evolutionary controller fine-tunes gait parameters to improve performance. This type of control scheme is deferred to future work, along with a comparison to other model-based methods such as adaptive control and extremum seeking control. Related to the second mitigation strategy, a similar way to mitigate the second limitation is to ensure that initial conditions for gait parameters are always near converged solutions (i.e., the evolutionary process is limited to small time-scales only), which also implies that evolved gaits for a desirable target are good initial conditions for the next target, and that successive desirable targets are never too far apart.

6. Conclusions

Two Micro Air Vehicle (MAV) flapping wing gait evolution applications are presented in this paper. The first application, having an objective of hovering flight, demonstrates the efficacy of this paper's chosen evolutionary computation control technique in optimizing flapping wing gaits. It also serves as a test of algorithm resilience, and as a benchmark problem on which to compare related methods. The algorithm itself can be optimally search-efficient in dynamic environments, trading off prior information about the search space for search effort savings as quickly as possible. Optimality arises from a maximization of search entropy, which results from minimizing a fitness-expectation of information. The process lacks bias and is model-independent, thereby avoiding erroneous predispositions of the search that result when a utilized model is rendered incorrect by environment changes. The technique is sufficient for stochastic behavior that is in turn sufficient for resilience to search objective variations caused by the environment changes.

The second application validates the capability of the chosen evolutionary computation technique to achieve on-line control for a realistic, complex problem. The objective of this application is to vary wing beat kinematics to ensure that the trajectory of the center of mass of the flapping wing, which moves under the influence of coupled aerodynamic (lift and thrust) forces generated by a flapping wing model in unsteady air flow, follows a desired trajectory. This paper's demonstration of a successful solution to tackling this physically-motivated problem, in combination with satisfactory algorithm metrics and performance on the first application, confirm the existence of an efficient and resilient method of controlling the gaits of a flapping wing MAV.

Acknowledgments

The authors thank Hikaru Aono, Chang-Kwon Kang, Wei Shyy, and Patrick Trizila for their assistance with the models of MAV flapping wing flight that were used in this paper. This work was supported in part by a Natural Sciences and Engineering Research Council of Canada Post-Graduate Scholarship, NSERC PGS D3-358609-2008. We are also very grateful for anonymous reviewer feedback that resulted in a significantly improved paper.

References

Arabagi, V., Hines, L., Sitti, M., 2012. Design and manufacturing of a controllable miniature flapping wing robotic platform. The International Journal of Robotics Research 31, 785–800. Doi:10.1177/0278364911434368.

Augustsson, P., Wolff, K., Nordin, P., 2002. Creation of a learning, flying robot by means of evolution, in: Proceedings of the 2002 Genetic and Evolutionary Computation Conference, pp. 1279–1285.

Beyer, H.G., Schwefel, H.P., 2002. Evolution strategies: A comprehensive introduction. Natural Computing 1, 3–52. Doi:10.1023/A:1015059928466.

Boddhu, S.K., Gallagher, J.C., 2010. Evolving neuromorphic flight control for a flapping-wing mechanical insect. International Journal of Intelligent Computing and Cybernetics 3, 94–116. Doi:10.1108/17563781011028569.

Brémaud, P., 1999a. Markov Chains: Gibbs fields, Monte Carlo Simulation and Queues. Springer. Ch. 7.

Brémaud, P., 1999b. Markov Chains: Gibbs fields, Monte Carlo Simulation and Queues. Springer. Ch. 2.

Brunton, S.L., Rowley, C.W., 2013. Empirical state-space representations for Theodorsen's lift model. Journal of Fluids and Structures 38, 174–186. Doi:10.1016/j.jfluidstructs.2012.10.005.

Caetano, J.V., de Visser, C.C., de Croon, G.C.H.E., Remes, B., de Wagter, C., Verboom, J., Mulder, M., 2013. Linear aerodynamic model identification of a flapping wing MAV based on flight test data. International Journal of Micro Air Vehicles 5, 273–286. Doi:10.1260/1756-8293.5.4.273.

Chung, S.J., Dorothy, M., 2010. Neurobiologically inspired control of engineered flapping flight. Journal of Guidance, Control, and Dynamics 33, 440–453. Doi:10.2514/1.45311.

Conn, A., Burgess, S., Hyde, R., Ling, C.S., 2006. From natural flyers to the mechanical realization of a flapping wing micro air vehicle, in: Proceedings of the 2006 IEEE International Conference on Robotics and Biomimetics, pp. 439–444. Doi:10.1109/ROBIO.2006.340232.

Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C., 2009. Introduction to Algorithms. MIT Press. third edition. Pp. 1170.

De Croon, G., De Clercq, K., Ruijsink, R., Remes, B., De Wagter, C., 2009. Design, aerodynamics, and vision-based control of the DelFly. International Journal of Micro Air Vehicles 1, 71–97. Doi:10.1260/175682909789498288.

Deng, X., Schenato, L., Sastry, S.S., 2006a. Flapping flight for biomimetic robotic insects: Part II -- flight control design. IEEE Transactions on Robotics 22, 789–803. Doi:10.1109/TRO.2006.875483.

Deng, X., Schenato, L., Wu, W.C., Sastry, S.S., 2006b. Flapping flight for biomimetic robotic insects: Part I – system modeling. IEEE Transactions on Robotics 22, 776–788. Doi:10.1109/TRO.2006.875480.

Doman, D.B., Oppenheimer, M.W., Sigthorsson, D.O., 2010. Wingbeat shape modulation for flapping-wing micro-air-vehicle control during hover. Journal of Guidance, Control, and Dynamics 33, 724–739. Doi:10.2514/1.47146.

Doncieux, S., Hamdaoui, M., 2011. Evolutionary algorithms to analyse and design a controller for a flapping wings aircraft, in: Doncieux, S., Bredèche, N., Mouret, J.B. (Eds.), New Horizons in Evolutionary Robotics. Springer Berlin Heidelberg. volume 341 of *Studies in Computational Intelligence*, pp. 67–83. Doi:10.1007/978-3-642-18272-3_6.

Duhamel, P.E.J., Pérez-Arancibia, N.O., Barrows, G.L., Wood, R.J., 2013. Biologically inspired optical-flow sensing for altitude control of flapping-wing microrobots. IEEE/ASME Transactions on Mechatronics 18, 556–568. Doi:10.1109/TMECH.2012.2225635.

Ellington, C., 1984. The aerodynamics of hovering insect flight. IV. aerodynamic mechanisms. Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences 305, 79–113. Doi:10.1098/rstb.1984.0052.

Ellington, C.P., 1999. The novel aerodynamics of insect flight: applications to micro-air vehicles. Journal of Experimental Biology 202, 3439–3448.

Ellington, C.P., van den Berg, C., Willmott, A.P., Thomas, A.L.R., 1996. Leading-edge vortices in insect flight. Nature 384, 626–630. Doi:10.1038/384626a0.

Fenelon, M.A.A., 2009. Biomimetic flapping wing aerial vehicle, in: Proceedings of the 2008 IEEE International Conference on Robotics and Biomimetics, pp. 1053–1058. Doi:10.1109/ROBIO.2009.4913146.

Forrester, A.I.J., Keane, A.J., 2009. Recent advances in surrogate-based optimization. Progress in Aerospace Sciences 45, 50–79. Doi:10.1016/j.paerosci.2008.11.001.

Gallagher, J.C., 2013. An islands-of-fitness compact genetic algorithm approach to improving learning time in swarms of flapping-wing micro air vehicles, in: Kim, J.H., Matson, E.T., Myung, H., Xu, P. (Eds.), Robot Intelligence Technology and Applications 2012. Springer Berlin Heidelberg. volume 208 of *Advances in Intelligent Systems and Computing*, pp. 855–862. Doi:10.1007/978-3-642-37374-9_82.

Gallagher, J.C., Humphrey, L.R., Matson, E., 2014. Maintaining model consistency during in-flight adaptation in a flapping-wing micro air vehicle, in: Kim, J.H., Matson, E.T., Myung, H., Xu, P., Karray, F. (Eds.), Robot Intelligence Technology and Applications 2. Springer International Publishing. volume 274 of *Advances in Intelligent Systems and Computing*, pp. 517–530. Doi:10.1007/978-3-319-05582-4_45.

Gallagher, J.C., Oppenheimer, M.W., 2012. An improved evolvable oscillator and basis function set for control of an insect-scale flapping-wing micro air vehicle. Journal of Computer Science and Technology 27, 966–978. Doi:10.1007/s11390-012-1277-1.

Garrick, I.E., 1936. Propulsion of a flapping and oscillating airfoil. Technical Report 567. NACA.

Gogulapati, A., Friedmann, P.P., Martins, J.R.R.A., 2014. Optimization of flexible flapping-wing kinematics in hover. AIAA Journal 52, 2342–2354. Doi:10.2514/1.J053083.

Goppert, J., Gallagher, J.C., Hwang, I., Matson, E., 2014. Model checking of a flapping-wing mirco-air-vehicle trajectory tracking controller subject to disturbances, in: Kim, J.H., Matson, E.T., Myung, H., Xu, P., Karray, F. (Eds.), Robot Intelligence Technology and Applications 2. Springer International Publishing. volume 274 of *Advances in Intelligent Systems and Computing*, pp. 531–543. Doi:10.1007/978-3-319-05582-4_46.

Hunt, R., Hornby, G.S., Lohn, J.D., 2005. Toward evolved flight, in: Proceedings of the 2005 Genetic and Evolutionary Computation Conference, pp. 957–964. Doi:10.1145/1068009.1068172.

Jaynes, E.T., 1957. Information theory and statistical mechanics. The Physical Review 106, 620–630. Doi:10.1103/PhysRev.106.620.

Jaynes, E.T., 1981. Entropy and search theory, in: Proceedings of the First Maximum Entropy Workshop.

Jin, Y., 2011. Surrogate-assisted evolutionary computation: Recent advances and future challenges. Swarm and Evolutionary Computation 1, 61–70. Doi:10.1016/j.swevo.2011.05.001.

Keennon, M., Klingebiel, K., Won, H., Andriukov, A., 2012. Development of the nano hummingbird: A tailless flapping wing micro air vehicle, in: Proceedings of the 50th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition. Doi:10.2514/6.2012-588.

Khan, Z.A., Agrawal, S.K., 2005. Force and moment characterization of flapping wings for micro air vehicle application, in: Proceedings of the 2005 American Control Conference, IEEE. pp. 1515–1520. Doi:10.1109/ACC.2005.1470180.

Khan, Z.A., Agrawal, S.K., 2007. Control of longitudinal flight dynamics of a flapping-wing micro air vehicle using time-averaged model and differential flatness based controller, in: Proceedings of the 2007 American Control Conference, pp. 5284–5289. Doi:10.1109/ACC.2007.4283052.

Lim, D., Jin, Y., Ong, Y.S., Sendhoff, B., 2010. Generalizing surrogate-assisted evolutionary computation. IEEE Transactions on Evolutionary Computation 14, 329–355. Doi:10.1109/TEVC.2009.2027359.

Lin, S., Hsiao, F., Chen, C., Shen, J., 2010. Altitude control of flapping-wing MAV using vision-based navigation, in: Proceedings of the 2010 American Control Conference, pp. 21–26. Doi:10.1109/ACC.2010.5531448.

Mahjoubi, H., Byl, K., 2013. Trajectory tracking in the sagittal plane: decoupled lift/thrust control via tunable impedance approach in flapping-wing MAVs, in: Proceedings of the 2013 American Control Conference, pp. 4951–4956. Doi:10.1109/ACC.2013.6580606.

Malhan, R., Benedict, M., Chopra, I., 2012. Experimental studies to understand the hover and forward flight performance of a MAV-scale flapping wing concept. Journal of the American Helicopter Society 57, 1–11. Doi:10.4050/JAHS.57.022003.

de Margerie, E., Mouret, J.B., Doncieux, S., Meyer, J.A., 2007. Artificial evolution of the morphology and kinematics in a flapping-wing mini-UAV. Bioinspiration & Biomimetics 2, 65–82. Doi:10.1088/1748-3182/2/4/002.

Meerkov, S.M., 1979. Mathematical theory of behavior — individual and collective behavior of retardable elements. Mathematical Biosciences 43, 41–106. Doi:10.1016/0025-5564(79)90103-2.

Menezes, A.A., 2010. Selective Evolutionary Generation Systems: Theory and Applications. Ph.D. thesis. University of Michigan.

Menezes, A.A., Kabamba, P.T., 2014. Optimal search efficiency of Barker's algorithm with an exponential fitness function. Optimization Letters 8, 691–703. Doi:10.1007/s11590-013-0608-7.

Menezes, A.A., Kabamba, P.T., 2016. Efficient search and responsiveness trade-offs in a Markov chain model of evolution in dynamic environments. Mathematical Biosciences 276, 44–58. Doi:10.1016/j.mbs.2016.03.002.

Milano, M., Gharib, M., 2005. Uncovering the physics of flapping flat plates with artificial evolution. Journal of Fluid Mechanics 534, 403–409. Doi:10.1017/S0022112005004842.

Mouret, J.B., Doncieux, S., Meyer, J.A., 2006. Incremental evolution of target-following neuro-controllers for flapping-wing animats, in: Nolfi, S., Baldassarre, G., Calabretta, R., Hallam, J.C.T., Marocco, D., Meyer, J.A., Miglino, O., Parisi, D. (Eds.), From Animals to Animats 9. Springer. volume 4095 of *Lecture Notes in Computer Science*, pp. 606–618. Doi:10.1007/11840541_50.

Nielsen, E.J., Diskin, B., 2013. Discrete adjoint-based design for unsteady turbulent flows on dynamic overset unstructured grids. AIAA Journal 51, 1355–1373. Doi:10.2514/1.J051859.

Olhofer, M., Yankulova, D., Sendhoff, B., 2011. Autonomous experimental design optimization of a flapping wing. Genetic Programming and Evolvable Machines 12, 23–47. Doi:10.1007/s10710-010-9107-0.

Orlowski, C.T., Girard, A.R., 2012. Dynamics, stability, and control analyses of flapping wing micro-air vehicles. Progress in Aerospace Sciences 51, 18–30. Doi:10.1016/j.paerosci.2012.01.001.

Pérez-Arancibia, N.O., Duhamel, P.E.J., Ma, K.Y., Wood, R.J., 2015. Model-free control of a hovering flapping-wing microrobot. Journal of Intelligent & Robotic Systems 77, 95–111. Doi:10.1007/s10846-014-0096-8.

Pérez-Arancibia, N.O., Whitney, J.P., Wood, R.J., 2013. Lift force control of flapping-wing microrobots using adaptive feedforward schemes. IEEE/ASME Transactions on Mechatronics 18, 155–168. Doi:10.1109/TMECH.2011.2163317.

Persson, P.O., Willis, D.J., Peraire, J., 2010. The numerical simulation of flapping wings at low Reynolds numbers, in: Proceedings of the 48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition. Doi:10.2514/6.2010-724.

Queipo, N.V., Haftka, R.T., Shyy, W., Goel, T., Vaidyanathan, R., Tucker, P.K., 2005. Surrogate-based analysis and optimization. Progress in Aerospace Sciences 41, 1–28. Doi:10.1016/j.paerosci.2005.02.001.

Ratti, J., Jones, E., Vachtsevanos, G., 2011. Fixed frequency, variable amplitude (FiFVA) actuation systems for micro aerial vehicles, in: Proceedings of the 2011 IEEE International Conference on Robotics and Automation, pp. 165–171. Doi:10.1109/ICRA.2011.5979699.

Regan, W., Van Breugel, F., Lipson, H., 2006. Towards evolvable hovering flight on a physical ornithopter, in: Proceedings of the 10th International Conference on the Simulation and Synthesis of Living Systems.

Roberts, J.W., Moret, L., Zhang, J., Tedrake, R., 2010. Motor learning at intermediate Reynolds number: Experiments with policy gradient on the flapping flight of a rigid wing, in: From Motor Learning to Interaction Learning in Robots. Springer, pp. 293–309. Doi:10.1007/978-3-642-05181-4_13.

Rudolph, G., 1994. Convergence analysis of canonical genetic algorithms. IEEE Transactions on Neural Networks 5, 96–101. Doi:10.1109/72.265964.

Salles, R., Schiele, A., 2004. Analysis of flapping wing robots for planetary exploration: An evolutionary approach, in: Proceedings of the 8th ESA Workshop on Advanced Space Technologies for Robotics and Automation.

Sane, S.P., Dickinson, M.H., 2001. The control of flight force by a flapping wing: Lift and drag production. Journal of Experimental Biology 204, 2607–2626.

Schenato, L., 2003. Analysis and control of flapping flight: From biological to robotic insects. Ph.D. thesis. University of California, Berkeley.

Shannon, C.E., 1948. A mathematical theory of communication. Bell System Technical Journal 27, 379–423 and 623–656.

Shim, Y.S., Kim, C.H., 2006. Evolving physically simulated flying creatures for efficient cruising. Artificial Life 12, 561–591. Doi:10.1162/artl.2006.12.4.561.

Shyy, W., Aono, H., Chimakurthi, S.K., Trizila, P., Kang, C.K., Cesnik, C.E.S., Liu, H., 2010. Recent progress in flapping wing aerodynamics and aeroelasticity. Progress in Aerospace Sciences 46, 284–327. Doi:10.1016/j.paerosci.2010.01.001.

Shyy, W., Lian, Y., Tang, J., Viieru, D., Liu, H., 2008. Aerodynamics of Low Reynolds Number Flyers. Cambridge University Press. Pp. xvii.

Taha, H.E., Hajj, M.R., Nayfeh, A.H., 2012. Flight dynamics and control of flapping-wing MAVs: A review. Nonlinear Dynamics 70, 907–939. Doi:10.1007/s11071-012-0529-5.

Theodorsen, T., 1935. General theory of aerodynamic instability and the mechanism of flutter. Technical Report 496. NACA.

Trizila, P., Kang, C.K., Visbal, M., Shyy, W., 2008. A surrogate model approach in 2D versus 3D flapping wing aerodynamic analysis, in: Proceedings of the 12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference. Doi:10.2514/6.2008-5914.

Van Breugel, F., Lipson, H., 2005. Evolving buildable flapping ornithopters, in: Proceedings of the 2005 Genetic and Evolutionary Computation Conference.

Vandenheede, R.B.R., Bernal, L.P., Morrison, C.L., Gogulapati, A., Friedmann, P.P., Kang, C.K., Shyy, W., 2014. Experimental and computational study on flapping wings with bio-inspired hover kinematics. AIAA Journal 52, 1047–1058. Doi:10.2514/1.J052644.

Weng, L., Cai, W., Zhang, M.J., Liao, X.H., Song, D.Y., 2007. Neural-memory based control of Micro Air Vehicles (MAVs) with flapping wings, in: Liu, D., Fei, S., Hou, Z.G., Zhang, H., Sun, C. (Eds.), Advances in Neural Networks — ISNN 2007. Springer. volume 4491 of *Lecture Notes in Computer Science*, pp. 70–80. Doi:10.1007/978-3-540-72383-7_10.

Wood, R.J., 2008. The first takeoff of a biologically inspired at-scale robotic insect. IEEE Transactions on Robotics 24, 341–347. Doi:10.1109/TRO.2008.916997.

Yan, J., Wood, R.J., Avadhanula, S., Sitti, M., Fearing, R.S., 2001. Towards flapping wing control for a micromechanical flying insect, in: Proceedings of the 2001 IEEE International Conference on Robotics and Automation, pp. 3901–3908. Doi:10.1109/ROBOT.2001.933225.

Yang, L.J., Hsu, C.K., Hsiao, F.Y., Feng, C.K., Shen, Y.K., 2009. A Micro-Aerial-Vehicle (MAV) with figure-of-eight flapping induced by flexible wing frames, in: Proceedings of the 47th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition. Doi:10.2514/6.2009-875.