

Research Article

Qualitative Functional Decomposition Analysis of Evolved Neuromorphic Flight Controllers

Sanjay K. Boddhu and John C. Gallagher

Department of Computer Science and Engineering, Wright State University, Dayton, OH 45435, USA

Correspondence should be addressed to Sanjay K. Boddhu, sboddhu@cs.wright.edu

Received 21 February 2012; Accepted 21 May 2012

Academic Editor: P. Balasubramaniam

Copyright © 2012 S. K. Boddhu and J. C. Gallagher. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In the previous work, it was demonstrated that one can effectively employ CTRNN-EH (a neuromorphic variant of EH method) methodology to evolve neuromorphic flight controllers for a flapping wing robot. This paper describes a novel frequency grouping-based analysis technique, developed to qualitatively decompose the evolved controllers into explainable functional control blocks. A summary of the previous work related to evolving flight controllers for two categories of the controller types, called autonomous and nonautonomous controllers, is provided, and the applicability of the newly developed decomposition analysis for both controller categories is demonstrated. Further, the paper concludes with appropriate discussion of ongoing work and implications for possible future work related to employing the CTRNN-EH methodology and the decomposition analysis techniques presented in this paper.

1. Introduction

Most, if not all, existing bird-sized and insect-sized flapping-wing vehicles possess only a small number of actively controlled degrees of freedom. In these vehicles, the bulk of the wing motions are generated via a combination of actively driven linkages (motors and armatures, piezoelectric beams, etc.) and passively driven elements (wing flex or rotation via dynamic pressure loading, etc.) [1, 2]. The number of controlled degrees of freedom is often minimized to simplify control and to limit the number of bulky actuators carried on board. In theory, both bird-sized [1] and insect-sized [2] robots can sustain stable flight with controllers generating actuation signals for only few degrees of freedom. But it would require taking advantage of every possible degree of freedom available in the robot to achieve sophisticated maneuvers that are possible in their biological counterparts. Thus, there exists a possibility that applying a learning or adaptable controller techniques [1, 3–6] to the control of the insect-sized flapping wing vehicles, hereafter referred to as Microlevel Flapping Wing Robots (MFWRs), will likely to produce more biomimetic control and maneuver patterns that evade traditional controller design. One can imagine

two basic approaches to the “adaptable controller” problem. First, one might attempt to hybridize an adaptive system to a traditional controller in the hope that the combined system could learn the specific needs of an individual vehicle by augmenting a base controller. Second, one might attempt to construct an adaptable controller that could learn acceptable control laws tabula rasa either all-at-once or via a staged approach. Even if tabula rasa methods could be made to work, one would incur a responsibility to explain the operation of controllers that, though functional, might operate in ways not conformant with existing explanatory paradigms. In previous work [4, 7–9], the authors were able to demonstrate controllers could be “learned from scratch” by verifying the idea within a framework of neuromorphic evolvable hardware. Further, this previous work also demonstrated the feasibility of neuromorphic adaptive hardware implementations that provide computational advantage over existing adaptive control techniques using similar neural substrates [10, 11]. The work mentioned in this paper focuses more intensely on that problem of explaining what those controllers do and how they do it. It will discuss subsequent work that was undertaken to analyze those evolved flight controllers with newly developed frequency-based and

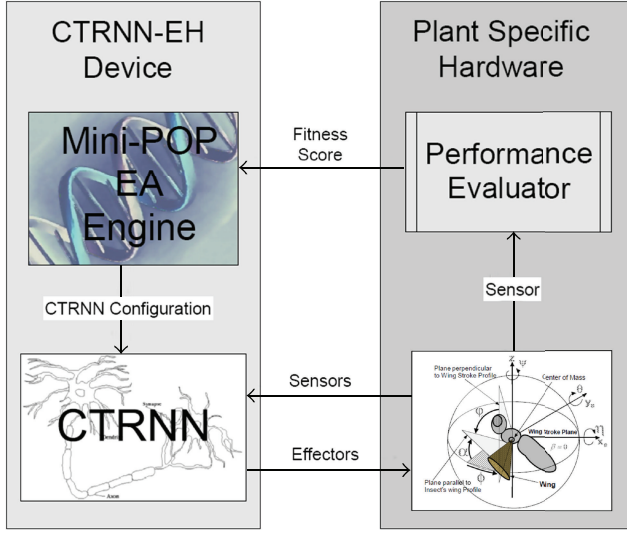


FIGURE 1: Schematic of CTRNN-EH Framework.

preexisting modularization decomposition methods. In this vein, the background knowledge necessary to understand the terminology, methods, and approaches employed as part of the above-mentioned CTRNN-EH framework are briefly explained in Section 2. Following is Section 3 describing the specific methods and models employed to successfully evolve CTRNN-EH flight controllers for a specific MFWR model. Section 4 describes the details of the proposed analysis methods and their applicability to the evolved flight controllers, followed by Section 5 with concluding remarks on the current work and ongoing future work.

2. Background and Previous Work

2.1. CTRNN-EH Framework. The CTRNN-EH framework introduced in the previous works [5, 12, 13] is summarized schematically in Figure 1. The CTRNN-EH framework is a neuromorphic variant of the standard Evolvable Hardware paradigm using Continuous Time Recurrent Neural Networks (CTRNNs) as the reconfigurable hardware substrate. CTRNNs are networks of Hopfield continuous model neurons [13] with unconstrained connection weight matrices. CTRNN neural activity and outputs are governed by an n th degree differential equation of the form:

$$\tau \frac{dy_i}{dt} = -y_i + \sum_{j=1}^N w_{ji} \sigma(y_j + \theta_j) + s_i I_i(t), \quad (1)$$

where y_i is the state of neuron i , τ_i is time constant of neuron i , w_{ji} is the strength of the connection from the j th to i th the neuron, θ is a bias term, $\sigma(x) = 1/(1 + e^{-x})$ is the standard logistic activation function, and $I_i(t)$ represents a weighted sensory input with strength s_i . The set of parameters defining a neuron I (incoming weights, time constant, and bias) are called an individual *neuron configuration*, and a collection of these individual neuron configurations for a

given network is called a *network configuration* or *CTRNN configuration*. These CTRNNs have been proven to be universal approximators of any smooth dynamics [14]. The practical benefit of this is that in principle, any possible control law can be approximated arbitrarily well given enough CTRNN neurons. In practice, even small networks of CTRNN neurons are capable of producing complex dynamical behavior. Based on the Evolvable Hardware (EH), principle of evolving optimized and desired configurations in a reconfigurable substrate using evolutionary algorithm techniques. CTRNNs are evolved to produce, the right control signal, using the evolutionary algorithms. The training of the CTRNNs is finding the appropriate parameter settings, that is, configuring the neuron settings in the network. The evolutionary algorithms search the given possible settings of the neuron and find the optimal settings for the network as a whole, to produce the required control signals. The CTRNNs functioning with optimal settings produced by the EA is called as the evolved controller for the given control problem being dealt. Based on previous work, the Minipop algorithm [15] is chosen as the evolutionary algorithm for the CTRNN-EH framework. The Minipop algorithm is a light weight evolutionary algorithm driven by mutation and hypermutation [15], more details of the same can be found in [13, 15].

2.2. Previous Work. The above-mentioned CTRNN-EH framework has been successfully employed to control legged locomotion in both real and simulated hexapod walkers [3, 16] by author's colleagues. These efforts concentrated on solving a learning locomotion control problem for the hexapod robots with twelve degrees of freedom, from scratch, without any preknowledge of the robot's physical characteristics (like weight and any leg damages). Conceptually, each leg of the hexapod (with two degrees of freedom in its actuators) would require optimal oscillatory patterns in its two actuators, with appropriate phase relations to aid in generating forces to move the hexapod forward or backward directions. Moreover, any controller that claims to provide optimal locomotion controller for the hexapod has to take into consideration the required optimal oscillatory dynamics in each leg as well as the needed collaborative dynamics among all the six legs to generate optimal and energy-efficient motion in the hexapod [3]. This complex mix of local and distributed locomotion pattern generation problem was successfully addressed by the CTRNN-EH architecture mentioned in [6]. Moreover, the evolved oscillatory CTRNN-EH locomotion controllers in those experiments embedded a large amount of practical functionality in very small numbers of neurons. These evolved controllers were capable of optimally controlling variously weighted bodies with or without damaged legs. The controllers could do this without reevolution and could adapt their dynamics on the fly by entraining to external sensory input [6, 13]. Further the work conducted to understand these evolved CTRNN-EH controllers (for legged locomotion) has resulted in a set of dynamical module analysis concepts [5, 6, 16] that can be employed to predict and explain the behavior of these

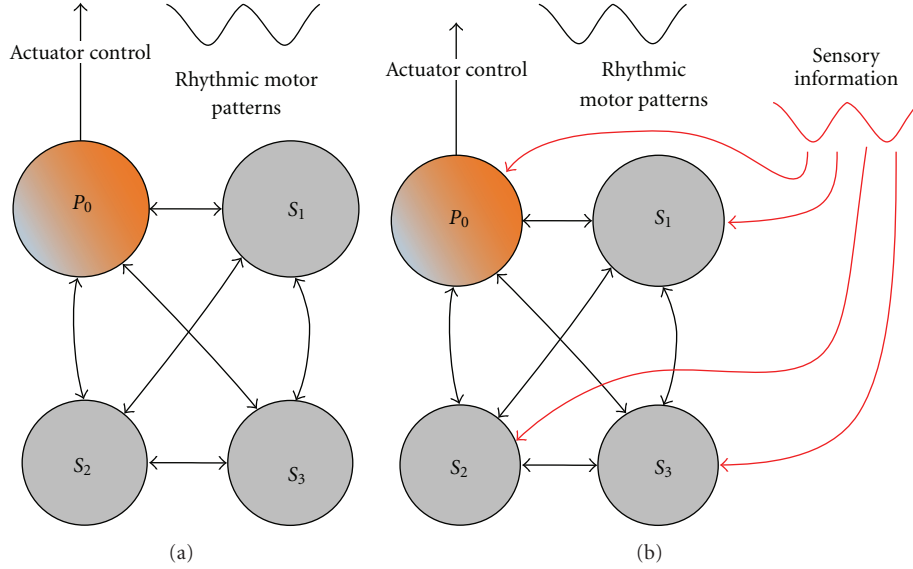


FIGURE 2: Illustrative representation of the assembly of neurons forming the autonomous neuromorphic (shown in (a)) and nonautonomous Neuromorphic (shown in (b)) controllers. “P” notation is used for primary neuron directly controlling the actuator of the robot under control and “S” is the secondary neuron, which aids in generating appropriate external dynamics required for the primary neuron. The numbering of the neurons in the network is arbitrarily chosen as appropriate.

evolved controllers, and controllers with similar nature, in terms of their functional sustainability and failure.

However, conceptually the flapping-wing flight problem shares the requirement of generating optimal oscillatory dynamics for desired flight behavior; with the hexapod walker problem, the former has inherent instability in its body dynamics, introduced by virtue of the medium of its flight (i.e., in three, dimensional space with constantly varying center of mass). This possible inherent instable body dynamics present in flapping-wing vehicles might make the CTRNN-EH based learning more challenging to be effective than when applied to the hexapod walkers, to generate optimal actuator dynamics. Further, the dynamical module analysis [6, 16] that was successful to understand the evolved locomotion controllers might not be applicable for the possible CTRNN-EH flight controllers. Nonetheless, the capabilities of CTRNN-EH controllers to produce smooth dynamics and provide provisions to adapt and modulate those produced dynamics observed in the previous work [5] sufficiently justifies the needed efforts presented in the authors published paper [4, 7–9] to evolve flapping flight controllers. Although providing the indetail description of varied possible modes of the CTRNN-EH controllers is beyond the scope of this paper, two basic modes are defined below, which are more pertinent to understand the experiments presented here in the paper.

Autonomous Controllers. These are neural network configurations that produce oscillatory signals without any external sensory inputs [3, 5]. As shown in the concept illustrative Figure 2(a), these configurations can generate autonomous and periodic dynamics in the neural network without any external triggers/sensor inputs; thus these would be referred

to as autonomous neuromorphic controllers in the later sections.

Nonautonomous Controllers. These are neural network configurations that can produce appropriate oscillatory patterns only when coupled to some other oscillatory system. They are more completely discussed in [8]. As shown in the concept illustrative Figure 2(b), these configurations can generate varied dynamics in the neural network only in sync with specific external triggers/sensor inputs provided to them, thus these would be referred as nonautonomous neuromorphic controllers in the later sections.

3. Evolved Flapping Wing Flight Controllers

This section describes authors’s successful efforts aimed at evolving autonomous and non-autonomous CTRNN-EH controllers for a number of flapping-wing vehicle flight modes [4, 7, 9]. The section will begin with a brief description of the microlevel flapping winged robot (MFWR) model employed, followed by description of MFWR-specific CTRNN-EH control architecture and the evolution strategy applied at evolving different flight controllers.

3.1. Microlevel Flapping Winged Robot (MFWR). Micromechanical flying insect model was developed by MFI team at UC Berkeley [2] to facilitate the investigative study to build a real flapping wing robot with a 100 mg mass and a 25 mm wings span. Based on the available MFI literature on that robot’s wing aerodynamics and body dynamics, we reconstructed a model with linear actuator dynamics called Microlevel Flapping Winged Robot (MFWR) [4]. Our model was verified in simulation and found to be a match for the

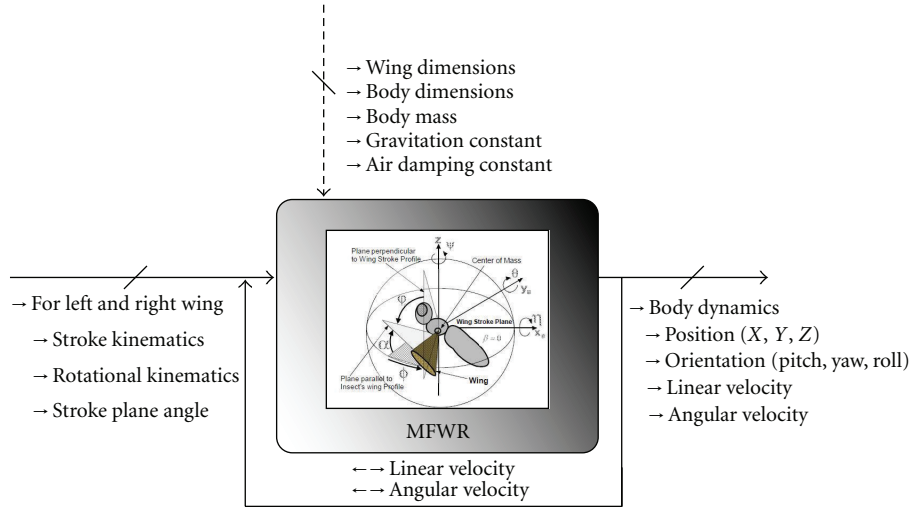


FIGURE 3: Schematic of the simulated Microlevel Flapping Winged Robot (MFWR).

Berkeley model in that we could reproduce their published flight envelopes and behaviors. The *Wing Aerodynamics* (WA) module of model is based on the mathematical model, developed from empirical study conducted on the RoboBly [17]. The WA module generates the aerodynamic forces and torques for given wing kinematics. The *Body Dynamics Module* takes the aerodynamics forces and torques generated by the wing kinematics and integrates them along with the dynamical model of the MFI body, thus computing the body's position and the attitude as a function of time. The readers are directed to [18] for detail descriptions of these modules.

In brief, the implemented MFWR model takes wing (left and right) actuation parametric inputs like stroke and rotation trajectories and produces the position and attitude information of the MFWR in the world coordinate system as shown in Figure 3. Additionally, the model also takes the body linear velocity and angular velocity from the previous simulation step making it an internal feedback system. The MFWR model has been simulated with realistic and envisioned physical and environmental parameters like robot's mass of 100 mg, envisioned wing length of 25 mm, acceleration due to gravity value of 9.8 m/sec^2 , air dampening coefficient of $62.3 \times 10^{-6} \text{ N-sec/m}$, stroke angle range of -60 to $+60$, rotational angle range of -90 to $+90$, and with some constant parameters derived and mentioned in [2, 17]. Further, the differential equations characterizing the internal dynamics of the robot model have been computed using Runge-Kutta (RK4) numerical method.

3.2. Control Architecture and Evolutionary Algorithm Specifications. After some preliminary experimentation conducted with the MFWR model, the custom CTRNN-EH control architecture shown in Figure 4 was chosen to be less-redundant architecture and with enough flexibility to embed into its dynamics the optimal control laws required for different flight modes [4]. The actuation dynamics modeled

for each wing can presently control the stroke and rotational parameters of the wing, and the stroke plane of the wing is fixed at a constant angle. This distributed CTRNN architecture shown in Figure 4 has a central core network block that can produce signals, which are delivered to each wing trajectory actuation after being processed by a delay network block. Figure 4 shows the interfacing of the controller architecture with the MFI wing parameters. The delay networks are placed to produce asymmetric/symmetric actuations of left wing with respect to right wing or vice versa to produce net nonzero/zero torques and forces in Y-direction [2, 4]. For the set of experiments and results, being described in this paper, a fullyconnected eight neuron network is chosen for the central core network block and the evolutionary algorithm is employed to evolve the central core network and/or the delay duration in outer delay networks [4]. Further, the central core can accept sensory input, from the MFWR's status or external command, which is feed to each neuron in the core. The next subsection provides the details of the evolutionary algorithm employed in the present experiments.

As mentioned earlier, each individual CTRNN neuron is specified by one bias, one time constant, and eight weighted connections from all neurons in the network (seven connections to other neurons, one self-connection, and one sensor). Thus, the central core network, with eight-neurons, is fully specified by the numeric value settings of eighty-eight parameters, with eleven parameters for each neuron in a fully connected eight-neuron network. The aforementioned minipop EA is implemented with a population size of 4 and a mutation rate of 0.005. The genome length chosen was equal to total number of bits employed to encode a given CTRNN configuration. Each neuron parameter in the configuration is encoded in eight bits, which aggregates to a genome length of 704 bits to represent the central core network. The delay input interval duration for gate networks is encoded in an eight-bit string. Employing the aforementioned architecture and algorithm specifications

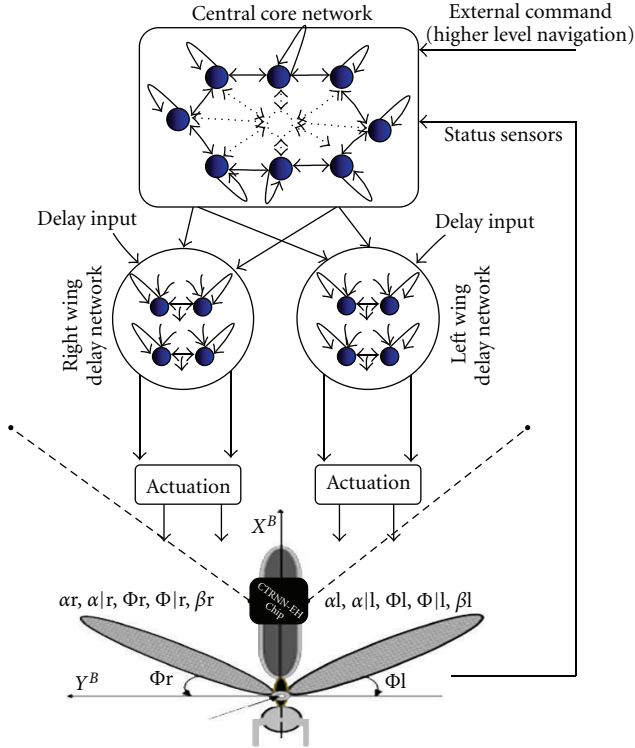


FIGURE 4: Interfacing CTRNN Architecture to MFWR Model to evolve and analyze flight controllers. In the figure, α (with range -90 to 90) indicates the instantaneous rotational angle of the wing, Φ (with range -60 to 60) indicates the instantaneous stroke angle of the wing, and β (with range -45 to 45) indicates the instantaneous stroke plane angle of the wing. The parameters with “|” indicate the rate of change for the parameter it is referring to and these parameters with “r” and “l” indicate the right (r) and left (l) wing parameters, respectively.

the next sections provide the details of the evolutionary runs and evaluation criterion applied to evolving autonomous and nonautonomous controllers.

3.3. Evolved Autonomous Flight Controllers. Three kinds of autonomous flight controllers, namely, cruising, altitude gain, and steering, were successfully evolved using the aforementioned architecture and algorithm, but with type-specific fitness evaluation criterion. For example, an acceptable behavior of MFWR under an evolved cruise mode controller is to produce motion in a forward direction that is greater than the motion in altitude or sideward directions. Moreover, it should also maintain zero angular velocity along the three vehicle frame axes (zero pitch, roll, and yaw). The later criteria of the expected controller can be met by employing preevolved CTRNN-EH gate networks with symmetric delays. But the first and primary criteria of the controller should be evolved in central core, since this is the only module capable of generating any dynamics to drive the wings. Thus, an evaluation function to capture this established cruise criteria should observe the motion of the MFWR under the control of the potential controller

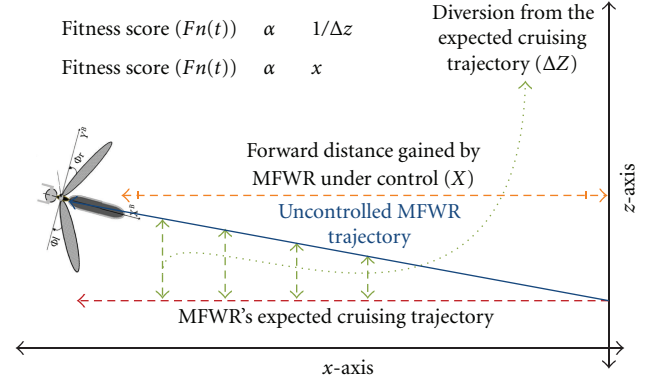


FIGURE 5: The figure shows the relation between the expected cruise behavior of MFWR and the fitness evaluation function employed to evolve the CTRNN-EH controllers to achieve the same behavior under control. An acceptable cruise controller has to propel the MFWR in forward direction and minimize the variation in the altitude. Thus, the fitness score employed to evolve the cruise controllers should reward any forward motion (in x -direction) and penalize any variations in altitude (in z -direction).

and reward the controller on generation of the forward motion and penalize it on generation of altitude variations. A pictorial representation of the expected autonomous cruise behavior and the established relation to its potential evaluation function is shown in Figure 5. Thus, the below evaluation function is designed with a minimizing fitness strategy [4], to capture the expected cruise behavior in MFWR:

$$\frac{\sum_{i=0}^{i=N} (|P_{zi}| - |P_{yi}| - P_{xi})}{N}, \quad (2)$$

where P_{zi} , P_{xi} , and P_{yi} are the instantaneous positional data of MFWR moment in Z (altitude), X (forward), and Y (sideward) directions under the control of the wing kinematics generated by the controller and N is the total number of time steps present in each evaluation period.

It can be observed that the above evaluation function captures the expected forward motion by placing constraints on the controller to maximize P_{xi} term, because it is a negating summation variable in the above minimizing fitness strategy function. Further, the altitude sustainability constraint is enforced by $|P_{zi}|$ term, which captures the averaging absolute measure of the variation in the altitude across the evaluation time, and evolved cruising controller's fitness should minimize this factor so as to favor the overall fitness value contributed by the P_{xi} term. Thus, at least theoretically, the established fitness evaluation function for evolving cruising mode controllers rewards the forward motion of the MFWR and penalizes the variations in its altitude, when placed under the control. Figure 6 shows the behavior of the MFWR under an evolved cruise controller, which has successfully evolved to produce appropriate stroke and rotation kinematics in the wings of the MFWR. Based on the above-mentioned fitness evaluation strategy, that is, to capture the flight mode behavior in terms of the positional information of the MFWR over a specified period

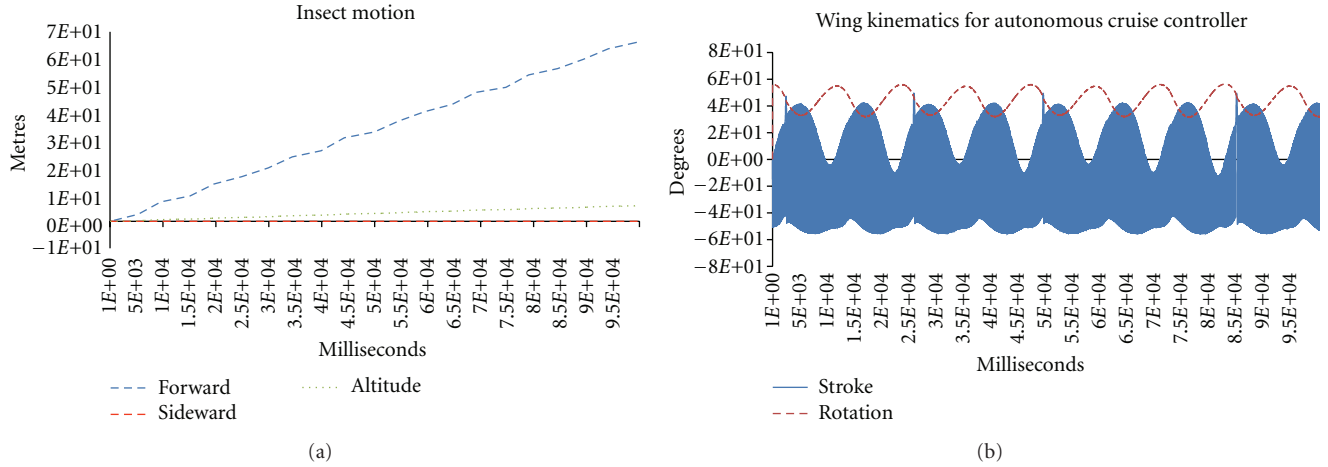


FIGURE 6: The motion of the MFWR in three dimensions (shown in (a)) controlled by a Cruise Mode Controller actuated wing kinematics (shown in (b)). Here the stroke kinematics has relatively higher beats rate than that of the rotation for the above controller shown in the figure, thus the oscillation cycles are cluttered, making it hard to visualize them with respect to the rotation kinematics.

of control, other two autonomous controllers, Altitude-gain and Steer, were also evolved successfully [8, 9]. Further details of the autonomous flight mode controller experiments can be found in [7–9].

3.4. Evolved Nonautonomous Flight Controllers. Two kinds of non-autonomous flight controllers were successfully evolved, namely, adaptive cruise mode controllers and polymorphic controllers [7, 8]. The adaptive cruise mode controllers were similar to their autonomous counterpart except that they were forced to sense the altitude of the MFWR and adapt accordingly during the evolution. The polymorphic controllers on the other hand were forced to change the behavior of the core central CTRNN module between autonomous altitude-gain controllers and autonomous cruise mode controllers, based on the external command. Both of these controllers employed the same fitness evaluation criterion, which was mentioned in the context of the autonomous controllers, on a varied range of the sensory inputs. An evolved nonautonomous cruise mode controller's effect on the wing kinematics of the MFWR is shown in Figure 7(a), along with the corresponding positional data of the MFWR and the sensory input. Also, one of the evolved CTRNN-EH polymorphic flight controller's effects on the wing kinematics is shown in Figure 7(b), along with the corresponding positional data of the MFWR robot and the external sensory input for invoking the desired modes.

4. Analysis of the Evolved Flight Controllers

The evolved CTRNN-EH flight controllers would be better accepted for practical deployment, at least for engineers, if their functionality can be explained using known general principles of engineering. As with all evolvable hardware-based methods, there exists a possibility that the acceptance of the evolved flight controllers, merely in terms of fitness score value (which is based on closely approximating the

acceptable overall body trajectory behavior), could have been exploited the possible underlying noise in the MFWR model to gain optimal controller status. Thus, the first possible analysis to accept the evolved flight controller is to diligently observe and validate the insect's temporal behavior when coupled with the evolved controller's dynamics and determine if they satisfy the known principal physical characteristics of the MFWR model flight behavior. Further, it would be of interest to explain the evolved controllers by possible decomposition of the CTRNN-EH layer in terms of logical control blocks. The next subsections deal with analyzing the evolved controllers with two deduced approaches mentioned below.

4.1. Acceptability Analysis. During the course of this work, it was deduced that the acceptability of the physical behavior of the MFWR flight, produced by the evolved flight controllers could be readily understood by qualitatively contrasting them, with the information discerned from the empirical study conducted on the MFI insect model [2, 18]. During the mentioned empirical study, it was demonstrated that an appropriately parameterized wing's rotational trajectory (the parameters being frequency, amplitude, and phase) can produce thrust in the wing motion plane that can counter air damping and drag on the insect's body, which in turn leads to proportional motion of the robot in forward direction. Additionally, it has been deduced that an appropriate and steady stroke trajectory envelope in the wing kinematics, at positive rotational position and rate of the wing (i.e., upstroke of the wing), can produce positive lift in MFI (which would counter the gravitational forces and leads to rise in the altitude), and the same stroke envelope at negative rotational position and rate would generate antilift (which leads to drop in the altitude). Thus, any designed or evolved controllers for MFWR model should at least qualitatively satisfy this empirically deduced criterion, established for the MFI insect model flight behavior.

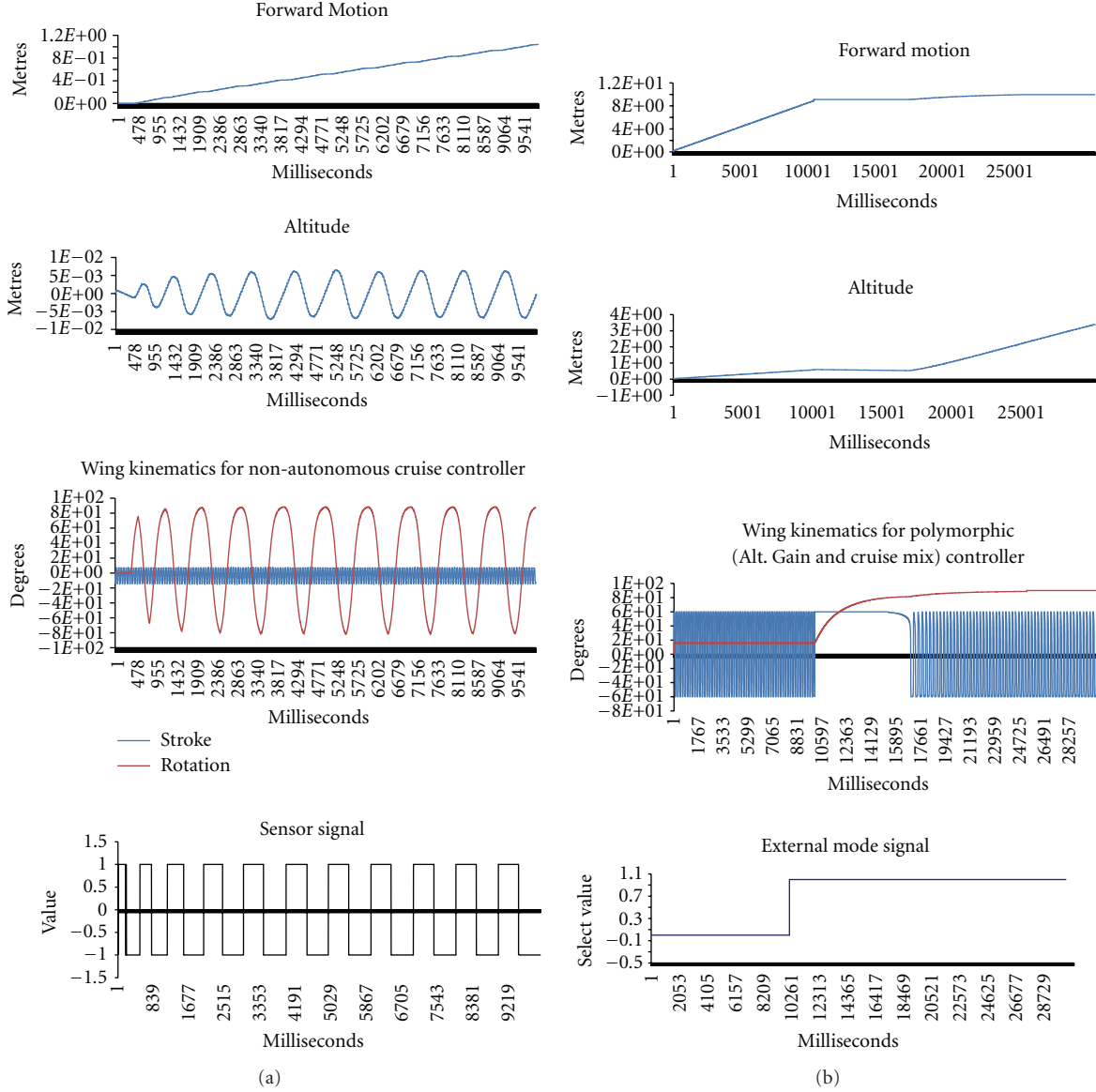


FIGURE 7: (a) shows the Wing Kinematics generated by the Central Core network of a Non-Autonomous Cruising Controller and the corresponding insect motion and sensory update (the stroke kinematics relatively has higher beats rate than that in rotation for the above controller). (b) shows the Wing Kinematics generated by the Central Core network of a Polymorphic flight Controller and the corresponding insect motion, when acted upon by external sensory inputs. One can see the initial cruising behavior is been switched to Alt. Gain behavior (with brief switching delay in wing kinematics).

It was demonstrated that the autonomous and nonautonomous controllers, evolved merely based on simple fitness evaluation functions, produced an acceptable physical behavior in the MFWR model in terms of overall body trajectory [8, 9]. Furthermore, when diligently observed, the wing trajectory (rotational and stroke) produced by the evolved flight controllers (for a given fitness criteria) seems to abide with the empirically established wing trajectory criterion (from MFI insect model flight behavior study). A detailed description of acceptability analysis is not in the scope of this paper and readers are directed to [8, 9] for more details on this analysis performed on individual flight mode controllers.

4.2. Qualitative Functional Decomposition Analysis. It would be of interest to interpret the evolved controllers by possible decomposition into easily explainable logical control blocks, and there exists a previous work [5, 16, 19] that relies on identifying the internal dynamics of the CTRNN-EH controller into Central Pattern Generators (CPGs) and Reflexive Pattern Generators (RPGs). To summarize succinctly, one can perceive the CPG patterned CTRNN controller as the collection of neuron modules that have been evolved appropriately at individual neuron level, to produce autonomous oscillatory dynamics, without any external oscillations or bias. The possibility of a two-neuron CTRNN-EH controller producing autonomous oscillatory dynamics has been

demonstrated in [16, 19], and further the later work in the same realm [5, 6] provided a logical CPG template, in which neurons inhibit each other with a time delay, which further leads to continuously destabilizing each other to generate oscillatory dynamics. Further, an RPG patterned CTRNN-EH controllers can be perceived as the collection of neuron modules that have been evolved appropriately to produce oscillatory dynamics in presence of the external oscillations or bias.

The evolved autonomous altitude gain, cruising, and steering and controllers are suspected to fall under the CPG template and could be decomposed into a collection of explainable oscillatory and nonoscillatory neuron groups that produced desired control of the evolved flight behaviors.

On other hand, the nonautonomous cruising mode controllers and polymorphic mode controller (as a whole) are likely to fall under the RPG template and could be decomposed into a collection of sensor-dependent or -independent oscillatory neuron groups. Thus, it would be necessary to find and separate the possible independent and dependent oscillatory control modules in an evolved controller that could aid in characterizing a given controller using known CPG or RPG templates. Further this decomposition process could provide a qualitative view and human understandable structure of the lower-level coordination among these separated modules, which primarily govern the behavior of a given evolved controller. In this vein, a three-step frequency-based analysis procedure is proposed to qualitatively decompose the evolved controllers.

Dynamics-Deprived Neuron Elimination. To simplify the process of decomposing, the evolved controllers into a group of functional units, a step-by-step neuron elimination technique, shown in Figure 8, has been employed to possibly reduce the size of the existing 8-neuron CTRNN-EH controllers. As shown in Figure 8, one can assign a role to individual neurons in a given CTRNN-EH controller architecture, based on their functional value. The neurons that are connected directly to the effectors module of the MFWR can be designated as primary neurons, and others can be designated as secondary neurons. It is obvious that primary neurons cannot be dynamics-deprived neurons, but some of the secondary neurons that saturate to minimum or maximum of neuron output level during flight controller period qualify to be dynamics-deprived neurons. These detected dynamics-deprived neurons can be folded into the existing neurons by modifying the biases appropriately (i.e., “Bias-Forwarding”). Once the reduced architecture’s dynamics qualitatively match the dynamics of the original complete network, the reduce network can be employed for further decomposition process.

Frequency-Based Grouping. Based on the previously mentioned general principle of acceptable controller dynamics, it was deduced that the steady oscillatory dynamics in the wing (stroke or rotation) dictate the flight behavior. Thus, based on this controller acceptability knowledge, it would be appropriate to group the neurons in the reduced network,

based on their individual time constants, into no more than two groups (one each for rotation and stroke). As shown in the Figure 9, the clustering criteria are based on the idea that the neurons with relatively lower time constants (i.e., higher frequency) are separated from the neurons with relatively higher time constants (i.e., lower frequency). As shown in Figure 9, the grouping of the neurons will simplify the decomposition process in the way that one can logically relate the individual wing kinematics (stroke or rotation) to the individual neuron groups (clustered) based on the qualitative difference in the frequency and phase of the wing kinematics.

Lesion Study. Once the frequency clustered neuron control modules are obtained for a given controller, it is necessary to understand the interactions of the individual neurons within those control modules and with the other existing control modules to qualitatively deduce the underlying governing principle of the controller functionality. Thus, in this lesion study, a general method of diligently observing the variations in the dynamics of an individual or group of neurons, while some of its connections are amputated from rest of the network, has been adopted. Though the number of lesion operations cannot be quantified and will vary depending on the complexity of the evolved controller, but as shown in Figure 10, the initial intuitive regions of the amputations across all the evolved controllers would be between the frequency clustered neuron groups to verify their interdependency followed by a series of further lesions, like intragroup lesion study, wherever deemed appropriate for the controller in the context. Employing the above-mentioned qualitative decomposition process; the best five of every evolved controller in each category have been analyzed as discussed below.

4.3. Analysis of Autonomous Controllers

4.3.1. Autonomous Cruise Mode Controllers. This section provides the detailed qualitative decomposition process for one of the best evolved autonomous cruise mode controllers, using the above-mentioned three general steps. For qualitative comparisons and to better understand the controller decomposition an unaltered original eight-neuron architecture of the controller to be analyzed is shown in Figure 11. The architecture has two primary neurons 0 and 1, which are directly connected to the stroke and rotation effectors of the MFWR and the neurons from 2 to 7 are the secondary neurons of the controller, whose role in governing the controller behavior would be determined as part of this decomposition process. The original outputs of each neuron in the controller’s architecture, which are responsible for producing the desired cruising behavior in MFWR, are shown in Figure 12.

Moreover, the flight trajectory of the MFWR under the control of the original controller in the context is shown in Figure 13, which would be useful for qualitative comparisons that would be performed later when the original architecture of the controller has been simplified for analysis. As shown in Figure 12, it can be observed that the secondary neurons

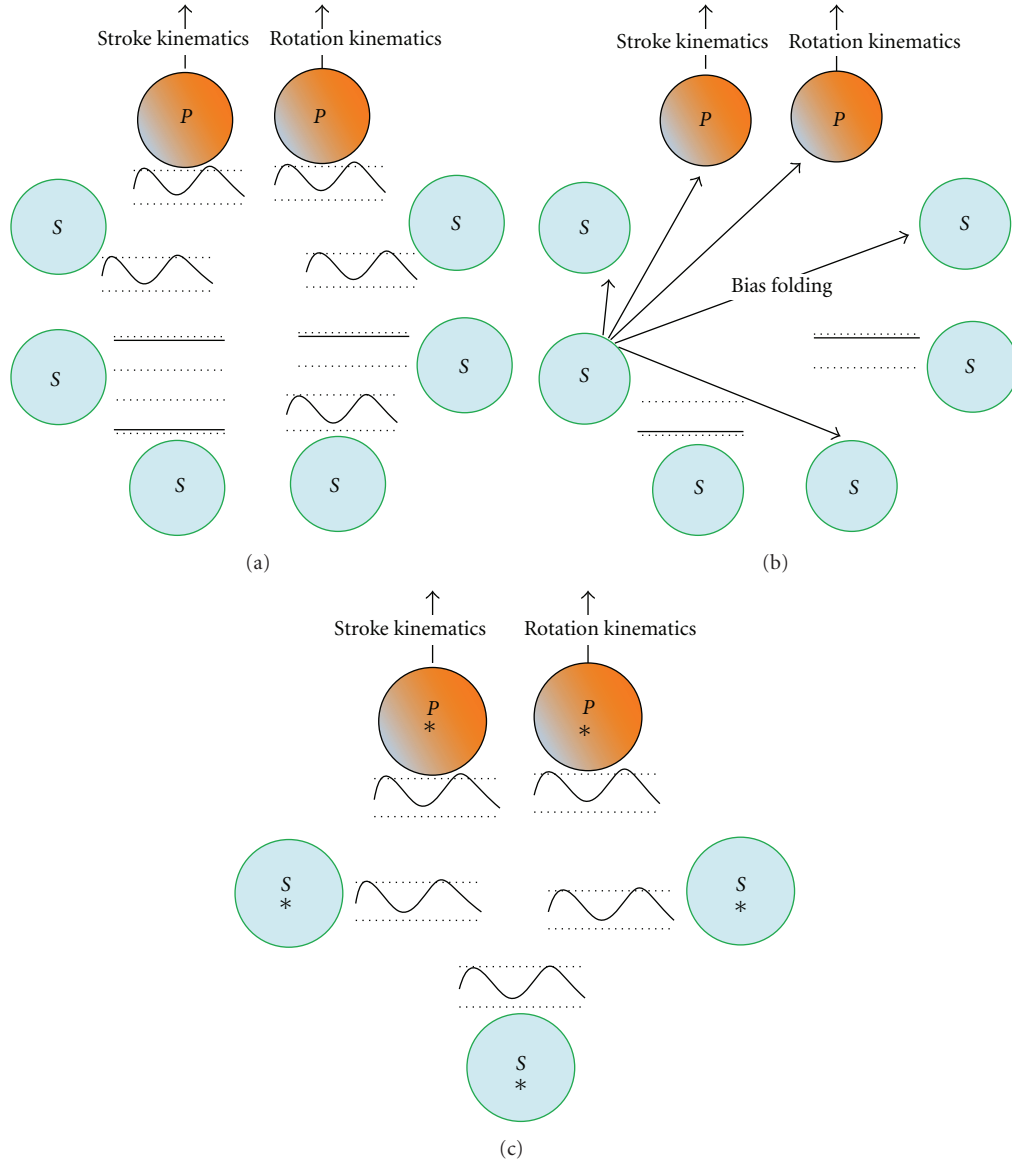


FIGURE 8: A pictorial representation of the “dynamics-deprived neuron elimination” process. The primary neurons are labeled as “ P ” and subsidiary neurons are labeled as “ S ”. A three step process is adopted here, starting with eliminating the neurons with saturated dynamics in them as shown in (a), followed by folding the saturated output of the eliminated neurons as bias into the survival neurons as shown in (b). The final step shown in (c) is to verify the qualitative match of the dynamics produced by the bias modified (labeled with “ $*$ ”) individual neurons to their counterparts in the original architecture.

3, 5, and 6 seem to be saturated at constant output value during the flight control. Though it can be deduced, at least, from observations that these three neurons may not have contributed to the overall output dynamics produced by the controller, a detailed step-by-step process mentioned in the “Dynamics-deprived Neuron Elimination” procedure is necessary to rule out the possibility that these neurons might have played a critical role during the initialization of the controller by providing transient dynamics before saturating in the steady state. The obvious neurons that are contributing to the controller dynamics are 0, 1, 2, 4, and 7, but there exist distinct differences in the output

envelope and frequency characteristics of 0 and 1 neurons from 2, 4, and 7, which could be used for “frequency-based grouping” process later on the successful reduction of the architecture size, as shown in Figure 17. Since, the candidate dynamics-deprived neurons are determined by the neuron output state observations; the biases of the neurons 0, 1, 2, 4, and 7 are modified appropriately, as pictorially represented in Figure 8(b), by treating the individual input weight of the survival neuron from each eliminated neuron as an additional bias value to its output state. The resultant reduced neuron architecture of five neurons in it is pictorially represented in Figure 14. To further validate

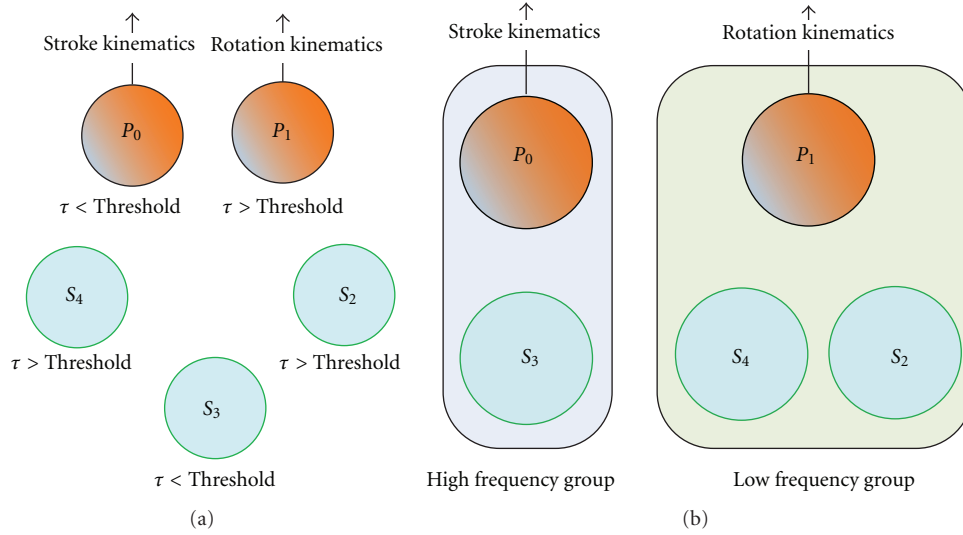


FIGURE 9: A pictorial representation of the “frequency-based grouping” process. The first step of the process, as shown in (a), is to determine a relative threshold time constant (τ) for the reduced network, reduced by “dynamics-deprived neuron elimination” process, followed by grouping the neurons in the architecture based on the frequency of the output produced by individual neuron (i.e., the neurons with time constant less than the relative threshold are clustered into high frequency group, and neurons with time constants more than the relative threshold are clustered into low frequency group) as shown in (b).

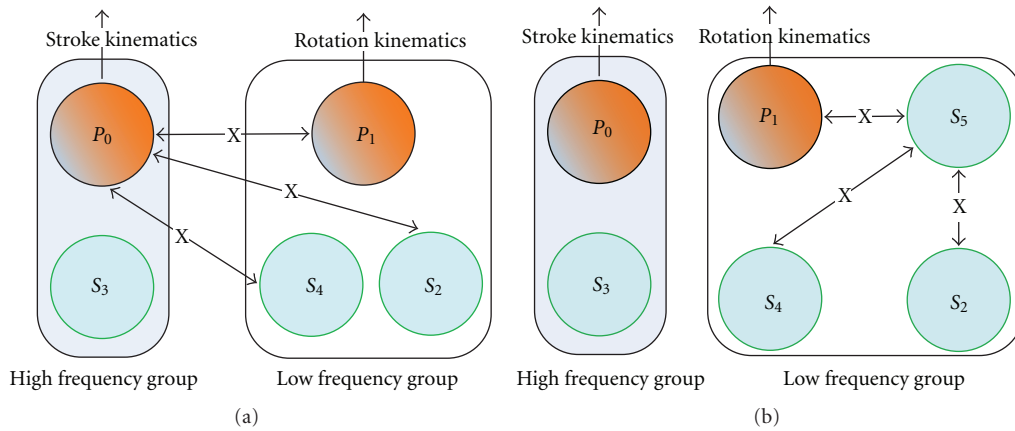


FIGURE 10: A pictorial representation of the “lesion study” process. The lesion study is based on the idea that it is possible to determine the underlying governing functional principle of the network with rigorously observing the behavior changes in the network for appropriate combinations of the amputations. Based on the complexity of the controller, the lesion study can be performed between neurons in distinct frequency groups, which is performing intergroup amputations, shown in (a), or between the neurons in the same frequency group, that is, intragroup amputations shown in (b).

that the dynamics-deprived neuron elimination process is applicable for this controller, two qualitative comparisons are necessary, primarily the architecturally reduced controller should at least qualitatively control the MFWR trajectory behavior that was intended by the original controller, and, moreover, the survival neuron output state envelopes during the flight control should match their output state envelopes from the original architecture.

The later condition eliminates the possibility that the reduced architecture could have changed dramatically and lost its internal dynamics, although it could have satisfied the primary condition to produce the desired cruise behavior in MFWR. Thus, the reduced five neuron controller is evaluated

against the MFWR, and the individual neuron output state envelope of the five-neurons is captured and shown in Figure 15, and accordingly the trajectory of MFWR under the control of the reduced controller is shown in Figure 16. It can be observed that there exists an acceptable qualitative match between the produced neuron outputs in the reduced five-neuron controller to its counterparts original eight neuron controller, including the in sync variations of the frequency and amplitude in the primary stroke neuron and the neuron 4 (of original architecture) to its new neuron position 3.

Moreover, the most convincing evidence that the reduced controller qualitatively controls the MFWR trajectory to

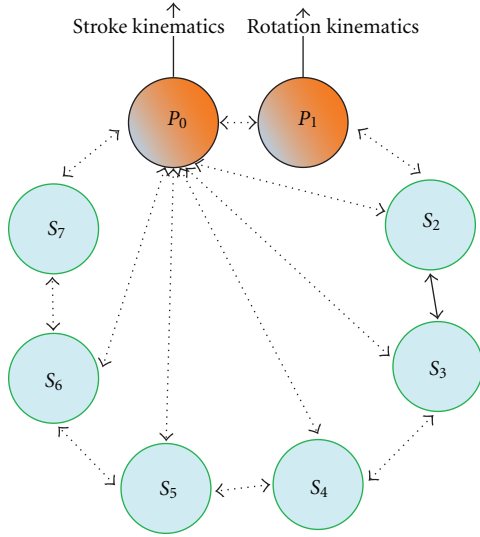


FIGURE 11: A pictorial representation of the fully connected eight neuron architecture of the autonomous cruise mode controller chosen for qualitative decomposition analysis. As mentioned earlier, the stroke and rotation neurons are marked “P” as primary and “S” as secondary for the other neurons and numbered accordingly from 0 to 7.

produce desired cruise behavior justifies that the dynamics-deprived neuron elimination process is applicable for this controller. Thus, moving forward with the reduced five-neuron architecture, applying frequency-based grouping would be uncomplicated, since it can be observed from the five neuron output envelopes that the primary stroke neuron and third secondary neuron seem to share a peculiar in sync frequency and amplitude variations, intuitively belonging to high frequency group. Moreover, the evolved time constant for both of these neurons is same and is 0.010000 units and on other hand, the neurons 1 and 4 along with the rotation primary neuron can be allocated to low frequency group with corresponding time constants 10.546157, 9.558393, and 20.176863, respectively. Thus, if a relative time constant threshold of 9 units is chosen, then there exist two distinct frequency-based groups as shown in Figure 18. After the grouping of the primary stroke neuron and third secondary neuron in a comparable frequency group, further interpretation on their interconnection weight revealed that there strongly inhibit each other, and further there exists a strong possibility that these two neurons can form a two-neuron (high frequency) oscillator with any other input dynamics from the low frequency group (consists of rotation primary neuron, fourth secondary neuron, and second secondary neuron). Thus, an intergroup lesion study, as shown in Figure 10(a), to amputate the neuron connections between the high frequency group neurons and the low frequency group neurons is performed. When this amputated network is evaluated, the above intuitive possibility of two-neuron oscillator formation in the high frequency group was validated along with a revelation of two-neuron oscillator formation in a low frequency group, as shown in Figure 18. It can be observed that the primary stroke neuron and the third

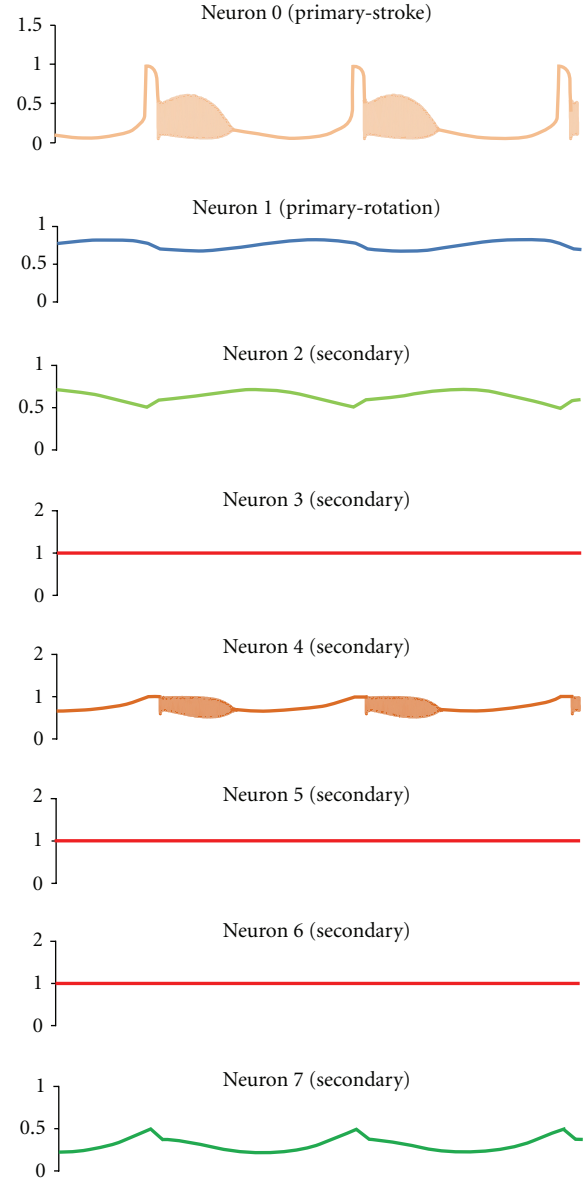


FIGURE 12: The above figure shows the neuron output state dynamics of each neuron in the fully connected original eight neuron controllers produced during the flight control of the MFWR to provide optimal cruise behavior.

secondary neurons oscillate at same frequency consistently, and their output amplitude is more than the lower frequency group consisting of primary rotation neuron and second secondary neuron along with a saturated fourth secondary neuron during the amputated evaluation. It is evident that the fourth secondary neuron's dynamics are not completely isolated from the high frequency stroke oscillator group (primary stroke neuron and third secondary neuron) since this fourth neuron has shown perfect oscillatory behavior when the reduced network was fully connected through this neuron. Moreover, when this amputated network controller is coupled to the MFWR, it has been observed that the controller was able to control the MFWR trajectory in an

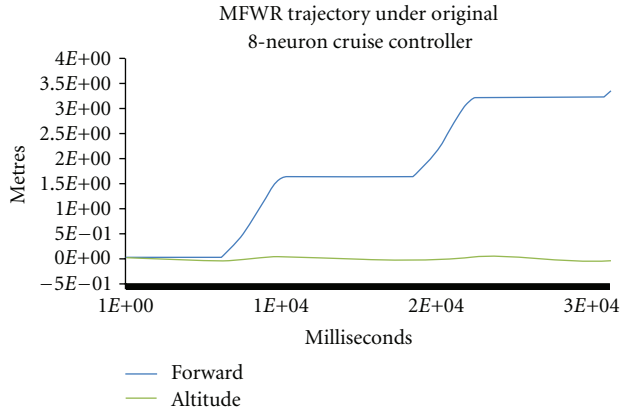


FIGURE 13: The MFWR trajectory produced by the original fully connected eight-neuron controller. It can be observed that the evolved controller was successful in producing forward motion in the MFWR without any overall gain in the altitude.

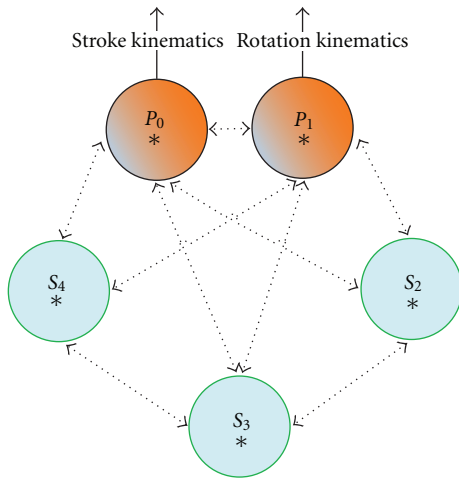


FIGURE 14: A pictorial representation of the reduced five-neuron architecture of the cruise controller referred in Figure 12. It should be noticed that the primary neurons and secondary neurons retain their position in the network, but neurons 4 and 7 from the original network are positioned in 3rd and 4th locations respectively. The “*” indicates that the neurons in this “dynamics-deprived neuron elimination” process-based architecture differ from the original neuron in the way that, their bias has been accounted for the eliminated neuron’s saturated output effect on them. So, at least in steady state this reduced network should perform functionally equivalent to the original architecture.

acceptable cruise mode behavior template seen before in Figures 16 and 15, but gradually it drifted from the acceptable behavior and resulted in significant rise in MFWR’s altitude, with a rise rate proportional to the MFWR forward motion rate, as shown in Figure 19. It would of interest to do a diligent comparison of the dynamics of the fully connected reduced five-neuron controller to that of the amputated controller shown in Figures 18 and 15 from behavior change perspective in each neuron output states during the flight evaluations. Though one might argue that quantitatively the

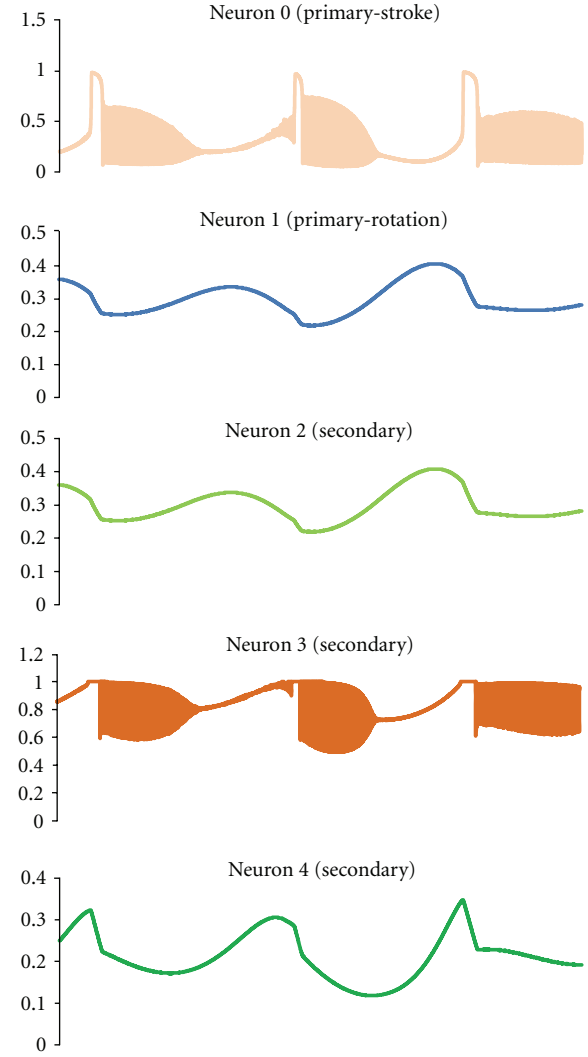


FIGURE 15: The above figure shows the neuron output state dynamics of each neuron in the reduced five-neuron architecture of the cruise controller architecture shown in Figure 12. It can be noticed that qualitatively the output dynamics of each neuron do not differ significantly from their original behavior shown in Figure 13.

dynamics of neurons 0 to 3 (includes primary stroke and rotation neurons and two secondary neurons) are different in both the scenarios, for qualitative analysis purposes these neurons do project similar oscillatory dynamics, but the drastic difference of the dynamics is observed in the fourth secondary neuron, which seized to oscillate when amputated from the high frequency neuron group indicating a strong connection to the controller’s performance degradation noticed in the Figure 24. When analyzing the neuron outputs of the fully connected controller, the dynamics of stroke neuron group (high frequency group) were altered periodically by a slow moving signal, with a period equivalent to the rotation neuron group (low frequency group). Moreover, it was already demonstrated that there exists the fourth secondary neuron in this low frequency group (rotation)

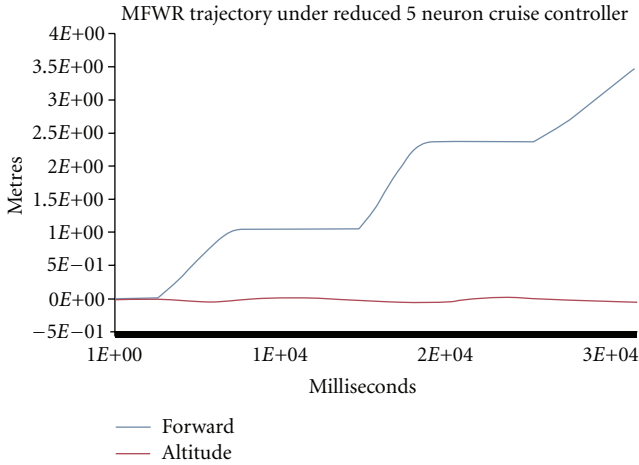


FIGURE 16: The MFWR trajectory produced by the reduced five neuron controller. It can be observed that the architectural reduced controller was successful in producing qualitatively same cruise behavior possible by the fully connected eight neuron network. The MFWR trajectory produced by the eight-neuron network is shown in Figure 14.

that is susceptible and depends on the dynamics of the high frequency group (stroke), and though both groups were capable of producing independent oscillatory dynamics to control the MFWR trajectory, the control lasted for a short period of time in absence of the possible dynamics modification by the fourth secondary neuron. Further, an optimal control behavior was only possible with inclusion of this fourth secondary neuron, which now can be treated as a monitoring neuron, that was evolved appropriately to take the responsibility of performing complex dynamics computation across both the oscillator groups and provide the high frequency group with periodic signals to alter its amplitude and frequency to satisfy the cruise behavior in the MFWR. Additionally, it was mentioned earlier that the fourth neuron has an intermediate time constant value of 9.558, which makes it have enough temporal summation ability, the ability which could have possibly made it an observer (sink in the dynamics of the other neurons) of the other neurons and further have sufficient internal dynamics (sufficient firing rate to generate spikes) to modify their dynamics at slower but in a strong way thus modifying their frequency and amplitude periodically.

Based on the above analysis, it can be deduced that the evolved autonomous cruise mode controllers can be qualitatively explained as a composition of two steady and independent frequency oscillators, one governing the stroke kinematics of the wing with higher beat rate and another it is rotation with lower beat rate, in presence of a monitoring neuron which periodically tunes the amplitude and frequency of the stroke oscillator, which periodicity synchronized with the rotation oscillator. A pictorial representation of the above deduced compositional template is shown Figure 20. All of the 5 best evolved autonomous cruise mode controllers were reducible from an eight neuron to a five-neuron architecture using the “dynamics-deprived neuron elimination” process.

Further, four of them had distinct frequency features in their architecture that could be exploited by the “Frequency-based Grouping” process and were successfully reduced to two kinematics control modules. Only three of the best five controllers complied with the decomposed template discussed above, with steady independent oscillator blocks and a monitoring neuron, and others performed the same functionality with closely dependent oscillator blocks that were not complaint with frequency-based clustering criteria. Nonetheless, the rigorous intragroup lesion study on them exhibited the presence of monitor neuron, which aided in controlling the amplitude of the rotation dynamics for acceptable cruise behavior.

4.3.2. Autonomous Altitude Gain Mode and Steer Mode Controllers. The above decomposition analysis mentioned in the context of the cruise mode controllers is performed on the entire best five autonomous altitude gain mode and steer mode controllers. The individual controller architectures were reducible from an 8-neuron to 4-neuron architecture using “Dynamics-deprived Neuron Elimination” process in both categories. Only some of the best altitude gain controllers were complaint with clustering criteria and thus two functional templates were derived using the lesion study performed on the individual neurons in the reduced network. As shown in Figure 21(a), this derived functional template employed single oscillatory control group encompassing both the primary neurons in it, performing close-looped oscillations required for wing kinematics, with aid of two subsidiary neurons in the network resembling a typical CPG-like control module described earlier in the section. This decomposition template is applicable for the steer controller’s entire central core and only for two of the altitude gain controllers. The other template, shown in Figure 21(b), had only the stroke primary neuron in an oscillatory control group with a saturated rotation kinematics in separate control module, which is applicable for only altitude gain controllers.

4.4. Analysis of Nonautonomous Controllers

4.4.1. Nonautonomous Cruise Mode Controllers. This section provides the detailed qualitative decomposition process for one of the best evolved non-autonomous cruise mode controllers. The applicability of the established three-step decomposition using Dynamics-Deprived Neuron Elimination, Frequency-based grouping, and Lesion Study methods will be presented and possible oscillatory level decomposition will be deduced. For qualitative comparisons and to better understand the controller decomposition, an unaltered original eight-neuron architecture of the controller to be analyzed is shown in Figure 22. The architecture has two primary neurons 0 and 1, which are directly connected to the stroke and rotation effectors of the MFWR and the neurons from 2 to 7 are the secondary neurons of the controller, whose role in governing the controller behavior would be determined as part of this decomposition process. Apart from the interconnections among the neurons, every

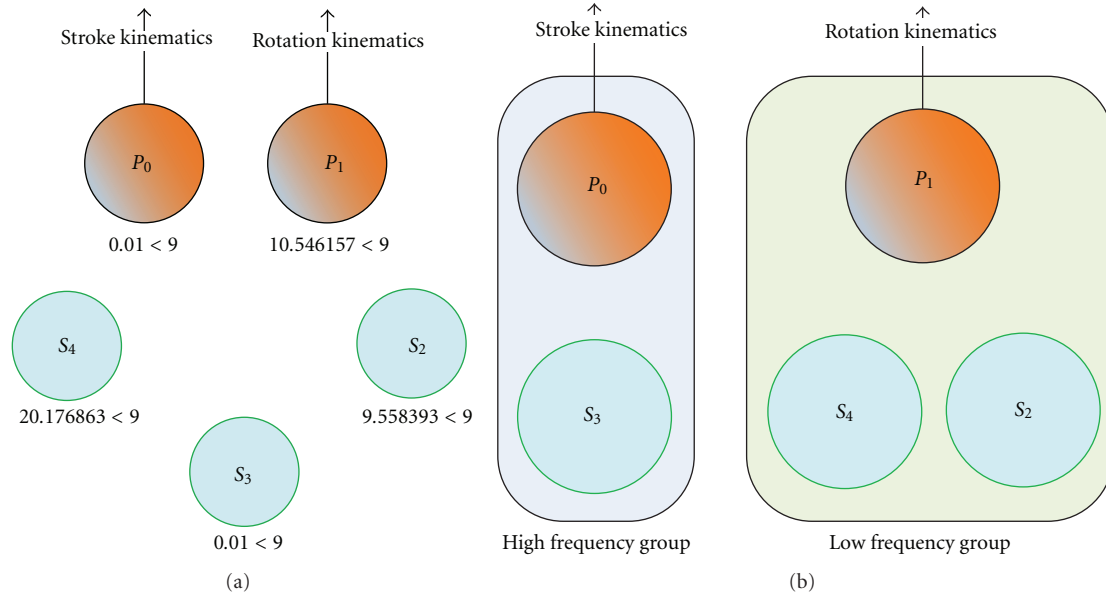


FIGURE 17: A pictorial representation of the “frequency-based grouping” process for the cruise mode controller shown in Figure 12. The first step of the process, as shown in (a), is to compare each neuron’s time constant with determined relative threshold of 9.0 units, followed by grouping the neurons in the architecture based on the frequency of the output produced by individual neuron (i.e., the neurons with time constant less than the relative threshold are clustered into high frequency group and neurons with time constants more than the relative threshold are clustered into low frequency group) as shown in (b).

neuron is connected to an external altitude sensor, which is modeled to provide a relative altitude status of the MFWR from its initial altitude during the evaluation. The original outputs of each neuron in the controller’s architecture, which are responsible for producing the desired cruising behavior in MFWR, along with the altitude sensor output, are shown in Figure 23. Moreover, the flight trajectory of the MFWR under the control of the original controller in the context is shown in Figure 24, which would be useful for qualitative comparisons that would be performed later when the original architecture of the controller has been simplified for analysis. As shown in the Figure 23, it can be observed that the secondary neurons 3 and 6 seem to be saturated at constant output value during the flight control. Though it can be deduced, at least, from observations that these three neurons, may not have contributed to the overall output dynamics produced by the controller, a detailed step-by-step process mentioned in the “dynamics-deprived neuron elimination” procedure is necessary to rule out the possibility that these neurons might have played a critical role during the initialization of the controller by providing transient dynamics before saturating in the steady state. The obvious neurons that are contributing to the controller dynamics are 0, 1, 2, 4, 5, and 7, but there exist distinct differences in the output envelope and frequency characteristics of 0, 2, 5, and 7 neurons from 1 and 4, which could be used for “frequency-based grouping” process later on the successful reduction of the architecture size. Moreover, as mentioned during the initial physical validation step of the evolution process, there exists an entrainment of the primary rotation neuron output state in amplitude and frequency with that of

the sensor status output characteristics. Since the candidate dynamics-deprived neurons are determined by the neuron output state observations, the biases of the neurons 0, 1, 2, 4, 5, and 7 are modified appropriately, by treating the individual input weight of the survival neuron from each eliminated neuron as an additional bias value to its output state. The resultant reduced neuron architecture of six neurons in it is pictorially represented in Figure 25. To further validate that the dynamics-deprived neuron elimination process is applicable for this controller, two qualitative comparisons are necessary; primarily the architecturally reduced controller should at least qualitatively control the MFWR trajectory behavior that was intended by the original controller, and, moreover, the survival neuron output state envelopes during the flight control should match their output state envelopes from the original architecture. As mentioned earlier, the later condition eliminates the possibility that the reduced architecture could have changed dramatically and lost its internal dynamics, although it could have satisfied the primary condition to produce the desired cruise behavior in MFWR.

Further, the interesting entrainment behavior between the sensor output and the rotation neuron output (and if possible the third (old designated position-fourth) secondary neuron output) should be maintained, at least qualitatively. Thus, the reduced six-neuron controller is evaluated against the MFWR, and the individual neuron output state envelope of the six neurons and the sensor status are captured and shown in Figure 26, and accordingly the trajectory of MFWR under the control of the reduced controller is shown in Figure 27. It can be observed that there exists an acceptable qualitative match between the produced neuron outputs in

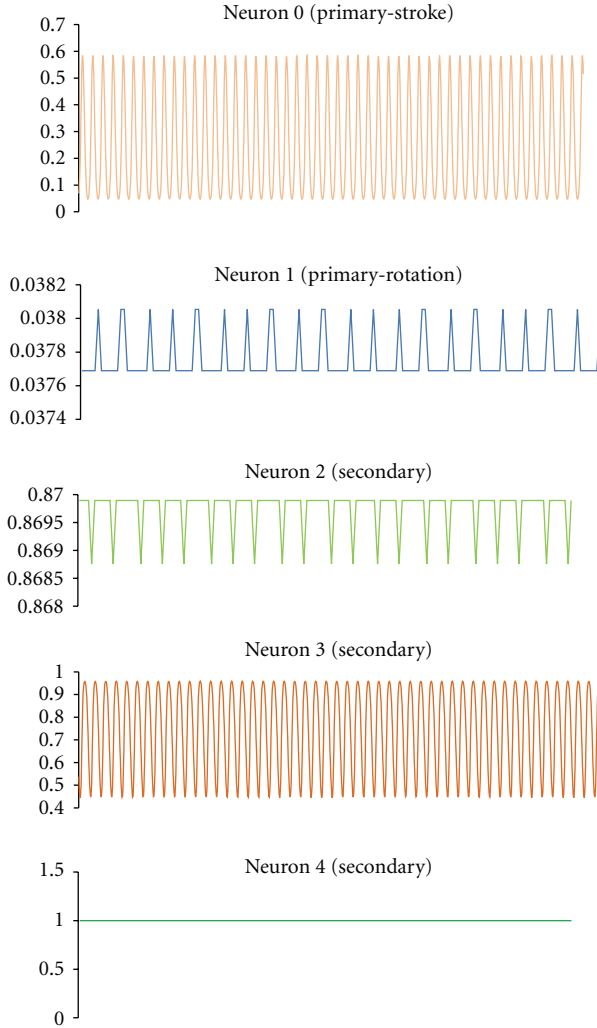


FIGURE 18: The output state dynamics of each neuron in the interfrequency group amputated network, amputated as part of the lesion study on the reduced five-neuron network. It can be observed that the primary stroke neuron and the third secondary neuron produced perfect in sync oscillations forming a two-neuron independent oscillator. On the other hand, the primary rotation neuron with second neuron formed a feeble two neuron oscillator. It should be noticed that the fourth neuron dynamics are saturated, in the amputated network, compared to its original oscillatory behavior seen in Figure 16 of the fully connected reduced controller.

the reduced six-neuron controller to its counterparts in the original eight-neuron controller, including the in entrainment behavior between the primary stroke neuron and the sensor status.

Thus, moving forward with the reduced six-neuron architecture, applying frequency-based grouping would be complicated, since it can be observed from the six-neuron output envelopes that the primary stroke neuron, along with second, fourth, and fifth secondary neurons, seems to share the same frequency bandwidth, intuitively belonging to high frequency group.

Moreover, the evolved time constants for these neurons are in the range of 0.010000 to 0.050000 units. But, on the

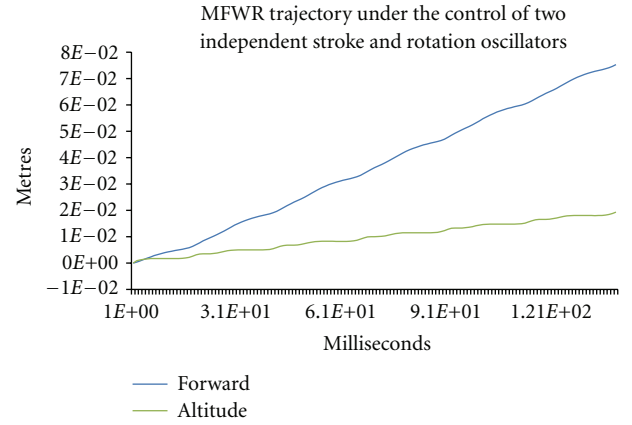


FIGURE 19: The trajectory of MFWR produced under the control of the amputated cruise controller. It can be noticed that the controller, with two independent oscillators for stroke, and rotation produces an acceptable cruise behavior during initial phases of the flight, but immediately loose its ability to control and reduce the altitude variations, in absence of the monitor neuron.

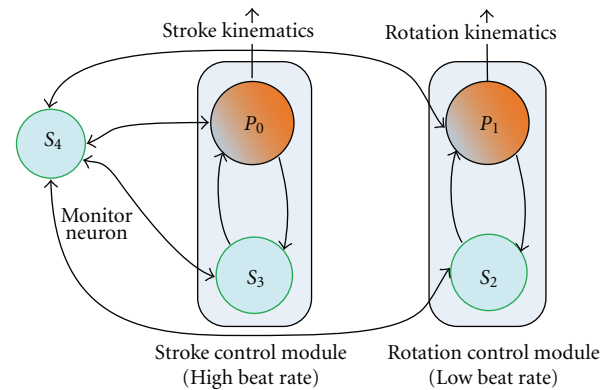


FIGURE 20: The Qualitative functional decomposition template derived for the autonomous cruise mode. Most of the autonomous cruise mode controllers can be decomposed into the above-shown template with a high frequency stroke control oscillator module and a low frequency rotation control oscillator along with an intermediate neuron called monitor neuron, which is responsible to coordinate and fine-tune the amplitude and frequency of the stroke oscillator with a period derived from the rotation oscillator. This functional template explains the general evolved behavior of the amplitude and frequency modulation of the stroke kinematics with rotation period for optimal cruise control of MFWR.

other hand, the rotation primary neuron and the third secondary neuron can be allocated to low frequency group with corresponding time range of 10.034 to 16.532 units. Moreover, since the sensor module output can be treated as a pseudoneuron (with dynamics equivalent to the MFWR model and with interneuron connections to the primary neurons only), there exist two options to decompose the architecture further. The first approach is to group the sensor pseudoneuron into the low frequency group and perform the intergroup lesion study, which will provide the insight into the high frequency group oscillator's (if at all the group

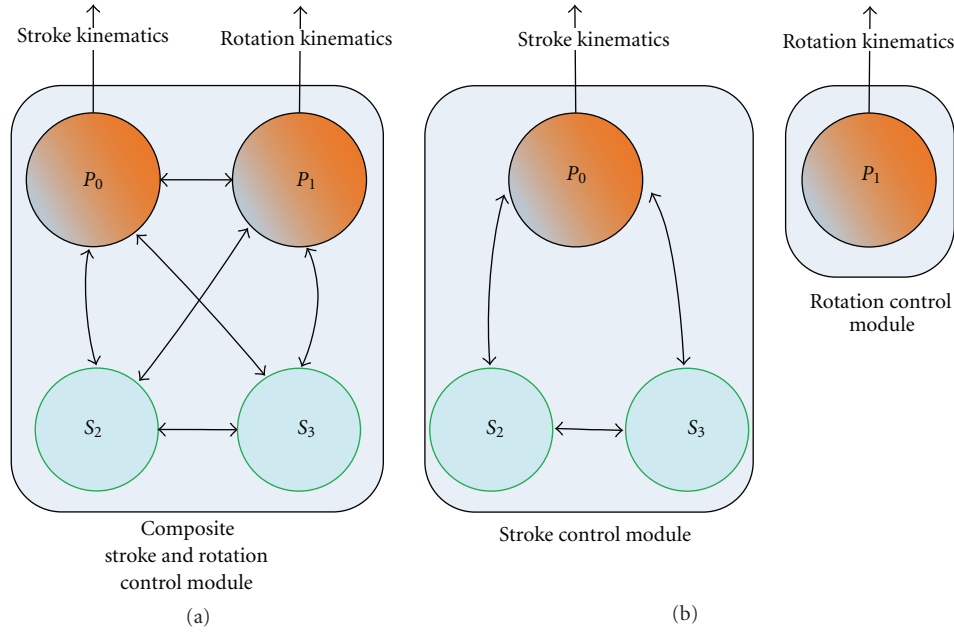


FIGURE 21: The qualitative functional decomposition derived for the evolved autonomous altitude gain controllers and steer controllers. Most of the steer controller's wing kinematics can be decomposed with a typical CPG-like functional template shown in (a) as a closely coupled stroke and rotation oscillators with steady beat rate and steady amplitude. Most of the altitude gain controllers can be decomposed with the functional template shown in (b), with a dedicated stroke oscillator along with a saturated rotation control module.

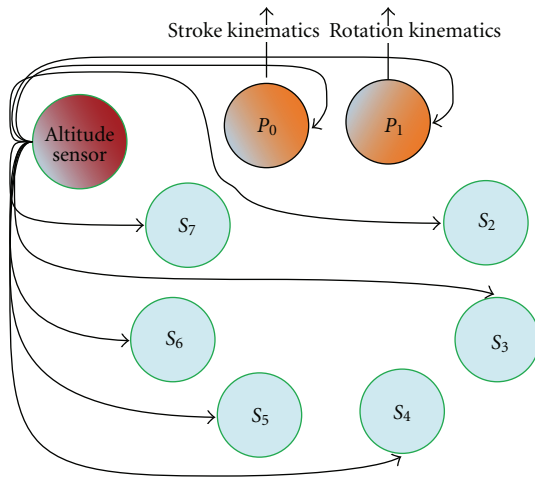


FIGURE 22: A pictorial representation of the fully connected eight-neuron architecture with a single altitude sensor, of the nonautonomous cruise mode controller chosen for qualitative decomposition analysis. Following the general neuron representation, the stroke and rotation neurons are marked with "P" as primary and "S" as secondary for the other neurons and numbered accordingly from 0 to 7.

exhibits independent oscillatory nature) dependency on the sensor state and further the same dependency can be derived by performing intragroup lesion study on the low frequency group by amputating the sensor pseudoneuron. The second approach is to group only the real neurons by completely ignoring the sensor signal (i.e., amputating the sensor

signal) into a high and low frequency groups and study their behavior independently, checking for independent oscillatory behavior, in the absence of the external sensor signal, followed by introducing the sensor signal to detect any significant behavior changes for deducing any possible independent control modules. Though both approaches would yield the same conclusions, the second approach is chosen since the sensor dynamics of the MFWR can be treated separately from the actual neuron dynamics, in two easy steps of complete sensor-independent neuron dynamics decomposition (frequency grouping and intragroup lesion study) followed by the sensor status injection into the possible neuron-level decomposed modules. Thus, moving forward, the six-neuron architecture is disconnected from the external sensor and a frequency-based grouping, with groups mentioned earlier is performed as shown in the pictorial representation Figure 28. Further, an intergroup lesion study is performed, as described earlier (shown in Figure 10(a)), and the outputs of the each neuron in the two groups are presented in Figure 29. It can be noticed that these two frequency groups have indeed self-sufficient dynamics in them to be independent oscillators, with two frequency groups, the high-frequency group and the low-frequency group as demonstrated in above analysis. Moreover, the outputs of each neuron in the high-frequency group match their original output envelope from the reduced six-neuron architecture suggesting that this group's internal dynamics is immune to the external sensor dynamics. But the same cannot be deduced for the low-frequency group, which has high time constants and have already shown its affinity to entrain with the sensor signal. Moreover, this intragroup amputated

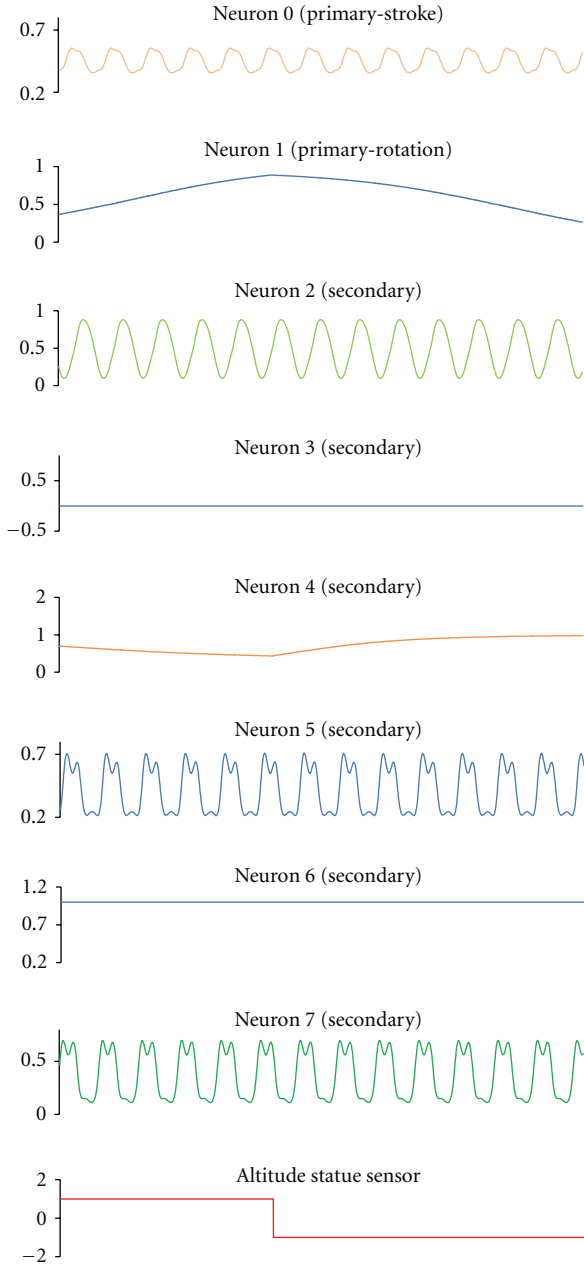


FIGURE 23: The above figure shows the neuron output state dynamics of each neuron in the fully connected original eight-neuron controller and the external altitude sensor, produced during the flight control of the MFWR to provide optimal cruise behavior. It can be observed that the output states of neurons 1 and 4 entrain with altitude sensor in phase and out of phase, respectively.

stroke and rotation neuron group independent oscillators have been partially successful in controlling the MFWR's expected cruise behavior as shown in Figure 30, in which it can be observed that the altitude of the MFWR is lost with the progression of the time, though the rate of the altitude drop is very much less than the rate of forward motion gain. The next step in this process to deduce any possible modular control structure is to inject the sensor signal dynamics

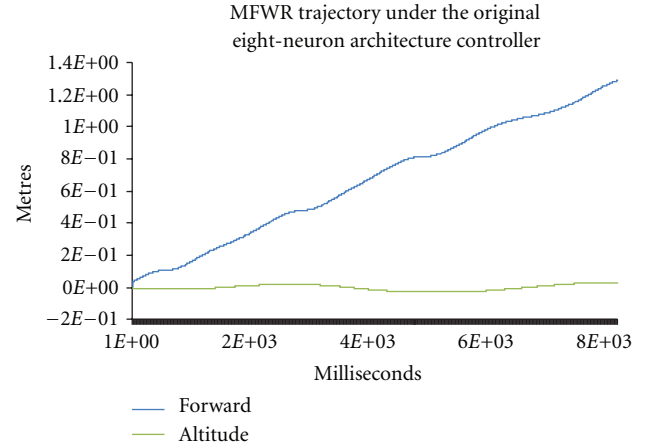


FIGURE 24: The MFWR trajectory produced by the original fully connected eight-neuron nonautonomous controller. It can be observed that the evolved controller was successful in producing forward motion in the MFWR without any overall gain in the altitude.

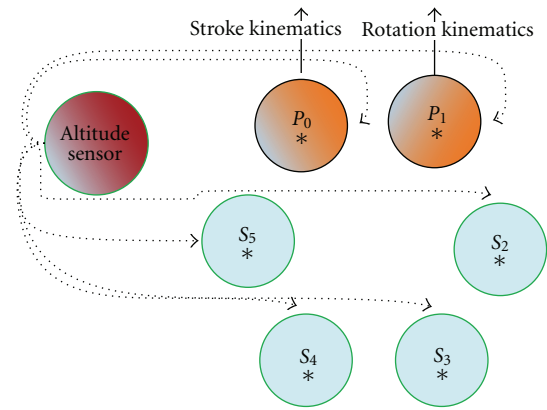


FIGURE 25: A pictorial representation of the reduced six-neuron architecture of the cruise controller referred in Figure 22. It should be noticed that the primary neurons and second secondary neuron retain their position in the network but the neurons 4, 5, and 7 from the original network are positioned in 3rd, 4th, and 5th locations respectively. The "*" indicates that the neurons in this "dynamics-deprived neuron elimination" process-based architecture differ from the original neuron in the way that, their bias has been accounted for the eliminated neuron's saturated output effect on them. So, at least in steady state, this reduced network should perform functionally equivalent to the original architecture.

into the established two frequency groups and check for the entrainment behavior and controller's expected cruise mode acceptability on MFWR. While performing the agreed final step in the decomposition process, it was deduced that injecting the sensor dynamics only into low-frequency group is sufficient to produce qualitatively acceptable cruise mode behavior in MFWR. Thus, a general qualitative functional decomposition template shown in Figure 31 is derived explaining the evolved non-autonomous cruise mode controllers, as a combination of two independent oscillators, of which the high-frequency oscillator controlled the stroke

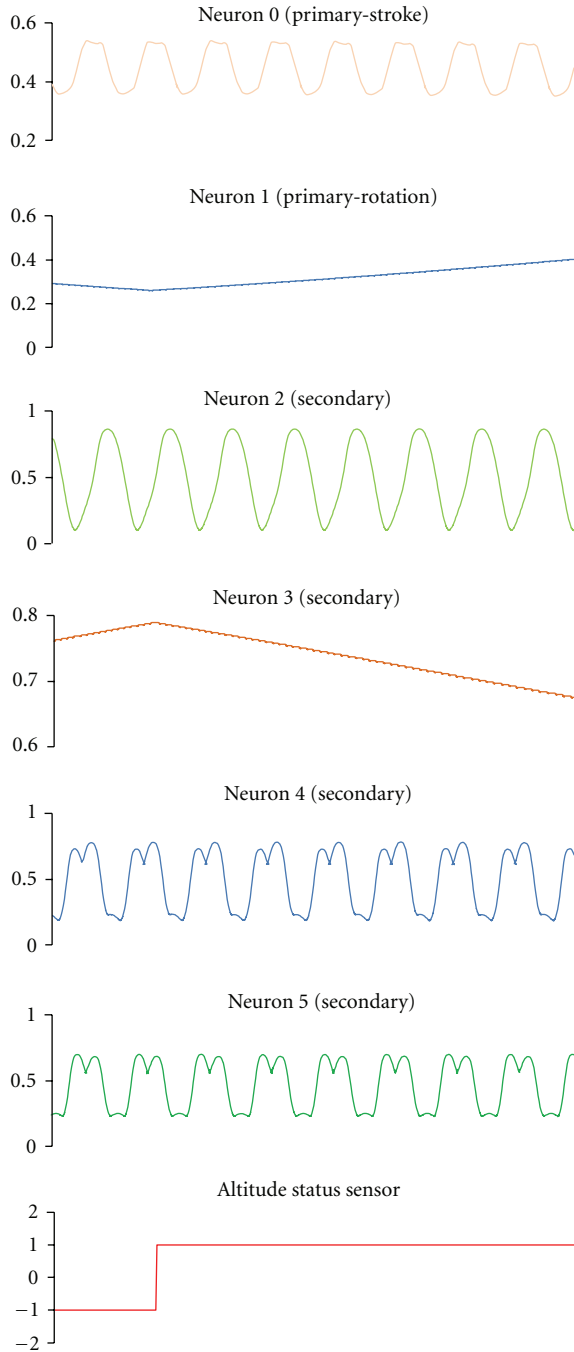


FIGURE 26: The above figure shows the neuron output state dynamics of each neuron in the reduced six neuron architecture of the cruise controller architecture shown in Figure 22. It can be noticed that the qualitative output dynamics of each neuron do not differ significantly from their original behavior shown in Figure 23. Moreover, It can be observed that the output states of neuron 1 and 3 entrain with altitude sensor in phase and out of phase, respectively.

kinematics of the wing with steady amplitude and frequency and the low-frequency oscillator, which was evolved to monitor the altitude variations in the MFWR, through the available external sensor module, altered the rotation dynamics continuously to limit the variations in the altitude of the MFWR and simultaneously provided the forward

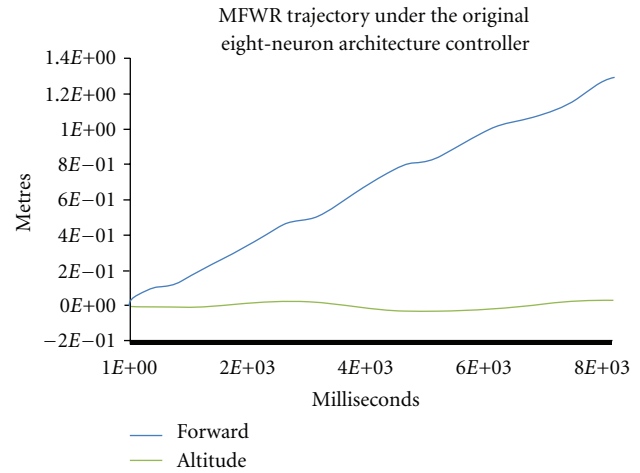


FIGURE 27: The MFWR trajectory produced by the reduced six-neuron controller. It can be observed that the architectural reduced controller was successful in producing qualitatively same cruise behavior possibly by the fully connected eight-neuron network. The MFWR trajectory produced by the eight-neuron network is shown in Figure 24.

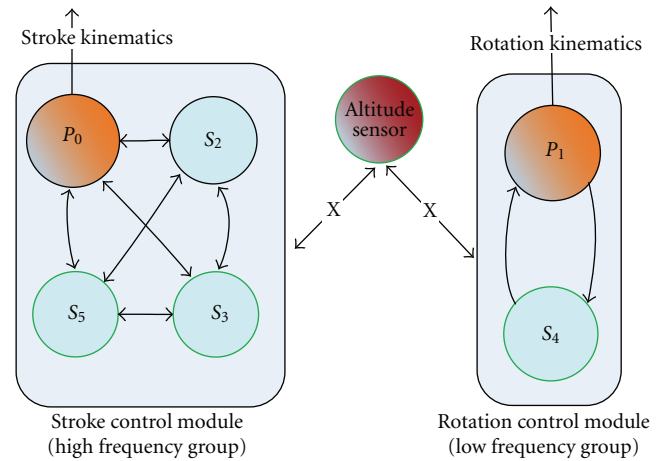


FIGURE 28: A pictorial representation of the "frequency-based grouping" process combined with intergroup lesion study in absence of the external sensor input for the nonautonomous cruise mode controller shown in Figure 22.

motion in it, by generating required lift and antilift with the behavior verified by the general principles of the empirical study (mentioned in the acceptability analysis). Three of the best five evolved controllers followed the deduced template, and the other two followed more closed template that is only different from the predominant one in that the stroke frequency group has a dependency on the external sensor status.

4.4.2. Nonautonomous Polymorphic Controllers. Since the polymorphic controllers embed in their architecture both the autonomous altitude gain and cruise mode controllers, which can be invoked as a separate controllers in isolation with a static external signal not a continuous dynamic signal, the qualitative functional decomposition templates

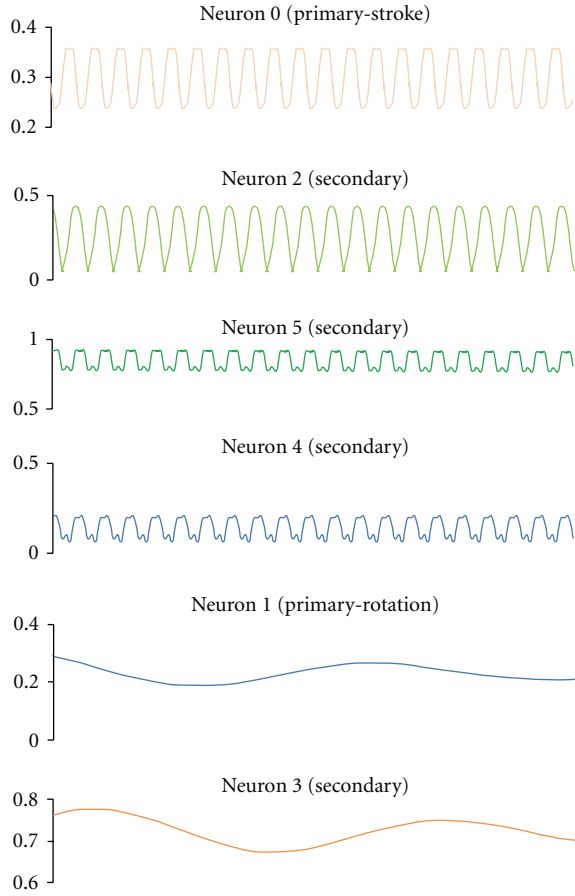


FIGURE 29: The above figure shows the neuron output state dynamics of each neuron in the low frequency group (1 and 3) and high frequency group (0, 2, 4, and 5) after intergroup amputation is performed and evaluated in absence of the external altitude sensor signal (represented in Figure 28). It can be noticed that each qualitative group is self-sufficient to generate internal dynamics to sustain steady oscillations independent of the external sensor signal. But, nonetheless low frequency group neurons seem to be susceptible to external sensor signal due to their high time constants.

presented for the autonomous cruise and altitude gain controllers in the previous section would be applicable for decomposing the polymorphic controllers into two isolated general templates pictorial represented in Figure 21. To further validate the above presented templates, an evolved polymorphic controller's neuron outputs have been evaluated in isolation for cruise and altitude gain command (external sensor value of "0" and "1", resp.) and presented in Figure 32(a) and Figure 32(b), respectively. It can be observed from Figure 32(a) that there exist neurons 2, 3, 6, and 7 which meet dynamics-deprived criteria, and further when their saturated outputs are bias folded into neurons 0, 1, 4, and 5, the dynamics of the reduced four neuron network and its effect on MFWR behavior, namely, cruising behavior, matched the original eight-neuron network's generated behavior. Moreover, as anticipated, the reduced four-neuron network with same output dynamics frequency for

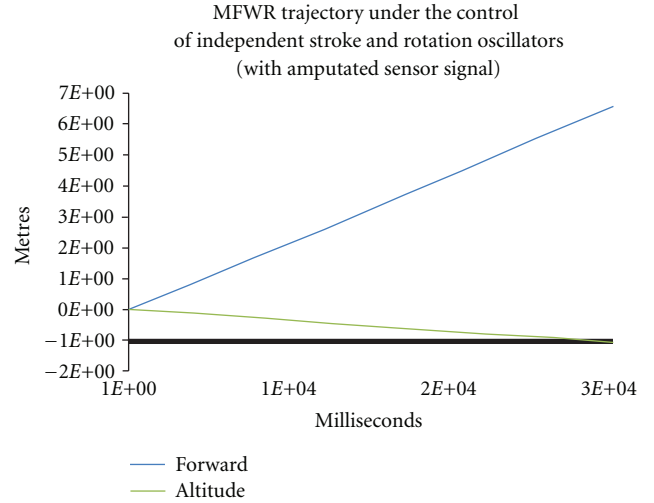


FIGURE 30: The MFWR trajectory produced by the controller during the lesion study performed with the techniques shown in Figure 28. It can be observed that the amputated two independent stroke and rotation oscillators, in absence of the external sensor, were partially successful in controlling the MFWR to have acceptable cruise behavior in it, suggesting the requirement of the external sensor ingestion to achieve the acceptable cruise control.

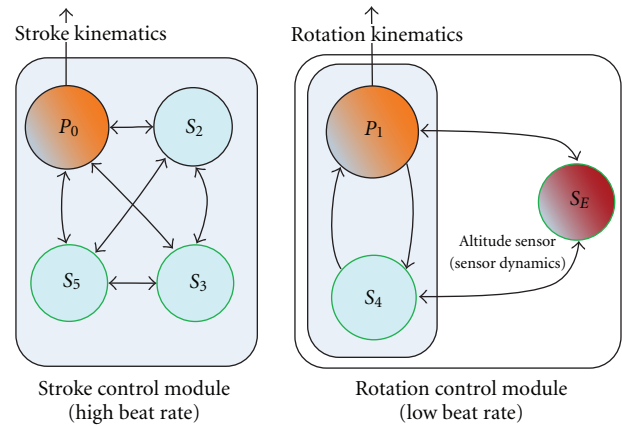


FIGURE 31: The Qualitative functional decomposition template derived for the non-autonomous cruise mode controllers. Most of the nonautonomous cruise mode controllers can be decomposed into the above-shown template, as a combination of two independent oscillator, of which the high-frequency oscillator, controlled the stroke kinematics of the wing with steady amplitude and frequency and the low-frequency oscillator which was evolved to monitor the altitude variations in the MFWR, through the available external sensor module, altered the rotation dynamics continuously to limit the variations in the altitude of the MFWR and simultaneously provided the forward motion in it, by generating required lift and anti-lift with the behavior verified by the general principles of the empirical study.

all the neurons falls under the composite stroke and rotation control module template presented in Figure 21(a).

Moving forward, it can be observed from Figure 32(b), that there exist neurons 1, 2, 3, 4, and 7, which meet dynamics deprived criteria, and further when their saturated outputs are bias folded into neurons 0, 5, and 6, the dynamics

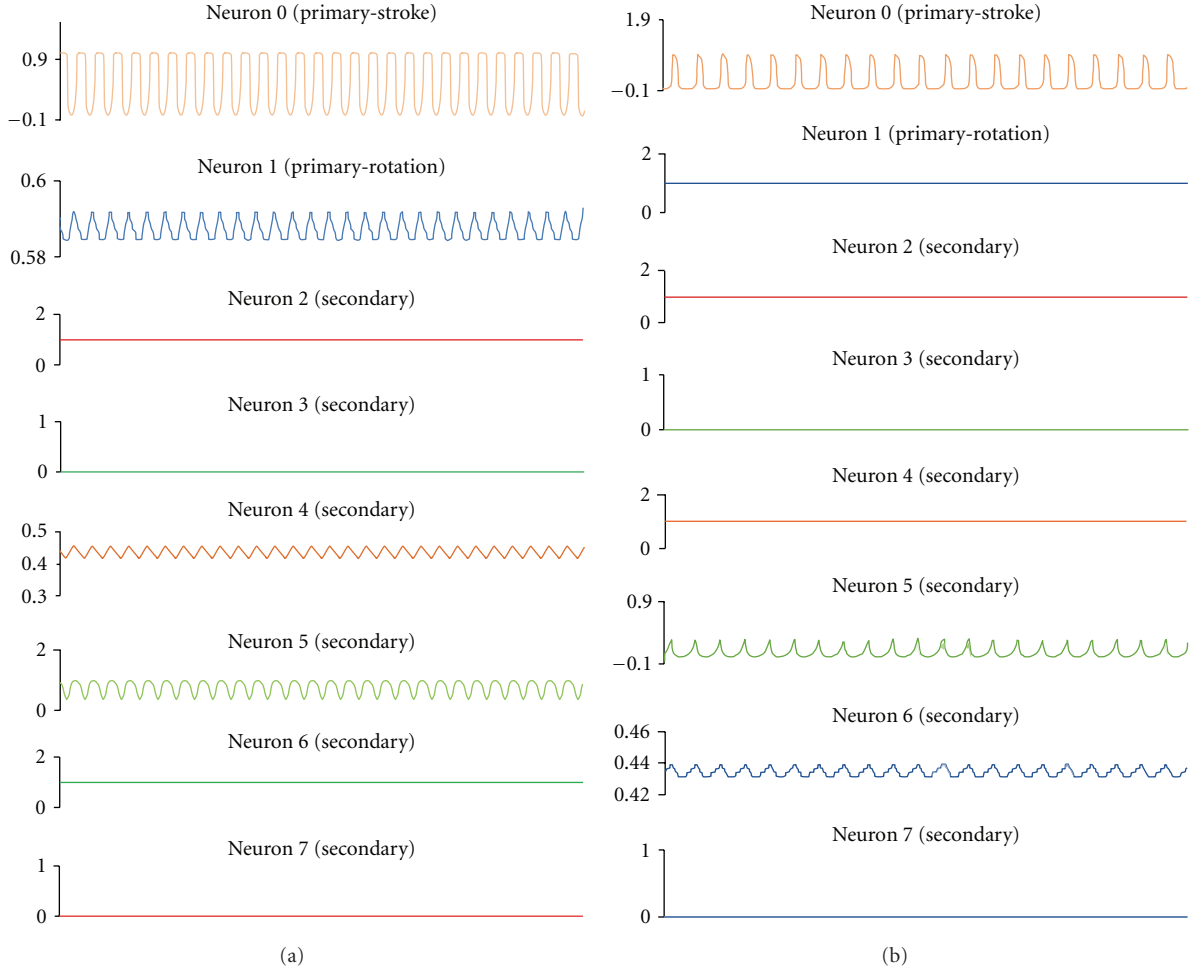


FIGURE 32: (a) shows the neuron output state dynamics of each neuron in the polymorphic controller when presented with a cruise command, whose the external sensor signal is “0.” It can be seen that the neurons 0, 1, 4, and 5 form a composite module with same frequency and would not comply with established criteria for frequency-based grouping. But, the cruise mode controller has been verified to form a single frequency composite stroke and rotation control module as shown in Figure 21(a) and generating appropriate cruise behavior in the MFWR. (b) shows the neuron output state dynamics of each neuron in the polymorphic controller when presented with an altitude gain command whose external sensor signal is “1.” It can be seen that the neurons 0, 5, and 6 forms same frequency group for stroke control and a constant rotation produced by saturated neuron 1. The altitude gain mode controller form a two separate independent stroke and rotation control blocks template as shown in Figure 21(b) and generates appropriate altitude gain behavior in the MFWR.

of the reduced four-neuron network and its effect on MFWR behavior, namely, altitude gain behavior, matched the original eight-neuron network’s generated behavior. Moreover, as anticipated, the reduced four-neuron network with two independent stroke and rotation control module template presented in Figure 21(b). Thus, when presented with an appropriate external command (sensor) value, the static command would be folded into the exiting neurons in the polymorphic controller architecture, as an appropriately evolved external bias that is responsible to shift the dynamics of the rotation and stroke neurons between autonomous altitude gain and cruise mode controllers, generating appropriate wing kinematics in the MFWR as shown in Figure 7(b). Further, it can be observed by comparing the output neuron dynamics of the cruise mode (shown in Figure 32(a)) and the altitude gain mode (shown in Figure 32(b)) that the external sensor’s dynamics modification process is evidently

observed when the dynamically active neuron 4 in cruise mode saturates in altitude gain mode, and vice versa for dynamics of output of the neuron 6.

5. Conclusion

In this paper, we have summarized author’s prior efforts using the Neuromorphic Evolvable Hardware (CTRNN-EH) framework to successfully evolve locomotion and different flight mode controllers, with detailed emphasis on the flight mode controllers. Further, a new frequency-based analysis procedure has been introduced to analyze the different evolved flight mode controllers, besides providing a brief qualitative analysis suggesting the acceptability of the evolved controllers for the given flight mode in the context. Moreover, the proposed frequency-based analysis methodology has been successfully applied to the evolved

autonomous and nonautonomous controllers, and it has been demonstrated that the methodology can be indeed used to decompose the evolved controllers into logically explainable control blocks for further control analysis. Finally, it can be perceived from the presented results and discussion that the proposed Neuromorphic Evolvable Hardware (CTRNN-EH) and frequency-based analysis methodologies can be employed to control problems that are similar to the flapping flight domain, using tabularasa approach. Though, it is not always an appropriate recommendation to employ a tabularasa approach to the control problems at hand; it can serve as an only approach where a suitably impressive closed-form traditional controller does not exist. Moreover, the above-proposed CTRNN-EH methodologies have also been successfully employed to design and evolve hybrid controllers, with evolvable module in the base traditional controller being evolved to supplement the control characteristics of the traditional controllers with rich dynamics of CTRNNs [20]. These CTRNN-EH-based hybrid controllers have been shown to increase the overall robustness and efficacy of the base traditional controllers to handle unforeseen changes in the assumed environment of the controller and the controlled vehicle [21].

References

- [1] V. B. Floris and H. Lipson, "Evolving buildable flapping ornithopters," in *Proceedings of Genetic and Evolutionary Computation Conference (GECCO '05)*, 2005.
- [2] L. Schenato, X. Deng, W. C. Wu, and S. Sastry, "Virtual Insect Flight Simulator (VIFS): a software testbed for insect flight," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '01)*, vol. 4, pp. 3885–3892, May 2001.
- [3] J. C. Gallagher, "An evolvable hardware layer for global and local learning of motor control in a hexapod robot," *International Journal on Artificial Intelligence Tools*, vol. 14, no. 6, pp. 999–1017, 2005.
- [4] S. K. Boddhu and J. C. Gallagher, "Evolved neuromorphic flight control for a flapping-wing mechanical insect model," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '08)*, IEEE Press, Hong Kong, China, June 2008.
- [5] H. J. Chiel, R. D. Beer, and J. C. Gallagher, "Evolution and analysis of model CPGs for walking: I. Dynamical modules," *Journal of Computational Neuroscience*, vol. 7, no. 2, pp. 99–118, 1999.
- [6] J. C. Gallagher, "Evolution and analysis of non-autonomous neural networks for walking: reflexive pattern generators," in *Proceedings of the Congress on Evolutionary Computation*, IEEE Press, Seoul, South Korea, May 2001.
- [7] S. K. Boddhu and J. C. Gallagher, "Evolving non-autonomous neuromorphic flight control for a flapping-wing mechanical insect," in *Proceedings of the IEEE Workshop on Evolvable and Adaptive Hardware (WEAH '09)*, Nashville, Tenn, USA, April 2009.
- [8] S. K. Boddhu and J. C. Gallagher, "Evolving neuromorphic flight control for a flapping-wing mechanical insect," *International Journal of Intelligent Computing and Cybernetics*, vol. 3, no. 1, pp. 94–116, 2010.
- [9] S. K. Boddhu, *Evolution and analysis of neuromorphic flapping-wing flight controllers [Ph.D. thesis]*, Wright State University, 2010.
- [10] D. Sbarbaro-Hofer, D. Neumerkel, and K. Hunt, "Neural control of a steel rolling mill," *IEEE Control Systems Magazine*, vol. 13, no. 3, pp. 69–75, 1993.
- [11] C. M. Lin and C. F. Hsu, "Supervisory recurrent fuzzy neural network control of wing rock for slender delta wings," *IEEE Transactions on Fuzzy Systems*, vol. 12, no. 5, pp. 733–742, 2004.
- [12] S. K. Boddhu, J. C. Gallagher, and S. A. Vigraham, "A commercial off-the-shelf implementation of an analog neural computer," *International Journal on Artificial Intelligence Tools*, vol. 17, no. 2, pp. 241–258, 2008.
- [13] S. K. Boddhu, J. C. Gallagher, and S. Vigraham, "A reconfigurable analog neural network for evolvable hardware applications: Intrinsic evolution and extrinsic verification," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '06)*, IEEE Press, Vancouver, Canada, July 2006.
- [14] K. I. Funahashi and Y. Nakamura, "Approximation of dynamical systems by continuous time recurrent neural networks," *Neural Networks*, vol. 6, no. 6, pp. 801–806, 1993.
- [15] G. R. Kramer and J. C. Gallagher, "An analysis of the search performance of a mini-population evolutionary algorithm for a robot-locomotion control problem," in *Proceedings of the IEEE Congress on Evolutionary Computation (IEEE '05)*, pp. 2768–2775, IEEE Press, September 2005.
- [16] R. D. Beer, H. J. Chiel, and J. C. Gallagher, "Evolution and analysis of model CPGs for walking: II. General principles and individual variability," *Journal of Computational Neuroscience*, vol. 7, no. 2, pp. 119–147, 1999.
- [17] M. H. Dickinson, F. O. Lehmann, and S. P. Sane, "Wing rotation and the aerodynamic basis of insect flight," *Science*, vol. 284, no. 5422, pp. 1954–1960, 1999.
- [18] L. Schenato, X. Deng, and S. Sastry, "Flight control system for a micromechanical flying insect: architecture and implementation," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '01)*, vol. 2, pp. 1641–1646, May 2001.
- [19] R. D. Beer, "On the dynamics of small continuous time recurrent neural networks," in *Adaptive Behavior*, vol. 3, 4, pp. 469–509, The MIT Press, Boston, Mass, USA, 1995.
- [20] J. C. Gallagher, B. David Doman, and W. Michael Oppenheimer, "The technology of the gaps: an evolvable hardware synthesized oscillator for the control of a flapping-wing micro air vehicle," *IEEE Transactions on Evolutionary Computation*. In press.
- [21] J. C. Gallagher and M. Oppenheimer, "An improved evolvable oscillator for all flight mode control of an insect-scale flapping-wing micro air vehicle," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '11)*, IEEE Press, 2011.

