Evolving swimming controllers for a simulated lamprey with inspiration from Neurobiology

Auke Jan Ijspeert* , John Hallam*, David Willshaw†

*Dept. of Artificial Intelligence, U. of Edinburgh, 5 Forrest Hill, Edinburgh EH1 2QL

†Centre for Cognitive Science, U. of Edinburgh, 2 Buccleuch Place, Edinburgh EH8 9LW

{aukei, john}@dai.ed.ac.uk, david@cns.ed.ac.uk

Abstract

This paper presents how neural swimming controllers for a simulated lamprey can be developed using evolutionary algorithms. A Genetic Algorithm is used for evolving the architecture of a connectionist model which determines the muscular activity of a simulated body. This work is inspired by the biological model developed by Ekeberg which reproduces the Central Pattern Generator observed in the real lamprey [Ekeberg 93]. In evolving artificial controllers, we demonstrate that a Genetic Algorithm can be an interesting design technique for neural controllers and that there exist alternative solutions to the biological connectivity. A variety of neural controllers are evolved which can produce the pattern of oscillations necessary for swimming. These patterns can be modulated through the external excitation applied to the network in order to vary the speed and the direction of swimming. The best evolved controllers cover larger ranges of frequencies, phase lags and speeds of swimming than Ekeberg's model. We also show that the same techniques for evolving artificial solutions can be interesting tools for developing neurobiological models. In particular, biologically plausible controllers can be developed with ranges of oscillation frequency much closer to those observed in the real lamprey than Ekeberg's hand-crafted model.

Keywords: Neural control; genetic algorithm; simulation; central pattern generator; swimming; lamprey.

1 Introduction

As locomotion is one of the most important aspects of living systems, the control of movements has been extensively studied both in Biological Sciences and Engineering. Neurobiology has shown the diversity of very effective motion controllers present in the animal kingdom. These controllers, composed of interconnected neurons, are able to create the oscillatory activity necessary for motion as well as to adapt to dynamic environments using sensory information. The controllers are generally distributed and present remarkable robustness and flexibility. Inspired by these interesting features, roboticists have applied *Artificial Neural Networks* (ANN) to control robots, using abstract mathematical models of neurons for processing the signals needed for control. Advances in Neurobiology and in Robotics are now at a point where models of biological controllers can be directly implemented in robots; neural controllers inspired by insects have, for instance, been used to control the walking of hexapod robots [Espenschied *et al.* 93, Cruse *et al.* 95]. Robotics benefits from biology as a source of inspiration on how to solve control problems, and Neurobiology may in return benefit from these experiments by receiving an evaluation of the capacity of a biological model to produce a specific behaviour. Another interesting interaction is the use of ANN techniques for investigating the computations performed by the central nervous system or for developing neurobiological models. As an example of the latter, an adapted backpropagation algorithm has been used to develop a biologically plausible connectionist model reproducing the local bending reflex of the leech [Lockery & Sejnowski 93].

Our research is interested in biologically inspired neural locomotion controllers for autonomous agents and how to develop controllers using a Genetic Algorithm (GA). In particular, we study here *anguiliform* swimming, one of the simplest forms of locomotion, which is used by eels and lampreys. Motion is generated by traveling undulations of the body, without the use of fins. Although simple, this type of locomotion presents the basic features of any natural motion control, namely the creation and the modulation of sets of oscillations with characteristic frequencies and phase lags.

This work is inspired by Ekeberg's connectionist model of the lamprey's swimming controller [Ekeberg 93]. Using a GA, we develop artificial controllers by evolving the connections and synaptic weights between neurons similar to those of Ekeberg's model. We are interested in evaluating how efficient GAs can be for the design of neural controllers. A possible application of such a design technique could, for instance, be the automatic generation of neural controllers for a specific robot body. We are also interested in investigating whether there exists alternative neural configurations to the biological one found in the lamprey. Finally, we investigate whether GAs could be used to help neurobiological modeling by instantiating variables which are difficult to measure (the synaptic weights of a connectionist model, for instance).

In this paper, we firstly present the current knowledge of the lamprey's swimming controller and the connectionist model made by Ekeberg. We then develop alternative solutions in three evolutionary stages, with firstly the evolution of a segmental oscillator, secondly the evolution of the couplings between a set of segmental oscillators, and finally, the evolution of sensory feedback from stretch sensitive cells. We show that there exists a variety of different connectivities for efficient swimming controllers and that it is possible to evolve controllers capable of producing patterns of oscillations over large ranges of frequencies and phase lags. In a preliminary experiment, we also show that it is possible to evolve sensory feedback pathways from stretch sensitive cells which allow the simulated lamprey to cross a speed barrier. As an example of how the GA could be used to develop biological models, we then repeat the evolutions by adding constraints on the encoding in order to evolve controllers which preserve the segmental connectivity observed in the real lamprey. The GA is therefore used to generate one significant part of the model that Ekeberg has defined by hand –the synaptic weights. We find that Ekeberg's synaptic weights can be optimized for the given controller topology to produce oscillations over a significantly larger frequency range, much closer to that observed in the real lamprey. We also evolve sensory feedback from stretch sensitive cells which comes out to be very similar to that found in the real lamprey.

2 Biological background and Ekeberg's connectionist model

The lamprey is one of the earliest and simplest vertebrates. It has no paired fins and swims by propagating an undulation along its body, from head to tail. The network of neurons located in its spinal cord which controls the motion has been extensively studied [Buchanan & Grillner 87, Grillner *et al.* 88, Grillner *et al.* 91]. Physiological experiments on isolated spinal cords have shown that this network of neurons is a Central Pattern Generator (CPG), which means that it can generate patterns of oscillation without oscillating excitation either from the brain or from sensory feedback. Because small parts of the spinal cord can be isolated (up to 2 segments) and still produce oscillations when subjected to an excitatory bath, the CPG is considered to be an interconnection of local (segmental) oscillators. The complete controller is formed of approximately 100 of these segments. The CPG can produce oscillations at frequencies varying between 0.25 and 10 Hz; and it maintains constant wavelength of the undulation (usually corresponding to the length of the body) as the frequency is changed.[1]

The CPG controlling the swimming has been modelled at several levels of abstraction, namely at a neurobiological [Ekeberg *et al.* 91, Wallen *et al.* 92], a connectionist [Buchanan 92, Ekeberg 93] and an abstract oscillator level [Kopell 95, Williams & Sigvardt 95]. Neurobiological models using relatively realistic models have demonstrated that the current state of knowledge of the lamprey is sufficient to produce models whose results agree with the physiological observations. Connectionist models of the CPG use less realistic neuron models which capture only the main feature of neurons, that is, their ability to change the frequency of action potential spikes in their axon (the output) depending on the sum of activity in their dendrites (the total input). They have demonstrated that the connectivity in itself is sufficient to produce many of the oscillation patterns observed in the real lamprey without the need for complicated neuronal mechanisms. Finally, at the most abstract level, the swimming controller can be modeled as a chain of mathematical oscillators in order to study which kind of couplings can produce a phase relation between segments which is constant over the whole spinal cord of the lamprey and also remains a fixed proportion of a cycle when the frequency of the oscillations is changed.

The work presented here is situated at the connectionist level. A connectionist model has the advantages of being abstract enough to be tractable, in the sense that the simulation of the complete CPG (segments over the whole spinal cord) and of a mechanical simulation of the body of a lamprey do not require too much computation time and too many parameter choices. At the same time it is concrete enough to give some interesting insights into the functioning of the CPG of the real lamprey (as mentioned above, connectionist models can produce most of the oscillation patterns observed in the real lamprey). Finally, the connectionist models made for modelling the lamprey's CPG are very similar to those currently used for the control of some robots [Lewis *et al.* 93]. Therefore, learning how to design them may be directly applicable to the design of a controller for a swimming robot, for instance.

We have reproduced Ekeberg's two-dimensional connectionist model [Ekeberg 93]. This model has inspired our work and it is the biological model with which our evolved controllers will be compared. We will refer to it as the *biological model*. The model combines a simulation of the neural controller with a mechanical simulation of a lamprey in water allowing direct evaluation of how neural activity is transformed into mechanical movements. It also offers the possibility of investigating the effect of sensory feedback from stretch sensitive cells. We present here the model and give some quantitative results (Ekeberg's paper [*ibid.*] presents mainly qualitative behaviours of the model) which will be compared with the performances of the evolved swimming controllers.

[1]This maintenance of constant wavelength over a large range of frequencies is an interesting aspect of the CPG. It means that the phase lag between segments is maintained at a constant proportion of the period of oscillations and that the time delays between segments vary considerably depending on the frequency. The lags between segments can not therefore be due to delays in conduction along descending axons because axons have, more or less, fixed velocities of conduction.

2.1 Neural controller

The neural controller is composed of 100 interconnected segmental networks (Figure 1). Each segmental network is an oscillator of 8 neurons of four types: two motoneurons (MN), two excitatory interneurons (EIN), two contralateral inhibitory interneurons (CIN) and two lateral inhibitory interneurons (LIN). Each neuron unit in the model represents a population of functionally similar neurons in the real lamprey. Neurons receive excitatory input from the brainstem. The weights of the segmental connections are given in Table 1. The complete controller is formed by interconnecting 100 copies of this segmental network. An interconnection consists of extending the connection from one neuron to another in one segment to the corresponding neuron in neighbouring segments (Table 1). A neuron unit is modeled as a leaky integrator with a saturating transfer function. Its output u corresponds to the mean firing frequency of the population it represents ($\in [0, 1]$) and is calculated as follows:

$$\dot{\xi}_+ = \frac{1}{\tau_D}(\sum_{i \in \Psi_+} u_i w_i - \xi_+) \tag{1}$$

$$\dot{\xi}_- = \frac{1}{\tau_D}(\sum_{i \in \Psi_-} u_i w_i - \xi_-) \tag{2}$$

$$\dot{\vartheta} = \frac{1}{\tau_A}(u - \vartheta) \tag{3}$$

$$u = \begin{cases} 1 - \exp\{(\Theta - \xi_+)\Gamma\} - \xi_- - \mu\vartheta & (u > 0) \\ 0 & (u \leq 0) \end{cases} \tag{4}$$

where Ψ_+ and Ψ_- represent the groups of pre-synaptic excitatory and inhibitory neurons respectively, ξ_+ and ξ_- are the delayed 'reactions' to excitatory and inhibitory input and ϑ represents the frequency adaptation observed in some real neurons. The parameters of each type of neuron are given in Table 2. These parameters, as well as the connection weights of Table 1, have been defined so that the simulation of the model fits physiological observations [Ekeberg 93].

When the segmental networks receive adequate excitation from the brainstem, they oscillate regularly with the left and right neurons out of phase. The frequency of oscillations increases with the level of excitation. The interconnections between segments leads to traveling waves of oscillations when the most rostral segments receive extra excitation compared to the others. The lag between segments relative to the period of oscillation is then constant over the whole spinal cord, except for the first and last ten segments at the extremities where it decreases towards zero. The value of the lag can be varied with the amount of extra excitation, and the higher the extra excitation, the larger the lag (or the shorter the wavelength). Interestingly, the frequency of oscillation and the wavelength of the undulation can be changed nearly independently. The model can therefore reproduce the capacity of the real lamprey to cover a whole range of different frequencies of oscillations while keeping the wavelength of the undulation constant. Our simulations show that frequencies between 1.6 and 5.5 [Hz] and phase lags per segment between 0.0% (no travelling wave) and 2.4% of the period can be obtained (the wavelength is then smaller than 50% of the 100-segment body).

2.2 Mechanical simulation

We have also reproduced Ekeberg's mechanical model of the lamprey's body interacting with water [Ekeberg 93]. The body is made of 10 rigid links connected through one-degree of freedom joints, with each mechanical link corresponding therefore to 10 neural

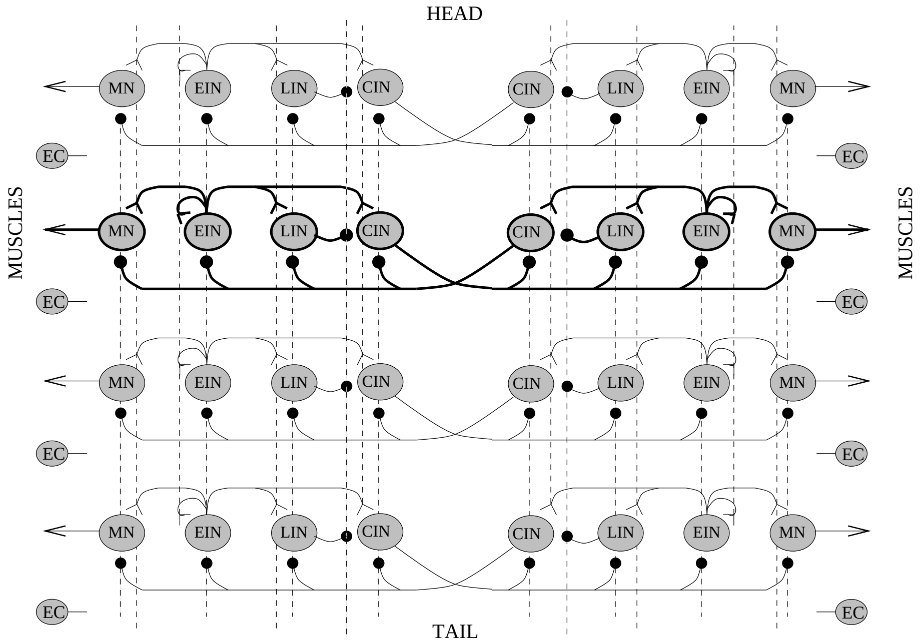


Figure 1: Biological controller. The controller is made of 100 interconnected segmental oscillators (only 4 segments shown) composed of 8 neurons each. Four types of neurons are present in the oscillators: 3 types of interneurons EIN, CIN and LIN and the motoneurons MN. The controller can receive feedback from the stretch sensitive edge cells EC. (See text for explanation.)

from: / *to:*	EINl	CINl	LINl	EINr	CINr	LINr	BS
EINl	0.4 [2, 2]	-	-	-	-2.0 [1, 10]	-	2.0
CINl	3.0 [2, 2]	-	-1.0 [5, 5]	-	-2.0 [1, 10]	-	7.0
LINl	13.0 [5, 5]	-	-	-	-1.0 [1, 10]	-	5.0
MNl	1.0 [5, 5]	-	-	-	-2.0 [5, 5]	-	5.0
EINr	-	-2.0 [1, 10]	-	0.4 [2, 2]	-	-	2.0
CINr	-	-2.0 [1, 10]	-	3.0 [2, 2]	-	-1.0 [5, 5]	7.0
LINr	-	-1.0 [1, 10]	-	13.0 [5, 5]	-	-	5.0
MNr	-	-2.0 [5, 5]	-	1.0 [5, 5]	-	-	5.0

Table 1: Biological configuration, as given in [Ekeberg 93]. Excitatory and inhibitory connections are represented by positive and negative weights respectively. Left and right neurons are indicated by l and r. BS stands for brain stem. The extension of the segmental connection to neighbour segments is given in brackets (extensions to the rostral and caudal direction, respectively).

Neuron type	Θ	Γ	τ_D	μ	τ_A
EIN	-0.2	1.8	30 ms	0.3	400 ms
CIN	0.5	1.0	20 ms	0.3	200 ms
LIN	8.0	0.5	50 ms	0.0	-
MN	0.1	0.3	20 ms	0.0	-

Table 2: Neuron parameters, as given in [Ekeberg 93]. Θ is the threshold, Γ the gain, τ_D the time constant of the dendritic sums, μ the coefficient of frequency adaptation and τ_A, the time constant of the frequency adaptation.

segments. Muscles are connected to each link (Figure 2) and are modeled as a combination of springs and dampers. The motoneurons can 'contract' muscles by increasing their spring constant, thus reducing their resting length[2]. The acceleration of a link is due to three forces: the torques due to the muscles, the forces linked to the constraints of the rigid links and the forces due to the water. It is assumed that the speed of the water relative to the body is sufficiently high for the forces exerted by the water to be mainly inertial forces (proportional to the density of the fluid and the square of the relative speed). Propulsion through water is obtained when the traveling waves of neural activity are transformed into traveling undulations of the body. The head to tail propagation of the undulation leads to forces from the surrounding water which propel the fish forward.

Our simulation is identical to Ekeberg's, except for some mechanical parameters values. It appeared that, given the geometry of the body[3], the masses of the links m_i should be increased by a factor of π in order to satisfy the assumption that the density of the body is the same as that of the water. Also, the coefficients of the drag forces ($\lambda_{\perp}$,$\lambda_{||}$) seem to be a factor 10 too small compared to the assumptions of a near cylindrical body (drag coefficient C close to 1). Table 3 gives the mechanical values we used in our simulations. Note that we increased the muscles' coefficients by a factor of π to compensate for the larger masses and the stronger water forces. Because of these new values, we obtain speeds of swimming which are approximately 40% smaller for the two quantitative examples given in [Ekeberg 93]. However, the qualitative behaviours of the two simulations are identical with, namely, identical influence on the speed of swimming of the global excitation and the extra excitation on the first segments.

Figure 3 shows a simulation of both the neural controller and the mechanical model of the body. We have simulated the biological model over the whole range of frequencies and phase lags it can produce, and the highest speed it can reach is 0.50 m/s (global excitation of 0.65 with 70% extra excitation on first segments producing oscillations at 5.5 Hz and relative lags between segments of 1.2%).

The mechanical simulation allows a direct evaluation of the efficiency of the CPG to control swimming (in terms of speed of swimming, for instance). It also allows to study the effect of sensory feedback from the edge cells which are stretch sensitive cells located on both sides of the spinal cord [Viana Di Prisco *et al.* 90]. These cells can be modeled as outputing a signal proportional to the local curvature of the body. While in [Ekeberg 93] the effects of the edge cells on the controller are negligible (very low synaptic weight), in [Ekeberg *et al.* 95], it is shown that feedback can help the lamprey to cross a speed barrier (water with local speed opposite to the direction of swimming) by coordinating the neural activity with the actual movements of the body and preventing a change of direction.

[2]The muscles of a mechanical link receive input from the motoneurons in the middle of the corresponding neural segments.

[3]Each link is 30 mm long and has an elliptical cross section. The height of each link is constant (30 mm) and the width is 20 mm for the four first links and then decreases linearly towards the tail.

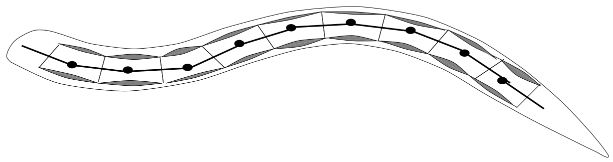


Figure 2: Biomechanical simulation. A two dimensional body of a lamprey is modeled as a set of rigid links connected through one-degree of freedom joints. The muscles are simulated as a combination of dampers and springs. Neural activity is transformed into muscular activity by the motoneurons changing the spring constants of the muscles.

link	w_i [mm]	m_i [g]	I_i [g mm^2]	$\lambda_\perp$ [Ns^2/m^2]	$\lambda_\parallel$ [Ns^2/m^2]
1	20.0	14.1	1414	0.45	0.3
2	20.0	14.1	1414	0.45	0.2
3	20.0	14.1	1414	0.45	0.1
4	20.0	14.1	1414	0.45	0.0
5	17.2	12.2	1137	0.45	0.0
6	15.0	10.6	944	0.45	0.0
7	11.7	8.3	691	0.45	0.0
8	8.3	5.9	465	0.45	0.0
9	5.0	3.5	271	0.45	0.0
10	1.7	1.2	90	0.45	0.0

Table 3: Corrected parameters for the mechanical simulation (see [Ekeberg 93]).

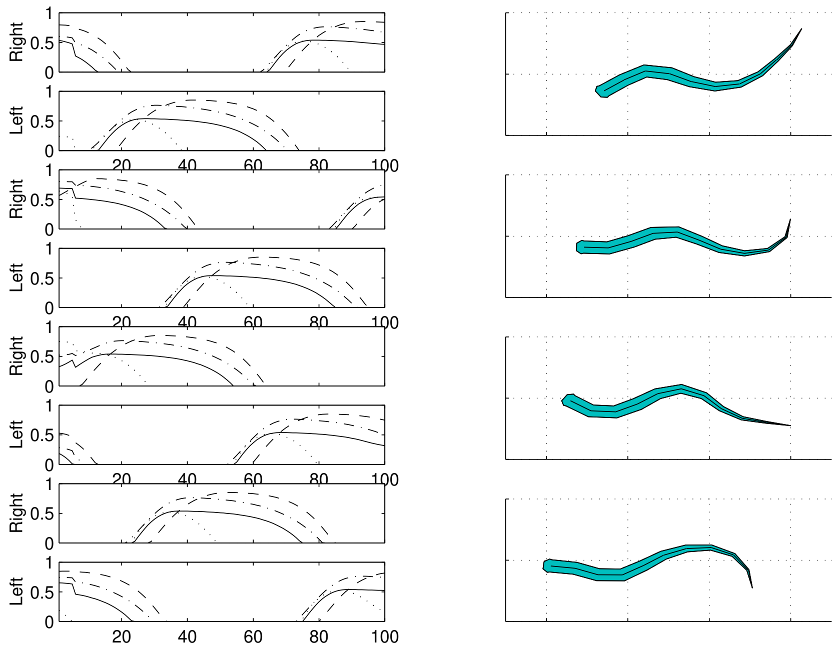


Figure 3: Biomechanical simulation of the lamprey. The level of excitation is 0.4 (arbitrary value) over the spinal cord, with 70% extra on the 5 first segments. The segments oscillate at 4.0 Hz and the speed of swimming is 0.40 m/s. A) Neural activity of the 800 neurons along the spinal cord. B) Mechanical simulation. There is 50ms between each snapshot, and the doted vertical lines are separated by 100mm.

3 Evolving Central Pattern Generators

We develop alternative swimming controllers by using the GA to define suitable connections and synaptic weights between neurons similar to those of Ekeberg's model. The similarity of the neuron models will allows us to directly compare the performances of the evolved solutions with those of the biological model. The controllers are evolved in three stages. First, segmental oscillators are evolved; then multi-segmental controllers are generated by evolving the couplings between copies of a chosen segmental oscillator; and, finally, connections providing sensory feedback from stretch sensitive cells are added. The first stage requires only the simulation of the neural activity within a segment, while the two last are realised with a simulation of the whole controller together with the mechanical simulation of the body interacting with water. This decomposition in stages was motivated by properties of the biological controller we wanted to reproduce. Firstly, segmental networks can oscillate independently when completely isolated from the others and from sensory feedback. Secondly, being a CPG, the whole (multi-segmental) controller can produce coordinated traveling waves of oscillations in a completely isolated spinal cord indicating that there is no need for sensory feedback for the coordination between segments. Finally, it appears that the role of feedback from stretch sensitive cells is mainly necessary for coordinating the neural activity with the actual movements of the body when these are disturbed by the environment. The third stage therefore includes the simulation of a speed barrier which can only be crossed when sensory feedback is provided to the CPG.

The controllers are evaluated depending on their ability to control swimming. In order to obtain similar performance to the biological controller, we define the following goals for the evolved controllers:

- production of stable oscillations over the spinal cord,
- generation of phase lags between segments,
- frequency of oscillation dependent on global excitation,
- lag between segments dependent on the proportion of extra excitation of the most rostral segments,
- independence of lag and frequency,
- large ranges of frequencies, lags and speeds of swimming,
- capacity to incorporate sensory feedback from stretch sensitive cells for crossing a speed barrier.

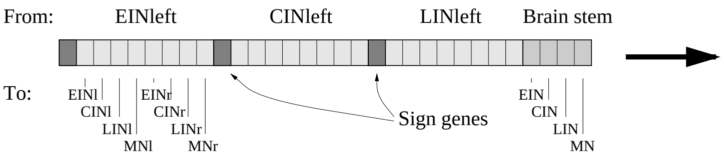
Section 3.1 presents how artificial solutions are evolved and Section 3.2 shows how the same technique can be used as help for developing neurobiological models. The main difference between the two sets of experiments is that for the artificial solutions only the neural models are fixed and both the connections and the synaptic weights are evolved, while for the second set of experiments solutions which preserve the biological segmental connectivity are evolved.

3.1 Artificial solutions

3.1.1 First evolutionary stage: Segmental oscillators

A segmental oscillator is generated by evolving both the connectivity between the 8 neurons composing the biological segment and the *sign* (excitatory and inhibitory) of each interneuron. The quality of a solution is evaluated depending on its ability to produce regular oscillations whose frequency can be varied with the level of excitation.

Encoding A chromosome represents the connectivity between the 8 neurons of the biological model. The number of neurons and their dynamics (defined by the 5 parameters of equations 1-4) are thus fixed and only the sign of neurons and the synaptic weights of the connections are evolved. We impose a left-right symmetry. A chromosome is a string of 31 genes which are real numbers between 0 and 1 (Figure 4). There are three *sign genes* which define whether the interneurons are inhibitory or excitatory.[4] The motoneurons are excitatory. The other genes directly encode the synaptic weights and correspond (via a linear transformation) to a real value between -5 and 0 or between 0 and 15, depending on the value of the sign gene (the interneuron is inhibitory if the sign gene is smaller than 0.5 and excitatory otherwise). Finally the four last genes of a chromosome determine the synaptic weights of the connections coming from the brain stem and correspond to a real value between -5 and 15. The weight boundaries (-5 and 15) have been defined so that they include the range of weights of the biological model (-2 and 13). Note that, although we will keep the denominations EIN, CIN and LIN to distinguish between the different types of interneurons (the different neuron dynamics given by equations 1-4), they lose their meaning as a description of the function or even the sign these neurons had in the biological model.



	EINl	CINl	LINl	EINr	CINr	LINr	Brain Stem
EINl	1.3	-0.8			-2.7		4.3
CINl		-3.5	-4.2				10.8
LINl	7.4						3.0
MNl	6.0				-3.0		5.8
EINr				1.3	-0.8		4.3
CINr		-2.7			-3.5	-4.2	10.8
LINr				7.4			3.0
MNr		-3.0		6.0			5.8

Figure 4: Encoding of a segmental network.

Genetic Algorithm The algorithm used is a variation of the standard GA (see for instance [Goldberg 89]) with the usual binary encoding being replaced by a real number encoding. The algorithm starts with a randomly generated initial population. At each generation, a fixed number of children are generated using three operators, *crossover*, *mutation* and *pruning*. The *crossover* operator chooses couples of parents, with a rank-based probability, for breeding and creates couples of children either by two-point crossover (probability *Prob_Xover*) or by simply copying the two parents. The *mutation* operator mutates each gene of the children with a probability *Prob_mut* and the mutation consists of adding or substracting a small random number within a mutation range:

$$new\_value = old\_value + Mut\_Range \cdot rand$$

where *rand* is a random number $\in [-0.5, 0.5]$. Finally, the *pruning* operator is applied which sets, with a probability *Prob_prun*, a gene to the value corresponding to a null weight. The children are then evaluated, and the size of the population is kept constant

[4]We chose to have neurons which are either inhibitory or excitatory (rather than neurons which could both inhibit and excite other neurons) in order to develop networks under the same constraints as the biological model, in the sense that biological neurons do not usually emit both excitatory and inhibitory neurotransmitters.

by rejecting, at each generation, the worst solutions of the increased population (old population plus children). The GA parameters used for the evolution of segmental oscillators are given in Table 4.

Population size	100
Number of children	30
Crossover probability	0.5
Mutation probability	0.4
Mutation range	0.2
Pruning probability	0.1

Table 4: GA parameters for evolving segmental oscillators

Evaluation We define a fitness function that rewards three desired characteristics for segmental oscillators:

1. The production of regular oscillations of the motoneurons, with one peak of activity per period and with the left and right neurons out of phase.
2. A frequency of oscillation which can be varied and which increases monotonically with the level of external excitation.
3. A minimal set of connections. In [Ijspeert *et al.* 97], we evolved fully connected solutions, the weakest connections of which proved unnecessary for the creation of oscillations and could be removed without affecting the neural activity.

The mathematical definition of the fitness function is given in Appendix A1. Solutions are evaluated by fixed-duration simulations (simulated time of 3000 ms) with asymmetric initial conditions (all left neurons excited). Note that the evaluation of the fitness of a segmental oscillator is only based on the neural activity of the motoneurons because they are the only neurons influencing the muscular activity. This means that the interneurons can have any activity and that solutions with fewer than 8 active neurons can be developed.

Results We carried out 10 evolutions with populations of 100 chromosomes. The evolutions were stopped after 500 generations, an arbitrarily chosen limit which was sufficient for the best solutions of each run to reach a fitness value higher than 0.3. Each evolution started with a different randomly generated initial population. When the neurons are randomly connected, there is little chance of the resulting network oscillating. In our case, only 3 solutions within the 1000 initially generated solutions produce regular oscillations, and this only at a fixed frequency. Within 500 generations, all evolutions converged to final populations composed of networks producing regular oscillations with variable frequency. Similary to the biological model, the frequency of oscillations and the amplitude of the motoneuron signals increase with the level of tonic excitation. When the oscillator receives too little excitation, the motoneurons stay inactive, and when the excitation level is too high they stabilize, after a few oscillations, in an asymmetrical state with one active and one inactive motoneuron. The fittest solutions of all runs cover a larger range of frequencies than the biological model, reaching lower and higher frequencies. The results are summarized in Table 5.

The ten runs did not all converge to the same network configuration, but rather to several different solutions with similar fitness values (see Appendix B for a description of the evolved configurations). These best solutions differ not only in their connectivity, but

	Fitness value	Frequency range in [Hz]	Oscillating neurons	Number of connections
Biol. model	0.18	[1.7, 5.6]	EIN+,CIN−,LIN−	26
run1	0.44	[1.0, 8.4]	EIN−,CIN+,LIN−	32
run2	0.39	[1.0, 6.8]	EIN−,LIN−	34 → 24
run3	0.43	[1.1, 7.7]	EIN−,CIN−	34 → 26
run4	0.47	[1.2, 9.9]	EIN+,CIN−,LIN−	38
run5	0.40	[1.1, 7.9]	EIN−,CIN−,LIN−	34
run6	0.51	[1.1, 11.3]	EIN−,CIN−	40 → 24
run7	0.54	[1.2, 8.0]	EIN−,CIN−	32 → 24
run8	0.31	[1.1, 5.8]	EIN−,CIN−	36 → 22
run9	0.60	[1.0, 10.4]	EIN−,CIN+,LIN−	32
run10	0.44	[1.3, 8.6]	EIN−,CIN−	28 → 22

Table 5: Summary of results for the evolved segmental networks. The table gives, for the best solution of each population, the lowest and the highest frequency at which the segmental network can oscillate, the interneurons active in the oscillator with the sign of their influence and the number of connections of the oscillator. When fewer than 6 interneurons are active, the number of connections after complete removal of the inactive neurons are given.

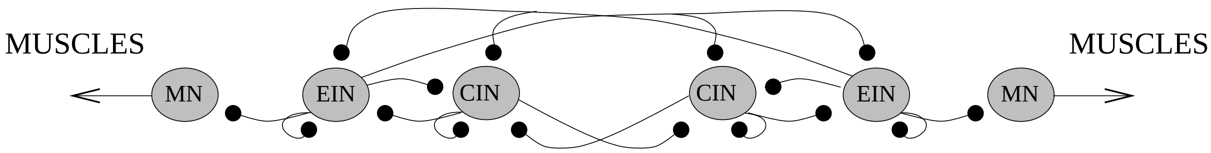
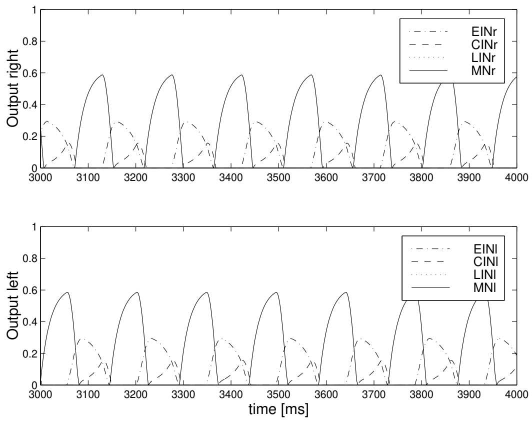


Figure 5: Segmental network configuration to which four out of ten runs converged (runs 3,6,8 and 10).



from: to:	EINl	CINl	LINl	EINr	CINr	LINr	BS
EINl	−1.9	−5.0	-	−4.2	−0.1	-	12.1
CCINl	−2.8	−1.7	-	−5.0	−2.1	-	4.7
LINl	-	-	-	-	-	-	-
MNl	−5.0	-	-	-	-	-	4.8
EINr	−4.2	−0.1	-	−1.9	−5.0	-	12.1
CCINr	−5.0	−2.1	-	−2.8	−1.7	-	4.7
LINr	-	-	-	-	-	-	-
MNl	-	-	-	−5.0	-	-	4.8

Figure 6: *Left:* oscillations in an evolved segmental network (run 6). *Right:* corresponding neural configuration.

also in the sign of the interneurons and in the number of neurons active in the creation of oscillations. Six evolutions converged to networks with only 6 rather than 8 active neurons. These solutions are composed of the two motoneurons and four inhibitory interneurons of two different types which create the oscillations. This demonstrates that excitatory interneurons are not necessary (at least in a connectionist model) for the production of oscillations over a large frequency range. Most often the inactive neurons are the LINs, which is probably due to their high threshold: they have more chance to be inactive unless they receive some strong excitatory input. Amongst these six runs, four (runs 3,6,8 and 10) converged to similar configurations based on an identical oscillator structure (when the possible permutations due to the left-right symmetry are taken into account) composed of inhibitory EIN and CIN and the motoneurons (Figure 5). Although there are some variations of the values of the connection weights, the solutions have all the same connections with similar strengths and present the same behaviour. Figure 6 shows the simulation and the configuration of one of them.

Although most of the evolved oscillators differ significantly from the biological segmental network, one controller (the best solution of run4) presents strong similarities with it. That solution is composed of the same interneurons —same number and with the same sign— as the biological model. Furthermore, although it has more connections than the biological model, it has a similar structure with inhibitory connections from the CIN to all the neurons on the contralateral side and a strong inhibitory connection from the LIN neurons to the CIN neurons on the same side acting as a burst terminator (as with the biological model, the LIN becomes active later in the cycle and inhibits the ispilateral CIN allowing the activity on the other side to start).

The pruning operator of the GA has led to solutions using significantly fewer than the 56 possible connections. The solutions have therefore few weak (low weight) connections which do not play an important role in the creation of oscillations. In the case of solutions which use fewer than 8 neurons, the majority of the connections to and from the inactive neurons have been cut; for these solutions Table 5 gives the number of connections before and after the removal of the inactive neurons.[5]

3.1.2 Second evolutionary stage: Multi-segmental controllers

We generate complete CPGs by evolving the interconnections between fixed segmental networks. The best segmental oscillators of each evolution of the previous evolutionary stage are chosen. The evaluation of complete CPGs is based on simulations of both the neural activity and the mechanical movements of the body. The quality of the solutions will depend on their capacity to control swimming at different speeds, frequencies of oscillation, and lags between segments.

Encoding and GA A chromosome encodes how the segments are interconnected. Similarly to the biological model, interconnections are extensions of segmental connections to the corresponding post-synaptic neurons in neighbouring segments. For each segmental connection, the extensions in each direction are encoded into a chromosome with values between 0 (no extension to neighbour segments) and a maximal extension of 12 (this value has been chosen to include the maximal extension of the biological model which is 10). The number of genes is therefore twice the number of segmental connections. As previously, a left-right symmetry is assumed. The connection weights of the segmental

[5]The fact that there are still some weak connections which could be removed without affecting the oscillation is probably due to the fact that the mutation operator can at any time give a non-zero value to a weight which had previously been set to zero. The mutation operator therefore slows down the convergence to the minimal set of connections.

networks are fixed except for a rescaling depending on the extension of the interconnection: the weight of a connection to a neuron is divided by the number of segments it receives input from.

The same GA as for the evolution of segmental oscillators is used but without the pruning operator. Although the genes are transformed into integers when the chromosome is decoded, they are real numbers in the GA. The GA parameters for the evolutions are given in Table 6.

Population size	40
Number of children	12
Crossover probability	0.5
Mutation probability	0.4
Mutation range	0.2

Table 6: GA parameters for evolving multi-segmental controllers.

Evaluation Multi-segmental controllers are evaluated for their capacity to control swimming in the mechanical simulation. The required features for the controllers are:

1. generation of stable oscillations in the 100 segments with coordinated phase differences for the creation of traveling undulations of the body,

2. ability to change the speed of swimming by changing either the frequency of oscillation or the wavelength of the undulation,

3. ability to change the frequency and the wavelength independently, by changing, respectively, the global excitation level and the amount of extra excitation on the most rostral segments.

In order to allow comparisons with the biological controller, a special emphasis is given to the capacity to swim with a wavelength corresponding to the length of the body (phase lag per segment of 1%), and therefore the capacity to change the frequency over a large range is only measured for lags close to 1%. The mathematical definition of the fitness function is given in Appendix A2.

Results We carried out 10 evolutions based on the best evolved segmental oscillators of the previous evolutionary stage. Each population is made of 40 chromosomes and evolutions are stopped when the fitness of the fittest solution of the population ceases to increase significantly (between 75 and 300 generations depending on the run).

The random coupling of the initial populations has a different effect depending on the segmental network. Some segmental networks are quite robust and produce stable oscillations despite the random couplings, but most of them fail to produce stable oscillations. A general observation is that, among the randomly connected segmental oscillators which can produce stable oscillations, very few can have the phase between segments varied with the extra excitation on the first segments.

All runs successfully converged to controllers capable of generating stable oscillations with phase lags between segments for relatively large ranges of frequency and phase lags. Similarly to the biological model, the evolved CPGs produce lags between segments which are constant over the whole spinal cord, except for the extremities (because of the lack of extensions from the boundaries, the segments at the extremities tend to oscillate in phase). The lags increase when extra excitation is applied to the most rostral segments.

After the evolutions, we tested the best solutions of each run over a whole range of levels of excitation and extra excitation in order to determine their exact ranges of speed, frequency and phase lag. The results are summarized in Table 7.

The multisegmental controllers cover approximately the same range of frequency as the corresponding isolated segmental networks. Five evolved controllers cover a larger range of phase lags than Ekeberg's solution, with a maximum lag of 3.8% for the solution of run4. The speed range covered by the evolved solutions is also generally larger than the range of Ekeberg's model (8 solutions reach higher speeds). Note that some solutions swim slowly backwards when all segments oscillate in phase (no extra excitation on the first segments), because of the kind of wriggling they then perform. Also, some solutions produce lags even without extra excitation of the first segments.

	Fitness value	Frequency range in [Hz]	Relative lag range in [%]	Speed range in [m/s]
Biol. model		[1.6, 5.6]	[0.0, 2.4]	[−0.03, 0.50]
run1	0.07	[1.9, 8.5]	[0.0, 1.9]	[−0.04, 0.59]
run2	0.09	[1.0, 6.8]	[0.1, 3.1]	[0.06, 0.58]
run3	0.07	[1.1, 7.8]	[0.0, 2.0]	[0.08, 0.49]
run4	0.13	[0.8, 10.1]	[0.0, 3.8]	[−0.03, 0.53]
run5	0.26	[1.1, 8.4]	[0.0, 3.6]	[−0.10, 0.55]
run6	0.02	[1.1, 7.0]	[0.0, 1.3]	[0.07, 0.51]
run7	0.02	[1.2, 7.5]	[0.0, 2.0]	[−0.03, 0.52]
run8	0.10	[1.1, 5.8]	[0.0, 3.0]	[−0.03, 0.56]
run9	0.13	[2.0, 11.0]	[−1.7, 3.5]	[−0.07, 0.50]
run10	0.18	[3.3, 8.7]	[0.3, 2.2]	[0.18, 0.60]

Table 7: Summary of results for the evolved multi-segmental controllers. The table gives, for the best solution of each population, the ranges of frequency, phase lag and speed it can produce.

Although most solutions cover larger ranges of frequencies and phase lags than the biological model, the independence of control of these variables is usually less good. For several solutions, for instance, increasing the global level of excitation not only increases the frequency of oscillations but also influences the relative lag between segments (usually decreasing it). This problem is illustrated by the best solution of run9 which can reach high frequencies and high lags, but not both together, which explains why its maximum speed is relatively low (a combination of high frequency and high lag is necessary for high speeds in our mechanical simulation).[6] Our fitness function has not prevented this relative lack of independence because, although it rewards independence, it does it only for the small area of the global excitation versus extra excitation space that is visited for the evaluation. The dependence occurs mainly at high levels of global excitation and extra excitation which are not visited for the evaluation. This relative problem could probably be solved by sampling more combinations of both types of excitation.

There are no clear conclusions to make from the way the segmental networks are coupled (see Appendix C for a description of the evolved configurations). The extents of the interconnections vary considerably from one solution to another, and there is, for instance, no systematic coupling favouring one direction. The extents of the coupling and the asymmetries vary depending on the types of the connected neurons, the influence of the connection, the phase differences between the activities of the neurons within the segmental oscillator, *etc.* A few general observations can, however, be made. On average,

[6] Note that the best solution of run9 also has the idiosyncrasy of producing, for some levels of global excitation, a caudal to rostral traveling wave when it receives no extra excitation. As soon as some extra excitation is given, the undulation travels again from head to tail.

the ipsilateral connections between neurons of the same type (the self-connections) extend more rostrally than caudally while the contralateral connections between neurons of the same type extend more caudally. Also a large majority of the inhibitory connections sent to the motoneurons extend more caudally. Finally the CPGs based on similar segmental oscillators have also similar couplings, with the same asymmetries of coupling for most segmental connections.

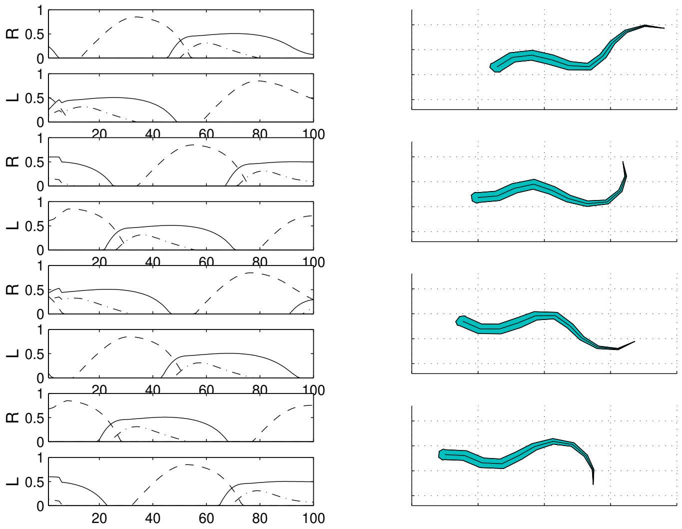


Figure 7: Simulation of one of the evolved swimming controllers (best of run7). The frequency of oscillations is 4.0 Hz, the phase lag between segments is 1.1% of a cycle duration and the speed of swimming is 0.41 m/s. *Left:* Neural simulation, the continuous line correspond to the MN, the dashed line corresponds to the CIN and dotted-dashed line to the EIN. *Right:* Mechanical simulation, there are 60 ms between each snapshot, and the dotted vertical lines are separated by 100 mm.

Figure 7 shows a simulation of one of the evolved multi-segmental controllers. Interestingly, the evolved CPGs can also perform turning although this aspect has not been taken into account in the evaluation functions. Similarly to the biological controller, giving a higher excitation level to one side of the spinal cord leads to a difference of the motoneuron amplitude and, because of stronger muscle contractions on one side, to a change in the direction of swimming. In the case of the evolved controllers, some left-right permutations of the interneurons may be necessary for optimising the turning ability (because of the left-right symmetry, these permutations have no effect on straight swimming). For instance, for avoiding antagonistic effects on the motoneuron amplitude when extra excitation is applied, the neurons sending inhibitory connections to a motoneuron should be on the contralateral side. Finally, initial tests on some of the evolved controllers show that it is possible to produce backwards swimming by giving more excitation to the most caudal segments and therefore inducing a tail to head undulation, but only very limited lags and speeds can then be obtained.

3.1.3 Third evolutionary stage: Sensory feedback from stretch sensitive cells

The CPGs evolved so far receive no feedback from the mechanical simulation. In the real lamprey, sensory feedback is provided to the CPG by stretch sensitive cells (the edge cells) situated on both side of the spinal cord. There are both inhibitory and excitatory edge cells and they emit signals proportional to their elongation [Viana Di Prisco *et al.* 90]. Ekeberg showed [Ekeberg *et al.* 95] that this sensory feedback is necessary for crossing a barrier of water with local speed opposite to the direction of swimming. We show here a preliminary experiment on how to evolve this sensory feedback for the artificial controllers in order to allow them to cross a similar speed barrier.

Encoding and GA A chromosome encodes how each segmental oscillator receives sensory feedback from two stretch sensitive cells situated at the sides of the corresponding part of the spinal cord. It encodes the synaptic weights of the connections from one edge cell to the 8 (or 6) neurons of the corresponding segmental network. We impose symmetry between both sides of the segmental network as well as over the whole spinal cord (the 100 segments). Each connection can be either excitatory or inhibitory and genes are decoded into synaptic weights between -2 and 5. Note that we have fixed these boundaries to be relatively small in order to prevent the sensory feedback disturbing the production of oscillations. The same GA with the same parameters as for the first evolutionary stage is used (Table 4), but without the pruning operator.

Evaluation Solutions are rewarded depending on their ability to cross a speed barrier. The barrier is 150 mm wide (half the length of the simulated lamprey) and the speed of the water flow opposite to direction of swimming is set to be 40% higher than the speed of swimming of the lamprey. The fitness function is defined to reward:

1. progression through the barrier in the direction of swimming,
2. minimal deviation in the direction,
3. minimal difference of speed between swimming with and without sensory feedback.

This last point was added to prevent important changes in the swimming patterns when sensory feedback is given to the controller. Evaluations are realised at a chosen combination of frequency of oscillations and phase lag between the segments (approximately in the middle of their respective range). The mathematical definition of the fitness function is given in Appendix A3.

Results We carried out two sets of 5 runs (populations of 40 chromosomes) using the best controller of run7 of the previous evolutionary stage[7]. In the first set of runs, sensory feedback is evolved to all 6 neurons forming the segmental oscillator (2 EINs, 2 CCINs and 2 MNs), while in the second set sensory feedback is only provided to the interneurons. The chosen levels of excitation produce swimming (without sensory feedback) at a frequency of oscillations of 4.0Hz, a relative lag of 1.1% and a corresponding speed of 0.41 m/s. The speed of the barrier was set to 0.57 m/s (40% higher than the speed of swimming). Without sensory feedback, the lamprey is not able to cross the speed barrier (Figure 8). The increase in local forces at the entrance to the barrier leads to extra bending of the upper part of the body and forces the lamprey to change its direction of swimming. The lamprey eventually swims perpendicularly to its initial direction.

In the first set of runs (sensory feedback to both interneurons and motoneurons), all runs converged within 80 generations to controllers able to cross the speed barrier (see Appendix D for a description of the evolved configurations). The connections from the stretch sensitive cells are not identical in all the evolved controllers. All solutions have, however, evolved excitatory connections from the stretch sensitive cells to the interneurons inhibiting the motoneurons on the contralateral side, therefore preventing excessive

[7]This controller was chosen because it combines relative large ranges of frequencies, lags and speeds and an independence of control of the frequency and the lag similar to the biological model.

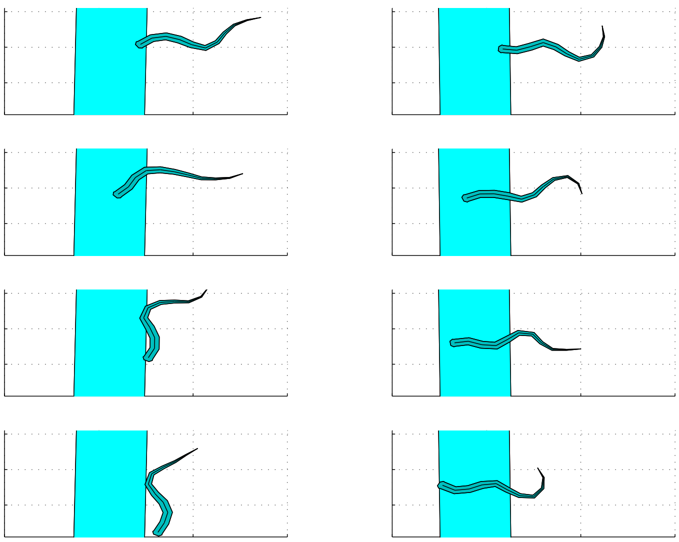


Figure 8: Effect of a speed barrier on swimming without sensory feedback.

bending. The solutions have also developed strong excitatory connections from the stretch sensitive cells to the motoneurons. Interestingly, these connections are not only ipsilateral (to prevent excessive bending) but also contralateral. The overall effect is that the amplitudes of the motoneuron signals are approximately 20% larger than without sensory feedback. The excitatory feedback to the motoneurons leads thus to a stiffer body of the lamprey, which is therefore less sensitive to the extra forces due to the barrier.

In the second set of runs, this direct feedback from the stretch sensitive cells to the motoneuron is not allowed. In a first try with the speed barrier of 0.57m/s, none of the runs converged (within 50 generations) to controllers capable of crossing the barrier. The speed was therefore reduced to 0.53m/s (30% larger than the speed of swimming without sensory feedback). After 50 generations, 2 of the 5 runs converged to solutions capable of crossing the barrier[8]. The controllers of the two successful runs are able to cross the speed barrier without increase of the amplitude of the motoneurons. These controllers therefore manage to cross the barrier without increasing the stiffness of the body but only by correcting the phases of the motoneuron oscillations. The best controllers of both runs have, for instance, stretch sensitive cells which inhibit the inhibitory interneuron connected to the motoneuron on the same side, and which excite the inhibitory neuron connected to the motoneuron on the contralateral side. The stretch senstive cells therefore act as burst terminators which tend to switch the motoneuron activity from one side to the other when the latter is too much extended.

3.2 GAs as tool for neurobiological modeling

The method presented previously can also be used to develop neurobiological connectionist models. We will here restrict the space of possible solutions to controllers which preserve the segmental connectivity and the type of neurons observed in the real lamprey. Given the desired activity of the network, the neuron models and the segmental connectivity, the GA is used to determine the synaptic strengths and the extents of interconnections. The evolutionary process will thus define a significant part of the connectionist model that Ekeberg has set up by hand.

[8] The three unsuccessful runs had a premature convergence to controllers which, although they crossed most of the barrier, were unable to cross it completely. The loss of diversity in the chromosomes meant that the GA could not improve those local maxima.

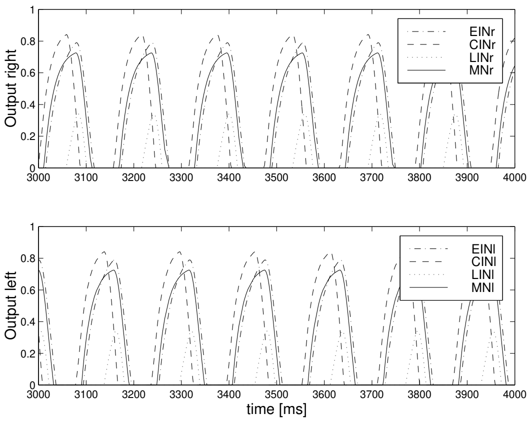
3.2.1 Segmental oscillator

Segmental oscillators with the biological connectivity are evolved. The same GA (with the same parameters, as given in Table 4) for evolving artificial segmental oscillators is used with the exception that the chromosome only encodes the connections observed in the real lamprey, i.e. the 26 connections of Ekeberg's model, and that no pruning operator is used. The type of each connection (excitatory or inhibitory) is also fixed and corresponds to the physiological observation. Ten evolutions with populations of 100 chromosomes were carried out for 500 generations.

	Fitness value	Frequency range in [Hz]
Biol. model	0.18	[1.7, 5.6]
run1	0.58	[1.0, 8.6]
run2	0.70	[0.9, 10.5]
run3	0.63	[0.9, 9.5]
run4	0.56	[1.0, 8.3]
run5	0.71	[0.9, 10.8]
run6	0.71	[1.0, 10.3]
run7	0.54	[1.4, 11.6]
run8	0.71	[0.9, 11.0]
run9	0.67	[0.9, 9.8]
run10	0.70	[1.0, 10.6]

Table 8: Summary of results for the evolved segmental oscillators with biological connectivity.

All populations converged to best solutions covering larger ranges of frequency than the segmental oscillator with Ekeberg's values (Table 8). These evolved segmental oscillators cover most of the observed frequencies of the real lamprey (between 0.25 and 10 Hz), with, for instance, the solution with highest fitness oscillating between 0.9 and 11.0 Hz. Figure 9 shows the simulation and the configuration of that best oscillator. None converged to the same set of connection weights as Ekeberg's segmental oscillator. Interestingly, five other runs converged to a network similar to that of the best run, with very similar weights (almost identical inhibitory weights) and similar fitness values. The corresponding segmental network seems thus to be an important local fitness maximum.



from: to:	EINl	CINl	LINl	EINr	CINr	LINr	BS
EINl	2.4	-	-	-	−2.3	-	9.6
CINl	10.0	-	−4.1	-	−1.3	-	10.3
LINl	4.2	-	-	-	−4.1	-	6.6
MNl	-	-	-	-	−1.8	-	3.4
EINr	-	−2.3	-	2.4	-	-	9.6
CINr	-	−1.3	-	10.0	-	−4.1	10.3
LINr	-	−4.1	-	4.2	-	-	6.6
MNr	-	−1.8	-	-	-	-	3.4

Figure 9: Simulation and configuration of the best evolved oscillator with biological connectivity. This oscillator can oscillate at frequencies between 0.9 and 11.0 Hz. Five (out of ten) other runs converged to very similar configurations.

A general observation from these runs is that the increase of the frequency range compared to the original segmental oscillator is due to stronger inhibitory links, especially the inhibitory connections from CIN to contralateral LIN and from LIN to ipsilateral CIN (see Appendix E for a description of the evolved configurations). It also seems that the excitatory connection from EIN to ipsilateral MN can be removed without disturbing the oscillatory capacities of the network (see the configuration of Figure 9, for instance). Interestingly, one of the evolved segmental networks had non-oscillating EINs showing that, even with the biological connectivity, it is possible to remove the excitatory interneurons and still produce oscillations over a large range of frequency. A similar observation was made by Jung [Jung *et al.* 96], who observed that, in a connectionist model similar to this one, with appropriate changes in the tonic drives to the LIN and CIN neurons, stable oscillatory output could be obtained even after eliminating the EIN neurons.

This experiment shows that the GA can successfully evolve segmental oscillators which are as biologically plausible as Ekeberg's oscillator (same connectivity and same neural behaviour) but with a significantly larger range of oscillation frequency, much closer to that observed in the real lamprey.

3.2.2 Multi-segmental controller

We have also evolved the interconnectivity between fixed segmental oscillators corresponding to the segmental network of Ekeberg's biological model. The existing interconnections between segments in the real lamprey are not perfectly known yet and there is some controversy in the mechanisms which lead to phase lags which are constant over the spinal cord and independent of the frequency of oscillations (see discussion). Evolving the couplings between the biological segmental network may therefore give some insight to which kind of interconnections can produce this type of phase lag.

	Fitness value	Frequency range in [Hz]	Relative lag range in [%]	Speed range in [m/s]
Biol. model		[1.6, 5.6]	[0.0, 2.4]	[-0.03, 0.50]
run1	0.22	[1.7, 5.5]	[0.0, 3.5]	[0.06, 0.53]
run2	0.21	[1.7, 5.6]	[0.0, 3.5]	[0.07, 0.53]
run3	0.22	[1.7, 5.5]	[0.0, 3.4]	[0.07, 0.53]
run4	0.21	[1.7, 5.6]	[0.2, 3.5]	[0.06, 0.53]
run5	0.22	[1.7, 5.5]	[0.0, 2.6]	[0.09, 0.52]

Table 9: Summary of results for the evolution of multi-segmental controllers with Ekeberg's segmental network.

We use the same methods as for the second evolutionary stage. The fixed segmental oscillator corresponds to Ekeberg's segmental network and a chromosome encodes the extensions to neighbouring segments of the segmental connections. We realised 5 runs with populations of 40 chromosomes. All five runs converged, in less than 75 generations, to complete controllers with similar performances to Ekeberg's biological model (see Table 9). The solutions cover larger ranges of lags and can reach slightly higher speeds. The best solution, for instance, can have its lag per segment varied between 0.0 and 3.5% of the oscillation period[9] and reach a speed of 0.53 m/s. Similarly to Ekeberg's model, the frequency of oscillation depends mainly on the global level of excitation and the lag between segments is mainly controlled by the extra excitation on the five first segments. Finally, Ekeberg's model and these five solutions all have a very similar relation between the speed of swimming and the levels of global excitation and extra excitation.

[9]Phase lags in an isolated spinal cord of a real lamprey can be varied between approximately -3% to +3% by varying local concentration of excitatory bathes [Matsushima & Grillner 92].

from: / to:	EINl	CINl	LINl	EINr	CINr	LINr	BS
EINl	0.4 [5, 3]	-	-	-	-2.0 [1, 0]	-	2.0
CINl	3.0 [11, 2]	-	-1.0 [3, 9]	-	-2.0 [0, 5]	-	7.0
LINl	13.0 [4, 1]	-	-	-	-1.0 [9, 4]	-	5.0
MNl	1.0 [4, 1]	-	-	-	-2.0 [11, 11]	-	5.0
EINr	-	-2.0 [1, 0]	-	0.4 [5, 3]	-	-	2.0
CINr	-	-2.0 [0, 5]	-	3.0 [11, 2]	-	-1.0 [3, 9]	7.0
LINr	-	-1.0 [9, 4]	-	13.0 [4, 1]	-	-	5.0
MNr	-	-2.0 [11, 11]	-	1.0 [4, 1]	-	-	5.0

Table 10: Evolved interconnections between biological segmental network (best of run2). The table gives the weights of the segmental network (identical to Ekeberg's) and, between brackets, the extensions of the evolved interconnections in the rostral and caudal direction respectively.

These solutions have all similar intersegmental couplings and are significantly different from that of Ekeberg's model in which only the CIN have asymmetric projections (favouring the caudal direction, *cf.* Table 1)(see Appendix F for a description of the evolved configurations). Table 10 gives the interconnections of one of them. Although the exact extents of the interconnections vary from one solution to another, they present strong similarities in the rostral/caudal asymmetries of interconnections. For instance, all solutions have asymmetries favouring the caudal direction for the ipsilateral LIN to CIN and contralateral CIN to CIN connections. They also have asymmetries favouring the rostral direction for the ipsilateral EIN to EIN, ipsilateral EIN to CIN and contralateral CIN to EIN connections. The fact observed with the artificial controllers that connections between neurons of the same type extend more caudally if the connection is contralateral and more rostrally if the connection is ipsilateral (self-connection) is thus also true here. It is interesting to compare the evolved interconnections with the couplings observed so far in the real lamprey. As summarized in [Wadden *et al.* 97], it has been observed that LINs have long caudal projections (up to 50 segments), CINs project mainly caudally (at least 14 segments) with some rostral extensions, and EINs have relatively short projections (between 5 and 7 segments) in both directions. The respective extents of the extensions relatively to the different neurons of a segmental network are not known. The evolved controllers have interconnections which respect to some extent these observations. For the solution given in Table 10, for instance, only the long rostral projections of the EIN to CIN, the CIN to LIN and CIN to MN strongly disagree with the biological observations. The others correspond to the observed biological interconnections if we hypothesize that the projections observed in the real lamprey from one type of neuron not necessarily target all neuron types in other segments in an identical way. For instance, although CIN neurons have globally long caudal projections and short rostral projections, it may be that their projections to EINs are mainly rostral and that the existing caudal projections do not target EINs.

Note that not all couplings produce the desired phases between segments. For instance, a symmetric closest neighbour coupling (each connection extended to the upper and lower neighbour segment) leads to phases between segments which depend on the global level of excitation and not the extra excitation as would be desired (increasing the extra excitation on the first segments influences the lag only locally). Another example is asymmetric closest neighbour coupling with extension only to the caudal(rostral) direction in which case only a temporary phase lag between segments occurs which progressively disappears from head to tail(tail to head) until all segments oscillate in phase.

3.2.3 Sensory feedback

Similarly to the artificial controllers, it is possible to evolve sensory feedback from stretch sensitive cells to the segmental oscillators. We realised a set of 5 runs (populations of 40 chromosomes, 100 generations) which evolved sensory feedback to the 6 interneurons and the 2 motoneurons of Ekeberg's segmental network. The speed of the speed barrier was fixed to be 40% higher than the speed of swimming without sensory feedback.

	Evolved		Biological	
	ECl	ECr	ECl	ECr
EINl	−1.0	5.0		
CINl	2.4	−0.9	+	−
LINl	0.8	−1.0	+	−
MNl	3.9	3.4	+	

Table 11: Comparison between the established biological sensory feedback [Viana Di Prisco *et al.* 90] and that evolved by the best run. EC stands for Edge Cell, and l and r indicate the left and right side of the spinal cord. The biological feedback comes from both inhibitory and excitatory edge cells. The signs of the established connections are given by + and −.

All runs converged to controllers capable of crossing the speed barrier. The solutions vary in the exact value of the weights but are almost identical in the respective signs of the connections (of the 8 possible connections, 5 connections have the same sign in all runs and the 3 others have the same sign in all runs except one)(see Appendix G for a description of the evolved configurations). Similarly to the corresponding runs with an artificial controller, they all developed strong excitatory connections from the stretch sensitive cells to the motoneurons. It is interesting to note that the evolved sensory feedback pathways correspond very closely to those observed in the real lamprey. It has been shown that there exist both inhibitory and excitatory edge cells in the lamprey with the inhibitory edge cells projecting contralaterally and the excitatory cells projecting ipsilaterally [Viana Di Prisco *et al.* 90]. Table 11 compares the evolved connections of the best run with the connections established with paired intracellular recordings. For all the established biological connections, the evolved controllers have developed sensory feedback connections with the same sign. Note that signs of the evolved connections from the edge cells to the EINs and LINs are somewhat counterintuitive because these connections counteract the effect of the other connections which tend to switch the neural activity to the side which is excessively extended.

4 Discussion

By developing efficient CPGs which can produce coordinated oscillations for the swimming of a simulated lamprey, the GA has proved to be an interesting design technique. We will here review the method and discuss the evolved controllers.

4.1 Discussion of the Method

Controllers were evolved in different stages in order to simplify the problem by decomposing it into subproblems. The decomposition was motivated by similar decompositions in the real lamprey and gave the possibility of direct comparison between evolved solutions and the biological model. One advantage of such a decomposition is to reduce the time needed to evolve a complete controller by avoiding evaluation of a whole controller made of segmental oscillators which do not oscillate correctly even when isolated. Decomposition presents the risk, however, of misleading the evolution as the fittest solutions of one stage may not necessarily be the best starting elements for the next stage. The segmental oscillators evolved in our first stage, for instance, have not been rewarded on their capacity to be interconnected, which may have constituted a handicap for the next stages.

The simple encoding scheme which directly encodes synaptic weights and extents of interconnections into a chromosome meant that search was performed in a well defined and limited space of possible solutions (fixed neuron models, fixed maximum number of neurons and left-right symmetry). Despite these limitations, the multidimensional search space was still large enough to generate a variety of interesting Central Pattern Generators. One way to enlarge the search space could be to include the parameters of the neuron dynamics into the encoding. Initial tests show that the GA is able to evolve both these neuron parameters and the connectivity and generate working CPGs. Another interesting extension would be to use an indirect encoding, such as a developmental program, which would allow modularity and not limit the number of neurons (see [Kodjabachian & Meyer 95] for a review).

Defining the fitness functions probably constitutes the most important part of the method. Our objectives were to develop CPGs capable of producing stable oscillations and of modulating the speed of swimming by varying, through a few external signals, the frequency and the phase lag of the oscillations. Once the objectives were defined, relatively little iteration was needed to define a suitable set of fitness functions. The functions are products of factors between 0 and 1 rewarding qualitative and quantitative aspects, which means that solutions which perform evenly in all aspects receive the highest fitness value. Both neural (e.g. the MN frequency) and mechanical (e.g. the speed of swimming) variables are rewarded, with the importance of neural aspects gradually decreasing and those of mechanical aspects increasing over the evolutionary stages. Ideally, only mechanical aspects should be examined as they alone effectively determine the quality of a swimming controller; rewarding neural activities presents the risk of biasing the search to particular network architectures. However, rewarding only mechanical variables would not have allowed the staged evolutionary approach discussed above.

The fact that the GA was able to find sets of synaptic weights and interconnections for CPGs under the constraints of the biological observations showed that it has the potential to be a useful tool for developing neurobiological models. Although it is possible, in this case, to define satisfactory values by hand (as Ekeberg did), the GA proved useful for searching the different possibilities and to optimize the unknown values for specific criteria, such as the range of frequencies for instance. From a general point of view, note that, although the GA can be very useful for demonstrating that a model can produce a specific behaviour (by finding efficient sets of unknown variables), it is less useful for invalidating a hypothetical model as an inability to find successful variable instantiations may be due to failings of the model or to problems with the GA set up, or both.

The main advantage of using a GA for the design of neural controllers is its ease of implementation and use. There are, for instance, no mathematical restrictions on the fitness function. Unlike most error functions minimized by learning algorithms for dynamical neural networks (like variations of the backpropagation algorithm, see [Pearlmutter 95], for instance), the fitness function does not need to be differentiable or even continuous. There is also no need to produce a specific oscillation (limit cycle) that the network should learn. This has allowed us to evolve controllers which produce oscillations only qualitatively specified. Constraints on the generated solutions can easily be added or removed; the type of evolved controllers can be modified at three levels: the encoding (e.g. by adding or removing constraints on the connectivity), the GA operators (e.g. adding a pruning operator) or the fitness function (e.g. adding or removing factors). Another interesting aspect of the GA is the parallel search of the parameter space it performs (each chromosome is in a different place in that space), reducing the risk of convergence to local maxima. This parallel search and the stochastic nature of GAs makes them particularly suitable for visiting the space of possible solutions and designing, potentially, a variety of solutions.

4.2 Discussion of the evolved controllers

Several interesting CPGs have been successfully evolved. A first observation is thus that there exists a variety of possible connectivities which produce the neural activities necessary for swimming.[10] We have demonstrated that there exist solutions alternative to the biological CPG which can control swimming with the same efficiency. These solutions differ from Ekeberg's biological model in terms of which neurons are interconnected, of the excitatory or inhibitory influence of the interneurons and even of the number of neurons involved in the rhythmogenesis.

We have evolved a diversity of artificial segmental oscillators with significantly larger ranges of frequencies than Ekeberg's segmental oscillator. Although one oscillator has a similar structure to the biological one, most evolved networks have interneurons that play other roles than in the biological network and have other signs. There is therefore no preferred role and sign for each dynamics of the different neuron models. A majority of oscillators are based on only four inhibitory interneurons, showing that, at least at the connectionist level, excitatory interneurons are not necessary for the production of oscillations over large frequency ranges.

The evolution of multi-segmental oscillators showed that segmental networks could be interconnected in several ways and produce stable oscillations with the necessary phase lags for swimming. Similary to the biological model, swimming at different speeds and with different wavelengths of undulations can be produced by varying simple external signals (the excitation applied to the CPG). Turning can also be induced when one side of the controller receives more excitation than the other. The evolved controllers can generally cover larger ranges of frequency, phase and speed than Ekeberg's model but with less independence in the control of the frequency and the phase between segments. A general observation on the coupling between segments is that there is no systematic asymmetry of interconnection favouring one direction but rather asymmetries in both directions which depend on the segmental connection. Note that the coupling of both Ekeberg's and the evolved controllers is a little bit crude in the sense that, except for the segments at the extremities, the synaptic weights of the extensions of a segmental connection are the same for all connected segments. It could be interesting to evolve, for each coupling, a normalising function which would determine the synaptic strengths depending on the length of the projection, in order to allow, for instance, a decrease of the influence of a segment on another with the distance separating them.

The evolutions of connections from stretch sensitive cells illustrates how sensory feedback can be used by the evolved CPGs for coordinating the neural activity with the actual movements of the body and allowing the lamprey to cross a speed barrier. The feedback increases the stiffness of the body and acts as a burst terminator which switches the neural activity to the side which is excessively bent. These preliminary experiments were realised with fixed external commands to the CPG (swimming straight) and the crossing of the barrier was therefore due to the passive properties of the CPG with sensory feedback. As active corrections in the commands may also help the lamprey to cross the barrier,

[10] A similar observation was made by Lockery for the leech bending reflex, where many different networks could produce a physiologically accurate local bending input-output function [Lockery & Sejnowski 93].

further experiments should include these in order to determine the respective importance of sensory feedback and higher commands from the brain for crossing a barrier.

The development of controllers which preserve the biological segmental connectivity has led to several observations which may interest neurobiologists. Firstly, the generation of segmental oscillators with the biological connectivity shows that Ekeberg's set of synaptic weights can be modified in order to optimize the frequency range. Frequency ranges much closer to those observed in real lamprey (between 0.25 and 10 Hz) can be obtained. The main observation is that the increase of the frequency range is obtained through an increase of the strength of some inhibitory connections (the contralateral CIN to LIN and ispilateral LIN to CIN connections). We also find, in agreement with Jung [Jung *et al.* 96], that the biological connectivity can produce oscillations without the excitatory neurons (provided that the other synaptic weights are readjusted). Moreover, we demonstrate here that removing the excitatory neurons does not prevent the biological connectivity from being able to cover a large frequency range of oscillations.

We have also evolved the couplings between the biological segmental networks which can produce phase lags which are constant over the spinal cord and independent of the frequency of oscillations, as observed in the real lamprey. There are opposing views on the origin on these phase lags in the real lamprey, in particular whether they are due to differences in the intrinsic frequencies of the oscillators or to the nature of the coupling, and, in the latter case, whether the dominant coupling is ascending or descending (see for instance [Matsushima & Grillner 92, Wadden *et al.* 97, Kopell 95, Williams & Sigvardt 95]). Here, at a connectionist level, we have developed CPGs which rely both on asymmetries of the couplings and on differences of intrinsic frequencies for creating travelling waves. The coupling is optimized for waves to travel from head to tail and the extra excitation of the most rostral segments determines the exact phase lag. Extra excitation of the first segments amounts to an increase of the intrinsic frequency of those segments which therefore tend to lead the others. As predicted theoretically for abstract oscillators, the higher the intrinsic frequency difference, the larger the phase difference between all segments. The optimization of intersegmental coupling for head to tail propagation is illustrated, firstly, by the difficulty of inducing backwards propagation when the most caudal segments receive extra excitation, and secondly, by the fact that some controllers spontaneously propagate waves from head to tail even without extra excitation. The evolved CPGs seem to have no dominant coupling (for instance, no systematic asymmetry) and the interconnections between segments combine both caudal and rostral asymmetries of couplings. Also, there seems to be no simple way of coupling segmental networks and obtaining the independence between the wavelength of the undulation and the frequency of oscillations. A simple closest neighbour coupling (symmetric or asymmetric), for instance, can not produce that feature. Finally, the evolved couplings are, to some extent, similar to those observed in the real lamprey if we hypothesize that the observed projections from one neuron type in one segment to neurons in other segments represent the total of all projections from that neuron and do not necessarily mean that the projections to different types of neurons must be identical.

Interestingly, the evolution of sensory feedback to Ekeberg's controller led to connections which correspond very closely to those observed in the real lamprey, with all connections from excitatory and inhibitory stretch sensitive cells established in the real lamprey being also present with the same sign in the evolved controllers. Crossing a speed barrier may therefore be a good example of the situation for which sensory feedback has been developed through natural evolution for the real lamprey.

5 Conclusion

This paper has described how a Genetic Algorithm can be used to develop connectionist neural networks for the control of anguiliform swimming. Inspired by Ekeberg's connectionist model of the Central Pattern Generator of the lamprey, alternative artificial solutions have been successfully evolved. This work shows firstly that the Genetic Algorithm can be an interesting design technique for dynamical recurrent networks applied to locomotion control. With the technique, we have evolved networks which efficiently control the swimming of a simulated lamprey by producing stable oscillations whose frequencies and phase lags can be modulated by simple external signals. This work also illustrates that the biological connectivity is not unique and that it is possible to connect the interneurons of Ekeberg's model differently and obtain other efficient solutions. In particular, alternative solutions which cover larger frequency, lag and speed ranges can be obtained. Finally, we show how the same technique for evolving artificial solutions could potentially be used as a tool for developing neurobiological models. By incorporating biological knowledge of the real lamprey into the encoding and the evaluation function of the Genetic Algorithm, the evolutionary process can be used to find variables which are difficult to determine, in this case synaptic weights, for the production of a desired behaviour. Solutions which satisfy the biological constraints and produce oscillations over a larger range of frequency than the original biological model have, for instance, been successfully evolved.

6 Acknowledgements

Auke Jan Ijspeert's PhD is partly supported by grants from the Swiss National Research Fund and from the European Commission (Marie Curie Fellowship). Facilities were provided by the University of Edinburgh. Many thanks to Aude Billard and Richard Reeve for comments on earlier versions of the manuscript.

A Fitness functions

A.1 Fitness of segmental oscillators

$$Fitness = \mathit{fit\_oscil} \cdot \mathit{fit\_connectivity} \cdot \mathit{fit\_freq} \in [(0.05)^5, 1]$$

where

$$\mathit{fit\_oscil} = \mathit{fit1} \cdot \mathit{fit2} \cdot \mathit{fit3} \in [(0.05)^3, 1]$$

$$\mathit{fit\_connectivity} = \begin{cases} -1.357 \cdot \mathit{con\_ratio} + 1.407 & \text{if } \mathit{con\_ratio} \in [0.3, 1] \\ 0.05 & \text{if } \mathit{con\_ratio} = 1 \\ 1 & \text{if } \mathit{con\_ratio} < 0.3 \end{cases}$$

$$\mathit{fit\_freq} = \begin{cases} 0.05 \quad \text{if } \mathit{fit\_oscil} < 0.45, \text{ otherwise:} \\ \begin{cases} 0.0864 \cdot \frac{\mathit{max\_freq}}{\mathit{min\_freq}} - 0.0364 & \text{if } \frac{\mathit{max\_freq}}{\mathit{min\_freq}} \in [1, 12] \\ 0.05 & \text{if } \frac{\mathit{max\_freq}}{\mathit{min\_freq}} = 1 \\ 1 & \text{if } \frac{\mathit{max\_freq}}{\mathit{min\_freq}} > 12 \end{cases} \end{cases}$$

The function *fit_oscil* is made of three factors which reward activity of the motoneurons which is varying (*fit1*), regular (*fit2*) and out of phase between left and right motoneurons (*fit3*). These factors are functions of statistical measures of the motoneuron signals (such as the standard deviation of the neuron output, the number of zeros, the difference of area under the output curves,...). They are bounded between 0.05 (bad behaviour) and 1 (good behaviour) and vary linearly between these values for the corresponding variables. The bounds for each variable have been determined by hand on 40 examples of different behaviours from initial experiments. It is possible to fix these bounds such that *fit_oscil* clearly makes the difference between interesting solutions and the others. A limit of 0.45 is thus determined above which a solution is certain to oscillate regularly with opposite behaviour between left and right motoneurons.[11] The function *fit_connectivity* rewards solutions with reduced connectivity. It is bounded between 0.05 and 1 and varies linearly between these values depending on the connectivity ratio (*con_ratio*), the ratio between the number of connections and the maximum number of possible connections (56). The smaller the connectivity the higher the reward, with a maximum reward of 1 for solutions with a connectivity ratio smaller than 0.3 (fewer than 18 connections). The factor *fit_freq* rewards solutions which can oscillate over a large range of frequencies when the excitation from the brain stem is varied. The range of frequencies is only measured if the solution oscillates regularly and with opposite behaviour between left and right (*fit_oscil* > 0.45) at the default level of excitation. The frequency range, the ratio between maximum and minimum frequencies, is measured by making a set of simulations at different levels of excitation. Only ranges in which the frequency and the amplitude of the motoneuron signals increase monotonically with the excitation level are rewarded.[12] Note that because the duration of a simulation is fixed and that a minimum of oscillations is required for verifying the stability of the signals, the lowest measurable frequency is approximately 1.0 Hz.

[11] Of the 40 examples of neural activities, 17 correspond to interesting behaviours. With the chosen fixed bounds of *fit1, fit2* and *fit3*, all the good behaviours have a *fit_oscil* value higher than 0.6 and all the others have a value lower than 0.25, with most lower than 0.1.

[12] The condition that the amplitude of the neural activity should not decrease when the frequency increases is added to prevent an antagonistic effect on the speed of swimming, as tests with the mechanical simulation showed that, in general, increases of the frequency or the amplitude of the motoneuron signals increase the speed of swimming.

A.2 Fitness of multi-segmental controllers

$$Fitness = min\_fit\_oscil \cdot fit\_lagcontrol \cdot fit\_freqcontrol \cdot fit\_speed$$

where

- *min_fit_oscil* is the minimum *fit_oscil* value between segments 1,10,20,..,100 when the controller is simulated (simulations of 3000 ms) with an excitation level of *excit0*, corresponding to the middle of the range of excitations with which the segmental oscillator can oscillate, and without extra excitation on the first segments.

- $fit\_lagcontrol = \begin{cases} 0.05 + \frac{lag\_range1}{1+freq\_range1} & \text{if} < 1 \\ 1 & \text{otherwise} \end{cases}$
  Lag_range1 and *freq_range1* are measured by making several simulations at a fixed level of excitation (*excit0*) and with an increasing amount of extra excitation. The lag range is non-zero only if the oscillations are regular in all segments (*min_fit_oscil*>0.45) and if the lag increases monotonically with the amount of extra excitation. The lag range is divided by the frequency range in order to reward solutions which can change the lag between segments independently of the frequency.

- $fit\_freqcontrol = \begin{cases} 0.05 + \frac{freq\_range2}{1+lag\_range2} & \text{if} < 1 \\ 1 & \text{otherwise} \end{cases}$
  Freq_range2 and *lag_range2* are measured by making several simulations with excitation levels varying around *excit0* and with a fixed amount of extra excitation *extra0*. The value *extra0* is calculated from the previous lag measurement if *lag_range1* includes a lag of 1%. *Extra0* is then defined by taking the value of extra excitation corresponding to the lag closest to 1%. The frequency range is non-zero only if the value *extra0* exists, and if the frequency increases monotonically with the level of excitation.

- $fit\_speed = \begin{cases} 0.05 + speed\_range & \text{if} < 1 \\ 1 & \text{otherwise} \end{cases}$
  Speed_range corresponds to the range of speeds covered by all the simulations made for the definition of *lag_freqcontrol* and *fit_freqcontrol*.

The different value ranges are measured as the difference between the maximum and the minimum value and are normalised by a value corresponding to a target value. The targets have been fixed to 2.5% for the lag ranges, 10 Hz for the frequency ranges and 0.8 m/s for the speed range.

A.3 Fitness for sensory feedback

$$Fitness = fit\_progression \cdot fit\_deviation \cdot fit\_speed\_constancy$$

where

- *fit_progression* varies linearly between (and is limited to) 0.05 and 1 when the maximum progression of the head of the lamprey varies between 0 and 450 mm (width of the speed barrier plus the body length). The progression is measured as the distance (in the direction of swimming) between the head of the lamprey and the entry point to the barrier,

- *fit_deviation* varies linearly between 0.05 and 1 as the cosine of the deviation angle (measured between the initial direction of swimming and the direction after entry into the barrier) varies between 0 and 1,
- *fit_speed_constancy* varies linearly between 0.05 and 1 when the relative difference between the speeds with and without sensory feedback varies between 50% and 0%.

B Artificial solutions: evolved segmental oscillators

	EINl	CCINl	LINl	EINr	CCINr	LINr	BS
EINl	-	8.6	-3.8	-4.3	-	-0.1	-0.1
CCINl	-	7.8	-2.9	-0.5	-	-	2.9
LINl	-	10.3	-	-4.9	6.8	-1.5	-
MNl	-0.7	-	-	-4.6	-	-0.8	10.2
EINr	-4.3	-	-0.1	-	8.6	-3.8	-0.1
CCINr	-0.5	-	-	-	7.8	-2.9	2.9
LINr	-4.9	6.8	-1.5	-	10.3	-	-
MNr	-4.6	-	-0.8	-0.7	-	-	10.2

Table 12: **Evolved artificial segmental oscillator**, best of run1

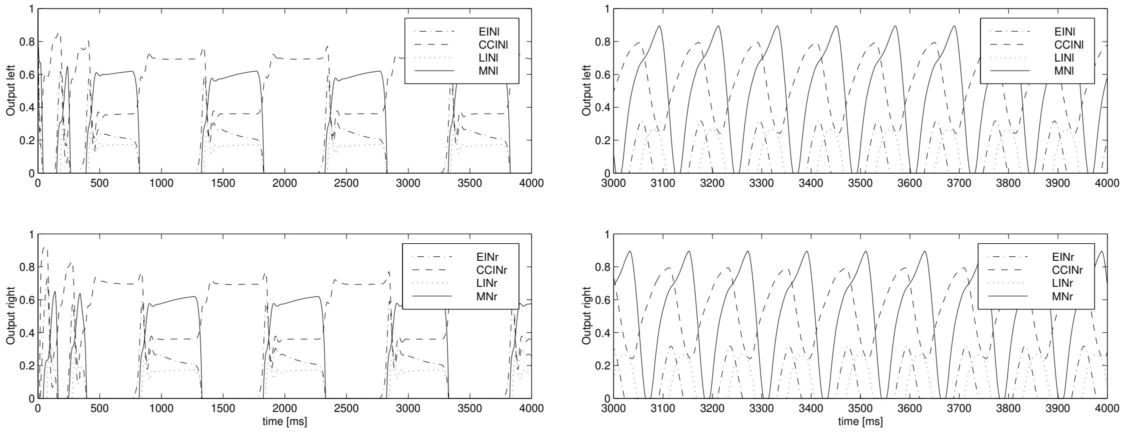


Figure 10: **Run1:** (a) Lowest frequency, (b) Highest frequency

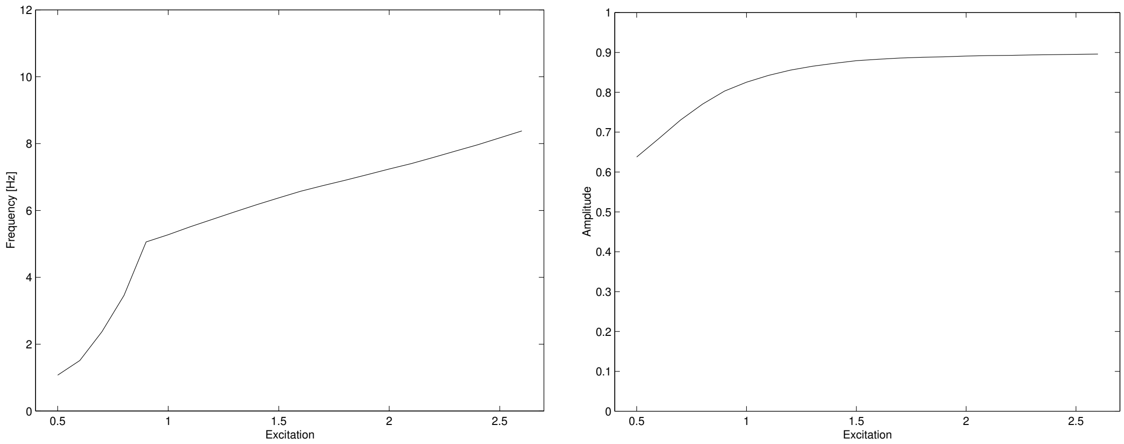


Figure 11: **Run1:** (a) Frequency vs excitation (b) Amplitude vs excitation

	EINl	CCINl	LINl	EINr	CCINr	LINr	BS
EINl	-	-	-	-2.2	-	-4.6	4.3
CCINl	-	-	-	-	-	-	-
LINl	-3.3	-	-0.5	-0.4	-	-1.7	15.0
MNl	-0.2	-	-0.3	-4.4	-	-	11.3
EINr	-2.2	-	-4.6	-	-	-	4.3
CCINr	-	-	-	-	-	-	-
LINr	-0.4	-	-1.7	-3.3	-	-0.5	15.0
MNl	-4.4	-	-	-0.2	-	-0.3	11.3

Table 13: **Evolved artificial segmental oscillator**, best of run2

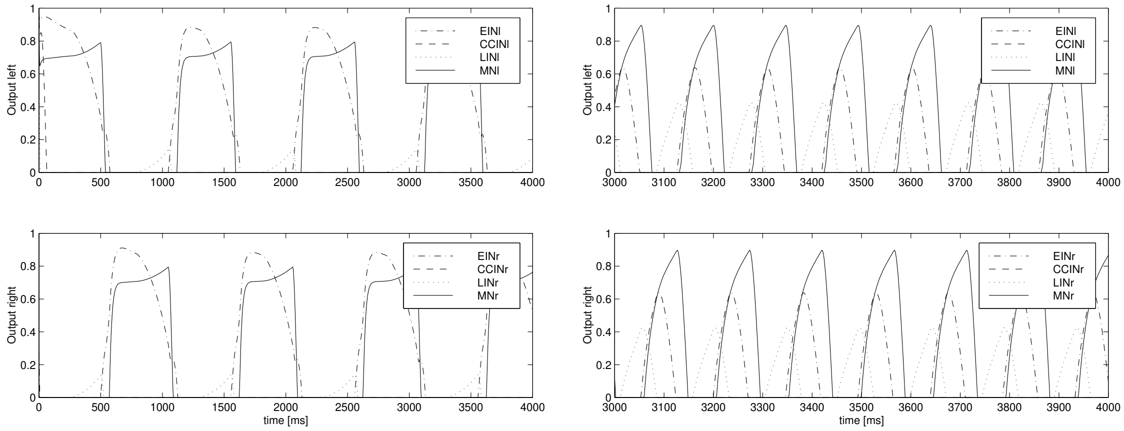


Figure 12: **Run2:** (a) Lowest frequency, (b) Highest frequency

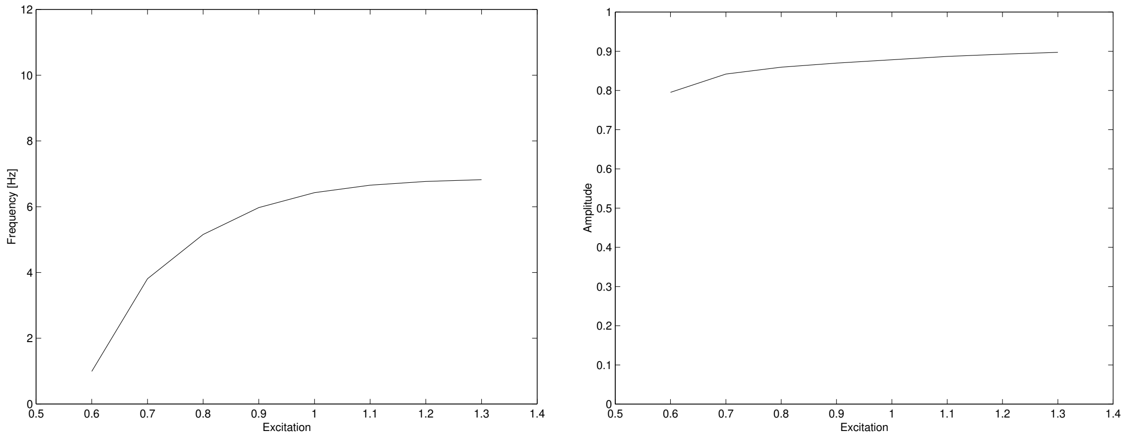


Figure 13: **Run2:** (a) Frequency vs excitation (b) Amplitude vs excitation

	EINl	CCINl	LINl	EINr	CCINr	LINr	BS
EINl	-1.4	-4.0	-	-4.5	-	-	2.4
CCINl	-1.4	-1.0	-	-4.1	-4.0	-	3.7
LINl	-	-	-	-	-	-	-
MNl	-	-	-	-4.9	-3.5	-	7.6
EINr	-4.5	-	-	-1.4	-4.0	-	2.4
CCINr	-4.1	-4.0	-	-1.4	-1.0	-	3.7
LINr	-	-	-	-	-	-	-
MNl	-4.9	-3.5	-	-	-	-	7.6

Table 14: **Evolved artificial segmental oscillator**, best of run3

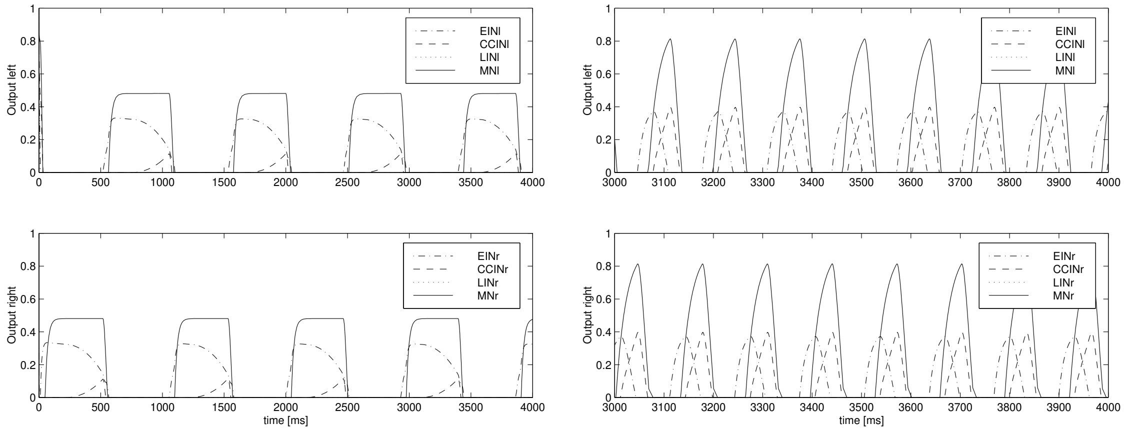


Figure 14: **Run3:** (a) Lowest frequency, (b) Highest frequency

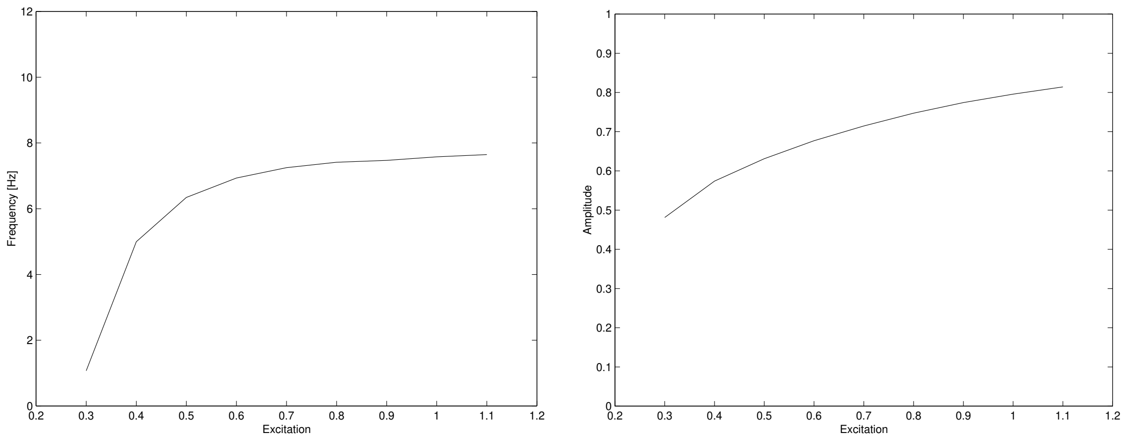


Figure 15: **Run3:** (a) Frequency vs excitation (b) Amplitude vs excitation

	EINl	CCINl	LINl	EINr	CCINr	LINr	BS
EINl	-	-0.1	-0.2	9.6	-2.2	-	-
CCINl	4.4	-1.1	-5.0	-	-2.9	-	-
LINl	11.5	-1.3	-2.2	-	-3.2	-3.4	7.2
MNl	-	-0.2	-0.1	-	-5.0	-1.3	6.4
EINr	9.6	-2.2	-	-	-0.1	-0.2	-
CCINr	-	-2.9	-	4.4	-1.1	-5.0	-
LINr	-	-3.2	-3.4	11.5	-1.3	-2.2	7.2
MNl	-	-5.0	-1.3	-	-0.2	-0.1	6.4

Table 15: **Evolved artificial segmental oscillator**, best of run4

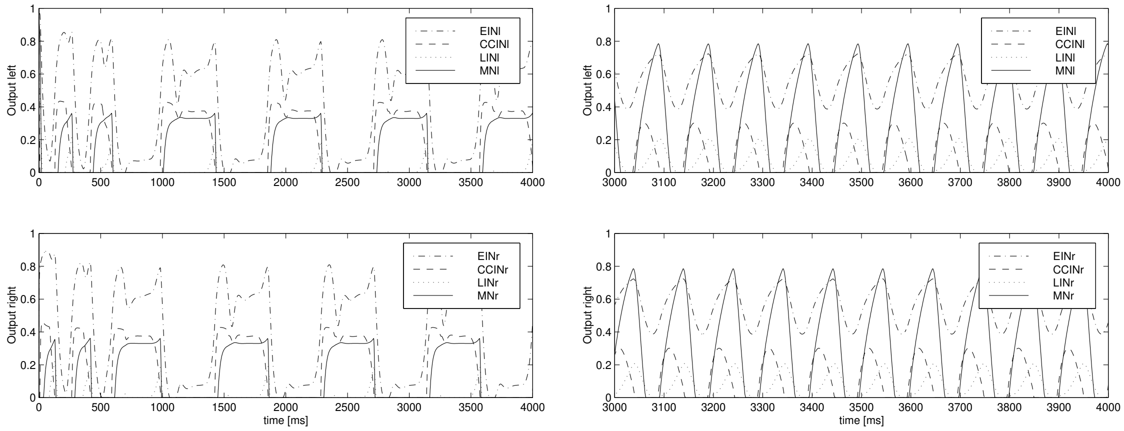


Figure 16: **Run4:** (a) Lowest frequency, (b) Highest frequency

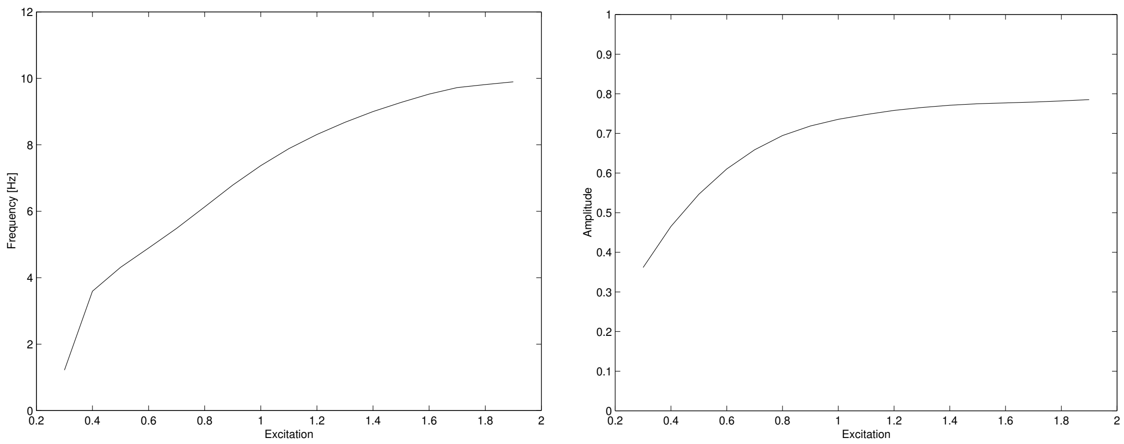


Figure 17: **Run4:** (a) Frequency vs excitation (b) Amplitude vs excitation

	EINl	CCINl	LINl	EINr	CCINr	LINr	BS
EINl	-0.7	-2.2	-	-	-	-3.0	-
CCINl	-	-1.8	-	-	-4.4	-3.9	7.2
LINl	-	-3.5	-1.7	-0.4	-2.1	-2.1	15.0
MNl	-4.9	-	-	-0.3	-4.9	-	9.1
EINr	-	-	-3.0	-0.7	-2.2	-	-
CCINr	-	-4.4	-3.9	-	-1.8	-	7.2
LINr	-0.4	-2.1	-2.1	-	-3.5	-1.7	15.0
MNl	-0.3	-4.9	-	-4.9	-	-	9.1

Table 16: **Evolved artificial segmental oscillator**, best of run5

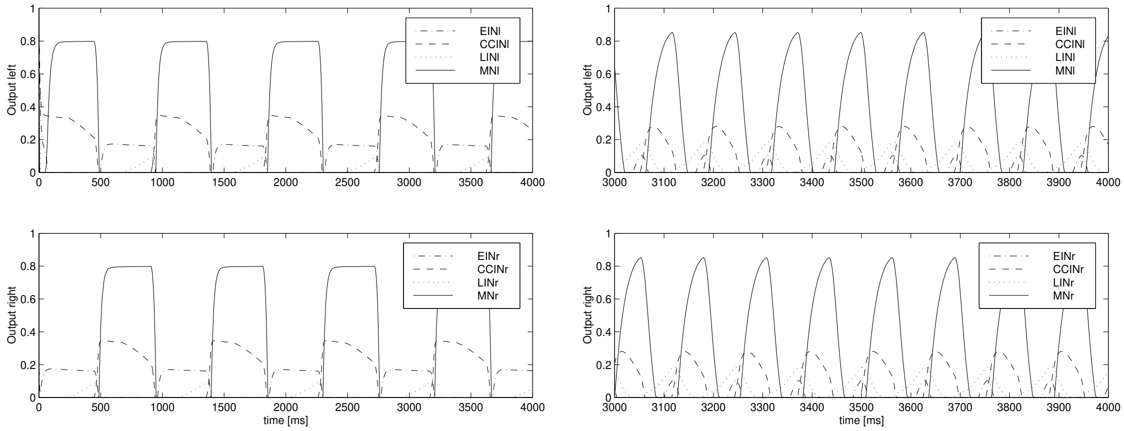


Figure 18: **Run5:** (a) Lowest frequency, (b) Highest frequency

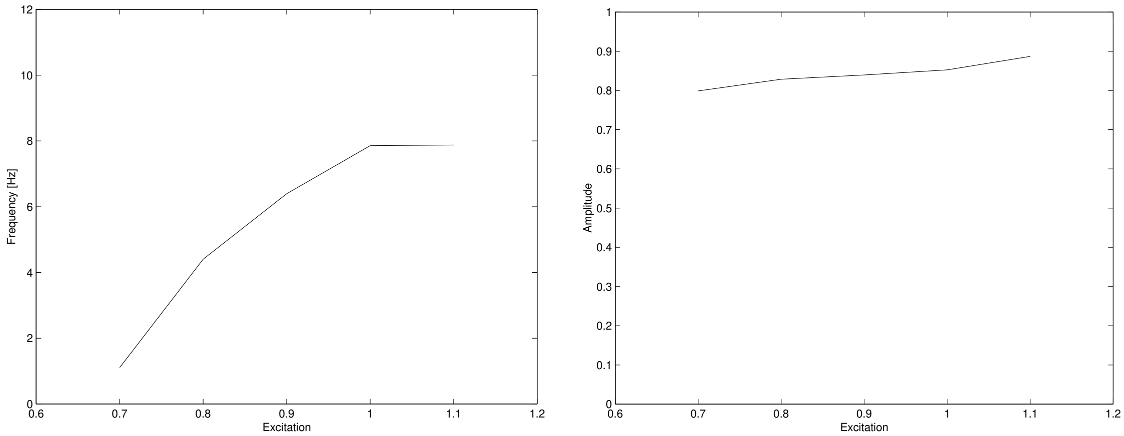


Figure 19: **Run5:** (a) Frequency vs excitation (b) Amplitude vs excitation

	EINl	CCINl	LINl	EINr	CCINr	LINr	BS
EINl	-1.9	-5.0	-	-4.2	-0.1	-	12.1
CCINl	-2.8	-1.7	-	-5.0	-2.1	-	4.7
LINl	-	-	-	-	-	-	-
MNl	-5.0	-	-	-	-	-	4.8
EINr	-4.2	-0.1	-	-1.9	-5.0	-	12.1
CCINr	-5.0	-2.1	-	-2.8	-1.7	-	4.7
LINr	-	-	-	-	-	-	-
MNl	-	-	-	-5.0	-	-	4.8

Table 17: **Evolved artificial segmental oscillator**, best of run6

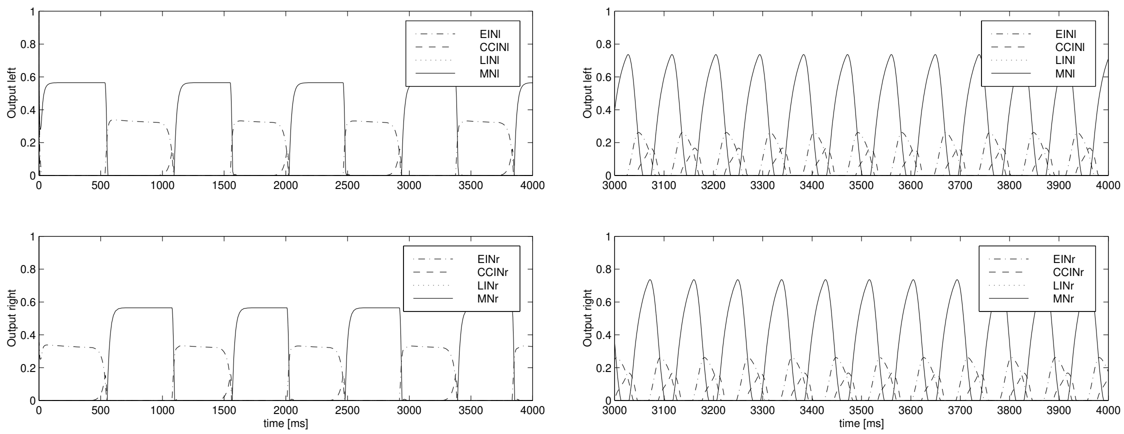


Figure 20: **Run6:** (a) Lowest frequency, (b) Highest frequency

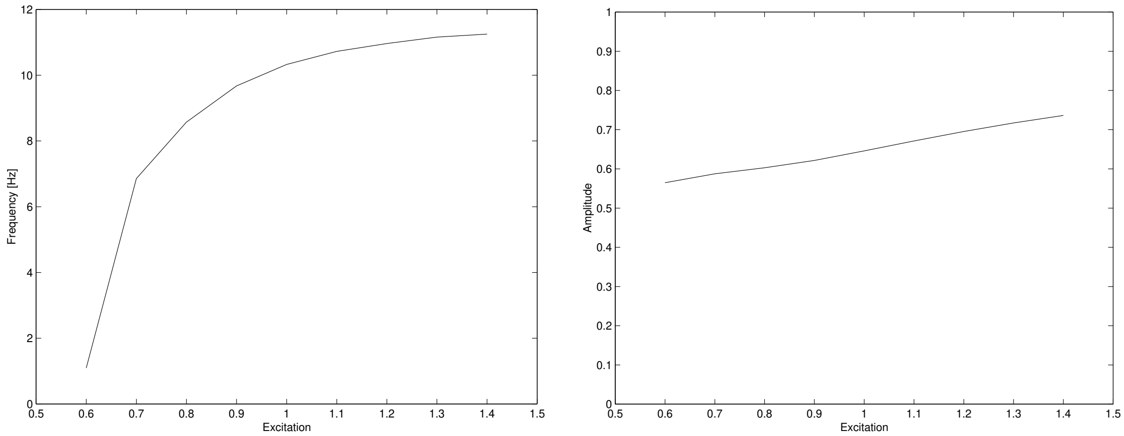


Figure 21: **Run6:** (a) Frequency vs excitation (b) Amplitude vs excitation

	EINl	CCINl	LINl	EINr	CCINr	LINr	BS
EINl	-0.8	-3.8	-	-0.9	-0.7	-	0.8
CCINl	-	-	-	-3.5	-3.7	-	13.6
LINl	-	-	-	-	-	-	-
MNl	-0.4	-3.2	-	-	-	-	3.8
EINr	-0.9	-0.7	-	-0.8	-3.8	-	0.8
CCINr	-3.5	-3.7	-	-	-	-	13.6
LINr	-	-	-	-	-	-	-
MNl	-	-	-	-0.4	-3.2	-	3.8

Table 18: **Evolved artificial segmental oscillator**, best of run7

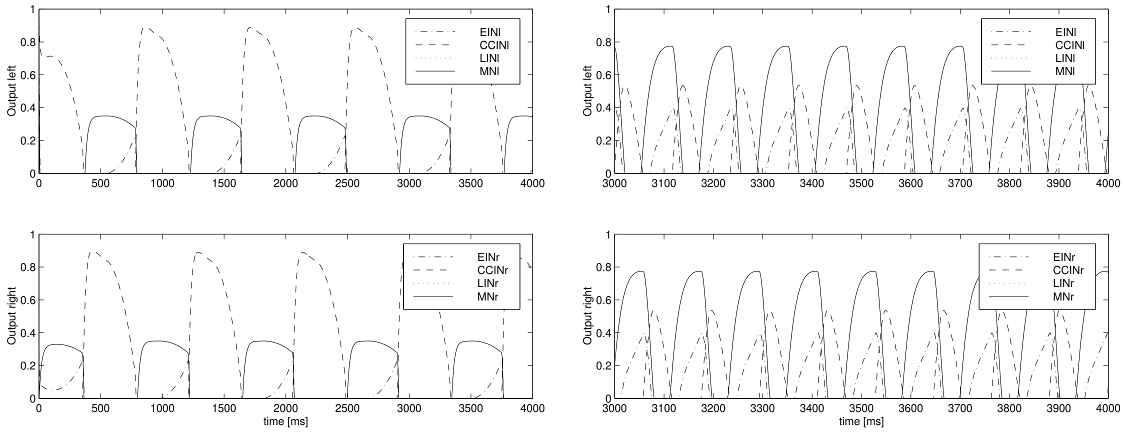


Figure 22: **Run7:** (a) Lowest frequency, (b) Highest frequency

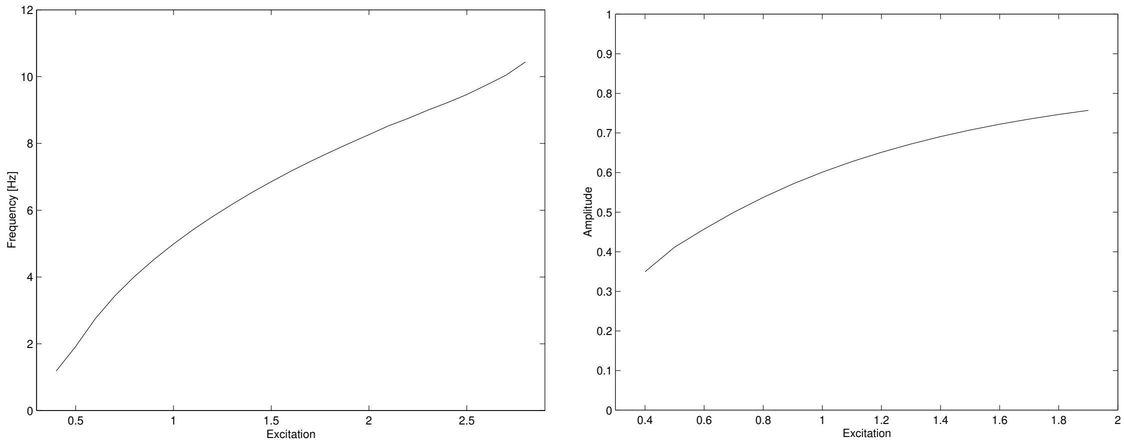


Figure 23: **Run7:**(a) Frequency vs excitation (b) Amplitude vs excitation

	EINl	CCINl	LINl	EINr	CCINr	LINr	BS
EINl	-0.3	-3.6	-	-3.0	-	-	4.5
CCINl	-0.7	-0.2	-	-2.7	-3.8	-	2.7
LINl	-	-	-	-	-	-	-
MNl	-	-	-	-4.2	-	-	8.6
EINr	-3.0	-	-	-0.3	-3.6	-	4.5
CCINr	-2.7	-3.8	-	-0.7	-0.2	-	2.7
LINr	-	-	-	-	-	-	-
MNl	-4.2	-	-	-	-	-	8.6

Table 19: **Evolved artificial segmental oscillator**, best of run8

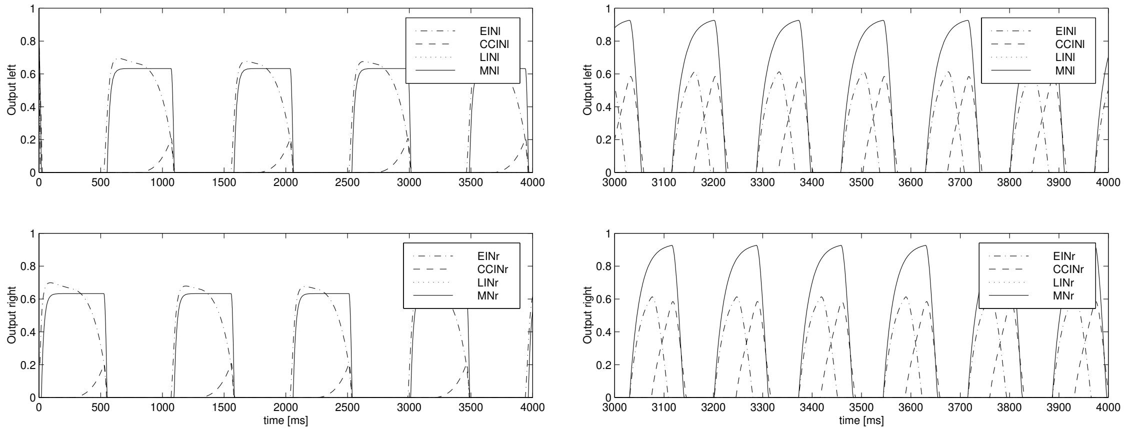


Figure 24: **Run8:** (a) Lowest frequency, (b) Highest frequency

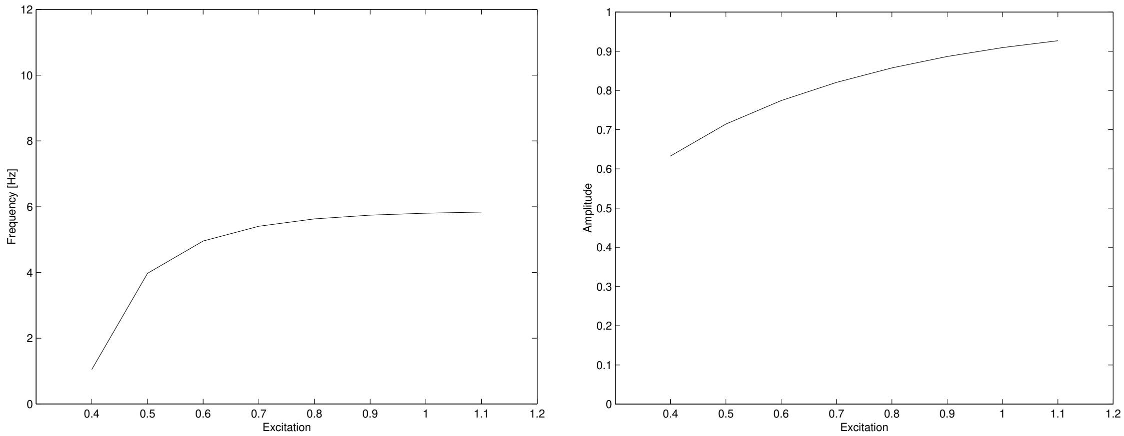


Figure 25: **Run8:** (a) Frequency vs excitation (b) Amplitude vs excitation

	EINl	CCINl	LINl	EINr	CCINr	LINr	BS
EINl	-	-	-5.0	-2.7	9.5	-	-
CCINl	-	-	-3.6	-0.5	11.4	-2.3	5.5
LINl	-0.4	-	-2.9	-2.5	10.1	-2.2	11.0
MNl	-	-	-	-4.3	-	-	4.0
EINr	-2.7	9.5	-	-	-	-5.0	-
CCINr	-0.5	11.4	-2.3	-	-	-3.6	5.5
LINr	-2.5	10.1	-2.2	-0.4	-	-2.9	11.0
MNl	-4.3	-	-	-	-	-	4.0

Table 20: **Evolved artificial segmental oscillator**, best of run9

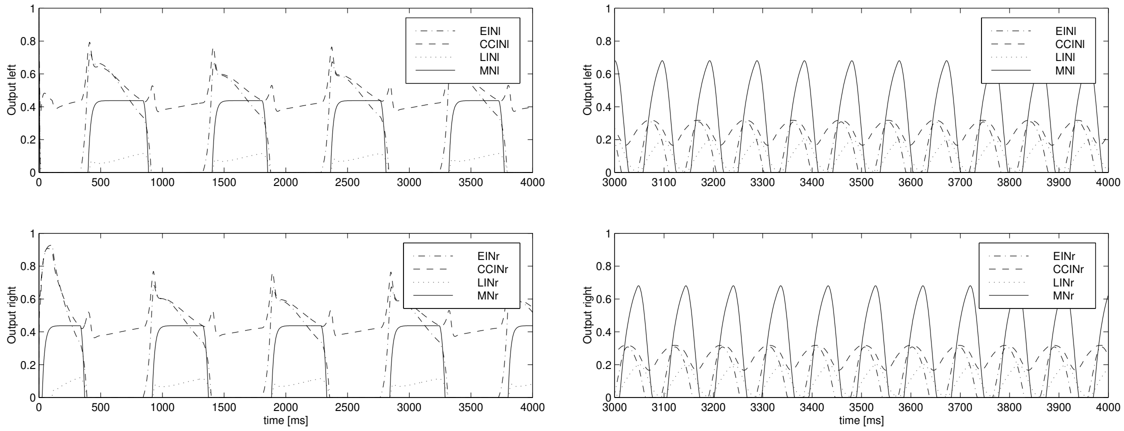


Figure 26: **Run9:** (a) Lowest frequency, (b) Highest frequency

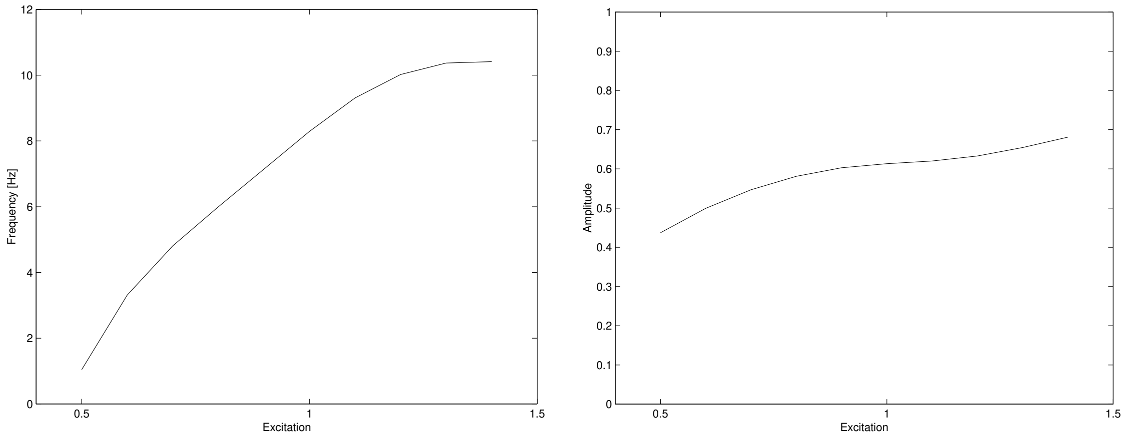


Figure 27: **Run9:** (a) Frequency vs excitation (b) Amplitude vs excitation

	EINl	CCINl	LINl	EINr	CCINr	LINr	BS
EINl	-	-0.2	-	-3.2	-4.6	-	6.7
CCINl	-3.7	-2.3	-	-0.9	-2.6	-	2.4
LINl	-	-	-	-	-	-	-
MNl	-4.5	-	-	-	-	-	3.7
EINr	-3.2	-4.6	-	-	-0.2	-	6.7
CCINr	-0.9	-2.6	-	-3.7	-2.3	-	2.4
LINr	-	-	-	-	-	-	-
MNl	-	-	-	-4.5	-	-	3.7

Table 21: **Evolved artificial segmental oscillator**, best of run10

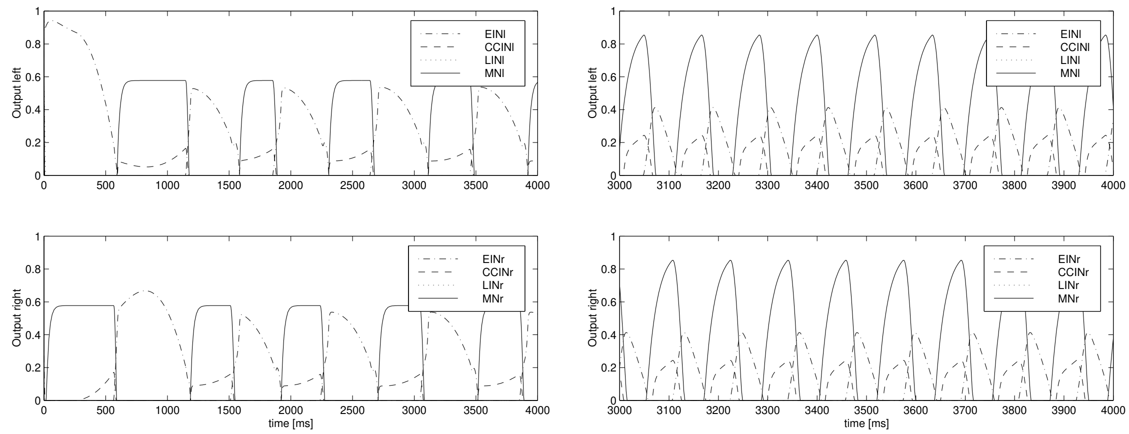


Figure 28: **Run10:** (a) Lowest frequency, (b) Highest frequency

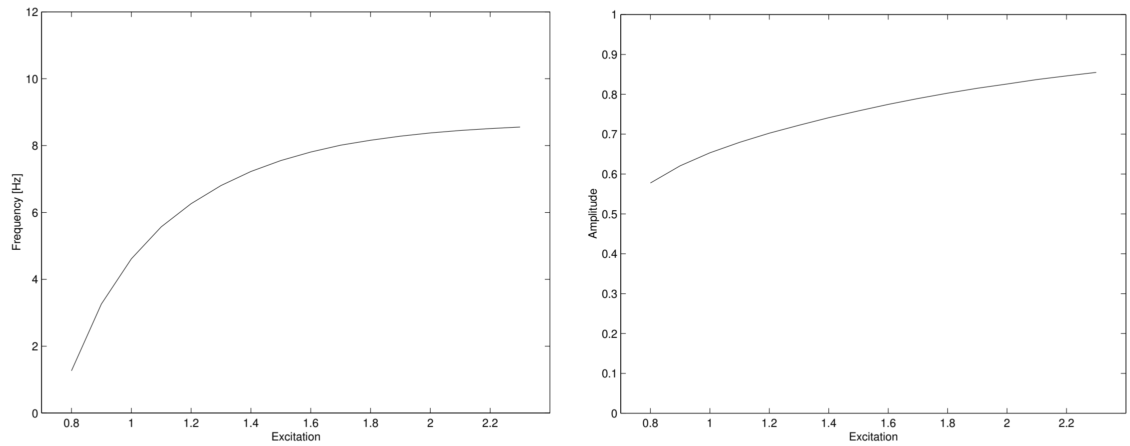


Figure 29: **Run10:** (a) Frequency vs excitation (b) Amplitude vs excitation

C Artificial solutions: evolved multi-segmental controllers

	EINl	CCINl	LINl	EINr	CCINr	LINr	BS
EINl	-	8.6 [2, 5]	-3.8 [0, 2]	-4.3 [2, 12]	-	-0.1 [9, 4]	-0.1
CCINl	-	7.8 [4, 5]	-2.9 [3, 12]	-0.5 [9, 1]	-	-	2.9
LINl	-	10.3 [8, 11]	-	-4.9 [6, 8]	6.8 [2, 7]	-1.5 [2, 9]	-
MNl	-0.7 [4, 2]	-	-	-4.6 [11, 5]	-	-0.8 [3, 12]	10.2
EINr	-4.3 [2, 12]	-	-0.1 [9, 4]	-	8.6 [2, 5]	-3.8 [0, 2]	-0.1
CCINr	-0.5 [9, 1]	-	-	-	7.8 [4, 5]	-2.9 [3, 12]	2.9
LINr	-4.9 [6, 8]	6.8 [2, 7]	-1.5 [2, 9]	-	10.3 [8, 11]	-	-
MNr	-4.6 [11, 5]	-	-0.8 [3, 12]	-0.7 [4, 2]	-	-	10.2

Table 22: **Evolved artificial multi-segmental controller**, best of run1

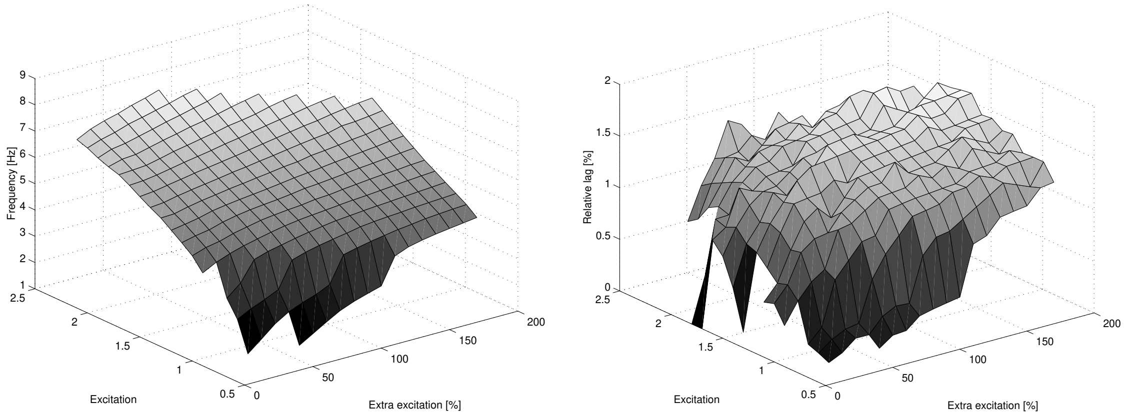


Figure 30: **Run1:** (a) Frequency, (b) Relative lag

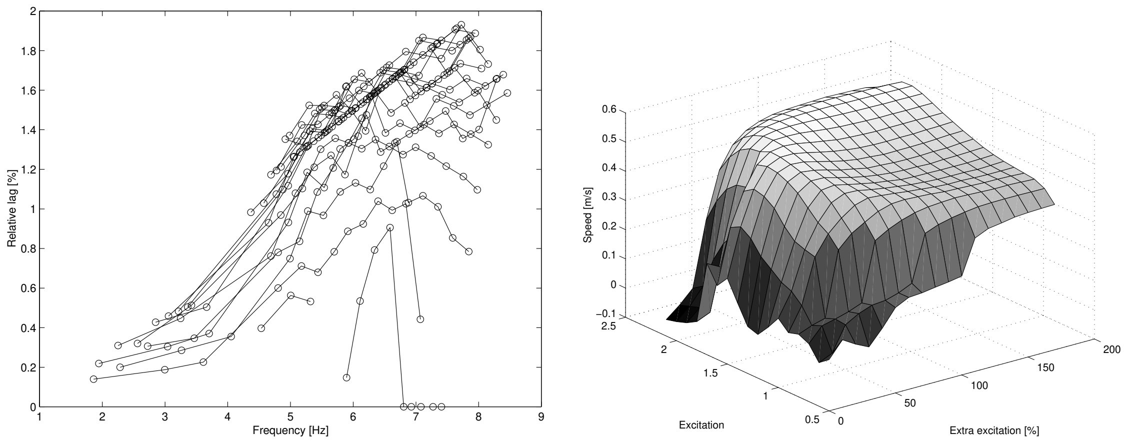


Figure 31: **Run1:** (a) Frequency-lag measurement (b) Speed

	EINl	CCINl	LINl	EINr	CCINr	LINr	BS
EINl	-	-	-	-2.2 [4, 6]	-	-4.6 [3, 6]	4.3
CCINl	-	-	-	-	-	-	-
LINl	-3.3 [3, 1]	-	-0.5 [6, 12]	-0.4 [10, 10]	-	-1.7 [6, 3]	15.0
MNl	-0.2 [9, 2]	-	-0.3 [12, 6]	-4.4 [8, 4]	-	-	11.3
EINr	-2.2 [4, 6]	-	-4.6 [3, 6]	-	-	-	4.3
CCINr	-	-	-	-	-	-	-
LINr	-0.4 [10, 10]	-	-1.7 [6, 3]	-3.3 [3, 1]	-	-0.5 [6, 12]	15.0
MNr	-4.4 [8, 4]	-	-	-0.2 [9, 2]	-	-0.3 [12, 6]	11.3

Table 23: **Evolved artificial multi-segmental controller**, best of run2

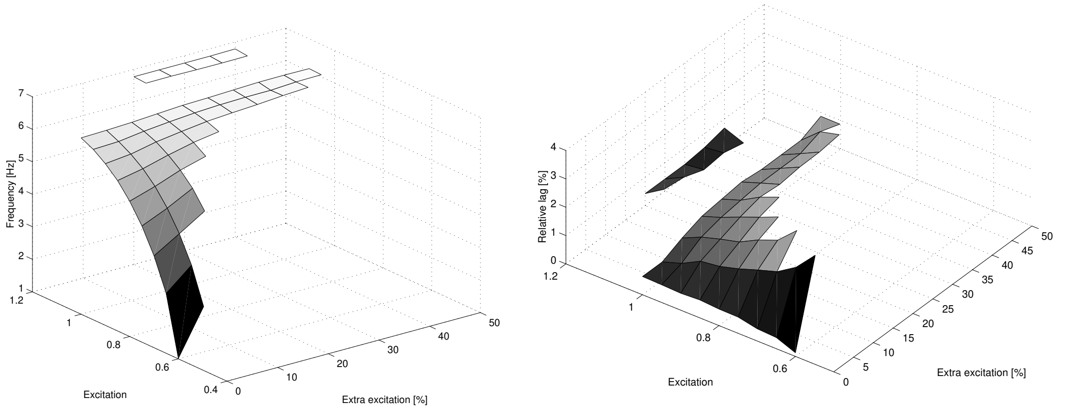


Figure 32: **Run2:** (a) Frequency, (b) Relative lag

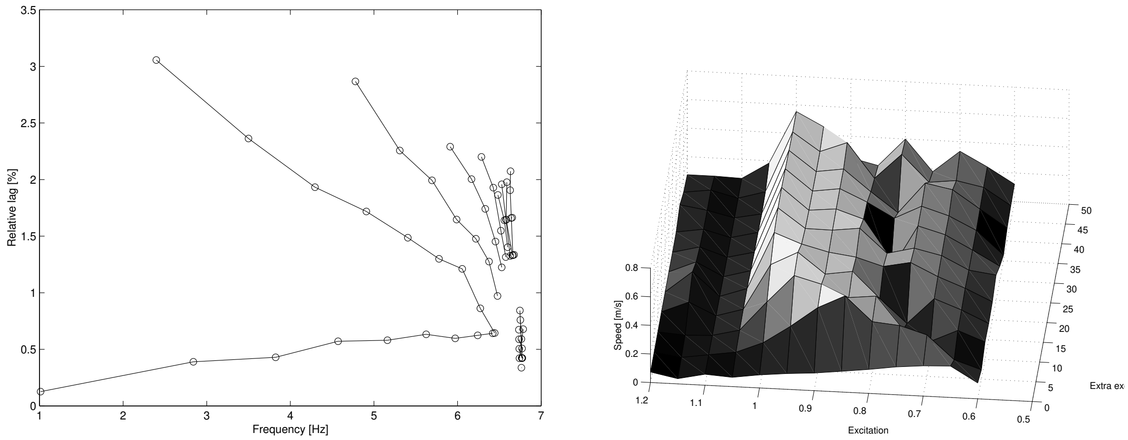


Figure 33: **Run2:** (a) Frequency-lag measurement (b) Speed

	EINl	CCINl	LINl	EINr	CCINr	LINr	BS
EINl	-1.4 [10, 0]	-4.0 [2, 4]	-	-4.5 [4, 5]	-	-	2.4
CCINl	-1.4 [2, 12]	-1.0 [4, 10]	-	-4.1 [5, 6]	-4.0 [4, 1]	-	3.7
LINl	-	-	-	-	-	-	-
MNl	-0.0 [5, 9]	-	-	-4.9 [7, 6]	-3.5 [8, 9]	-	7.6
EINr	-4.5 [4, 5]	-	-	-1.4 [10, 0]	-4.0 [2, 4]	-	2.4
CCINr	-4.1 [5, 6]	-4.0 [4, 1]	-	-1.4 [2, 12]	-1.0 [4, 10]	-	3.7
LINr	-	-	-	-	-	-	-
MNr	-4.9 [7, 6]	-3.5 [8, 9]	-	-0.0 [5, 9]	-	-	7.6

Table 24: **Evolved artificial multi-segmental controller**, best of run3

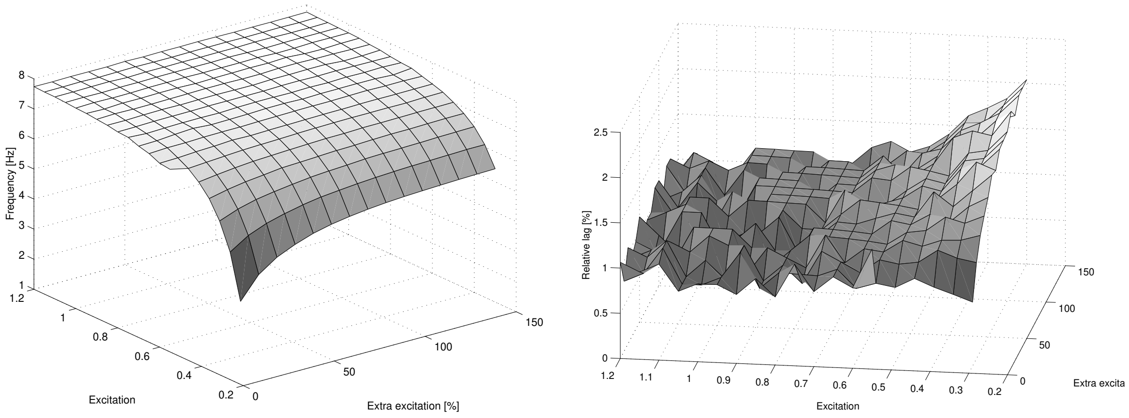


Figure 34: **Run3:** (a) Frequency, (b) Relative lag

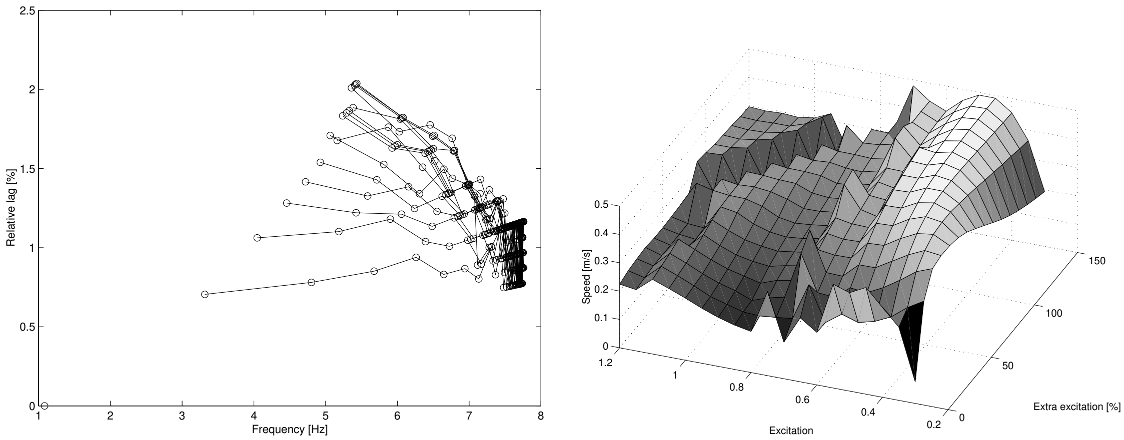


Figure 35: **Run3:** (a) Frequency-lag measurement (b) Speed

	EINl	CCINl	LINl	EINr	CCINr	LINr	BS
EINl	-	-0.1 [11, 8]	-0.2 [1, 11]	9.6 [9, 10]	-2.2 [4, 8]	-	-
CCINl	4.4 [3, 10]	-1.1 [10, 0]	-5.0 [8, 8]	-	-2.9 [2, 3]	-	-
LINl	11.5 [2, 5]	-1.3 [9, 12]	-2.2 [4, 4]	-	-3.2 [3, 1]	-3.4 [6, 8]	7.2
MNl	-	-0.2 [10, 3]	-0.1 [4, 9]	-	-5.0 [3, 6]	-1.3 [6, 3]	6.4
EINr	9.6 [9, 10]	-2.2 [4, 8]	-	-	-0.1 [11, 8]	-0.2 [1, 11]	-
CCINr	-	-2.9 [2, 3]	-	4.4 [3, 10]	-1.1 [10, 0]	-5.0 [8, 8]	-
LINr	-	-3.2 [3, 1]	-3.4 [6, 8]	11.5 [2, 5]	-1.3 [9, 12]	-2.2 [4, 4]	7.2
MNr	-	-5.0 [3, 6]	-1.3 [6, 3]	-	-0.2 [10, 3]	-0.1 [4, 9]	6.4

Table 25: **Evolved artificial multi-segmental controller**, best of run4

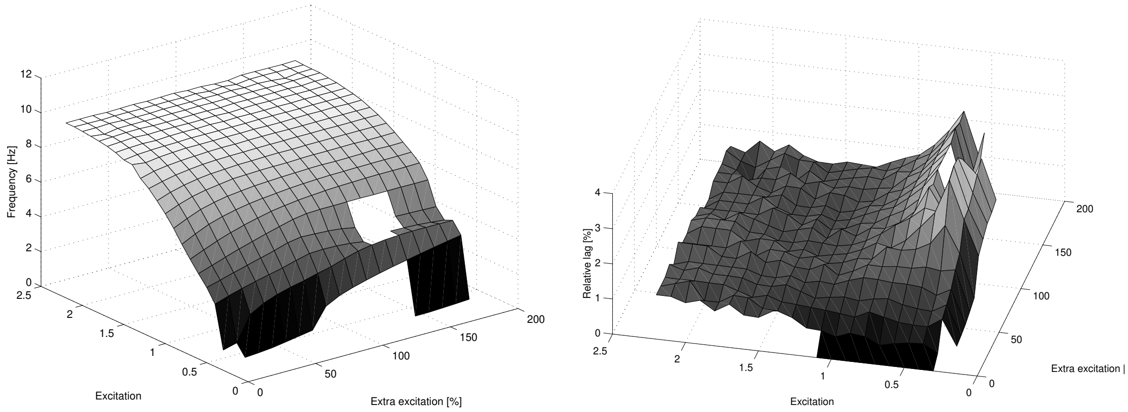


Figure 36: **Run4:** (a) Frequency, (b) Relative lag

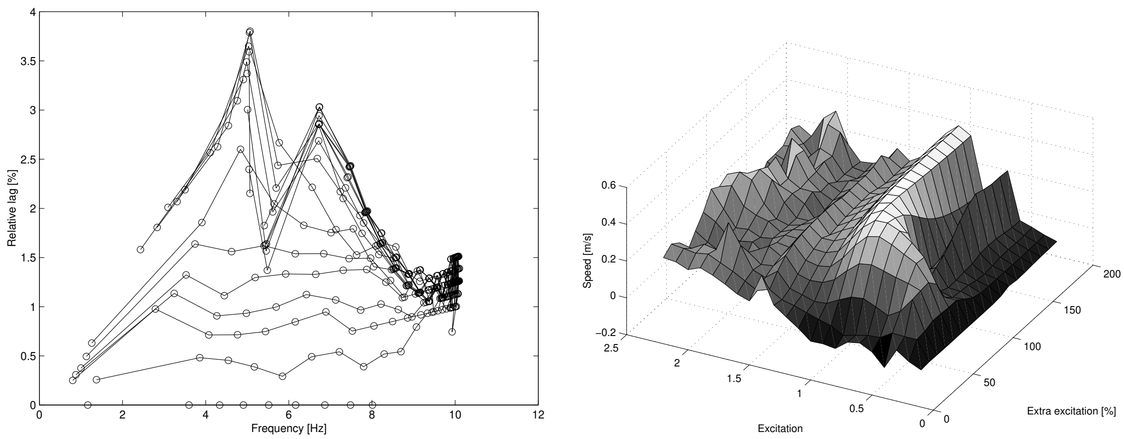


Figure 37: **Run4:** (a) Frequency-lag measurement (b) Speed

	EINl	CCINl	LINl	EINr	CCINr	LINr	BS
EINl	-0.7 [5, 6]	-2.2 [9, 8]	-	-	-	-3.0 [2, 6]	-
CCINl	-	-1.8 [9, 0]	-	-	-4.4 [6, 6]	-3.9 [4, 12]	7.2
LINl	-	-3.5 [7, 1]	-1.7 [6, 2]	-0.4 [5, 9]	-2.1 [3, 3]	-2.1 [7, 11]	15.0
MNl	-4.9 [12, 2]	-	-	-0.3 [7, 10]	-4.9 [0, 3]	-	9.1
EINr	-	-	-3.0 [2, 6]	-0.7 [5, 6]	-2.2 [9, 8]	-	-
CCINr	-	-4.4 [6, 6]	-3.9 [4, 12]	-	-1.8 [9, 0]	-	7.2
LINr	-0.4 [5, 9]	-2.1 [3, 3]	-2.1 [7, 11]	-	-3.5 [7, 1]	-1.7 [6, 2]	15.0
MNr	-0.3 [7, 10]	-4.9 [0, 3]	-	-4.9 [12, 2]	-	-	9.1

Table 26: **Evolved artificial multi-segmental controller**, best of run5

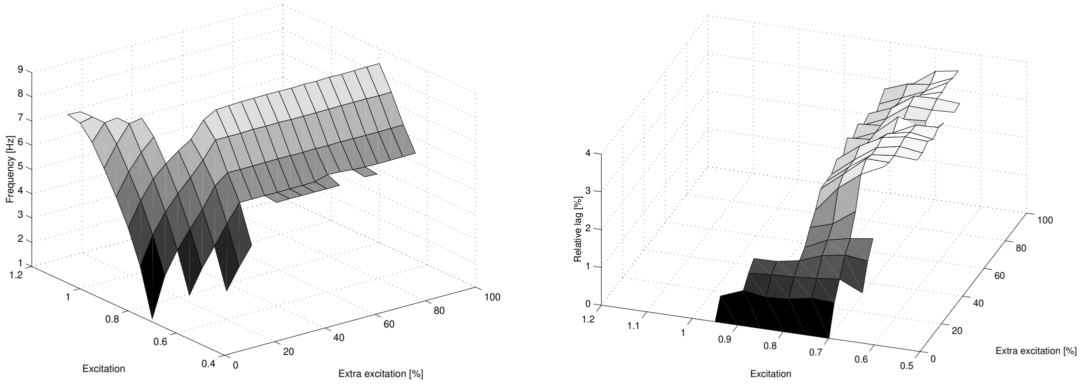


Figure 38: **Run5:** (a) Frequency, (b) Relative lag

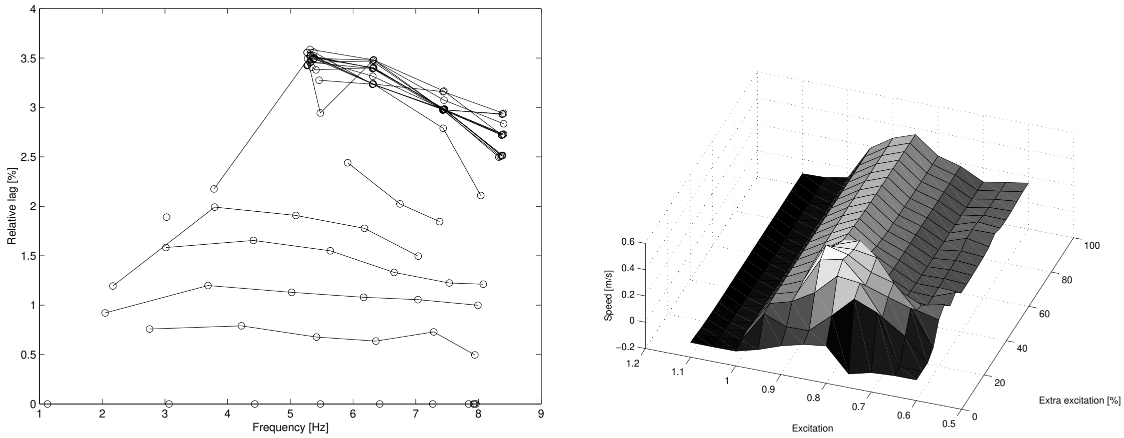


Figure 39: **Run5:** (a) Frequency-lag measurement (b) Speed

	EINl	CCINl	LINl	EINr	CCINr	LINr	BS
EINl	-1.9 [11, 4]	-5.0 [6, 8]	-	-4.2 [4, 8]	-0.1 [5, 8]	-	12.1
CCINl	-2.8 [5, 5]	-1.7 [7, 0]	-	-5.0 [8, 2]	-2.1 [3, 2]	-	4.7
LINl	-	-	-	-	-	-	-
MNl	-5.0 [4, 0]	-	-	-	-	-	4.8
EINr	-4.2 [4, 8]	-0.1 [5, 8]	-	-1.9 [11, 4]	-5.0 [6, 8]	-	12.1
CCINr	-5.0 [8, 2]	-2.1 [3, 2]	-	-2.8 [5, 5]	-1.7 [7, 0]	-	4.7
LINr	-	-	-	-	-	-	-
MNr	-	-	-	-5.0 [4, 0]	-	-	4.8

Table 27: **Evolved artificial multi-segmental controller**, best of run6

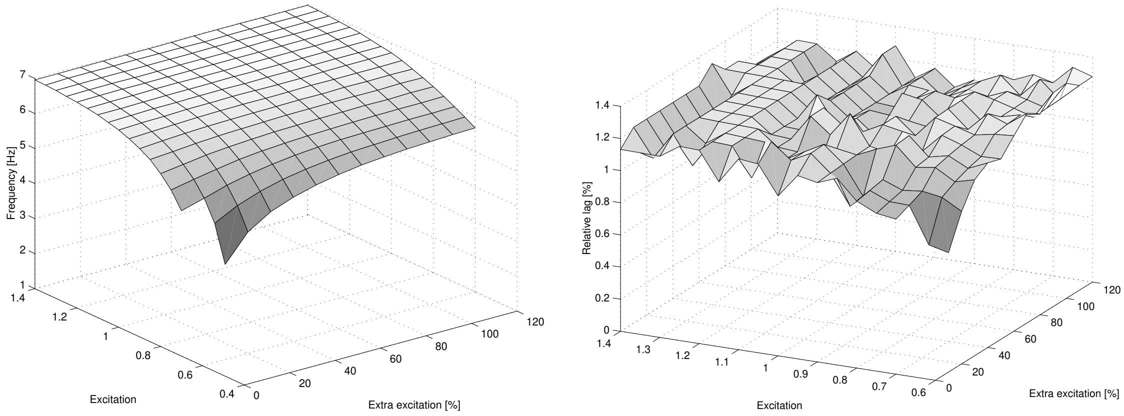


Figure 40: **Run6:** (a) Frequency, (b) Relative lag

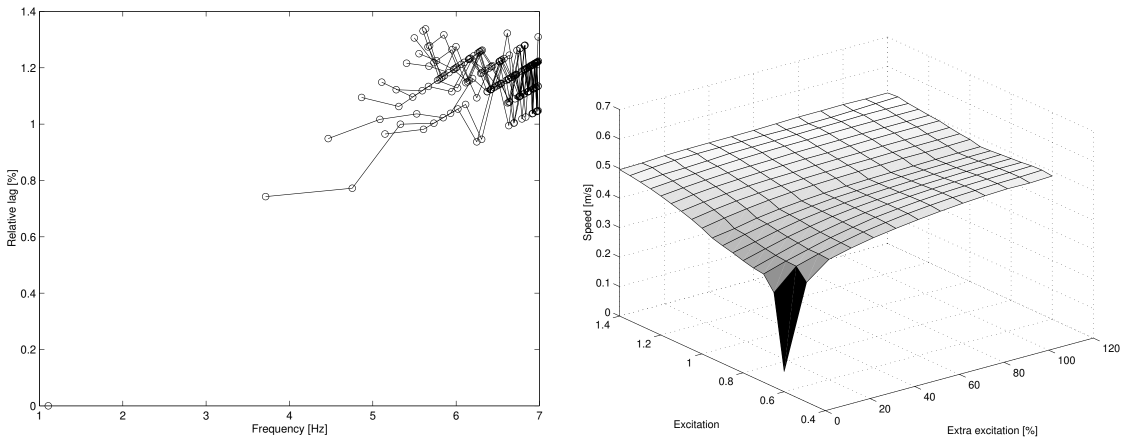


Figure 41: **Run6:** (a) Frequency-lag measurement (b) Speed

	EINl	CCINl	LINl	EINr	CCINr	LINr	BS
EINl	-0.8 [12, 4]	-3.8 [12, 10]	-	-0.9 [5, 10]	-0.7 [1, 10]	-	0.8
CCINl	-0.0 [12, 8]	-	-	-3.5 [2, 2]	-3.7 [9, 9]	-	13.6
LINl	-	-	-	-	-	-	-
MNl	-0.4 [9, 2]	-3.2 [8, 1]	-	-	-	-	3.8
EINr	-0.9 [5, 10]	-0.7 [1, 10]	-	-0.8 [12, 4]	-3.8 [12, 10]	-	0.8
CCINr	-3.5 [2, 2]	-3.7 [9, 9]	-	-0.0 [12, 8]	-	-	13.6
LINr	-	-	-	-	-	-	-
MNr	-	-	-	-0.4 [9, 2]	-3.2 [8, 1]	-	3.8

Table 28: **Evolved artificial multi-segmental controller**, best of run7

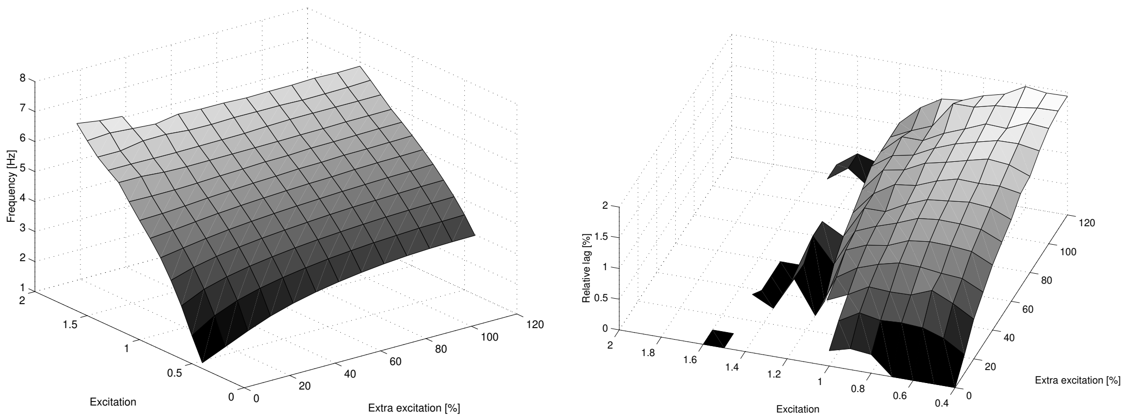


Figure 42: **Run7:** (a) Frequency, (b) Relative lag

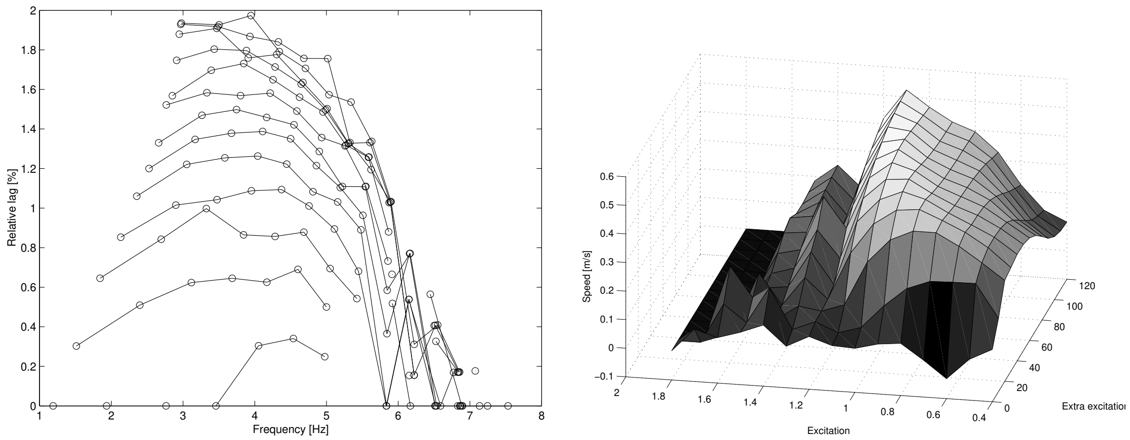


Figure 43: **Run7:** (a) Frequency-lag measurement (b) Speed

	EINl	CCINl	LINl	EINr	CCINr	LINr	BS
EINl	-0.3 [11, 1]	-3.6 [0, 8]	-	-3.0 [0, 5]	-	-	4.5
CCINl	-0.7 [0, 5]	-0.2 [3, 11]	-	-2.7 [11, 1]	-3.8 [6, 2]	-	2.7
LINl	-	-	-	-	-	-	-
MNl	-	-	-	-4.2 [12, 4]	-	-	8.6
EINr	-3.0 [0, 5]	-	-	-0.3 [11, 1]	-3.6 [0, 8]	-	4.5
CCINr	-2.7 [11, 1]	-3.8 [6, 2]	-	-0.7 [0, 5]	-0.2 [3, 11]	-	2.7
LINr	-	-	-	-	-	-	-
MNr	-4.2 [12, 4]	-	-	-	-	-	8.6

Table 29: **Evolved artificial multi-segmental controller**, best of run8

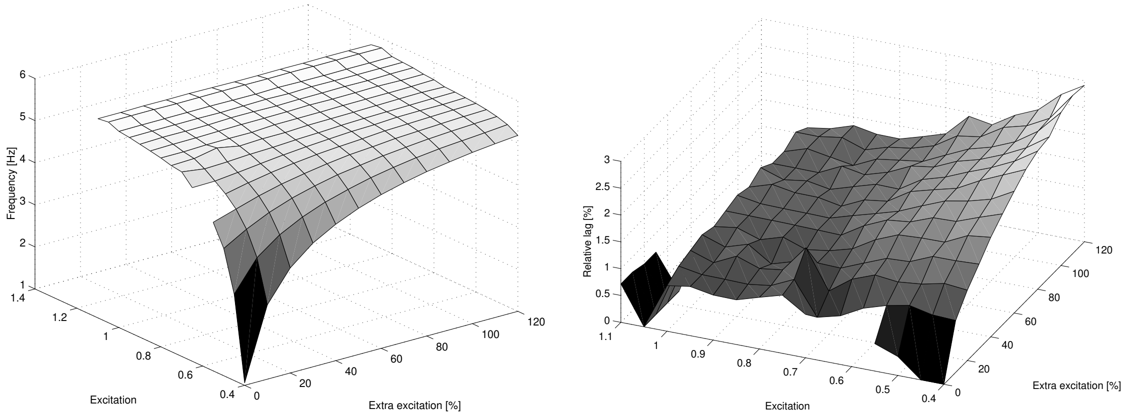


Figure 44: **Run8:** (a) Frequency, (b) Relative lag

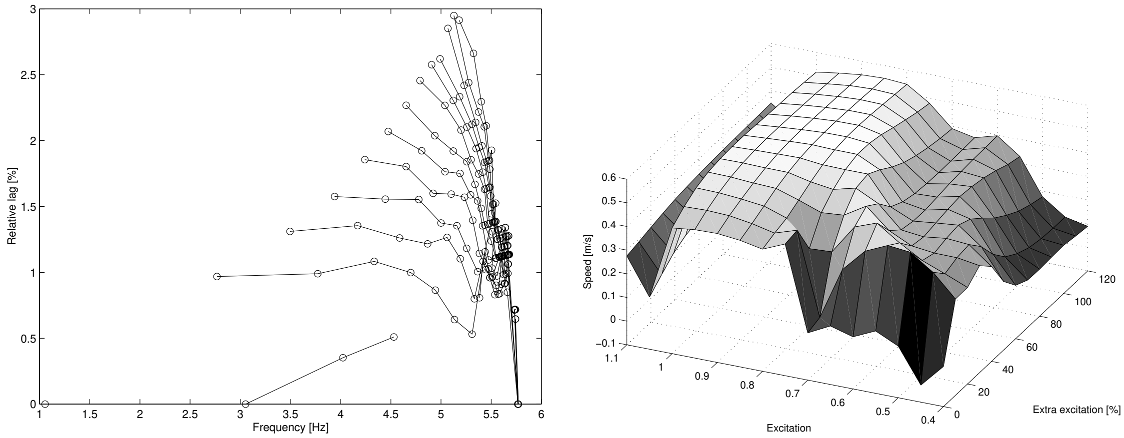


Figure 45: **Run8:** (a) Frequency-lag measurement (b) Speed

	EINl	CCINl	LINl	EINr	CCINr	LINr	BS
EINl	-	-	-5.0 [5, 2]	-2.7 [1, 4]	9.5 [10, 9]	-	-
CCINl	-	-	-3.6 [5, 6]	-0.5 [2, 1]	11.4 [2, 12]	-2.3 [4, 1]	5.5
LINl	-0.4 [8, 3]	-	-2.9 [11, 11]	-2.5 [5, 7]	10.1 [11, 12]	-2.2 [2, 8]	11.0
MNl	-	-	-	-4.3 [1, 5]	-	-	4.0
EINr	-2.7 [1, 4]	9.5 [10, 9]	-	-	-	-5.0 [5, 2]	-
CCINr	-0.5 [2, 1]	11.4 [2, 12]	-2.3 [4, 1]	-	-	-3.6 [5, 6]	5.5
LINr	-2.5 [5, 7]	10.1 [11, 12]	-2.2 [2, 8]	-0.4 [8, 3]	-	-2.9 [11, 11]	11.0
MNr	-4.3 [1, 5]	-	-	-	-	-	4.0

Table 30: **Evolved artificial multi-segmental controller**, best of run9

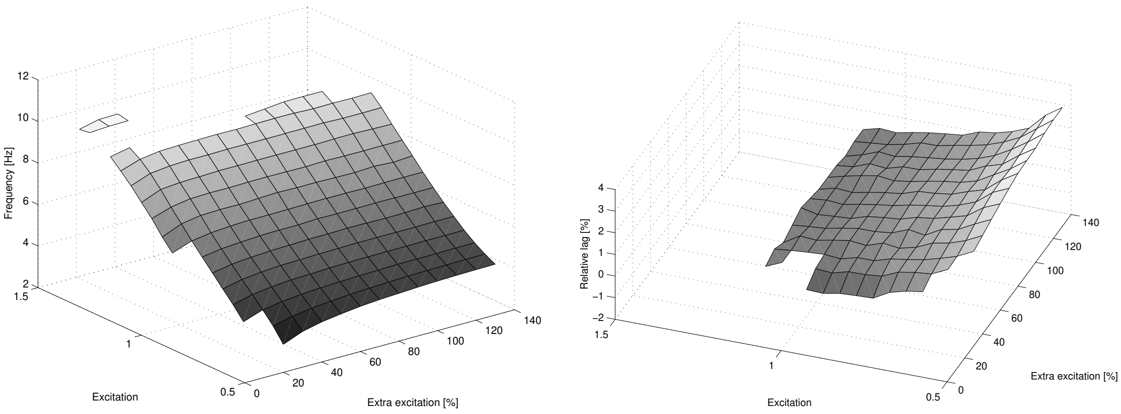


Figure 46: **Run9:** (a) Frequency, (b) Relative lag

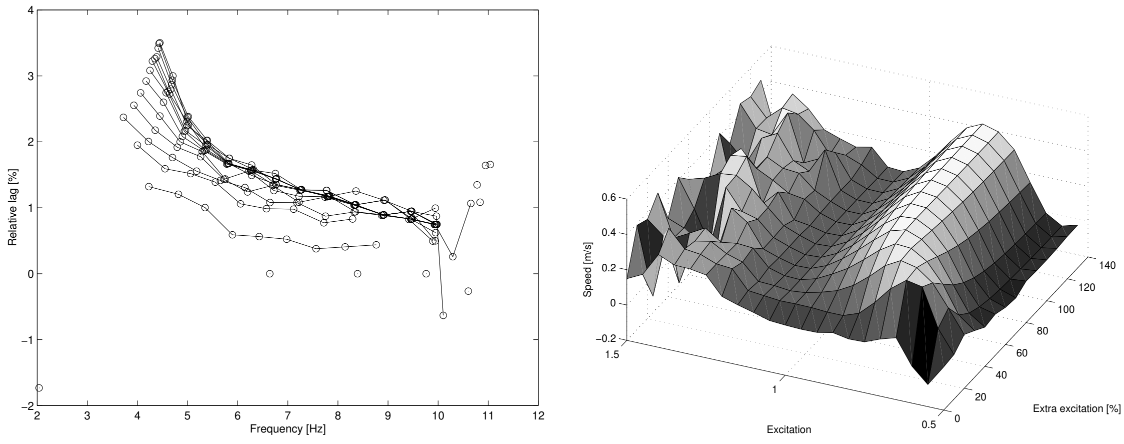


Figure 47: **Run9:** (a) Frequency-lag measurement (b) Speed

	EINl	CCINl	LINl	EINr	CCINr	LINr	BS
EINl	-	-0.2 [1, 4]	-	-3.2 [6, 5]	-4.6 [0, 9]	-	6.7
CCINl	-3.7 [9, 1]	-2.3 [6, 11]	-	-0.9 [7, 0]	-2.6 [2, 7]	-	2.4
LINl	-	-	-	-	-	-	-
MNl	-4.5 [4, 3]	-	-	-	-	-	3.7
EINr	-3.2 [6, 5]	-4.6 [0, 9]	-	-	-0.2 [1, 4]	-	6.7
CCINr	-0.9 [7, 0]	-2.6 [2, 7]	-	-3.7 [9, 1]	-2.3 [6, 11]	-	2.4
LINr	-	-	-	-	-	-	-
MNr	-	-	-	-4.5 [4, 3]	-	-	3.7

Table 31: **Evolved artificial multi-segmental controller**, best of run10

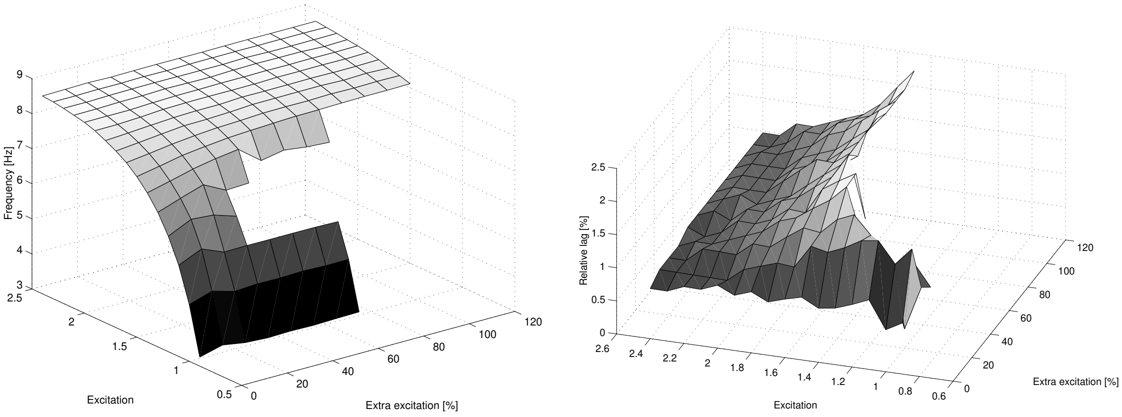


Figure 48: **Run10:** (a) Frequency, (b) Relative lag

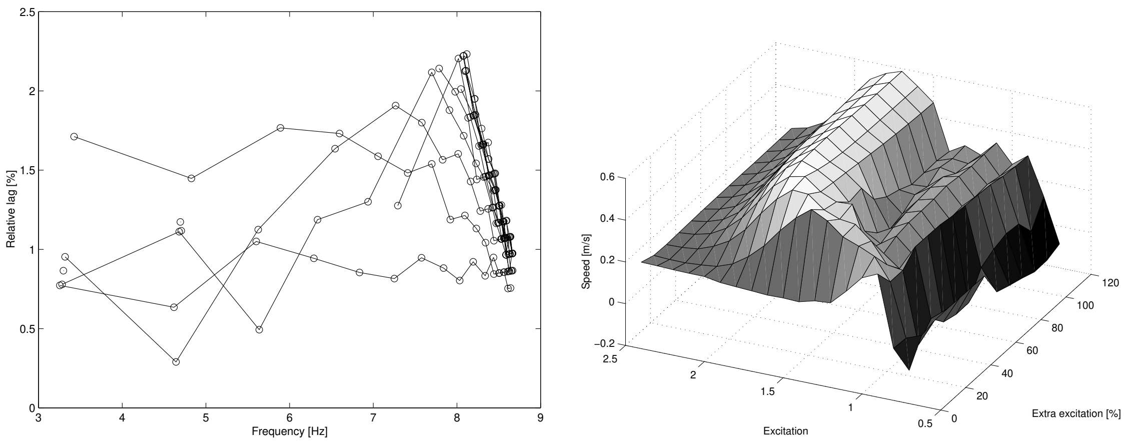


Figure 49: **Run10:** (a) Frequency-lag measurement (b) Speed

D Artificial solutions: evolved sensory feedback

	EINl	CCINl	LINl	EINr	CCINr	LINr	ECl	ECr	BS
EINl	-0.8	-3.8	-	-0.9	-0.7	-	0.0	0.8	0.8
CCINl	-	-	-	-3.5	-3.7	-	1.6	3.0	13.6
LINl	-	-	-	-	-	-	-	-	-
MNl	-0.4	-3.2	-	-	-	-	4.2	-0.3	3.8
EINr	-0.9	-0.7	-	-0.8	-3.8	-	0.8	0.0	0.8
CCINr	-3.5	-3.7	-	-	-	-	3.0	1.6	13.6
LINr	-	-	-	-	-	-	-	-	-
MNr	-	-	-	-0.4	-3.2	-	-0.3	4.2	3.8

Table 32: **Evolved sensory feedback for artificial controller No 7**, best of run1

	EINl	CCINl	LINl	EINr	CCINr	LINr	ECl	ECr	BS
EINl	-0.8	-3.8	-	-0.9	-0.7	-	0.1	3.4	0.8
CCINl	-	-	-	-3.5	-3.7	-	2.2	2.3	13.6
LINl	-	-	-	-	-	-	-	-	-
MNl	-0.4	-3.2	-	-	-	-	4.2	1.9	3.8
EINr	-0.9	-0.7	-	-0.8	-3.8	-	3.4	0.1	0.8
CCINr	-3.5	-3.7	-	-	-	-	2.3	2.2	13.6
LINr	-	-	-	-	-	-	-	-	-
MNr	-	-	-	-0.4	-3.2	-	1.9	4.2	3.8

Table 33: **Evolved sensory feedback for artificial controller No 7**, best of run2

	EINl	CCINl	LINl	EINr	CCINr	LINr	ECl	ECr	BS
EINl	-0.8	-3.8	-	-0.9	-0.7	-	0.5	4.1	0.8
CCINl	-	-	-	-3.5	-3.7	-	2.0	4.1	13.6
LINl	-	-	-	-	-	-	-	-	-
MNl	-0.4	-3.2	-	-	-	-	2.8	2.9	3.8
EINr	-0.9	-0.7	-	-0.8	-3.8	-	4.1	0.5	0.8
CCINr	-3.5	-3.7	-	-	-	-	4.1	2.0	13.6
LINr	-	-	-	-	-	-	-	-	-
MNr	-	-	-	-0.4	-3.2	-	2.9	2.8	3.8

Table 34: **Evolved sensory feedback for artificial controller No 7**, best of run3

	EINl	CCINl	LINl	EINr	CCINr	LINr	ECl	ECr	BS
EINl	-0.8	-3.8	-	-0.9	-0.7	-	0.1	0.7	0.8
CCINl	-	-	-	-3.5	-3.7	-	4.2	0.5	13.6
LINl	-	-	-	-	-	-	-	-	-
MNl	-0.4	-3.2	-	-	-	-	1.4	2.4	3.8
EINr	-0.9	-0.7	-	-0.8	-3.8	-	0.7	0.1	0.8
CCINr	-3.5	-3.7	-	-	-	-	0.5	4.2	13.6
LINr	-	-	-	-	-	-	-	-	-
MNr	-	-	-	-0.4	-3.2	-	2.4	1.4	3.8

Table 35: **Evolved sensory feedback for artificial controller No 7**, best of run4

	EINl	CCINl	LINl	EINr	CCINr	LINr	ECl	ECr	BS
EINl	-0.8	-3.8	-	-0.9	-0.7	-	0.1	5.0	0.8
CCINl	-	-	-	-3.5	-3.7	-	-0.6	0.0	13.6
LINl	-	-	-	-	-	-	-	-	-
MNl	-0.4	-3.2	-	-	-	-	3.3	3.6	3.8
EINr	-0.9	-0.7	-	-0.8	-3.8	-	5.0	0.1	0.8
CCINr	-3.5	-3.7	-	-	-	-	0.0	-0.6	13.6
LINr	-	-	-	-	-	-	-	-	-
MNr	-	-	-	-0.4	-3.2	-	3.6	3.3	3.8

Table 36: **Evolved sensory feedback for artificial controller No 7**, best of run5

	EINl	CCINl	LINl	EINr	CCINr	LINr	ECl	ECr	BS
EINl	-0.8	-3.8	-	-0.9	-0.7	-	0.2	2.7	0.8
CCINl	-	-	-	-3.5	-3.7	-	0.0	1.0	13.6
LINl	-	-	-	-	-	-	-	-	-
MNl	-0.4	-3.2	-	-	-	-	-	-	3.8
EINr	-0.9	-0.7	-	-0.8	-3.8	-	2.7	0.2	0.8
CCINr	-3.5	-3.7	-	-	-	-	1.0	0.0	13.6
LINr	-	-	-	-	-	-	-	-	-
MNr	-	-	-	-0.4	-3.2	-	-	-	3.8

Table 37: **Evolved sensory feedback for artificial controller No 7**, best of run6

	EINl	CCINl	LINl	EINr	CCINr	LINr	ECl	ECr	BS
EINl	-0.8	-3.8	-	-0.9	-0.7	-	0.1	0.4	0.8
CCINl	-	-	-	-3.5	-3.7	-	-0.3	3.2	13.6
LINl	-	-	-	-	-	-	-	-	-
MNl	-0.4	-3.2	-	-	-	-	-	-	3.8
EINr	-0.9	-0.7	-	-0.8	-3.8	-	0.4	0.1	0.8
CCINr	-3.5	-3.7	-	-	-	-	3.2	-0.3	13.6
LINr	-	-	-	-	-	-	-	-	-
MNr	-	-	-	-0.4	-3.2	-	-	-	3.8

Table 38: **Evolved sensory feedback for artificial controller No 7**, best of run 7

	EINl	CCINl	LINl	EINr	CCINr	LINr	ECl	ECr	BS
EINl	-0.8	-3.8	-	-0.9	-0.7	-	0.3	0.3	0.8
CCINl	-	-	-	-3.5	-3.7	-	-1.1	1.9	13.6
LINl	-	-	-	-	-	-	-	-	-
MNl	-0.4	-3.2	-	-	-	-	-	-	3.8
EINr	-0.9	-0.7	-	-0.8	-3.8	-	0.3	0.3	0.8
CCINr	-3.5	-3.7	-	-	-	-	1.9	-1.1	13.6
LINr	-	-	-	-	-	-	-	-	-
MNr	-	-	-	-0.4	-3.2	-	-	-	3.8

Table 39: **Evolved sensory feedback for artificial controller No 7**, best of run8

	EINl	CCINl	LINl	EINr	CCINr	LINr	ECl	ECr	BS
EINl	-0.8	-3.8	-	-0.9	-0.7	-	-0.0	0.0	0.8
CCINl	-	-	-	-3.5	-3.7	-	-0.2	0.1	13.6
LINl	-	-	-	-	-	-	-	-	-
MNl	-0.4	-3.2	-	-	-	-	-	-	3.8
EINr	-0.9	-0.7	-	-0.8	-3.8	-	0.0	-0.0	0.8
CCINr	-3.5	-3.7	-	-	-	-	0.1	-0.2	13.6
LINr	-	-	-	-	-	-	-	-	-
MNr	-	-	-	-0.4	-3.2	-	-	-	3.8

Table 40: **Evolved sensory feedback for artificial controller No 7**, best of run9

	EINl	CCINl	LINl	EINr	CCINr	LINr	ECl	ECr	BS
EINl	-0.8	-3.8	-	-0.9	-0.7	-	0.2	2.7	0.8
CCINl	-	-	-	-3.5	-3.7	-	3.8	1.2	13.6
LINl	-	-	-	-	-	-	-	-	-
MNl	-0.4	-3.2	-	-	-	-	-	-	3.8
EINr	-0.9	-0.7	-	-0.8	-3.8	-	2.7	0.2	0.8
CCINr	-3.5	-3.7	-	-	-	-	1.2	3.8	13.6
LINr	-	-	-	-	-	-	-	-	-
MNr	-	-	-	-0.4	-3.2	-	-	-	3.8

Table 41: **Evolved sensory feedback for artificial controller No 7**, best of run10

E Biological segmental connectivity: evolved segmental oscillators

	EINl	CCINl	LINl	EINr	CCINr	LINr	BS
EINl	7.6	-	-	-	-2.6	-	9.5
CCINl	-	-	-4.4	-	-1.2	-	8.8
LINl	6.3	-	-	-	-2.3	-	4.4
MNl	-	-	-	-	-1.4	-	3.2
EINr	-	-2.6	-	7.6	-	-	9.5
CCINr	-	-1.2	-	-	-	-4.4	8.8
LINr	-	-2.3	-	6.3	-	-	4.4
MNr	-	-1.4	-	-	-	-	3.2

Table 42: **Biological connectivity, segmental oscillator**, best of run1

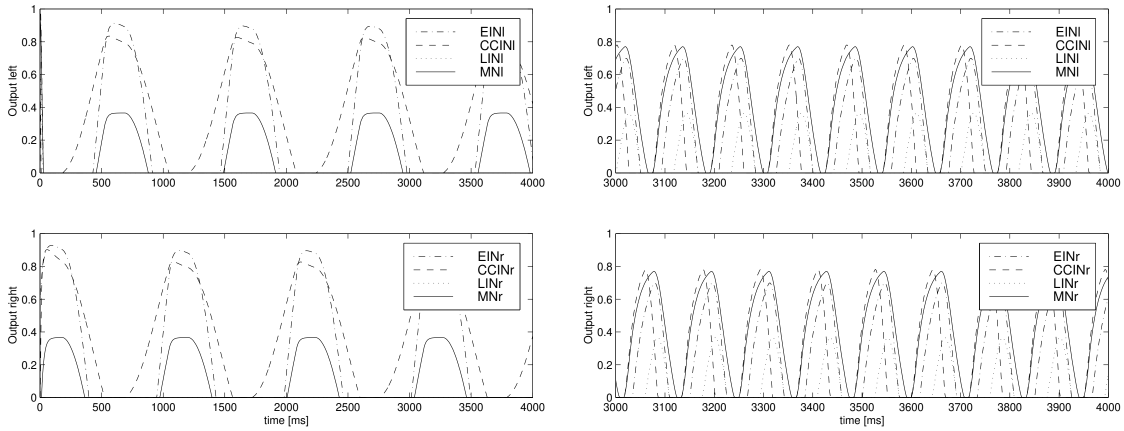


Figure 50: **Run1:** (a) Lowest frequency, (b) Highest frequency

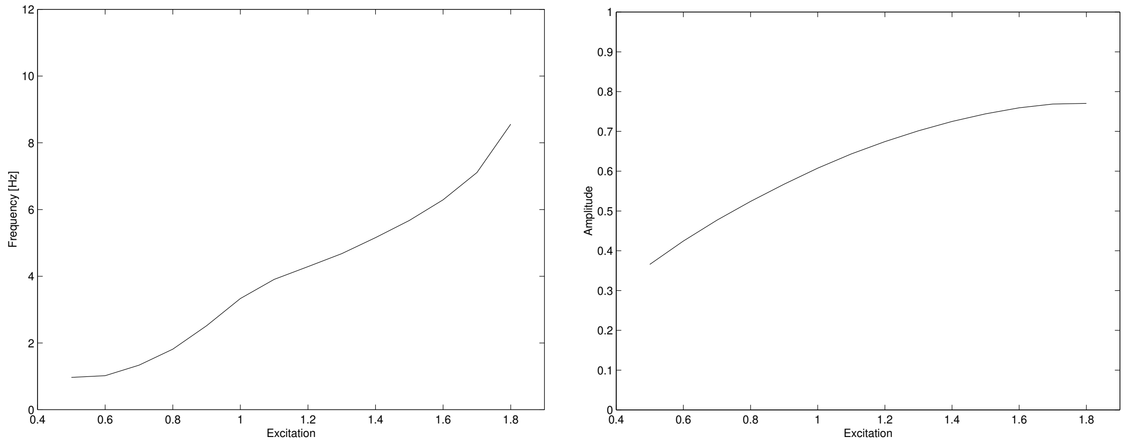


Figure 51: **Run1:** (a) Frequency vs excitation (b) Amplitude vs excitation

	EINl	CCINl	LINl	EINr	CCINr	LINr	BS
EINl	3.9	-	-	-	-3.4	-	10.3
CCINl	11.3	-	-4.8	-	-1.3	-	13.0
LINl	4.9	-	-	-	-4.3	-	5.0
MNl	1.9	-	-	-	-1.9	-	2.0
EINr	-	-3.4	-	3.9	-	-	10.3
CCINr	-	-1.3	-	11.3	-	-4.8	13.0
LINr	-	-4.3	-	4.9	-	-	5.0
MNr	-	-1.9	-	1.9	-	-	2.0

Table 43: **Biological connectivity, segmental oscillator**, best of run2

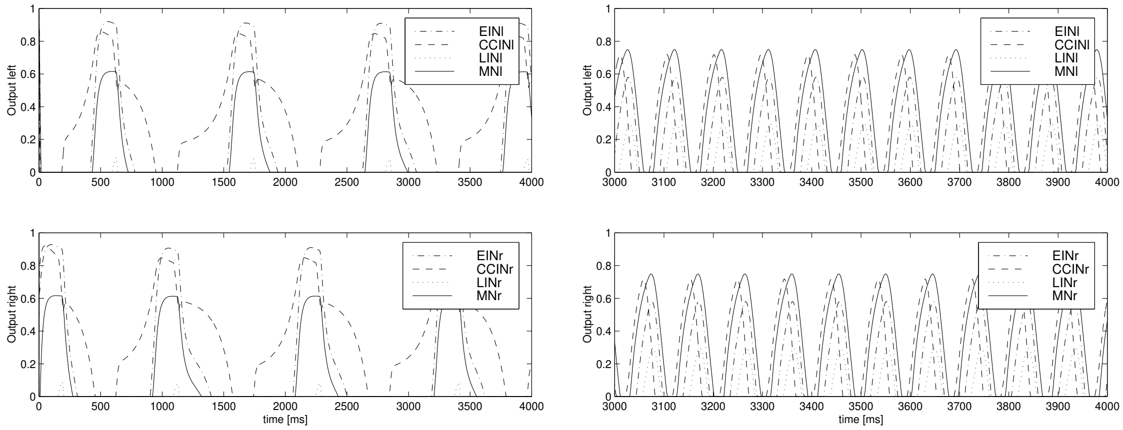


Figure 52: **Run2:** (a) Lowest frequency, (b) Highest frequency

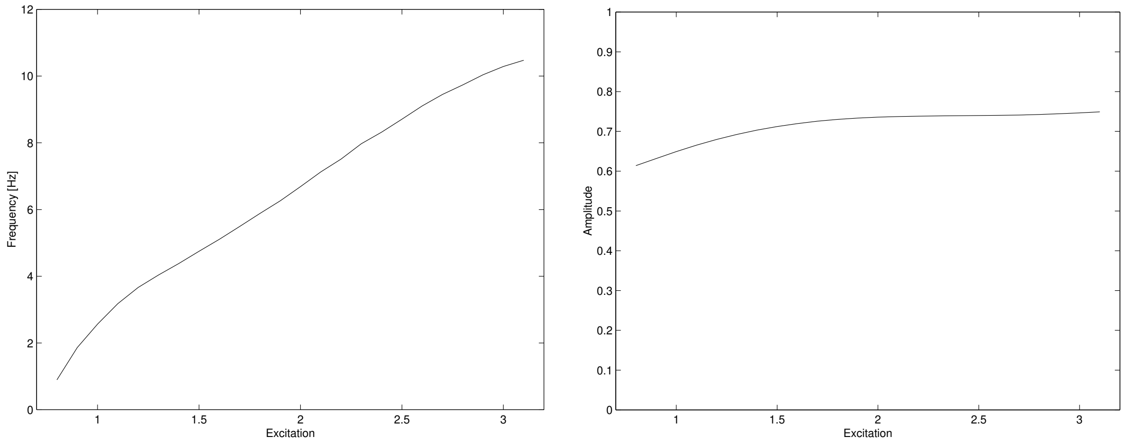


Figure 53: **Run2:** (a) Frequency vs excitation (b) Amplitude vs excitation

	EINl	CCINl	LINl	EINr	CCINr	LINr	BS
EINl	10.1	-	-	-	-3.8	-	3.8
CCINl	12.8	-	-4.0	-	-1.3	-	11.1
LINl	6.0	-	-	-	-4.2	-	4.3
MNl	-	-	-	-	-1.8	-	3.0
EINr	-	-3.8	-	10.1	-	-	3.8
CCINr	-	-1.3	-	12.8	-	-4.0	11.1
LINr	-	-4.2	-	6.0	-	-	4.3
MNr	-	-1.8	-	-	-	-	3.0

Table 44: **Biological connectivity, segmental oscillator**, best of run3

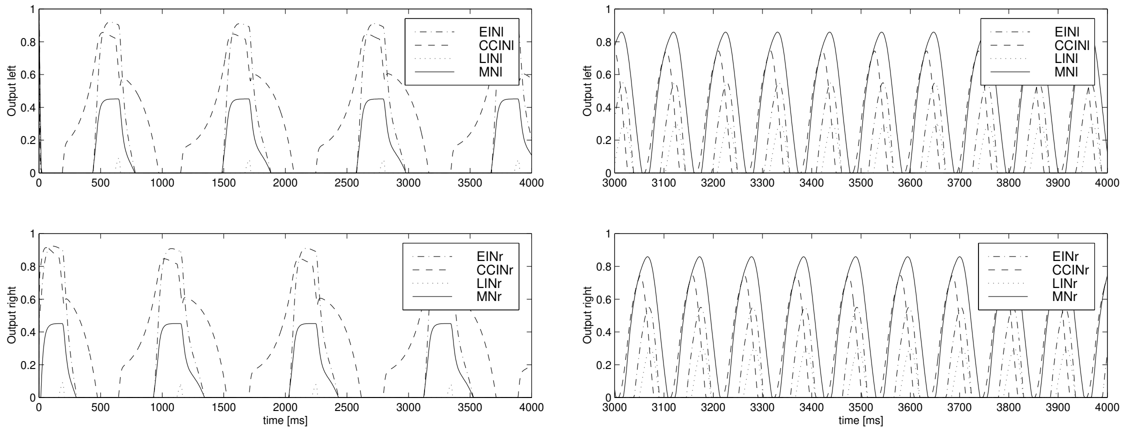


Figure 54: **Run3:** (a) Lowest frequency, (b) Highest frequency

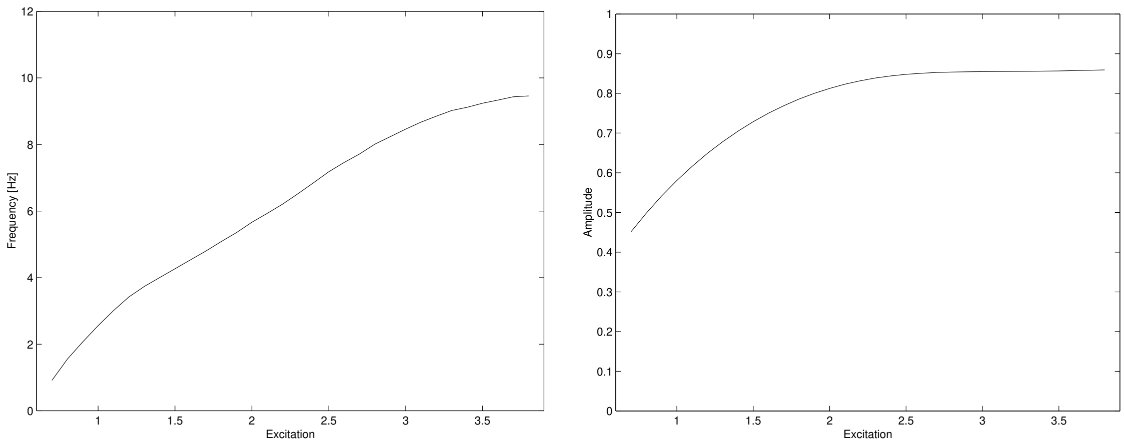


Figure 55: **Run3:** (a) Frequency vs excitation (b) Amplitude vs excitation

	EINl	CCINl	LINl	EINr	CCINr	LINr	BS
EINl	-	-	-	-	-3.6	-	7.4
CCINl	10.4	-	-3.8	-	-1.2	-	13.1
LINl	5.3	-	-	-	-1.4	-	5.4
MNl	-	-	-	-	-1.5	-	4.1
EINr	-	-3.6	-	-	-	-	7.4
CCINr	-	-1.2	-	10.4	-	-3.8	13.1
LINr	-	-1.4	-	5.3	-	-	5.4
MNr	-	-1.5	-	-	-	-	4.1

Table 45: **Biological connectivity, segmental oscillator**, best of run4

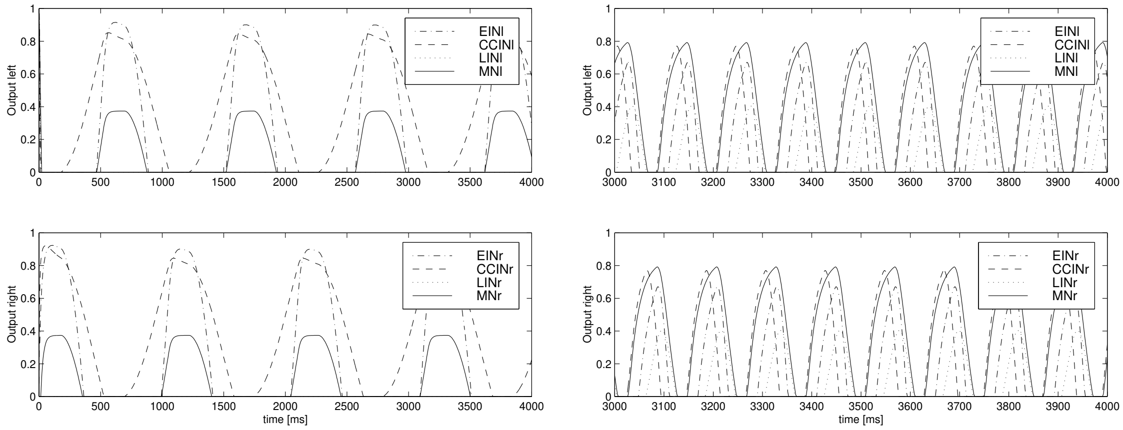


Figure 56: **Run4:** (a) Lowest frequency, (b) Highest frequency

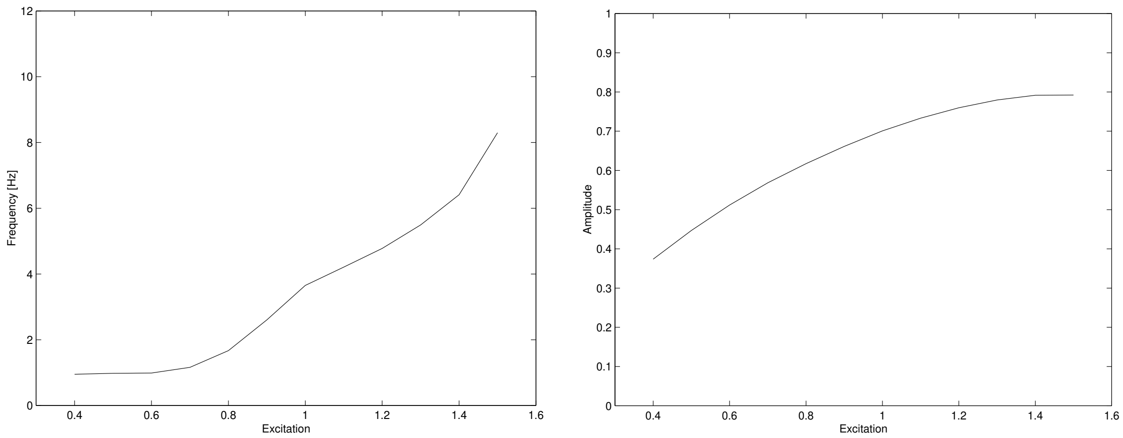


Figure 57: **Run4:** (a) Frequency vs excitation (b) Amplitude vs excitation

	EINl	CCINl	LINl	EINr	CCINr	LINr	BS
EINl	6.8	-	-	-	-0.0	-	7.1
CCINl	3.4	-	-4.5	-	-1.3	-	7.6
LINl	3.7	-	-	-	-4.2	-	6.9
MNl	-	-	-	-	-2.0	-	4.5
EINr	-	-0.0	-	6.8	-	-	7.1
CCINr	-	-1.3	-	3.4	-	-4.5	7.6
LINr	-	-4.2	-	3.7	-	-	6.9
MNr	-	-2.0	-	-	-	-	4.5

Table 46: **Biological connectivity, segmental oscillator**, best of run5

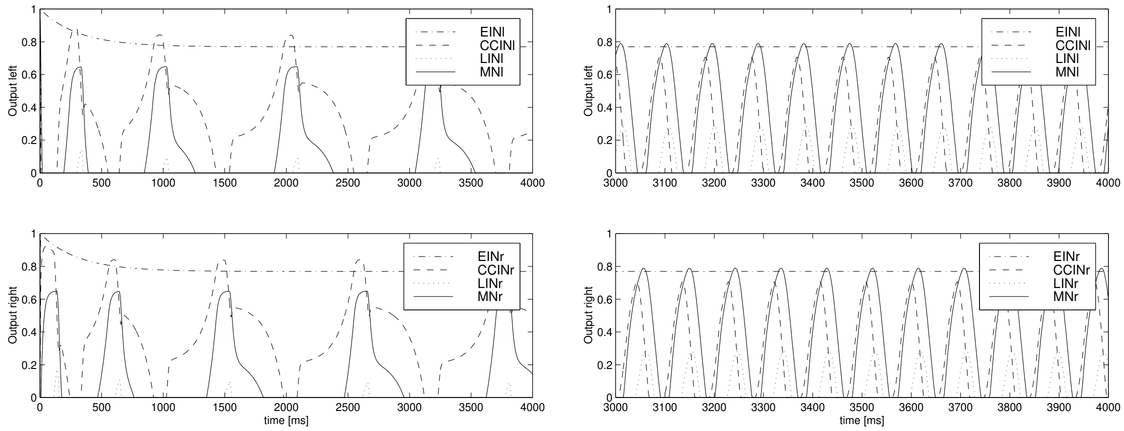


Figure 58: **Run5:** (a) Lowest frequency, (b) Highest frequency

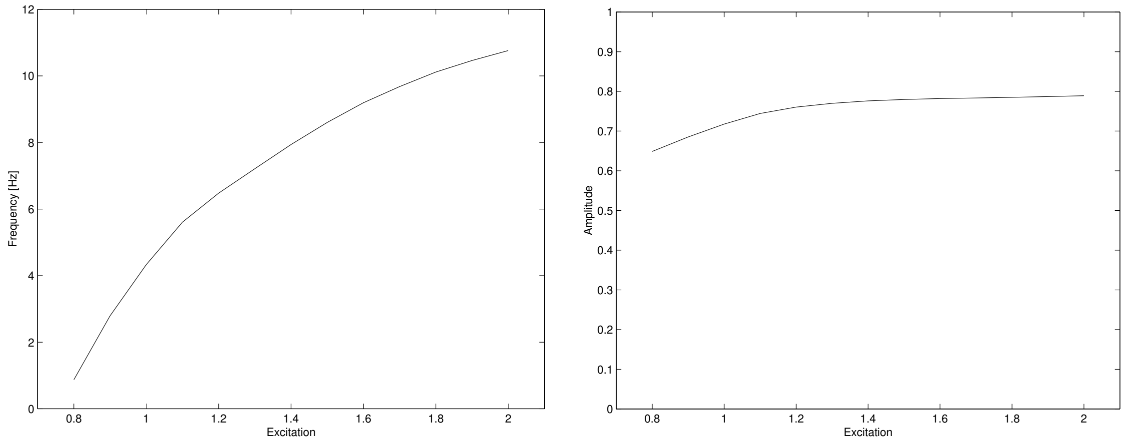


Figure 59: **Run5:** (a) Frequency vs excitation (b) Amplitude vs excitation

	EINl	CCINl	LINl	EINr	CCINr	LINr	BS
EINl	14.3	-	-	-	-1.0	-	-
CCINl	5.2	-	-4.3	-	-1.3	-	14.8
LINl	7.2	-	-	-	-4.2	-	3.4
MNl	2.0	-	-	-	-1.8	-	1.5
EINr	-	-1.0	-	14.3	-	-	-
CCINr	-	-1.3	-	5.2	-	-4.3	14.8
LINr	-	-4.2	-	7.2	-	-	3.4
MNr	-	-1.8	-	2.0	-	-	1.5

Table 47: **Biological connectivity, segmental oscillator**, best of run6

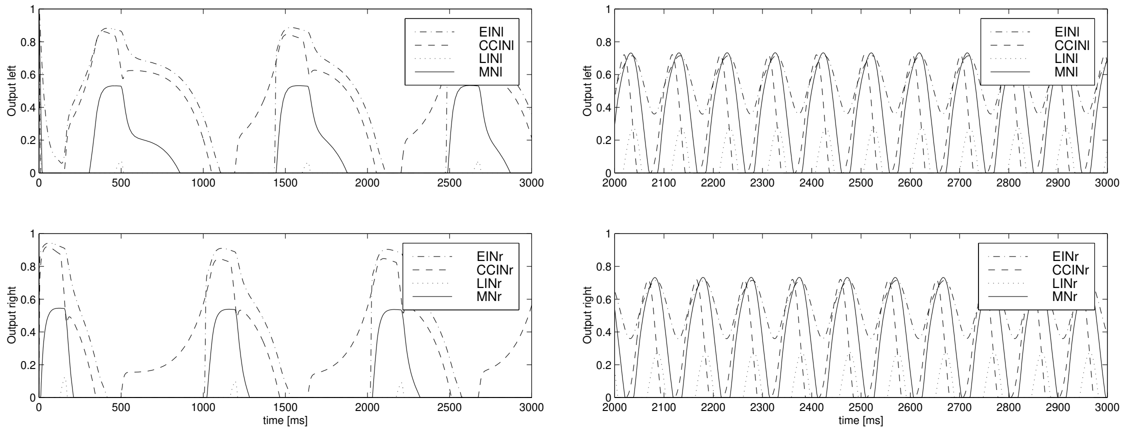


Figure 60: **Run6:** (a) Lowest frequency, (b) Highest frequency

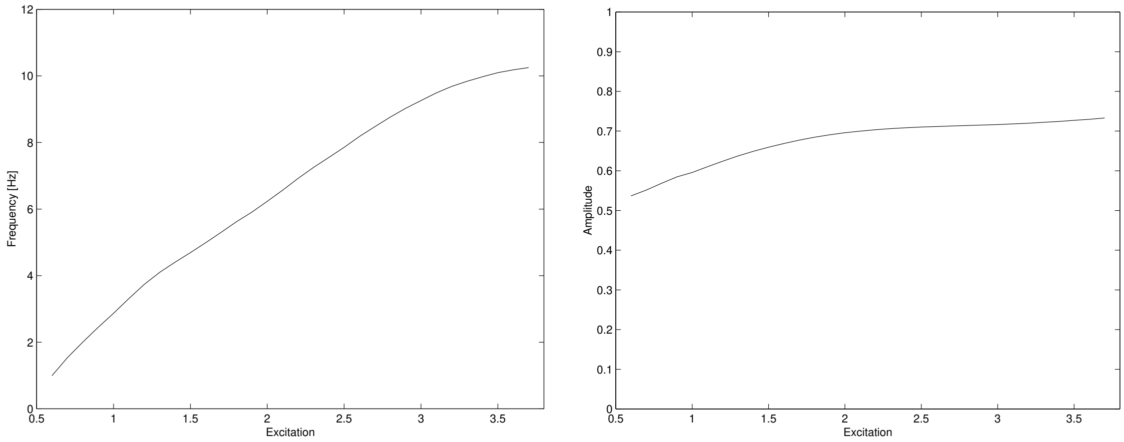


Figure 61: **Run6:** (a) Frequency vs excitation (b) Amplitude vs excitation

	EINl	CCINl	LINl	EINr	CCINr	LINr	BS
EINl	4.5	-	-	-	-1.7	-	6.5
CCINl	6.4	-	-3.3	-	-1.6	-	-
LINl	9.4	-	-	-	-5.0	-	5.4
MNl	1.9	-	-	-	-2.6	-	2.7
EINr	-	-1.7	-	4.5	-	-	6.5
CCINr	-	-1.6	-	6.4	-	-3.3	-
LINr	-	-5.0	-	9.4	-	-	5.4
MNr	-	-2.6	-	1.9	-	-	2.7

Table 48: **Biological connectivity, segmental oscillator**, best of run7

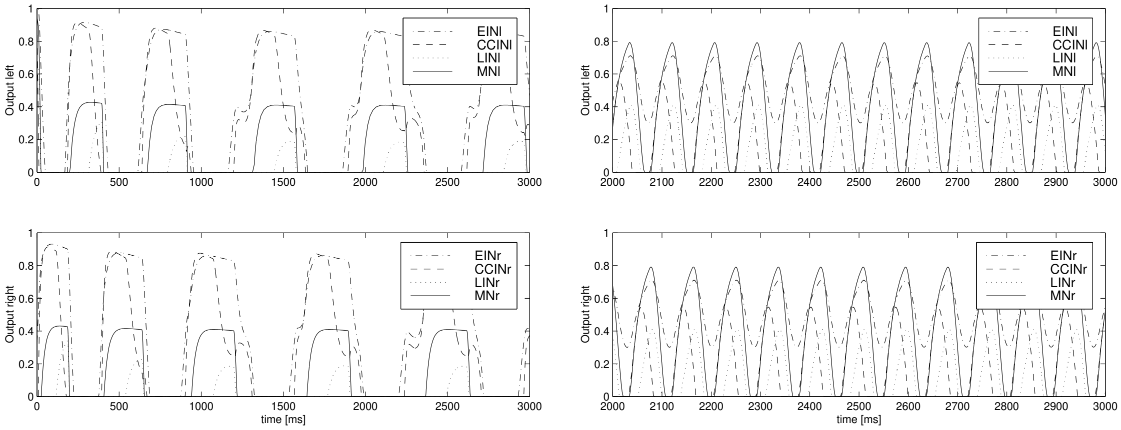


Figure 62: **Run7:** (a) Lowest frequency, (b) Highest frequency

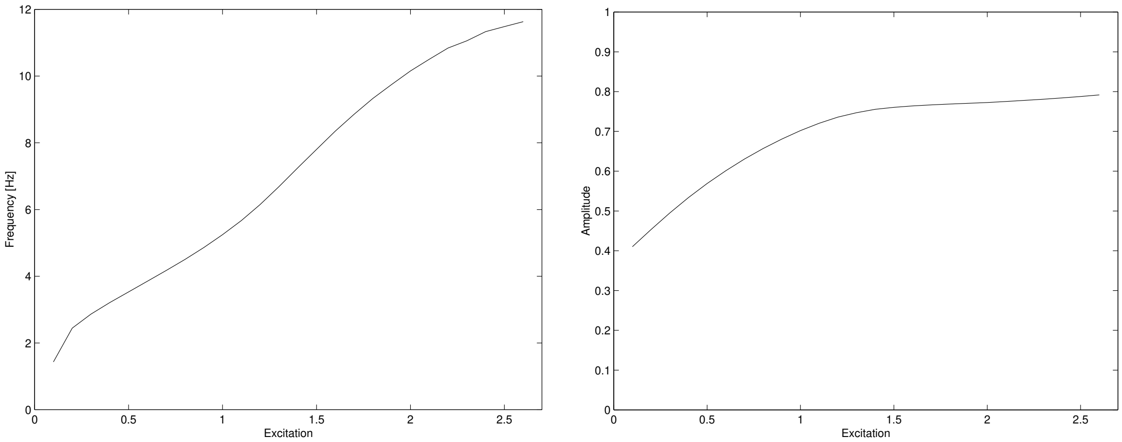


Figure 63: **Run7:** (a) Frequency vs excitation (b) Amplitude vs excitation

	EINl	CCINl	LINl	EINr	CCINr	LINr	BS
EINl	2.4	-	-	-	-2.3	-	9.6
CCINl	10.0	-	-4.1	-	-1.3	-	10.3
LINl	4.2	-	-	-	-4.1	-	6.6
MNl	-	-	-	-	-1.8	-	3.4
EINr	-	-2.3	-	2.4	-	-	9.6
CCINr	-	-1.3	-	10.0	-	-4.1	10.3
LINr	-	-4.1	-	4.2	-	-	6.6
MNr	-	-1.8	-	-	-	-	3.4

Table 49: **Biological connectivity, segmental oscillator**, best of run8

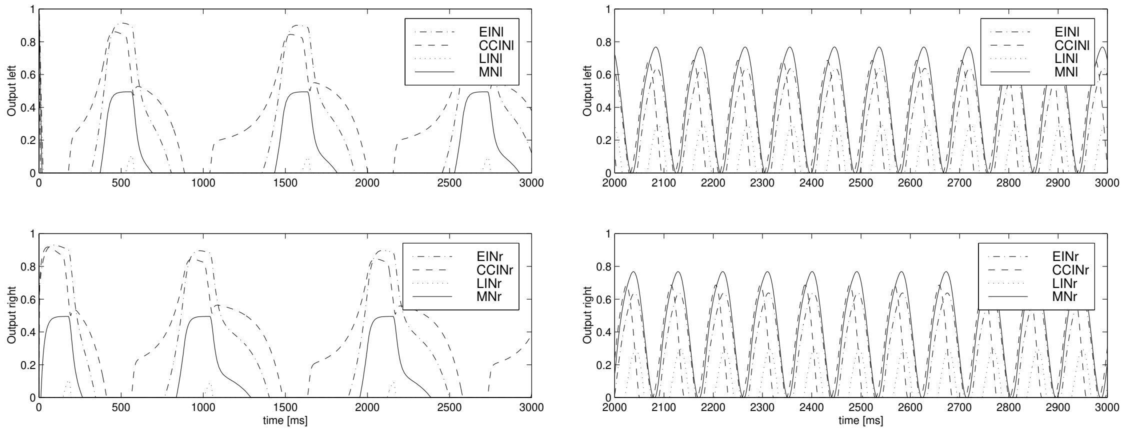


Figure 64: **Run8:** (a) Lowest frequency, (b) Highest frequency

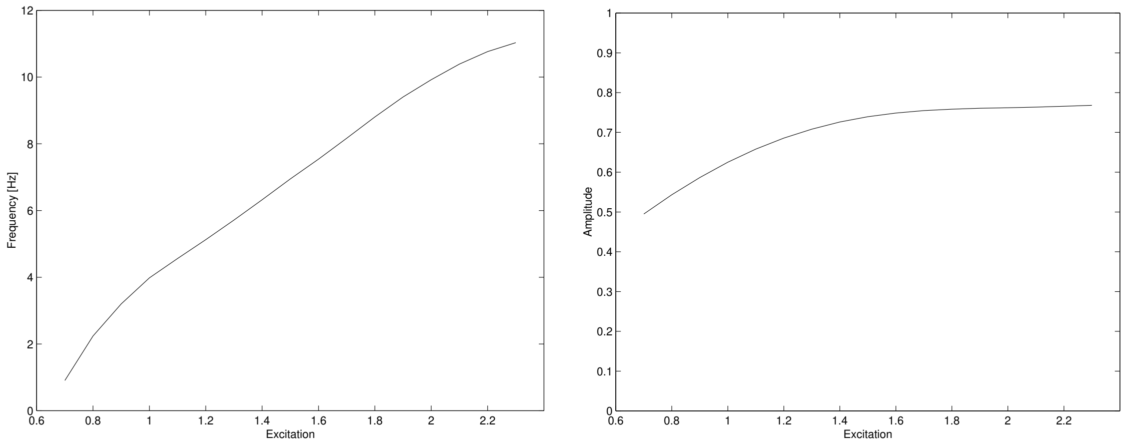


Figure 65: **Run8:** (a) Frequency vs excitation (b) Amplitude vs excitation

	EINl	CCINl	LINl	EINr	CCINr	LINr	BS
EINl	6.6	-	-	-	-3.4	-	1.1
CCINl	-	-	-4.5	-	-1.3	-	10.0
LINl	4.1	-	-	-	-4.3	-	5.8
MNl	-	-	-	-	-2.0	-	3.3
EINr	-	-3.4	-	6.6	-	-	1.1
CCINr	-	-1.3	-	-	-	-4.5	10.0
LINr	-	-4.3	-	4.1	-	-	5.8
MNr	-	-2.0	-	-	-	-	3.3

Table 50: **Biological connectivity, segmental oscillator**, best of run9

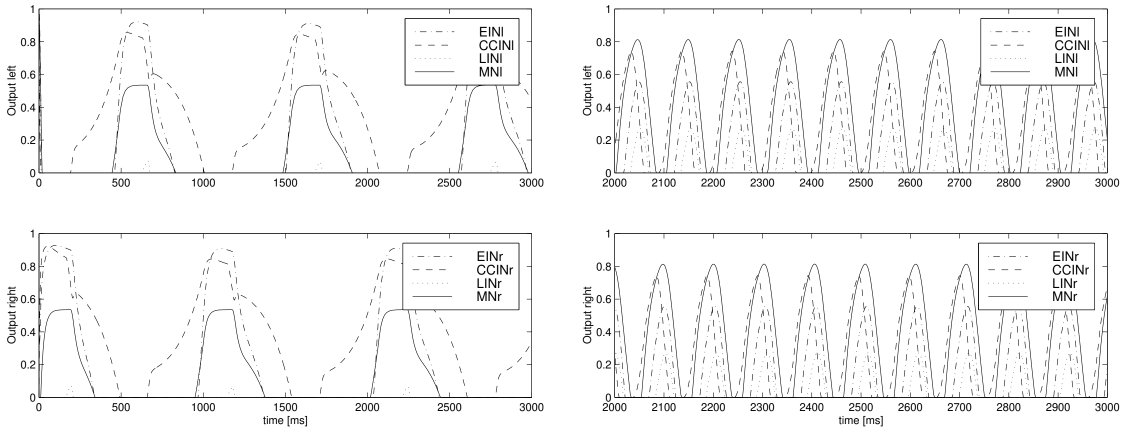


Figure 66: **Run9:** (a) Lowest frequency, (b) Highest frequency

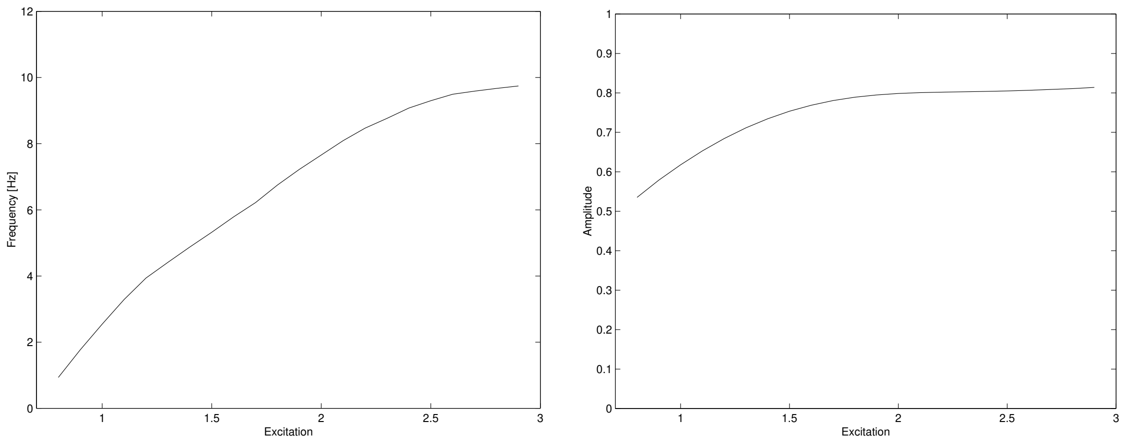


Figure 67: **Run9:** (a) Frequency vs excitation (b) Amplitude vs excitation

	EINl	CCINl	LINl	EINr	CCINr	LINr	BS
EINl	14.3	-	-	-	-1.5	-	-
CCINl	12.0	-	-4.0	-	-1.3	-	13.6
LINl	5.2	-	-	-	-4.1	-	5.3
MNl	-	-	-	-	-1.9	-	3.3
EINr	-	-1.5	-	14.3	-	-	-
CCINr	-	-1.3	-	12.0	-	-4.0	13.6
LINr	-	-4.1	-	5.2	-	-	5.3
MNr	-	-1.9	-	-	-	-	3.3

Table 51: **Biological connectivity, segmental oscillator**, best of run10

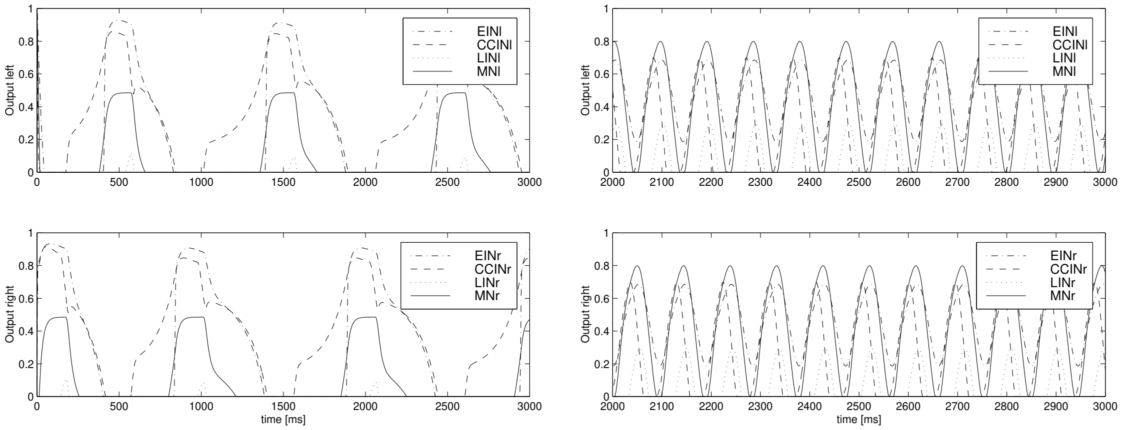


Figure 68: **Run10:** (a) Lowest frequency, (b) Highest frequency

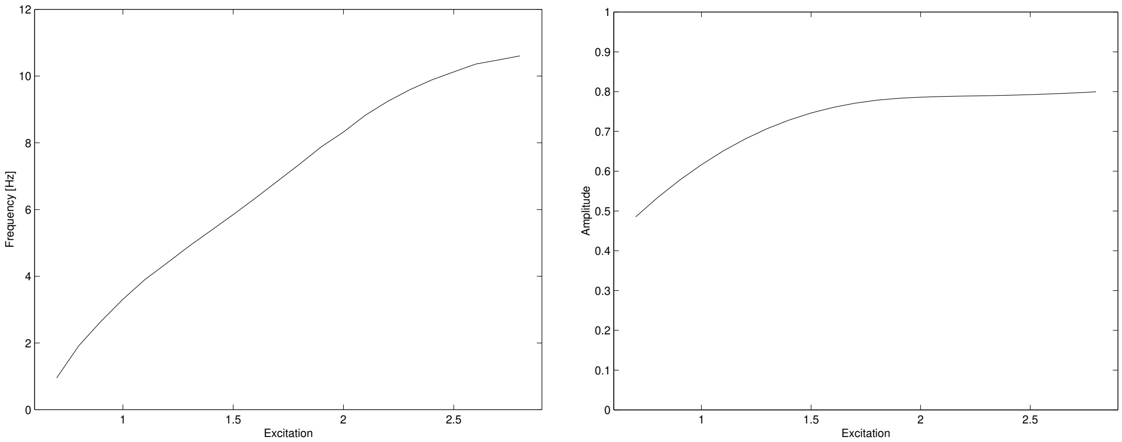


Figure 69: **Run10:** (a) Frequency vs excitation (b) Amplitude vs excitation

F Biological segmental connectivity: evolved multi-segmental controller

	EINl	CCINl	LINl	EINr	CCINr	LINr	BS
EINl	0.4 [10, 2]	-	-	-	-2.0 [10, 5]	-	2.0
CCINl	3.0 [3, 1]	-	-1.0 [5, 7]	-	-2.0 [1, 6]	-	7.0
LINl	13.0 [1, 6]	-	-	-	-1.0 [2, 4]	-	5.0
MNl	1.0 [3, 9]	-	-	-	-2.0 [11, 8]	-	5.0
EINr	-	-2.0 [10, 5]	-	0.4 [10, 2]	-	-	2.0
CCINr	-	-2.0 [1, 6]	-	3.0 [3, 1]	-	-1.0 [5, 7]	7.0
LINr	-	-1.0 [2, 4]	-	13.0 [1, 6]	-	-	5.0
MNr	-	-2.0 [11, 8]	-	1.0 [3, 9]	-	-	5.0

Table 52: **Biological connectivity, inter-segmental coupling**, best of run1

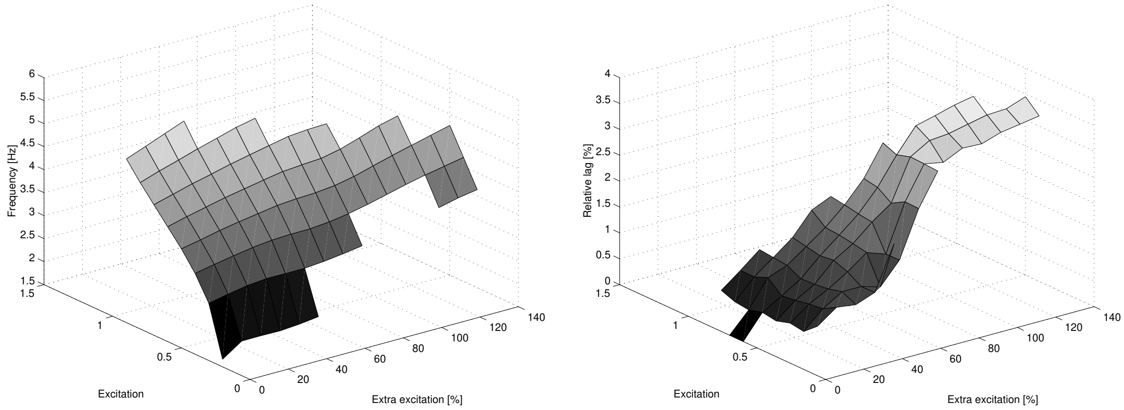


Figure 70: **Run1:** (a) Frequency, (b) Relative lag

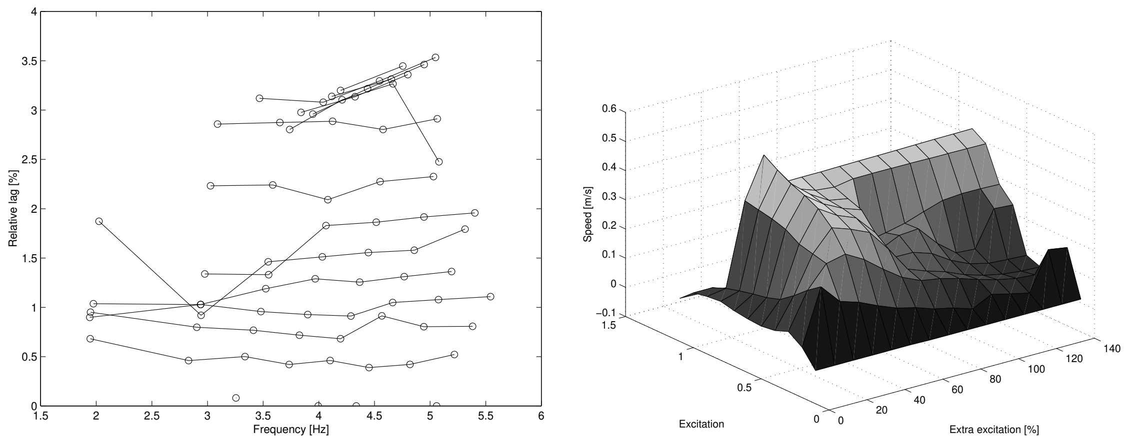


Figure 71: **Run1:** (a) Frequency-lag measurement (b) Speed

	EINl	CCINl	LINl	EINr	CCINr	LINr	BS
EINl	0.4 [5, 3]	-	-	-	-2.0 [1, 0]	-	2.0
CCINl	3.0 [11, 2]	-	-1.0 [3, 9]	-	-2.0 [0, 5]	-	7.0
LINl	13.0 [4, 1]	-	-	-	-1.0 [9, 4]	-	5.0
MNl	1.0 [4, 1]	-	-	-	-2.0 [11, 11]	-	5.0
EINr	-	-2.0 [1, 0]	-	0.4 [5, 3]	-	-	2.0
CCINr	-	-2.0 [0, 5]	-	3.0 [11, 2]	-	-1.0 [3, 9]	7.0
LINr	-	-1.0 [9, 4]	-	13.0 [4, 1]	-	-	5.0
MNr	-	-2.0 [11, 11]	-	1.0 [4, 1]	-	-	5.0

Table 53: **Biological connectivity, inter-segmental coupling**, best of run2

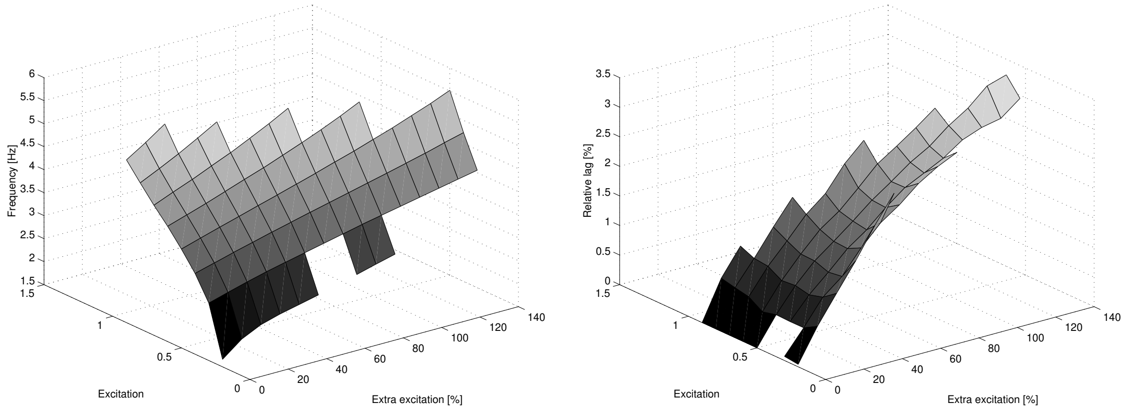


Figure 72: **Run2:** (a) Frequency, (b) Relative lag

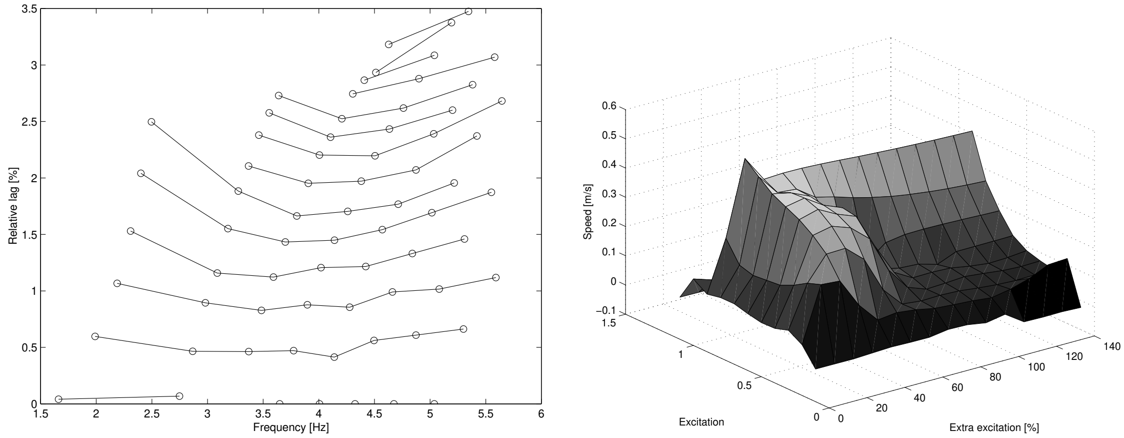


Figure 73: **Run2:** (a) Frequency-lag measurement (b) Speed

	EINl	CCINl	LINl	EINr	CCINr	LINr	BS
EINl	0.4 [1, 0]	-	-	-	-2.0 [9, 5]	-	2.0
CCINl	3.0 [2, 2]	-	-1.0 [5, 8]	-	-2.0 [2, 7]	-	7.0
LINl	13.0 [3, 6]	-	-	-	-1.0 [11, 11]	-	5.0
MNl	1.0 [5, 10]	-	-	-	-2.0 [4, 4]	-	5.0
EINr	-	-2.0 [9, 5]	-	0.4 [1, 0]	-	-	2.0
CCINr	-	-2.0 [2, 7]	-	3.0 [2, 2]	-	-1.0 [5, 8]	7.0
LINr	-	-1.0 [11, 11]	-	13.0 [3, 6]	-	-	5.0
MNr	-	-2.0 [4, 4]	-	1.0 [5, 10]	-	-	5.0

Table 54: **Biological connectivity, inter-segmental coupling**, best of run3

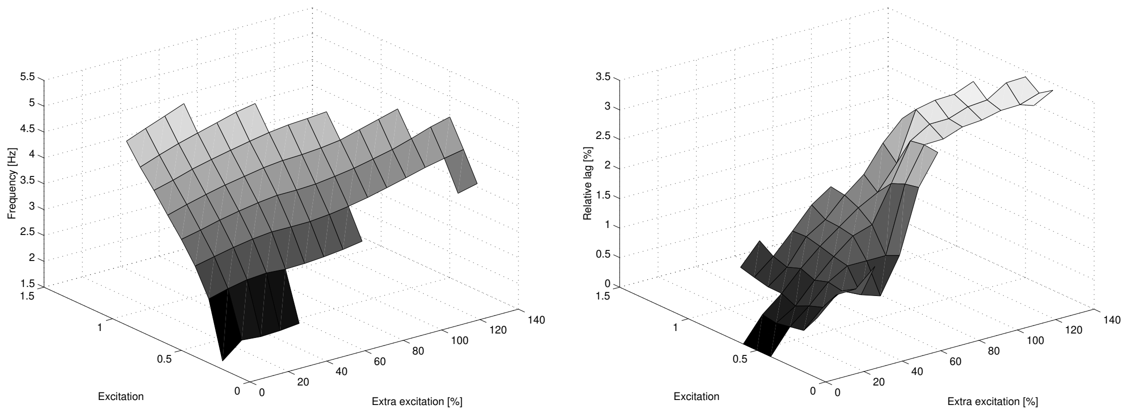


Figure 74: **Run3:** (a) Frequency, (b) Relative lag

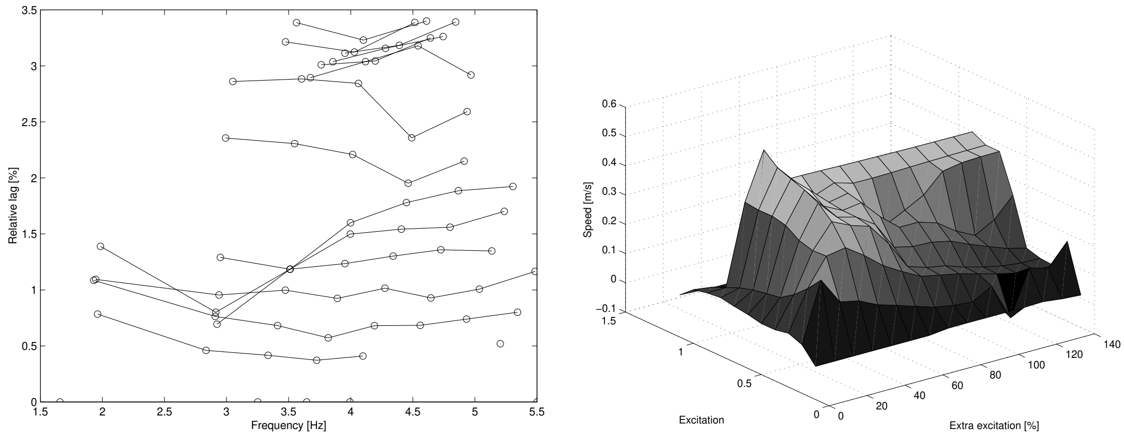


Figure 75: **Run3:** (a) Frequency-lag measurement (b) Speed

	EINl	CCINl	LINl	EINr	CCINr	LINr	BS
EINl	0.4 [11, 5]	-	-	-	-2.0 [8, 3]	-	2.0
CCINl	3.0 [12, 8]	-	-1.0 [1, 3]	-	-2.0 [0, 5]	-	7.0
LINl	13.0 [4, 7]	-	-	-	-1.0 [5, 8]	-	5.0
MNl	1.0 [2, 10]	-	-	-	-2.0 [8, 1]	-	5.0
EINr	-	-2.0 [8, 3]	-	0.4 [11, 5]	-	-	2.0
CCINr	-	-2.0 [0, 5]	-	3.0 [12, 8]	-	-1.0 [1, 3]	7.0
LINr	-	-1.0 [5, 8]	-	13.0 [4, 7]	-	-	5.0
MNr	-	-2.0 [8, 1]	-	1.0 [2, 10]	-	-	5.0

Table 55: **Biological connectivity, inter-segmental coupling**, best of run4

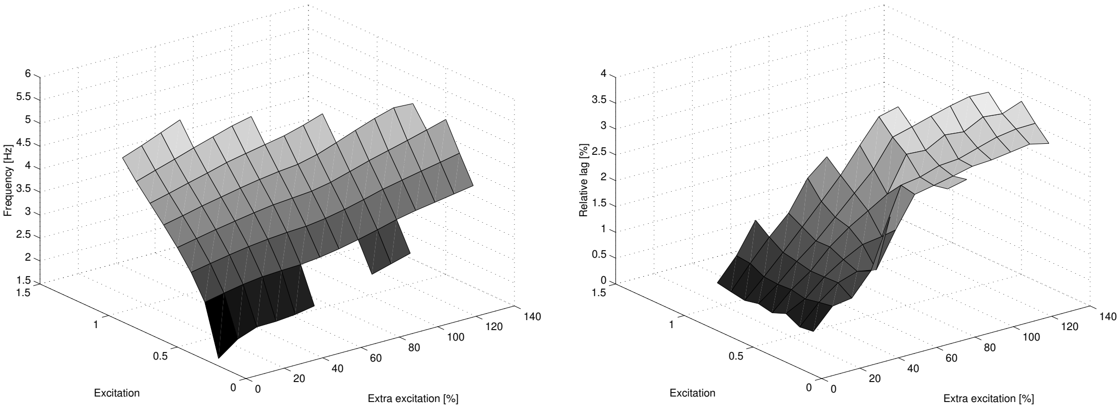


Figure 76: **Run4:** (a) Frequency, (b) Relative lag

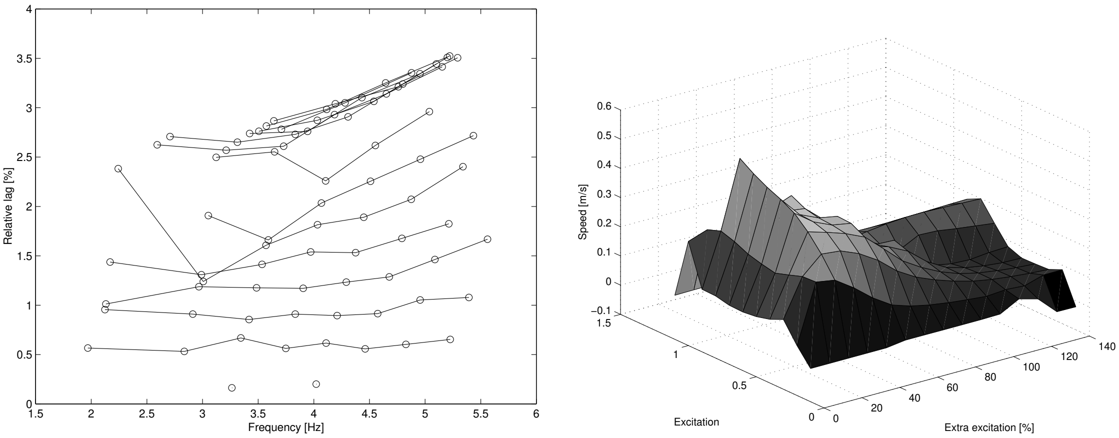


Figure 77: **Run4:** (a) Frequency-lag measurement (b) Speed

	EINl	CCINl	LINl	EINr	CCINr	LINr	BS
EINl	0.4 [5, 2]	-	-	-	-2.0 [4, 2]	-	2.0
CCINl	3.0 [11, 6]	-	-1.0 [7, 7]	-	-2.0 [5, 8]	-	7.0
LINl	13.0 [4, 8]	-	-	-	-1.0 [4, 2]	-	5.0
MNl	1.0 [11, 1]	-	-	-	-2.0 [5, 6]	-	5.0
EINr	-	-2.0 [4, 2]	-	0.4 [5, 2]	-	-	2.0
CCINr	-	-2.0 [5, 8]	-	3.0 [11, 6]	-	-1.0 [7, 7]	7.0
LINr	-	-1.0 [4, 2]	-	13.0 [4, 8]	-	-	5.0
MNr	-	-2.0 [5, 6]	-	1.0 [11, 1]	-	-	5.0

Table 56: **Biological connectivity, inter-segmental coupling**, best of run5

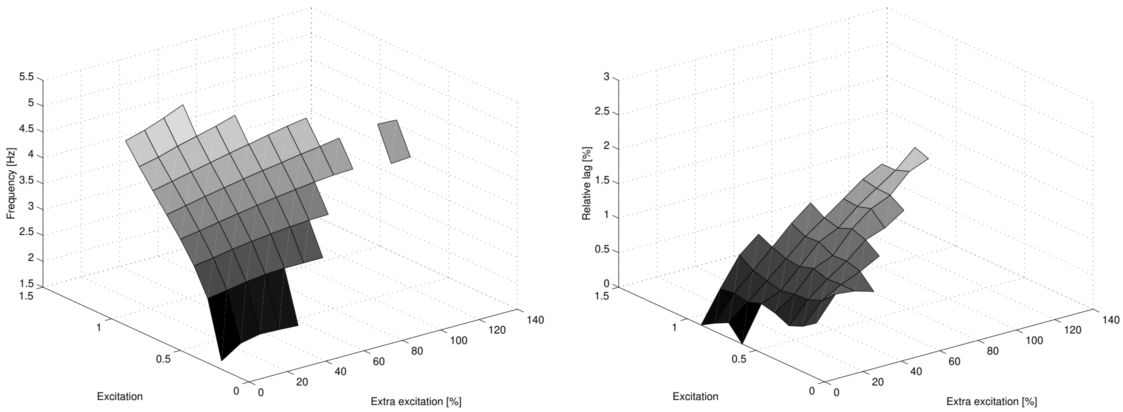


Figure 78: **Run5:** (a) Frequency, (b) Relative lag

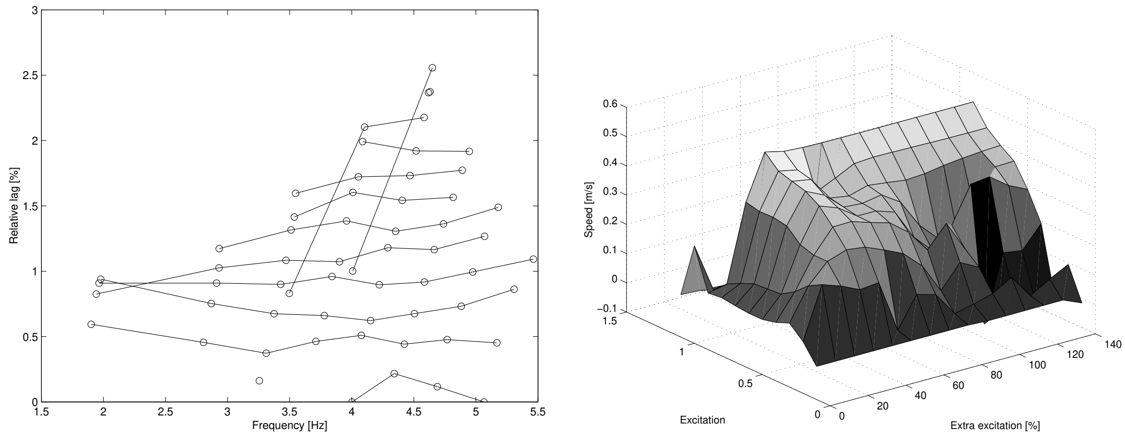


Figure 79: **Run5:** (a) Frequency-lag measurement (b) Speed

G Biological segmental connectivity: evolved sensory feedback

	EINl	CCINl	LINl	EINr	CCINr	LINr	ECl	ECr	BS
EINl	0.4	-	-	-	-2.0	-	-0.4	1.2	2.0
CCINl	3.0	-	-1.0	-	-2.0	-	0.1	-0.2	7.0
LINl	13.0	-	-	-	-1.0	-	4.7	4.7	5.0
MNl	1.0	-	-	-	-2.0	-	4.1	4.0	5.0
EINr	-	-2.0	-	0.4	-	-	1.2	-0.4	2.0
CCINr	-	-2.0	-	3.0	-	-1.0	-0.2	0.1	7.0
LINr	-	-1.0	-	13.0	-	-	4.7	4.7	5.0
MNr	-	-2.0	-	1.0	-	-	4.0	4.1	5.0

Table 57: **Biological connectivity, sensory feedback**, best of run1

	EINl	CCINl	LINl	EINr	CCINr	LINr	ECl	ECr	BS
EINl	0.4	-	-	-	-2.0	-	-0.4	-1.2	2.0
CCINl	3.0	-	-1.0	-	-2.0	-	3.3	-2.0	7.0
LINl	13.0	-	-	-	-1.0	-	1.5	3.6	5.0
MNl	1.0	-	-	-	-2.0	-	0.3	4.3	5.0
EINr	-	-2.0	-	0.4	-	-	-1.2	-0.4	2.0
CCINr	-	-2.0	-	3.0	-	-1.0	-2.0	3.3	7.0
LINr	-	-1.0	-	13.0	-	-	3.6	1.5	5.0
MNr	-	-2.0	-	1.0	-	-	4.3	0.3	5.0

Table 58: **Biological connectivity, sensory feedback**, best of run2

	EINl	CCINl	LINl	EINr	CCINr	LINr	ECl	ECr	BS
EINl	0.4	-	-	-	-2.0	-	-0.8	1.6	2.0
CCINl	3.0	-	-1.0	-	-2.0	-	0.9	-0.9	7.0
LINl	13.0	-	-	-	-1.0	-	0.1	3.2	5.0
MNl	1.0	-	-	-	-2.0	-	4.5	3.5	5.0
EINr	-	-2.0	-	0.4	-	-	1.6	-0.8	2.0
CCINr	-	-2.0	-	3.0	-	-1.0	-0.9	0.9	7.0
LINr	-	-1.0	-	13.0	-	-	3.2	0.1	5.0
MNr	-	-2.0	-	1.0	-	-	3.5	4.5	5.0

Table 59: **Biological connectivity, sensory feedback**, best of run3

	EINl	CCINl	LINl	EINr	CCINr	LINr	ECl	ECr	BS
EINl	0.4	-	-	-	-2.0	-	-0.4	2.5	2.0
CCINl	3.0	-	-1.0	-	-2.0	-	3.1	4.0	7.0
LINl	13.0	-	-	-	-1.0	-	1.5	1.9	5.0
MNl	1.0	-	-	-	-2.0	-	3.2	1.7	5.0
EINr	-	-2.0	-	0.4	-	-	2.5	-0.4	2.0
CCINr	-	-2.0	-	3.0	-	-1.0	4.0	3.1	7.0
LINr	-	-1.0	-	13.0	-	-	1.9	1.5	5.0
MNr	-	-2.0	-	1.0	-	-	1.7	3.2	5.0

Table 60: **Biological connectivity, sensory feedback**, best of run4

	EINl	CCINl	LINl	EINr	CCINr	LINr	ECl	ECr	BS
EINl	0.4	-	-	-	-2.0	-	-1.0	5.0	2.0
CCINl	3.0	-	-1.0	-	-2.0	-	2.4	-0.9	7.0
LINl	13.0	-	-	-	-1.0	-	0.8	-1.0	5.0
MNl	1.0	-	-	-	-2.0	-	3.9	3.4	5.0
EINr	-	-2.0	-	0.4	-	-	5.0	-1.0	2.0
CCINr	-	-2.0	-	3.0	-	-1.0	-0.9	2.4	7.0
LINr	-	-1.0	-	13.0	-	-	-1.0	0.8	5.0
MNr	-	-2.0	-	1.0	-	-	3.4	3.9	5.0

Table 61: **Biological connectivity, sensory feedback**, best of run5

References

[Buchanan & Grillner 87] J.T. Buchanan and S. Grillner. Newly identified 'glutamate interneurons' and their role in locomotion in the lamprey spinal cord. *Science*, 236:312–314, 1987.

[Buchanan 92] J.T. Buchanan. Neural network simulations of coupled locomotor oscillators in the lamprey spinal cord. *Biological Cybernetics*, 66:376–374, 1992.

[Cruse *et al.* 95] H. Cruse, D.E. Brunn, Ch. Bartling, J. Dean, M. Dreifert, T. Kindermann, and J. Schmitz. Walking: A complex behavior controlled by simple networks. *Adaptive Behavior*, 3(4):385–418, 1995.

[Ekeberg 93] Ö. Ekeberg. A combined neuronal and mechanical model of fish swimming. *Biological Cybernetics*, 69:363–374, 1993.

[Ekeberg *et al.* 91] Ö. Ekeberg, P. Wallen, L. Brodin, and A. Lansner. A computer-based model for realistic simulations of neural networks i: The single neuron and synaptic interaction. *Biological Cybernetics*, 65:81–90, 1991.

[Ekeberg *et al.* 95] Ö. Ekeberg, A. Lansner, and S. Grillner. The neural control of fish swimming studied through numerical simulations. *Adaptive Behavior*, 3(4):363–384, 1995.

[Espenschied *et al.* 93] K.S. Espenschied, R. Quinn, H.J. Chiel, and R.D. Beer. Leg coordination mechanisms in the stick insect applied to hexapod robot locomotion. *Adaptive Behavior*, 1(4):455–468, 1993.

[Goldberg 89] D. E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley, 1989.

[Grillner *et al.* 88] S. Grillner, Buchanan J.T., P. Wallen, and L. Brodin. Neural control of locomotion in lower vertebrates. In A. H. Cohen, S. Rossignol, and S. Grillner, editors, *Neural control of rhythmic movements in vertebrates*, pages 1–40. Jon Wiley & Sons, 1988.

[Grillner *et al.* 91] S. Grillner, P. Wallen, and L. Brodin. Neuronal network generating locomotor behavior in lamprey: Circuitry, transmitters, membrane properties, and simulation. *Annu. Rev. Neurosci.*, 14:169–199, 1991.

[Ijspeert *et al.* 97] A.J. Ijspeert, J. Hallam, and D. Willshaw. Artificial lampreys: Comparing naturally and artificially evolved swimming controllers. In *Proceedings of the Fourth European Conference on Artificial Life, ECAL97*, pages 256–265. MIT Press, 1997.

[Jung *et al.* 96] R. Jung, T. Kiemel, and A.H. Cohen. Dynamical behavior of a neural network model of locomotor control in the lamprey. *J. of Neurophysiology*, 75:1074–1086, 1996.

[Kodjabachian & Meyer 95] J. Kodjabachian and J.-A. Meyer. Evolution and development of control architectures in animats. *Robotics and Autonomous Systems*, 16:161–182, 1995.

[Kopell 95] N. Kopell. Chains of coupled oscillators. In M.A. Arbib, editor, *The handbook of brain theory and neural networks*, pages 178–183. MIT Press, 1995.

[Lewis *et al.* 93] M.A. Lewis, A.H. Fagg, and G.A. Bekey. Genetic algorithms for gait synthesis in a hexapod robot. In Y.F. Zheng, editor, *Recent trends in mobile robots*. World Scientific, 1993.

[Lockery & Sejnowski 93] S.R. Lockery and T.J. Sejnowski. The computational leech. *Trends in Neuroscience*, 16(7):283–290, 1993.

[Matsushima & Grillner 92] T. Matsushima and S. Grillner. Neural mechanisms of intersegmental coordination in lamprey: local excitability changes modify the phase coupling along the spinal cord. *J. of Neurophysiology*, 67:373–388, 1992.

[Pearlmutter 95] B.A. Pearlmutter. Gradient calculations for dynamic recurrent neural networks: a survey. *IEEE Transactions on Neural networks*, 6(5):1212–1228, 1995.

[Viana Di Prisco *et al.* 90] G. Viana Di Prisco, P. Wallen, and S. Grillner. Synaptic effects of intraspinal stretch receptor neurons mediating movement-related feedback during locomotion. *Brain Research*, 530:161–166, 1990.

[Wadden *et al.* 97] T. Wadden, J. Hellgren, A. Lansner, and S. Grillner. Intersegmental coordination in the lamprey: simulations using a network model without segmental boundaries. *Biological Cybernetics*, 76:1–9, 1997.

[Wallen *et al.* 92] P. Wallen, Ö. Ekeberg, A. Lansner, L. Brodin, H. Traven, and S. Grillner. A computer-based model for realistic simulations of neural networks ii: The segmental network generating locomotor rhythmicity in the lamprey. *J. of Neurophysiology*, 68:1939–1950, December 1992.

[Williams & Sigvardt 95] T.L. Williams and K.A. Sigvardt. Spinal cord of lamprey: generation of locomotor patterns. In M.A. Arbib, editor, *The handbook of brain theory and neural networks*, pages 178–183. MIT Press, 1995.