Artificial Lampreys: Comparing Naturally and Artificially Evolved Swimming Controllers

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 studies and compares naturally and artificially evolved neural networks controlling the motion of an animal: the anguilliform swimming of a lamprey. The swimming controller of the lamprey, extensively studied and modeled by Grillner and Ekeberg [8, 5], is reproduced. A real number Genetic Algorithm is used to develop alternative artificial controllers composed of neurons similar to those of the biological model. We examine the feasibility of using Genetic Algorithms to evolve neural controllers and evaluate the quality of the evolved networks by comparing them with the biological controller. Results show that artificial controllers can be obtained with architectures other than the biological one, and which, in some ways, are more efficient.

1 Introduction

Interactions between Neuroscience and Artificial Intelligence (AI) are currently growing as inspirations from Neuroscience become frequent in AI and models and techniques from AI are increasingly used in Neuroscience. Our research is particularly interested in neural locomotion controllers for autonomous agents which are biologically inspired and how to develop controllers using Evolutionary Algorithms.

Nature has developed a diversity of very effective motion controllers which are able to create the oscillatory activity necessary for motion as well as to adapt, using sensory information, to dynamic environments. These controllers are generally distributed and present remarkable robustness and flexibility. These interesting features have led roboticists to create biologically inspired neural locomotion controllers for robots or simulated agents [1]. Models of walking controllers in stick insects have, for instance, been used to control hexapod robots [3].

In return, AI techniques can be used to give some insights on Neuroscience measurements and give some hints about the computation performed. In the case of locomotion control, the local bending reflex of the leech has, for instance, been reproduced by a biologically plausible recurrent neural network optimised with an adapted backpropagation algorithm [9].

A recent AI technique to develop adapted controllers is evolving neural configurations using an Evolutionary Algorithm. This technique has been used successfully to develop walking controllers for hexapod agents [2] or biped agents [4], for instance.

This paper examines the lamprey's swimming controller which has been studied extensively by Grillner and his colleagues [8]. The mathematical model of the biological controller given in [5] is reproduced. Artificial controllers are created by using a Genetic Algorithm (GA) to evolve neural networks composed of neurons similar to those of the biological model. There is a double motivation for evolving artificial solutions. The first one is to study the alternative possibilities which Nature could have chosen to obtain controllers with an adapted behaviour, the second one is to evaluate how good a GA is as a tool for creating neural controllers for a specific task. Comparing the new solutions with the biological model will determine the quality of the evolved solutions.

2 Biological Controller

The lamprey is a fish without paired fins which swims by creating an undulation along its body. The corresponding neural activity is created by a Central Pattern Generator (CPG) situated in the spinal cord. The CPG is composed of segments capable of generating oscillatory signals which are coupled together in a way that propagates waves of motoneuron activity from head to tail [8]. Several mathematical models have been developed which are able to reproduce the neural activity measured on real lampreys [8, 5].

The mathematical model of the biological controller given in [5] is simulated, using MATLAB. The model simulates two-dimensional swimming and is based on two simplifications. Firstly, populations of similar real neurons are represented by single neuron units; secondly the output of a neuron unit is not spiking but corresponds to the mean firing frequency of the population ($\in [0, 1]$).

The model presented here corresponds to that model except that stretch sensitive edge cells (EC) are not included. These cells are not necessary for the creation of oscillations and play a role only when the lamprey swims in unstable water where they coordinate the neural activity with actual movements of the body [6].

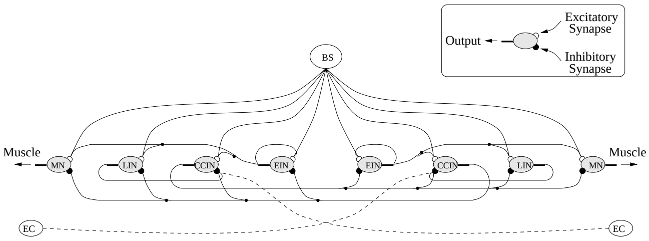

Figure 1: Biological segmental network. A segmental network is made of 8 neurons of 4 different types (see text). The biological model in this paper corresponds to the model described in [5], except that edge cells (EC) are not included. A dash means no connection.

	EINl	CCINl	LINl	EINr	CCINr	LINr	BS
EINl	0.4	-	-	-	-2.0	-	2.0
CCINl	3.0	-	-1.0	-	-2.0	-	7.0
LINl	13.0	-	-	-	-1.0	-	5.0
MNl	1.0	-	-	-	-2.0	-	5.0
EINr	-	-2.0	-	0.4	-	-	2.0
CCINr	-	-2.0	-	3.0	-	-1.0	7.0
LINr	-	-1.0	-	13.0	-	-	5.0
MNr	-	-2.0	-	1.0	-	-	5.0

Table 1: Biological connection weights, as given in [5]. A row corresponds to the weights of the pre-synaptic connections. Excitatory and inhibitory connections are represented by positive and negative weights respectively. Left and right neurons are indicated by *l* and *r*.

Segmental oscillators (Figure 1) are made of 8 neurons of 4 different types: excitatory interneurons (EIN), contralateral inhibitory interneurons (CCIN), lateral inhibitory interneurons (LIN) and motoneurons (MN). All receive excitation from the brainstem (BS). The output of the motoneurons is connected to lateral muscles and waves of motoneuron activity are transformed into waves of muscular contraction and thus undulations of the body. The connection weights between the neuron units are given in Table 1. The self-connections of the EIN neurons express the mutual excitation which happens among the population of neurons represented by the single EIN unit.

A neuron unit is a leaky integrator with a saturating transfer function defined by three differential equations. The output u of such a neuron 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. Neuron types have different times of reaction, thresholds and frequency adaptation. These parameters as well as the connection weights of Table 1 have been defined so that the simulation of the model fits physiological observations [5].

Neuron type	Θ	Γ	τ_D	μ	τ_A
EIN	-0.2	1.8	30 ms	0.3	400 ms
CCIN	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 [5]. 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.

Oscillatory activity appears in segmental networks when an adequate level of excitation is applied from the brainstem. There are no oscillations if the excitation level is too low, and, when the level is too high, the neurons saturate after a few oscillations. In between, the system oscillates with a frequency proportional to the excitation. Our simulations[1] show that frequencies between 1.7 Hz and 5.6 Hz can be obtained, which does not cover the observed range on real lampreys completely (between 0.25Hz and 10Hz [10][2]). A typical oscillation is shown in Figure 2. The frequency of the oscillations in the segmental networks directly determines the speed of the fish in the water. The ability to change frequency is thus an important aspect of the model as it allows the lamprey to control its speed and adapt it to the environment.

[1] The differential equations are solved using a fourth order Runge-Kutta method with adaptive step size.

[2] It has been shown that voltage-sensitive N-Methyl-D-Aspartate receptors play an important role in slow oscillations[10]. As these are not included in this model, the lowest frequencies of the biological range can not be reproduced.

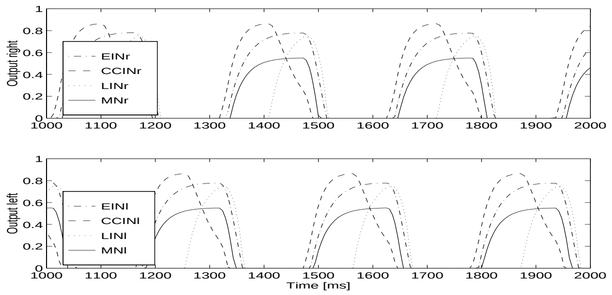

Figure 2: Typical activity of the 8 neurons of the biological segmental network.

A complete controller is composed of approximately 100 interconnected segments. Interconnections are extensions of the segmental connections to (up to ten) neighbour segments. The extent of each connection is given in [5]. Travelling waves of oscillations are created due to an asymmetry of the extensions favouring the caudal direction. Extra excitation on the five segments closest to the head increases the lag between segments and reduces the undulation wavelength. Wavelengths as small as approximately 50% of a 100-segment body can be obtained. The most efficient swimming is obtained with undulations whose wavelength corresponds to the length of the body, and indeed this is what is observed in many fishes [7]. Interestingly, the frequency of the oscillation and the wavelength of the undulation can be changed nearly independently.

This model can also perform turning or even backward swimming. Turning is obtained by exciting one side more than the other over the whole spinal cord, which results in higher amplitudes of signal going to the muscles of one side. Backward swimming is possible by giving some extra excitation to the most caudal (rather than rostral) segments, which causes the travelling wave to go backwards.

3 Evolving controllers

Artificial controllers are developed using a GA. The development of solutions is done in two stages. First segmental oscillators are evolved, then complete controllers are created by evolving the interconnections between fixed segmental networks. The GA encodes the connectivity of potential solutions and solutions are evaluated by simulating their neural activity over a fixed time and rewarding specific behaviours. There is no mechanical simulation of the lamprey's swimming. To evaluate controllers without mechanical simulation is possible, firstly, because no sensory feedback is necessary for the creation of oscillations and, secondly, because the body and muscular structure of the lamprey are sufficiently simple that good and bad controllers can be distinguished by evaluating their neural activity. The main features of controllers on which we will concentrate are the possibilities of varying the frequency of oscillations and the wavelength of spatial undulations. The frequency of oscillations in each segment directly determines the speed of swimming and therefore control of the frequency of oscillations means control of speed. The control of the wavelength of the undulation allows variation of the 'type' of swimming in order to adapt it to the environment and, for instance, to find the most efficient swimming for different external conditions.

3.1 Segmental oscillators

3.1.1 Genetic Algorithm implementation

The same basic GA is used for both design stages. Solutions are encoded in fixed-length chromosomes made of real number genes $\in [0, 1]$. Parents are chosen with a rank-based probability for breeding, and children are created either by a two-point crossover operator (probability *Prob_Xover*) or by simply copying the two parents (probability $1 - Prob\_Xover$). Genes are mutated with a probability *Prob_Mut* and given a new value:
$new\_value = old\_value + Mut\_Range \cdot rand$
where *rand* is a random number $\in [-0.5, 0.5]$. If a new value is outside the range $[0, 1]$, it is set to the closest limit. A selective pressure is created by keeping the size of the population constant and rejecting the worst solutions at each generation. The GA and the simulations are implemented with MATLAB for this design stage.

3.1.2 Encoding Assumptions

Artificial segmental oscillators are created by evolving the connectivity between neurons similar to the biological model. The problem of evolving segmental oscillators is defined by several assumptions:

1. Only the connectivity (i.e. the weights of all the possible connections between neurons) is evolved.

2. The number of neurons is fixed to 8.

3. The types of neurons (parameters of eq. 1–4) are fixed. There are two neurons of each type as in the biological model[3].

4. Weights are bounded between two fixed values, one negative, one positive.

5. Symmetry of the connections between left and right neurons is imposed.

[3] Note that the names of the neuron types EIN, CCIN and LIN describe the function of these neurons in the biological model and this function may change in other configurations. We will keep them only to describe the corresponding neuron parameters.

6. Segmental oscillators have no sensory feedback (no edge cells and no mechanical simulation).

The weight bounds are chosen to include the biological range of weights. The symmetry between left and right neurons reduces the dimension of the search space by a factor of 2.

A segmental oscillator is represented by a chromosome through a direct encoding of each connection weight into a gene, a real number between 0 and 1. A chromosome is translated into a configuration by linearly transforming the gene's value into a value between the two fixed weight bounds and transforming the rescaled vector into the matrix giving the weights of all the possible connections.

3.1.3 Fitness function

The evaluation of the fitness of a segmental network is based on qualitative features of the neural activity of motoneurons. Only these neurons are considered as their signals determine the muscular activity along the body. An evaluation consists of fixed-length simulations with asymmetric initial conditions[4].

A fitness function is developed which rewards the following behaviour for the motoneurons activity:

1. regular oscillations,
2. periods corresponding to an alternation of an active and a resting phase, with a single peak of activity per period,
3. opposite behaviour between left and right motoneurons,
4. monotonic relation between excitation and oscillation frequency,
5. large range of frequencies.

As already mentioned, controllers with a large range of frequencies are rewarded because the frequency of oscillations determines the speed of swimming and therefore large ranges of frequency offer more flexibility in the control of speed. Note that because the activity in the interneurons (neurons other than the motoneurons) is not considered, oscillators with fewer than 8 active neurons can be developed.

The mathematical definition of the fitness is as follows:

$$\textit{fitness} = \begin{cases} \textit{fit\_oscil} \cdot & (1 + \textit{freq\_range}) \\ & \text{if } \textit{fit\_oscil} > 0.5 \\ \textit{fit\_oscil} & \text{otherwise} \end{cases}$$

where

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

The function *fit_oscil* rewards the three first points of the desired behaviour of a solution, with the functions *fit1, fit2* and *fit3* rewarding respectively varying outputs, regularity and opposite behaviour between left and right motoneurons. These three functions are bounded between 0.05 (bad behaviour) and 1 (good behaviour), and vary linearly between these values for one or several variables. The bounds for each variable, determining when the value of the variable is bad, good or in between, 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 0.5 is thus determined above which a solution is certain to oscillate regularly with opposite behaviour between left and right motoneurons[5].

The value *freq_range* corresponds to the range of frequency (in Hz) in which the solutions oscillate regularly (*fit_oscil* higher than 0.5) with a frequency increasing with the excitation level. An evaluation consists thus of a first simulation at a fixed level of excitation (equal to 1) which determines *fit_oscil*, followed, if *fit_oscil* is higher than 0.5, by a set of simulations at different excitation levels (0.1 steps) in order to determine *freq_range*.

3.1.4 Results

As first tests showed that there existed many different solutions with similar fitness values, three experiments were done with different encodings in order to study different neural configurations: encoding all the possible connections between the 8 neurons; encoding all the possible connections except feedback from motoneurons; and encoding only the biological connections. For each experiment, five evolutions are performed with different initial populations of 100 chromosomes. The number of generations for each experiment is chosen so that frequency ranges higher than 7Hz are reached in all runs. An *elite population* is then created, made of the 20 best solutions of each initial evolution. This new population is then further evolved until frequency ranges higher than 14Hz are reached. The basic GA parameters for each experiment are given in Table 3.

Experiment1, fully connected network: All the possible connections between the 8 neurons and the brainstem (without feedback from neurons to the brainstem) are encoded, resulting in 72 possible connections[6]. Due to the left/right symmetry assumption, a chromosome has thus 36 genes. This encoding allows feedback

[4] All the left neurons of a segment are excited.

[5] On 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.

[6] Each neuron can have a self-connection.

Population size	100
Number of children	30
Weight bounds	[-5,15]
Crossover probability	0.5
Mutation probability	0.4
Mutation range	0.4

Table 3: GA parameters for evolving segmental oscillators

from the motoneurons to the other neurons, which is not the case in the biological model where the only output connection from motoneurons goes to muscles.

As mentioned, the evolutions all started with different initial populations. Within these 500 randomly generated configurations, only 4 produce varying outputs (of which one could be a potential controller, i.e. a solution with regular asymmetric oscillations and variable frequency). This means that approximately 1% of the 36-dimension variable space sampled here corresponds to configurations with varying outputs among which the GA must find interesting controllers.

After 200 generations, all evolutions successfully converged to potential controllers, which produce regular oscillations, opposite behaviour between left and right motoneurons and variable frequency. There exists a large diversity of controller configurations and behaviours within the final populations. Configurations vary in terms of weight values, connection types and number of active neurons; none is similar to the biological configuration nor to the biological configuration with a swap of function between the neurons types. The number of neurons participating in the oscillations varies between 4,6 and 8. The signal shapes of the neurons other than motoneurons (the interneurons) can be very different, with for instance, signals that have more than one peak per period, or that oscillate without resting phases.

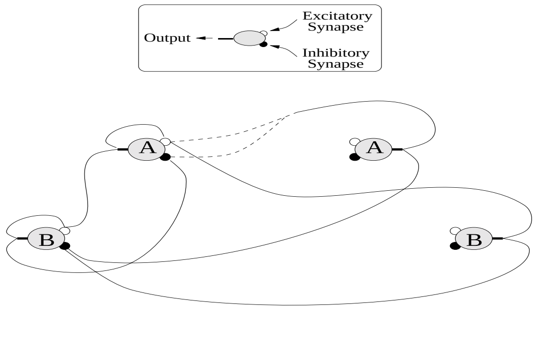

Figure 3: Four-neuron oscillator. Variations (broken lines) of a 4-neuron structure (2 neurons A and 2 neurons B) found in several evolved segmental oscillators. Only the input connections of the 2 left neurons are given, the others can be found by symmetry.

The frequency ranges of the best solutions of each evolution after 200 generations lay between 7.1 Hz and 9.6 Hz. These results are summarised in Table 7. These five best solutions are able to create the oscillations with fewer than 8 neurons (either 6 or 4 neurons) for most of the frequencies in the frequency range. Some neurons are active at low frequencies and do not participate (they stay at zero) in the oscillations at higher frequencies. These neurons can be removed and the solutions still oscillate at most of the frequencies except the lowest. Interestingly, the resulting 4-neuron configurations of runs 1, 2 and 3 present a similar structure (when the left/right symmetry is taken in account and the type of neuron is ignored), having the same type of connections, excitatory or inhibitory, for all the connections, except one for the solution of run1. These runs seem thus to have converged to solutions built on a similar 4-neuron oscillator structure (Figure 3).

The best solution after evolving the *elite population* for 150 generations, has a very large frequency range, 16.4Hz (from 2.8Hz to 19.2Hz), which is thus more than 4 times larger then the frequency range of the biological model (3.9Hz). The behaviour and the connectivity of that solution are given in Figure 4 and Table 4.

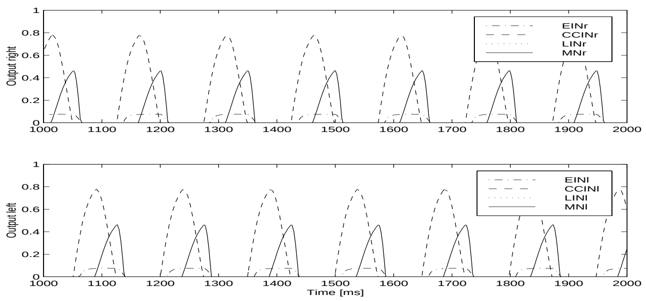

Figure 4: Fully connected network: Simulation of the best solution of exp.1. Frequency=6.7Hz. At higher frequencies, only the MN and CCIN neurons are active.

	EINl	CCINl	LINl	MNl	EINr	CCINr	LINr	MNr	BS
EINl	-5.0	3.2	11.2	8.3	-0.7	-1.4	-5.0	7.0	-2.7
CCINl	10.0	14.7	10.4	-3.2	-3.0	-1.7	0.7	3.6	10.2
LINl	5.0	2.0	5.9	8.0	12.3	8.3	-3.2	-3.9	4.7
MNL	-5.0	1.9	10.3	15.0	2.3	-4.9	7.3	-0.7	9.2
EINr	-0.7	-1.4	-5.0	7.0	-5.0	3.2	11.2	8.3	-2.7
CCINr	-3.0	-1.7	0.7	3.6	10.0	14.7	10.4	-3.2	10.2
LINr	12.3	8.3	-3.2	-3.9	5.0	2.0	5.9	8.0	4.7
MNr	2.3	-4.9	7.3	-0.7	-5.0	1.9	10.3	15.0	9.2

Table 4: Fully connected network: Connection weights of the best solution of exp.1. A row corresponds to the weights of the pre-synaptic connections. This solution is based on a four-neuron oscillator made of MN and CCIN neurons.

The oscillations of this solution are mainly due to the interaction between CCIN neurons and motoneurons. By removing the other neurons, it is possible to create a solution which presents identical behaviour, except at low frequencies, and which oscillates at frequencies between 6.9Hz and 19.2Hz[7]. Note that such a controller with only 4 neurons is only possible because of the feedback from motoneurons. This 4-neuron structure is identical to that of the best solution of run1 in terms of connection types (excitatory or inhibitory).

Experiment2, fully connected network without MN feedback: Here all the possible connections are encoded except that feedback from motoneurons to the other neurons is not allowed. This situation is thus closer to the biological model in which there is no such feedback and in which the motoneurons do not participate in the creation of oscillations. 56 connections are thus encoded in 28-gene chromosomes. Within the initial 500 randomly generated configurations, only 3 produce varying outputs (less than 1% of the 28-dimension variable space) of which none could be a potential controller.

All evolutions, except one, successfully converged, within 150 generations, to potential controllers, which produce regular oscillations, opposite behaviour between left and right motoneurons and variable frequency. The failed evolution converged prematurely to a local maximum corresponding to a non-oscillating solution and did not manage to improve it within 150 generations. Again a diversity of neural configurations has been found.

The frequency ranges of the best solutions of the four other evolutions after 150 generations lay between 10.3 Hz and 17.5 Hz (See summary in Table 7). The oscillations of the best solution of run 5 are created by the EIN and CCIN neurons (LIN neurons stay inactive, at all frequencies). The corresponding 4-neuron structure is similar in terms of connection types (excitatory or inhibitory) to that underlying the oscillations of the best solutions of run1 and the evolved elite population of the first experiment (Figure 3).

The elite population is evolved for 80 extra generations. A best solution is thus created whose frequency range is 21.0Hz (from 2.8Hz to 23.8Hz) which is more than 5 times larger then the frequency range of the biological model. The behaviour and the connectivity of that solution are given in Figure 5 and Table 5. The complete controllers evolved in the second design stage will be based on this segmental network.

Experiment3, biological connections: In this experiment only the biological connections are encoded (the others are set to zero) and the bounds are fixed such that the types of connection, excitatory or inhibitory, are identical to the biological model. The 26 connections are encoded in 13-gene chromosomes. Within the initial 500 randomly generated configurations, 76 (13% of the sampled variable space) produce varying outputs of which 64 could be potential controllers (regular asymmetric oscillations and variable frequency). Having only the biological connections and fixing their type, excitatory or inhibitory, thus restricts the variable space to a much more favourable search space than having the complete connectivity encoded.

Within only 50 generations, all evolutions converged to interesting solutions. The shapes of the signals are very similar to those of the biological simulations, except for their amplitudes. The frequency range of the best solutions lay between 8.5Hz and 10.7Hz. This means that the range of frequencies of the biological model can be improved by changing the values of its connections a little and using a fitness function which optimises the frequency range. A general observation is that this improvement is obtained by increasing the strengths of the connections. Interestingly, the best solutions have all very similar weights for the inhibitory connections and have converged to a common underlying inhibitory structure.

After evolving the *elite population* for 50 extra generations, a best solution is created (Figure 6 and Table 6)

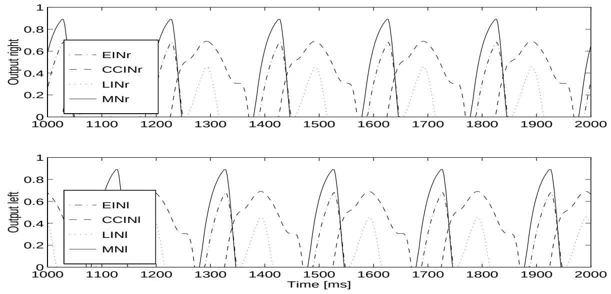

Figure 5: Fully connected network without MN feedback: Simulation of the best solution of exp.2. Frequency=5.0Hz.

	EINl	CCINl	LINl	EINr	CCINr	LINr	BS
EINl	12.5	-5.0	-4.4	6.9	10.8	14.1	9.1
CCINl	7.6	2.7	12.8	-0.5	0.9	-5.0	-0.2
LINl	10.3	9.6	2.7	-5.0	-1.3	-3.1	14.4
MNL	14.8	-5.0	-3.4	-1.6	14.9	1.1	14.2
EINr	6.9	10.8	14.1	12.5	-5.0	-4.4	9.1
CCINr	-0.5	0.9	-5.0	7.6	2.7	12.8	-0.2
LINr	-5.0	-1.3	-3.1	10.3	9.6	2.7	14.4
MNr	-1.6	14.9	1.1	14.8	-5.0	-3.4	14.2

Table 5: Fully connected network without MN feedback: Connection weights of the best solution of exp.2. A row corresponds to the weights of the pre-synaptic connections.

[7] This shows that, in these fully connected solutions, some connections or even some neurons are not necessary for creating oscillations. One possibility to create solutions with only the necessary connections (results not shown) is to add a mutation which randomly sets some connections to zero and to add a factor to the fitness function rewarding solutions with reduced connectivity.

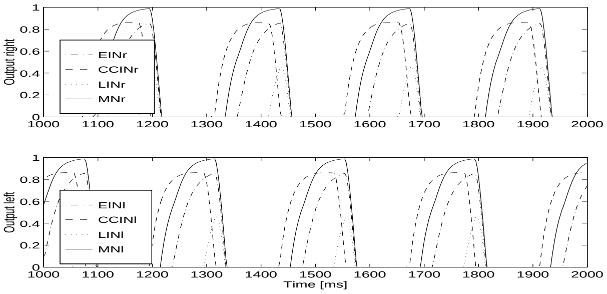

Figure 6: Biological connections: Simulation of the best solution of exp.3. Frequency=4.2Hz.

	EINl	CCINl	LINl	EINr	CCINr	LINr	BS
EINl	2.8	-	-	-	-4.5	-	1.7
CCINl	1.5	-	-4.3	-	-1.4	-	15.0
LINl	10.5	-	-	-	-4.4	-	11.5
MNL	14.7	-	-	-	-3.4	-	13.8
EINr	-	-4.5	-	2.8	-	-	1.7
CCINr	-	-1.4	-	1.5	-	-4.3	15.0
LINr	-	-4.4	-	10.5	-	-	11.5
MNr	-	-3.4	-	14.7	-	-	13.8

Table 6: Biological connections: Connection weights of the best solution of exp.3. A row corresponds to the weights of the pre-synaptic connections. The runs 1 to 5 converged to solutions whose inhibitory weights are very similar to this solution. There seems to be a common underlying inhibitory structure which optimises the frequency range.

whose frequency range is 11.9Hz (from 2.1Hz to 14.0Hz), which is approximately three times the range of the biological model and covers better the range of frequency observed in real lampreys, from 0.25Hz to 10Hz approximately. As observed before, the connections are stronger than in the biological model (absolute value of the weights on average 88% larger than the biological values).

Note that the weights of the biological model (in [5]) have been set by hand in order to fit the biological data (Ekeberg, personal communication). As the model is based on two important simplifications (a whole population of neurons is represented by one mathematical neuron unit and the output of a neuron unit is not a spiking action potential but the mean firing frequency), the weights of the biological model have no biological meaning except for showing that the model is able to reproduce the observed physiological behaviour of the CPG. Results shown here may give some insights into the strengths of the connections relative to each other in order to create oscillations which cover a range of frequencies which is closer to that observed in real lampreys.

Exp.	Run	N. of gen.	Range in Hz	from [Hz]	to [Hz]	Not oscil.
1	1	200	7.7	5.5	13.2	EIN,LIN
	2	200	8.4	1.9	10.3	EIN,CCIN
	3	200	7.1	2.4	9.5	EIN,CCIN
	4	200	9.6	4.2	13.8	CCIN
	5	200	7.9	3.0	10.9	CCIN
	elite	150	16.4	2.8	19.2	EIN,LIN
2	1	150	12.9	2.6	15.5	-
	2	150	-	-	-	-
	3	150	16.0	3.7	19.7	-
	4	150	10.3	2.7	13.0	-
	5	150	17.5	2.5	20.0	LIN
	elite	80	21.0	2.8	23.8	-
3	1	50	10.7	2.3	13.0	-
	2	50	10.2	1.7	11.9	-
	3	50	8.5	2.0	10.5	-
	4	50	8.8	2.1	10.9	-
	5	50	9.4	1.7	11.1	-
	elite	50	14.1	2.0	16.1	-

Table 7: Evolved segmental networks, summary of results. This table gives the range of frequency of the solutions with highest fitness value of each evolution. The neurons which do not oscillate for some of the excitation levels (usually the highest) are indicated.

3.2 Complete controllers

Multi-segmental controllers are developed by evolving the interconnections between fixed segmental oscillators. Two preliminary evolutions are realised, one with an evolved segmental network and one with the segmental network of the biological model. The first evolution will create a complete artificial controller. The evolved segmental network with the largest frequency range is chosen (best solution of experiment 2). The aim of the second evolution is to study the interconnectivity of the biological model. Because the physiological interconnections are not perfectly known [5], evolving the interconnections of the biological model may show whether there are several possibilities for interconnecting the biological segmental networks and creating travelling waves.

3.2.1 Genetic Algorithm implementation

The same basic GA as in the segmental oscillator design stage is used here, except that genes are transformed into integers representing the extent of an interconnection and that the GA and the simulations are implemented in C code.

3.2.2 Encoding assumptions

As in the biological model, segmental networks are interconnected through extensions of connections within a segmental network to neighbour segments. This means that a neuron which is connected to (rather, whose output is sent to) another neuron in one segment, can also have extensions to the corresponding neuron in neighbouring segments. The extent of these interconnections varies with each segmental connection. A connection weight is rescaled by dividing the weight of the connection in the segmental network by the number of neighbour segments it receives input from[8].

The encoding of a complete controller is based on the following assumptions:

1. The weights of the segmental network are fixed, except for the rescaling mentioned above.
2. Only the extents (in the rostral and caudal directions) of the interconnections are evolved.
3. These extents vary between zero and a fixed limit.
4. Symmetry of the interconnections between left and right neurons is imposed.

A complete controller is decoded from a chromosome by transforming and rounding a gene's value into an integer between 0 and the fixed maximum extent and transforming the resulting vector into the two matrixes giving the extent of each segmental connection in the rostral and caudal directions. The maximum extension is chosen to permit the maximum biological extension (10 segments).

3.2.3 Fitness function

As for the development for segmental oscillators, the evaluation of the fitness function is based on qualitative features of the neural activity of motoneurons.

We would like the complete controller to be such that

1. each segment oscillates regularly,
2. waves of neural activity propagate from head to tail,
3. the wavelength of the undulation can be varied by changing the amount of extra excitation on the segments closest to the head.

An evaluation consists of two simulations of a 100-segment controller with two different amounts of extra excitation on the five first segments, 0% and 100% of the level of excitation of the other segments (excitation equal to 1.0). The fitness is calculated as follows:

$$\mathit{fitness} = \begin{cases} \mathit{oscil\_behav}\cdot & (1 + \mathit{Lag(100\%)} - \mathit{Lag(0\%)}) \\ & \text{if } \mathit{oscil\_behav} > 0.5 \\ & \text{and both } \mathit{Lag()} \geq 0 \\ \mathit{oscil\_behav} & \text{otherwise} \end{cases}$$

where *oscil_behav* is calculated by measuring *fit_oscil* in segments 1,10,20,...,100 for both evaluations (with and without extra excitation) and taking the minimum measured value. The lag values correspond to the lag per segment relative to the period of oscillation, in percent (values typically vary between 0% and 2.5%). Note that the range of lags is not explicitly rewarded, but only the range between two fixed levels of extra excitation.

3.2.4 Results

The GA parameters of both experiments are given in Table 8. The evolutions of each experiment are stopped when wavelength ranges larger than that of the biological model are reached.

Population size	20
Number of children	6
Extensions bounds	[0,12]
Crossover probability	1.0
Mutation probability	0.4
Mutation range	0.4

Table 8: GA parameters for evolving complete controllers

Complete controller with evolved segmental network A complete artificial controller is created by evolving the interconnections between an evolved segmental network. The best solution of experiment 2 (Table 5), the solution with the largest frequency range of the three experiments, is chosen. As this segmental network has 48 connections between the 8 neurons, the dimension of the search space[9] for both rostral and caudal extensions is 48.

An evolution of a population of 20 chromosomes is realised for 10 generations. Within the initial randomly generated population, 17 of the 20 configurations have regular oscillations in all segments for both levels of extra excitation showing that such a segmental network can be interconnected in different ways and still oscillate regularly. However, most of these solutions present only very small lags, some of them going from the tail to the head. Others create travelling waves whose wavelength is not changed by the level of extra excitation. Only six solutions have a reasonable range of lags per segment (higher than 0.2% of the period).

After 10 generations only, the best solutions have ranges of lags per segment up to 2.3% of the period of oscillation, which is a little bit larger than the range of the biological model. All the solutions except one have extensions favouring the caudal direction on average, as is the case for the biological model. The best solution (Table 9) has lags per segment varying almost linearly with the amount of extra excitation on the first segments and lying between 0.1%(no extra excitation) and

[8] This rescaling compensates the weights for the neurons in the first and last segments which receive less input because they have fewer rostral and caudal extensions respectively.

[9] There are 48 possible extensions in both directions, and these are encoded in 48-gene chromosomes because of the symmetry assumption.

2.4%(100% extra excitation) of the period. Wavelengths as small as 42% of a 100-segment body can thus be obtained. The shortest wavelength of the biological model is approximately 50% of the body length. Figure 7 shows an example of the neural activity of the best evolved controller.

	EINl	CCINl	LINl	EINr	CCINr	LINr
EINl	6:10	0:2	4:1	4:10	0:9	6:6
CCINL	11:5	6:9	4:3	4:10	7:1	8:9
LINl	8:8	10:4	7:3	11:5	2:10	0:11
MNl	10:4	12:9	6:8	9:9	12:11	0:9
EINr	4:10	0:9	6:6	6:10	0:2	4:1
CCINr	4:10	7:1	8:9	11:5	6:9	4:3
LINr	11:5	2:10	0:11	8:8	10:4	7:3
MNr	9:9	12:11	0:9	10:4	12:9	6:8

Table 9: Complete evolved controller: rostral:caudal extensions of the best solution. There is an average asymmetry of the interconnections favouring the caudal direction.

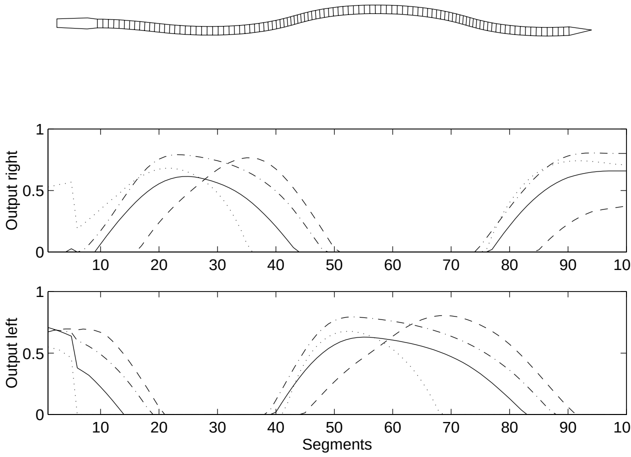

Figure 7: Complete evolved controller: Simulation of the best solution. A snapshot of the neural activity in a 100-segment body (800 neurons) is represented. A wave of neural activity, and therefore an undulation of the body, is created because of time lags between each segment. As in the biological model, these lags are due to the extra excitation on the five first segments and to the average asymmetry of the interconnections favouring the caudal direction. The body represented here is simply a set of trapezia whose parallel lengths are inversely proportional to the signal of the left and right motoneurons of each segment.

This preliminary experiment has shown that it is possible to evolve complete artificial controllers using a GA. The efficiency of swimming with this controller should be checked with a mechanical simulation of the body. Initial tests show that this artificial model is able to induce turning when one side of the spinal cord is more excited than the other, leading to different amplitudes of motoneuron signal, as with the biological model.

Complete controller with biological segmental network The extensions of the 18 connections between the 8 neurons of the biological segmental network are encoded into a 18-gene chromosome. An evolution of a population of 20 chromosomes is realised for 40 generations.

Again, there seem to be many possibilities for interconnecting the biological segmental network which result in regular oscillations in all segments. After 40 generations the population evolved to solutions with ranges of lags per segment similar to the biological model. The solutions all present an average asymmetry of interconnections favouring the caudal direction. There is a variety of different solutions among the final population, but none is similar to the complete biological controller, mainly because they have asymmetric extension for all connections, not only the connections going from the CCIN neurons as in the biological model (see [5]).

In summary, there are several possibilities for interconnecting the segmental network to obtain travelling waves with variable wavelengths for similar wavelength ranges. The interconnectivity of the biological model is only one of them.

4 Discussion

We have shown that a GA can be successfully used to develop artificial swimming controllers, in a relatively limited number of generations. GAs have thus proved to be an interesting design technique.

A first observation is that there exist many possible solutions other than just variations of the biological controller. Potential segmental oscillators vary in terms of weight values, connection types and even number of active neurons. This results in different kind of behaviours which have in common that the motoneurons present regular oscillations with asymmetric behaviour, but show differences in the activity of the other neurons (the interneurons), with different sequences of activity and signal shapes. The EIN, CCIN and LIN neurons have thus taken other functions than in the biological model where they were respectively excitatory interneurons, contralateral inhibitory interneurons and lateral inhibitory interneurons. None of the evolved solutions corresponds simply to the biological configuration with a swap of function between the neuron types. The only common structure which has been found in several solutions is the 4-neuron oscillator mentioned above (Figure 3). Preliminary results have also shown that there exists a variety of ways in which segmental oscillators can be interconnected in order to form complete controllers. The variety of different potential solutions was also observed for the leech bending reflex by Lockery who found that many different networks, with different sets of connections, could produce a physiologically accurate local bending input-output function [9].

The evolved controllers can be considered as more efficient in terms of frequency range than the biological model, as solutions have been found with frequency ranges several times larger than the frequency range of the biological model, and with the same range of wavelengths for the undulation. We have concentrated on the ability to vary the frequency of the oscillations and on large frequency ranges, because the frequency determines the speed of swimming and a large range of frequency means a greater flexibility of the controller. However, there is probably an upper limit for the frequency at which muscles can contract and mechanical simulations should be made to determine which highest frequencies can actually be performed. The upper limit observed for swimming lampreys is approximately 10Hz[10].

The fitness function could be extended in order to include aspects we have chosen not to consider in this first approach, such as the relation between the excitation level and the amplitude of the motoneuron signals, the shape of these signals, the inclusion of sensory feedback. The next step should in fact evaluate the mechanical behaviour rather than the neural behaviour in order to reward a controller by directly rewarding the effectiveness of swimming.

Our principal interest is to define a methodology for developing locomotion controllers for autonomous agents. But can this research be useful for Neuroscience? The results presented here are probably of limited interest for neuroscientists and the main points we showed are that a variety of potential solutions exists and that the frequency range of the biological model can be optimized by changing the weights of the connections of that model, resulting in a better coverage of the observed physiologically frequency range. But we believe that GAs can prove to be very useful in helping neuroscientists to model a system, by including knowledge from physiological measurements as constraints on the encoding and the fitness functions, and using a GA to determine unknown variables. The experiment in which we fix the biological types of connection and evolve the connection weights is an example of such a methodology.

5 Conclusion

This paper has examined the swimming controller of lampreys and developed alternative artificial solutions using a real number Genetic Algorithm. Artificial controllers composed of neurons similar to those of Ekeberg's biological model [5] have been created in two stages. Many different neural configurations for potential controllers have been shown to exist; the biological network is only one of them. Artificial controllers have been created which are more efficient, in terms of frequency range, than the biological model, with frequency ranges up to five times larger. We have also shown that the biological weights can be modified in order to increase the frequency range of the biological model and better match the physiological measurements. GAs have proved to be an interesting tool for developing adapted artificial controllers and for optimising the biological network.

[10] It is not clear if this limit is due to neural or mechanical limitations.

References

[1] R.D. Beer. *Intelligence as Adaptive Behavior, an Experiment in Computational Neuroethology.* Academic Press, 1990.

[2] R.D. Beer and J.C. Gallagher. Evolving dynamical neural networks for adaptive behavior. *Adaptive Behavior*, 1(1):91–122, 1991.

[3] 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.

[4] H. de Garis. Genetic programming: Building artificial nervous systems using genetically programmed neural network modules. In B.W. Porter and R.J. Mooney, editors, *Proceedings of the seventh international conference on machine learning*, pages 132–139. Morgan Kaufmann, 1990.

[5] O. Ekeberg. A combined neuronal and mechanical model of fish swimming. *Biol. Cybern.*, 69:363–374, 1993.

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

[7] 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.

[8] 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.

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

[10] P. Wallen, O. 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*, 6:1939–1950, December 1992.