Evolution of efficient swimming controllers for a simulated lamprey

Jimmy Hajime Or, John Hallam, David Willshaw[*] **Auke Ijspeert**[**]

[*]Divison of Informatics
University of Edinburgh
5 Forrest Hill, Edinburgh
Scotland EH1 2QL
{hajimeo, john}@dai.ed.ac.uk
david@anc.ed.ac.uk

[**]Computational Learning and Motor Control Lab
University of Southern California
Hedco Neuroscience Building
3641 Watt Way, Los Angeles
CA 90089-2520, USA
ijspeert@usc.edu

Abstract

This paper investigates the evolutionary design of efficient connectionist swimming controllers for a simulated lamprey. Efficiency is defined as the ratio of forward swimming speed to backward mechanical wave speed.

Using the lamprey model proposed by Ekeberg (1993) and extending the work of Ijspeert et al. (1999) on evolving lamprey swimming central pattern generators (CPGs) through genetic algorithms (GAs), we investigate the space of possible neural configurations which satisfies the property of high swimming efficiency. Techniques are devised to measure efficiency at various swimming speeds. The measurements are incorporated into the fitness function of Ijspeert's original GA and efficient controllers are evolved. Interestingly, the best evolved controller not only is capable of swimming in a similar manner to the real lamprey, but also with the same efficiency (about 0.8). Moreover, it can exhibit a wide range of controllable speeds and efficiencies.

1. Introduction

In recent years, there have been advances in understanding animal motor control due to better physiological measurement techniques, higher density microelectrodes and faster computers for simulations of the neural mechanisms which underlie behaviors. However, due to the complexity of the nervous systems, we are still far from being able to understand completely the neural control of higher vertebrates such as humans. The lamprey has been chosen for study by several neurobiologists because it is relatively easy to analyze: firstly, because while it has a brainstem and spinal cord with all the basic vertebrate features, the number of neurons in each category is an order of magnitude fewer than in other vertebrates, and secondly because its swimming gait is simple. Hence, findings on this *prototype vertebrate* can provide a better understanding of vertebrate motor control.

According to Sir James Lighthill, swimming speed and efficiency are the two qualities that fish must maintain in order to survive (Lighthill, 1970). If the swimming efficiency is low, the fish can quickly use up energy derived from food before they can find their next meal. The ability to maintain high swimming efficiency is especially important for lampreys because they do not eat during the long journey up-river from the sea to the breeding grounds (Williams, 1986). Blake (1983) suggested that efficiency is a good criteria to use when comparing the swimming performance of different fish. Its increase with speed is important to the evolutionary ecology of fish. Swimming efficiently is also important from a robotics point of view. An inefficient robotic lamprey can use up its battery power and sink in the ocean easily. Note that in the robotic implementation, it is important to maintain efficiency across a wide range of speeds.

Currently, there are several definitions of swimming efficiency (Sfakiotakis et al., 1999). However, the ratio of forward swimming speed (U) to backward mechanical wave speed (V) has been commonly used (Williams, 1986; Sfakiotakis et al., 1999). Since biological data is available for comparison, we are using the same definition to evolve controllers in this paper.

Over the past 15 years, neurobiologists have achieved a better understanding of the lamprey locomotive networks. However, nobody yet fully understands how the segmental oscillators inside the lamprey CPG are coupled. We believe it is a good idea to consider a few important properties related to the survival of the lamprey (such as swimming efficiency, robustness in speed against changes in body scales and noise in neural connections, etc.) and then use them as a guide towards the discovery of features in its neural organization that are related to such properties. Given that high swimming efficiency is important to both the real and the artificial lamprey, in this paper we extend the idea from Ijspeert's (1998) work on evolutions of lamprey swimming CPGs to evolve efficient swimming controllers for the model lamprey.

Experimental results are encouraging. Most of the evolved controllers are able to swim like the real fish and with high efficiency. Their speed vs. efficiency curves show that they can not only achieve a wide range of speeds but also be able to maintain a fairly constant efficiency (at least for speeds over 0.3 [m/s]).

Most importantly, the best evolved controller has achieved an efficiency of about 0.8, which is close to the one achieved by the real lamprey (Williams, 1986). Thus, through the use of GA, we have found intersegmental couplings which allow the model lamprey to swim at about the same efficiency as the real one. This result not only could provide inspiration to biologists to gain a better understanding of the intersegmental couplings of the real lamprey but also could inspire the development of more efficient swimming controllers for lamprey in both computer simulations and robotic hardware.

2. Background

This section briefly describes the Ekeberg neural and mechanical models. A more detailed description can be found in (Ekeberg, 1993; Or, 2002).

2.1 Neural model

Based on physiological experiments, Ekeberg (1993) hand-crafted a connectionist model for the lamprey swimming CPG. The network is made of 100 copies of interconnected segmental oscillators (Figure 1). Within each segmental oscillator, there are 8 neurons each of which is modeled using a leaky integrator with a saturating transfer function. The output u ($\in [0,1]$) is the mean firing frequency of the population the unit neuron represents. It is calculated using the following set of formulas:

$$\dot{\xi_+} = \frac{1}{\tau_D}\left(\sum_{i\in\Psi_+} u_i w_i - \xi_+\right) \quad (1)$$

$$\dot{\xi_-} = \frac{1}{\tau_D}\left(\sum_{i\in\Psi_-} u_i w_i - \xi_-\right) \quad (2)$$

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

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

where w_i represents the synaptic weights and 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 inputs and ϑ represents the frequency adaptation observed in real neurons. The values of the neural timing (τ_D, τ_A) and gain/threshold (Θ, Γ, μ) parameters and those for the connection weights are set up in such a way as that the simulation results from the model agree with physiological observations. For details of the neuron parameters, refer to (Ekeberg, 1993).

As the details of the intersegmental connections of the real lamprey CPG are not yet known, Ekeberg simplified the controller as follows. Except for the CIN neurons which have longer projections in the caudal direction, each neuron has symmetrical connections extending both rostrally and caudally. Since the neurons at both ends of the CPG receive fewer neural connections, synaptic weights are adjusted to account for this by dividing them by the number of segments a neuron receives input from.

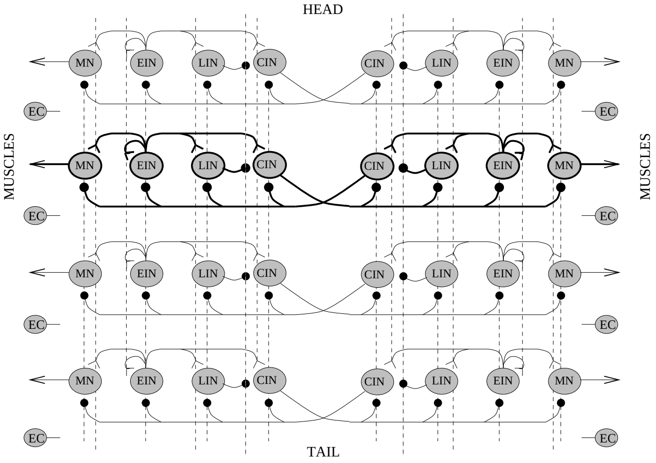

Figure 1: Configuration of the biological swimming controller. The controller is composed of 100 interconnected segmental oscillators (only four segments are shown here). Each segment consists of eight neurons of four types: motoneurons (MN), excitatory interneurons (EIN), lateral inhibitory interneurons (LIN) and contralateral inhibitory interneurons (CIN). A neuron unit represents a population of functionally similar neurons in the real lamprey. Each of them receives excitations from the lamprey brainstem. Connections with a fork ending represent excitatory connections while those with a dot ending represent inhibitory connections. In addition to input signals from the brainstem (not shown here), the controller receives feedback from the stretch sensitive edge cells (EC). Note that the EC cells are not considered in this paper.

The complete swimming CPG functions as follows: Global excitation from the brainstem stimulates all neurons in the CPG; sufficient stimulation results in oscillations in each individual segment at a frequency that depends on the strength of this global excitation signal. Extra excitation is supplied from the brainstem to the five most rostral segments of the CPG. The effect of this, interacting with intersegmental coupling, is to induce a roughly equal relative phase lag between successive segments in the CPG, with the result that caudally traveling waves of neural activity appear. The global excitation controls the amplitude of the motoneurons as well as the frequency of oscillation of the CPG. The extra excitation alters the intersegmental phase lag largely independently from the global excitation.

2.2 Mechanical model

Ekeberg (1993) further proposed a 2D mechanical lamprey model to study how the muscular activity induced by the model CPG affects swimming. The model lamprey is made to approximate the size and shape of the real one. It consists of 10 rigid body links with nine joints of one degree of freedom (Figure 2). Each link is assumed to be a cylinder with an elliptical cross-section. The link is represented by its center of mass coordinate (x_i, y_i) as well as the angle (φ) between it and the x-axis.

On each side of the body, muscles connect each link to its immediate neighbors. The muscles are modeled as a combination of springs and dampers. The outputs from the motoneurons control the spring constants of the corresponding muscles. As the neural wave travels along the body from head to tail, the successive contraction of muscles creates a mechanical wave. This in turn generates inertial forces from the surrounding water that propel the lamprey forward.

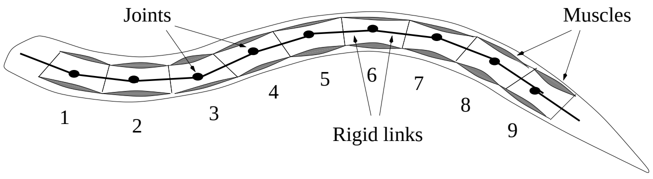

Figure 2: The mechanical lamprey model.

Each mechanical link has an elliptical cross section of constant height (30 [mm]) and variable width. Its length is 30 [mm]. The mass and moment of inertia of each link are calculated by assuming that the density of the lamprey is the same as that of water (Table 1).

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 1: Parameters for the mechanical simulation. w_i, m_i and I_i are the width, mass and inertia of link i respectively. $\lambda_\perp$ and $\lambda_\parallel$ are the λ factors used to calculate the water forces.

3. Methods

3.1 Genetic Algorithms

To evolve intersegmental couplings of 100 copies of any chosen segmental oscillator, the same real number GA with mutation and crossover operators described in Ijspeert et al. (1999) is used here. Two sets of experiments are conducted. The first set is based on evolving controllers with big maximum achievable efficiency while the second set is on evolving controllers with big minimum efficiency (hereby referred as the bigmax and bigmin approach respectively). The reason for conducting two sets of experiments is to investigate which approach produces better results (i.e. more controllers swimming at higher efficiency). In each set of experiments, the following four prototype controllers are used to seed the otherwise random initial populations of each run.

1. *The Biological Controller*, hand-crafted by Ekeberg (1993) based on physiological data.
2. *Controller 2*, evolved by Ijspeert et al. (1999) using Ekeberg's segmental oscillator and intersegmental couplings evolved by GA.
3. *Controller 3*, evolved by (Ijspeert et al., 1999), with both intra- and intersegmental connections evolved by GA.
4. *Hybrid Robust Controller*, which consists of the hybrid segmental oscillator[1] with the best evolved intersegmental couplings, for that oscillator, from (Or, 2002).

To test whether seeding the initial population, with its potential reduction of diversity, was necessary, we also ran:

5. *Hybrid Random*, in which the initial population was made up of individuals, using the Hybrid Segmental Oscillator, but with random intersegmental couplings, i.e. as in case 4 above but without the pre-evolved seed controller.

For each prototype controller, six evolutions (runs) are performed. This makes a total of 30 runs for each set of experiments. Note that the initial population of each run (for cases 1 .. 4) contains a prototype controller together with 39 other randomly generated individuals while for case 5 all 40 are randomly generated.

[1] In Or (2002), we included two segmental oscillators (one which can exhibit a large range of amplitude and the other which can oscillate at a large range of frequency) along with other randomly generated individuals in the initial population to evolve the hybrid of these two controllers, which we call the Hybrid Segmental Oscillator. Its motoneuron outputs have a larger minimum amplitude and a larger range of oscillation frequency than the motoneuron outputs from the rest of the evolved oscillators. The best complete controller obtained by evolving intersegmental couplings for this segmental oscillator is called the Hybrid Robust Controller.

3.2 Encoding

A chromosome is used to encode the intersegmental couplings among any given segmental oscillator and its neighbors. The couplings are extensions from a neuron in one segment to other post-synaptic neurons in the neighboring segments. For each neuron within a segment, the number of extensions in either the rostral or caudal direction is an integer value between 0 and 12. The reason for choosing this range is that it includes the maximum number of extensions (10) in the biological model. The extension is represented by a real-valued gene with range [0, 1]. The gene value is linearly mapped to [0, 12] and rounded to give the extension. Due to left-right symmetry, a chromosome of length 64 can be used to encode the intersegmental connections of the entire CPG.

3.3 Fitness calculation

We defined the fitness function which evolves efficient swimming controllers as a product of five factors so that they can be optimized at the same time. Each fitness factor varies linearly between 0.05 and 1.0 when the corresponding variable varies between the "bad" and "good" boundary. The five factors reward solutions with the abilities to:

1. generate stable and regular oscillations in the 100 segments of the CPG (i.e. *min_fit_oscil* > 0.45),
2. change the wavelength of undulation by changing the global excitation,
3. change the frequency of oscillation by changing the extra excitation
4. modulate the swimming speed by varying either the frequency of oscillation or the wavelength,
5. swim with high efficiency.

Note that the first two factors are exactly as the ones defined in (Ijspeert, 1998). The third and fourth factors are slightly modified to adapt for the experiments described in this paper.

An evaluation consists of neuromechanical simulations with different control inputs in order to determine how different key characteristics (such as frequency, speed and efficiency, *etc.*) vary. Note that the difference between experiments one and two is that, in the latter case, the fitness function is modified to search for the minimum non-zero efficiency value instead of the maximum efficiency.

3.4 Efficiency calculation

The efficiency and mechanical wave speed are defined as follows:

$$e = \frac{U}{V} \tag{5}$$

$$V = \frac{\lambda}{T} \tag{6}$$

where U, V, λ and T are the swimming speed [m/s], backward mechanical wave speed [m/s], mechanical wavelength [m] and the mechanical period [s] respectively. The forward swimming speed is calculated by fitting a circular arc to the positions and directions of the model lamprey head at two time instants which have the same relative phase in the swimming cycle.

Since both the neural and mechanical simulations are not stable at the beginning, the lamprey can end up swimming straight at any angle. The first step in computing the mechanical wave parameters is to rotate the original coordinate system so that the fish is swimming parallel to the x-axis. Following the transformation, we calculated the mechanical periods of body links 2 to 5. (The mechanical period of each body link can be calculated by using the difference in time instances at which two consecutive wave crests pass through the same link.) If the periods of three or more of the mechanical links are defined, the mechanical wavelength can be calculated using the method described by Videler (1993). Note that due to strange head and tail movements caused by the reduction in neural connections at the ends of the swimming CPG, the mechanical period of the second link, T_2, is used in Equation 6 to calculate the mechanical wave speed. Equation 5 can then be used to compute the swimming efficiency.

4. Results

We monitored the progress of the evolution weekly. Some of the most fit individuals from these evolutions were tested. When the fitness of these individuals stopped increasing significantly, we stopped the evolutions (after 2 months). The best individual of each run is tested over a range of global and extra excitations (under neuromechanical simulations) to determine the ranges of amplitude, frequency, phase lag, speed and efficiency which it can achieve. The corresponding surfaces are plotted for comparison.

Due to space limitations, the results for only 20 of the 60 evolved controllers are presented here. The criteria for choosing these controllers is a balance between high swimming efficiency and high fitness (recall that the fitness function has more factors than efficiency). Based on these criteria, two controllers from each prototype group are chosen for comparison. Since there are five prototypes and two sets of experiments, this makes a total of 20 controllers.

The results for the two set of experiments are presented in the following subsections. For details of the neural configuration and performance surfaces of each controller, refer to (Or, 2002).

4.1 Results of experiment 1: On evolving controllers with big maximum efficiency

The results for the 10 selected controllers based on the bigmax approach are summarized in Table 2.

The table indicates that except for the run3 controller and those evolved with the hybrid segmental oscillator as the prototype (the bottom four controllers), the rest of the evolved controllers can achieve a higher maximum efficiency than the corresponding prototypes. Among the 10 evolved controllers, the run5 controller has the highest efficiency value of 0.86.

4.2 Results of experiment 2: On evolving controllers with big minimum efficiency

The results for the 10 selected controllers based on the bigmin approach are summarized in Table 3. The table indicates that except for run7, run9 and run10 controllers (again all evolved with the hybrid segmental oscillator as the prototype), the rest of the evolved controllers are more efficient than their corresponding prototypes. Among the 10 evolved controllers, the run8 controller has the highest efficiency value of 1.03[2]!

5. Discussion of the methods

Rather than starting with random initial populations, we included a prototype controller in each initial population to guide the GAs to search for regions of possible solutions (controllers that can at least swim) in the search space. Although this approach can reduce the amount of time needed to evolve efficient controllers, there is the possibility that all the evolved controllers (under the same prototype) end up similar to each other. Fortunately, this did not pose a serious problem here. Most of the evolved controllers (even evolved with the same prototype) have different neural configurations and performance surfaces. Thus, adding the prototypes helped to accelerate the generation of interesting swimming controllers without significantly biasing the diversity of the evolved controllers. The biological controller, controller 2 and controller 3 prototypes all have fitness of zero under the new fitness function. As a result, they could not dominate the entire population. As for the hybrid robust prototype controller, it has a fitness of 0.11 and 0.06 in experiments one and two respectively, which is relatively low. The mutation and crossover operators of GAs could easily move the search to neighboring regions. Evolutions based on the hybrid random prototype were included just in case this approach failed. In general, if there were plenty of time and computing resources, it might be better to have an initial population with all randomly made individuals. This allows more different types of controllers to be evolved.

In experiment one, we evolved controllers based on big maximum efficiency. The reason for this is that we wanted to obtain controllers which are capable of swimming at high efficiency. Since we only considered positive efficiency to be valid, evolving controllers under this approach implicitly means evolving controllers with a larger efficiency range.

In experiment two, we evolved controllers based on big minimum efficiency. This approach implicitly forces all the measured efficiencies of the controller to be good because the GA is trying to pull up the worst efficiency each controller can achieve. Hence, it can be harder for the evolution system. However, the evolved controllers in this experiment should produce better results than those in experiment 1, and a comparison of the efficiency range achieved by the controllers in Table 2 and Table 3 shows that this is indeed correct. (Also, refer to Table 4 in the discussion which follows.)

To determine pulse regularity, the condition *min_fit_oscil* > 0.45 is used. The threshold value of 0.45 was derived in Ijspeert et al. (1999) based on experience. Generally speaking, this value is good enough to distinguish neural waves which oscillate regularly from those which do not. It seemed to be suitable for the implementation here at the beginning. However, at the end of the evolutions, we realized that the GA had found a way to break this condition to pull up the efficiency (see below). Fortunately, the threshold problem appears only in two of the 60 evolved controllers: run5 (experiment 1) and run8 (experiment 2).

Finally, the methods used to calculate the mechanical wavelength and efficiency have several limitations. According to Videler (1993), the measurement of kinematic parameters such as mechanical wavelength can be achieved accurately only as long as the mechanical wave crests propagating along the body are well pronounced and the amplitude is large even near the head. This should not pose a problem because these characteristics fit eel-like swimmers such as the lamprey. However, the two controllers with efficiency over 0.8 sometimes swim with a stiff body (due to pulse irregularity) in approximately the first half of the body. This is similar to the sub-carangiform swimming mode described in (Sfakiotakis et al., 1999). It looks like the main problem here is that the wavelength is not constant along the body (i.e. infinite wavelength in the rigid part of the body),

[2] Efficiency greater than one is impossible. This value is caused by the breakdown of the wavelength calculation algorithm. Thus, this value is later considered to be invalid (refer to Section 5. for details).

	Fitness	Amplitude range	Frequency range in [Hz]	Phase lag range in [%]	Speed range in [m/s]	Efficiency range
biological	0.00	[0.0, 0.8]	[1.6, 5.5]	[-0.1, 1.7]	[-0.09, 0.45]	[0.05, 0.58]
run1	0.11	[0.2, 0.8]	[1.6, 7.2]	[-2.9, 2.6]	[-0.09, 0.51]	[0.05, 0.61]
run2	0.10	[0.0, 0.8]	[1.3, 5.7]	[-1.4, 3.2]	[-0.03, 0.53]	[0.05, 0.64]
controller 2	0.00	[0.0, 0.8]	[1.7, 6.0]	[-3.1, 3.2]	[-0.09, 0.49]	[0.02, 0.60]
run3	0.13	[0.0, 0.8]	[1.4, 7.5]	[-2.6, 2.8]	[-0.09, 0.52]	[0.05, 0.59]
run4	0.11	[0.0, 0.8]	[1.6, 5.7]	[-2.2, 3.4]	[-0.03, 0.50]	[0.01, 0.63]
controller 3	0.00	[0.0, 0.6]	[1.3, 5.5]	[-0.2, 1.9]	[-0.08, 0.43]	[0.06, 0.58]
run5	0.10	[0.0, 0.6]	[1.4, 6.4]	[-2.3, 8.6]	[-0.07, 0.49]	[0.02, 0.86]
run6	0.06	[0.0, 0.6]	[1.5, 5.9]	[-0.0, 1.8]	[-0.08, 0.44]	[0.03, 0.64]
hybrid robust	0.11	[0.0, 0.8]	[1.8, 7.1]	[0.0, 3.1]	[-0.02, 0.49]	[0.08, 0.69]
run7	0.15	[0.0, 0.8]	[1.4, 7.1]	[-0.0, 2.8]	[-0.03, 0.48]	[0.18, 0.68]
run8	0.11	[0.0, 0.8]	[1.2, 7.1]	[0.0, 3.3]	[-0.02, 0.48]	[0.30, 0.61]
hybrid random						
run9	0.05	[0.0, 0.8]	[1.5, 7.6]	[-1.3, 2.9]	[-0.05, 0.48]	[0.07, 0.65]
run10	0.09	[0.0, 0.7]	[1.5, 7.0]	[-0.4, 2.2]	[-0.02, 0.38]	[0.07, 0.62]

Table 2: Summary of results for the evolved efficient controllers in experiment one. The table lists the performance of the best individual from each evolution. The evolution is based on the bigmax approach. Note that the hybrid random prototype generates irregular neural waves due to random couplings. As a result, all the parameters are undefined.

	Fitness	Amplitude range	Frequency range in [Hz]	Phase lag range in [%]	Speed range in [m/s]	Efficiency range
biological	0.00	[0.0, 0.8]	[1.6, 5.5]	[-0.1, 1.7]	[-0.09, 0.45]	[0.05, 0.58]
run1	0.11	[0.0, 0.8]	[1.6, 5.6]	[0.0, 6.4]	[-0.05, 0.48]	[0.05, 0.68]
run2	0.06	[0.2, 0.8]	[1.6, 5.5]	[-0.7, 1.8]	[-0.16, 0.47]	[0.05, 0.68]
controller 2	0.00	[0.0, 0.8]	[1.7, 6.0]	[-3.1, 3.2]	[-0.09, 0.49]	[0.02, 0.60]
run3	0.09	[0.0, 0.8]	[1.6, 5.5]	[-0.3, 2.1]	[-0.15, 0.51]	[0.06, 0.76]
run4	0.08	[0.0, 0.8]	[1.6, 5.5]	[-1.7, 3.1]	[-0.03, 0.48]	[0.03, 0.70]
controller 3	0.00	[0.0, 0.6]	[1.3, 5.5]	[-0.2, 1.9]	[-0.08, 0.43]	[0.06, 0.58]
run5	0.06	[0.1, 0.6]	[1.3, 7.9]	[-3.1, 3.3]	[-0.05, 0.48]	[0.03, 0.59]
run6	0.06	[0.0, 0.6]	[1.3, 6.9]	[0.0, 2.4]	[-0.08, 0.48]	[0.03, 0.58]
hybrid robust	0.06	[0.0, 0.8]	[1.8, 7.1]	[0.0, 3.1]	[-0.02, 0.49]	[0.08, 0.69]
run7	0.09	[0.0, 0.8]	[1.2, 7.0]	[-0.1, 3.5]	[-0.02, 0.49]	[0.02, 0.61]
run8	0.09	[0.0, 0.8]	[1.5, 7.1]	[0.0, 3.6]	[-0.05, 0.46]	[0.12, 1.03]
hybrid random						
run9	0.10	[0.0, 0.8]	[1.3, 7.0]	[-1.1, 5.3]	[-0.07, 0.46]	[0.23, 0.68]
run10	0.20	[0.0, 0.7]	[2.0, 7.1]	[0.0, 3.1]	[-0.02, 0.48]	[0.05, 0.63]

Table 3: Summary of results for the evolved efficient controllers in experiment two. The table lists the performance of the best individual from each evolution. The evolution is based on the bigmin approach. Note that the hybrid random prototype generates irregular neural waves due to random couplings. As a result, all the parameters are undefined.

as it should be during anguiliform swimming. Under this situation, the measurement algorithm breaks down, underestimating the mechanical wavelength, and efficiency over 1 was obtained.

6. Discussion of results

Using the fitness function presented in this paper, efficient swimming controllers have been evolved successfully. Most of them are more efficient than their corresponding prototypes. The neural configurations of the best individuals from the 20 evolutions are different even with the presence of the same prototype in the initial population. There is not much similarity in the way the segments are coupled.

Generally speaking, controllers based on the bigmin approach can achieve higher efficiency than those based on the bigmax approach. (This is true even when all 60 evolutions are taken into consideration.) Table 4 shows that all the evolved controllers have a maximum efficiency ≥ 0.58. Under the bigmax approach, three of the controllers have efficiency ≥ 0.65, the best of which has efficiency above 0.7. When the bigmin approach is used, six of the evolved controllers have efficiency ≥ 0.65, three of which have efficiency above 0.7. As the evolutions under the bigmin approach can produce more good solutions at the same time as those under the bigmax approach (they all started and terminated at the same time), it is better to evolve efficient controllers based on the bigmin approach.

Approach	$e \geq 0.58$	$e \geq 0.60$	$e \geq 0.65$	$e \geq 0.7$
Bigmax	10	9	3	1
Bigmin	10	8	6	3

Table 4: Comparison of performance of the bigmax and bigmin approaches in terms of the efficiencies of the controllers. Since there is a cheating controller in each experiment (see below), subtract one from each table element if they are considered to be invalid due to the breakdown of the mechanical wavelength calculation algorithm.

6.1 *Discussion of the evolved controllers*

In order to understand how the evolved controllers achieve high swimming efficiency, we have chosen the best five[3] for further investigation. We looked at the characteristics of their neural waves as well as the corresponding swimming patterns. Based on these investigations, the controllers can be classified into two groups. The first group includes the run5 and run8 controllers (from bigmax and bigmin respectively) while the second group involves the run2, run3 and run9 controllers (all from bigmin). Controllers from the former group are called the "*cheating controllers*" as some of their neural waves contain irregular oscillations.

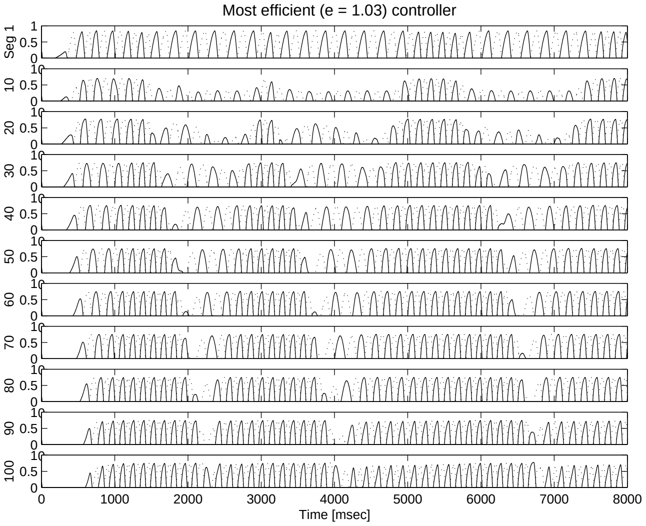

Figure 3: Irregular neural waves generated by the bigmin run8 controller. Solid lines represent outputs from the left motoneurons while dashed lines represent the outputs from the right motoneurons.

Figure 3 gives an example of the motoneuron output from a controller (bigmin run8) of this group. Its *min_fit_oscil* value is 0.452 (0.485 for the bigmax run5 controller). The irregular neural waves cause the mechanical wave calculation algorithm to break down and return very short mechanical wavelengths (less than 0.1 m). This results in very high efficiency values. From computer animations, we have found that a lamprey embedded with either of these two controllers swims alternately between sub-carangiform and anguilliform swimming modes. As for the three controllers in the second group, the outputs from the motoneurons are regular. Figure 4 shows the neural wave of a typical controller (run3) from this group.

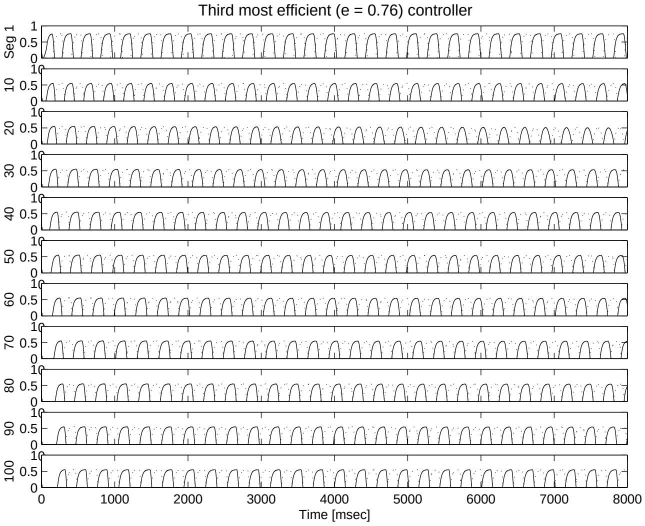

Figure 4: Neural wave of the bigmin run3 controller. Solid lines represent outputs from the left motoneurons while dashed lines represent the outputs from the right motoneurons

The average amplitude and oscillation frequency of the three controllers from the second group are about 0.57 and 3.6 [Hz] respectively while the average swimming speed is about 0.34 [m/s] (refer to Table 5). These values are relatively low when compared with those of controllers evolved without taking efficiency into consideration, as reported in (Ijspeert, 1998)[4].

As we require a controller which can both generate regular control signals and swim with high efficiency, the run3 controller evolved under the bigmin approach is considered to be the most efficient one. The connection weight matrix for this controller is shown in Table 6 while its performance surfaces are shown in Figure 5. Note that the maximum efficiency achieved by this controller is 0.76, which is close to 0.8 achieved by the real lamprey (Williams, 1986).

Although the efficiency is very high, the corresponding maximum swimming speed is about 0.32 [m/s], which is lower than typical maximum speeds of about 0.4 [m/s].

[3] Based on a balance between high efficiency and smooth performance surfaces.

[4] The average frequency and maximum speed for the evolved controllers reported in (Ijspeert, 1998; Ijspeert et al., 1999) are 8.2 [Hz] and 0.54 [m/s] respectively.

Rank	(global, extra)	Amp	Freq [Hz]	Phase lag [%]	Mec λ [m]	Speed [m/s]	Wave speed [m/s]	Efficiency
1	(1.0, 40%)	0.76	5.56	1.22	0.04	0.34	0.33	1.03
2	(0.6, 140%)	0.45	3.60	1.69	0.02	0.27	0.31	0.86
3	(0.4, 100%)	0.54	3.52	1.60	0.12	**0.32**	0.42	**0.76**
4	(0.5, 150%)	0.60	4.03	1.29	0.13	0.37	0.55	0.68
5	(0.6, 30%)	0.58	3.23	1.61	0.15	0.34	0.49	0.68
Sin		0.54	3.52	1.60	0.16	**0.34**	0.58	**0.59**

Table 5: Comparison of neural and mechanical parameters for the five controllers with largest maximum efficiency. The controllers are run8, run5, run3, run2 and run9 (listed in order from top to bottom). Except for the run5 controller, these are evolved based on the bigmin approach. The last line shows the performance of a sinusoidal controller with amplitude, frequency and phase lag matching the rank 3 controller. See text for discussion.

	MNl	EINl	LINl	CINl	CINr	LINr	EINr	MNr	BS
MNl	-	1.0 [4, 9]	-	-	-2.0 [4, 4]	-			5.0
EINl	-	0.4 [3, 3]	-	-	-2.0 [3, 5]	-	-	-	2.0
LINl	-	13.0 [4, 2]	-	-	-1.0 [11, 5]	-	-		5.0
CINl	-	3.0 [11,1]	-1.0 [4, 7]	-	-2.0 [5, 7]	-	-	-	7.0
CINr	-		-	-2.0 [5, 7]	-	-1.0 [4, 7]	3.0 [11, 1]	-	7.0
LINr	-	-	-	-1.0 [11, 5]	-	-	13.0 [4, 2]	-	5.0
EINr	-	-	-	-2.0 [5, 5]	-	-	0.4 [3, 3]	-	2.0
MNr	-	-	-	-2.0 [4, 4]	-	-	1.0 [4, 9]	-	5.0

Table 6: Connection weight matrix for the efficient swimming (bigmin run3) controller. 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 brainstem. The extensions from a neuron to those in neighboring segments are given in brackets. The first number indicates the number of extensions in the rostral direction while the second number indicates extensions in the caudal direction.

This means efficient energy utilization at the cost of speed, as observed in the real lamprey (Williams, 1986).

6.2 Comparison of the best evolved controller with a matched sinusoidal controller

A comparison of the best evolved controller with a matched sinusoidal controller can provide justification of using a neural controller evolved by GA, rather than a simpler analytic controller, to control an artificial lamprey. Table 5 lists the neural parameters and swimming performance of the five controllers with highest efficiency and one matched sinusoidal controller. By matching, we mean a sinusoidal controller whose amplitude, frequency and phase lag are tuned to match the corresponding parameters for the controller under consideration (in this case, the rank 3 one). The table shows that the relatively high efficiency achieved by the first two controllers is caused by the breakdown of the wavelength calculation algorithm. (Based on experience, wavelength much less than 0.1 m indicates that the motoneuron outputs are irregular. This in turns corresponds to the breakdown of the wavelength calculation algorithm which often return a high efficiency value.) As the efficiency of these two controllers is invalid, the sinusoidal controller is compared with the run3 controller which is listed third (but considered to be the best evolved controller) in the table.

The comparison shows that although the analytic sinusoidal controller is able to achieve a slightly higher speed (0.34 [m/s] vs. 0.32 [m/s]), its efficiency is much lower than the run3 controller (0.59 compared with 0.76). Figure 6 shows the joint drive generated by the run3 controller with superimposed joint drive generated by the matched sinusoidal controller.

While the sinusoidal controller and the neural controller are matched in terms of amplitude, frequency and phase lag, they are not matched in terms of pulse shape. The pulse duration of the run3 controller is longer than that of the sinusoidal controller. Since the swimming speed is inversely proportional to burst duration (Grillner and Kashin, 1976; Wallén and Willams, 1984) (later demonstrated for the Ekeberg mechanical model — unpublished results), this agrees with the result that the swimming speed of the run3 controller is lower than that of the matched sinusoidal controller. Given that the sinusoidal controller can achieve a higher swimming speed, its lower efficiency is caused by the larger wavelength of the induced mechanical wave (0.16 [m] vs. 0.12 [m]).

The reason why a lamprey driven by the run3 controller can have a shorter wavelength is due to the pulse shape. A comparison of the signals that control the first joint of the mechanical lamprey body shows that symmetrical pulses are generated by the sinusoidal controller. On the other hand, the pulses generated by the run3 controller are not symmetrical (Figure 6). Recall that the mechanical wavelength calculation algorithm requires the time instances at which a mechanical wave crest passes through different parts of the body. As the pulse shape can affect the amount of bending at each body link, the mechanical wavelengths of the two con-

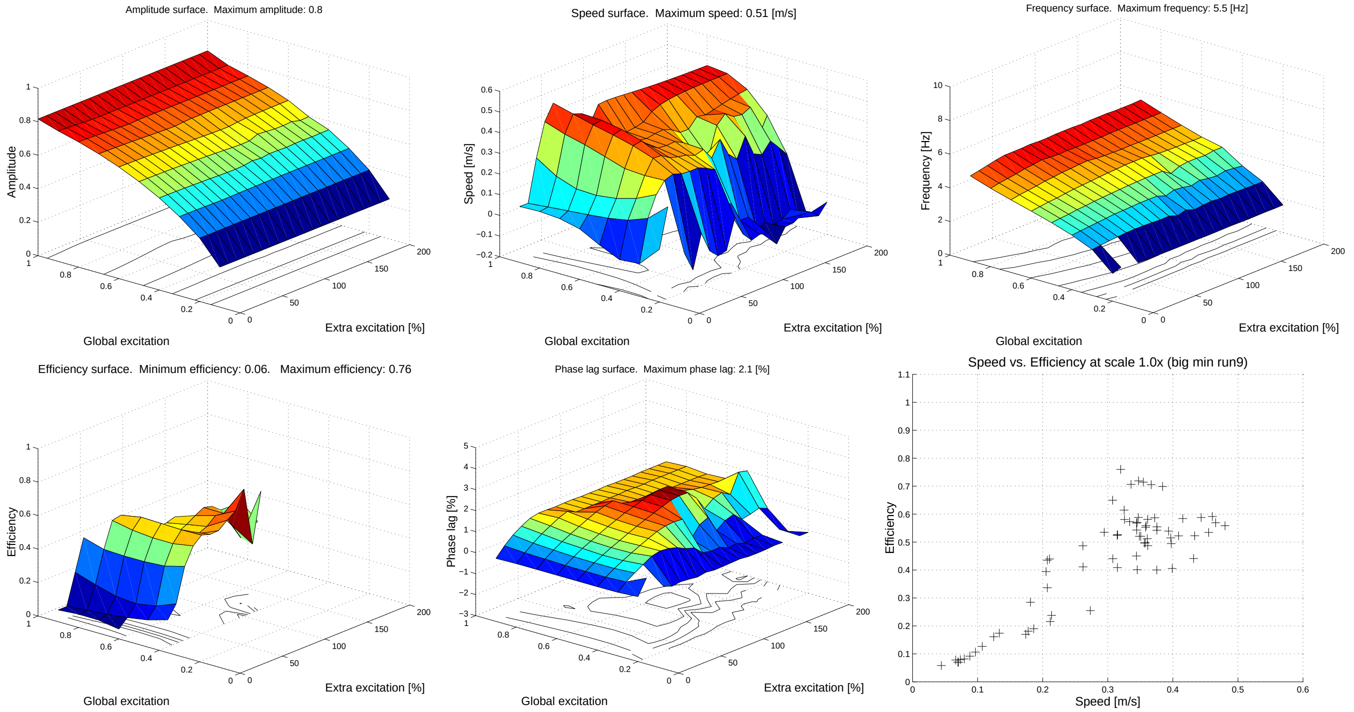

Figure 5: Performance surfaces of run3 controller. Note that efficiencies that cannot be measured (i.e. either $min\_fit\_oscil \leq 0.45$ or efficiency ≤ 0) are filtered. These filtered values correspond to the empty regions in the efficiency surface.

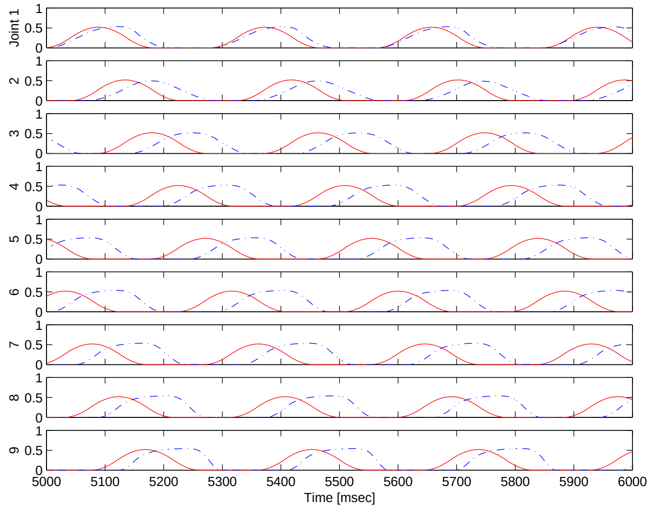

Figure 6: Averaged MNl activity (joint drive) from run3 controller with superimposed averaged MNl activity from a matched sinusoidal controller. The excitation combination used is (0.4, 100%). The amplitude is 0.54 while the frequency is 3.52 Hz. Phase lag is 1.60%. Dashed lines represent outputs from the run3 controller. Solid lines represent outputs from the matched sinusoidal controller

trollers can be different. It may be that the rather flat and asymmetrical pulse shape of the run3 controller allows it to achieve a shorter mechanical wavelength. Such controller is relatively difficult to hand-craft analytically. Note that another difference between the two controllers is that the wavelength in the neural controller varies slightly along the body, while it is perfectly constant in the sinusoidal controller.

6.3 Inherited property: Robustness in swimming efficiency against variations in speed

This section addresses the robustness in swimming efficiency against variation in speed. We want a robotic lamprey which can both achieve a wide range of swimming speed and be able to maintain high efficiency.

In order to determine the relationship between the swimming speed and efficiency of the evolved controllers, we have plotted the efficiency vs. speed curves of these controllers for speed ranges from 0.05 to 0.6 [m/s] in steps of 0.05 (refer to Figure 5 for the plot belonging to the bigmin run3 controller. For the plots of other controllers, refer to (Or, 2002).). Note that the scatter plots are used as the same speed can correspond to more than one efficiency value. For the different controllers, the results suggest that the relation between speed and efficiency can be classified into two types, namely:

- efficiency increases with speed

- efficiency increases with speed initially and then stays fairly constant (for speed over 0.3 [m/s])

Such relationships are favorable to both the real and robotic lampreys as mentioned in the introduction section.

7. Conclusion

In this paper, we successfully used GA to evolve efficient swimming controllers. Most of the evolved controllers exhibit a wide range of controllable speeds and efficiencies. Moreover, some of them are robust in swimming efficiency against speed. Under the same wave characteristics (amplitude, frequency and phase lag), the best evolved controller is able to drive the model lamprey more efficiently than the corresponding analytic sinusoidal controller. The difference appears to be due to subtle differences in the shapes of the signals between the sinusoidal controllers and the neural controllers. The evolved neural controllers can thus potentially be used to efficiently control a robotic lamprey such as the one developed by (McIsaac and Ostrowski, 1999). Based on experimental results, evolutions using the bigmin approach produce more controllers with higher swimming efficiency than those based on the bigmax approach. Most importantly, the GA has found a controller which can achieve the same efficiency as that observed in the real lamprey. Future work involves comparing the evolved neural organization with that of the real lamprey. The results will tell how close they are to each other.

8. Acknowledgments

Jimmy Or is supported by the British ORS Awards, the Canadian Natural Science and Engineering Research Council and the Canadian Space Agency. Facilities for this research were provided by the University of Edinburgh.

References

Blake, R. (1983). *Fish locomotion.* Cambridge University Press.

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

Grillner, S. and Kashin, S. (1976). On the generation and performance of swimming in fish. In Cohen, A. H., Rossignol, S., and Grillner, S., (Eds.), *Neural control of locomotion.* Plenum Press.

Ijspeert, A. J. (1998). *Design of artificial neural oscillatory circuits for the control of lamprey- and salamander-like locomotion using evolutionary algorithms.* PhD thesis, University of Edinburgh.

Ijspeert, A. J., Hallam, J., and Willshaw, D. (1999). Evolving swimming controllers for a simulated lamprey with inspiration from neurobiology. *Adaptive Behavior*, 7(2):151–172.

Lighthill, M. J. (1970). Aquatic animal propulsion of high hydrodynamic efficiency. *Journal of Fluid Mechanics*, 44:263–301.

McIsaac, K. A. and Ostrowski, J. P. (1999). A geometric approach to anguilliform locomotion: simulation and experiments with an underwater eel robot. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 1, pages 2843–2848. IEEE Press.

Or, J. (2002). *An investigation of artificially-evolved robust and efficient connectionist swimming controllers for a simulated lamprey.* PhD thesis, University of Edinburgh. Unpublished PhD thesis being examined.

Sfakiotakis, M., Lane, D., and Davis, J. (1999). Review of fish swimming modes for aquatic locomotions. *IEEE Journal of Oceanic Engineering*, 24(2):237–252.

Videler, J. J. (1993). *Fish swimming.* Chapman and Hall.

Wallén, P. and Willams, T. (1984). Fictive locomotion in the lamprey spinal cord in vitro compared with swimming in the intact and spinal animal. *Journal of Physiology*, 347:225–239.

Williams, T. L. (1986). Mechnical and neural patterns underlying swimming by lateral undulations: Review of studies on fish, amphibia and lamprey. In Grillner, S., Stein, P. S. G., Stuart, D., Forssberg, H., and Herman, R., (Eds.), *Neurobiology of vertebrate locomotion*, pages 141–155. Macmillan.