

# Robust simulation of lamprey tracking

Matthew Beauregard and Paul J. Kennedy

Faculty of IT, University of Technology, Sydney, PO Box 123, Broadway, NSW 2007,  
AUSTRALIA,  
`paulk@it.uts.edu.au`

**Abstract.** Biologically realistic computer simulation of vertebrates is a challenging problem with exciting applications in computer graphics and robotics. Once the mechanics of locomotion are available it is interesting to mediate this locomotion with higher level behavior such as target tracking. One recent approach simulates a relatively simple vertebrate, the lamprey, using recurrent neural networks to model the central pattern generator of the spine and a physical model for the body. Target tracking behavior has also been implemented for such a model. However, previous approaches suffer from deficiencies where particular orientations of the body to the target cause the central pattern generator to shutdown. This paper describes an approach to making target tracking more robust.

## 1 Introduction

Increasingly, digital media professionals are incorporating biologically realistic representations of artificial animals into films and computer games. Whilst representations of bodies, fur and skin have become increasingly realistic, it is challenging to model life-like movement and life-like behaviors such as tracking of targets. Vertebrate locomotion is a complex process that is difficult to imitate in simulated environments. Arms, legs, and spinal columns have many degrees of freedom that must be controlled in a co-ordinated way for stable locomotion to occur. However, life-like locomotion on its own is only the first step towards generating biologically realistic behavior. Other higher actions, beginning with target tracking and foraging, must be simulated on top of the locomotion to produce life-like behavior. Many of these types of behavior have been explored previously in the robotics and Artificial Life domains (eg. [1]), but usually with simple models of locomotion. The form of locomotion affects the higher level behavior. For example, swimming in a simulated fish involves movement of the head from side to side. Consequently, the inputs from simple models of eyes tracking an object in the environment change as the simulated animal's head moves. Target tracking must take this into account for robust behavior.

This paper examines target tracking behavior that has been implemented with specific inputs to a complex simulated fish spinal cord. Two previous approaches to tracking are described. However, these approaches suffer in that they do not sufficiently take into account head movements of the simulated animal and, consequently, are not robust in all situations. We describe a modification that makes the tracking more robust to movements of the target.

Digital characters in computer graphics are often composed of a skin (collection of surfaces) associated with a skeletal configuration or “armature” at various places. As the armature moves, the skin deforms with the resulting effect showing the character moving. Computation of movement of bones is often done with inverse kinematics [2]. Sequences of poses are compiled by an animator and replayed (with interpolation of the in between movement) to show the character performing actions. This gives precise control, allowing movements and deformations that are not physically possible in the real world. However, if the aim is life-like movement, these techniques may be tedious to apply.

A complementary approach is to model the anatomy of the animal in greater detail. The challenge is that detailed actual physiology of animals, at the neural level, is mostly unknown and very complex. One successful approach has been to start with simpler animals, for which knowledge of the neural pathways is available. The animal is modelled as a body and a spine consisting of connected oscillating clusters of recurrent neural networks. A (greatly) simplified model of simulated brain and eyes transmits impulses through the spine to motoneurons, which control tension in muscles along the body. The tension in the muscles subsequently applies forces to joints, which together with forces from an environment determine the position of the body. A system of differential equations represents the configuration of the spine and body. These are numerically integrated to determine the movement of body segments for particular neural and environmental inputs over time. Once locomotion from a simulated spine has been attained, target tracking can be implemented by modulating inputs to the spine from the eyes and simplified brain to produce locomotion towards targets.

One simple vertebrate that has been well studied is the lamprey, which is a jawless eel-shaped fish. It is primitive in an evolutionary sense with its major distinguishing feature being a large rounded sucker surrounding the mouth [3]. The spinal cord is a continuous column of neurons made up of around 100 clusters. Each cluster projects motoneurons to the surrounding muscles [4]. Lampreys swim by propagating a wave along the body from head to tail by phased muscular contraction. In the normal case, the wavelength of this travelling wave is constant and approximately corresponds to the length of the body; its frequency determines the speed of swimming. The lamprey has been studied over several decades (see [5] for a clear introduction and other papers in the same volume e.g. [6] for more details). A variety of simulations has been implemented and our model is based on one of these with minor modifications (see Section 2). Target tracking behavior appears to have been added to these previous simulations almost as an afterthought, mainly to illustrate the locomotion behavior. This is understandable given the significant amount of effort required just to build and simulate the model of locomotion. However, it is a shame because modulating biologically realistic locomotion systems with target tracking signals adds subtle and interesting complexities that must be addressed to maintain robustness and stability of the behavior. Section 3 reviews previous approaches to target tracking for lamprey models and section 4 describes our more robust approach.

## 2 Model

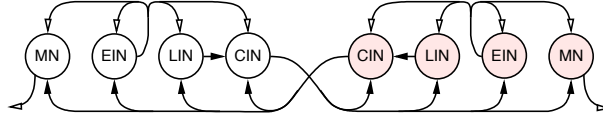
Ijspeert [4] groups neural models of the lamprey into three classes: biophysical, connectionist and mathematical. Biophysical models investigate the detailed low-level neurobiology (on the order on dozens of cells). Connectionist models are less realistic and are interested in connections *between* neurons. Mathematical models are more abstract and view the controller as a chain of oscillators with a focus on the couplings between them. Connectionist models are similar to dynamical recurrent neural networks and compute the mean firing rate of neurons. Our work takes the continuous time *leaky integrator* models used in [7] and [4] as a starting point. In contrast to these models, the neural and physical aspects of our simulation are combined into a single model, rather than two separate but interacting models. This gives significant improvements in simulation speed [8]. The rest of this section describes in detail the complex neural and physical model and where our model differs from [7] and [4]. Detailed knowledge is required to appreciate the effect of modulating inputs for target tracking and to allow replication of our work.

### 2.1 Neural model

Biologically the lamprey spinal cord is a continuous column of neurons without clear boundaries but it can be considered as roughly 100 discrete but interconnected oscillators (or segmental networks). The combined assembly is known as a central pattern generator (CPG). The main types of neuron involved are: motoneurons (MN) projecting to muscles, excitatory interneurons (EIN) projecting to ipsilateral neurons (ie. those on the same side of the segment), lateral inhibitory interneurons (LIN) projecting to ipsilateral neurons, contralateral inhibitory interneurons (CIN) projecting to contralateral neurons (ie. the other side of the segment) and excitatory brain stem (BS) neurons that project from the brain. The controller consists of 100 segmental networks. Each model neuron represents a population of functionally similar neurons. Actual connections between segments are not well known, so Ekeberg in [7] chooses a simplified, symmetric coupling (except for connections from the CINs which are longer tailward). Parameters for both inter- and intrasegmental connections and extents are given in Table 1 and most of the connections for one segment are shown in Fig. 1. In order to limit output from neurons and to compensate neurons in segments near ends of the body (and have fewer intersegmental inputs), synaptic weights are scaled by dividing by the number of input segments. Ekeberg [7] also suggested supplying extra excitation to the first few segments of the spinal column in order to help generate a phase-lagged oscillation down the spine. However, we found in simulations [8] that this is not necessary and reduced the speed of the lamprey swimming. Accordingly, extra excitation was not applied in our model. As mentioned in [4], if the excitation levels for each side of the lamprey neural model are set independently and differently, the lamprey turns. This observation forms the basis for an approach to modulating the lamprey neural assembly with outputs from simple eyes to result in target tracking behavior.

**Table 1.** Neural connection configuration. From [4] with additions from [7] and separately-controllable left- and right-side excitation. Negative weights indicate inhibitory connections. Extents of connections to neighbor segments are given in brackets (headward and tailward, respectively).

To ↓	From: EIN <sub>L</sub>	CIN <sub>L</sub>	LIN <sub>L</sub>	EIN <sub>R</sub>	CIN <sub>R</sub>	LIN <sub>R</sub>	BS <sub>L</sub>	BS <sub>R</sub>
EIN <sub>L</sub>	0.4 [2, 2]	-	-	-	-2 [1, 10]	-	2	0
CIN <sub>L</sub>	3 [2, 2]	-	-1 [5, 5]	-	-2 [1, 10]	-	7	0
LIN <sub>L</sub>	13 [5, 5]	-	-	-	-1 [1, 10]	-	5	0
MN <sub>L</sub>	1 [5, 5]	-	-	-	-2 [5, 5]	-	5	0
EIN <sub>R</sub>	-	-2 [1, 10]	-	0.4 [2, 2]	-	-	0	2
CIN <sub>R</sub>	-	-2 [1, 10]	-	3 [2, 2]	-	-1 [5, 5]	0	7
LIN <sub>R</sub>	-	-1 [1, 10]	-	13 [5, 5]	-	-	0	5
MN <sub>R</sub>	-	-2 [5, 5]	-	1 [5, 5]	-	-	0	5



**Fig. 1.** Schematic of neural connections for one segment. Filled arrows are inhibitory.

Each neuron is modeled as a leaky integrator with a saturating transfer function. Let  $u$  be the mean firing frequency of the population of neurons,  $\xi_+$  and  $\xi_-$  the delayed ‘reactions’ to excitatory and inhibitory input and  $\vartheta$  the frequency adaptation (decrease in firing rate over time given a constant input) observed in some real neurons. Let  $w$  be the synaptic weights of excitatory and inhibitory presynaptic neuron groups  $\psi_+$  and  $\psi_-$ ,  $\tau_D$  the time constant of dendritic sums,  $\tau_A$  the time constant of frequency adaptation,  $\mu$  a frequency adaptation constant,  $\Theta$  the threshold and  $\Gamma$  the gain. It is not possible to directly measure values of these parameters, so values (Table 2) are hand-tuned to produce output matching physiological observations. The neuron is modeled as

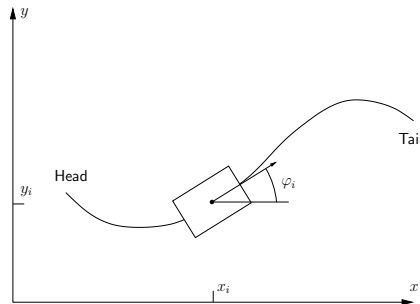
$$\begin{aligned}
 \dot{\xi}_+ &= \frac{1}{\tau_D} \left( \sum_{i \in \psi_+} u_i w_i - \xi_+ \right) \\
 \dot{\xi}_- &= \frac{1}{\tau_D} \left( \sum_{i \in \psi_-} u_i w_i - \xi_- \right) \\
 \dot{\vartheta} &= \frac{1}{\tau_A} (u - \vartheta) \\
 u &= 1 - e^{(\Theta - \xi_+) \Gamma} - \xi_- - \mu \vartheta \text{ if positive} \\
 &0, \text{ otherwise.}
 \end{aligned} \tag{1}$$

**Table 2.** Neuron parameters. From [7]. Symbols are explained in the text.

Neuron type	$\Theta$	$\Gamma$	$\tau_D$	$\mu$	$\tau_A$
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	-

## 2.2 Physical model

As with the neural model, the physical lamprey body is modeled similarly to [7] and [4]. It is represented by ten links with nine joints between them each with one degree of freedom. Ten neural segments act on one body segment or link. Each link is modeled as a right elliptic cylinder with the major axes of the ellipses aligned vertically. All links have length  $l$  of 30 mm, height 30 mm with the width starting at a maximum of 20 mm at the head and decreasing towards the tail. Muscles appear on both sides of the body, attached to the centers of each segment. Muscles are modeled with a spring-and-damper arrangement, where the force exerted by the muscle is set using the spring constant. Local body curvature varies linearly with muscle length. The body and the neural network are linked by having motoneuron excitation drive the muscular spring constants. The body is represented in our model in two dimensions as rectangles with joints at the midpoints of their sides. The position of a link  $i$  is described by  $(x_i, y_i, \varphi_i)$ , where  $x_i$  and  $y_i$  are the co-ordinates of the rectangle centre and  $\varphi_i$  is the angle of a line through the centre and the joint with respect to the  $x$ -axis (see Fig. 2). Constraint forces are used to keep body links together. The physical parameter values for the links are the same as those in [4].

**Fig. 2.** Co-ordinates describing the position of a link. From [7].

Movement of the body results from the interaction of three forces: torques  $T$  generated by the muscles, forces  $W_i$  from the water and constraint forces  $F_i$  and  $F_{i-1}$  that keep the body links together. These forces determine the acceleration

of the links according to Newton's law of motion. Change in position for links  $i \in \{1, \dots, N\}$  is determined by numerical integration of the equations of motion.

$$\begin{aligned}
m_i \ddot{x}_i &= W_{i,x} + F_{i,x} - F_{i-1,x} \\
m_i \ddot{y}_i &= W_{i,y} + F_{i,y} - F_{i-1,y} \\
I_i \ddot{\varphi}_i &= T_i - T_{i-1} - (F_{i-1,x} + F_{i,x}) \frac{l_i}{2} \sin \varphi_i \\
&\quad + (F_{i-1,y} + F_{i,y}) \frac{l_i}{2} \cos \varphi_i
\end{aligned} \tag{2}$$

As described above, muscles are modeled as springs directly connected to sides of links. Force exerted by each spring on its associated joint is determined not only by the local curvature of the body but also linearly by the output of the motoneurons in the associated segments. Let  $M_L$  and  $M_R$  be the left and right motoneuron activity and assign parameters  $\alpha$  ( $=3 \text{ N mm}$ ),  $\beta$  ( $=0.3 \text{ N mm}$ ),  $\gamma$  ( $=10$ ) and  $\delta$  ( $=30 \text{ N mm ms}$ ). As in [7] the torque is defined as

$$T_i = \alpha (M_L - M_R) + \beta (M_L + M_R + \gamma) (\varphi_{i+1} - \varphi_i) + \delta (\dot{\varphi}_{i+1} - \dot{\varphi}_i).$$

Speed of motion through water in our case is sufficiently high that we only account for inertial water force which is proportional to the square of the speed:

$$W = \rho v^2 \frac{A}{2} C$$

where  $\rho$  is the fluid density,  $v$  object speed,  $A$  area parallel to movement and  $C$  drag coefficient. The abbreviation  $\lambda = \rho \frac{A}{2} C$  is made in [7], together with the simplification  $W = W_\perp + W_\parallel = v_\perp^2 \lambda_\perp + v_\parallel^2 \lambda_\parallel$  and values of  $\lambda_\perp$  and  $\lambda_\parallel$  for links.

Body segments are constrained such that for adjacent segments, joints for the facing sides must be in the same position (ie., the links stay joined together). The joint position is expressed in terms of  $x_i$ ,  $y_i$  and  $\varphi_i$  for  $i \in \{1, \dots, n-1\}$ , so

$$\begin{aligned}
x_i + \frac{l_i}{2} \cos \varphi_i &= x_{i+1} - \frac{l_{i+1}}{2} \cos \varphi_{i+1} \\
y_i + \frac{l_i}{2} \sin \varphi_i &= y_{i+1} - \frac{l_{i+1}}{2} \sin \varphi_{i+1}
\end{aligned} \tag{3}$$

Equations (2) and (3) form a differential-algebraic equation (DE) system [7] typical of non-minimal coordinate systems that can be numerically integrated.

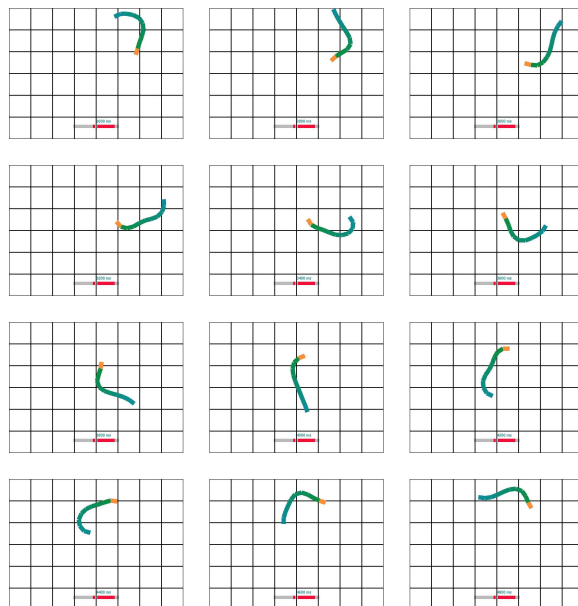
The models described above were implemented in C++ and rendered graphically in Python using PyOpenGL [9]. For details of the implementation and approaches to increasing simulation speed, together with typical model behavior see [8]. A 10 s simulation takes 620 s to run on an AMD 1800 CPU. After a start up period the steady state swimming speed and characteristics observed in our model are similar to those observed by [4].

### 3 Approaches towards target tracking

Even a simple, hard-coded oscillator can generate appropriate neural outputs for straight-line swimming. However, for applications such as target tracking,

the pattern generator must be able to accept and respond to changing conditions in the environment while still emitting a stable oscillation pattern. It must have inputs that cause it to change its outputs in relevant ways, and be able to accept a wide range of varying inputs without the oscillation collapsing and a subsequent stop of locomotion. In summary, the CPG must be robust to changes in the environment whilst producing reasonable behavior.

Ijspeert describes two different approaches to making the lamprey swim towards a target [4, 10]: (i) simple bearing-based tracking and (ii) exponentiated bearing-based tracking. Both approaches build on the observation that the lamprey can be made to turn by supplying different levels of excitation to the left and right halves of each segment (see Figure 3). Two notional “retinas” supply excitation that varies according to the relative bearing of the target. Neither case considers the distance to the target.



**Fig. 3.** Circling behavior with  $BS_L = 0.1$  and  $BS_R = 0.8$ . Gridlines 100 mm apart.

### 3.1 Simple bearing-based tracking

In this arrangement [4], retina output varies linearly with bearing to the target. The lamprey is considered to have a “dead zone” in that it does not respond to targets bearing more than  $150^\circ$  from the head axis. Any bearing more acute than this is linearly transformed into an excitation between 0 and 1 for the side of the body on which the target lies, and no excitation on the opposite side.

This model of vision is somewhat effective, but suffers from two major weaknesses. The most obvious is the dead zone: if the target is located out of the field of view, brainstem excitation drops to 0 resulting in a shutdown of the pattern generators. Even if the randomly moving target eventually reappears, the lamprey may not be able to restart its swimming. The spinal cord needs specific startup conditions or phased oscillation will not occur.

More subtle is the effect of linearly mapping the bearing to excitation. As the bearing to the target approaches  $0^\circ$ , so does the excitation from the brainstem. The more the lamprey turns to face the target, the less excitation is supplied, often resulting in shutdown. The case illustrated in [4] is a favorable one where the target crosses the lamprey’s field of view repeatedly, so that the lamprey spends very little time unexcited. However, not all random movements of the target will be so fortunate.

### 3.2 Exponentiated bearing-based tracking

A more complicated tracking system for a salamander simulation is developed in [10]. Here, the retinas are considered to have axes offset from the head’s major axis. (The offset is not given but would be about  $\pm 30^\circ$ .) Excitation again ranges between 0 and 1 but is calculated as  $R = e^{-\alpha \Delta\phi^2}$  where  $\alpha = 0.0005$  and  $\Delta\phi$  is the angle between the axis and the target. In effect, excitation is 67% for a target directly ahead of the lamprey, marginal (10%) at  $90^\circ$  away and nil at  $150^\circ$ . Retina output depends only on the bearing of the target from the major axis of the head, with an angular offset.

Observations of experiments with this tracking arrangement demonstrate that it performs far better than the linear mapping. The CPG receives levels of excitation sufficient to sustain pattern generation for almost the full range of bearings within the “dead zone” seen in linear mapping, and in particular a target directly in front of the lamprey causes enough excitation for effective pursuit. However the problem of the lamprey becoming becalmed when it reaches orientations where the target is in the dead zone remains.

## 4 Bearing-based tracking with foraging

The previous section relates two methods of implementing tracking behavior to modulate the inputs to a lamprey simulation. However, these approaches suffer from the problem that the lamprey can become becalmed in some situations.

Our solution to the becalming problem is to apply a foraging algorithm when the target disappears from the field of view. This is implemented by sending the lamprey into a fast circling action when the target disappears from view. One input to the spine (arbitrarily the right-side) is set to 0.8 and the other to 0.1 until the target is recovered. These values result in a fast, tight circling motion without being so excessively high as to swamp the CPG. In combination with the exponentiated tracking arrangement the lamprey becomes able to effectively track targets exhibiting a wide variety of random behaviour: straight, circling or weaving motion, and holding stationary or almost stationary (Figure 4).



### 4.1 Observations

Although exhaustive tests have not been conducted, in general the CPG withstands the constantly-changing brainstem inputs remarkably well. It should be noted that apart from the variation in input caused by the random movements of the target, the lamprey's head moves from side to side during normal swimming, causing a 6 Hz oscillation in the brainstem activation. That the CPG generally continues to function even when subjected to an oscillatory signal of the same frequency but not the same phase is testament to its stability.

This resistance is not perfect. In runs with a fixed target the lamprey will from time to time cease forward motion and propagate waves in place, or even briefly move in reverse. These effects are short-lived, with the pause lasting around 700 ms before being overcome by the natural phase delay of the CPG. In runs with a randomly moving target this effect is much rarer, though still in evidence. The random component in excitation levels that results from the random motion makes the particular confluence of oscillations described above less likely to occur. Random target motion also reduces the likelihood of triggering the permanent becalming conditions described with the previous approaches.

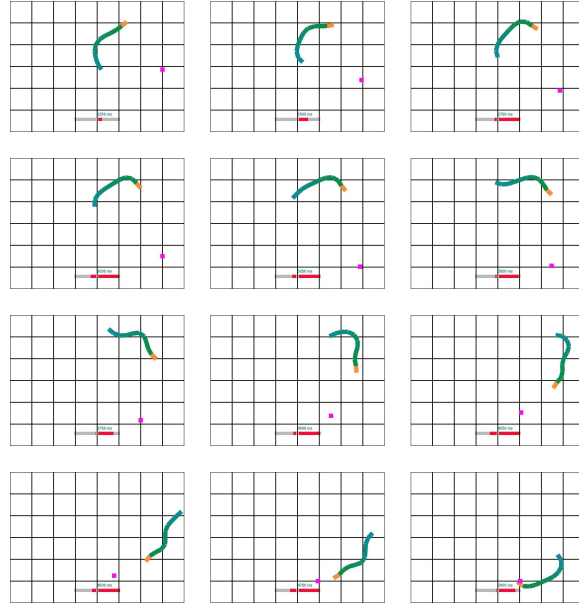
Finally, it should be noted that the arrangement of two sensors tied to two physical systems is distinctly reminiscent of Braitenberg's famous vehicles [11]. The lamprey exhibits the source-seeking, AGGRESSIVE behaviour of Vehicle 2b, despite being wired as a Vehicle 2a. This is of course because increased activity on a given side of the lamprey propels it towards that direction, while increased activity in the motor of a Braitenberg vehicle propels it away.

## 5 Conclusion

Simulation of lifelike locomotion has applications to computer graphics and robotics. Once locomotion is achieved it is interesting to simulate higher level behavior such as target tracking. Previous approaches to target tracking in neural simulations of a lamprey suffer from deficiencies where the lamprey loses neural input and is becalmed in some configurations. We present an approach to overcoming this problem based on an underlying foraging behavior and show simulation results. There are several opportunities for farther work, including identification of quantitative measures of the behavior despite the long simulation times involved. We are interested in determining whether other patterns of foraging than tight circling, such as slow random movements, also solve the problem. Also, we plan to explore other higher level behaviors and apply similar models to other skeletal configurations and armatures within computer graphics animation packages.

## References

1. Pfeifer, R., Scheier, C.: *Understanding Intelligence*. MIT Press (1999)
2. Foley, J.D., van Dam, A., Feiner, S.K., Hughes, J.F.: *Computer graphics: principles and practice*. 2nd in C edn. Addison-Wesley (1996)



**Fig. 4.** Target-tracking behavior. Gridlines are 100 mm apart.

3. Janvier, P.: Tree of Life: Hyperoartia. URL: <http://tolweb.org/tree?group=Hyperoartia> (1997)
4. Ijspeert, A.J.: Design of artificial neural oscillatory circuits for the control of lamprey- and salamander-like locomotion using evolutionary algorithms. PhD thesis, Department of Artificial Intelligence, University of Edinburgh (1998)
5. Lansner, A., Ekeberg, Ö., Grillner, S.: Realistic modeling of burst generation and swimming in lamprey. In Stein, P.S.G., Grillner, S., Selverston, A.I., Stuart, D.G., eds.: *Neurons, Networks and Motor Behaviour*. The MIT Press (1997) 165–172
6. Wallén, P.: Spinal networks and sensory feedback in the control of undulatory swimming in lamprey. In Stein, P.S.G., Grillner, S., Selverston, A.I., Stuart, D.G., eds.: *Neurons, Networks and Motor Behaviour*. The MIT Press (1997) 75–81
7. Ekeberg, Ö.: A combined neuronal and mechanical model of fish swimming. *Biological Cybernetics* **69** (1993) 363–374
8. Beauregard, M., Kennedy, P.J.: Fast simulation of animal locomotion: lamprey swimming. To appear in *Symposium on Professional Practice at 19th IFIP World Computer Congress, Santiago, Chile, 20–25 August 2006* (2006)
9. Fletcher, M.C., Liebscher, R.: PyOpenGL – the Python OpenGL binding. URL: <http://pyopengl.sourceforge.net/> (2005)
10. Ijspeert, A., Arbib, M.: Visual tracking in simulated salamander locomotion. *Proceedings of the Sixth International Conference of The Society for Adaptive Behavior (SAB2000)*, Paris, France, 11–15 September 2000 (2000)
11. Braitenberg, V.: *Vehicles*. The MIT Press (1984)