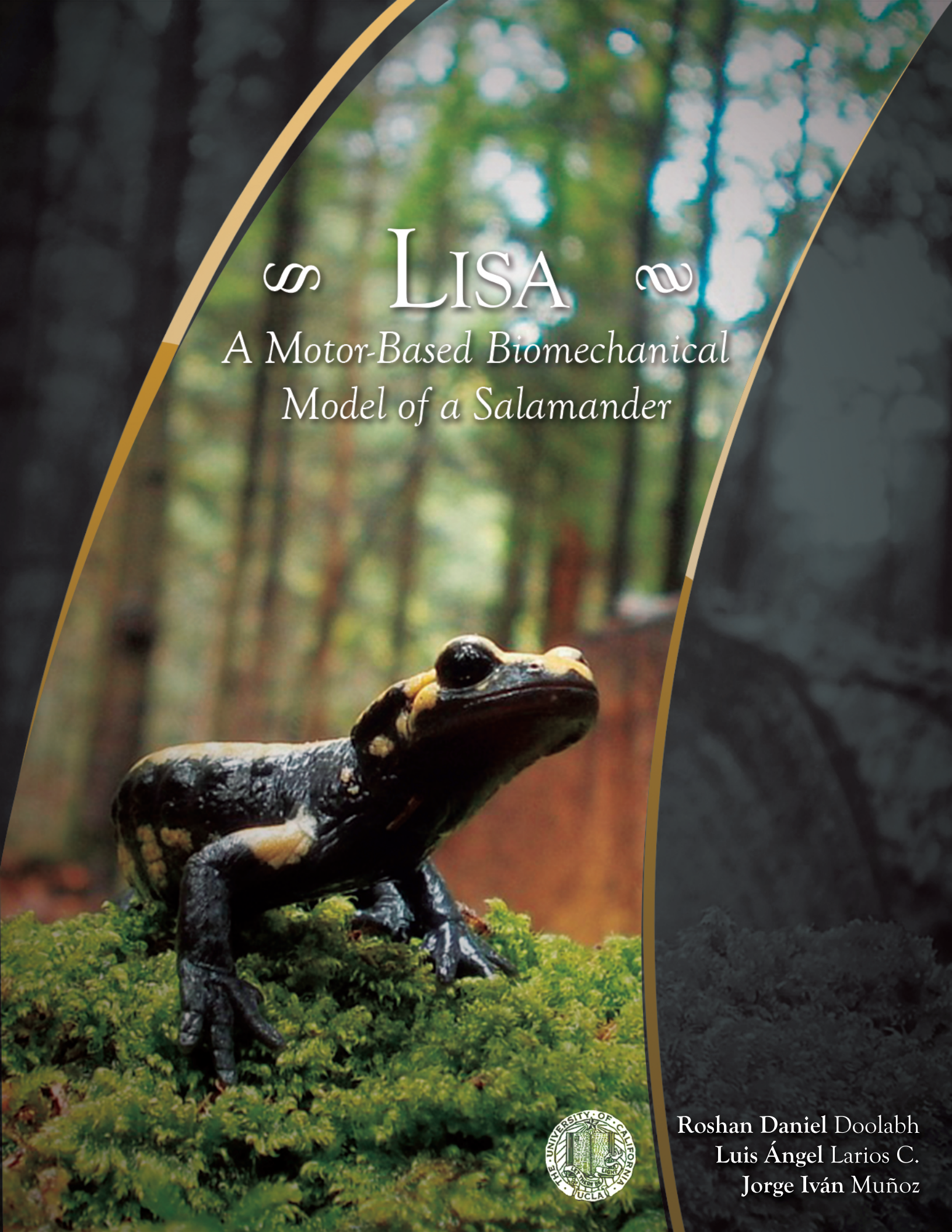


LISA

A Motor-Based Biomechanical Model of a Salamander

Roshan Daniel Doolabh
Luis Ángel Larios C.
Jorge Iván Muñoz

LISA

A Motor-Based Biomechanical Model of a Salamander

Artificial Life Term Project Report

March 19, 2014

Prof. Demetri Terzopoulos

COM SCI 275: Artificial Life

Term Project Report

LISA

A Motor-Based Biomechanical Model of a Salamander

Roshan Daniel Doolabh, Luis Ángel Larios Cárdenas, and Jorge Iván Muñoz

ABSTRACT

This article describes a 2D-based simulation of a salamander as an articulated body that interacts with its environment to achieve various locomotive states. The aim of the simulation is to implement a robust biomechanical model whose motion is comparable to the aquatic and terrestrial locomotion of a real salamander. Our model uses a motor-based approach to control the contraction and expansion of abstract muscles at the joints, as opposed to the traditional spring based models, which allows for a higher level of control.

1. INTRODUCTION

The physically-based simulation of a biomechanical model for a salamander was inspired by the work of Ijspeert, A. J., 2000. Salamanders possess the ability to trot using gaits similar to those of quadrupeds, or swim in an aquatic environment. While trotting, the salamander's body performs an S-shaped standing wave with nodes at its waist (Frolich, L. M., et al, 1991). The bending of the trunk allows for the salamander to increase the reach of its limbs, which are attached to the sides of the trunk. The S-wave trotting can be produced by coupling chain oscillators with coupling between the extremities and the middle of the chain (Ermentrout, B., et al, 1994). The aquatic locomotion of the salamander is based on the locomotive patterns of a previous lamprey model (Cohen, A. H., 1994).

Our biomechanical model is constructed of a chain of rigid body links that are connected through hinge joints wrapped by pairs of muscles to control them. We use sinusoidal muscle activation functions to generate the different phases of locomotion. The terrestrial motion simulates the observed S-shape movement of real salamanders and can perform general gaits such as trotting straight and turning left and right with ease by applying friction forces and muscle activation. Our creature is able to transition between the terrestrial locomotion and aquatic locomotion when prompted and exhibits the traveling wave pattern of the lamprey. The aquatic model propels itself forward using simplified hydrodynamic equations and is able to turn as well using muscle activation patterns.

2. RELATED WORK

Since the early nineties much research has devoted to the analysis of walking and swimming gaits of amphibians. In particular, salamanders have attracted most of the attention because they provide the opportunity to study the transition from aquatic to terrestrial motion. Also, from the robotics and physics-based simulations point of view, much valuable information can be gained from understanding a salamander's stance since the limb disposition on the body periphery accounts for better support and stability than other erect postures currently being used in biologically-inspired robots and webots.

Salamander swimming resembles that of a lamprey, for which there already exist plenty of studies that explain in detail how ***Central Pattern Generators*** (CPG) play the role of oscillators that continuously activate muscles. Thus, early research on the simulation of salamander locomotion can be traced back to models that actually implement CPGs to produce locomotion in lampreys.

The first neurobiological-inspired computational model of lamprey swimming was due to Ekeberg, Ö., 1993. In his work, Ekeberg developed a 2D biomechanical representation of the finless fish, which included a neural network whose connectivity had been previously established in other studies. The key contribution of Ekeberg's research consisted of coupling a CPG to an articulated, rigid-body-based mechanical lamprey that swam in a virtual aquatic environment. The simulated lamprey was able to react to external forces while maintaining its body structure by applying internal forces. Moreover, he introduced the use of springs into his model in order to generate torques on the fish's joints, imitating

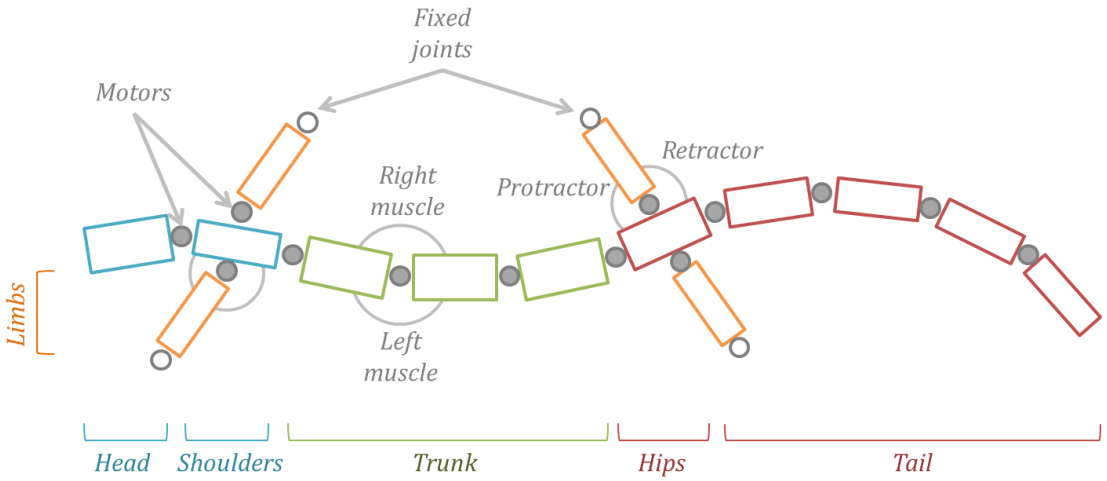


***Figure 1**, a diagram of our salamander's model.*

the contraction and relaxation of muscles that are responsible for movement.

Following Ekeberg's work, in 1998, Ijspeert, A. J., et al, extended the lamprey's CPG to control the activation of muscles in a 2D salamander. They used a genetic algorithm to evolve previously working lamprey-like swimming controllers into trotting and switching controllers in an incremental fashion. Their neuronal approach was able to mimic the S-shaped stance of a salamander when trotting and the traveling wave from head to tail when swimming. Similarly to Ekeberg, the activation of motoneurons caused the contraction of the corresponding muscles after increasing their spring stiffness constants. Furthermore, to create the sense of motion on dry ground, they added four more rigid links to the lamprey's body in order to make the model more stable. These links represented the salamander's limbs and friction was applied to the tips of the legs for which their flexor muscles were contracted. We have taken this idea of applying friction to propel our creature forward; however, our approach for muscle simulation supersedes traditional springs with a more stable means of simulation-motors. These motors simplify the control problem and work around the difficult task of defining appropriate values for spring constants.

A solo work by Ijspeert, A. J., in 2000 took his 1998 approach of a CPG-brained salamander into yet another 2D model where the neuronal controllers were removed in favor of sinusoidal functions. All features of spring-based muscles and hinge joints added little to the degrees of freedom in the leg links despite the claim that the salamander was a 3D model. In fact, the calves were attached to the upper limbs that the salamander lifted during the swing phase of trotting; nonetheless, the salamander's trunk was always in contact with the ground, simulating a realistic drag on the floor caused by friction. For this reason, lifting limbs did not provide another dimension of motion to the other two dimensions already existing in his 1998 model. However, we have taken his approach into account in our model; we represent the activation of abstracted muscles with sinusoidal functions. In essence, the only difference between Ijspeert's model and ours lies in the use of motors instead of springs as we will explain in detail in the next section.

3. Implementation

Our approach for modeling a salamander is defined in terms of the features provided by ***ODE*** (Open Dynamics Engine) (Smith, R., 2007), which is the platform we chose to produce a physics-based simulation. The creature is made of 10 rigid body segments (called links) that represent its head, shoulders, trunk, hips, and tail. Additionally, it has four legs with two links each, where the upper limbs are initially set perpendicular to the trunk line, and the lower legs, which support the salamander's weight, are always perpendicular to the upper limbs (mimicking calves). Figure 1 shows the model of our virtual amphibian.

From Figure 1 we can observe that the links along the head, shoulders, trunk, hips, and tail are connected through joints (shaded circles) that have only one degree of freedom and whose axes of rotation are perpendicular to this sheet of paper. These hinges represent constraints that keep the whole body together by just allowing a waving motion of the segments on the coronal plane of the amphibian. Likewise, the four upper limbs are connected, in pairs, to the shoulders and hip links respectively. The upper limbs are constrained to an arc motion since these joints have one degree of freedom. This also holds true for the rest of the mobile links. Furthermore, the axes of rotation for these links are parallel to all other hinges' axes located along the body. In addition, the calves cannot move from their initial location and we have attached them to the upper limbs with fixed joints, represented with empty circles in Figure 1.

Link	Length (m)	Width (m)	Height (m)	Mass (Kg)
Head	0.03	0.03	0.04	0.177
Shoulders	0.03	0.025	0.04	0.147
Trunk 1	0.03	0.03	0.04	0.177
Trunk 2	0.03	0.03	0.04	0.177
Trunk 3	0.03	0.03	0.04	0.177
Hips	0.03	0.022	0.04	0.13
Tail 1	0.03	0.02	0.04	0.118
Tail 2	0.03	0.016	0.04	0.094
Tail 3	0.03	0.01	0.04	0.059
Tail 4	0.03	0.005	0.04	0.029
Left fore-upper limb	0.025	0.01	0.01	0.02
Right fore-upper limb	0.025	0.01	0.01	0.02
Left hind-upper limb	0.025	0.01	0.01	0.02
Right hind-upper limb	0.025	0.01	0.01	0.02
Left fore-calf	0.02	0.01	-	0.012
Right fore-calf	0.02	0.01	-	0.012
Left hind-calf	0.02	0.01	-	0.012
Right hind-calf	0.02	0.01	-	0.012

***Table 1**, biological measurements for the salamander's rigid bodies.*

In the pursuit to simulate an accurate biomechanical model, the salamander's segments are ODE primitive **box-like** geometries (to handle collisions) attached to their respective dynamics bodies (to handle forces and torques). Each link in Figure 1 is represented as a rigid body with mass, length, width, and height that correspond to actual biological metrics from a real salamander. Table 1 summarizes these measurements for the salamander's segments; it is an adaptation of the information available in Ijspeert, A. 2000. Unlike the rest of the segments, calves are not box-like rigid bodies but are capsules where the specified widths are each capsules diameter. The reason for the choice of capsules to model lower legs will become evident following the section on trotting locomotion.

Unlike previous models (Ijspeert, A., 2000; Ekeberg, Ö., 1993; Harischandra, N., et al, 2010), we have abstracted the salamander's muscles into the activation of ***motors*** rather than implementing springs. Each hinge joint has a virtual ODE motor, controllable through ***desired velocities*** that the user can provide at every time step. Thus, looking at Figure 1, the salamander can be represented with 4 types of muscles that manipulate the state of its joints:

a) ***Left*** and ***right muscles*** wrap the joints along the head, shoulders, trunk, hips, and tail. This provides the sense of bending the body either to the left or to the right.
b) ***Protractor*** and ***retractor muscles*** change the state of the joints that connect upper legs to the actual body of the salamander. A protractor muscle, when active, yields limb forward motion, while a retractor muscle acts on the same limb but induces its motion in the opposite direction (e.g. pushes the leg backward).

We can consider this collection of muscles as a set of pairs of ***agonist*** and ***antagonist contracting fibers*** that produce a change in the current configuration angle of each movable joint in the salamander. Thus, by controlling and synchronizing the successive activation of muscles, we can generate different body shapes and even locomotion if we add friction to calves and include other external forces to act on the segments in response to the contraction (activation) of certain muscles in the body of the amphibian.

3.1. Control

In order to change the state of a salamander's joints we have to provide a desired velocity. ODE will ultimately take that desired velocity as input to generate a proportional torque to achieve the indicated desired velocity in each time step. However, this desired velocity must be chosen carefully and in synchrony with the rest of the joints if we want, for instance, to bend the creature's body into the natural "S" shape that real salamanders exhibit when trotting. In simplified terms, while some joints should bend in one direction, others should bend in the opposite direction or in another shifted angle that smoothly varies from joint to joint. Biologically, this is possible with the intervention of a Central Pattern Generator (Ekeberg, Ö., 1993), which is the leading mechanism behind the activation of the distinct muscles that control the joints of the amphibian.

For the sake of simplicity, we have replaced a CPG with two activation functions to control the change of state of each joint at all times. These activation functions or "neurosignals" enjoy the property of mutual exclusiveness so that at most one abstract muscle in an agonist-antagonist pair can be contracting at a given instant.

Another biological feature of CPGs is periodicity: by sending repeated patterns of activation to muscles over periods of time, muscles are continuously contracting and relaxing, favoring the emergence of locomotion. In our case, the neurosignals we have selected also possess the property of periodicity, and, mathematically, they are represented in the following way:

$$M_1(t) = \min(sin(\omega t - \lambda_i), 0)$$
$$M_2(t) = \min(sin(\omega t - \lambda_i + \pi), 0)$$

These neurosignals (Figure 2) are always positive but never active at the same time. Moreover, we can change the speed of activation of muscles if we provide a different frequency ω, and the i^{th} joint can be active with a slight phase with respect to its neighboring joints if we

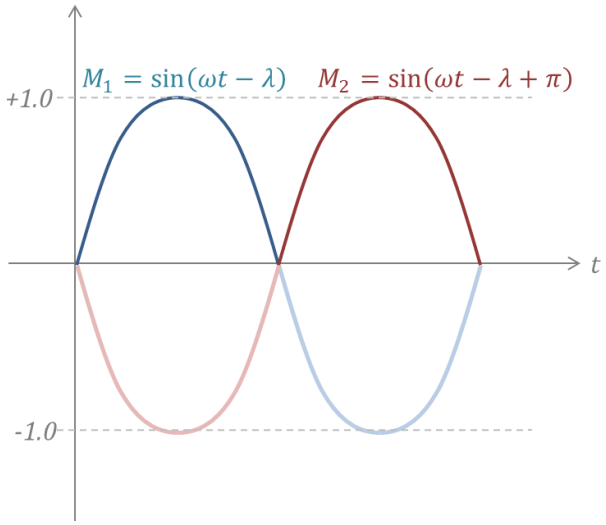


***Figure 2**, the two neurosignals that control each muscle (and, therefore, each motor-based join) in the salamander.*

set its λ_i to a value different than that of its neighbors $\lambda_i \neq \lambda_j, i \neq j$.

In our implementation, $M_1(t)$ has been assigned to the left and retractor muscles, while $M_2(t)$ controls activation of the right and protractor muscles.

Neurosignals are the key for changing the state of a joint. Since $M_1(t)$ and $M_2(t)$ are always independent, we can take advantage of their mutual exclusiveness. For instance, during the time that one of them is active, we can force the joint to bend in one direction, and start bending in the opposite direction once the other neurosignal becomes active. All that remains is to translate neurosignals into a ***deformation angle*** to express the sense of bending direction for a joint. In other words, we should transform neurosignals into the desired velocities that drive the motion of the motors attached to the salamander's joints.

Figure 3 describes how ODE treats deformation angles and desired (angular) velocities in a joint motor. The dynamics of how ODE was utilized can be best explained by picturing the illustrated motor-based hinge as one of the salamander's trunk joints viewed from above. The initial configuration of the joint (e.g. when the hinge was created) has both $link\ 1$ and $link\ 2$ attached to the joint along the dotted horizontal line. At that moment, ODE considers the initial deformation angle θ being zero. Also, we have defined the axis of rotation as the axis perpendicular to this sheet of paper, departing from the sheet plane towards the reader. In this particular example, if $link\ 1$ moves downwards, the yielded angle becomes positive after following the right hand convention had we applied a positive torque (with respect to the hinge axis of rotation) on $link\ 1$. As long as $link\ 1$ was the first link attached to the joint during its initial configuration and we wanted to generate a positive deformation angle, we

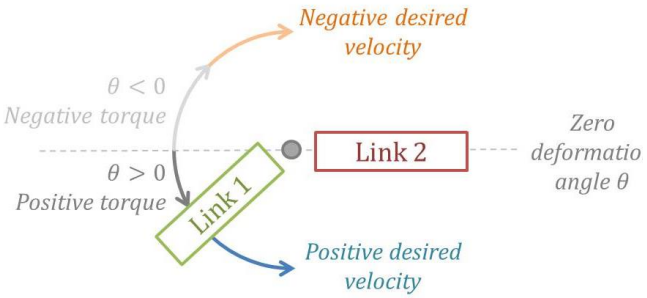


***Figure 3**, deformation angles, desired velocities, and torques in ODE joints.*

should provide the motor with a downwards ***desired angular velocity*** (a scalar quantity), and ODE would automatically compute the necessary torque to drive the hinge angular motion to a constant positive velocity.

Ultimately, the deformation angle in a joint is a synonym for direction, and we can define ***target deformation angles*** as bases to compute desired angular velocities. Since we already have neurosignal functions to indicate directions of motion, such functions can be used to compute desired velocities at every time step in the following way:

We consider a maximum deformation angle $0 < \theta_{max} < \frac{\pi}{2}$ for each mobile hinge joint. Also, we assume that, from Figure 1, all links along the head, shoulders, trunk, hips, and tail are attached to hinges from left to right, e.g. $link\ 1$ is the head and $link\ 2$ is the shoulders for the first joint, $link\ 1$ is the shoulders and $link\ 2$ is the first trunk link for the second joint, and so on; and for limbs, they are attached bottom – up, e.g. left fore-upper limb is $link\ 1$ and shoulders is $link\ 2$, shoulders is $link\ 1$ and right fore-upper limb is $link\ 2$, and so on. Then, a target deformation angle is defined as

$$\theta_{target} = \begin{cases} A_1 * \theta_{max} & if\ M_1(t) > 0 \\ -A_2 * \theta_{max} & if\ M_2(t) > 0 \end{cases}$$

where $A_1, A_2 \in [0,1]$ state how much of a maximum deformation angle we want a joint to bend in each direction. We require θ_{target} be negative for $M_2(t) > 0$ because, according to the ODE scheme described in Figure 3, this assignment is going to yield negative desired velocities when the second neurosignal is active.

Now, whenever a joint's neurosignal enters its period of activation, the creature should stop bending that joint in the direction indicated by the antagonist neurosignal (which should be turned off at this point), and compute a target deformation angle in the new desired direction. For example, if $M_1(t)$ was activated, we get the scenario of Figure 4. In that case, we want to be able to obtain a desired positive velocity at each time step so that we can

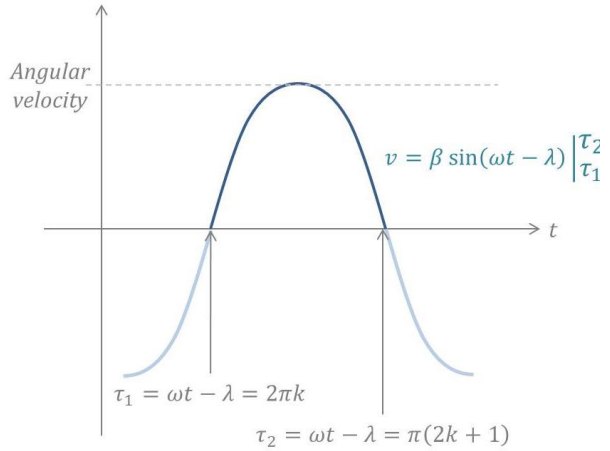


***Figure 4**, computing the desired velocity during the period that M_1 is active.*

safely "reach" the desired target deformation angle θ_{target} at the end of $\frac{\pi}{\omega}$ seconds. Specifically, when a change of activation in favor of $M_1(t)$ occurs, a new ***angular distance*** to traverse is calculated from the current deformation angle:

$$\Delta\theta = \theta_{target} - \theta_{current}$$

One way to ensure traversing $\Delta\theta$ in $\frac{\pi}{\omega}$ seconds consists of having a desired velocity function with the same shape of the neurosignal $M_1(t)$ because the latter is guaranteed to complete its activation within the time constraint. Also, $M_1(t)$ has the nice property of a smooth variation from 0 to a maximum value and then back to zero again. If this property is applied to the angular velocity, then the corresponding joint would start bending from a stationary point, peak its speed halfway, and come to a stop once it reaches the target deformation angle. Therefore, we couple the desired velocity with the joint's neurosignal to obtain the following expression

$$v(t) = \beta M_1(t) = \beta \sin(\omega t - \lambda_i)\Big|_{\frac{2\pi k+\lambda_i}{\omega}}^{\frac{\pi(2k+1)+\lambda_i}{\omega}}$$

where $(\omega t - \lambda_i) \in [2\pi k, \pi(2k+1)]$, $k \in \mathbb{N}$, and β is the unknown amplitude that multiplies the current neurosignal value $M_1(t)$ so that the joint is able to traverse the whole $\Delta\theta$ in a timely manner. Thus, in order to solve for β we use the fact that the integral of velocity is distance, which in this case is the desired angular displacement $\Delta\theta$. Therefore

$$\int_{\frac{2\pi k+\lambda_i}{\omega}}^{\frac{\pi(2k+1)+\lambda_i}{\omega}} \beta \sin(\omega t - \lambda_i)\, dt = \Delta\theta$$

$$\therefore \beta = \frac{(\omega\Delta\theta)}{2}$$

and the desired velocity at each time step during the activation of $M_1(t)$ is

$$v(t) = \frac{\omega\Delta\theta}{2} M_1(t) \geq 0$$

Similarly, when $M_2(t) > 0$

$$v(t) = \frac{\omega\Delta\theta}{2} M_2(t) \leq 0$$

because $\Delta\theta \leq 0$. It is important to note that $\Delta\theta$ is computed only once, at the right moment that a change of activation happens; that way, $v(t)$ remains in the same form during the semi-cycle in which the corresponding neurosignal $M_1(t)$ or $M_2(t)$ is non zero.

With our framework, it is possible to generate different gaits, such as trotting and swimming, vary the speed and shape of motion, and even allow for turning in water or land after the inclusion of external forces. In Ekeberg, Ö., 1993, turning occurs when the body of the salamander bends more in one lateral direction than another. This happens in the original implementations of spring-based models because the researchers use the amplitude of neurosignals to produce bending. In our approach, we get the same effect by providing asymmetrical values to A_1 and A_2 which indicate how close we want a joint to reach its maximal deformation angle in one or another direction. More specifically:

$$Locomotion = \begin{cases} Left & A_1 > A_2 \\ Right & A_2 > A_1 \\ Straight & A_1 = A_2 \end{cases}$$

And, in our experiments, we set an allowance $A_0 = 0.5$, such that $A_1 = A_2 = A_0$ for straight locomotion and $A_1, A_2 \in (A_0, 2A_0]$ for turning in either direction.

3.2. Trotting Gait

Although our salamander model is a 3D graphical construct, the dynamics of locomotion are constrained to 2D because all joints are hinges. We generate trotting by activating and deactivating friction in cross-opposite calves. The lower legs (not shown in Figure 1) are essentially ODE capsule primitives that are attached to the upper legs through fixed joints in a perpendicular angle. Accordingly, the current implementation of our salamander does not lift its legs to initiate or continue the swing phase when trotting. We have chosen capsules because the number of contacts generated during collision detection with the ground is minimal, which reduces the contact surface and favors dynamics stability.

As introduced in section 1, during trotting, a salamander bends its body into an S-shape. Forward propulsion occurs by contracting its retractor muscle on the fore-limb

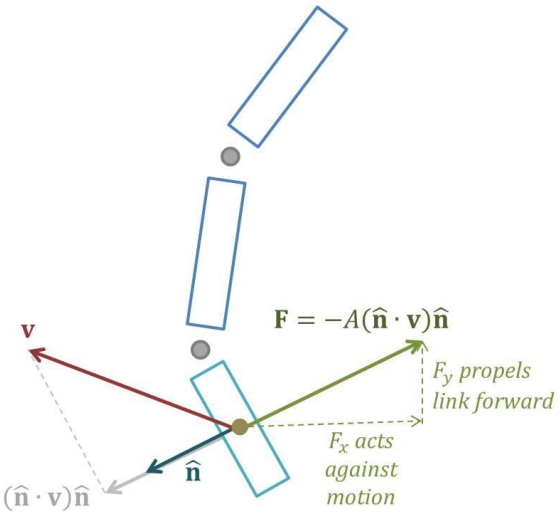


***Figure 5**, simplified hydrodynamic forces acting on a salamander's tail link.*

ipsilateral to the active muscles on the trunk, and by contracting its retractor muscle on the hind-limb that is contralateral to the activation of the muscles along the trunk.

Mathematically, λ_i, the phase of a joint in $M(t)$, is used to generate time-based shifted hinge activation signals. To achieve an S-shape we set $\lambda_i = 0$ for trunk joints ($i = 1,2,3,4$, if one enumerates joints from 0 to 8 in the head-to-tail direction in Figure 1), and $\lambda_i = \pi$ for $i = 0,5,6,7,8$. Intuitively, with such formulation, when the salamander bends its trunk to its left, both head and tail bends to its right, creating the visual appeal that characterizes the natural land motion of this kind of animals.

Furthermore, legs are synchronized with the rest of the body too, where limb muscles are in phase with their cross-lateral limb muscles, and out of phase with respect to their ipsilateral neighboring joints. This way, taking the trunk as the "bending" reference, $\lambda_j = 0$ for $j = 0,2$, i.e. left fore-limb and right hind-limb, and $\lambda_j = \pi$ for $j = 1,3$, i.e. right fore-limb and left hind-limb. Thus, in order to generate motion, we apply friction on the calves whenever their respective upper leg retractor muscle is in contraction, e.g. $M_1(t) > 0$, and remove it as soon as M_2 enters its activation period. Consequently, the alternate activation of the neurosignals translates into "impulses" of friction on calves. These friction applications push the salamander's body forward at the same time such that it balances side-to-side by synchronizing the rest of the links along its body.

3.3. Swimming

Salamanders adopt a lamprey-like gait when they propel in aquatic environments (Ijspeert, A. J., et al, 1998). When a lamprey swims, one can observe a traveling wave that linearly increases its amplitude as it propagates from head to tail. In our model, we set different activation phases for the head, shoulders, trunk, hips, and tail hinge joints in order to achieve the same visual effect. Thus, while swimming, $M_1(t)$ and $M_2(t)$ utilize $\lambda_i = i\frac{(2\pi)}{9}$, for $0 \leq i \leq 8$. Moreover, we tune the joints' amplitude by setting their respective allowance parameter A_0 to a linearly increasing value from 0 to 0.5. With this formulation, we can still use our approach for turning to either side if we make $A_1, A_2 \in [A_0, 2A_0]$, as it has been explained in section 3.1.

Unlike trotting, there is no friction applied to the salamander's body during swimming. The external forces that act on the body are simulated via simplified hydrodynamics, mostly inspired by the idea in Terzopoulos, D., et al, 1994. We treat each of the head, shoulders, trunk, hips and tail links of Figure 1 as flat surfaces with zero width, onto which fluid exerts pressure in a direction parallel to their outward normal vectors. We have excluded limb segments from fluid force calculations. Figure 5 illustrates this method.

In a broad sense, hydrodynamic forces are negatively proportional to the projection of the velocity onto the outward normal unit vector of a corresponding segment. These forces have the effect of counteracting motion at the same time that aid in propelling the body forward or towards any desired direction embedded in the current linear velocity of the links. Formally

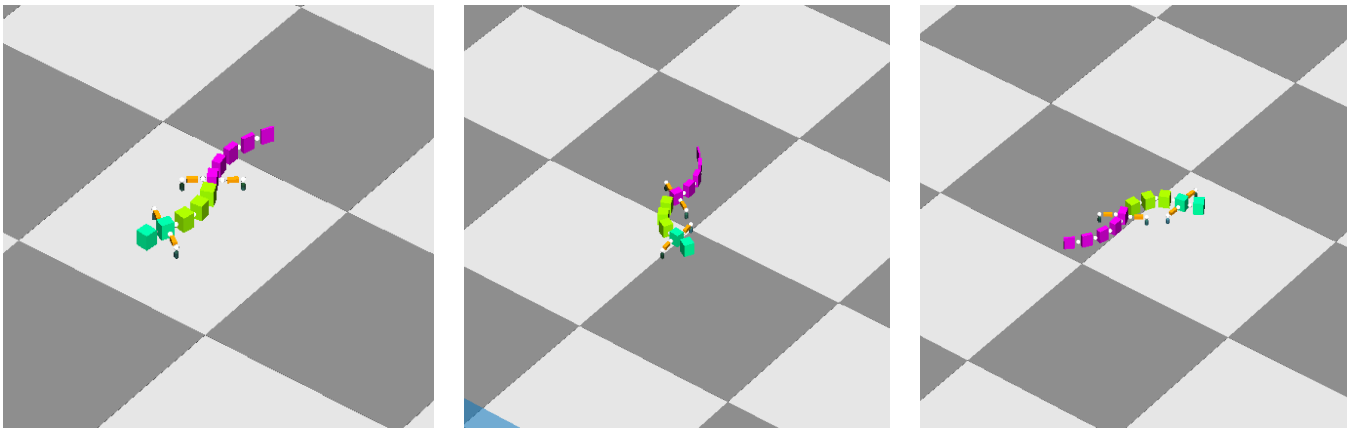
$$\mathbf{F} = -A(\hat{\mathbf{n}} \cdot \mathbf{v})\hat{\mathbf{n}}$$

where A is a proportionality constant that is usually associated to the area of the segment surface, $\hat{\mathbf{n}}$ is the outward unit normal of a lateral face of the segment, and $\mathbf{v}$ is the current linear velocity of such link. In our case, hydrodynamic forces are strictly applied to lateral faces, disregarding fluid forces acting on the top, bottom, front, and back faces of the box-like segments that make up the salamander's body. This way, as well as trotting, swimming is constrained to a plane that is parallel to the XZ plane of the simulated real world frame.

3.4. Transitioning Gait Modes

Transitioning from trotting to swimming and vice versa requires an immediate re-computation of target deformation angles, θ_{target}, and allowance amplitudes, A_0, at each joint. However, we cannot apply such changes as soon as the salamander is prompted to change locomotion mode. Doing so might trigger unexpected "breakages" along the salamander's spinal cord, producing undesired visual/unnatural transitions.

For this reason, when a change of locomotion mode is detected, we indicate joints to go back to their initial configuration state, i.e. $\theta_{target} = 0$, because that is the



(a) Trotting straight *(b) Turning to the left* *(c) Turning to the right*

***Figure 6**, Lisa in trotting locomotion mode.*

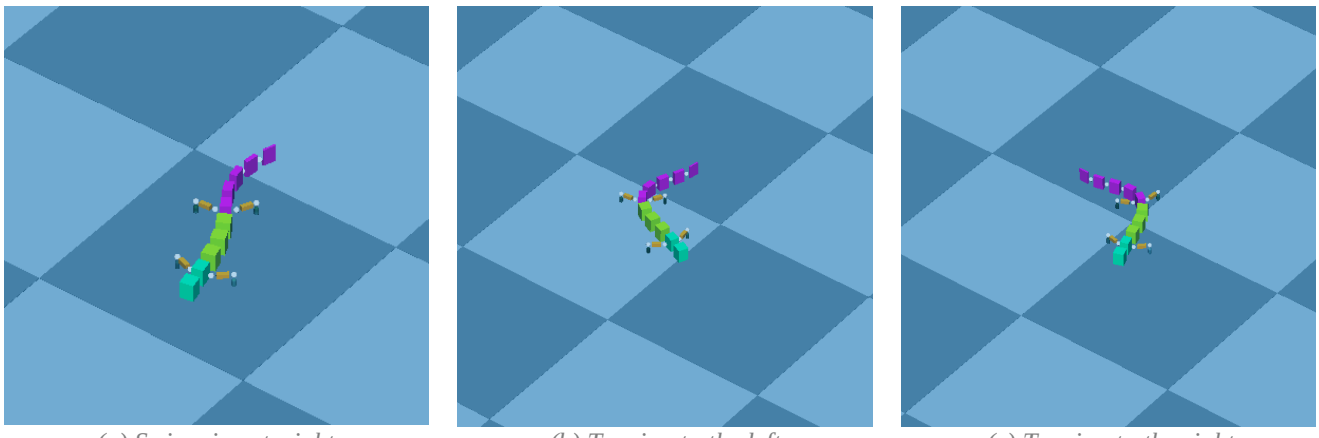
nearest reachable point that all joints can get to without inducing large, destabilizing angular velocities. In order to allow joints to achieve this goal state, we give them up to $\frac{\pi}{2}$ seconds. During this period, any joint that has reached its zero-angle has to stay there and wait for the rest to catch up. At the end of the transition time, locomotion may resume but now under the new configuration constraints provided by the new selected gait mode.

4. Results and Discussion

We have developed a controllable biomechanical model using the ODE library to handle the physics of collisions and force computations in the virtual world. The whole simulation has been written in C++ under the Microsoft Visual Studio 2010® environment, together with OpenGL® to render the salamander and its surroundings. We have created a console interface through which the user can give different commands to our model such as turning to the left or to the right, moving straight, toggling locomotion mode from trotting to swimming, changing the skin, and moving the camera with the arrow keys to track the salamander as it moves around.

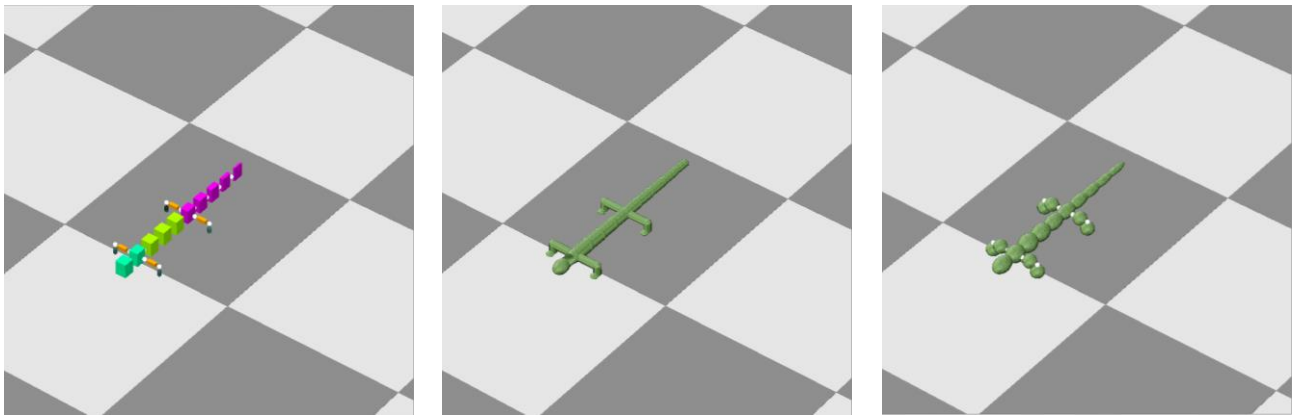
As indicated in previous section, the biomechanical model is capable of trotting in any desired direction. Figure 6 shows the salamander trotting straight (a), turning left (b), and turning right (c). In particular, making the creature turn while it was trotting posed a true challenge. In our early approach we got lots of "breakages" on the salamander's trunk because we were not able to come up with an appropriate set of parameters for the maximum allowed joint forces, f_{max}, and the friction coefficients responsible for locomotion on land. Initially, we were dealing with the same f_{max} for all of the non-leg hinges because we thought that such an idea might save us time spent towards tuning parameters. However, right after we gave joints individual maximum forces and made trunk joints stronger than the rest of the body, the salamander was able to smoothly turn concurrently and it continued to bend its body in the natural S-shape.

Also, our salamander can perform the same kind of trotting maneuvers in swimming mode. In section 3.3 we already explained that locomotion in an aquatic environment mostly follows the swimming pattern of a lamprey. On this mode, the head moves the least while the



(a) Swimming straight *(b) Turning to the left* *(c) Turning to the right*

***Figure 7**, Lisa in swimming locomotion mode.*



(a) ODE primitive geometries *(b) Textured cylindrical skin* *(c) Textured spherical skin*

***Figure 8**, Lisa's various skins.*

tail sweeps a wider arch providing thrust in the forward direction. Surprisingly, contrary to what we thought at the beginning, making our creature swim was less complex than making it trot. Furthermore, locomotion in our virtual aquatic means was even smoother and more visually attractive than trotting. Figure 7 shows the biomechanical model swimming straight, to the left, and to the right. The reader should be aware of the propagating sinusoidal function that appears in part (a) of the latter Figure, thus demonstrating that our motor-based salamander vividly mimics the graceful gaits of a real homologue amphibian.

Figures 6 and 7 show the biomechanical model with the actual ODE primitives on which the dynamics engine applies forces and torques and performs collision detection. We have developed another couple of "skins" illustrated in Figure 8. This demonstrates the flexibility of our model, where the dynamics body links represent the coupling with the physics of the environment while the visual part does not have to be necessarily the same as those primitive geometries. A more elaborate mesh built upon the dynamics skeleton of the creature would help increase the level of realism of the simulation.

The companion videos provide a multimedia experience with our biomechanical model in three situations: trotting, swimming, and an indecisive salamander that trots first to later delve into water and flee the scene.

5. Conclusion and Future Work

We simplified our model by moving away from the traditional spring-based model to prevent the fine tuning of many parameters. Since our model uses a motor-based approach, it allows for a higher level of control than our previously implemented spring model. Our articulated creature manages to perform all of the standard terrestrial and aquatic motions observed in a real salamander.

The next steps in our project would be to add behavior to our biomechanical model. Behaviors would include a field of view, the ability to search for food, avoid objects and recognize when it should transition between trotting and swimming. After these behaviors are implemented, we would like to provide different incentives to perform these behaviors such as hunger and curiosity. The ultimate goal of this work is to develop a physically-based animat, capable of learning and reasoning, similar to a real salamander but inside a virtual ecosystem.

6. Bibliography

Cohen, A. H. 1988. Evolution of the Vertebrate Central Pattern Generator for Locomotion. *In Neural Control of Rhythmic Movements in Vertebrates.*

Ekeberg, Ö. 1993. A Combined Neuronal and Mechanical Model of Fish Swimming. *In Biological Cybernetics 69, 363-374.*

Ermentrout, B., and Kopell, N. 1994. Inhibition-Produced Patterning in Chains of Coupled Nonlinear Oscillators. *In SIAM Journal of Applied Mathematics, 54(2):478-507.*

Frolich, L. M., and Biewener, A. A. 1991. Kinematic and Electromyographic Analysis of the Functional Role of the Body Axis during Terrestrial and Aquatic Locomotion in the Salamander Ambystoma Tigrinum. *In Journal of Experimental biology, 62:107-130.*

Harischandra, N.; Cabelguen, J.; and Ekeberg Ö. 2010. A 3D Musculo-Mechanical Model of the Salamander for the Study of Different Gaits and Modes of Locomotion. *In Frontiers in Neurorobotics, Volume 4, Article 112.*

Ijspeert, A. J.; Hallam, J.; and Willshaw, D. 1998. From Lampreys to Salamanders: Evolving Neural Controllers for Swimming and Walking. *In From Animals to Animats, Proceedings of the Fifth International Conference of the Society of Adaptive Behavior (SAB '98): 390-399.*

IJSPEERT, A. 2000. A 3-D Biomechanical Model of the Salamander. *In Proceedings of the Second International Conference on Virtual Worlds, ed. J Heudin, London: Springer, 225-234.*

SMITH, R. 2007. Open Dynamics Engine (ODE). *Available at http://www.ode.org/ as of January 21, 2014.*

TERZOPOULOS, D.; TU, X.; AND GRZESZCZUK, R. 1994. Artificial Fishes: Autonomous Locomotion, Perception, Behavior, and Learning in a Simulated Physical World. *In Artificial Life 1(4): 327-351.*