Bio-sensory motor systems

Raphael Cherney & Frédéric Wilhelm

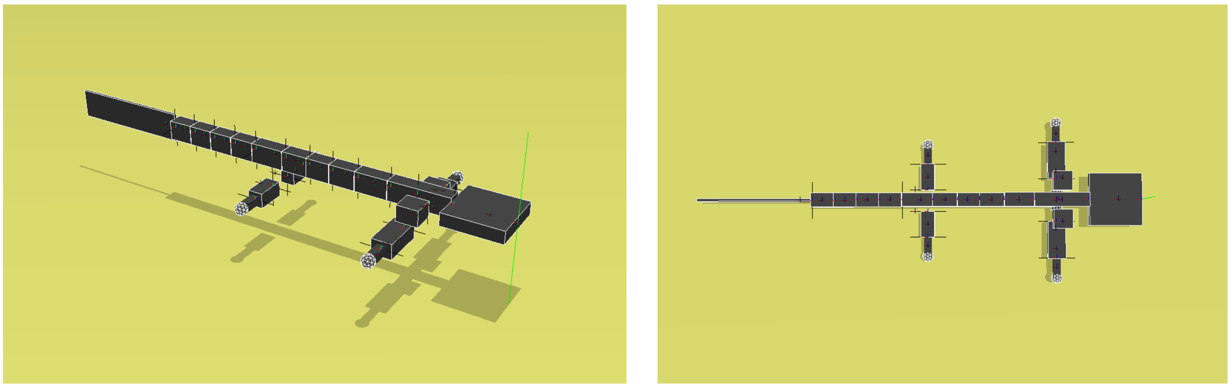
January 10, 2012

Abstract

In this paper, we investigate the locomotion and control of a 23 degree of freedom simulated salamander robot. We examine different the parameters controlling both walking and swimming using a sine-based controller and a central pattern generator (CPG). We find that, when optimizing the average speed through stematic tests and particle swarm optimization, the optimal gait and swimming trajectories are similar to the movement of actual salamanders. Finally, we improve our salamander by adding stereovision and model the biological vision system with a neural network so that the salamander is capable of reaching an object.

1 Robotic salamander model

All of the experiments descibred in this report were done in simulation using the Webots software by Cyberbotics Ltd. [5]. The software encorporates an integrated development environment (IDE), physics engine, and 3-dimentional graphics engine. Running tests in simulation has several advantages when compared to physical experiments: hardware prototyping is time consuming, commercial robots are expensive and fragile, simulation allows for rapid changing of the experimental setup, measuring physical quantities is easier in a virtual environment, and tests can be performed in faster than real-time. The salamander robot model we used has 23 active degrees of freedom: 11 along the spine, 3 for each hind limb, and 3 for each forelimb (Figure 1). The model is based on an actual, 26 degree of freedom robot in the Biorobotics Laboratory of the École Polytechnique Fédérale de Lausanne (EPFL). It was designed based on 3-dimentional locomotion data recorded of a live salamander so that it could accurately reproduce the swimming and walking motion of a biological speciemen. The similarity between the biological, physical, and simulated salamanders means that the simulation results can have both physical and biological significance. That is to say, our exploration of locomotion parameters in simulation can be used to create an effective controller for the pysical robot and/or gain insight into the biological workings of live salamanders. In the following sections, we examine the parameters associated with locomotion, implement a more advanced and robust controller, and ultimately create a simulated salamander with a biologically-inspired vision and locomotion. Along the way, we compare our results to biological data and try to extract meaning from the similarities and differences.



(a) Perspective view

(b) Top view

Figure 1: Webots model of salamander. The black lines indicate the servo axes of rotation.

2 Sine-based controller

The goal of this first section is to explore and optimize the different parameters that control salamander motion. We begin by noticing that, in animal locamotion, the trajectories of joint angles are continuous and periodic. That is so say that transitions are smooth (not jerky) and repetative. Furthermore, we observe that the overall gait (e.g. walk, trot, gallop) is periodic such that individual joint trajectories can simply be defined by a relationship to a common periodic signal. From these observations, we can create a simple sine-based controller in which each joint angle is calculated by applying a particular phase shift, scaling factor, and offset to a common sine wave signal. Such a sine-based controller can be described by:

$$\theta_i = A_i \sin\left(\omega_i t + \phi_i\right) + \overline{\theta}_i \tag{1}$$

where θ_i is the desired joint angle for motor i, A_i is the amplitude, ω_i is the frequency, ϕ_i is the phase lag, and $\overline{\theta}_i$ is the offset. Given this simple controller, we can easily implement relatively complex motions in the body by defining the relationship of each joint to the central frequency. Unfortunately, while this model can very easily syncronize motion between limbs, it has poor biological meaning (as all relative motion is strictly defined), and needs a completely different model for swimming and walking. Nevertheless, it is useful for evaluating and optimizing externally visible parameters such as the amplitude, frequency, offset, and phase lag that still come into play with more advanced controllers.

2.1 Swimming

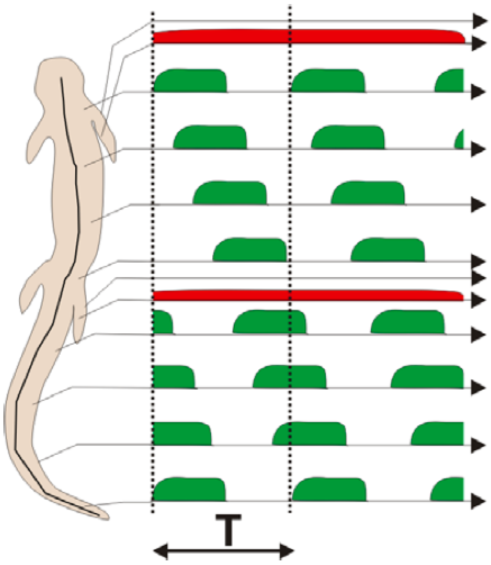


Figure 2: Traveling wave in axial muscles of swimming salamander (Ijspeert ?).

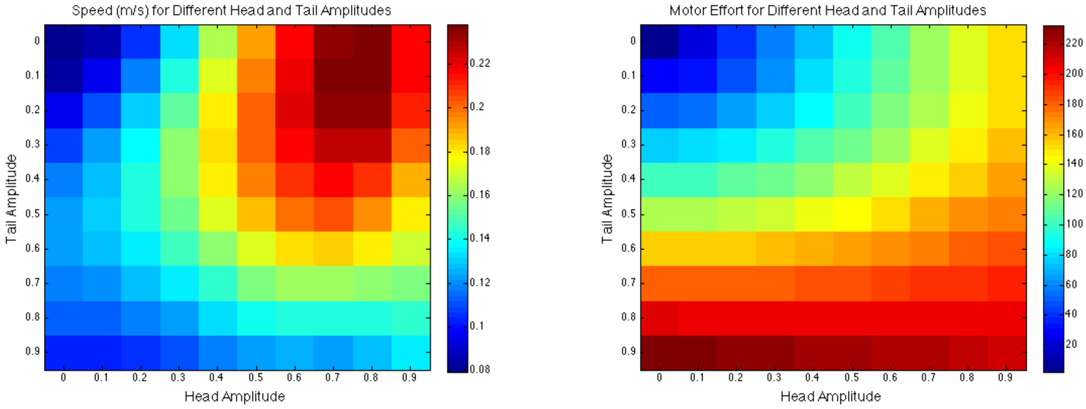
We begin by looking at the salamender swimming motion. It is relatively simple and is dominated by a traveling wave down the spine – much like a lapray (Figure 2). Based on the scientific data presented in ???, we assume a constant phase lag between body segments (a good assumption except near the pectoral and pelvic girdle [2]). Given N segments and uniformly increasing phase lags $\phi_{i+1} - \phi_i = \Delta\phi$, we find that the head-tail phase lag is $N\Delta\phi$. One cycle is 2π so the number of cycles on the body is $k = \frac{N\Delta\phi}{2\pi}$, known as the wave number. Knowing that salamanders typically have a wave number of $k = 1$, we calculate the phase lag between segments $\Delta\phi = \frac{2\pi k}{N} = \frac{2\pi}{11} \approx 32.72°$. Each segment has a slightly greater phase lag than the pervious one giving $\phi_i = i * \Delta\phi$. We also observe that the amplitude of oscilation tends to increase from head to tail. In order to test the implications of this observation, we implemented a linearly increasing amplitude from head to tail with an maximum value of A_{tail} at the last segment (and a corresponding linearly increasing aplitude from tail to head with a maximum value of A_{head}). By adjusting the relative tail versus head movement, we can test whether a constant, increaing, or decreasing amplitude along the body is better.

2.1.1 Testing

We performed a series of systematic tests, modifying two parameters at a time to see how they affected the quality of locomotion. As a measure of the quality of locomotion motion, we meaured the average speed of the robot during each simulation run ($speed = \frac{distance\,from\,starting\,point}{simulation\,time}$) and the motor effort (as measured by the integral of the torques applied by the simulated motors). Based on our observations, we had several hypotheses we hoped to answer:

- To what extent can a simple sine-based controller recreate salamander swimming motion?
- Is having a larger amplitude toward the tail benefitial for motion?

A simple implementation of a travelling wave along the spine with a wave number of one reveals that we can, quite effectively, create at least a high-level approximation of salamander swimming motion using sine-based controllers. The motion we observe is both effective and visibly similar to that of biological specimens. We also notice that a higher amplitude at the tail end of the body was more effective. Figure 3 shows the speed and motor effort measurements for one of our systematic tests. We see that the optimal speed is achieved with essentially no movement of the head and an increaing amplitude as we approach the last segment. We also observe that the amount of energy required is greater for motion of the head than motion of the tail. In fact, the most energy intensive motion is moving the front of the salamander while keeping the tail rigid. We hypothesize that this is due to the large size of the head and increased effort it takes to move it in such an inefficient way (the tail is almost working against the fore-body segments).

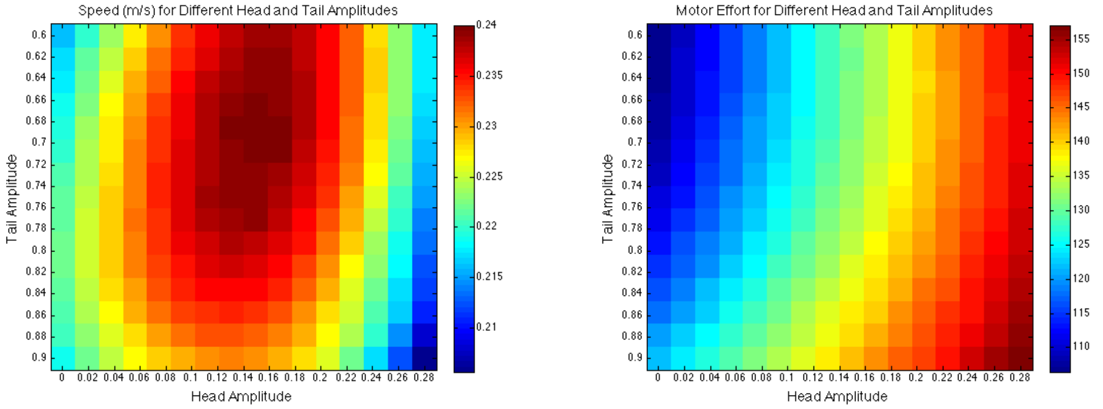


(a) Speed

(b) Motor effort

Figure 3: Systematic tests reveal that lower head movements are advantageous (for both speed and energy consumption) and that the optimal speed uses relatively large tail motion

We suspected that salamander motion is optimize for speed and efficiency, so we attempted to optimized the parameters of our sine based model before making comparisons. In particular, we optimized for the average speed of travel given a contant frequency. Figure 4 shows the results of this systematic test. We find that the optimal speed for a frequency of $\omega = 0.5$ Hz is 0.2401 m/s with $A_{head} = 0.08$ and $A_{tail} = 0.76$. These values have little biological corresponence other than to tell us that an increasing amplitude from head to tail (with only minimal head movement) is optimal.



(a) Speed

(b) Motor effort

Figure 4: Fine-grain search to optimize speed.

2.1.2 Biological comparison

The finding that the optimal speed occurs with increasing amplitudes along the spine corresponds to the observed motion of actual swimming salamanders. Figure 5 shows video capture of real salamander swimming and the simulated salamander (parameters optimized for speed). Overall, the two match quite well, but there is slightly more head motion in the real salamander and the real tail is more flexible, allowing for more efficient thrust.

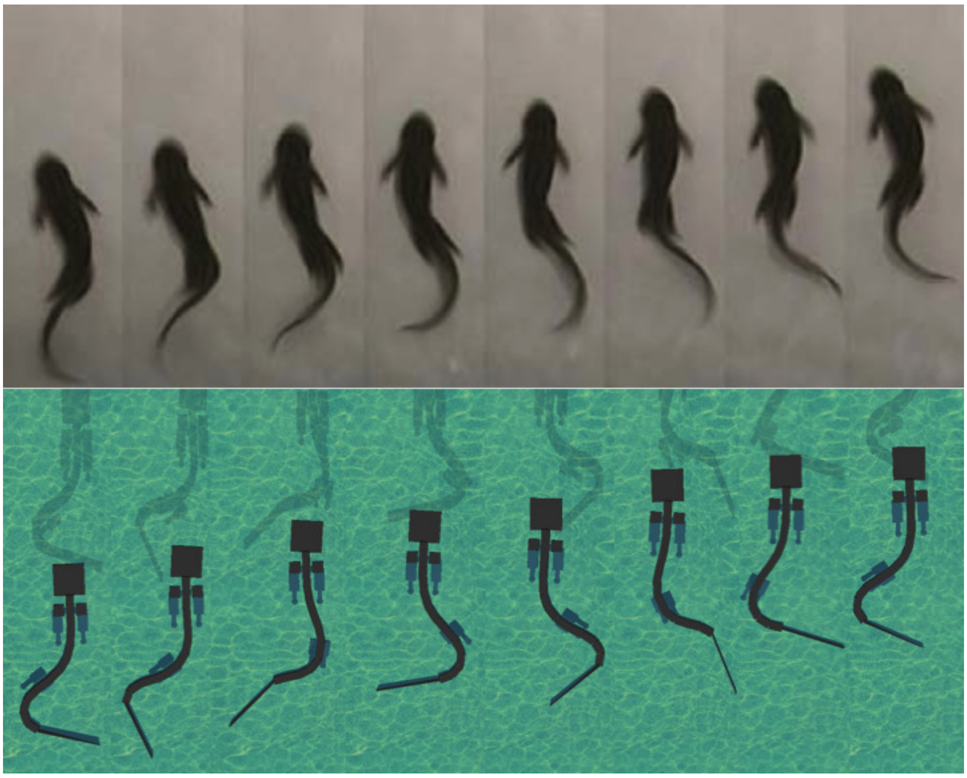


Figure 5: Swimming mode of real salamander (top) and simulated salamander (bottom).

2.2 Walking

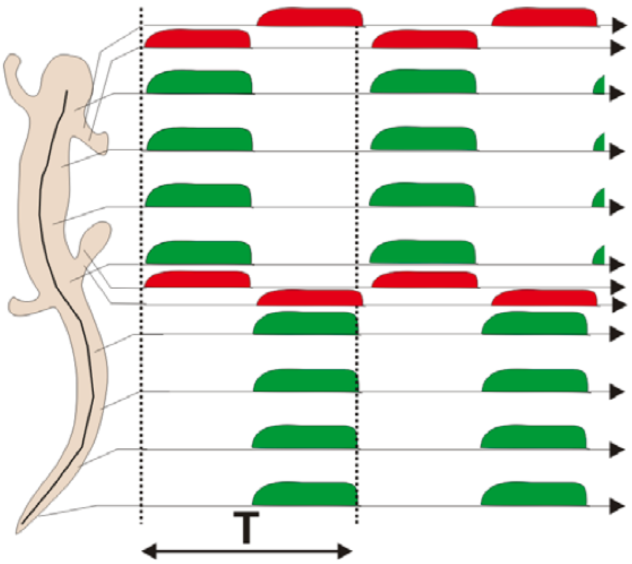


Figure 6: Standing wave signal in muscles of walking salamander (Ijspeert ?).

The walking motion of salamanders is quite a bit more complex than swimming. Modeling the coordination of the legs can quickly become quite complicated depending on the different gaits. Nevertheless, a sine-based controller can effectively synchonize the motion throughout the step cycle to achieve effective, life-like walking. This basic motion, as measured in actual salamanders is acieved by having diagonal limbs move synchronously together along some stepping trajectory and lateral limbs working in anti-phase (phase lag $\phi = \pi$). Implementing a standing wave of the body allowing for greater reach of the legs and can greatly increase the speed of motion. Figure 6 shows the signal activity that controls the contraction of the muscles that create this standing wave (and walking motion) in actual salamanders. There are many parameters that control the coordination of joints to achieve walking, even with a simplistic and relatively ridged sine-based controller. Instead of arbitrarily changing parameters with little biological significance, we wanted to answer the following questions through out experimentation:

- To what extent can a simple sine-based controller recreate the salamander walking motion?
- To what extent is the walking motion of salamanders optimized?

2.2.1 Initial testing

We began by implementing a simple sine based controller to control the walking motion. The spine segments all move with an common amplidue of A and a standing wave is created by introducing phase lag of $\phi_i = \pi$ for the body segments between the legs (i=1 through 5). The legs themselves each follow a half-elipse trajectory centered around a point at a position away from the body. The duty cycle (amount of time with the leg on the ground throughout the period) and phase lag of these leg trajectories can also be adjusted. After performing several systematic tests we achieved a reliable and life-like walking trot gait.

2.2.2 Particle swarm optimization (PSO)

In order to determine whether or not the walking motion of salamanders is optimized, we decided to optimize our simulated salamander and then compare the results to biological data. Due to the size and interdependence of the search space, rather than perform a series of systematic tests, we opted to use a more advanced and applicable optimization method: particle swarm optimization, or PSO. PSO is an iterative algorithm that attempts to optimize a fitness function by flying particles through the search space with their position representing a candidate solution. Each iteration a particle updates its position based on its velocity, the best candidate solution found by the particle, and the best solution found by any particle (global neighborhood, or GBEST). We used an 8-dimensional search space with the dimensions being the duty cycle of each leg (4), the phase lag relative to one leg (3), and the amplitude of the spine oscilation (1). We attempted to optimize locomotion as measured by the average speed. The position of the ith particle in 8-dimentional space is represented as $\mathbf{x}_i = (x_{i1}, x_{i2}, ..., x_{i8})$ and its velocity is given by $\mathbf{v}_i = (v_{i1}, v_{i2}, ..., v_{i8})$ (both of which are randomly initialized). Each iteration we update the position and velocity as decribed by the following equations:

$$\mathbf{v}_i = c_3 \left(\mathbf{v}_i + c_1 * \text{rand}() * (\mathbf{p}_i - \mathbf{x}_i) + c_2 * \text{rand}() * (\mathbf{p}_g - \mathbf{x}_i)\right) \tag{2}$$

$$\mathbf{x}_i = \mathbf{x}_i + \mathbf{v}_i \tag{3}$$

where rand() is a random number between 0 and 1, $\mathbf{p}_i$ is the position of the best solution found by particle i, $\mathbf{p}_g$ is the position of the global best solution, c_1 is the cognitive factor (attraction to personal best solution), c_2 is the social factor (attraction to global best solution), and c_3 is the constriction factor ($c_3 < 1$) which causes the particles to slow over time and perform more localized searching.

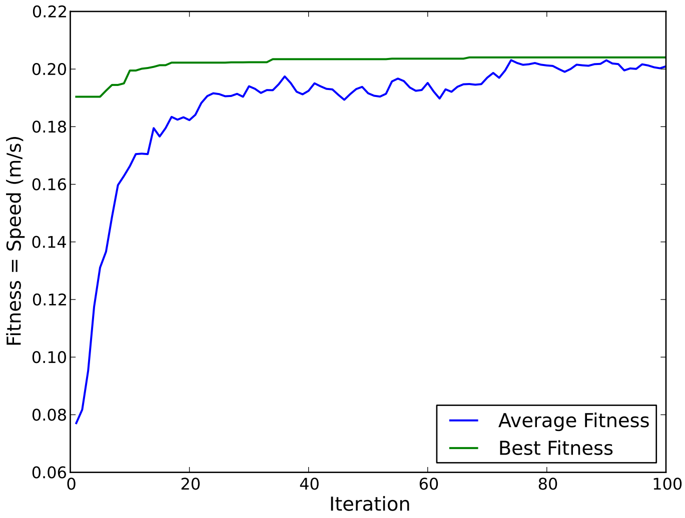


Figure 7: The particle swarm optimisation quickly maximizes the robot's walking speed.

Figure 7 shows the average and best fitness (average speed) results from our particle swarm optimization over the 100 iterations (20 particles, 10 second simulation, $\omega = 0.5$Hz, $c_1 = 1$, $c_2 = 1$, $c_3 = 0.5$). The method is able to quite quickly optimize the parameters to maximize our fitness function (good global best solution after only 10 iterations). We found a maximum speed of 0.204 m/s, which is almost exactly the same as our swimming velocity for the same frequency. We then took these optimized parameters and compared the result to biological data.

2.2.3 Biological comparison

When we compare our optimized sine-based controller to biological data, the results are quite surprising. Qualitatively, we see many similarities between our evolved controller and actual salamander locomotion. They both use a walking trot gait with similar duty cycles and phase lags. They also travel approximately the same number of body lengths per cycle. Figure 8 shows both the a real salamander motion and our simulated sine-based using speed-optimized parameters.

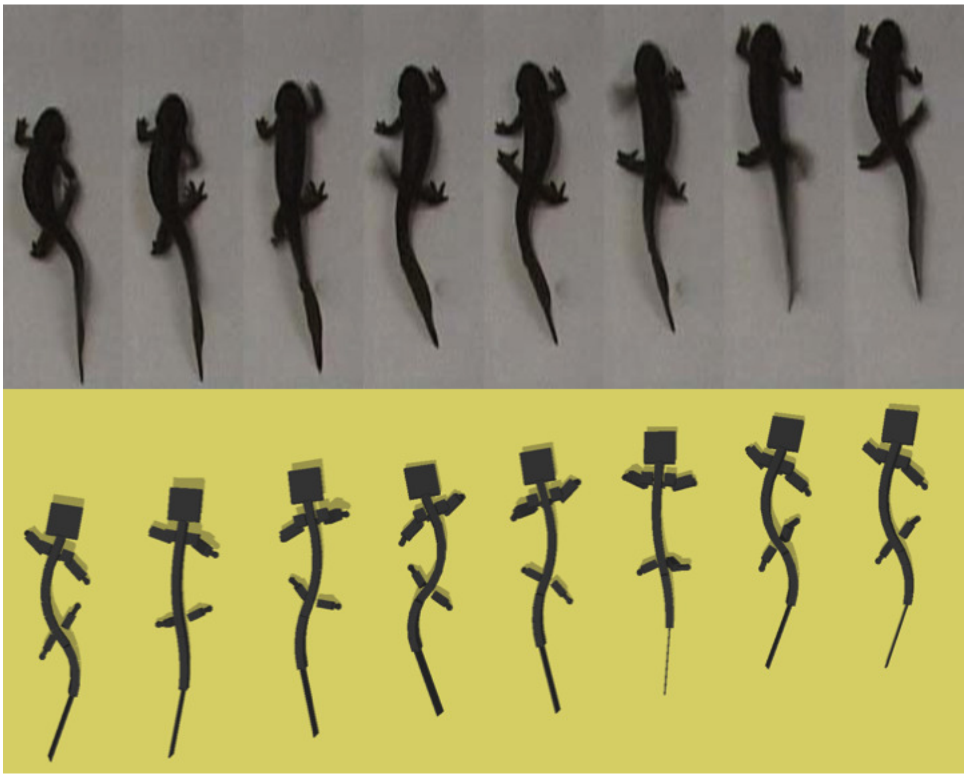
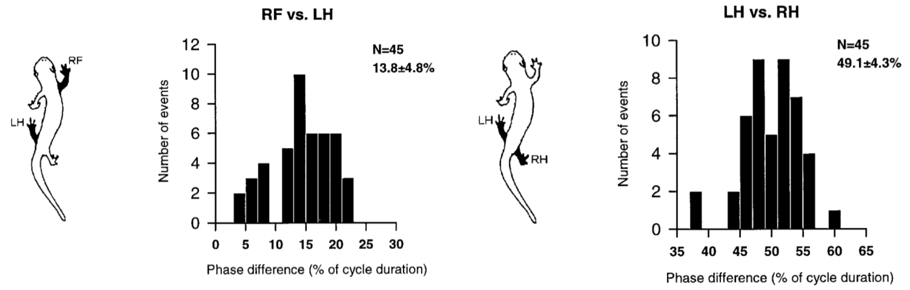


Figure 8: Walking mode of real salamander (top) and simulated salamander (bottom).

Figure 9 shows biological data from [2] showing the phase difference between diagonal (RF and LH) and lateral (LH and RH) limbs. This data shows that diagonal limbs typically have a phase difference of 13.8±4.8% when walking. The phase difference between the RF and LH limbs of our optimized controller is $\frac{3.337847-2.606619}{2\pi} = 11.6\%$. For lateral coordination, we find that the phase difference between the LH and RH limbs is a little off from the biological value of 49.1±4.3% at 36.1% (likely due to differences in the duty cycles). Nevertheless, we see that the model is able to capture some of the more intricate relationships between limbs. In particular, if optimizing for speed results in similar phase lags to what we see in nature, this could suggest that the reason salamanders walk the way they do is to maximumize their speed (and in turn increase their fitness).



(a) Diagonal coordination

(b) Lateral coordination

Figure 9: Interlimb coordination during overground stepping, showing percent phase difference between limbs [2].

3 CPG controller

In this second part, we use a Central Pattern Generator to control the gait of our robot. CPGs are a bio-inspired way to generate rythmic patterned outputs. We try here to investigate the ability of the CPG to describe and reproduce the gait of the real salamander and compare it with the previous results of the sine-based controller. Then, we try to use the CPG to describe the transition between swimming and walking with a simple model.

3.1 Model

Our model is inpired from [4] and is composed of a body CPG (with only one oscillator per degree of freedomunlike two in [4]) and a limb CPG implemented as a system of coupled nonlinear phase oscillators with controlled amplitude:

$$\begin{aligned} \dot{\theta}_i &= 2\pi\omega_i + \sum_{j\neq i} r_j w_{ij} \sin(\theta_j - \theta_i - \phi_{ij}) \\ \dot{r_i} &= a_i(R_i - r_i) \\ x_i &= X_i + r_i \cos(\theta_i) \end{aligned} \tag{4}$$

For simplicity, we set the same frequency ω and amplitude convergence rate a for all oscillators. We also restrict to symmetric coupling (a biologically coherent hypothesis), i.e. $\phi_{ij} = -\phi_{ji}$ for all couple i, j of oscillators. According to this model, the oscillators should synchronize and the phase differences $\theta_j - \theta_i$ should converge toward the defined phase bias ϕ_{ij}. The amplitude r_i should converge toward R_i.

We use the hypothesis described in [4] in our model:

- **Hypothesis 1:** The body CPG spontaneously produces travelling wave when activated with a tonic drive.
- **Hypothesis 2:** The strengths of the couplings from limb to body oscillators are stronger than those from body to body oscillators and from body to limb oscillators. Thus, the limb CPGs override the natural tendency of the body to produce travelling waves.

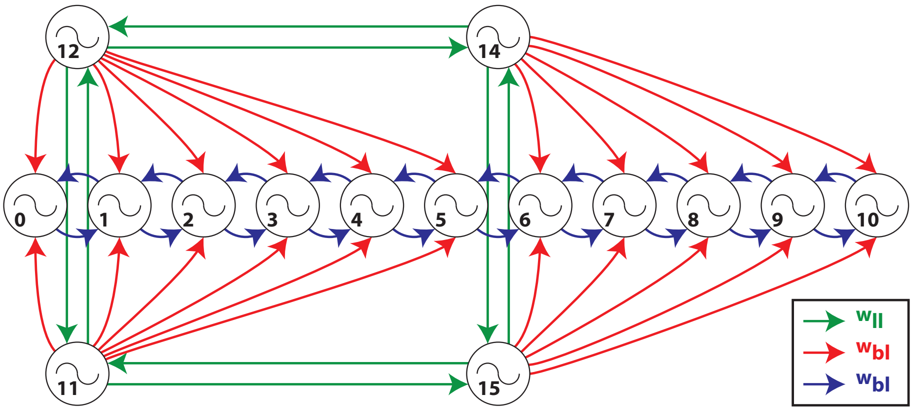


Figure 10: Configuration of the CPG model, and notations for the coupling strength (see 3.3).

- **Hypothesis 3:** Limb oscillators stops oscillate above a frequency threshold. Thus, they do not override the limb CPG and the salamander is switching to a swimming gait.
- **Hypothesis 4:** Limb oscillators have lower intrinsic frequency than the body oscillators. Therefore, the frequency increase rapidly when switching from walking to swimming.

For simplicity, we will not use this last hypothesis, and we just consider that the limbs stop oscillating above a certain frequency threshold.

The configuration of our CPG model is illustrated in Figure 10. As shown, it consists of a chain of 11 oscillators driving the spine motors and 4 oscillators for the limbs.

In accordance with our model, we use a significantly lower coupling strength between the body oscillators than with the limbs. The coupling phase (i.e., the phase lag after convergence) between neighboring body oscillators is constant so as to generate a travelling wave, and, throught equation 5 is directly linked to the **wave number**[1] k and to the number $\tilde{N}$ of body oscillators (11 in our case).

$$\phi_{i-1,i} = \frac{2\pi\tilde{N}}{k}, i = 2..\tilde{N} \tag{5}$$

The following plot show the two behaviours of the CPG. We tested it using Matlab®, with a wave number of 1, coupling strenth of 0.5, 5 and 2 for the coupling body/body, body/limb and limb/limb respectively. The frequency threshold for limb oscillation was set to $\omega_T = 1.2$. Thus, we tested walking with $\omega = 1$ and swimming with $\omega = 1.4$.

[1] the number of waves on the body at a given time

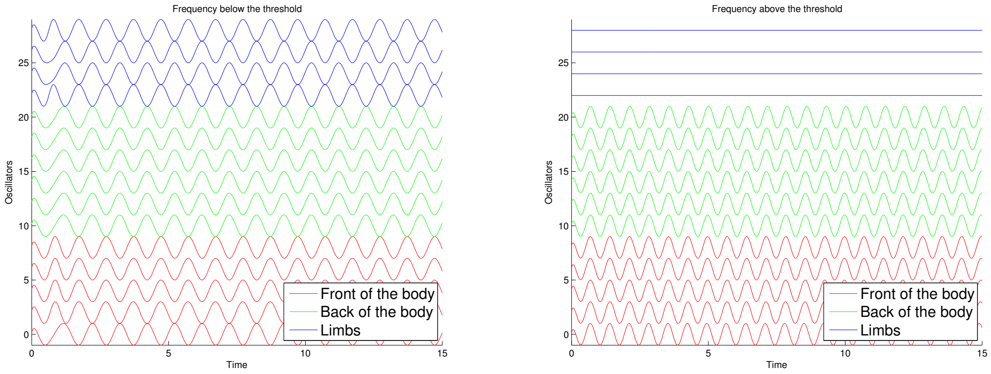


Figure 11: *Left:* Walking gait ($\omega < \omega_T$). *Right:* Swimming gait ($\omega > \omega_T$).

On the first figure, we see that each half of the body are in phase, and the two parts are in opposite phase. We thus obtain our waking gait. On the second figure, the limbs have stopped oscillating, and after a transient state we have a constant phase lag withing neighboring spine oscillators, which account for the travelling wave of the swimming gait.

We are also able to study the transition between the two gait regimes, by linearly increasing ω from 1.0 to 1.4. On Figure 12, we see that the CPG oscillate with a walking regime during the first half of simulation time, with an slightly increasing frequency, then jumps to swimming gait after a short transient state.

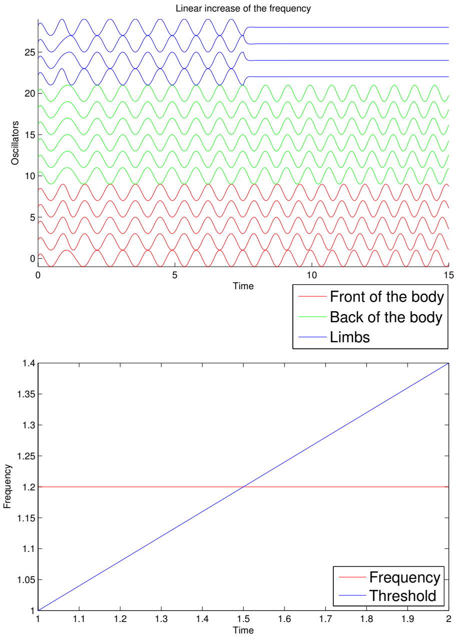


Figure 12: CPG behaviour when we increase the frequency. When it reaches the threshold, the limb CPG stops oscillating (after a transient state) and the body CPG converges to a travelling wave.

3.2 Study of the dynamical system

The CPG model is a dynamical system, but the high number of variables makes it difficult to study formally. In order to study the dynamical system, we will first focus on a more simple model with just two oscillators 1 and 2, and the same notations as previously.

Let us note $\theta = \theta_2 - \theta_1$, $\phi = \phi_{2,1}$ and $\gamma = r_1 w_{2,1} + r_2 w_{1,2}$. We then have

$$\dot{\theta} = -\gamma sin(\theta - \phi),$$

which is a one-variable dynamical system, whose phase diagram is just a sine function plotted on Figure 13.

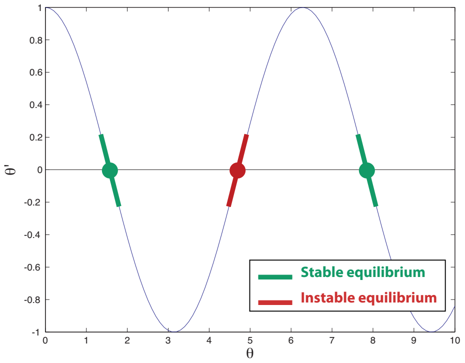


Figure 13: Phase diagram

Points of equilibrium are found at $\dot{\theta} = 0$, which leads to $\theta = \phi + k\pi$. For $\theta = \phi + 2k\pi$, the equilibrium is stable since $\frac{d\dot{\theta}}{d\theta} = -\gamma < 0$, but it is instable for $\theta = \phi + (2k+1)\pi$ since $\frac{d\dot{\theta}}{d\theta} = \gamma > 0$. This means our CPG is in instable equilibrium when some couple of oscillators i, j have phase differences opposite to the defined phase bias $\phi_{j,i}$ (i.e. of $\phi_{j,i} + \pi$).

In the plots of Figure 11, we initialized the phase states θ_i randomly close to zero, in order to take into account the limited precision of the motors. If we initialize them to 0 (exactly), we obtain the plot on Figure 14 for the walking gait. On the first half, all the limbs are in phase instead of being in opposite phase for neighboring limbs. This is an instable equilibrium since, for instance, the two fore limbs have a phase bias of 0 instead of π. At the half of the simulation, we randomly perturbate the phases θ_i, and the system goes to stable equilibrium.

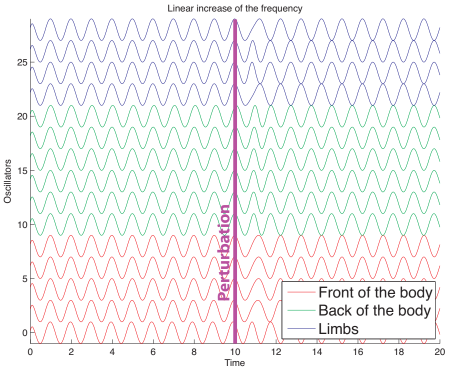


Figure 14: Instable configuration in walking mode. After a slight perturbation, the CPG goes to stable equilibrium.

The transient phase between the two regimes is directly linked with the convergence rate a, which determines how fast the amplitude of the limbs will converge to zero when switching to swimming gait. On Figure 15, we plot the phase differences between the different oscillators and the first oscillator (the head), in order to compare the transition between the two gaits. In the first half (walking), the phase difference are the same within the front half of the body and within the back half (standing waves).After the transition, the CPG converges to a state when the phase differences are linearly increasing along the body (travelling waves). We see the transition becomes more abrupt if we increase the factor a from 0.2 to 2, but we do not see any visible change from 2 to 10. This is because the transition is now limited by the coupling strength w_{ij}, which determines how fast the phases θ_i will update.

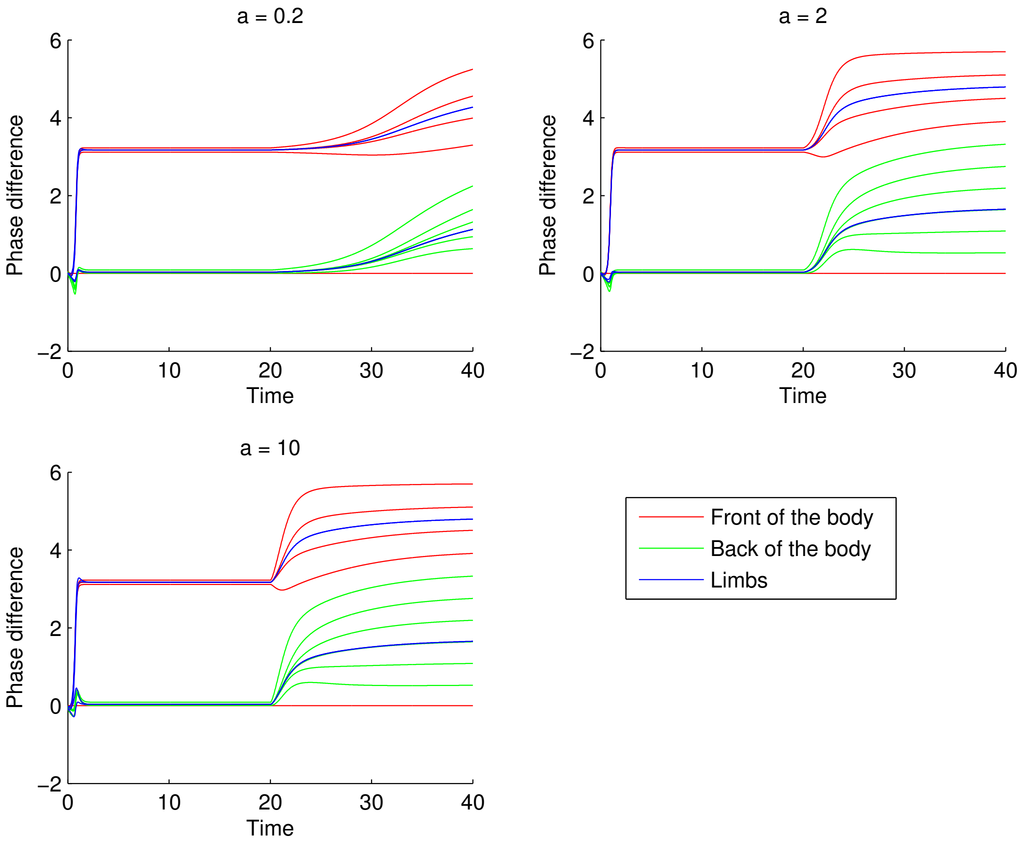


Figure 15: Phases differences with the first oscillator (the head of the salamander). we see the transition between the walking gait (standing waves ; same phase difference within each half of the body) and the swimming gait (travelling waves ; regularly increasing phase differences along the chain of the body oscillators).

3.3 Salamander robot behaviour

We now try to address the question of the coupling between the CPG oscillators. Our systematic analysis in Part 2 gives us good parameters for the amplitude, offset and coupling phase of the different oscillators. Starting with this, and as discussed previously, the coupling strength and the amplitude convergence rate will influence the transition between the two gaits. Our goal is to study the effect of these parameters on the transition using our simulation in Webots®.

We therefore focus our study on three parameters:

1. The coupling strength between the segments of the body w_{bb},
2. The coupling strength of the body with the limbs w_{bl},
3. The coupling strength between the different limbs w_{ll}.

Using Figure 10, w_{bb} accounts for the coupling strength $w_{0,1}$, $w_{1,2}$, $\cdots$, $w_{9,10}$ and their symmetrical, w_{bl} accounts for $w_{i,11}$, $w_{i,12}$, $w_{i,13}$, $w_{i,14}$ where $i = 0 \cdots 10$, and w_{ll} for $w_{11,12}$, $w_{12,13}$, $w_{13,14}$, $w_{14,11}$ and their symmetrical.

We first tested the influence of the coupling strength within the limbs, w_{ll} on the walking gait (this influence, of course, is null on the swimming gait, since the limbs are deactivated). On Figure 16, we notice that w_{ll} does not influence the gait, at least in the long term, since the speed and the stability remains almost constant. The measure were done using our Webots®simulation, with $w_{bb} = 0.5$, $w_{bl} = 5$, and a frequency below the threshold so that we obtain a walking gait. This can be explained by the fact that the limbs are not influenced by the rest of the body. Thus, the limb CPG will converge toward the same equilibrium regardless of w_{ll}, which only influence the convergence rate. The rest of the body will synchronize with the limbs thanks to w_{bl}, also regardless of w_{ll}.

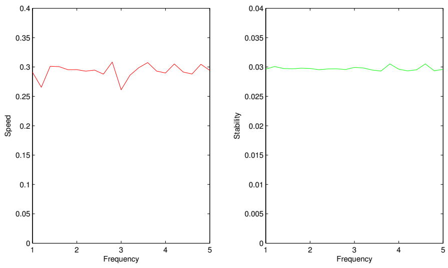


Figure 16: Influence of w_{ll} on the speed and the stability of the walking salamander.

The same is obtained with the swimming gait and the coupling within the body, w_{bb} (Figure 17). The coupling strength do not change the equilibrium position, but only changes the convergence rate toward this equilibrium.

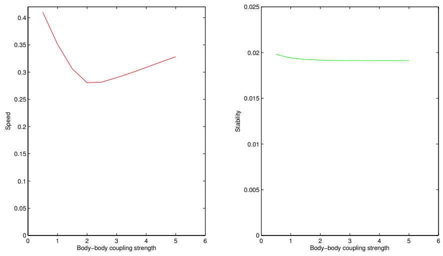


Figure 17: Influence of w_{bb} on the speed and the stability of the walking salamander.

Our following step is to conduct a systematic analysis of the influence of w_{bl} and w_{ll} on the gait. We choose arbitrarity to set w_{ll} to 5, which is big enough to avoid a long transient state.

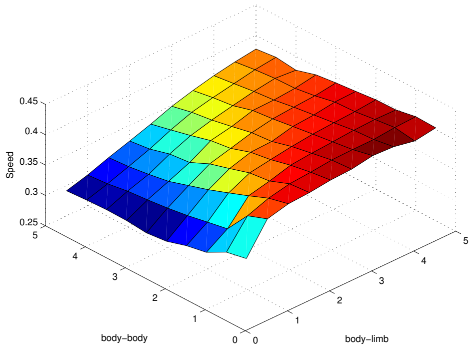


Figure 18: Influence of w_{bl} and w_{bb} on the speed and the stability of the salamander. As expected, the fastest gait is when w_{bl} is maximized and w_{bb} is minimized, so that the travelling waves effect is almost cancelled by the standing waves.

3.3.1 Gait transition

We now study the transition between the walking gait and the swimming gait, that is allowed by our model. This transition is done with an increase of the frequency ω, so that the limbs' amplitude is set to zero, and the body's movement switches to travelling waves.

We thus create a Webots$^{\circledR}$environment with a sloped ground and water (Figure 19). We make our salamander walk in direction of the water. We see the transition between the two gaits on Figure 20. When the salamander enters the water, it should continue walking while the ground is high enough. When the water is deep enough, the salamander switches to travelling waves and begins swimming. Our model is very simple and the salamander does not detect the water and the deepness. Instead, the controller knows the position of the water and knows at which position is should switch between the two gaits. A future improvement would be to detect when the contact between the feet and the ground stops before switching to a swimming gait [1].

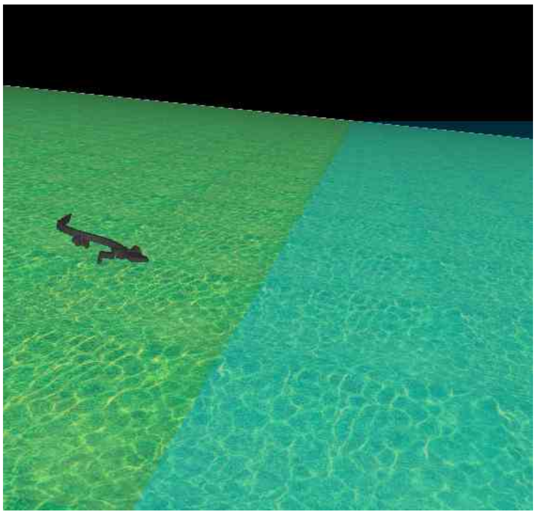


Figure 19: Sloped ground with water for the study of the gait transition

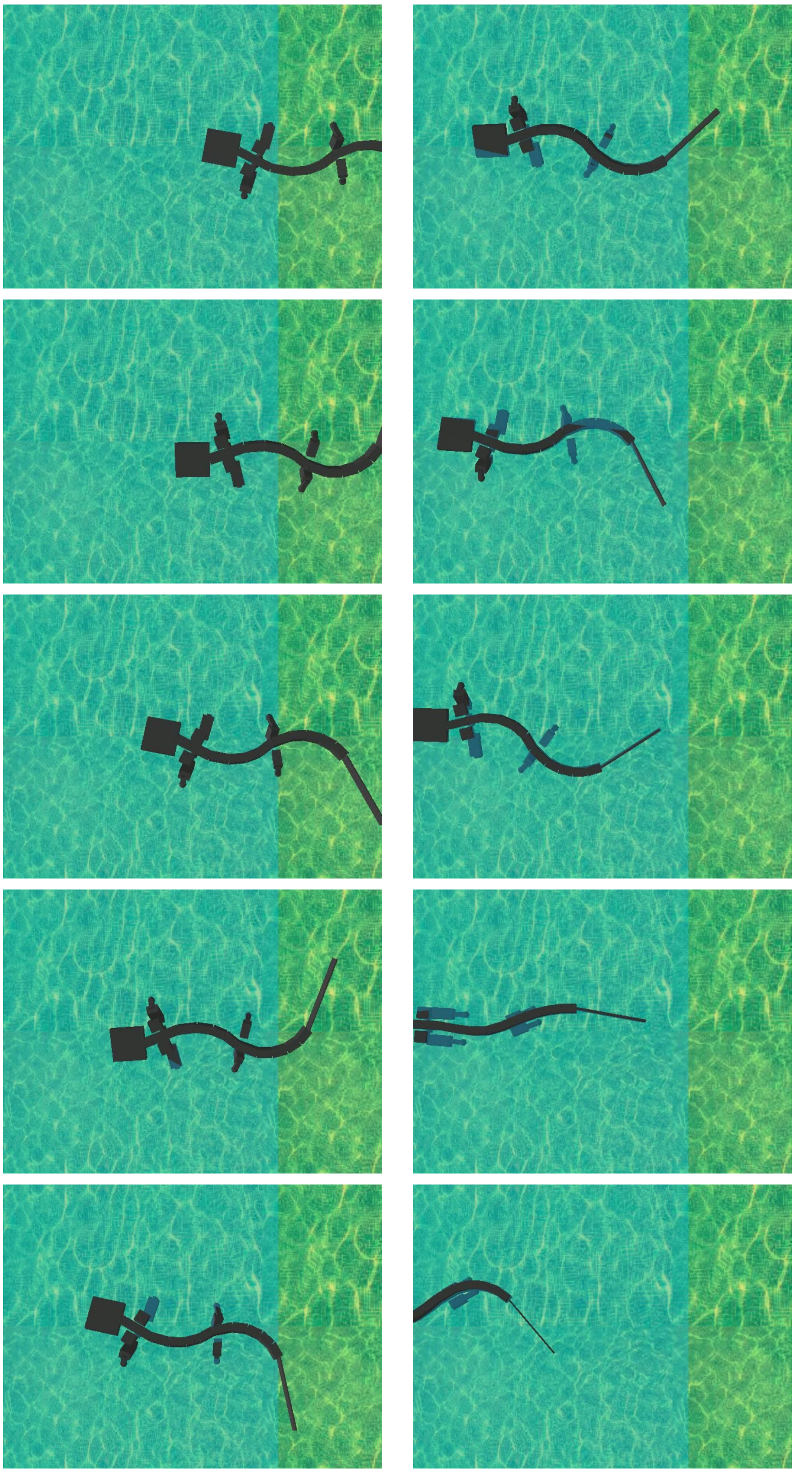


Figure 20: Transition between swimming and walking gait. For this simple model, the salamander knows the position of the surface and do not detect it.

3.4 Results and discussion

Our results with this CPG model can lead us to conclude that CPG models presents many advantages compared to sine-based controllers. The power of dynamical systems allows smooth changes of parameters, as well as good robustness toward perturbations, without jerky movements that could harm the mechanics of a real robot. The coupling parameters between the oscillators are very convenient to model several gaits and transitions within them. Finally, This model has the advantages of being close to biological mechanism that were designed by millons of years of evolution. To mention some disadvantages of CPG models, let us notice that parameters are sometimes difficult to choose, and require simulation or experiments. The use of optimization methods (like PSO) or evolutionary algorithms could be used to optimize them.

4 Adding vision

4.1 Model of the salamander visual system

The final aspect of salamander locomotion that we chose to investivate was adding a higher level of control by implementing a simple vision system. The model that we implemented is based on unpublished work by Petreska and Ijspeert and on [3]. It uses disparity in the left and right visual fields to track and move towards a target. For this part, we added two camera sensors to the head to act as the eyes (retina) of the salamander. In the model, we assume that the retinal ganglion cells have a square receptive field and are only sensitive to changes in the visual field (implemented as the difference in intensity between two time steps).

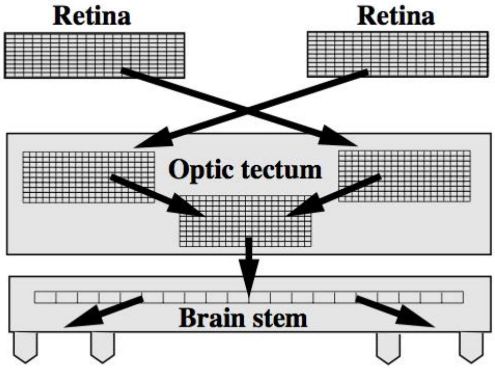


Figure 21: Visual system of the salamander. The tectum cells and the brainstem are modelled as an array of neurons.

Our salamander uses two eyes, each with a 64x64 array of cells. The signals from these cells are passed to a 32x32 array of tectum cells, (Figure 21). Each tectum cell is a neuron that takes a 3x3 array of retinal inputs and effectively averages the values, creating a more compact representation of what the eye is seeing. The brainstem then computes the absolute difference between the left and right optic tectum maps, providing some measure of depth. Finally, based on the total activity on each half of the brainstem, the brain outputs a tonic input (offset) to the spinal chord which causes it to move toward areas with high level of percieved the motion (as is caused by objects in the visual field).

The direction of movement of the salamander is controlled by the offsets X_i (see equation 4) for the body segments. It allows to generate an asymetrical gait which changes the direction of walking of the salamander.

4.2 Results and discussion

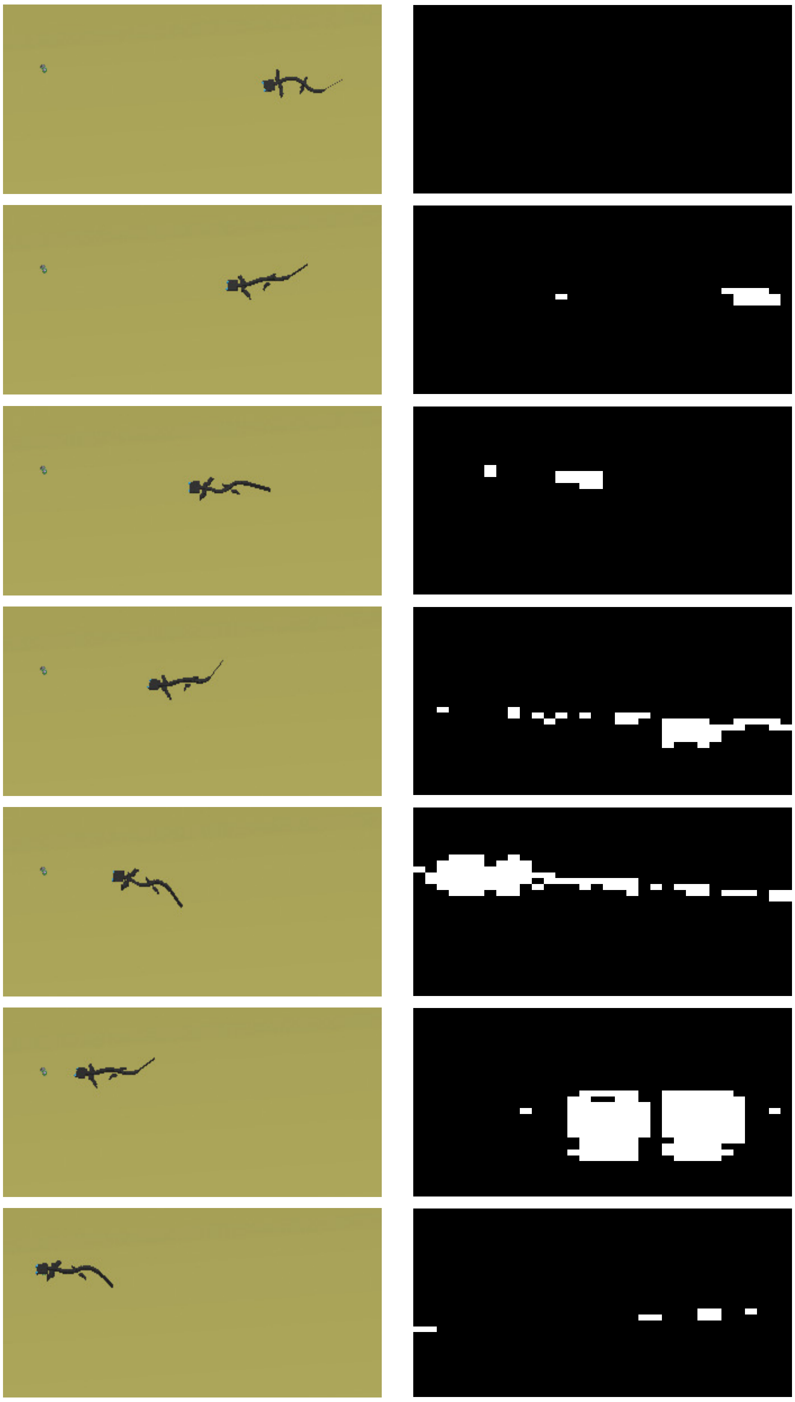


Figure 22: Walking toward a target using vision. *Left:* Salamander walking toward the target. *Right:* brain-stem activity. The salamander turns to the side where the number of firing neurons, corresponding to the number of pixels that changes a lot from one frame to another, is bigger. Note that we had to use an uniform ground to prevent it to activate too much cells and to make the object indistinguishable.

As represented on Figure 22, the object (a can) create areas of activity (firing neurons). As the salamander comes close to the object, the area of activity becames wider.

This study account for only one object. Some trials with two objects (of approximately the same size) lead to the convergence toward the closest object. The salamander first hesitates between the two while they are both far (and of about the same size on the cameras). When approached, the closest becomes more significant on the brainstem, and our salamander thus converge toward it.

4.3 Improvement

An easy way to improve the visual system is to use a finer discretization of the possible directions. Here, we only discretize with two direction: left and right. We could as well use n direction, by splitting the brainstep array horizontally by n, and associate a direction for each subdivision. We could also change our binary controll (one direction an not any other), by assigning to each direction a weight proportionnal to the number of firing neurons in the corresponding subdivision of the screen. This could also improve the behaviour with several object (a big object on the back first attracts attention, before an eventual smaller object becomes closes and changes the direction of attention of the salamander.

References

[1] M A Ashley-Ross. Kinematics of the transition between aquatic and terrestrial locomotion in the newt taricha torosa. *Journal of Experimental Biology*, 207(3):461–474, 2004.

[2] Isabelle Delvolve, Tiaza Bem, and Jean-Marie Cabelguen. Epaxial and Limb Muscle Activity During Swimming and Terrestrial Stepping in the Adult Newt, Pleurodeles waltl. *J Neurophysiol*, 78(2):638–650, August 1997.

[3] A.J. Ijspeert and M. Arbib. Locomotion and visually-guided behavior in salamander: a neuromechanical study. In G.T. McKee and P.S. Schenker, editors, *Proceedings of Sensor Fusion and Decentralized Control in Ro botics Systems III, (SFCRSIII)*, volume 4196 of *SPIE Proceedings*. SPIE, 2000.

[4] Auke Jan Ijspeert, Alessandro Crespi, Dimitri Ryczko, and Jean-Marie Cabelguen. From swimming to walking with a salamander robot driven by a spinal cord model. *Science*, 315(5817):1416–1420, 2007.

[5] Webots. http://www.cyberbotics.com. Commercial Mobile Robot Simulation Software.