

MODELING THE SALAMANDER SWIMMING GAIT WITH VIRTUAL MUSCLES ON A ROBOTIC PLATFORM

MASTER THESIS



Author:
Jonathan GRIZOU

Professor:
Auke JAN IJSPEERT
BIOROBOTICS LABORATORY (BIOROB)

Supervisors:
Jérémie KNÜSEL
Konstantinos KARAKASILIOU

September 13, 2011

Abstract

This project intends to model the interaction between muscle activation and body curvature for salamander locomotion. The final objective is to reproduce a swimming dynamic on the salamander robot, a nine segments bio-inspired robot capable of swimming and walking, by driving the resulting musculoskeletal model with recorded electromyography.

At first, it has been extracted kinematics of several salamanders from slow motion X-ray movies. Afterwards the musculoskeletal model architecture has been developed based on previous works and robot constraints. Then, using the Webots simulator, the model parameters have been optimized according to statistical electromyography and kinematics. Finally, the model has been successfully tested on the salamander robot in the lab.

Acknowledgements

First of all, I would like to thank professor Auke Jan Ijspeert for providing me the opportunity to do my Master project in his lab.

Thanks to Jérémie and Kostas to have supervised and advised me during this project.

Thanks to Alessandro for the many helps concerning computers and robotics issues, and his quick acting on the robot firmware v6 update.

Thanks to Jesse for all the nice software tools developed.

Thanks to Steeve for the very interesting talks we had together and the introduction to muscle and muscle modeling.

And finally thanks to all the BIOROB team members for the good spirit maintained, it has been a great experience to work in this lab.

Remarks

A compact disc should be enclosed with this report, it includes many videos which are helpful to visualize important result.

Contents

1. Introduction	1
1.1. Overview	1
1.2. Method	2
1.3. Report	2
1.4. Robot	3
2. Kinematics and EMG	5
2.1. Slow motion X-Ray movies	5
2.1.1. Tracking	5
2.1.2. Calibration	7
2.1.3. Segmentation	8
2.2. Extracting useful data	10
2.2.1. Fitting of a sinus function	10
2.2.2. Resulting data	12
2.3. EMG	14
2.4. Discussion	19
2.4.1. X-Ray data	19
2.4.2. EMG	19
2.4.3. Timing	19
3. Muscle/Joint Model	21
3.1. Joint model	21
3.1.1. Salamander morphology	21
3.1.2. Previous Work	22
3.1.3. Ideal muscle/joint model	25
3.2. Muscle model	26
3.2.1. What is a muscle ?	26
3.2.2. Muscle modeling	28
3.2.3. Muscle activation	31
3.2.4. Energy estimation	32
3.3. Constrain imposed by the robotic platform	32
3.3.1. Effect of robot segmentation	32
3.3.2. Effect of frequency limitation	37
3.4. Summary	37
3.5. Discussion	37
4. Optimization	38
4.1. Particle Swarm Optimization	38
4.2. Webots Model	38
4.2.1. Tail modeling	39
4.2.2. Shaped robot	42
4.3. Firsts attempts	45
4.3.1. Reducing the parameter space	46
4.4. Protocol	48

4.5. Results	49
4.6. Improvements	58
4.7. Discussion	59
5. Robot	60
5.1. Robot architecture	60
5.2. Problems	60
5.3. Solutions	61
5.3.1. Lookup table	61
5.3.2. Code improvement	61
5.3.3. No acknowledge	62
5.3.4. A broadcast protocol	62
5.3.5. Residual problems	62
5.4. Results	63
5.5. Discussion	63
6. Future works	64
6.1. Webots	64
6.1.1. 30 segments + tail model	64
6.1.2. Real size, shape and weight model	64
6.2. Optimisation	64
6.2.1. Energy optimization	64
6.2.2. Muscles have weight	65
6.3. Model	65
6.3.1. Slow and fast muscle	65
6.3.2. Tendon, spring and damping	65
6.4. Control	66
6.4.1. EMG as coupled oscillators	66
6.4.2. EMG feedback	66
6.5. Robot	66
6.5.1. Realistic shaped robot	66
6.6. Robot enhancement	66
6.7. Discussion	67
7. Conclusion	68
Bibliography	72
A. Appendix	73
A.1. Tracking	73
A.2. EMG	75
A.3. Correlation analysis	77

Nomenclature

c	Muscle Model : Constant
d	Joint Model : Length of one segment
F_{HPE}	Muscle Model : Passive property
f_l	Muscle Model : Length coefficient
F_{max}	Muscle Model : Maximal force
F_{MF}	Muscle Model : Force generated by the contractile element
f_v	Muscle Model : Velocity coefficient
K	Muscle Model : Curvature constant
l_{CE}	Muscle Model : Length of the contractile element
l_{opt}	Muscle Model : Optimal length, muscle fiber length at rest
N	Muscle Model : Relative amount of force reached at a lengthening velocity v_{max}
r	Joint Model : Width of one segment
v_{CE}	Muscle Model : Speed of the contractile element
v_{max}	Muscle Model : Maximum speed of muscle fiber
w	Muscle Model : Stretch constant

List of Figures

1.1.	Timing of muscle strain and activation from [19]	1
1.2.	Optimization process	2
1.3.	Lamprey robot	3
1.4.	Salamander robot	3
1.5.	Robot dimensions	4
1.6.	Centipede and fish robot	4
2.1.	One X-Ray frame after tracking	6
2.2.	X-Ray calibration cube	7
2.3.	Depth dependent calibration schema	8
2.4.	One frame after calibration	8
2.5.	Animal and robot segmentation	9
2.6.	One frame after segmentation	10
2.7.	Sinus fitting on smoothed data	11
2.8.	Sinus fitting with wrong amplitude	12
2.9.	Median Amplitude	13
2.10.	Median PhaseLag	13
2.11.	Final angle kinematics	14
2.12.	Delvolv� EMG recording [22]	15
2.13.	Delvolv� dependence of rostrocaudal phase lag on position of myomeres along the body axis from [22]	16
2.14.	Reconstructed phase lag	17
2.15.	Reconstructed duty cycle	17
2.16.	Reconstructed EMG for right side	18
3.1.	Salamander skeletal layout, provided by Kostas	21
3.2.	Salamander muscle repartition from [21]	22
3.3.	Fish myomere from [51]	22
3.4.	Muscle/Joint Model used by O. Ekeberg from ??	23
3.5.	Muscle/Joint Model used by A. J. Ijspeert from ??	23
3.6.	One segment	24
3.7.	Simple Muscle/Joint	24
3.8.	Linear Muscle/Joint	25
3.9.	Complex Muscle/Joint	25
3.10.	Implemented Muscle/Joint	26
3.11.	Muscle structure from [38]	27
3.12.	Sarcomere properties from [38]	28
3.13.	Sarcomere force velocity property from [38]	28
3.14.	Hill muscle model equation from [32]	29
3.15.	Muscle model	29
3.16.	Muscle passive property	30
3.17.	Muscle force length relation	30
3.18.	Muscle force velocity relation	30
3.19.	Muscle contractile element force	30

3.20. Muscle model Matlab plot	31
3.21. Muscle activation to stimulation function from [60]	31
3.22. Joint angle amplitude increase due to segmentation	33
3.23. Joint model segmentation effect	33
3.24. Joint model	34
3.25. Musle/Joint Model : Force to Torque relation	34
3.26. Muscle/Joint ReSegmentation Choices	35
3.27. New w value in function of the angle	36
3.28. The effect of changing the w muscle parameter	36
4.1. Optimization setup	38
4.2. Experimental setup for the tail characterization	39
4.3. Experimental analysis	40
4.4. Torque versus joint angle for tail characterization, rigid and soft side	41
4.5. Soft side all joint spring constant, tracking number 3	41
4.6. The robot model on Webots with tail	42
4.7. A walking gate X-Ray frame	43
4.8. Tracking profile	44
4.9. Body length to body width profile of a real salamander	44
4.10. Actual versus realistic robot dimension	45
4.11. Realistic shaped Webots model	45
4.12. Real salamander mass distribution	47
4.13. Weight profile	48
4.14. Expected and obtained joint angle after optimization	51
4.15. Resulting angle amplitude compare to statistical data	52
4.16. Resulting angle phase lag compare to statistical data	52
4.17. Resulting lateral displacement compare to statistical data	53
4.18. Muscle stimulation and body dynamics	54
4.19. Timing comparison	55
4.20. Muscle working zone	56
4.21. Energy consumption	57
4.22. Proposed muscle/joint model	59
5.1. Robot time step instability at 30 ms and stability at 10 ms	61
6.1. The 30 segment robot model on Webots with tail	64
6.2. Muscle/joint model with muscle, tendon, spring and damping	66
A.1. Tracking summary sheet	73
A.2. Tracking summary sheet	74
A.3. Delvolve data summary sheet	75
A.4. Delvolve data summary sheet	76
A.5. Correlation analysis	77

List of Tables

2.1. Median Kinematics Value	14
4.1. Median Spring Constant	42
4.2. Width robot value	44
4.3. Width profile value	47
4.4. Weight profile value	48
4.5. Optimised Parameters	50
4.6. Swimming data	53
4.7. Speed data	54

1. Introduction

Scientific work, as philosophy, start by an astonishment. The one which motivated this work is about life, about movement. The question is : how does a salamander locomote ? This astonishment has been scientifically addressed since several tens of years, from biology to control theory, salamander is one of the more studied animal because it is likely to be the one who did the transition from water to land.

1.1. Overview

Francis [24] in 1934 wrote a state of the art books about the anatomy of the salamander. Blight [10, 9] in 1977 published a complete review about vertebrate swimming movements and show that anguilliform fish swim by propagating a wave of contraction for head to tail. Many others scientists bring their view and analysis about anguilliform swimming [48, 26, 54, 18, 27, 28, 25]. But kinematic is the tip of the iceberg about locomotion, the final result of complex interaction of muscles, tendons, and bones, but also a complex neural networks system, feedback information, etc.

Therefore researchers start looking inside the animal and electromyography (EMG) recording has been performed on salamanders, a reference work has been published in 1997 by Delvolv   [22] but also [52, 25].

From those studies, a new astonishment emerges from the specific timing between muscle activation and body movement as described in [19, 57, 53]. As show in figure 1.1 (A), the traveling speed of EMG is higher than the bending wave speed. In others words, the role of muscle activation is different along the salamander body.

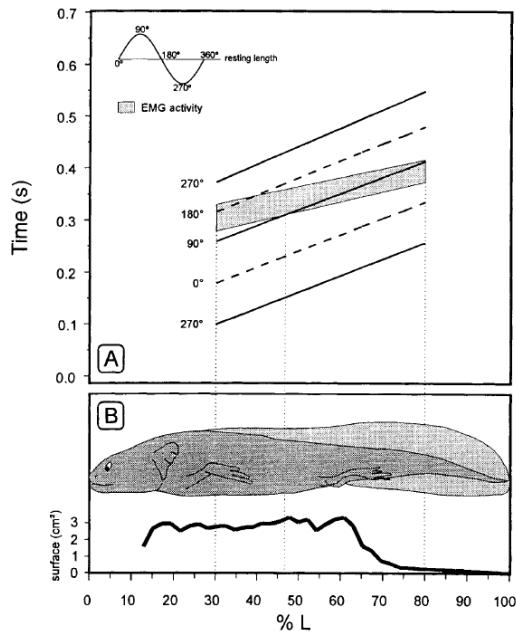


Fig. 2. Steady swimming in *Ambystoma mexicanum*. A. Diagram showing the strain cycle (oblique solid and dashed lines) and EMG activity (grey surface) by means of linear regressions through the data points. The meaning of the strain phases is explained in the text and indicated in the inset. B. Transverse surfaces of epaxial muscle (both sides) along the body, measured from 3 mm thick slices. The muscle distribution shows a sharp decline in the tail region, which is in contrast to the much smoother profile for lamprey and eel (HESS, 1983).

Figure 1.1.: Timing of muscle strain and activation from [19]

This non intuitive relation is a nowadays interrogation, people wants to model the muscle to body interaction in order to understand the role of each part of the body (for biologist), but also for being able to study the locomotion control from a natural perspective. As the missing link between EMG and kinematics are muscles and body architecture, detailed studies of muscle repartition [14, 47, 6] and functional role of muscles [42, 2, 7, 21, 5, 13, 51] have been done. Moreover many studies of the salamander locomotion control show the important role of central pattern generator to produce stable walking and swimming gait [23], and can also explain the transition from swimming to walking [36].

A step further has been to model the body to water interaction, the body dynamics, by developing software mechanical model of a salamander [11, 16, 15, 58], models which enable scientist to study, test and validate different gaits and modes of locomotion [30] and complex central pattern generator (CPG) architecture [23, 36, 35].

Finally, one others interesting direction has been the development of a robotic platform [17] which enable researcher to cross the reality gap and validate all previous studies in the real word [36].

The study of each pieces of the puzzle is quite advanced, but the system as a one have yet to be reconstructed. The goal of this project is to make a new step in this direction by focusing on two aspects : (1) the relative timing of muscle activation and body curvature and (2) the robotic platform tool. The aspect (1) because reproducing this dynamic will enable future research on neural networks, EMG and feedback control according to the effect on the animal kinematic. The aspect (2) in order to use the salamander robot to validate the results or test hypothesis on real interaction condition.

1.2. Method

The method detailed in this report is as follow. First to develop a generic muscle/joint model for the salamander robot which depend on few parameters. Then to tune the musculoskeletal model parameters via a simulation software called Webots [45] and an optimization algorithm called Particule Swarm Optimization [39]. Optimization algorithm try to find the best set of parameters value to increase a fitness function. As our model intend to model the muscle to body interaction, input will be real EMG data and output will be real kinematics data. The all process is resume in figure 1.2. Finally the model will be implemented on the robotic platform.

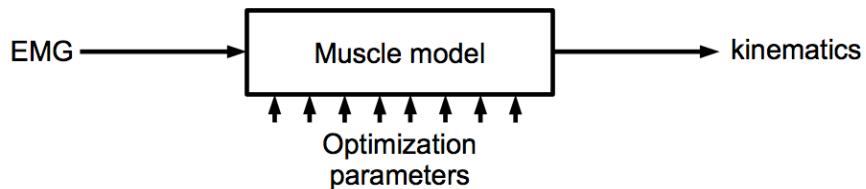


Figure 1.2.: Optimization process

1.3. Report

The two following chapter (2 and 3) respectively describe what are the morphological data we used and how we set the muscle/joint model. Both of them are strongly linked together. Kinematics and EMG come first because some data from there are used to finalize the muscle/joint model. Chapter 4 describe the optimization process, the resulting performance and an analysis of them. Chapter 5 explain the robotic implementation.

1.4. Robot

The salamander robot is a modular robot. Three different types of module are needed :

- body module
- limb module which is bigger but have one rotating limb on each side
- head module which includes the main computational power and do not have any actuator

Each module is connected to the others via a rotating joint thought which pass a CAN bus. The head is controlling the communication on the bus and centralize the informations. Each segment is able to command his rotating joint in position or in torque but also to feedback the position, the speed, and the torque.

Using those modules we can build several kind of robot. For example, the lamprey robot is made of one head plus eight body modules (figure 1.3).



Figure 1.3.: Lamprey robot

The salamander robot is composed of one head plus six body modules plus two limb modules and finally a plastic tail to complete a realistic size scale robot (figure 1.4).



Figure 1.4.: Salamander robot

The overall dimensions of the modules and the salamander robot are described in figure 1.5.

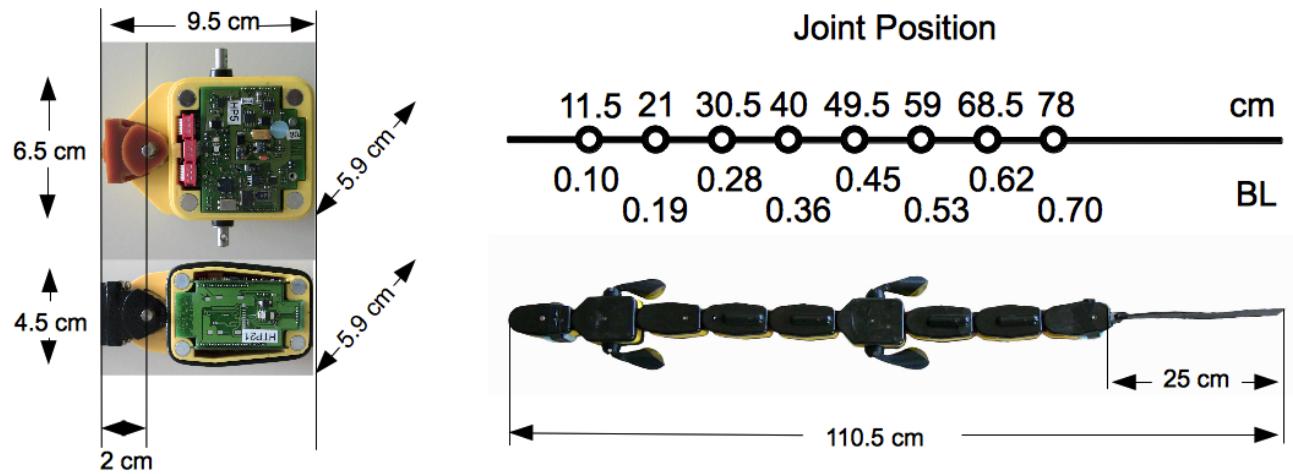


Figure 1.5.: Robot dimensions

A centipede robot can also be created by using only limb modules (figure 1.6). As well as a fish robot composed of only one of each module (figure 1.6).



Figure 1.6.: Centipede and fish robot

2. Kinematics and EMG

We need biologically relevant data to validate and tune our musculoskeletal model as explained in 1.2. We need kinematics data as a reference for the salamander swimming gait, and EMG data as input to the model. This chapter explain how kinematics data have been extracted from slow motion X-Ray movies and EMG data obtained and reconstructed based on 1997 Delvolv  paper [22].

2.1. Slow motion X-Ray movies

Few months before I integrate the lab, Kostas recorded several slow motion X-ray movies of swimming and walking salamanders. Side and top view have been recorded for each experiments. It was the first time a salamander has been tracked this way, it allows us to rebuild the all kinematics based on the animal's bones.

Note : You can find samples in the CD's X-Ray folder ¹.

2.1.1. Tracking

Unfortunately the tracking is not automatic, and has be done by hand via the mTracking software developed by Kostas. Tracking is a bored action which consist to click for every frame of every movies the position of the bones.

The tracked points are :

- tip of nose
- left jaw
- right jaw
- left scapula back
- left scapula front
- left shoulder
- left elbow
- left wrist
- left root middle finger
- left tip middle finger
- right scapula back
- right scapula front
- right shoulder
- right elbow

¹CD/X-Ray/Roentgen_A_VisarioG2_006_001.mp4 and Roentgen_B_VisarioG2_011_001.mp4

- right wrist
- right root middle finger
- right tip middle finger
- left hip
- left knee
- left ankle
- left foot root finger
- left foot tip finger
- right hip
- right knee
- right ankle
- right foot root finger
- right foot tip finger
- 1 to 50 points for the tail

The following figure 2.1 show one frame after tracking.

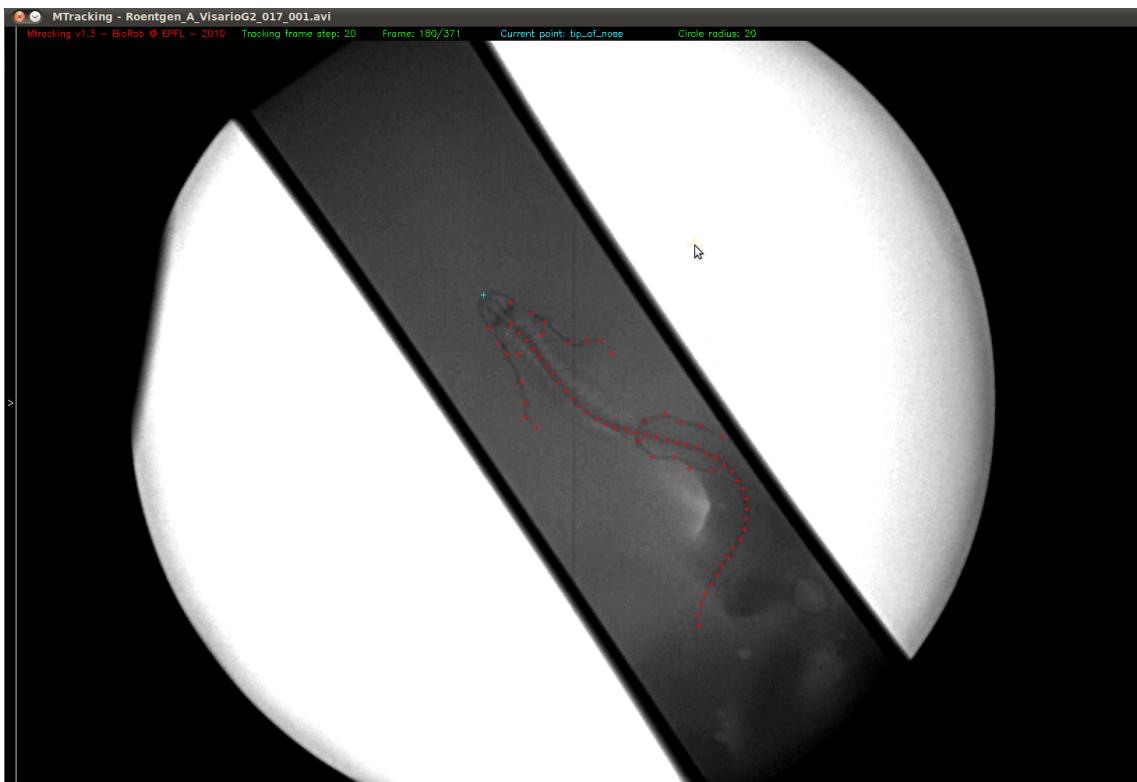


Figure 2.1.: One X-Ray frame after tracking

You can find in appendix A.1 a description of the tracked movies in terms of frame step, number of cycle tracked, quality of the video, and comments.

This tracking software extracts the coordinate of tracked points in terms of pixels. Therefore we need to calibrate the resulting data in order to deal with correct dimension.

At this step we can already create nice 3D animation of the animals kinematics ².

Note : For this project only the vertebral tracked point are usefull. The others points have been tracked for further studies.

2.1.2. Calibration

Two steps are necessary for a good calibration process. First of all the distortion due to lens effect need to be corrected. This have been done by Jérémie and Kostas, they provide me a lookup table correcting it.

The second step is to rescale the data from pixel to centimeter. This operation is not straight forward because this pixel to cm ratio depend on how far the animal is from the objective of the camera. To this end a fixed cube composed of regularly spaced metal dot has been recorded as a reference (figure 2.2). The space between two dot is constant and equal to 10 cm.

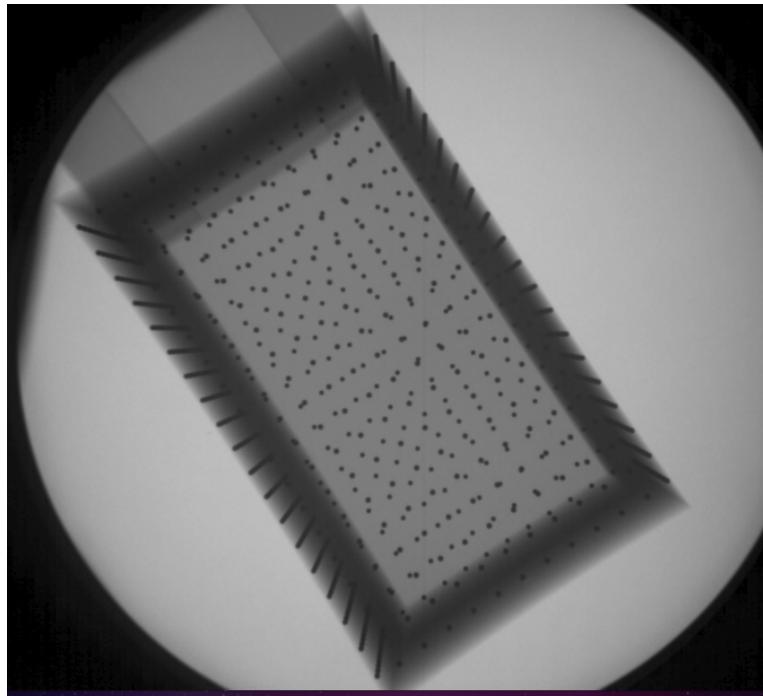


Figure 2.2.: X-Ray calibration cube

The depth of field effect is clearly visible, the top dot have a bigger shadow on the screen and looks more spaced than the bottom one. In others words, for each layer of the space, a different pixel to cm ratio. Using this calibration box we are able to identify the pixel to cm ratio at the top and bottom plane. Thales theorem confirm the linear relation between depth and pixel to centimeter ratio. Therefore we need to use the side view to infer the correct pixel to cm ratio to use for the top view and vice et versa. To explain the problem we can refer to the following figure 2.3.

²CD/X-Ray/movie3D28011.mp4 , movieXY28011.mp4 , movieXZ28011.mp4 and movieZxZ28011.mp4

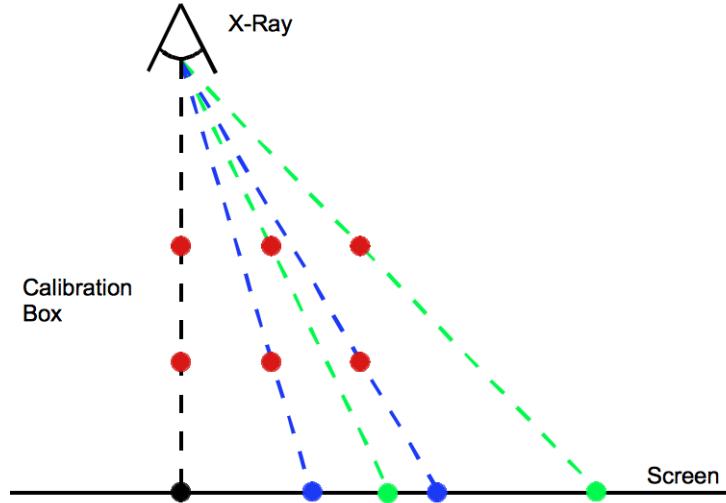


Figure 2.3.: Depth dependent calibration schema

The projector is the source of electromagnetic radiation and the screen is the X-ray sensitive material used to record movies. Note that the optics lens have been omitted in this graph for a better understanding. The side view is used to infer the depth of each tracked point relative to the bottom of the cube and using the linear relation between depth and pixel to cm ratio we can rescale our data to the correct dimension. The following figure show the result of this calibration process for the tracking frame of figure 2.1.

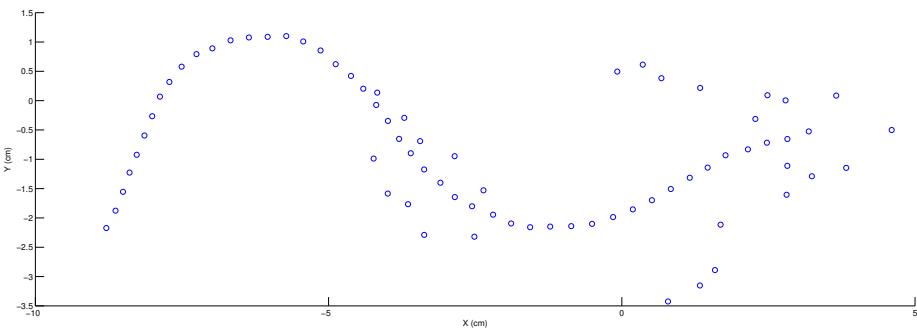


Figure 2.4.: One frame after calibration

You can note the overall dimension of the animal is about 17 cm which correspond to the tracked animals size.

2.1.3. Segmentation

This study focus only on reproducing a swimming gate, to do so biological data about the real animal kinematic are needed. The characterization of the swimming gate is done by segmenting the body

and observing three main properties :

- Angle variation between segments
- Phase lag between segments
- Lateral displacement of joint according to the animal swimming direction

Remember, we want to recreate the kinematic on the salamander robot. As the robot is composed of nine segments we need to recut the salamander in 9 segments. The size of each segment should correspond to the robot size in relative dimension. Salamanders tracked measure about 17cm and robot is 110.5 cm long.

Before re-segmenting the data an important approximation must be done, as the robot can only perform two dimensional movement, the all 3D kinematics has been projected to the (x,y) axis. This approximation can only be done because a swimming salamander is almost horizontal. It disturb a bit the correctness of the segmentation but no others solution are possible.

As the robot is almost 10 times bigger than the recorded animals we need to find a good morphological reference which can be use on both robot and tracked animals. This reference is usually the body length of the animal. This value is two difficult to obtain in the tracked movies because the end of tail tracking is not viable. Water is not transparent to X-Ray, and the end of the tail is quite fine and in most of the case it is impossible to discern the very end of the tail. Therefore we can not accurately extract the salamander body length and the reference used is the length form the first vertebra (head base) to the hind limbs (cross of spinal cord with left to right hind limb line).

In the animals there is exactly 16 vertebrae in this portion but our robot use only 4.5 segment in this interval (figure 2.5).

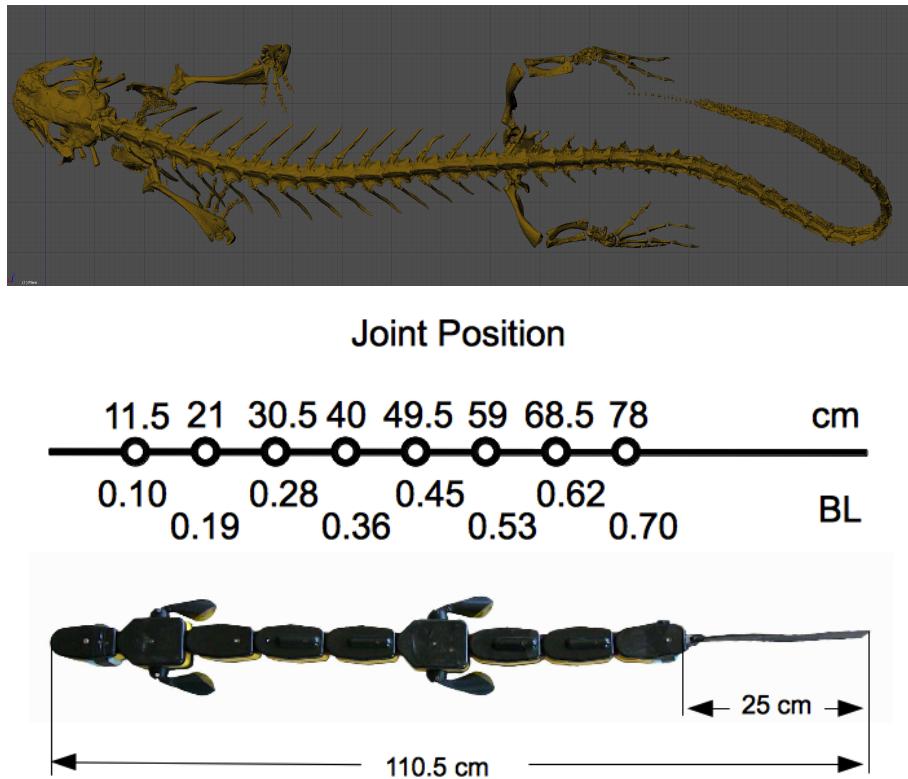


Figure 2.5.: Animal and robot segmentation

The size of one segment is this reference distance divided by 4.5. Based on this segment size we can cut the spine in 8 segment of equal size. The head segment is always independent, his size may vary

and will not be the same as the others. Note that head size is not important, we only consider the angle variation, the head span from the tip of nose to the first vertebra and is a rigid part. Whatever its size, the angle should remain the same. The resulting segmentation is as shown in figure 2.6.

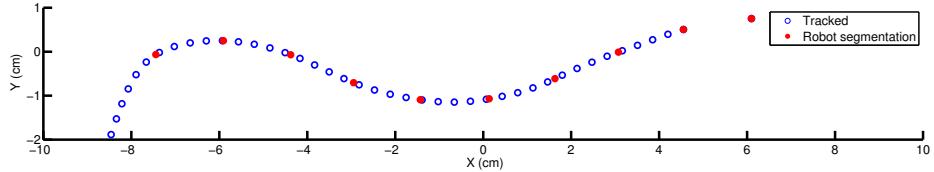


Figure 2.6.: One frame after segmentation

This process have been repeated for all frames tracked. At this step we can already plot nice animation of the animals kinematic associate to angle variations ³. Note that the angle evolution is periodic and look like a sinus function.

2.2. Extracting useful data

Having the correct dimension and segmentation, relevant swimming kinematics data can be extracted. Those data must represent each angle variation according to time. Assuming this function can be approximate by a sinus function, frequency, amplitude and phase are the only needed information. The sinus function assumption have been done after close look to the data, and will be visible in the following subsections.

2.2.1. Fitting of a sinus function

Angle thought time evolution is periodic, the tracking resolution is about 20 frame per cycle and is not perfect. Therefore the first task is to smooth the result. The Matlab *csaps* function have been used with the suggested smoothness factor. Assuming tracking errors are more important than tracking resolution, this technique attenuate the irregularities due to tracking errors. However we can lose some information but this project is not an attempt to study in details the kinematics of the animals and non relevant or small variation should be ignore.

To extract frequency, amplitude and phase of each angle evolution, the fitting tool box of Matlab has been used. The idea is to ask Matlab to fit a home made function on the data. From visual analysis, sinus function is the more appropriate.

The function to approximate is $A \sin(wt + \varphi)$. The interpretation of the results is easy :

- A is the amplitude (rad)

³CD/X-Ray/movieAngleTail28011.mp4 and movieAngleXY28011.mp4

- w is $2 \times \pi \times$ frequency (Hz)

- φ is the phase (rad)

Unfortunately the Matlab fitting tools are not magic and starting values are keys to a good fitting process. In fact, starting value must be very close to the final value, starting value have been set this way :

- $A_0 = \frac{\max - \min}{2}$
- w_0 is set using a Fast Fourier Transform
- φ_0 is set to 0

This setting led to very successful result as display in figure 2.7.

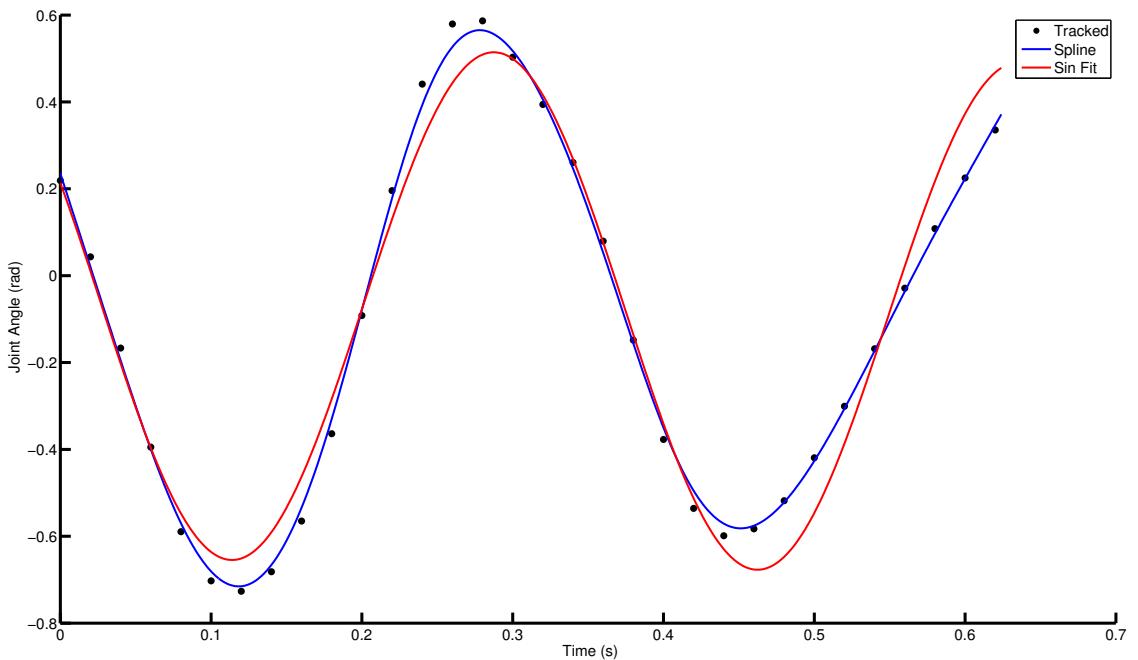


Figure 2.7.: Sinus fitting on smoothed data

Unfortunately most of the time the amplitude given by the sinus fitting process is not perfect and often underestimated (see figure 2.8).

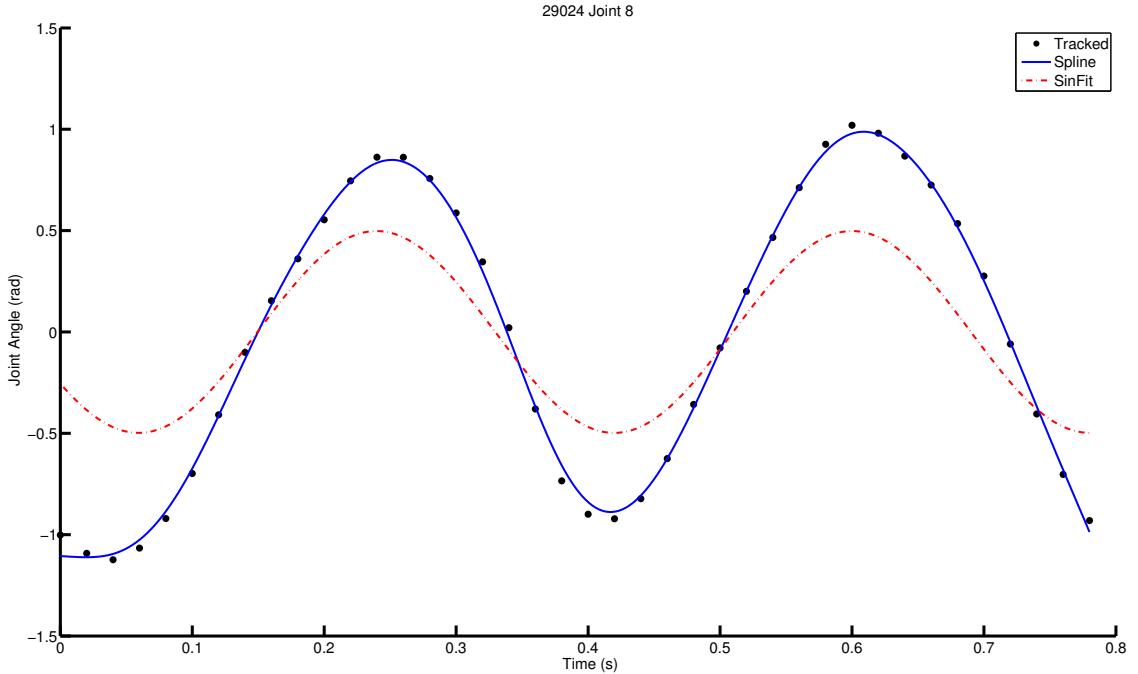


Figure 2.8.: Sinus fitting with wrong amplitude

Therefore the amplitude is determined as $A = A_0 = \frac{\max - \min}{2}$. Moreover the angle phase between each joint lag can be determined form the resulting phase data.

For further analysis, the traveled distance and speed of the animal have been extracted. As well as the lateral displacement of each joint. This have been simply extracted by rotating and translating the animals on the x axis. This way as the animals are moving in straight line, we can directly infer the lateral displacement which is the distance to the x axis.

Result of this sinus fitting process are very accurate but can we directly use them for the robot. In facts the tracking salamanders frequency range from 3 Hz to 7 Hz. Does the kinematics evolve depending on the frequency ? To answer this question, a correlation analysis have been perform on the data. We test the correlation between frequency and speed, angle amplitude and phase lag (see appendix ??). This analysis show a strong correlation between frequency and speed, but no relevant correlation between frequency and angle parameters. It confirm the facts presented by Gary B. Gillis in [26] and K. D'Aout and P. Aerts in [18], salamanders increase their speed by frequency variation and not by amplitude variation.

This is important for two reason, (1) we can discard the frequency parameters and build statistics from all the available data and (2) because the salamander robot can not swim to more than 1.2Hz. As not animals swim to this frequency it could have been very difficult to infer the gait at such a frequency.

Therefore data have all been rescaled to 1Hz and angle amplitude and phase have been averaged to get a final median kinematics of a swimming salamander.

2.2.2. Resulting data

Median value and standard variation are display in the following graph 2.9 and 2.10.

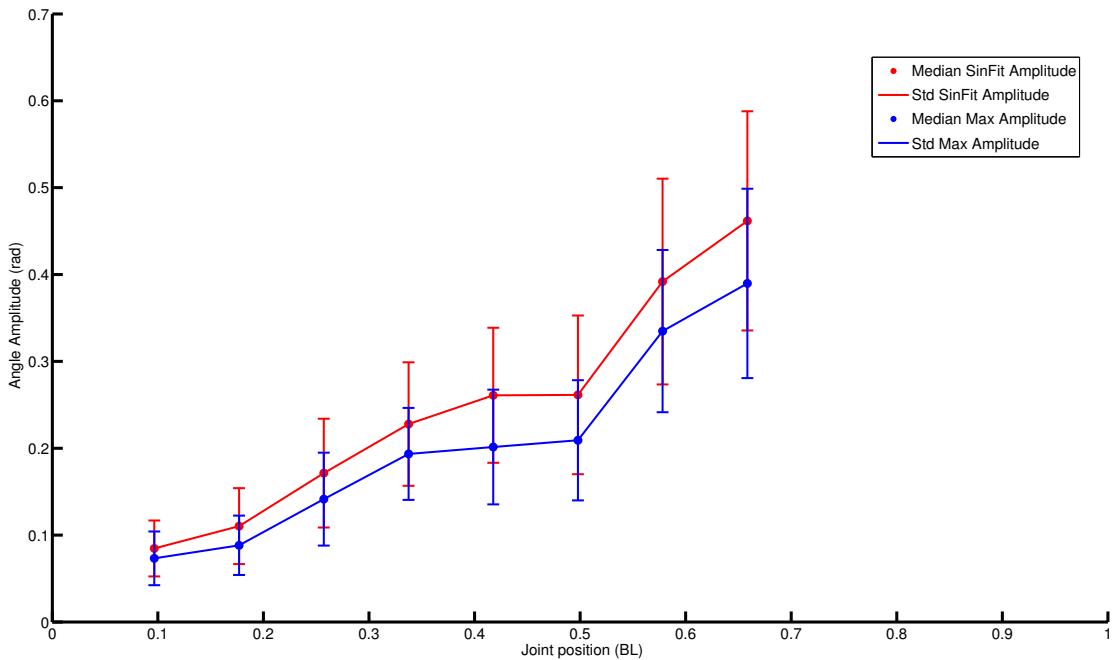


Figure 2.9.: Median Amplitude

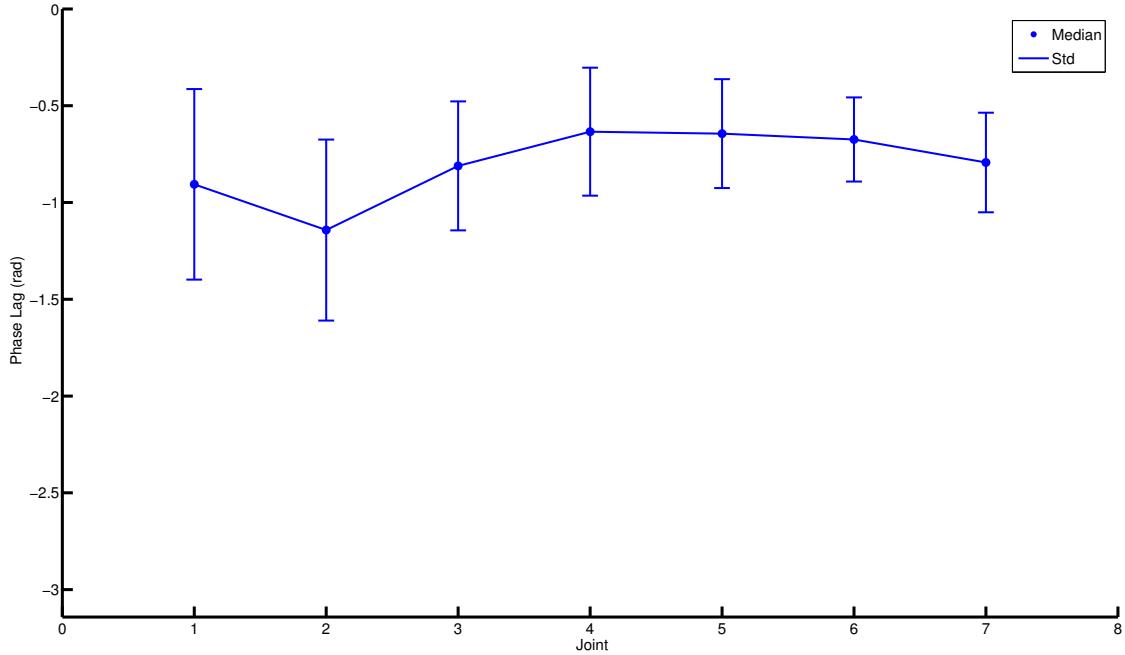


Figure 2.10.: Median PhaseLag

Based on this median values (table 2.1), we can recreate angles kinematics with the initial $A \sin(wt + \varphi)$ function as in figure 2.11.

Median Value (rad)	Joint 1	Joint 2	Joint 3	Joint 4	Joint 5	Joint 6	Joint 7	Joint 8
Amplitude Max	0.085	0.110	0.171	0.228	0.260	0.261	0.392	0.462
Amplitude SinFit	0.073	0.088	0.141	0.193	0.201	0.209	0.335	0.390
Phase Lag	Nan	0.91	1.14	0.81	0.63	0.64	0.67	0.79

Table 2.1.: Median Kinematics Value

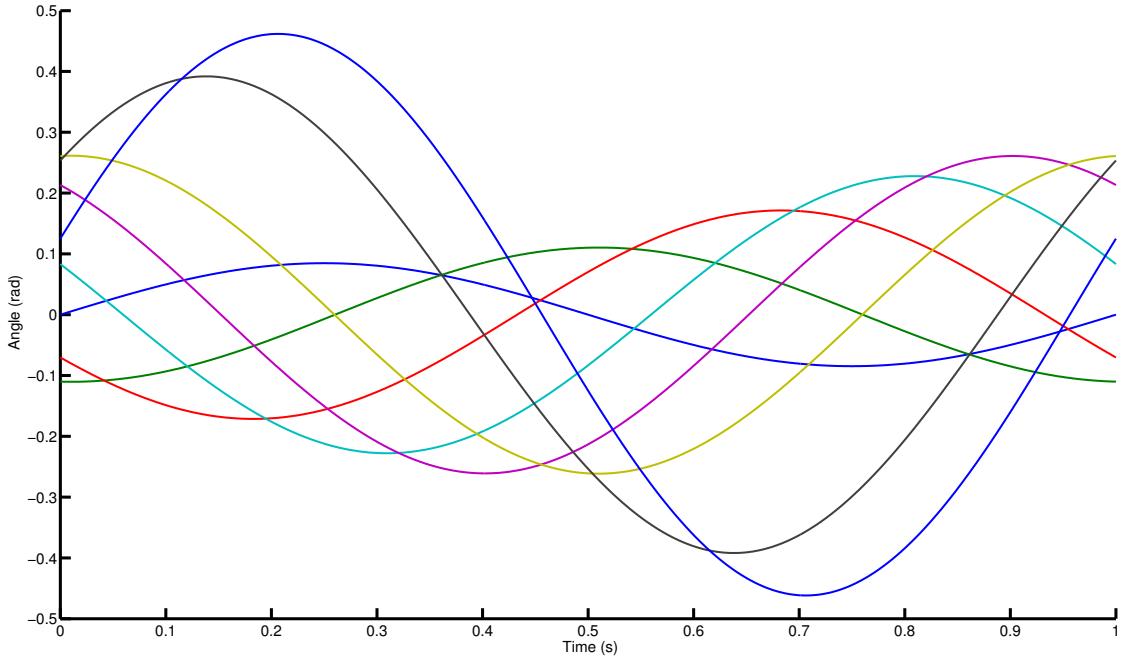


Figure 2.11.: Final angle kinematics

Note that only one cycle is needed because the function is periodic.

Note : The use of a sinus solve many problems. First of all we cannot be sure that the frequency do not change a bit the shape of the angle evolution. Tools to identify the correlation of a shape thought frequency are too complex for our problem. Moreover if those tools exist they will not solve the problem because the robot cannot swim to 3Hz and the frequency change shape effect will not be predictable as no salamander swim to 1Hz in nature

2.3. EMG

Kinematics data are the desired output of our model, input remain to be set. Activation of a muscle is done by motor neuron who created an electrical stimulation to the muscle. Those electrical stimulation are considered as the command of the muscle and Electromyography is a technic which permit to record this activity. The initial idea was to use synchronized kinematics and EMG recording. Unfortunately those data were not available at the time. Fortunately many studies transcript detailed EMG recording for salamanders, particularly the 1997 Delvolv   study [22]. Note that this recoding (figure 2.12) have been perform on epaxial muscles which are the one mainly responsible for a lateral bending of the body.

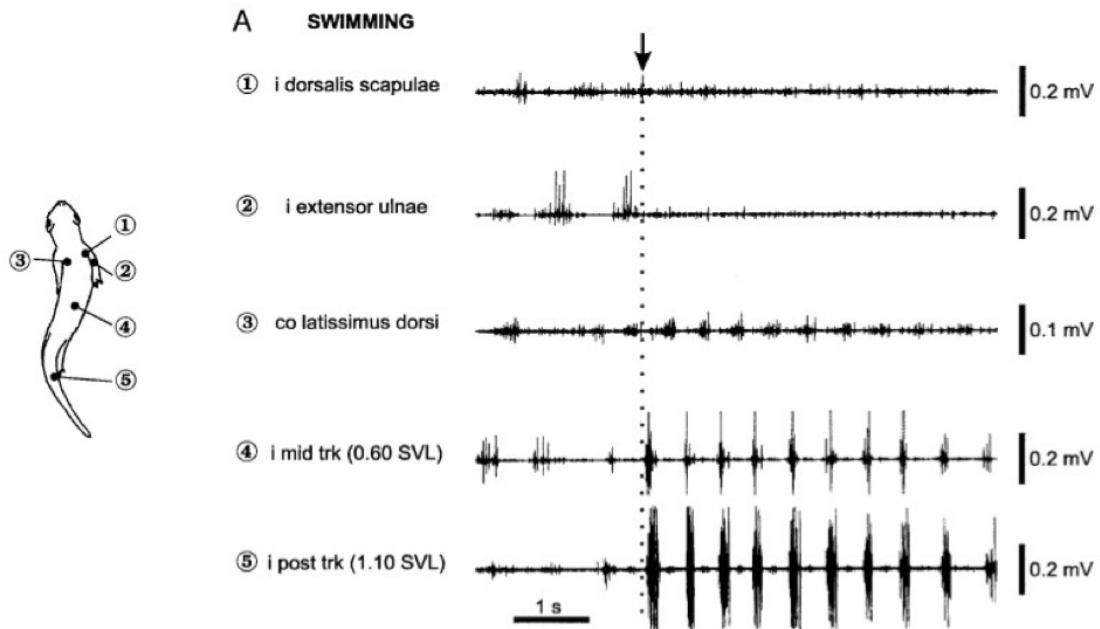


Figure 2.12.: Delvolvé EMG recording [22]

In this paper we can find tables (cf p641 [22]) which indicate the duty cycle of activation in function of the body position. And plots (figure 2.13) which indicate the delay of activation depending of a reference and the body position.

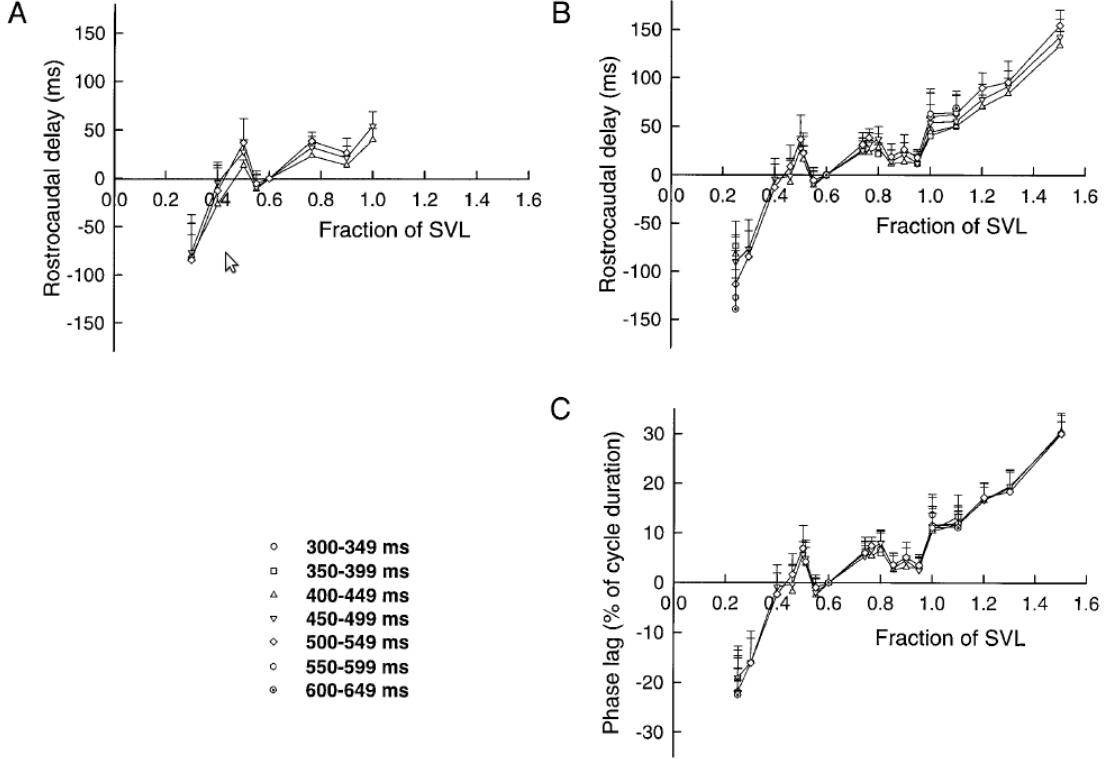


FIG. 6. Intersegmental coordination during swimming. *A* and *B*: dependence of rostrocaudal delay (ordinate) on position of myomere along body axis (abscissa). Each symbol represents mean rostrocaudal delay obtained for observed range of swimming cycle duration (indicated in *inset*). Negative values of delay: myomeres activated before reference myomere (i.e., 0.60 SVL recording site). Positive values: myomeres activated after reference myomere. *C*: dependence of rostrocaudal phase lag (ordinate) on position of myomere along body axis (abscissa). Rostrocaudal phase lag was calculated as rostrocaudal delay divided by corresponding cycle duration. Each symbol represents mean value of rostrocaudal phase lag in range of swimming cycle durations indicated in *inset*. Data were from single individual in *A* and pooled from 5 individuals in *B* and *C*. In *A*–*C*, vertical bars indicate SDs of delay and of phase lag, respectively.

Figure 2.13.: Delvolv   dependence of rostrocaudal phase lag on position of myomeres along the body axis from [22]

We need to transform those data from snout-vent length (SVL) to body length (BL) as this is the unit I choose for all my data. The SVL to BL factor chosen is the one of the robot. Note that this factor is size dependent, small animals have a higher $\frac{SVL}{BL}$ ratio, which mean their tail are smaller compare to trunk size than bigger animals. The robot $\frac{SVL}{BL}$ ratio is about 0.48. The Delvolv   data have been averaged and can be found in appendix A.2. The resulting relations are shown in the following graphs 2.14 and 2.15.

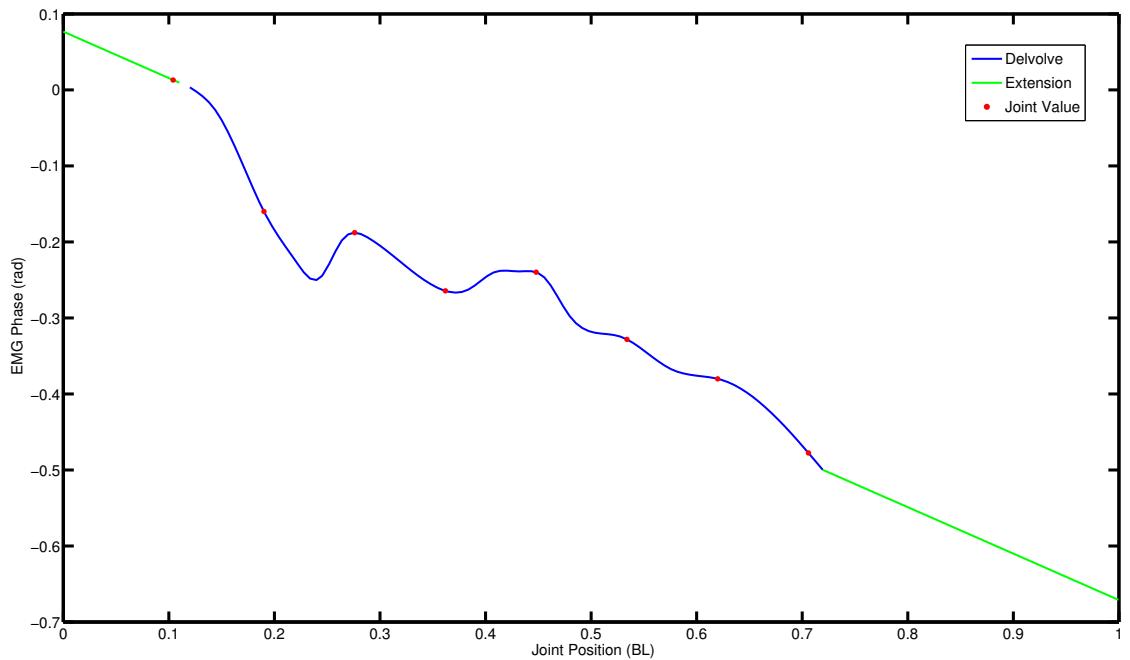


Figure 2.14.: Reconstructed phase lag

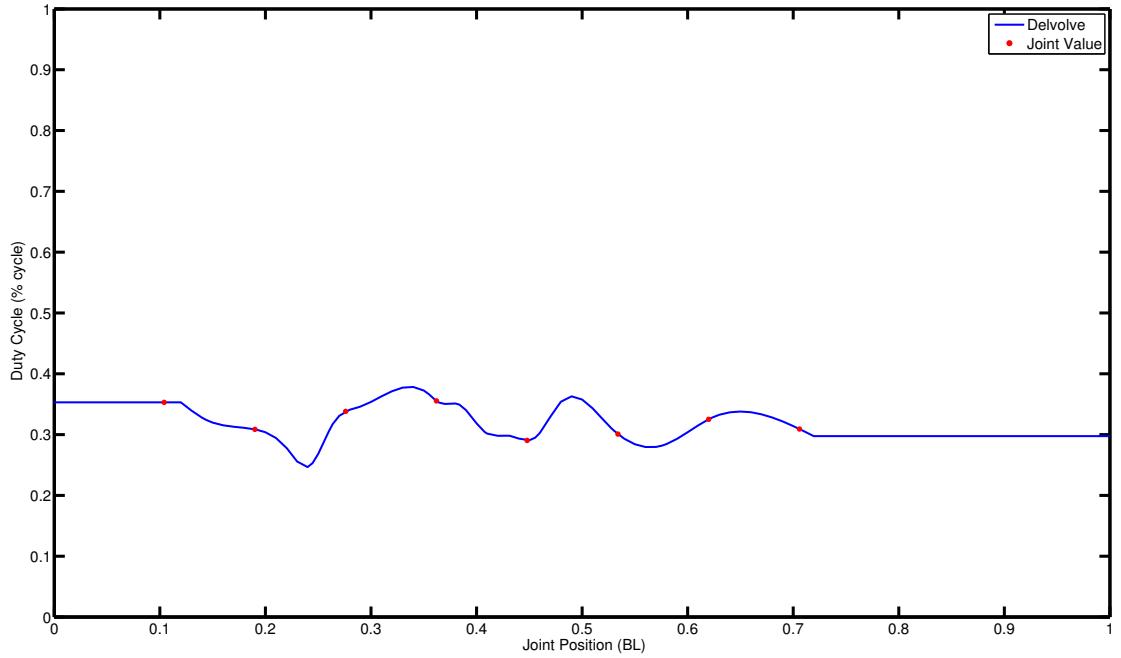


Figure 2.15.: Reconstructed duty cycle

Note that all those graph are expressed in body length and in percentage of cycle duration which permit to reconstruct any kind of EMG based on the desired frequency and the joint position along the body. The red dot on figure 2.14 and 2.15) correspond to the joint body length position relative

to the robot.

One information is still missing, EMG are periodic signal, we know the duty cycle and the phase lag but the shape of the signal is not given. Example of EMG shape can be found in figure 2.12, we can note a sawtooth profile composed of many spike. We will not try to reproduce this complex profile. The general pattern look like a positive sinus during the activation and is equal to zero elsewhere and will be modeled this way.

As we previously set the kinematics swimming data to 1Hz, the EMG are also set to 1 Hz. Keep in mind that kinematics as well as EMG are, for this study, supposed as frequency independent. We also assume a maximal robot swimming frequency of 1.2Hz.

We should note that when biologist record EMG data, they cannot deduce the amplitude of signal. The same signal can have two different amplitude depending on the way the sensor is placed in the muscle of the animals. As no information are available, we assume a constant level of activation. In our muscle muscle, muscle activation is scaled from 0 to 1, therefore I arbitrarily choose to use an EMG amplitude of 0.8 to keep the ability to increase the frequency. An increase of frequency led logically to an increase of activation level, at a higher frequency the muscle should contract faster and with more force because the water damping will be higher. For a better understanding of muscle activation see chapter 3.

The musculoskeletal model will have a pair of muscle for each joint, one for the left side and one for the right side. The phase lag between left and right activation is logically set to 0.5 cycle.

Based on duty cycle and phase lag data from Delvolv , the positive sinus shape and the 1Hz frequency we can reconstruct a median EMG activity for all joint of the robot (figure 2.16).

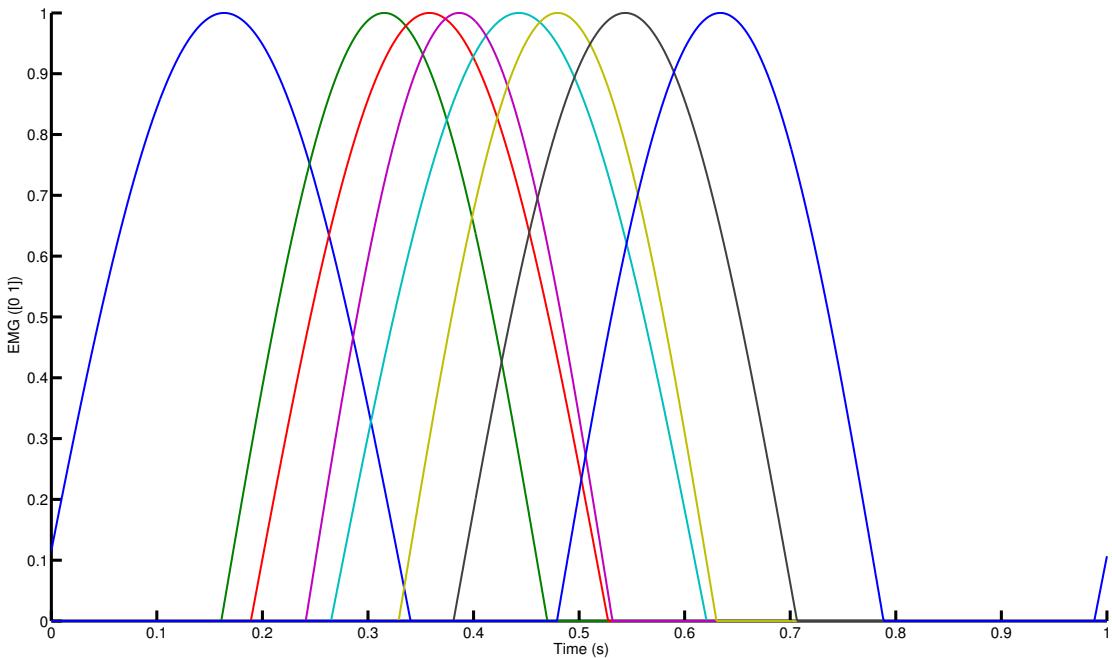


Figure 2.16.: Reconstructed EMG for right side

Note : In this graph the amplitude is 1 because it will be set in the simulation software.

2.4. Discussion

2.4.1. X-Ray data

Little approximations have been done during the course of the kinematics extraction, in particular when we switch from 3D to 2D. They are necessary approximation because the robot can only perform planar undulation. Even if we wanted to simulate a 3D virtual modeling on Webots, it will be extremely difficult to solve the problems of control, musculoskeletal modeling, water interaction. The facts we segmented the body in bigger segment than the animal vertebra size also lead to little approximation. The body length position of each joint is not always the same because of this re-segmentation. No problem when the body is straight, but when the body is fully bended, the two same segment are in fact representing a bit more body length than in the straight position. Many little others approximation can be found (uncertainty of tracking, water visual deformation) but in fine, all those data are smoothed and we will only consider median characteristic of all data. Those approximations could have influence for further study about the animals kinematics only, but not in our case.

In an other direction, I have performed few more studies and statistics about the animals kinematics, some result show that an important bias have been integrated in the experiment. When we want to analyze the swimming efficiency as in [18], for some experiment we obtain value superior to 1, which is impossible unless the salamander have been launched in the pool. It appear that it has been the case. As it is extremely difficult to ask a salamander to swim as we want and when we want, a ramp has been used to propel salamanders and force them swimming. This analysis has been done quickly but this experimental bias should be kept in mind for further studies and at least checked again.

2.4.2. EMG

An important question must be at least asked. As our model will have a different segmentation than the real animal, should we extend the duty cycle because of the much important area covered by each muscle ? I had hard time thinking to this problem. Should I use those value this way or should I modify them, and if I should, according to what ? I am not a biologist and my answer to this question will lack of experience and knowledge.

Those EMG measurements are single point in the body. We can definitively see the pattern of muscle activation thought the Delvolv  article. But one particular muscle extend over several millimeters or even centimeters along the animals body. Are muscles activated only when the needles record activity or during a little more time ? If we place an electrode in the middle of one particular muscle, we can assume that a part of the muscle have been activated a bit before we recorded and a bit after if the EMG are traveling along the body.

First of all what we called traveling wave of muscle activation should not be seen as an electrical wave going thought the body, it is a complex mechanism of bunches of neurons linked together to produce this pattern. Secondly, the use of a needle to record EMG can induce error in interpretation, the needle suggest we are recording electrical simulation to one precise location. But in facts the measurement cover a surrounding area around the needle.

To know the overall activation of the muscle, should we sum the overall activity taking place along the muscle length ? I have no answer to this question, I just know that the only possible change is to increase the duty cycle because the activation should remain between 0 and 1.

The final and only argument is that if I don't have precise argument, I should not change anything. I am very concern about it, I still don't know the correct way but I decided to use the Delvolv  as they are.

2.4.3. Timing

It is disappointing that we couldn't use synchronized EMG and kinematics data because that add one other variable in the optimization process : the relative timing. Therefore I decided to transform this lack in an advantage. Some data are available about the relative timing of EMG and mechanical

waves [19, 57, 53] . But not enough precise to use them. Thus if after our optimization process we are able to retrieve the relative timing shown in paper, it will be an indication that our model is able to reproduce the main characteristic of the muscle to body interaction during a swimming gait.

3. Muscle/Joint Model

This chapter first present how the joint model have been design based on previous work. Then the muscle model and finally, the muscle/joint model modification due the robotic platform constrain.

3.1. Joint model

The joint model is mainly the choice of how the muscles are attached to the body and how to deduce from it the torque applied at the joint.

3.1.1. Salamander morphology

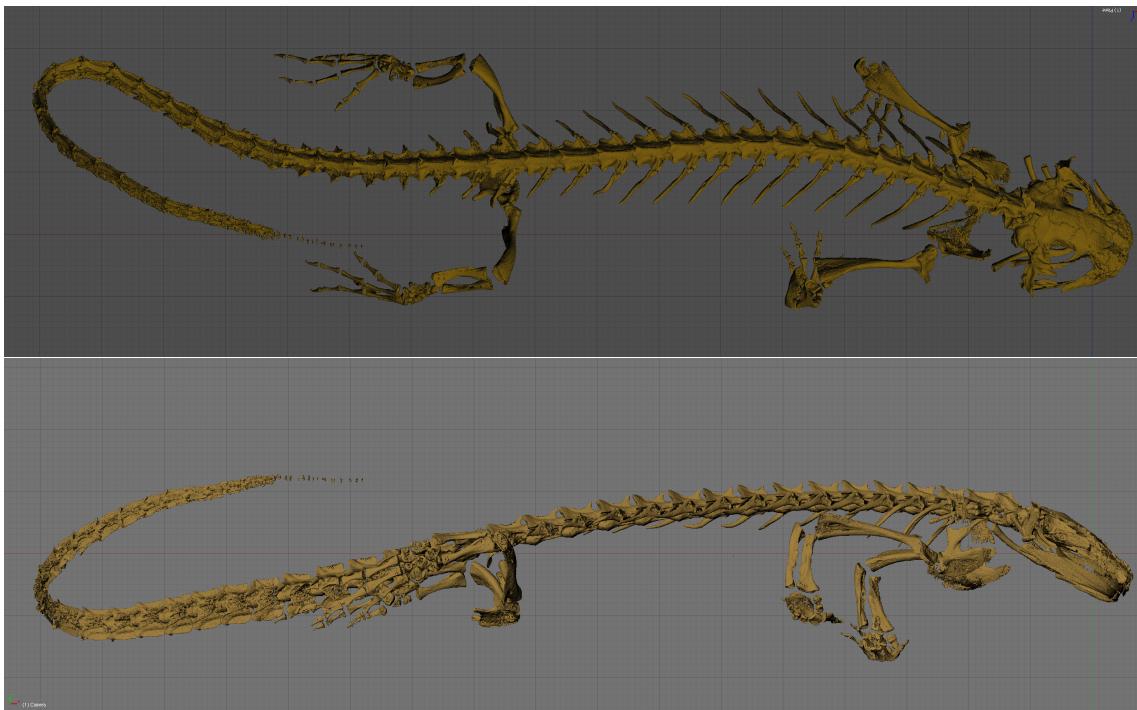


Figure 3.1.: Salamander skeletal layout, provided by Kostas

Before modeling the salamander, we should be aware of the internal interaction taking place in a real salamander and many studies have been pursue about the role and repartition of each individual muscle in the all body dynamic [24, 2, 7, 14, 47, 5, 13, 51, 21]. There is more than 10 muscle per vertebra as display in figure 3.2 (A).

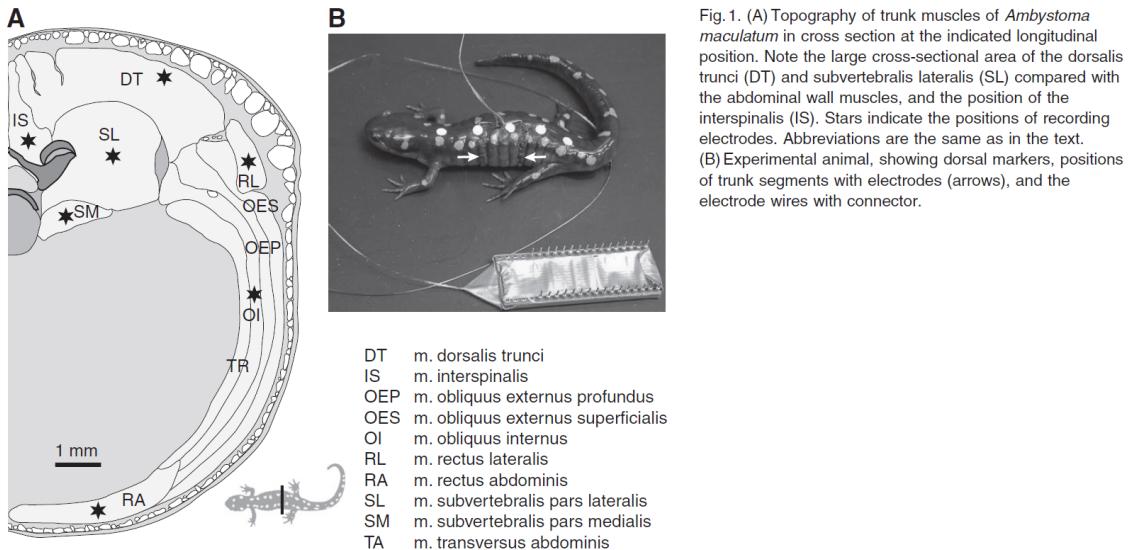


Figure 3.2.: Salamander muscle repartition from [21]

The joint model represent the way muscles interact with the all body. Mainly with the bones but also with others tissues and others muscles. In fish muscle are mainly attached to a fine layer of tendonious tissues called myomere as display in figure 3.3.

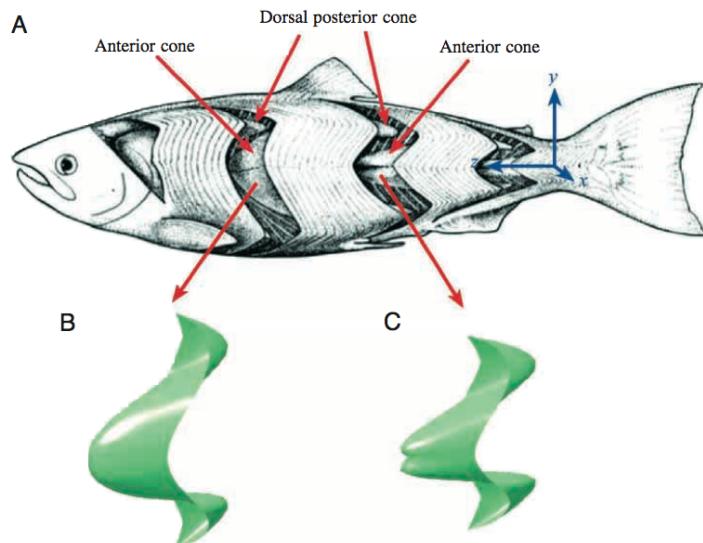


Figure 3.3.: Fish myomere from [51]

The complexity of such a structure is extremely difficult to model. Even if we wanted to reproduce it, we do not have enough data about the muscles, tendons, tissues properties. This kind of model has never been done and is faraway to be. It ask a lot of work on the biology side, a very strong mathematical background and an amazing computational power. Therefore people who models salamander swimming use much more simpler model. Simpler but able to reproduce the main characteristic of a swimming gait.

3.1.2. Previous Work

Many researchers worked on the salamander modeling, more or less complex musculoskeletal model have been created [30, 23, 35, 58, 43, 11, 16, 15]. I decided to present only one which is the one I used,

this model is simple, easy to understand and can model very accurately the body dynamics.

This model have been used by O. Ekeberg (figure 3.4) and A. J. Ijspeert (figure 3.5) and many others.

ates x_i , y_i , and φ_i : x_i and y_i denote the position of the midpoint of the link, while φ_i denotes the angle from the x -axis (see Fig. 2A). This set of parameters constitutes

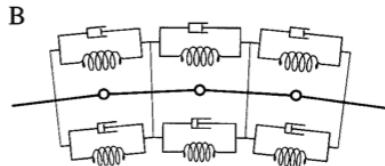
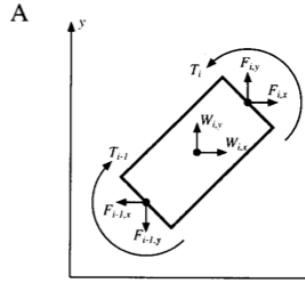
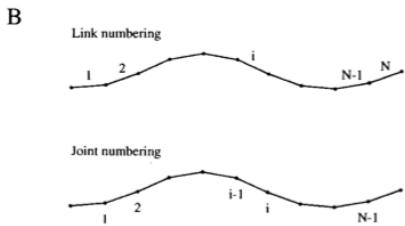
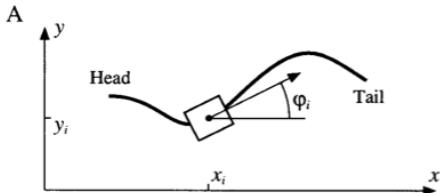


Fig. 3. **A** Forces acting on link i : muscular torques T_i and T_{i-1} , water forces \vec{W}_i , and inner forces from neighboring links \vec{F}_i and \vec{F}_{i-1} . **B** The

Figure 3.4.: Muscle/Joint Model used by O. Ekeberg from ??

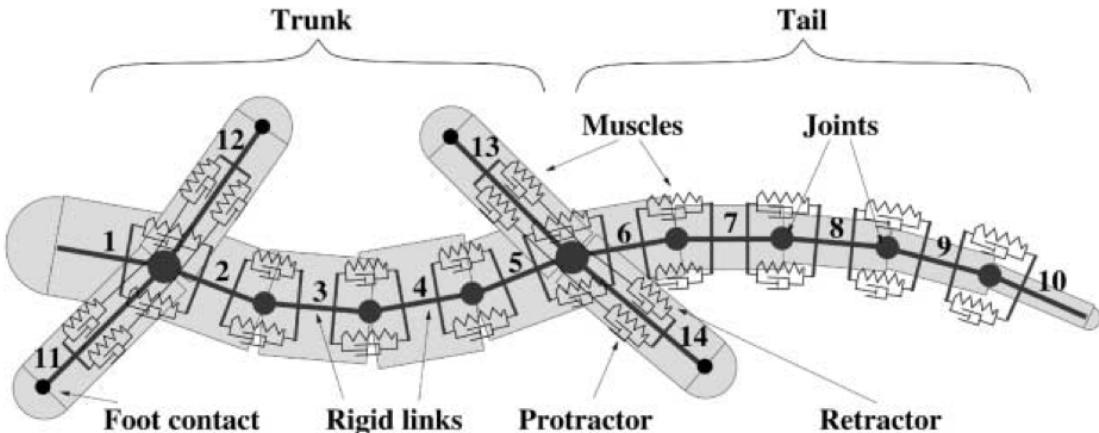


Figure 3.5.: Muscle/Joint Model used by A. J. Ijspeert from ??

This model is a planar model, composed of many segments linked together by a rotating joint. This structure allows the segment to form all the shape needed for anguilliform swimming, if the number of segments is reasonable. Moreover this structure is exactly how the robot is built. As real vertebrae, those segments are composed of ribs, one on each side, of equal length. The muscles will be attached to these ribs. One segment is defined as follows in figure 3.6 and are linked together by muscle as in figure 3.10.

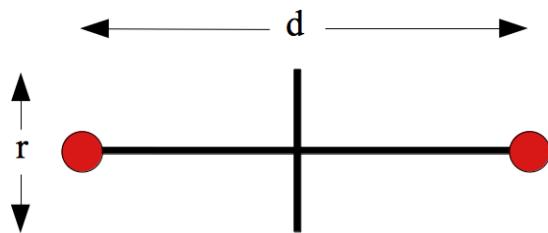


Figure 3.6.: One segment

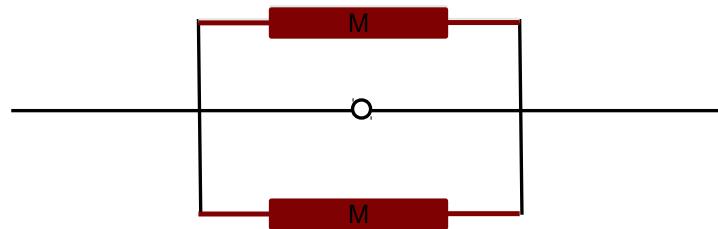


Figure 3.7.: Simple Muscle/Joint

In O. Ekeberg and A. J. Ijspeert works, the muscle model was a linear muscle model composed of a shiftable spring constant and a damping. We can represent this model as in figure 3.8.

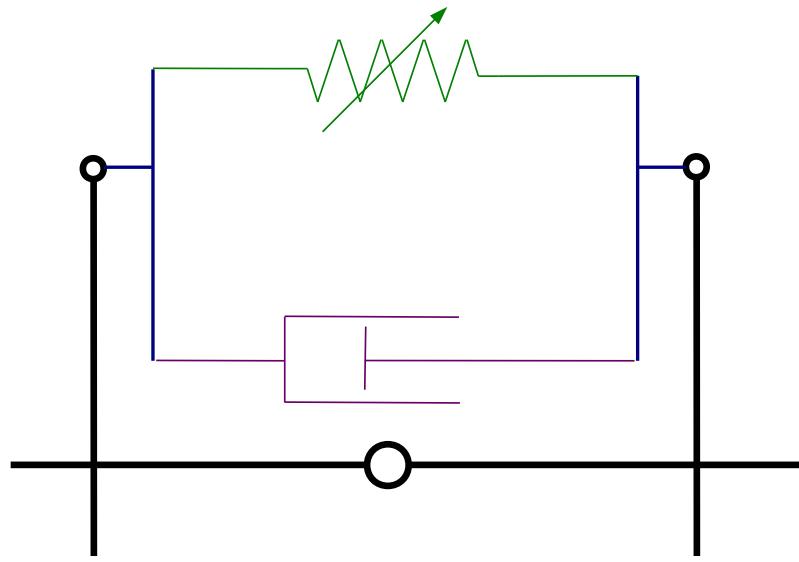


Figure 3.8.: Linear Muscle/Joint

The mechanical structure remain but the muscle model will be improved to match better real properties of muscle. (figure 3.9).

3.1.3. Ideal muscle/joint model

Note that ideally, the muscle/joint model should include tendons, spring and damping constant from the tissues. And should look like figure 3.9.

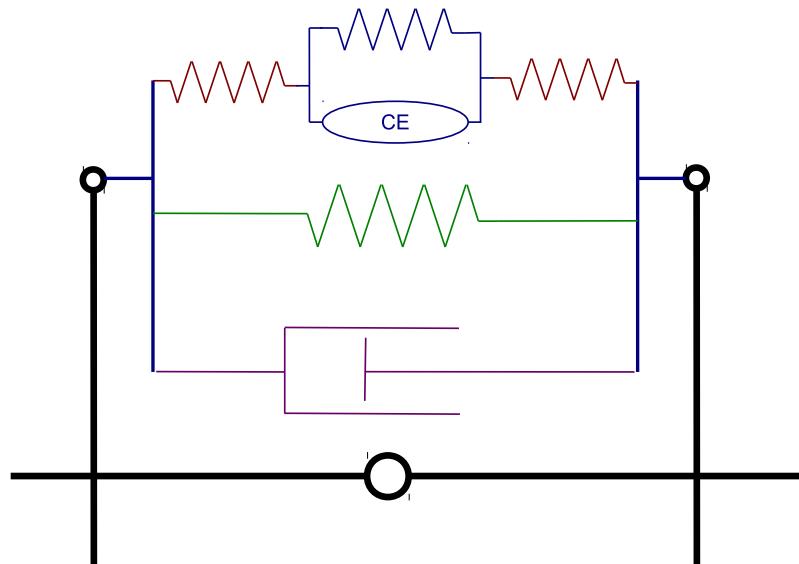


Figure 3.9.: Complex Muscle/Joint

But this model is far too much complex to tackle the problem for two reason : (1) the optimization will have to deal with a lot more parameters and (2) if we get result we will not be able to know what behavior is due to the tendon, to the spring or to the muscle. By consequence, I decided to start by the figure 3.10 configuration.

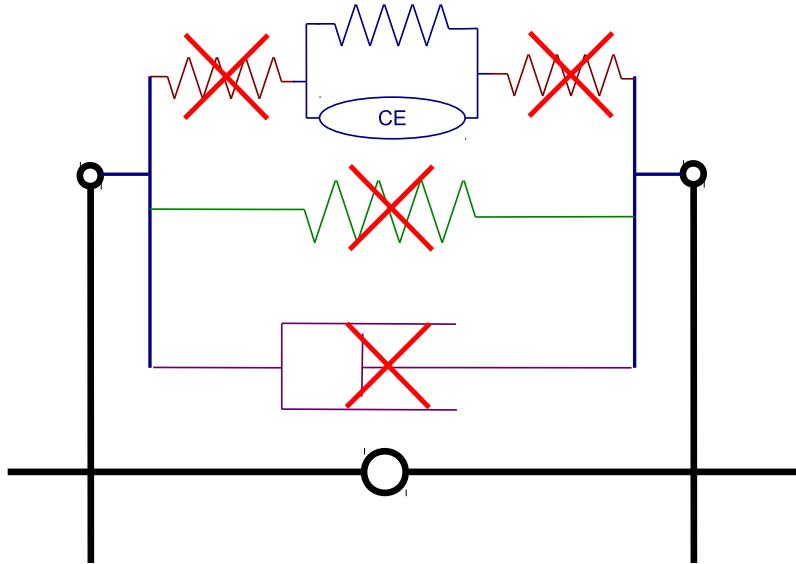


Figure 3.10.: Implemented Muscle/Joint

3.2. Muscle model

To model a muscle we need to know how it works. Muscle is long time interrogation and more than needed studies are available [32, 60, 38, 44, 40].

3.2.1. What is a muscle ?

In this subsection, I will only present the minimal knowledge and important concept needed to understand the muscle properties. For more details, I suggest to read the good muscle summary done by S. Berger in his master thesis report [46]. For deeper understanding refer to the previously given references.

The main concept is that a muscle is built of thousands and thousands of tiny contractile units called sarcomeres (figure 3.11). The properties of one muscle is the addition of the properties of those small units. To build a strong muscle, sarcomeres are placed in parallel and to increase the stretchability sarcomeres are serially connected. Thus to build a model of a muscle we need to know the properties of one sarcomere.

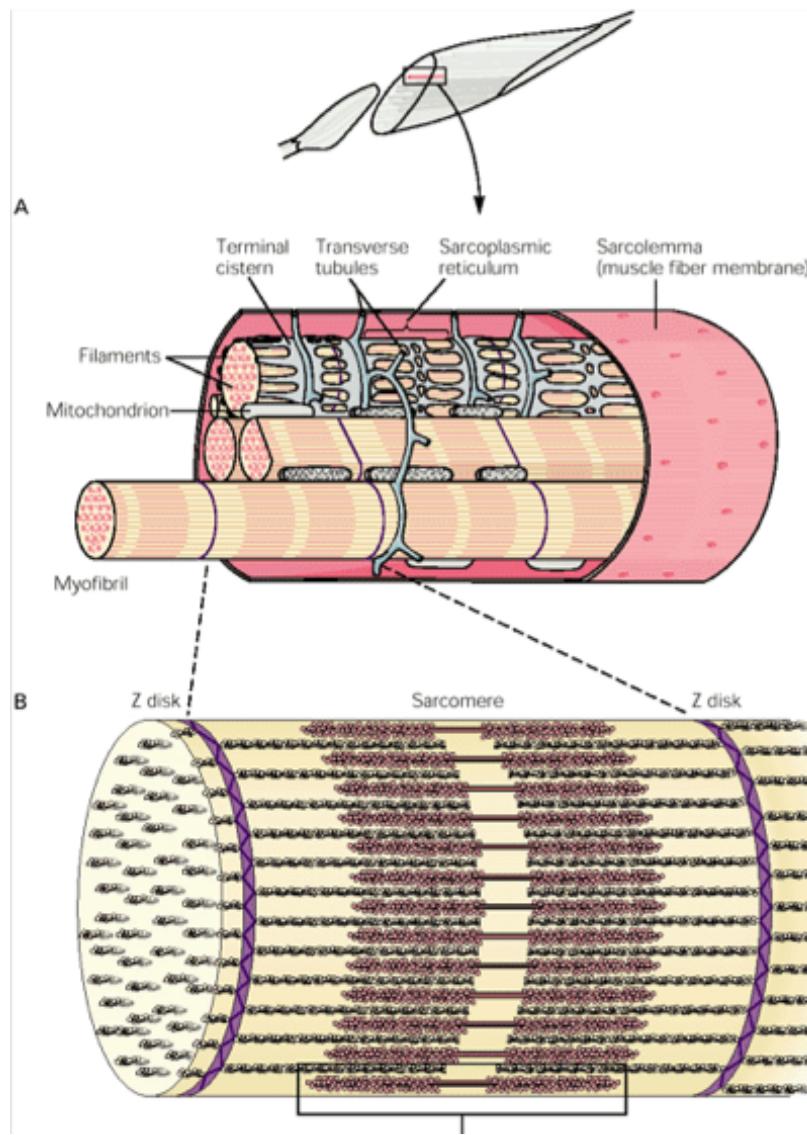


Figure 3.11.: Muscle structure from [38]

One sarcomere have three main properties :

- Passive property : in the sarcomere tiny proteins act as springs and generate a passive force if elongated (cf figure 3.12)
- Active force to length relation : when activated a muscle produce a force proportional to the activation level but dependent on the current length state of the muscle. This bell shape relation is linked with the surface of interaction between actin and myosin (cf figure 3.12).
- Active force to velocity relation : this property is what give the intrinsic stability role of muscle, it act like an imbedded reflex (see [29]). By itself a muscle which is extending will have more force for the same activation level and in the same other condition. In contrary, when the muscle is shortening, the muscle will produce less forces in the same condition. Depending on the velocity of muscle contraction or lengthening, the sliding motion between actin and myosin change their interaction condition (cf figure 3.13).

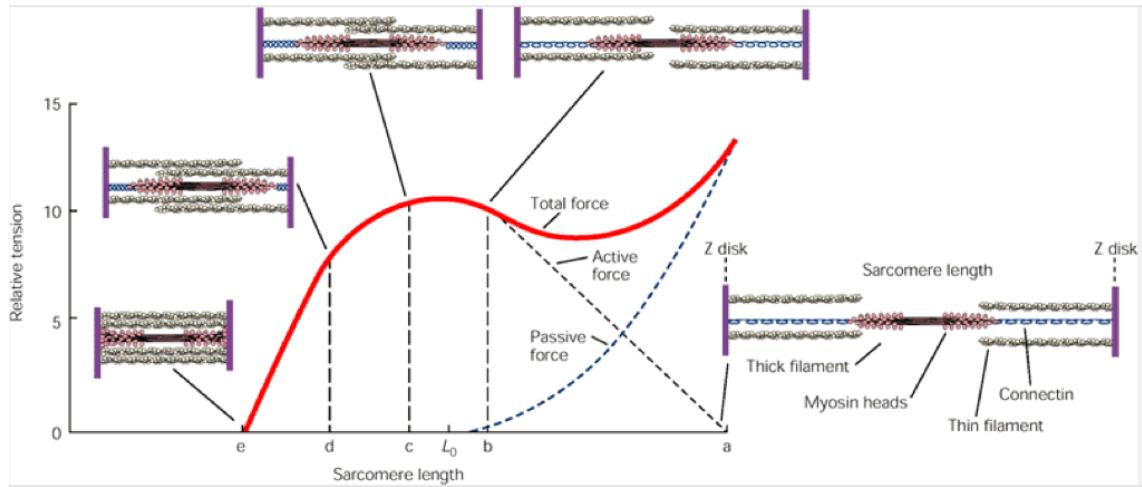


Figure 3.12.: Sarcomere properties from [38]

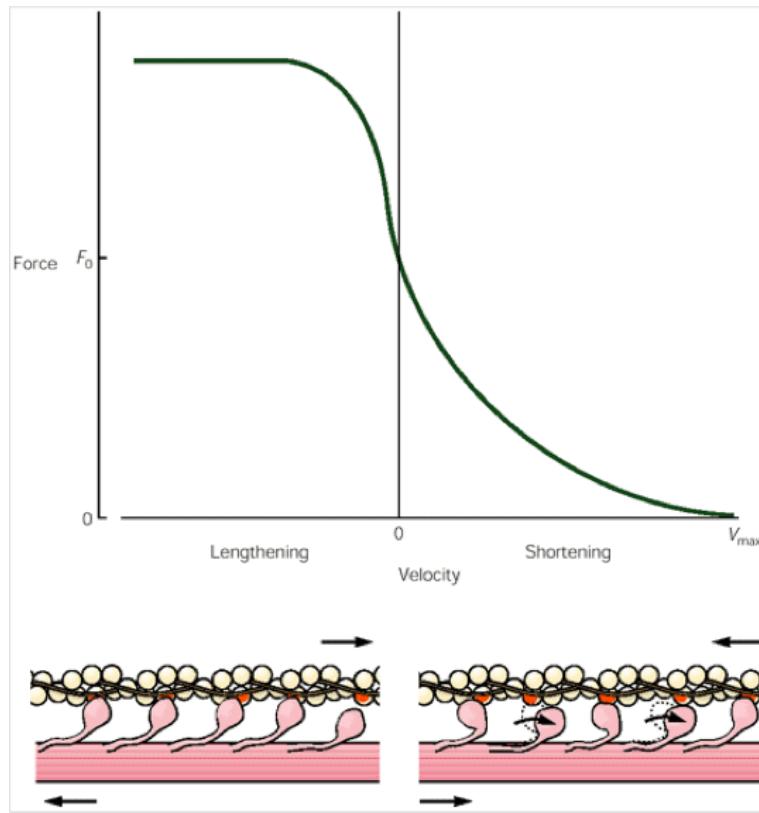


Figure 3.13.: Sarcomere force velocity property from [38]

Modeling a muscle resume in modeling the sarcomere properties describe here. This is the topic of the following subsection.

Note : Muscle have almost universal properties. Very little changes in sarcomere properties occur from salamander to human.

3.2.2. Muscle modeling

There is mainly three different muscle model which have been develop. Each of them for different purpose and have increasing degree of complexity.

The first model is the linear one. This model is constitute of two terms, a classical damping and a spring terms where the traditional ax function is change with with a $ax + b$. The b terms is proportional to the level of activation of the muscle. We will not use this model because the idea of this project is to use a realistic model, which should (1) enable us to estimate the energy consumption based on biologically relevant models and (2) bring a realistic interaction between EMG command and kinematics

The second model is the so called Hill muscle model. This model is based on the description from section 3.2.1, and reproduce the non linearity of the muscle properties.

The third one is the more complex, it is called the Huxley model, it take in to account the all dynamics of sarcomere and is very complex and costly to compute. It is mainly a tool for biologist which want to studies particular muscles in details.

According to [59] the choice between Hill or Huxley model for mechanical simulation is obvious. The Huxley model is too computationally costly and do not improve a lot the general accuracy of the model, therefore, the Hill muscle model have been chosen.

A hill model is usually writen in this terms (figure 3.14):

$$(v + b)(F + a) = b(F_0 + a)$$

Figure 3.14.: Hill muscle model equation from [32]

But the parameters are difficulty identifiable and expressible into human understandable value. This is why others model based on the same muscle properties have been developed. I decided to use the following model which have been mainly use for human muscle modeling. This choice has been motivated by three main reason :

- An other master student previously in the lab was using it since 4 months which is a good point to avoid getting stuck on stupid issues.
- Muscle properties are universal, the difference between frog, rat and human sarcomere are small and negligible for our dynamical use.
- This model have the advantage to express all the properties of the muscle in clear and simple terms.

The model can be schematically represented as follow :

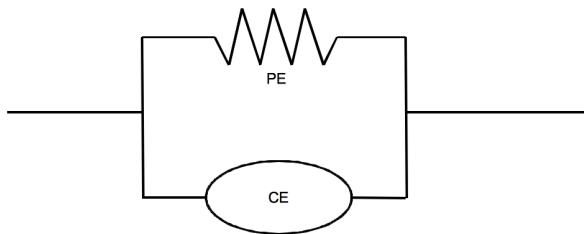


Figure 3.15.: Muscle model

The spring represent the passive properties of the sarcomere and the contractile element the active one. The total force of a motor unit is define as the sum of the spring and the contractile element force.

The model equation are as follow, figure 3.16 represent the characteristic of the spring. Figure 3.17 the force length relation, and figure 3.18 the force velocity relation. Finally figure 3.19 represent the contractile element force, depending on force length, force velocity, activation and muscle maximal force.

$$F_{HPE} = \begin{cases} F_{max} \cdot \left(\frac{l_{CE} - l_{opt}}{l_{opt} \cdot w} \right)^2, & \text{if } l_{CE} > l_{opt} \\ 0, & \text{else} \end{cases}$$

Figure 3.16.: Muscle passive property

$$f_l(l_{CE}) = \exp \left(c \cdot \left| \frac{l_{CE} - l_{opt}}{l_{opt} \cdot w} \right|^3 \right)$$

Figure 3.17.: Muscle force length relation

$$f_v(v_{CE}) = \begin{cases} \frac{v_{max} - v_{CE}}{v_{max} + K \cdot v_{CE}}, & \text{if } v_{CE} > 0 \text{ (shortening)} \\ N + (N - 1) \cdot \frac{v_{max} + v_{CE}}{7.56 \cdot K \cdot v_{CE} - v_{max}}, & \text{else (lengthening)} \end{cases}$$

Figure 3.18.: Muscle force velocity relation

$$F_{MF} = F_{max} \cdot A \cdot f_l(l_{CE}) \cdot f_v(v_{CE})$$

Figure 3.19.: Muscle contractile element force

The meaning of each variable is easily understandable :

- F_{max} (N) is the muscle maximal force, it is proportional to the number of sarcomere associate in parallel.
- l_{opt} (m) is the muscle optimal length, it is proportional to the number of sarcomere in serially connected.

Others parameters are fixed and are the muscle properties :

- v_{max} (Lopt/s) is hte maximum speed of muscle fiber. This value is about 2 for mixed skeletal muscle of many species of fish (see [50]).
- w is the stretch constant is usually set to 0.56
- c is a constant equal to $\ln(0.05)$
- K is the curvature constant equal to 5
- N is the Relative amount of force reached at a lengthening velocity v_{max} and is equal to 1.5

The model equation are of course dependent to the current length of the muscle l_{CE} , the current velocity of the muscle v_{CE} and the activation level A .

Using Matlab we can visualize this model (figure 3.20). By eyes, those graph look to correspond to the muscles properties detailed in section 3.2.1.

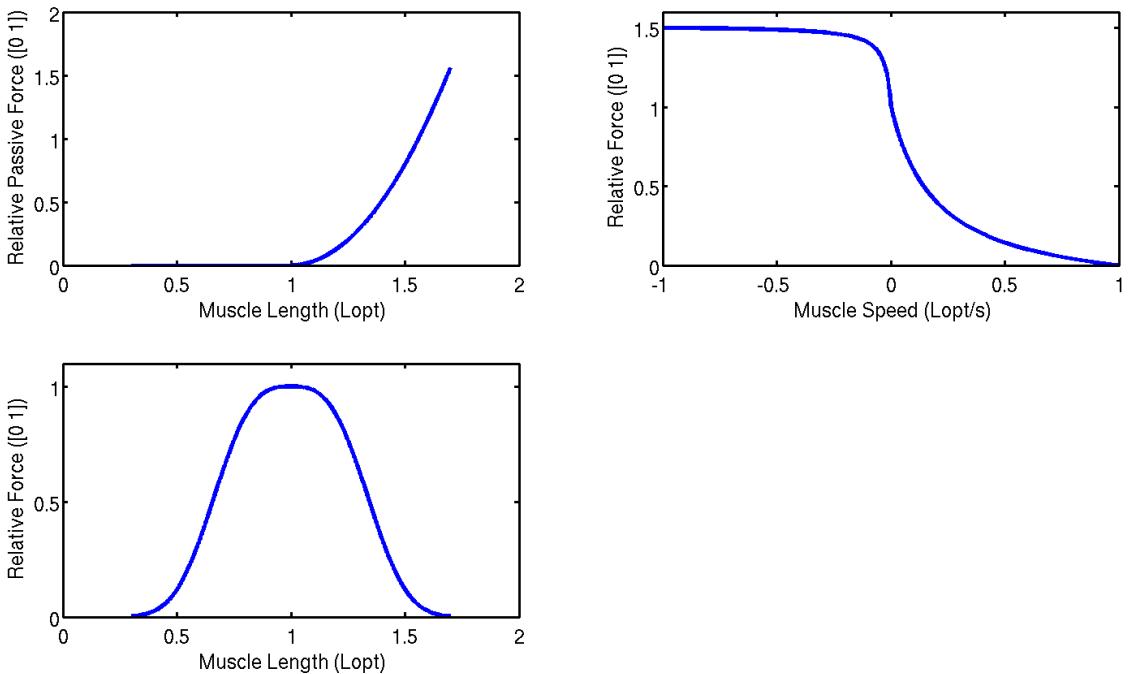


Figure 3.20.: Muscle model Matlab plot

The model is finally set, but the activation variable of the model, which is the input, is not directly the EMG value. There is a complex transformation from electrical stimulation to chemical action in the muscle. This activation have been modeled in many way and the next subsection describe quickly the model I use.

3.2.3. Muscle activation

A muscle is activated electrically by motor neurons but it is not the electricity by itself which contract the muscle. In one word, electrical activation discharge some Ca^+ ion which flow down to sarcomere and produce the actin to interact with the myosin. For details about this interaction you can read the reference from section 3.2.1. in terms of physical consequence, it mainly delay a bit the activation from about 10 ms, but an important factor is that the raising time constant is different form the decrease time constant. In short, the stimulation to activation follow a first order differential equation with variable time constant. I decided to use the Zajac model because it is the simplest one to model this time constant variation, the model equation is as follow :

$$u(t) = \text{electrical excitation (EMG)}$$

$$a(t) = \text{chemical excitation}$$

$$\frac{da(t)}{dt} + \left[\frac{1}{T_{act}} \cdot (B + (1-B) \cdot u(t)) \right] \cdot a(t) = \left(\frac{1}{T_{act}} \right) \cdot u(t)$$

$$0 < B = \text{const} < 1$$

Figure 3.21.: Muscle activation to stimulation function from [60]

The B variable determine the ration between the raising and the decreasing time constant. Classical value for muscle are $B = 0.2$ and $Tact = 10 \text{ ms}$, which are the one I used for my model.

3.2.4. Energy estimation

Using a realistic muscle model enable us to estimate the metabolic energy consumption of muscle as biologist do. The model I decided to use is taken form [8], because his parameter are close to the one used by the muscle model. I will comment the problem I had using this model in the next chapter about the optimization.

3.3. Constrain imposed by the robotic platform

My goal is not to stay on simulation software but to be able to switch on real robot, this robot has many constrain and we have to think what are the consequences for our model. The salamander robot has two main characteristic which constrain our modeling, the segmentation and the maximum operating frequency.

3.3.1. Effect of robot segmentation

As we have see before, the spine of a living salamander is composed of 40 vertebra, while the robot has only 8 segment (excluding the head) and a piece of plastic to complete the tail. That made a ratio of 3.5 vertebra for one robotic segment. But what are the consequences of this re-segmentation ?

Intuitively, we can grasp the effect of segmentation by thinking in angular position. The more the segment are small the less they angle amplitude will be. In others words, the more the segment are long the more they angle amplitude will be. Despite this effect, the general characteristics remain the same, a increase of the amplitude from head to tail and the same phase lag relative to body position. But the muscle will interact in a different way with the joint model.

The first idea is to quantify this angle amplitude difference. As it has been perform in section 2.1.3, we will segment again our X-Ray data but this time as a real salamander. with 16 vertebra in the trunk. The median amplitude difference are display in figure 3.22.

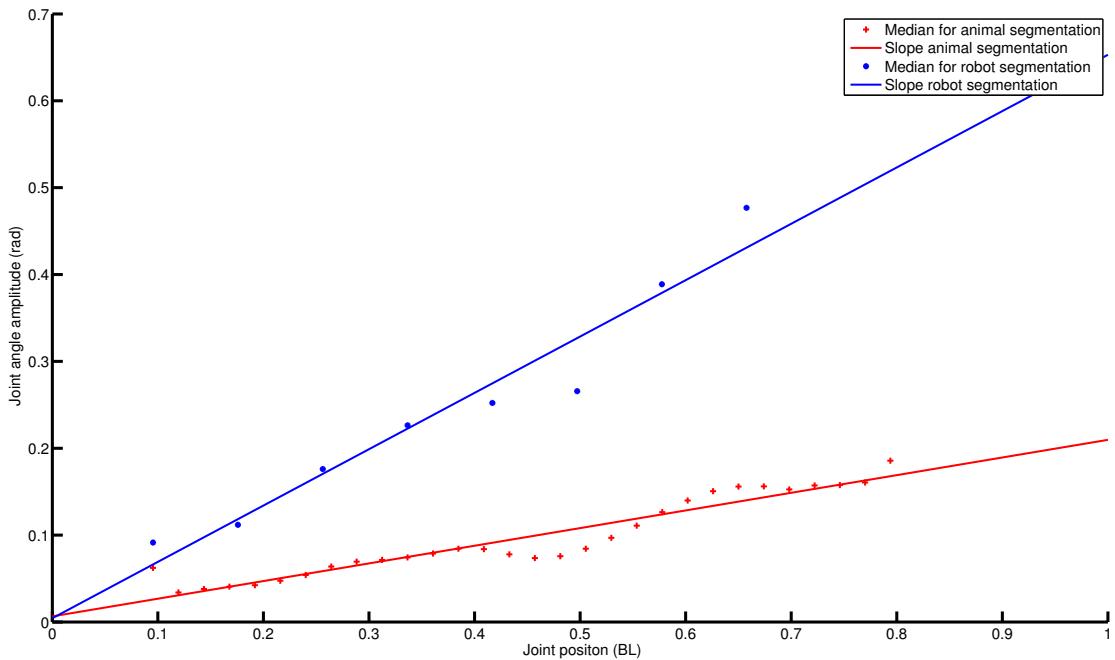


Figure 3.22.: Joint angle amplitude increase due to segmentation

The slope ratio between robot and real segmentation is equal to 0.32.

Note that if we cut the spine in only two segments, it doesn't make sense and the effect described here will no more be correct, for each specific case, specific segmentation, all this analysis should be done again

Joint model change

The angle amplitude increase changes the way muscle interact with the joint model, the figure 3.23 illustrate this change.

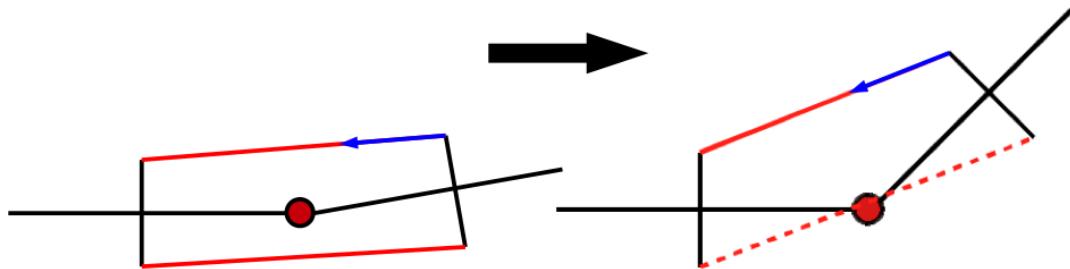


Figure 3.23.: Joint model segmentation effect

With a real segmentation the action of the muscle is almost always parallel to the body axis but with the robot segmentation this approximation can no more be true. Therefore I decided to change my joint model in order to keep the same muscle to torque relation. The new model assumes a bending of the muscle fiber in order to keep the muscle parallel to the body axis as shown in figure 3.24.

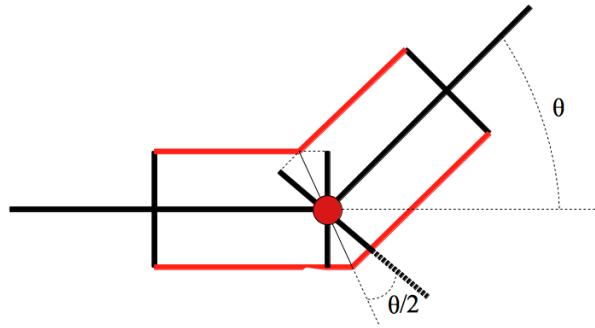


Figure 3.24.: Joint model

When we are dealing with muscle to body interaction, we do not talk about force created by the muscle but the torque created. An other advantage of this model is that the torque is easily deducible from the muscle force (cf figure 3.25).

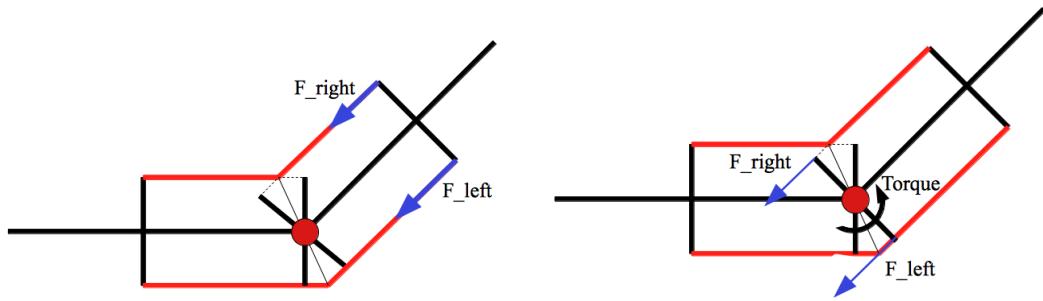


Figure 3.25.: Muscle/Joint Model : Force to Torque relation

The torque due to the muscle force is given by the following equation 3.1.

$$Torque = \frac{(F_{right} - F_{left}) \times r}{2} \quad (3.1)$$

The current muscle length of the muscle is also easy to compute by the following equation 3.2. \pm sign is to represent left and right side difference.

$$l_{CE} = d \pm r \times \tan(\theta/2) \quad (3.2)$$

The current speed will be deduced from the difference between previous and current muscle length divide by the time step.

Muscle model change

The angle amplitude increase have direct effect on the muscle working zone. Due to robot segmentation the muscles of both side will be more extended than with a real segmentation. To be clear, we have seen before that a muscle can work on a limited length range around its optimal length. This is more than logical, a muscle cannot compress to a 0 cm length neither extend to the infinite. From this observation, three different directions are possible (cf figure 3.26).

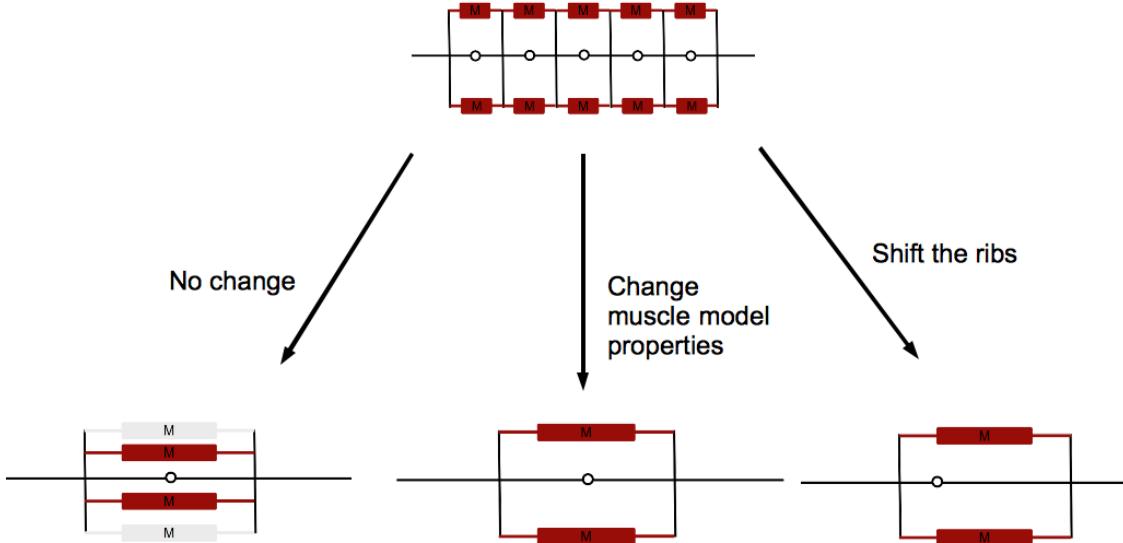


Figure 3.26.: Muscle/Joint ReSegmentation Choices

- Keep the muscle model as it his. The muscle will tend to be attached closer to the body axis and will result in an increase of the muscle maximal force which is not very relevant.
- Change the stretchability variable w of our muscle model. Using this trick, the muscle model will no more be realistic in quantitative terms, but its general behavior will remain in the same in our context.
- Shift the ribs, for now the ribs are situated in the middle of the segment. But when nature need to increase the amplitude of movement of any body part, the attaching point are shifted. This is the only possibility the nature have because muscle are real and cannot compress to 0 length or extend to two time its optimal length.

The choice one is discarded as well as the third. I decided to discard the third possibility because it will add a new unknown to our system (the ribs shift). Therefore how to change the muscle model that it can perform on a wider length change. The parameter w is responsible of the stretchability and by changing its value we can have the muscle model operate in a wider zone.

To define the new value I decided that one muscle attached to two segment should apply the same force with the original $w = 0.56$ and an angle corresponding to a real segmentation as with the new w and the robot segmentation angle. For the sake of simplicity I assume the same model for both segmentation.

Based on equation from figure 3.17, and equation 3.2, the following equation need to be verified :

$$\exp(c \mid \frac{l_{CE1} - l_{opt}}{l_{opt}w_{0.56}} \mid^2) = \exp(c \mid \frac{l_{CE2} - l_{opt}}{l_{opt}w_{???}} \mid^2) \quad (3.3)$$

Where l_{CE1} et l_{CE2} respectively depend on an angle θ and a $\frac{1}{0.32}\theta$ bigger angle. Remember realistic segmentation led to angle 0.32 time smaller than with the robot segmentation. To simplify the equation, I consider that the muscle optimal length is equal to the length of one segment ($l_{opt} = d$). The ribs length is constant as the segment length. Using all those tricks, the new w can be computed this way :

$$w_{???} = \frac{\tan(\frac{k\theta}{2})}{\tan(\frac{\theta}{2})} \times \frac{d_{realsegmentation}}{d_{robotsegmentation}} \times w_{0.56} \quad (3.4)$$

Where k is the angle ratio from real segmentation to robot segmentation equal to 0.32. The ratio $\frac{d_{realsegmentation}}{d_{robotsegmentation}} = 4.5/16 = 0.2812$. Due to the \tan function, the result can not be generalized. But we can estimate the value for discrete values, the real segmentation angle range from 0.03 to 0.19 radian. Figure 3.27 show the needed w change for several values in this range.

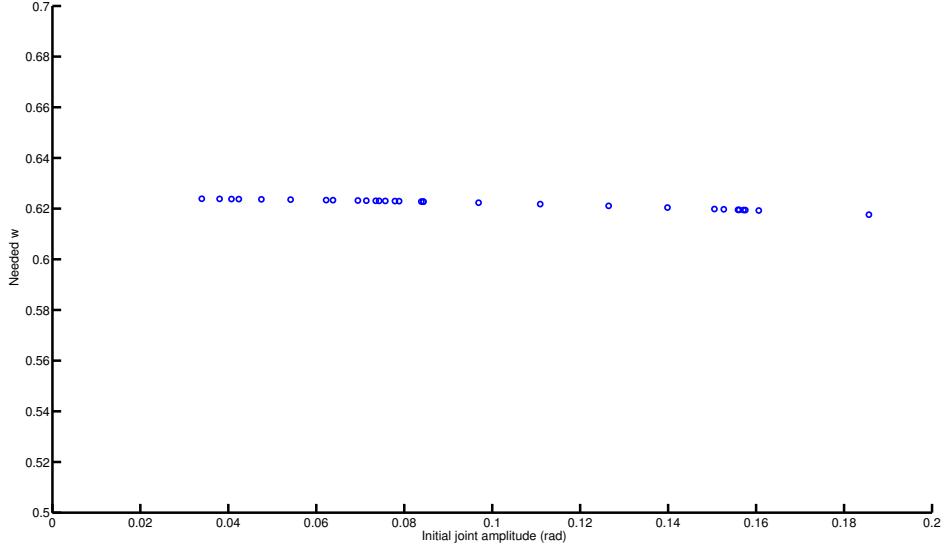


Figure 3.27.: New w value in function of the angle

Fortunately, the variation is very small, the average value is 0.62. This new value changes the muscle model as in figure 3.28.

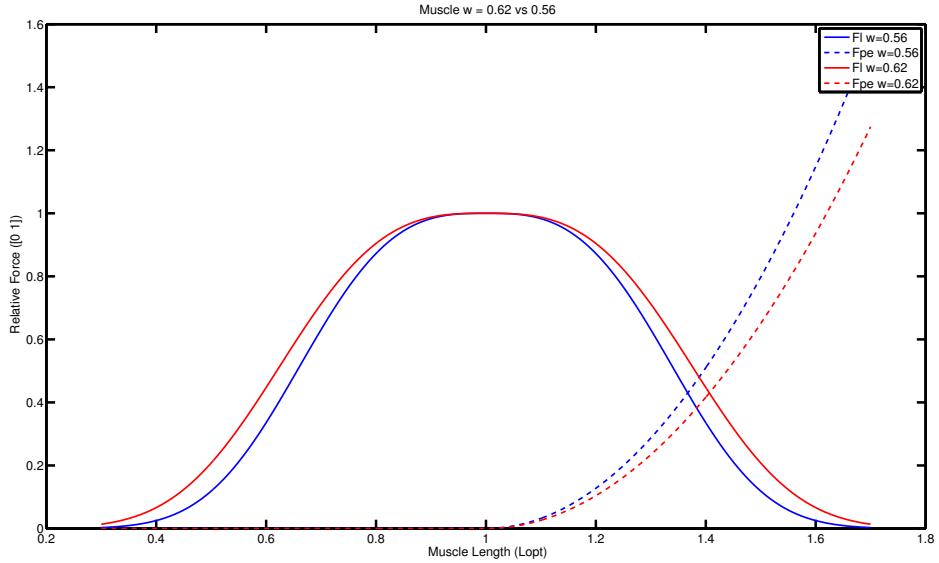


Figure 3.28.: The effect of changing the w muscle parameter

As expected, this new value for w allows the new muscle to operate in a wider zone than a normal muscle.

3.3.2. Effect of frequency limitation

The robot can not swim to the same frequency than salamanders and is limited to about 1.2 Hz. The frequency have no effect on the working zone but on the shortening or lengthening speed relation should evolve. The muscle contracting velocity will be lower than in the animals, so in a first approach I should change the v_{max} parameter. But in the mean time I am increasing the angle amplitude, which mean that for the same frequency, a muscle will extend more and by consequence increase his speed of lengthening (idem shortening). The median salamander swimming frequency was about 3.5 Hz in the tracked movies and I am bringing it back to 1 Hz. The angle amplitude increase is $\frac{1}{0.32} = 0.313$. One effect cancel the others by consequence no change have been applied to v_{max} which remain equal to 2 (l_{opt}/s).

3.4. Summary

Assuming a symmetry in muscle from left to right side, the remaining parameters for each joint are :

- F_{max} the muscle maximal force
- l_{opt} the muscle optimal length
- r the joint width (length of the ribs)

3.5. Discussion

Even if I am sure the optimization process can be done and will give good result in terms of accuracy without this muscle/joint change, I took the decision to applied and justify those choices. To explain my approach from the beginning I wanted to use a reverse approach, starting form a real segmentation salamander, tune the muscle parameter and find a way to change the muscle parameter and the joint model to keep the same behavior for the robot segmentation. To this end a 30 segment robot plus tail have been created, but the computational cost was too heavy and optimization took two week, which force me to use the inverse approach presented here. Therefore to validate my choice, the 30 segmented model should be tested with the parameter we will obtain from the optimization scaled to the new segmentation. For example, if one segment is 3 time smaller, we should came back to the original joint model and muscle w parameter and use the same F_{max} and l_{opt} parameters divided by 3 while keeping the same value for r . Unfortunately I did not have time to perform this test.

4. Optimization

The optimization process consist to tune the open parameters of our model in order to obtain the correct output for a fixed input (figure 4.1). Data collected in chapter 2 are respectively input for the EMG and desired output for kinematics X-Ray data. The optimization parameters for each joint are :

- F_{max} the muscle maximal force
- l_{opt} the muscle optimal length
- r the joint width (length of the ribs)

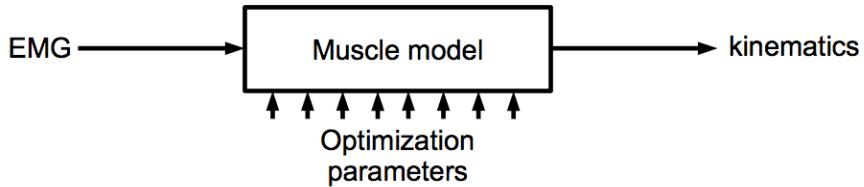


Figure 4.1.: Optimization setup

4.1. Particle Swarm Optimization

PSO is the optimization algorithm chosen, not because this algorithm is particularly performant but because it is widely use at BIOROB and many useful tools have been developed. For detailed information about PSO, the reader should read [39]. To simply describe the algorithm, its a way to explore the parameter space using particle. Particle can be seen as little ball displaying randomly on the parameter space. At the starting time, the first iteration, each one of them have one speed in each dimension of the parameter space. Using the fitness function, we can run all the experiment corresponding to the particle position in the parameter space and estimate the value of the fitness function which add one other dimension. Then we update the particle speed in order to move a bit in the direction of the best particle but also in the direction of the nearest best particle and also to keep a bit of original speed. This is why this algorithm name include *Swarm*, because comportment of particle depend on the behavior of the others. We can imagine many and many version off how to update those speed, and I am sure a lot have been done on this topic, but it is out of my concern in this project.

4.2. Webots Model

Webots [45] is a simulation tool designed at EPFL for robotic simulation. It is a powerful tool using the open source dynamical engine ODE. The experimented user have full control of the physic of the world. I have been using a Webots model of the robot developed at the lab. The body to water interaction is very advance and based on M. Porez thesis work [12]. However one element was missing, the tail which is present on the lab robot and which is a very important part to have a good swimming dynamics. Therefore the first step is to model this tail.

4.2.1. Tail modeling

The robot tail is made of plastic, a layer of plastic of 250x59x1 mm (l x h x d) rigidly attached to the last segment of the robot. The tail have been designed by hand and tune by Kostas. No data are available about its properties. And Webots, as powerful it is, cannot model the intrinsic elasticity of a materials. So my approach have been to model the tail as the robot body, in a segmented way. Assuming I can divide the tail in several segments linked together by a rotating joint, I could just characterize the passive spring constant of each joint. If enough segment are used, the resulting model should behave as the real tail.

To characterize those spring constant, the experiment is simple, using a dynamometer and a video camera I can track the bending and the force applied for any position. Figure 4.2 show the experimental setting. Note that the force applied by the dynamometer is located to the tip of the tail.



Figure 4.2.: Experimental setup for the tail characterization

Then using the same technics used for the X-Ray analysis, I can track, calibrate, reconstruct and segment the tail. Tracking point are :

- One point in the dynamometer axis to have the direction of the force.
- Tip of tail
- Start of tail
- 1 to 10 point on the tail, regularly spaced but not accurately placed.

I choose model the tail in 10 pieces. The more we use many segment the more the computational cost will increase and the simulation time with it, 10 is a good compromise. Once tracked, calibrated and segmented, for each frame we can deduce :

- the direction and location of the force
- the intensity of the force (N) by reading the value
- the angle between each segment (rad)

Using the direction and the intensity of the force we can deduce the corresponding torque (N.m) to each joint (figure 4.3). Doing so we assume we are in a static system, which is not perfectly true because I decided to work on movie and not on picture, tracked point are stop motion but this effect is negligible as I was moving slowly and the dynamometer resolution is low.

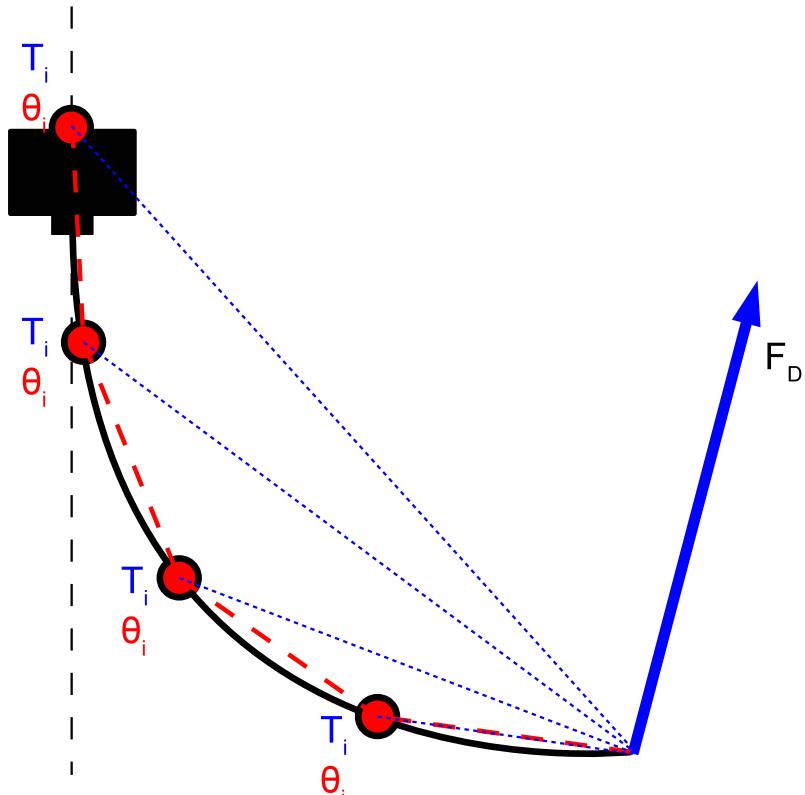


Figure 4.3.: Experimental analysis

For each joint, and each tracked frame I can plot the graph of torque versus angle. Which is the characteristic wanted for a spring joint torque. As expected the characteristic is linear. But the right and left side of the tail are not symmetric. One side is more rigid than the other, the difference is quite big and can be feel by hand. Reproducing this asymmetry is not at all needed and I decided to average the result. Figure 4.4 display all those informations. Note that the important information here is the slope, expressed in $\frac{N.m}{rad}$.

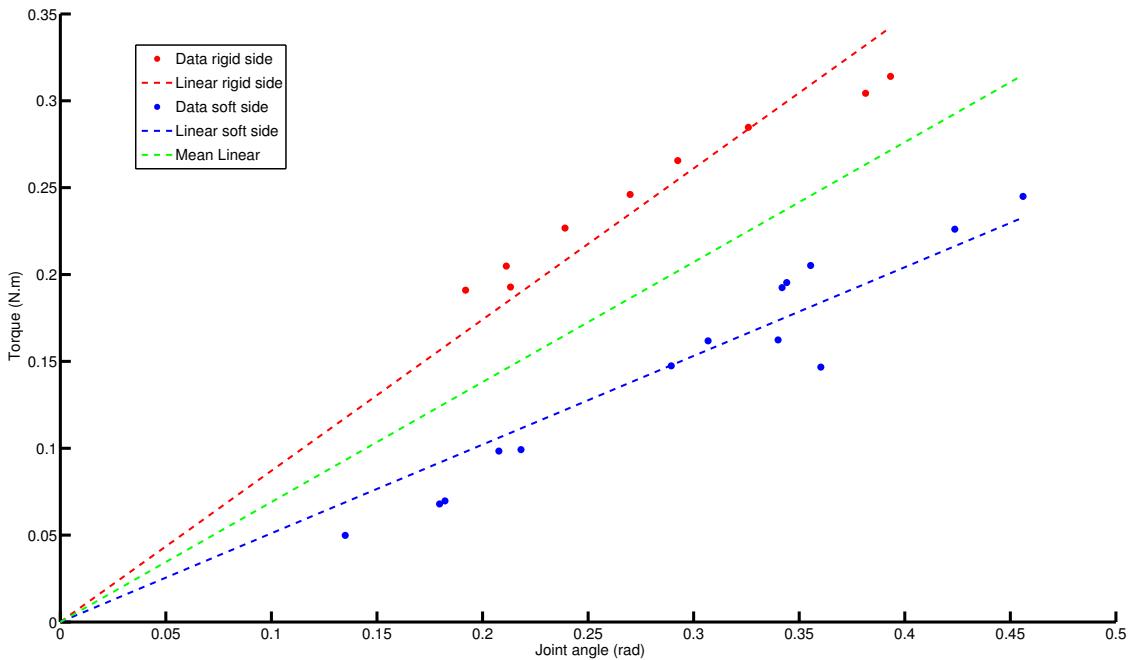


Figure 4.4.: Torque versus joint angle for tail characterization, rigid and soft side

Three tracking have been perform for each side of the tail, and the result have been averaged. Final tail characteristic look like figure 4.5.

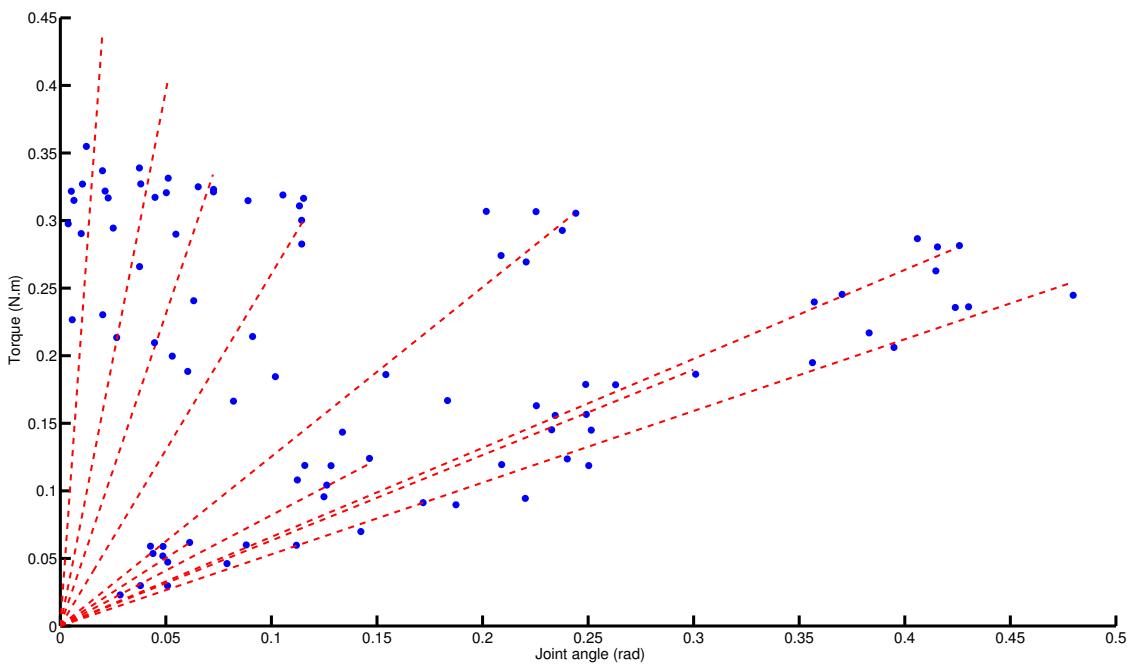


Figure 4.5.: Soft side all joint spring constant, tracking number 3

Table 4.1 resume the averaged spring constant for each joint.

Joint	1	2	3	4	5	6	7	8	9	10
Spring Constant ($\frac{N.m}{rad}$)	22.50	8.22	4.79	2.76	1.48	0.88	0.71	0.77	0.95	1.12

Table 4.1.: Median Spring Constant

Do not forget that there is a spring joint value which is very high on the fixation of the tail to the last module. That is why for 10 segment we have 10 value and not 9.

Using directly those value give instantaneous and astonishing result. The dynamic behavior of the tail is much more realistic than expected ¹.

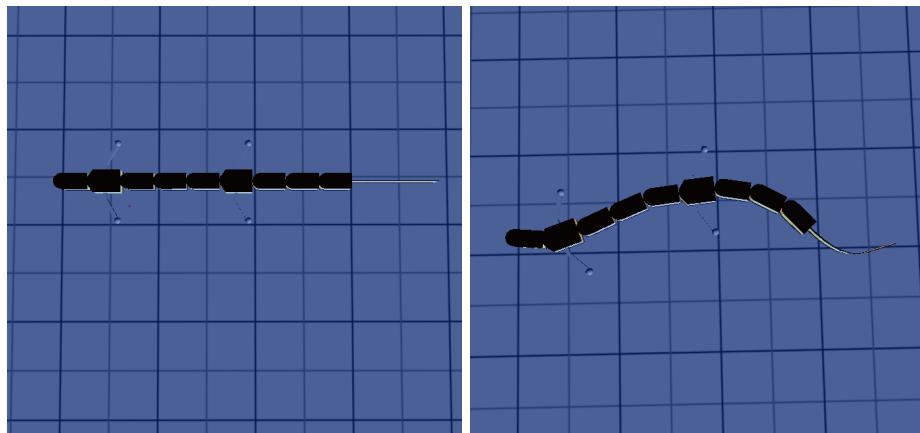


Figure 4.6.: The robot model on Webots with tail

About the damping coefficient once again I assume that the damping of water is a lot bigger than the one of the salamander tail joint. Characterizing the damping is a quite difficult task because is ask for dynamical experiment. It have been proposed to film the tail freely oscillating form an initial bended position to a straight position. We will in this way see the attenuation of the oscillation and could derive the damping factor form dynamical equation of the system. We all agree that with no damping the tail will never stop oscillate but the tail is simulated only in water for now and damping of water is more important.

Unfortunately I have not perform any other test than a visual comparison. And this kind of test are also very difficult to perform. First of all because it is needed to test two model at the same time, the water interaction model and the tail model, but also to perform the exact same experiment in the artificial word and in the real word which is very difficult and time consuming. I think visual test remain enough for this tail. We can see that the behavior is the same as in the robot, and the probability of getting such a visual result if both of the model are wrong is very low. Of course it is not a proof but I decided to handle with it.

4.2.2. Shaped robot

An interesting direction is to modify the robot morphology to test the effect of it. This change is easily feasible in Webots, so we decided to implement a realistic shaped robot with the same segmentation and the same body length but the scaled shape of a real salamander. I used again the data from the X-Ray moves, but this time form walking video were we can see better the bones and the animals shape (figure 4.7).

¹CD/Tail/tail_webots_slowmotion.mpeg compare to CD/Tail/run085005.mp4



Figure 4.7.: A walking gate X-Ray frame

From this frame we can track the side and the spine of the salamander and by using the same process as in 2 calibrate the data to determine the profile of the salamander. As always the reference unit is the body length. In order to be able to scale and reuse this profile I scale it between [0 1] for the body position but also from [0 1] for the body width. And store the body length to body with ratio of 0.13. Figure 4.8 show the tracking profile on Matlab and figure 4.9 the [0 1] profile information.

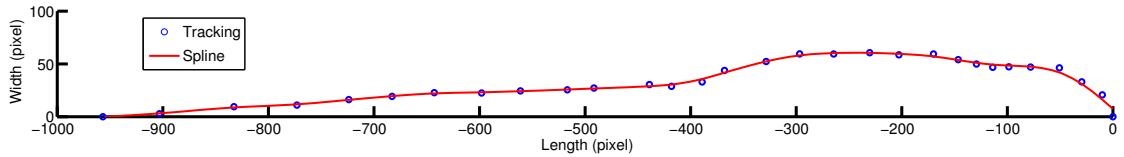


Figure 4.8.: Tracking profile

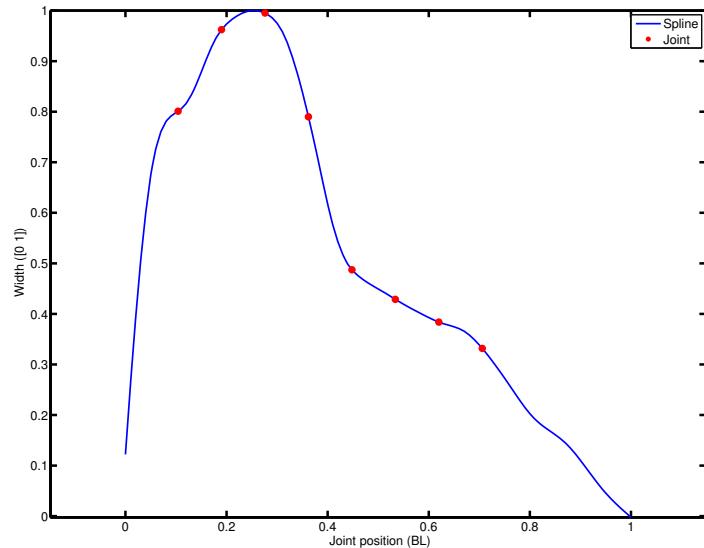


Figure 4.9.: Body length to body width profile of a real salamander

By averaging the width value for each body segment, we can deduce the resulting width scaled to the robot (table ??).

Segment	Head	1	2	3	4	5	6	7	8
Width robot (cm)	7.9	11.8	13.7	13.4	9.5	6.6	5.8	5.3	4.2

Table 4.2.: Width robot value

As show in figure 1.5 of chapter 1.4, the original width of body module is 4.5 cm and 6.5 cm for limb module.

The resulting robot is as follow :

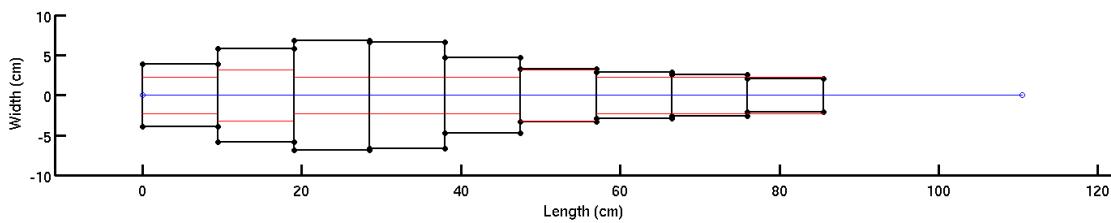


Figure 4.10.: Actual versus realistic robot dimension

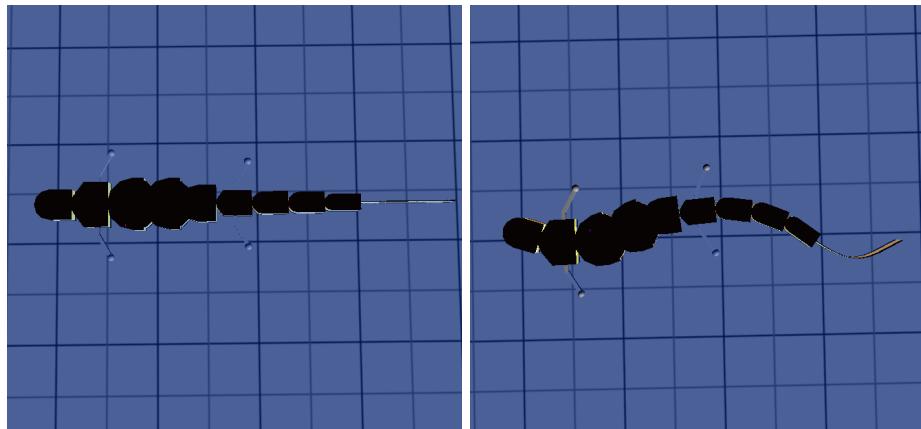


Figure 4.11.: Realistic shaped Webots model

Note : the mass move in proportion with the volume of the segment.

Note : we can not be sure that a real animal of the same size as the robot will have the same shape.

4.3. Firsts attempts

First of all, hand tuning of the open parameter led to interesting visual result ². The first optimization experiment almost no constrain the parameter space, and use a complex fitness function $\frac{\text{distance}}{\text{error} \times \text{energy}}$.

²CD/Opti/webots_first_attemp.mpeg

Where *distance* is the euclidian distance between starting and ending point, *error* is the sum of the difference between actual and desired angle and *energy* is the sum of the energy consumption of all muscle. The time step was 1 ms, optimization last 10 seconds, with 200 population and 200 iteration. With no surprise the result are out of any realistic behavior.

Several problems can be extracted from all this observation :

- The time step of 1 ms led to sawtooth profile on the joint angle velocity. If the time step is too big, the muscle model state is "moving" too fast, and the force applied by the muscle react violently to high speed variation. After some trial, a time step of 0.1 ms resolve this problem.
- The energy optimization is chaotic and not suitable for optimization. The model used compute the rate of energy consumption. There is two main terms, the work produce by the muscle and the energy lost as heat. The problem come from the work computation, it enable the muscle to produce negative work. This negative work is usually compensated by a velocity heat rate. But the optimization found some position in the parameter space where those value do no compensated at all and led to negative energy consumption. Which force me to stop using the energy consumption estimation. Moreover using energy in optimization is difficult in my case, as usual the robot start by not moving to reach a zero energy consumption, and if we credit the robot of an fixed amount of energy to spend, it will win energy instead of using it.
- The error estimation is not equally spread. As the joint from the end of the body perform greater angular amplitude than the first one, the optimization tend to optimize them in priority. To solve this problem I weighted the error by joint inversely proportionally to the joint angle amplitude.
- By discarding the energy in the fitness function, the distance is no more essential to the fitness function, what matter the most it the reproduction of the kinematics.

The final fitness function is $\frac{1}{\text{error}}$, time step is 0.1 ms. With this setup, the resulting optimization are interesting. To diversify the kind of optimization setup and to decrease the number of parameter (3×8 for now), I decided to add a last constrain.

Note : Nothing prove that the 0.1 ms time step led to realistic result. For sure the 1 ms time step is too big for our model especially concerning the muscle velocity estimation. Ideally, the model should be study by a mathematician in terms of time step accuracy, by building this model and the equation describing it, the time step accuracy limit could be deduce with special mathematic tool. But this would include the study of ODE and is probably extremely complex.

4.3.1. Reducing the parameter space

The idea is simple, from the salamander morphology, we can deduce that muscle from the trunk are more powerful than the one from the tail, also they are in average attached faraway from the body axis (corresponding to the *r* parameter). And this repartition follow two profile, the mass distribution for the muscle force (F_{max} parameter) and the width profile for the *r* parameter. If we are able to determine those profile, only one value for all the F_{max} and *r* value will be needed. The result is a decrease of the number of parameter needed, but as the robot morphology is not the same as in real animal perhaps those profile are not relevant. It is also why the shaped robot have been created.

Note that the profile will not be fixed, only the relative value from one parameter to the others will be fixed but the maximal value can be change and the others will change in proportion. That is why the profile will be expressed as value between [0 1].

Width profile

The width profile have been previously extracted from one X-Ray walking frame in section 4.2.2. From this profile we can extract the relative position of the robot joint on the body. On figure 4.9 the red

dot correspond to the body length position of the robot joint and give us information about the body width at those locations (table 4.3).

Joint	1	2	3	4	5	6	7	8
Profile Value	0.800	0.962	0.995	0.790	0.487	0.429	0.384	0.332

Table 4.3.: Width profile value

This profile will be use for the r parameter.

Weight profile

The weight profile is based on a chopped salamander. The murder took placed in Bordeaux, in the Jean-Marie Cabelguen lab.

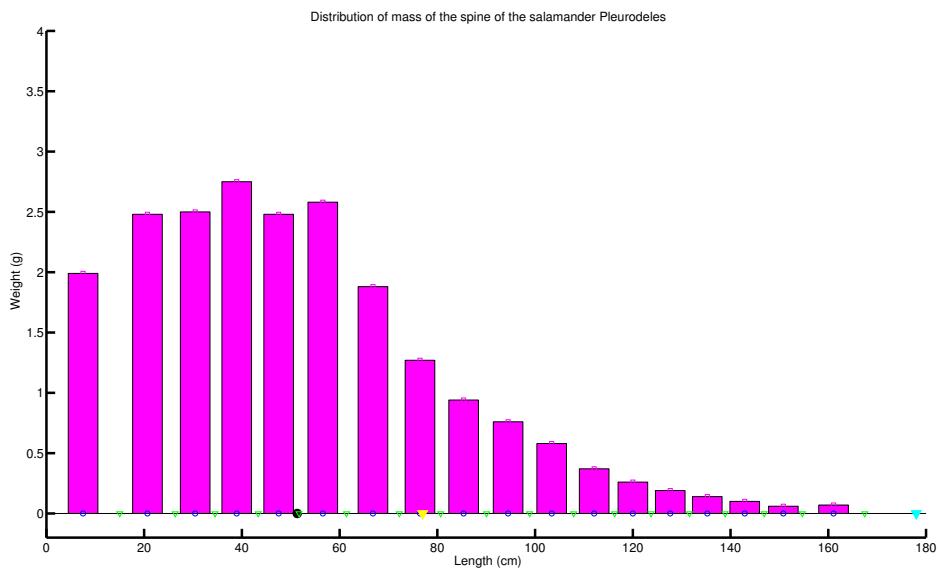


Figure 4.12.: Real salamander mass distribution

Based on this information, and following the same principle as for the with profile, the [0 1] profile can be computed.

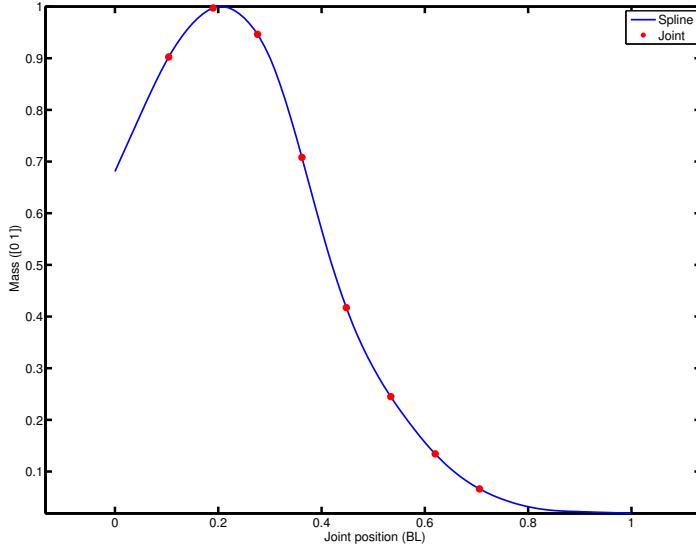


Figure 4.13.: Weight profile

The value corresponding to the joint position are display in table 4.4

Joint	1	2	3	4	5	6	7	8
Profile Value	0.902	0.997	0.946	0.708	0.417	0.245	0.134	0.066

Table 4.4.: Weight profile value

This profile will be use for the F_{max} parameter.

4.4. Protocol

Based on the first attempts and the experience I had form them I finally define a set of final optimization to run.

Common setting are :

- population = 200
- iteration = 200
- time step = 0.1 ms
- simulation time = 10 sec
- fitness function = $\frac{1}{error}$ Weighted according to joint angular amplitude, also error estimation start only after 5 second of simulation when the swimming gait is stable.

The available Webots model are :

- Normal robot without tail
- Normal robot with tail
- Shaped robot without tail
- Shaped robot with tail

It enable us to explore the effect of the tail, and of the shape of the robot.

The parameters spaces chosen :

- Full : all the parameter are open (F_{max} , l_{opt} and r for each segment)
 - $F_{max}[1:8]$ range from 0 to 200 N
 - $l_{opt}[1:8]$ range from $0.095 * 0.724$ to $0.095 * 1.615$ m (middle of the force length bell shape)
 - $r[1:8]$ from 0.001 to $0.0633 * 110.5 * 2$ m (using the max(radius)/BL find for the width profile and multiply by the BL of the robot 110.5 cm times 2 because r is the diameter not the radius)
- Full restrain : all the parameter are open but with restrained value range
 - $F_{max}[1:8]$ range from 0 to 200 N
 - $l_{opt}[1:8]$ range from $0.095 * 0.874$ to $0.095 * 1.526$ m (third of the force length bell shape)
 - $r[1:8]$ from 0.001 to $0.0633 * 110.5 * 2$ m

The use of restrain parameter space is motivated because the extreme value of l_{opt} are often found by the optimization process and are a bit extreme (see section 4.5).

- Profile : l_{opt} remain open for all joint but F_{max} and r are constrain according to their respective profile. Only two parameter are needed, $F_{maxprofile}$ and $r_{profile}$.
 - $F_{maxprofile}$ range from 0 to 200 N
 - $l_{opt}[1:8]$ range from $0.095 * 0.724$ to $0.095 * 1.615$ (middle of the force length bell shape)
 - $r_{profile}$ from 0.001 to $0.0633 * 110.5 * 2$ m

The F_{max} and r value for each segment is the value of $F_{maxprofile}$ and $r_{profile}$ respectively multiply by its corresponding profile value.

Note : A very important parameter is not explain in this section. This parameter is named Timelag and is key for our experiment but is difficult to explain. In few words, the question is how can I be sure of my error measurement is there is phase shift between my expected angle and the current angle. What is important is not the absolute error but the shape error. Or this shape error is very difficult to measure in real time if a phase lag is present between the two signal. To solve this problem, I allow the EMG to start at any moment of the cycle and I fixed the desired angle evolution. This way the optimization have to adapt the starting time of the EMG, to be in phase with the desired angle evolution. This is the purpose of the Timelag parameter which logically range form 0 to $\frac{1}{swimmingfrequency}$. it is important to notice that by this way we are absolutely not forcing the muscle activation to body dynamics timing. This timing is due to the others parameters and the system dynamics, we are just shifting the data to one direction or one other, the relative timing remain the same.

4.5. Results

The following optimization have been performed :

- 5 × Normal robot without tail and Full setup
- 5 × Normal robot with tail and Full setup

- 5 × Shaped robot without tail and Full setup
- 5 × Shaped robot without tail and Full setup
- 1 × Normal robot without tail and Profile setup
- 1 × Normal robot with tail and Profile setup
- 1 × Shaped robot without tail and Profile setup
- 1 × Shaped robot without tail and Profile setup
- 1 × Normal robot without tail and Full restrain setup
- 1 × Normal robot with tail and Full restrain setup
- 1 × Normal robot without tail and Full setup at 0.5 Hz
- 1 × Normal robot with tail and Full setup at 0.5 Hz

Much more optimization should be performed in order to get relevant statistical information but time have been my limitation. This chapter will explain the important information that can be extracted from those optimization. To this end, a complete document resuming the performed optimization should be enclosed with the report, its tittle is *Appendix : All Optimization*. This document have been auto-generated the notation inside are as follow : Opti_9seg_tail_w62_shaped_Full_1Hz_1. This experiment use the Full parameter space, on the shaped robot with tail. It run at 1Hz and is the number 1 performed. It confirm the use of the new w value for the muscle model.

At first, I will explain the different plot that will be found in the document by studying the Opti_9seg_tail_w62_Full_1Hz_2 experiment. Afterwards, I will give some general observation and link them to the appendix document for the sake of report size. At this step the reader will have to deal with the CD and the appendix documentation.

At first a link to the resulting swimming gait as well as the best optimization set of parameters are display.

Webots video : CD/Opti/Opti_9seg_tail_w62_Full_1Hz_2/Opti_9seg_tail_w62_Full_1Hz_2.mpeg

	Joint 1	Joint 2	Joint 3	Joint 4	Joint 5	Joint 6	Joint 7	Joint 8
Fmax (N)	182.367	4.776	193.689	187.303	68.481	113.315	117.791	156.856
Lopt (m)	0.070	0.071	0.071	0.075	0.096	0.095	0.095	0.105
r (m)	0.131	0.118	0.093	0.086	0.115	0.137	0.096	0.122
TimeLag (s)	0.714	0.714	0.714	0.714	0.714	0.714	0.714	0.714
error	0.127	0.127	0.127	0.127	0.127	0.127	0.127	0.127
Fitness	7.876	7.876	7.876	7.876	7.876	7.876	7.876	7.876

Table 4.5.: Optimised Parameters

Followed by a plot of the desired angle versus the obtained angle, this graph show in more visual terms the performance of the resulting set of parameters.

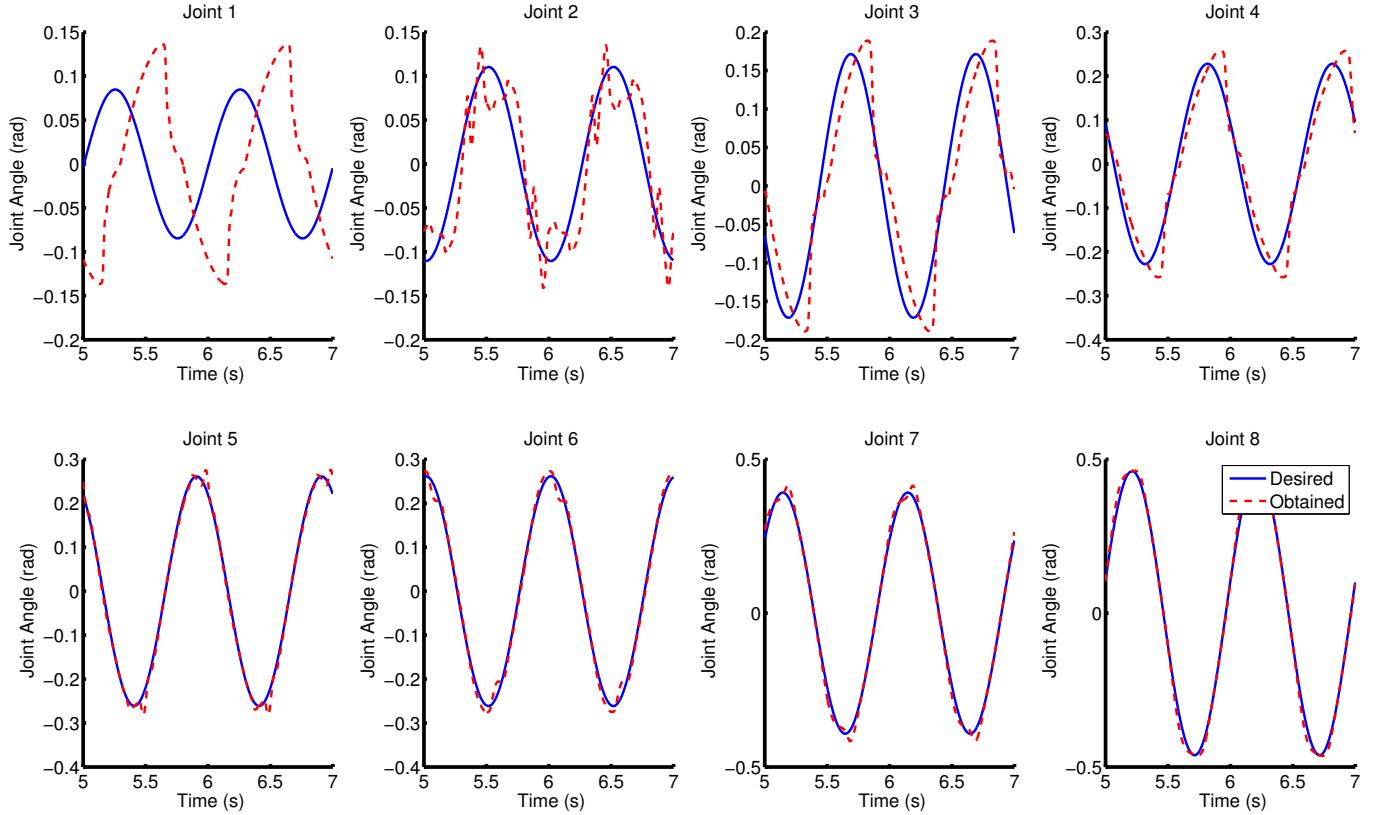


Figure 4.14.: Expected and obtained joint angle after optimization

This particular example show that the first 2 joint are not at all well fitted. The third and fourth one show a strange behavior close to the 0 angle. This will be explain by a future graph. The 4 last joint behave as expected.

The three next graph represent the swimming gait properties as extracted in the chapter 2. Angle amplitude, angle phase lag and lateral displacement are displayed and compare to the statistical data resulting form the X-Ray analysis.

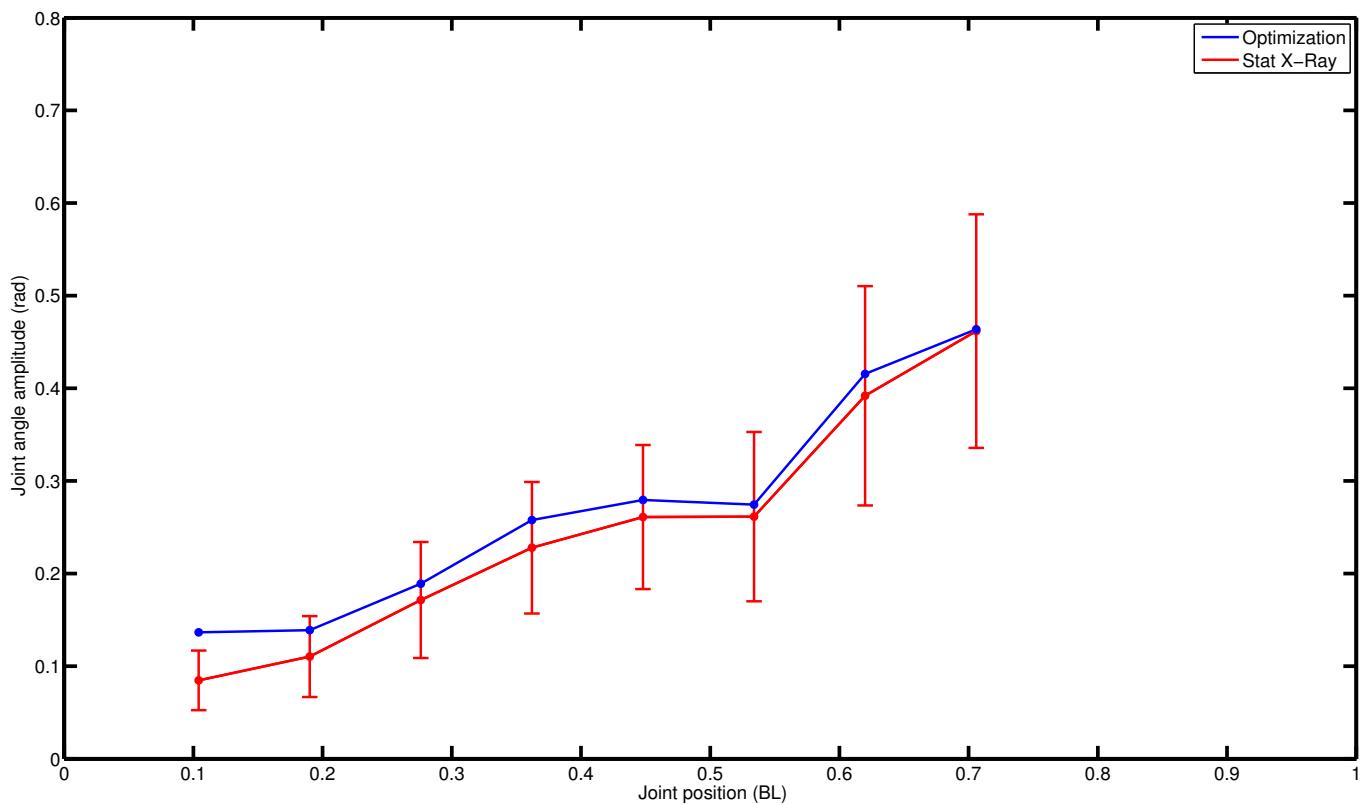


Figure 4.15.: Resulting angle amplitude compare to statistical data

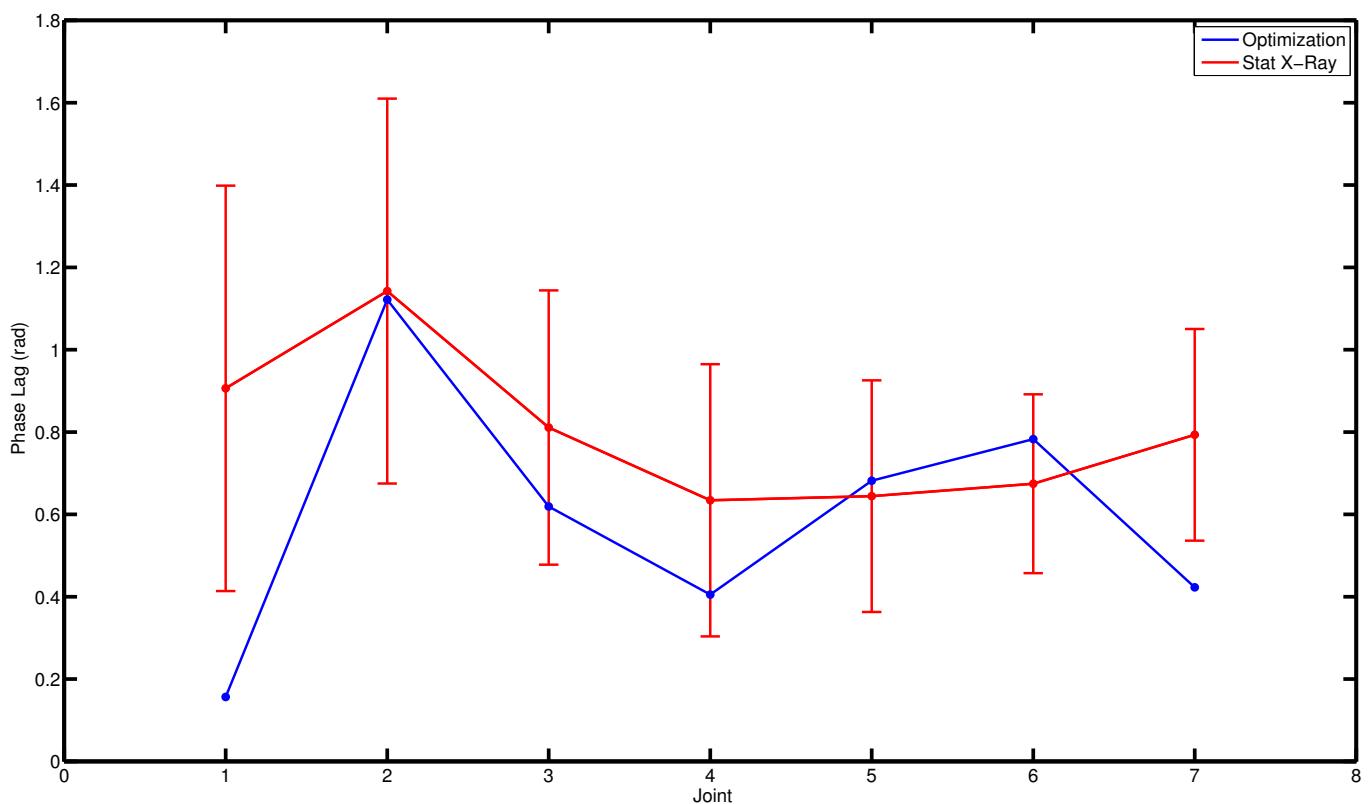


Figure 4.16.: Resulting angle phase lag compare to statistical data

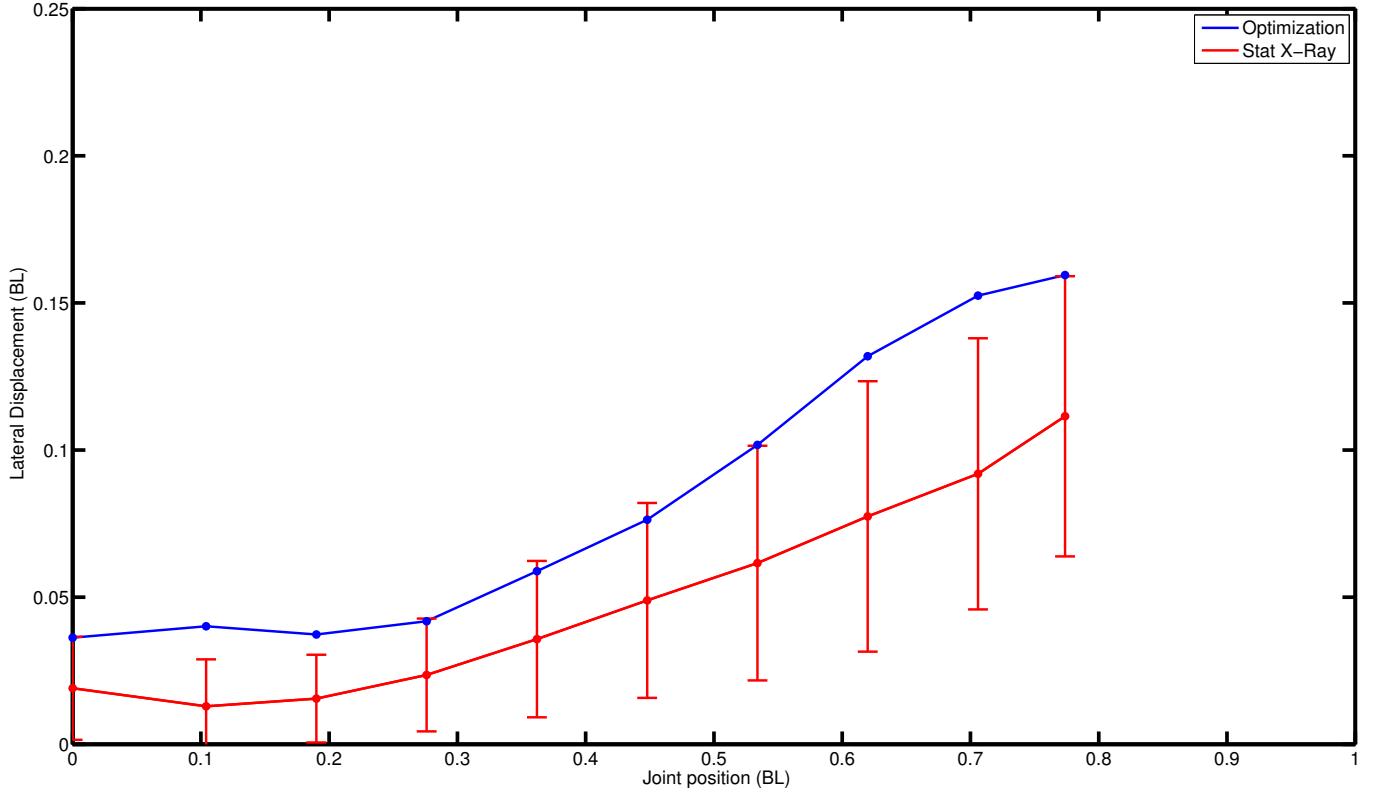


Figure 4.17.: Resulting lateral displacement compare to statistical data

The angle phase lag is not cleverly displayed. The phase lag is expressed in radian but should be given in cycle percentage. The x axis is labeled as joint number, but the number 1 represent the phase lag between the first and the second joint. That is why there is only 7 data. Ideally this axis should represent the body position between those two segment for example. One other representation is the one use by Delvolv  in [22], where the phase is represented depending on one reference on the body.

The displayed figure show that the behavior of this optimization is very similar to a realistic statistical swimming gait. Note that the lateral displacement have the same shape but a bigger amplitude. First of all this parameter is not at all optimized by the PSO algorithm. It is the pure result of the body to water interaction to the current body dynamics. Perhaps this difference is due to a the model which can not take in to account every world parameters. But it can also be due to the robotic shape, size and weight which differ widely from a real salamander size.

The kinematics part ended by tables summarizing those characteristic in numbers.

	Lateral Displacement (BL)	Angle Amplitude (rad)	Phase Lag (rad)
Joint 0	0.036	NaN	NaN
Joint 1	0.040	0.137	NaN
Joint 2	0.037	0.139	0.156
Joint 3	0.042	0.189	1.122
Joint 4	0.059	0.258	0.619
Joint 5	0.076	0.279	0.405
Joint 6	0.102	0.274	0.682
Joint 7	0.132	0.416	0.783
Joint 8	0.152	0.464	0.423
Joint 9	0.159	NaN	NaN

Table 4.6.: Swimming data

Robot Speed (BL/s)	Backward-travelling Wave Speed (BL/s)	Ratio
0.587	1.010	1.721

Table 4.7.: Speed data

This last table show the swimming efficiency. Unfortunately a dummy mistake make the ratio been the inverse as the usually displayed one. The real swimming efficiency is here $\frac{1}{1.721} = 0.58$. Which is quite low comparing to classical salamanders value. In [?] the estimated swimming efficiency is 0.75 ± 0.04 (mean + s.d.). Once again this can be due to the modeling or to the robotic shape, size and weight or to a lower efficiency of the resulting parameters or to the 1 Hz swimming frequency.

After the kinematics analysis, focus is bring to the relative timing of EMG and kinematics, to the muscle working zone and to the estimated energy consumption.

It began by a short Matlab video representing the optimization result with the muscle and the activation wave.

Matlab video : CD/Opti/Opti_9seg_tail_w62_Full_1Hz_2/BodyMuscleActivity_Opti_9seg_tail_w62_Full_1Hz_2.mp4

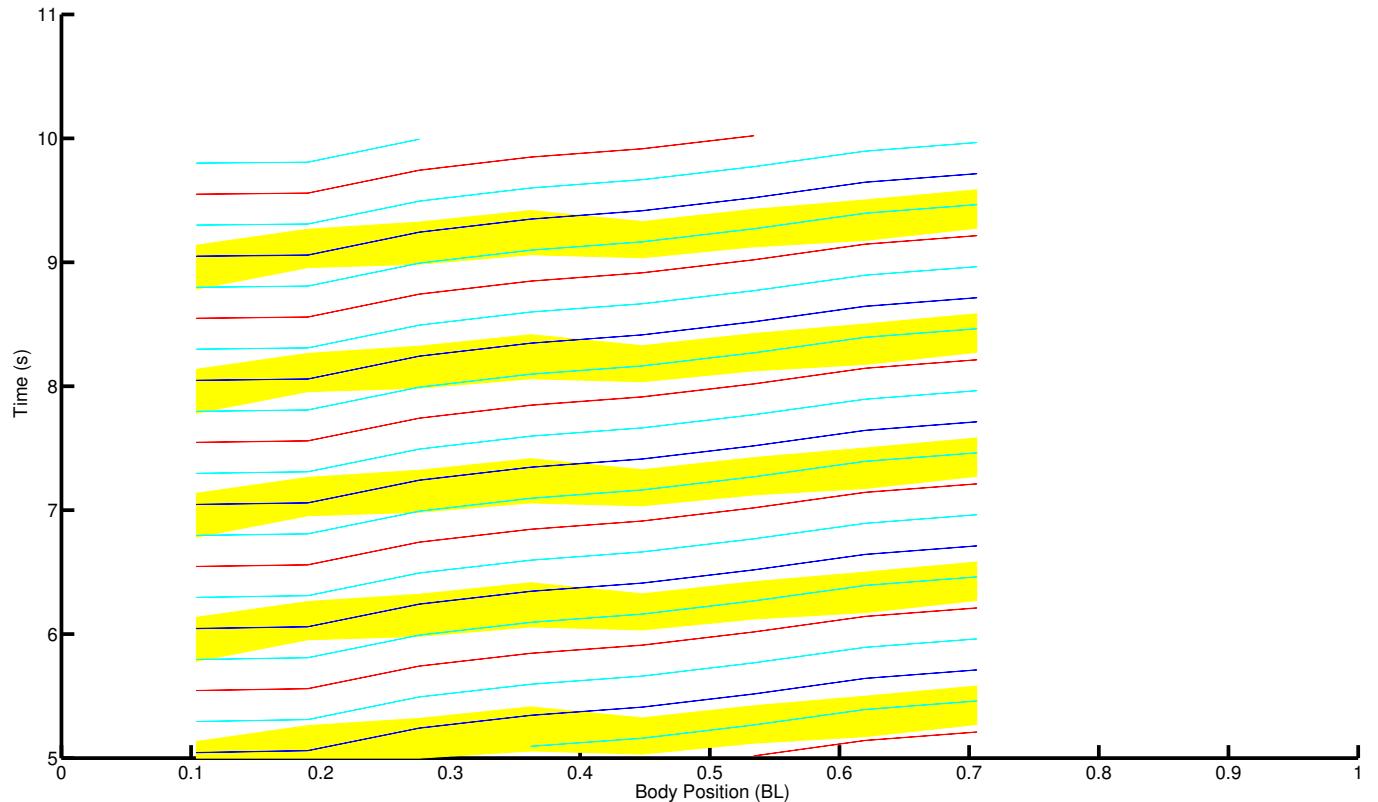


Figure 4.18.: Muscle stimulation and body dynamics

This graph use the same representation as in [19]. In short, this graph only consider one side of the body but the swimming is in a straight direction which led to symmetrical result. First the axis, as x axis, is displayed the body position in (BL) and as y axis is display the time. The red line on the graph represent the time when a joint is at his maximal angle and the corresponding muscle is fully extended. The blue line correspond to the opposite position with muscle totally compress. The yellow area represent the activation time of the corresponding muscle. To link this graph with the D'Aout paper [19], the red zone on the following figure 4.6 represent the yellow zone of figure 4.18. The following plot is not accurate is an help to the visualization.

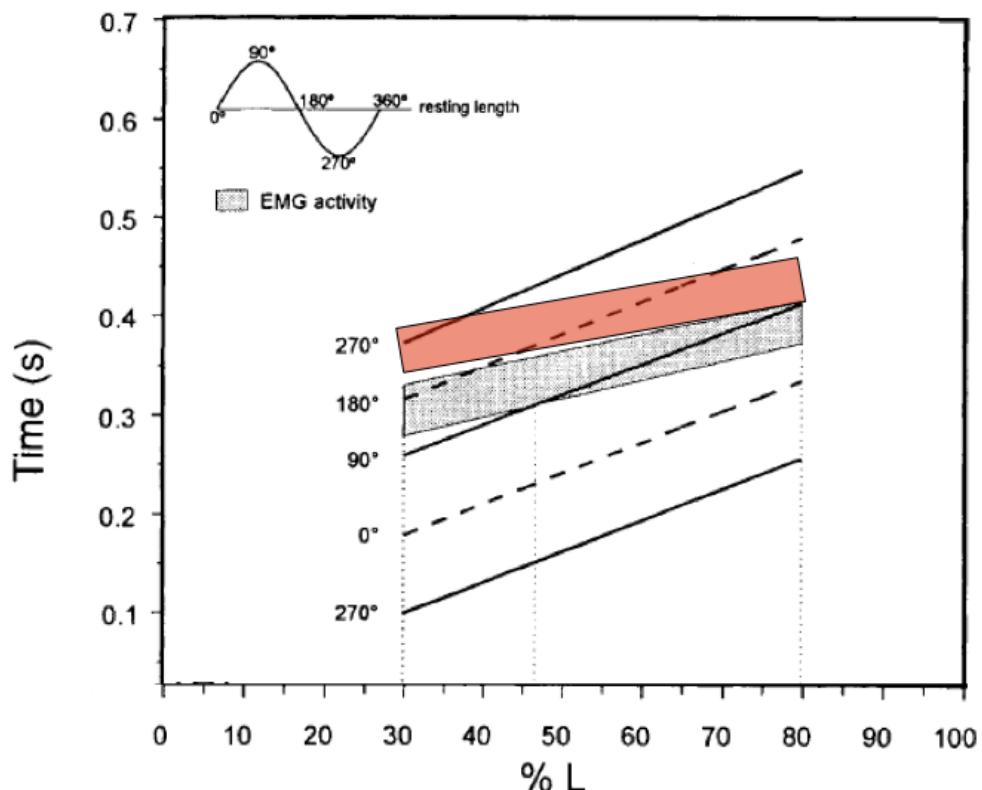


Figure 4.19.: Timing comparison

This particular plot will not be display in the appendix. In our case it show that the optimization result are not able to reproduce a realistic muscle activation to body curvature dynamic. We will see later what are the proposed solution. Next graph 4.20 show the working of the muscle during the swimming.

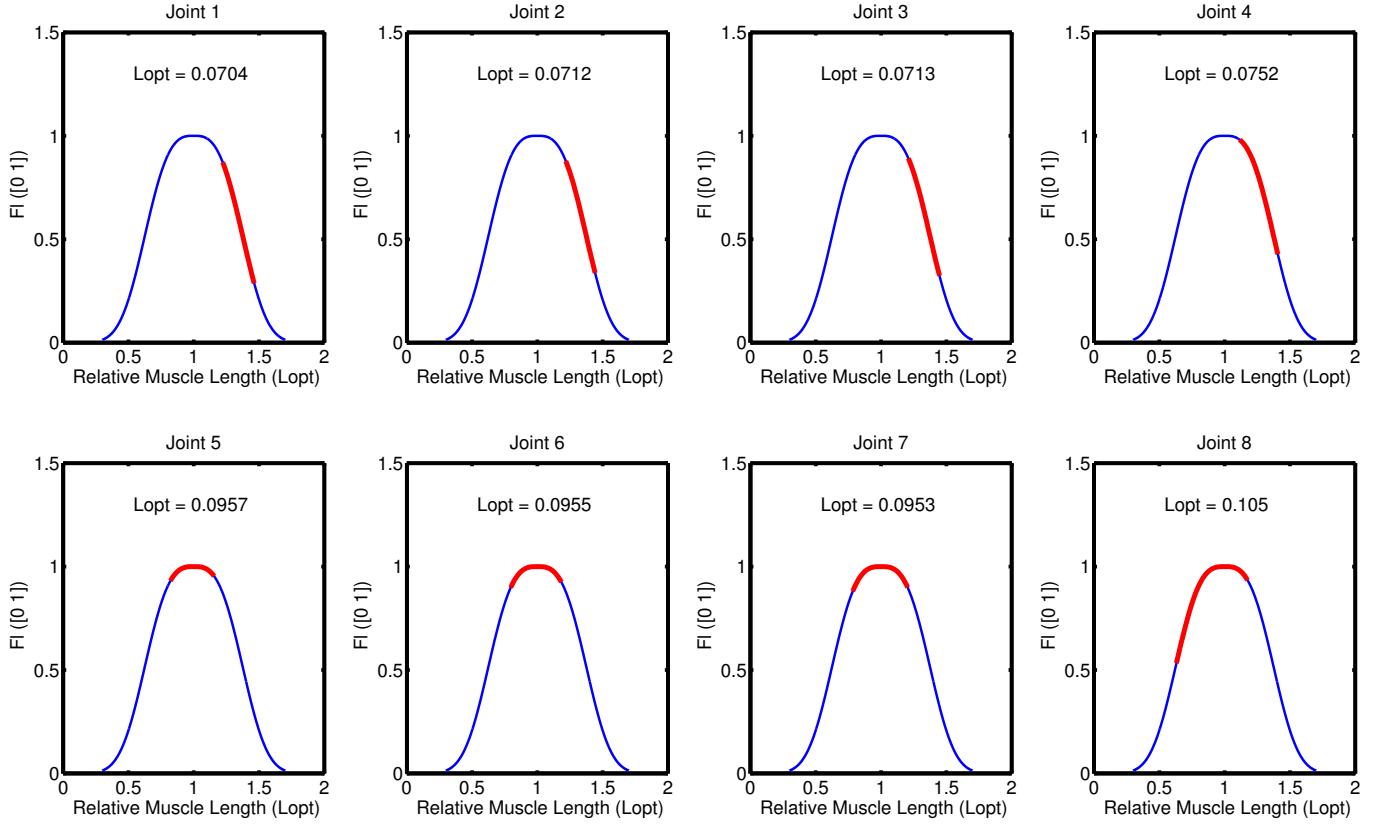


Figure 4.20.: Muscle working zone

This graph explain the strange angle evolution for the four first joint (figure 4.14). Those four first joint are using muscle in their elastic domain. At rest, a non negligible force is applied by muscle of both side of the joint which led to a totally different behavior. First those segment are very rigid. as soon as you contract one muscle it need to work against the antagonist muscle passive force which is not an easy task in this condition. And as soon as the muscle stop been activated, a strong force bring the segment back in a straight position. That explain the strange behavior close to 0 angle of the 3 and 4 joint figure 4.14 and also the shape of the angle evolution.

Finally the appendix document display the metabolic energy consumption of each joint (two muscle). We are not using this information but it could be interesting for some people. Moreover for some experiment we can notice that energy consumption is negative as explained before.

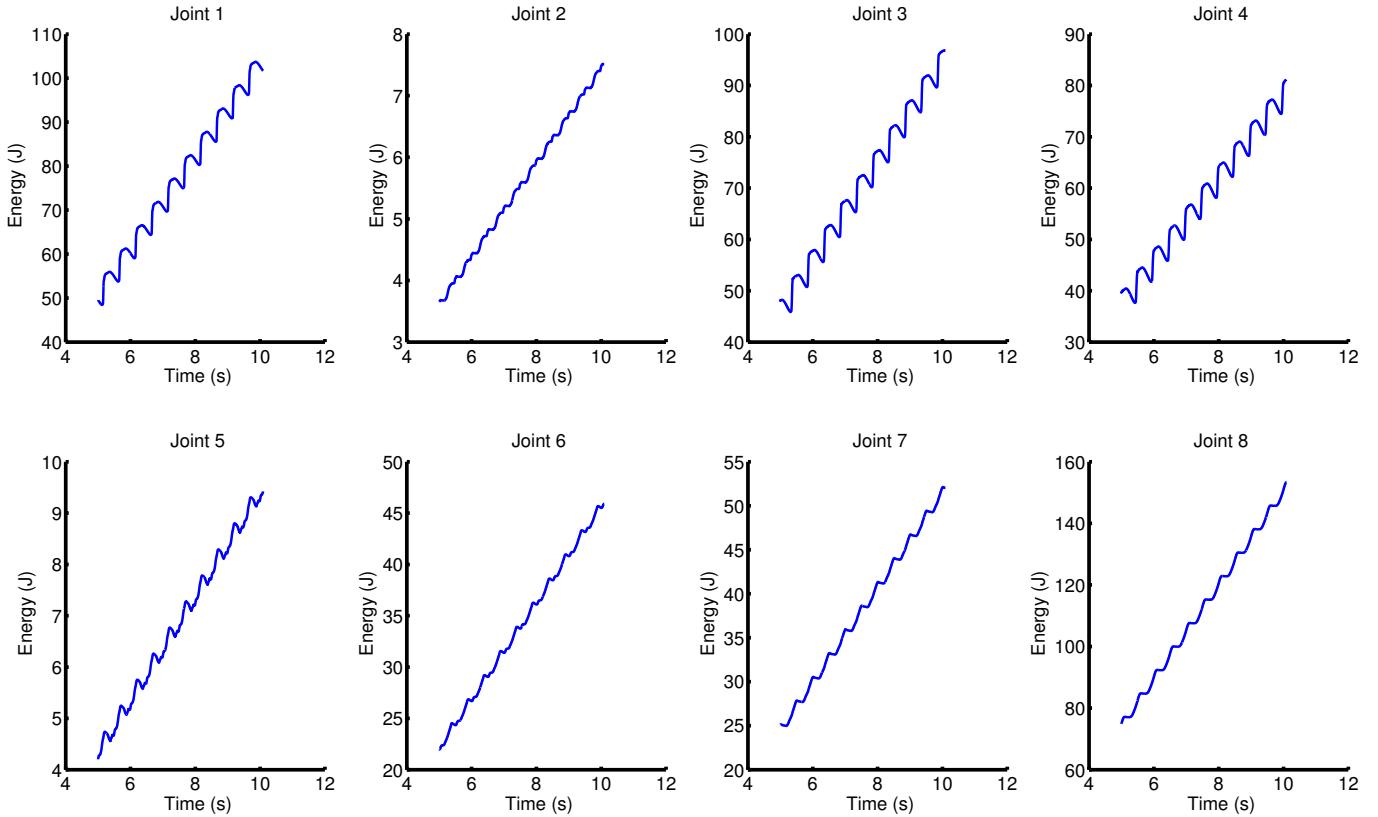


Figure 4.21.: Energy consumption

Those explanation should be enough to understand all the appendix and CD content. The next lines will describe a personal analysis of some result with few reference to the appendix. It is important to remember that not enough experiment have been run to use the resulting data as general conclusion. My observation are as follow :

- Tail effect : the tail have mainly two effect, (1) it stabilize the swimming gate and help the optimization algorithm to find good solution. This effect is mainly visible on the last segment of the body. (2) As expected it increase both the robot absolute speed and the swimming efficiency.
- The shaped robot seems to not have particular influence on the optimization performance. But have one important consequence on the lateral displacement characteristic which fit perfectly in the statistical value. In a way this is a good news for the water to body interaction modeling and the Webots model in general. However this effect is more probably due to the new weight of the robot that to the new shape. But this remain to be proven.
- In almost all experiment the muscle to body dynamics is different than the realistic one. Which certainly means something is missing in our model. Next section will propose a new model which could improve this point.
- In almost all experiment, the four front joint muscle are used in a very extreme working zone. This can be explain by a model limitation. But also by the EMG reconstruction limits. The way I reconstruct the EMG assume a muscle is activated only once a cycle or, the Delvolv e paper and others show that in the front part of the body co-contraction happen. It mean that when one muscle is activated the antagonist one is also activated, at a lower level but activated, that could explain the need for stiffness in the front part. Our model, more precisely our EMG input data, do not account for this special pattern and the optimization algorithm found a way to bypass it by increasing the spring effect of the muscle.

- The `Opti_9seg_tail_w62_shaped_Profile_1Hz_1` is interesting it show a better muscle to body timing dynamics. This pattern is very close to the one presented in papers. We can note that the 3 and 4 joint muscle working zone are back to a "normal" operating zone. But the angular evolution of the 2 first joint is still very chaotic. Interestingly this timing pattern is not present for the normal robot in `Opti_9seg_tail_w62_Profile_1Hz_1`. Thus it look like the shape associated to the Profile setup have interesting properties, but one experiment of both is not enough to make conclusion and I am not yet able to explain this difference. But few more optimization should be performed with those setup.

Note : The CD and appendix document are important source of information and interested reader should build their own opinion by exploring them.

4.6. Improvements

In the previous section, we have seen that many of the results show strange muscle working zone for the first four segment and a delay muscle to body timing dynamics. I propose two explication :

- The muscle working zone could be explained by the lack of co-contraction but also by the facts that with the current muscle/joint model, the all spring effect is due to the muscle spring constant. Or in animals some of this work is performed by tissues and ligaments. By consequence adding a independent spring in parallel to the muscle should solve a part of the problem but will add one parameter to the system. Unless biologist could tell us what is the ratio between spring effect from muscle and for tissues.
- The delayed timing properties could be explained by the lack of tendons. To explain this facts, I will use a simple experimental result. When I apply directly a force to an object, the object will move, but if to apply this force I use a spring, a delay will occur between the time of my action and the movement of the object. This is a negative delay, that mean if I want to move an object at a special time I will have to apply the force before. And that is exactly what is needed to bring back the red zone down to the realistic zone (cf). this hand explication is the effect tendons have and is the way they are modeled, as a first order differential equation. This change could solve the problem of timing but will this time increase importantly the computational cost of the model because muscle and tendons are linked together and they affects each others.

Moreover to anticipate with next chapter, the use of a damping for out-water experiment is needed. My first thought was to overlooked the animal intrinsic damping due to the tissue. This assumption is probably correct while the robot stay in water all the time. But when we want to test the rigidity of the body outside of the pool, the lack of damping lead to an instability in the system. Without damping one ideal model tend to maintain oscillation. The final proposed model is display below.

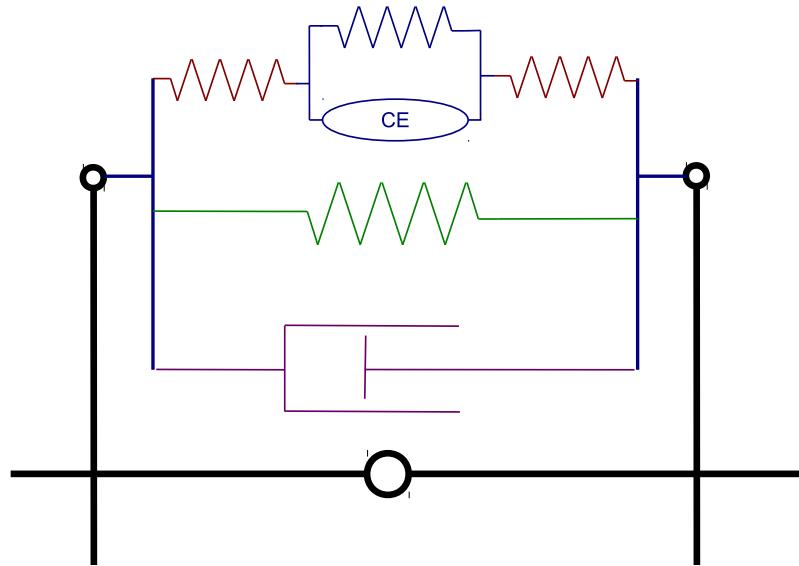


Figure 4.22.: Proposed muscle/joint model

4.7. Discussion

Discussion about the optimization meaning and modeling are part of the future work and mainly of the conclusion. To conclude this part I will say that not enough experiment have been perform due to a lack of time. This do not allow us to generalize from the few example and all my explanation should be put into more test. Moreover, a living creature is the fruit of long term optimization that take into account much more parameter that we could imagine. Integrating them into our optimization is impossible, but a better fitness function will help.

5. Robot

Optimization result allow us to dream of reproducing the result on the real robot in the lab. This chapter describe quickly the robot architecture, the problems that have been solved and the encouraging result.

5.1. Robot architecture

A new robot is currently in use in the BIOROB laboratory. No publication have yet been publish on his infrastructure, and it is not the purpose of this section. Reader can refer to [17] which describe the version 2 of the modular robot.

In chapter 1.4, a general description of the robot can be found. The robot is a modular robot, the head module embed the main computational power with an ARM and communicate to others modules via a bus CAN. Module are linked to the others via a rotating connector and are slave on the network. As always in real time system, the procedure is a loop of fixed or variable time step. In our case, this loop is composed of three steps :

- The head ask to each segment, its position, speed and torque value
- Based on previous and current state of the system, the muscle/joint model is computed and return the new torque to apply to each rotating joint
- The head send to each segment the new torque to apply

In a module the PIC which receive the message is not the one who act on the motors. Therefore those two PICs communicate via a I2C channel, which is much more slower than the CAN bus. Moreover, the head can only communicate to one module at a time. The protocol implemented for the CAN bus include an acknowledge system.

5.2. Problems

The main problem with the robotic platform is that we have to deal with real time. We cannot as in Webots use a time step of 0.1 ms. Hopefully the load of computing is far lower than on the computer, simply because environment does not have to be computed.

The source of problem could be :

- Differential equations : for too big time step, discrete resolution of differential equation tend to oscillate. Or the stimulation to activation relation is a differential equation (figure 3.21).
- Torque and open loop control : torque control is by definition unstable. If for any reason wrong value are read, the time step is too big, the torque will remain applied for a too long time and may damage the robot in addition to a very bad swimming gait.

The time step is the critical parameter we have to control. When I first implement the muscle/joint model on top of the code I receive, the time step was 40 ms and the robot was oscillating dangerously¹ for the hardware. With this initial time step, nothing could have been done with the robot. The

¹CD/Robot_test/Crazy_oscillation.m4v

transformation from stimulation to activation, which use a differential equation is unstable for a time step bigger than about 22ms of time step (figure 5.1).

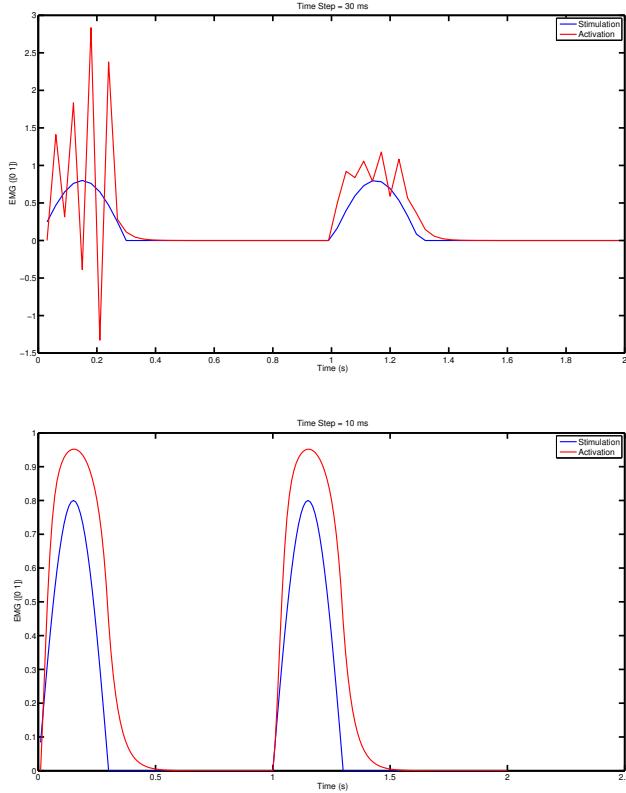


Figure 5.1.: Robot time step instability at 30 ms and stability at 10 ms

5.3. Solutions

In front of this problem, we need by any means to decrease the time step.

5.3.1. Lookup table

Function from the math library are sometime too costly for the use and the precision we need. Thus function like *sin()*, *tan()* and *exp()* have been implemented as lookup tables. Those lookup use a lot of memory but there is no problem of memory with the ARM in the head. That improve the time step of 1 or 2 ms only.

5.3.2. Code improvement

The first task is to replace the complex CPG code by a very simple EMG simulator. EMG are fixed in our stimulation and do not have to treat any feedback information. But the implemented CPG is a quite complex algorithm, and the only fact of suppressing it bring the time step to 30 ms. Then I decided to suppress some feed back information such as the torque, but also the rotating speed which will be calculated from previous and current angular position. This shortcut make us win a lot of time because of the I2C communication inside each module which take times. Therefore the only information asked to the module is the angular position, this bring the time step from 30 to 15 ms. Even at this quite low time step, result on the robot are not acceptable.

5.3.3. No acknowledge

The acknowledge include in the protocol is secure but is a time waster. We need to run faster so I decided to delete it. In facts the operation which take time is once again not the fact to send an acknowledge but the waiting time due to the I2C communication inside each module. For example, when the head want to set a torque to the module number 2, it should send a message containing the torque value. The module receive it, decrypt that it is a torque command, and send via the I2C channel the order to the second PIC which execute the command and feedback an acknowledge to the first PIC which send it back again to the head. And it is only after this process that the head can send an order to the next module. The acknowledge is better for stability and liability of the system but it takes time. We can compare that as switching from TCP to UDP in networks.

By stopping asking for acknowledge, time step switch form 15 to 11.5 ms. This modification does not implies a any modification to the existing protocol. The acknowledge was only an option. Note that it only make sense for when the head send command to module, not when they want to receive information. This time step improvement is no yet enough to obtain a good swimming gait ².

5.3.4. A broadcast protocol

It look now obvious that the head is losing a lot of time waiting for an information coming from one segment. Remember that the I2C is quite slow compared to the CAN bus. Moreover it have to wait for the reply to switch to the next module. The solution look simple, if the head could ask at the same time the angular position of all the modules and then wait for their reply, the time will decrease dramatically. Instead of waiting 8 times the slow I2C communication it will only wait for one. Therefore, we need to include a broadcast address in the network. This way we send the request and as soon as one module receive the value from the second PIC it can send it back to the head with his ID number.

This was not provided in the initial firmware of the modules. Thus we had to implement it. Two things must been add to the firmware : (1) an broadcast address, (2) a field *sdn* in the reply message for the ID number of the responding module (it was not necessary before because the head was in total control of the bus activity). Alessandro implemented the change in the module firmware and I implemented the part for the head. The broadcast address is 0x1F.

This modification led to a gain of 7 ms in term of time step. Finally the time step have been reduce from 40 ms to about 5 to 7 ms.

Note : After switching to the firmware v6, few troubles appear when I start the robot after a stop, motors applied by themselves a positive torque and the robot stop responding. Could it be due to a wrong initialization in my code, with some access to zone of memory not yet initialize or wrongly reset. I do not think so, I check my code many time and do not find any

Note : the CAN chips include all the low level collision detection and address filter.

5.3.5. Residual problems

It look like by trying to go too fast we somehow overflow the sender buffer. Thus between two torque setting message I had to add a pause of few μ s.

Instability could also come from the non accurate angle measurement. The measure by it self is correct but the joint between two segment is not perfectly rigid and deformation occur during swimming. A solution is to deduce the real angles via the tracking system already present in the lab. Asking for this information should be as fast as the internal process which remain to be tested.

²CD/Robot_test/Best_old_firmware.m4v

5.4. Results

The parameters used for all the following test are not the one obtained by the optimization process. The robot remain too unstable to use such extreme value. I use very simple value, $L_{opt} = 0.090$ (m), $F_{max} = 30$ (N) and $r = 0.095$ (m) for all the joint. Also the phase lag have been set with a constant phase lag of about 54 degree in order to have one sinus on the body assuming the relation between muscle stimulation and body bending will be direct. This choice is explained by experience and test, the relative timing is impossible to reproduce in the current state of the robot, this timing ask to run at 1 Hz and is too dynamic for the limited stability of the robot. Also the robot start being unstable at 0.6 Hz, sometimes it can go faster but the stability remain fragile.

The first important result is to touch the robot to feel the effect of the muscle model passive properties. I cannot transcript how the robot behave but you can imagine it on the CD's video (`CD/Robot_test/Salamander_passive_vertical.m4v` and `Lamprey_passive.m4v`). The passive characteristics of the muscle maintain a cohesion in the body and if we bend it, muscles passive elasticity tend to bring back the body in the straight position. This beam effect is very important to the swimming dynamics. Moreover it give a living aspect to the robot which is always great for demonstration.

Using this configuration, the robot can swim at 0.5 Hz (`CD/Robot_test/05Hz_middle_swim.m4v`) and even turn (`CD/Robot_test/Salamander_lowfreq_turn.m4v`).

You can enjoy the others video :

- Passive two segment salamander : `CD/Robot_test/two_segment_robot_muscle.m4v`
- Passive head with small instability : `CD/Robot_test/head_robot_muscle.m4v`
- Fish robot table demonstration : `CD/Robot_test/Fish_demo_turn.m4v`
- Fish robot with muscle model in the pool : `CD/Robot_test/Fish_Swim.m4v`

The best experiment was unfortunately not recorded, I success to run a 5 segments lamprey to 1.5 Hz, the time step was about 2-3 ms.

5.5. Discussion

For my point of view of electrical and computational engineering student I already find this robot amazing. Well designed, it learn me many tricks that I will remember but it is extremely difficult to have all the pieces of the puzzle working together. With this new version of the muscle model and the torque control, we are reaching the limits of the robotic platform. Even using all this modification and winning so much time step, I feel the robot still not stable. There is many possibility that data get lost or falsify and adding secure protocol slow down the loop.

A new version of the robot is in development, it will add a lot of computational power in every module, which will enable each segment to compute its own muscle/joint model. The time step could go down to 1 or 2 ms, which will be amazing and allow the head to compute more high level task such as avoiding obstacle, following a target, etc.

Finally, from my point of view, a special focus should be bring to the torque control mode. Controlling a motor in torque is not so easy, especially when torque is changing often over time. Motor reaction is quite complex because of the coil inside. Perhaps a detailed study or reading should be done about the torque control of motors. Moreover the motors torque controller should be faster than the global time step.

6. Future works

6.1. Webots

6.1.1. 30 segments + tail model

Webots allow us to create a real segmented robot, which discard all the problem coming from the segmentation of the current robot.

This robot have been created (figure 6.1) but with a time step of 0.1 ms, this increase in segmentation led to huge increase in simulation time. It limited the optimization process which take two full week for this configuration.

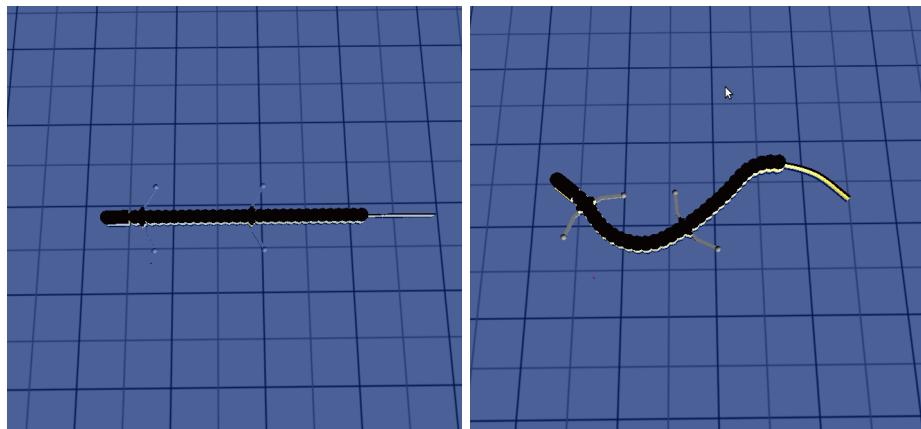


Figure 6.1.: The 30 segment robot model on Webots with tail

6.1.2. Real size, shape and weight model

The use of a real muscle/joint model will make more sense in a simulated real size, shape and weight Webots model. The ultimate simulation will be to weight and calibrate the bones of the animals. And develop a 3d model of it and optimize the muscle according to many factors, including energy and swimming efficiency. First of all the weight of the muscle according to their forces will be considered in the dynamics. Also the volume of the muscle will sculpt the shaped of the animal. The expected result are a global weight and shape as close as possible to a real salamander. But this is a full Phd thesis.

6.2. Optimisation

6.2.1. Energy optimization

Optimization according to energy consumption is not so easy if we consider the complex model form [8]. Others models and information can be found in [41, 1, 49, 34, 33, 20, 55]. Depending on how the optimization and the fitness function is set, the optimizer will find very unlikely and unrealistic gate which most often led to negative energy consumption ! And negative rate of energy consumption is the main weakness of the model. It is not a weakness for biologist because they always study muscle in real condition but optimization can go anywhere in the open parameter space and can exploit strange

model properties. For example, optimization results led to very extreme muscle working zone and muscle maximal force.

The solution is to restrain the parameter space to realistic values but what are realistic values? I know nothing about muscle working zone in real salamander, even in human. The insight of a biologist on this topic could be very helpful to set limitation to the parameter range and tell us what is feasible and what is not.

One other option is to use an different energy optimization function such as the work performed by the muscle. Which question the use of a real muscle model.

Note : In a certain extend energy optimization make sense but few studies show that a swimming gait is not always energy optimal.

6.2.2. Muscles have weight

Including the weight of muscle in the optimization could be an additional factor with the energy optimization. The problem remain that the robotic platform weight much more than bones, we should reduce the module weight and include before each simulation the weight of each module depending on the optimization muscle parameters.

6.3. Model

6.3.1. Slow and fast muscle

As human we feel the difference between running and walking, the difference is that we are not using the same kind of muscle for both type of activity. There is different kind of muscle fiber, some are slow but resistant as red muscle fiber, some are faster but less resistant as white muscle fiber. And of course the salamander muscle are composed of those two kind of muscle fiber. In [47] can be found a lot of details about muscle composition and repartition in an ambystoma tigrinum salamander.

It is well known that white muscle fiber are mainly use for quick and violent activity, such as escaping a danger. In general the more you increase the speed the more you use white muscle fiber. Because red muscle fiber are slow and cannot by themselves be responsible for a frequency increase. Therefore in order to study and reproduce a range of velocity dynamics, those two king of muscle should be modeled. Also the spacial placement of those muscle is important. The white fiber are essentially concentrate far away form the body axis in order to have a strong action when needed. And depending on the swimming frequency the percentage of both muscle activation could evolve as in paper [8].

6.3.2. Tendon, spring and damping

The reason to add tendon, spring and damping are explained in section 4.7, figure 6.2 remind the full model. I will not repeat them but suggest some way to do it. First of adding all this new function will increase the number of optimization variable and will led to much more difficult and variable results.

The work of J. Long [43] suggest that axial tendons are essential to the simple beam-like behaviors of fish. To do so he develop a musculoskeletal model of fish and test it with and without tendon. Before switching to tendon modeling this paper is a must read even if it use linear muscle model. I am sure tendons have an important role only by the facts that it exists and remain the body but not so much data are know about salamander tendons. The main question to ask is : what it length ratio between tendon and muscle in a salamander. If we answer this question we discard one optimization variable.

Characterizing the spring and the damping is much more difficult. Even if we were able to measure it on a real animal, the way to extend it to the 9 segment robot is difficult to define properly. What could be done for the springs constant is to characterize the amount of spring due to tissue versus the

one due to muscle and tendons. This value will once again decrease the number of variable. Once again transposing this ratio to the robot size and segmentation is perhaps not viable.

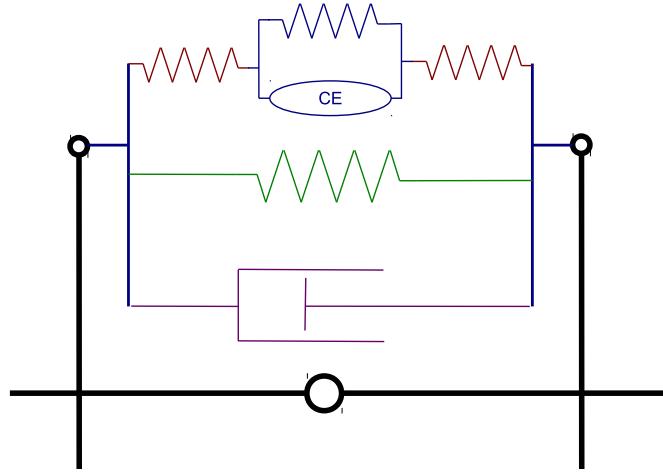


Figure 6.2.: Muscle/joint model with muscle, tendon, spring and damping

6.4. Control

6.4.1. EMG as coupled oscillators

Instead of the actual angular position CPG implemented and used to control the robot, a EMG central pattern generator should be developed. It will be good base for future improvement in control such as feedback information integration and perturbation correction.

6.4.2. EMG feedback

The overall goal of this project is to became a tool to study directly the neural networks and EMG data in relation with the effect they produce on the animal kinematics. A important goal is to study how feedback informations are integrated and treated inside the complex neural network system. How it stabilize and constrain the open loop fuzzy reaction and what are the effect on the animal kinematics.

6.5. Robot

6.5.1. Realistic shaped robot

By using foam, it would be interesting to reproduce the same shape and weight than the new realistic shaped Webots model. It will allow to compare the result and observe the influence of such a center of mass shift on the robotic swimming gate.

6.6. Robot enhancement

As explained in chapter 5, the robot should and wis planed to be improve. Adding computational power in each module is a priority. I insist that we should focus on the torque control. And finally if we wanted to win even more time, especially when feedback information will be implemented, a neighbor to neighbor communication could be an amazing time saver. This option is difficult to implement because of limited space, a solution could be adding wireless chips as Bleutooth or Xbee in every module, but the energy consumption increase is not negligible.

6.7. Discussion

As always, this project open more question than it reply to. But based on the current state of it many of these future work can be done in a matter of weeks.

Moreover we can note some Interesting contributions in the domain :

- non linear muscle model
- salamander (rather than lamprey)
- it works on a robotics platform
- EMG irregularities
- kinematics from X-rays movies
- optimization methodology to fit the muscle parameter.

But the study will only be complete when the relevant muscle activation to body dynamics timing will be reproduce by this indirect optimization way. Adding tendons will probably help us to reach this goal.

Current developing robot in the lab will have the ability to reproduce almost entirely the kinematics of a real salamander, even for the very complex walking gate. Will it be able to simulate all the muscle active in the salamander body, understanding the role of every muscles, every tendons, every tissues ? This will represent a huge work from an engineering point of view but also and probably more for mechanical engineers and biologist which will have to collect precise and exhaustive information about the properties of each single muscle tendons, bones and tissue of the salamander body. Its remain a dream for biologist, but roboticist do not care, because there is no need to simulate every part to reproduce the global behavior of a salamander. And if we reach this state, does the next step will be to connect a salamander brain to this new artificial body ? It is probably coming..

7. Conclusion

As a conclusion, I will not resume the report content but asking, and arguing around the following question : Does it make sense to use a non-linear muscle model for a 9 segmented robot ? This question should have been asked before the project, but it is only by working on it that very important weakness in the use of non-linear muscle model have been revealed. This question of the viability and the meaning of using a realistic muscle model for the robotic platform obsessed me during the all course of this project. I do not have clear answer, just a personal view and few arguments easily opposable.

The use of a hill model was mainly justified by two argument : (1) it enable us to estimate the energy consumption based on biologically relevant models and (2) it bring a realistic interaction between EMG command and kinematics. In a first approach, those arguments have no counter argument, but during the course of this project we have seen that the robotic platform, by its non realistic segmentation, constrain the model due to the limited stretchability of muscles.

The joint model have been change to solve the problem of muscle to ribs direction of force. In the final model the force is always applied parallel to the body axis and if this change is not applied we can feel that the muscle will need to be stronger and will not react the same way as in a real animal. Which discard both (1) and (2) argument.

The muscle model had also to be changed and applying no change to the muscle model would have tend to attach the muscle closer to the body axis on the ribs and would have result in a high augmentation of muscle force and a huge increase in energy consumption. Which discard both (1) and (2) argument. But changing the w parameter of the muscle model affect also the energy consumption modeling in a way we cannot predict. Therefore the energy information should be use only as a relative value. It could only be compare to an other value by a plus or minus comparison. Which could be done with any others models, linear models included.

Of course using a model does not mean trying to model a possible reality, because a model remain a model. While we are not trying to use this model to infer about what happen on the real animal I cannot see any others problems that I found this model too complex compare to a linear one. I believe that a muscle/joint modeling using a linear muscle plus tendon, spring and damping will perform as well as with a non-linear muscle model.

From my point of view, I would suggest two main direction. In one hand, the robot cannot be biologically relevant in terms of muscle/joint modeling but if proven that any model (linear or not) is able to reproduce the muscle to body dynamics, it should be used by keeping in mind that no deduction can be infer about the real animal muscle function, and that only because of the segmentation problem. In an other hand, the model developed in this report can be extended by adding tendon, spring and damping but used on a realistic size and segmentation salamander model. From which results will have some meaning.

Bibliography

- [1] BC Abbott, XM Aubert, and AV Hill. The absorption of work by a muscle stretched during a single twitch or a short tetanus. *Proceedings of the Royal Society of London. Series B-Biological Sciences*, 139(894):86, 1951.
- [2] J.D. Altringham and D.J. Ellerby. Fish swimming: patterns in muscle function. *Journal of Experimental Biology*, 202(23):3397, 1999.
- [3] J.D. Altringham and I.A.N.A. Johnston. Modelling muscle power output in a swimming fish. *Journal of experimental biology*, 148(1):395, 1990.
- [4] J.D. Altringham and I.A.N.A. Johnston. Scaling effects on muscle function: power output of isolated fish muscle fibres performing oscillatory work. *Journal of experimental biology*, 151(1):453, 1990.
- [5] E. Azizi, G.B. Gillis, and E.L. Brainerd. Morphology and mechanics of myosepta in a swimming salamander (siren lacertina)* 1. *Comparative Biochemistry and Physiology-Part A: Molecular & Integrative Physiology*, 133(4):967–978, 2002.
- [6] AF Bennett, T. Garland, and PL Else. Individual correlation of morphology, muscle mechanics, and locomotion in a salamander. *American Journal of Physiology-Regulatory, Integrative and Comparative Physiology*, 256(6):R1200, 1989.
- [7] W.O. Bennett, R.S. Simons, and E.L. Brainerd. Twisting and bending: the functional role of salamander lateral hypaxial musculature during locomotion. *Journal of Experimental Biology*, 204(11):1979, 2001.
- [8] L.J. Bhargava, M.G. Pandy, and F.C. Anderson. A phenomenological model for estimating metabolic energy consumption in muscle contraction. *Journal of biomechanics*, 37(1):81–88, 2004.
- [9] A.R. Blight. Undulatory swimming with and without waves of contraction. 1976.
- [10] A.R. BLIGHT. The muscular control of vertebrate swimming movements. *Biological Reviews*, 52(2):181–218, 1977.
- [11] G. Bowtell and T.L. Williams. Anguilliform body dynamics: modelling the interaction between muscle activation and body curvature. *Philosophical Transactions: Biological Sciences*, pages 385–390, 1991.
- [12] F. Boyer, M. Porez, and A. Leroyer. Poincaré–cosserat equations for the lighthill three-dimensional large amplitude elongated body theory: Application to robotics. *Journal of Nonlinear Science*, 20(1):47–79, 2010.
- [13] E.L. Brainerd and E. Azizi. Muscle fiber angle, segment bulging and architectural gear ratio in segmented musculature. *Journal of experimental biology*, 208(17):3249, 2005.
- [14] E.L. Brainerd and R.S. Simons. Morphology and function of lateral hypaxial musculature in salamanders. *American Zoologist*, 40(1):77, 2000.
- [15] J. Carling, T.L. Williams, and G. Bowtell. Self-propelled anguilliform swimming: simultaneous solution of the two-dimensional navier-stokes equations and newton’s laws of motion. *Journal of experimental biology*, 201:3143–3166, 1998.

- [16] J.Y. Cheng, TJ Pedley, and JD Altringham. A continuous dynamic beam model for swimming fish. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, 353(1371):981, 1998.
- [17] A. Crespi and A.J. Ijspeert. Amphibot ii: An amphibious snake robot that crawls and swims using a central pattern generator. In *Proceedings of the 9th international conference on climbing and walking robots (CLAWAR 2006)*, volume 11, pages 19–27. Citeseer, 2006.
- [18] K. D'Aout and P. Aerts. Kinematics and efficiency of steady swimming in adult axolotls (*ambystoma mexicanum*). *Journal of experimental biology*, 200:1863–1871, 1997.
- [19] K. D'Aout, P. Aerts, and F. De Vree. The timing of muscle strain and activation during steady swimming in a salamander, *ambystoma mexicanum*. *Netherlands Journal of Zoology*, 46, 3(4): 263–271, 1995.
- [20] MP De Looze, HM Toussaint, DA Commissaris, MP Jans, and AJ Sargeant. Relationships between energy expenditure and positive and negative mechanical work in repetitive lifting and lowering. *Journal of Applied Physiology*, 77(1):420, 1994.
- [21] S.M. Deban and N. Schilling. Activity of trunk muscles during aquatic and terrestrial locomotion in *ambystoma maculatum*. *Journal of Experimental Biology*, 212(18):2949, 2009.
- [22] I. Delvolv , T. Bem, and J.M. Cabelguen. Epaxial and limb muscle activity during swimming and terrestrial stepping in the adult newt, *pleurodeles waltl*. *Journal of neurophysiology*, 78(2): 638, 1997.
- [23]  . Ekeberg. A combined neuronal and mechanical model of fish swimming. *Biological cybernetics*, 69(5):363–374, 1993.
- [24] E.T.B. Francis. *The anatomy of the salamander*. The Clarendon Press, 1934.
- [25] L.M. Frolich and A.A. Biewener. Kinematic and electromyographic analysis of the functional role of the body axis during terrestrial and aquatic locomotion in the salamander *ambystoma tigrinum*. *Journal of Experimental Biology*, 162(1):107, 1992.
- [26] G. Gillis. Anguilliform locomotion in an elongate salamander (*siren intermedia*): effects of speed on axial undulatory movements. *Journal of experimental biology*, 200(4):767, 1997.
- [27] G.B. Gillis. Undulatory locomotion in elongate aquatic vertebrates: anguilliform swimming since sir james gray. *American zoologist*, 36(6):656, 1996.
- [28] J. Gray. Directional control of fish movement. *Proceedings of the Royal Society of London. Series B, Containing Papers of a Biological Character*, pages 115–125, 1933.
- [29] DFB Haeufle, S. Grimmer, and A. Seyfarth. The role of intrinsic muscle properties for stable hopping stability is achieved by the force-velocity relation. *Bioinspiration & Biomimetics*, 5: 016004, 2010.
- [30] N. Harischandra, J.M. Cabelguen, and  . Ekeberg. A 3d musculo-mechanical model of the salamander for the study of different gaits and modes of locomotion. *Frontiers in Neurorobotics*, 4, 2010.
- [31] H. Hatze. A myocybernetic control model of skeletal muscle. *Biological Cybernetics*, 25(2):103–119, 1977.
- [32] AV Hill. The mechanics of active muscle. *Proceedings of the Royal Society of London. Series B-Biological Sciences*, 141(902):104, 1953.

- [33] AV Hill. The effect of load on the heat of shortening of muscle. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 159(975):297, 1964.
- [34] E. Homsher, W. Mommaerts, and NV Ricchiuti. Energetics of shortening muscles in twitches and tetanic contractions. *The Journal of general physiology*, 62(6):677, 1973.
- [35] A. Ijspeert. A 3-d biomechanical model of the salamander. In *Virtual Worlds*, pages 225–234. Springer, 2000.
- [36] A.J. Ijspeert. A connectionist central pattern generator for the aquatic and terrestrial gaits of a simulated salamander. *Biological cybernetics*, 84(5):331–348, 2001.
- [37] A.J. Ijspeert, A. Crespi, D. Ryczko, and J.M. Cabelguen. From swimming to walking with a salamander robot driven by a spinal cord model. *Science*, 315(5817):1416, 2007.
- [38] E.R. Kandel, J.H. Schwartz, T.M. Jessell, S.A. Siegelbaum, and AJ Hudspeth. *Principles of neural science*, volume 3. Elsevier New York, 1991.
- [39] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Neural Networks, 1995. Proceedings., IEEE International Conference on*, volume 4, pages 1942–1948. IEEE, 1995.
- [40] G.A. Lichtwark and AM Wilson. Effects of series elasticity and activation conditions on muscle power output and efficiency. *Journal of experimental biology*, 208(15):2845, 2005.
- [41] G.A. Lichtwark and AM Wilson. A modified hill muscle model that predicts muscle power output and efficiency during sinusoidal length changes. *Journal of experimental biology*, 208(15):2831, 2005.
- [42] J.H. Long. Muscles, elastic energy, and the dynamics of body stiffness in swimming eels. *American zoologist*, 38(4):771, 1998.
- [43] J.H. Long et al. Force transmission via axial tendons in undulating fish: a dynamic analysis*. 1. *Comparative Biochemistry and Physiology-Part A: Molecular & Integrative Physiology*, 133(4): 911–929, 2002.
- [44] T.A. McMahon. *Muscles, reflexes, and locomotion*. Princeton Univ Pr, 1984.
- [45] O. Michel. Cyberbotics ltd. webotstm: Professional mobile robot simulation. *International Journal of Advanced Robotic Systems*, 1(1):39–42, 2004.
- [46] Berger S. Energy consumption optimization and stumbling corrective response for bipedal walking gait. Master’s thesis, EPFL, 2011.
- [47] N. Schilling and S.M. Deban. Fiber-type distribution of the perivertebral musculature in ambystoma. *Journal of Morphology*, 271(2):200–214, 2010.
- [48] M. Sfakiotakis, D.M. Lane, and J.B.C. Davies. Review of fish swimming modes for aquatic locomotion. *Oceanic Engineering, IEEE Journal of*, 24(2):237–252, 1999.
- [49] ICH Smith. Energetics of activation in frog and toad muscle. *The Journal of Physiology*, 220(3): 583, 1972.
- [50] Douglas A. Syme. Functional properties of skeletal muscle. In *Chapter 6, Volume 23, Fish Physiology: Fish Biomechanics*, volume 23, pages 179–240, 2006.
- [51] J.L. Van Leeuwen. A mechanical analysis of myomere shape in fish. *Journal of Experimental Biology*, 202(23):3405, 1999.

- [52] P. Wallén and TL Williams. Fictive locomotion in the lamprey spinal cord in vitro compared with swimming in the intact and spinal animal. *The Journal of physiology*, 347(1):225, 1984.
- [53] C. Wardle, J. Videler, and J. Altringham. Tuning in to fish swimming waves: body form, swimming mode and muscle function. *Journal of experimental Biology*, 198(8):1629, 1995.
- [54] R.J. Wassersug. Locomotion in amphibian larvae. *American Zoologist*, 29(1):65, 1989.
- [55] P.W. Webb. The swimming energetics of trout. i. thrust and power output at cruising speeds. *The Journal of experimental biology*, 55(2):489, 1971.
- [56] JB Wells. Comparison of mechanical properties between slow and fast mammalian muscles. *The Journal of Physiology*, 178(2):252, 1965.
- [57] L. Williams, S. Grillner, V.V. Smoljaninov, P.W. Allen, S. Kashin, and S. Rossignol. Short communication locomotion in lamprey and trout : the relative timing of activation and movement. *J. exp. Biol.*, 143:559–566, 1989.
- [58] T. Williams, G. Bowtell, and NA Curtin. Predicting force generation by lamprey muscle during applied sinusoidal movement using a simple dynamic model. *Journal of experimental biology*, 201(6):869, 1998.
- [59] JM Winters and L. Stark. Muscle models: what is gained and what is lost by varying model complexity. *Biological Cybernetics*, 55(6):403–420, 1987.
- [60] F.E. Zajac. Muscle and tendon: properties, models, scaling, and application to biomechanics and motor control. *Critical reviews in biomedical engineering*, 17(4):359, 1989.

A. Appendix

A.1. Tracking

Video	Kind	Animal number	Start frame	Stop frame	End frame	Nb_of_cycle	Circle Radius	Top	Side	Comments
28008	start-swim-stop	4	360	12	600	1	15	OK	OK	of side, use left instead of right
28011	swim	1	120	8	440	2	15	OK	OK	
28013	fall-swim	1					16			
28014	swim	2	400	10	600	1	16	OK	OK	
28015	swim	1	140	10	340	1	16	OK	OK	
28016	swim	2	280	7	420	1	16	OK	OK	
28017	fall-swim	1	80	5	260	2	16	OK	OK	
28018	fall-swim	2	100	4	240	2	16	OK	OK	
28019	fall-swim	?	100	6	220	1	17	OK	OK	
28020	swim	?	100	5	280	2	15	OK	OK	
#####	#####	#####	#####	#####	#####	#####	#####	###	###	
28001	fall-swim	2	180	8	340	1	15	OK	OK	
28002	swim	2	100	6	220	1	15	OK	OK	
28003	fall-swim	1					16			something strange
28004	fall-swim	1	180	9	360	1	17	OK	OK	
28005	fall-swim	1	100	7	240	1	15	OK	OK	
28006	swim	2	240	8	720	3	15	OK	OK	
28007	swim	4	120	4	280	2	15	OK	OK	From 208 to 268 step = 5
28009	swim	4	0	8	160	1	15	OK	OK	
28010	swim	4	140	7	280	1	15	OK	OK	
28011	swim	4	100	6	220	1	15	OK	OK	
28012	swim	4	140	7	280	1	15	OK	OK	little angle

Figure A.1.: Tracking summary sheet

Video	Kind	Animal number	Start frame	Step frame	End frame	Nb.of_cycle	Circle Radius	Top	Side	Comments
29013	swim	4	110	8	270	1	15	OK	OK	slowdown
29014	swim	4	160	6	280	1	15	OK	OK	little angle
29015	swim	4	260	12	500	1	15	OK	OK	little hit the wall
29016	swim	3	160	4	252	1	15	OK	OK	
29017	fall-swim	3	260	7	420	1	15	OK	OK	bdf
29018	swim	3					15		NO	
29019	swim	3					15		NO	
29020	swim	3	360	10	580	1	15	OK	OK	
29021	swim	3	560	12	800	1	15	OK	OK	
29022	fall-swim	3	140	13	400	1	15	OK	OK	
29023	swim	3	240	10	440	1	15	OK	OK	
#####	#####	#####	#####	#####	#####	#####	#####	#####	#####	#####
29024	no-tail-swim	5	300	10	700	2	15	OK	OK	
				Total cycle	37					
Rejected		Why?	Tracked not used		Why?	Animal number		Nb.of_cycle		
28001	u-walk		28009	swim and hit the wall		1	7			
28002	u-walk		28010	swim walk		2	9			
28003	u-walk		28012	swim and hit the wall		3	6			
28004	u-walk		28005	#####		4	10			
28005	u-walk		29008	Salamander 45°		5	2			
28006	u-walk					?	3			
28007	inexistent			Total Cycle	37					
#####	#####	#####	#####	#####	#####	#####	#####	#####		
29025	no-tail-u-walk									
29026	??-walk									
29027	??-walk									
29028	u-walk									

Figure A.2.: Tracking summary sheet

A.2. EMG

	SVL	BL	PhaselagAvg (% of cycle duration) ref 0.65SVL	PhaselagAvg (% of cycle duration) ref 0.25 SVL	Duty_cycle_avg
0.25	0.4796380.09	0.1199050523	0.3547304177		
0.3	0.4796380.09	0.1438914027	0.3211400259		
0.4	0.4796380.09	0.1918552036	0.3060394115		
0.46	0.4796380.09	0.2206354942	0.3016539343		
0.5	0.4796380.09	0.2398190045	0.2116663327		
0.51	0.4796380.09	0.2446153846	0.2393850827		
0.55	0.4796380.09	0.2638009005	0.3538828302		
0.5	0.4796380.09	0.287782054	0.3225066211		
0.74	0.4796380.09	0.3549321267	0.3091514203		
0.76	0.4796380.09	0.3645248869	0.3000790124		
0.8	0.4796380.09	0.3837104072	0.3854835595		
0.85	0.4796380.09	0.4076923077	0.2787122258		
0.9	0.4796380.09	0.4316742381	0.3207049244		
0.95	0.4796380.09	0.4556561086	0.2862534681		
1	0.4796380.09	0.4796380.09	0.3743337971		
1.1	0.4796380.09	0.52760181	0.3061644512		
1.2	0.4796380.09	0.575566109	0.2802400323		
1.7	0.4796380.09	0.6238294118	0.3900720019		
1.8	0.4796380.09	0.6238294118	0.2971234226		
1.5	0.4796380.09	0.7194570136			
30					
50					

Figure A.3.: Delvolv  data summary sheet

SVL	Burst duration	Cycle duration	Duty_cycle
0.25	149	427	0.348946136
0.3			
0.4			
0.46	138	427	0.323185012
0.5			
0.51	104	427	0.243559719
0.55	152	427	0.355971897
0.6	142	427	0.332552693
0.74	168	427	0.393442623
0.76	132	427	0.309133489
0.8	164	427	0.384074941
0.85	116	427	0.271662763
0.9	141	427	0.330210773
0.95	118	427	0.276346604
1	155	427	0.362997658
1.1	135	427	0.316159251
1.2	120	427	0.281030445
1.3	153	427	0.358313817
1.5	119	427	0.278688525
SVL	Burst duration	Cycle duration	Duty_cycle
0.25	164	473	0.348723044
0.3	154	473	0.325581395
0.4	156	473	0.329809725
0.46	139	473	0.293868922
0.5	103	473	0.217758985
0.51	111	473	0.234672304
0.55	174	473	0.367864693
0.6	159	473	0.33615222
0.74	196	473	0.414376321
0.76	137	473	0.289640592
0.8	183	473	0.386892178
0.85	129	473	0.272727273
0.9	150	473	0.317124736
0.95	120	473	0.253699789
1	177	473	0.374207188
1.1	146	473	0.308668076
1.2	134	473	0.283298097
1.3	150	473	0.317124736
1.5	140	473	0.295983087
SVL	Burst duration	Cycle duration	Duty_cycle
0.25	192	521	0.368522073
0.3	165	521	0.316698656
0.4	148	521	0.284069098
0.46	150	521	0.287907869
0.5	107	521	0.20537428
0.51	125	521	0.239923225
0.55	176	521	0.3378119
0.6	171	521	0.328214971
0.74	203	521	0.389635317
0.76	158	521	0.303262956
0.8			
0.85	152	521	0.291746641
0.9	164	521	0.314779271
0.95	140	521	0.268714012
1	201	521	0.385796545
1.1	153	521	0.293666027
1.2	144	521	0.276391555
1.3	164	521	0.314779271
1.5	165	521	0.316698656

Figure A.4.: Delvolve data summary sheet

A.3. Correlation analysis

<i>Angle_Amplitude</i>			<i>Lateral_Amplitude</i>		
type	R	P	type	R	P
median	0.3685	0.0492	median	0.147	0.4467
mean	0.24	0.2	mean	0.0966	0.61
Slope	0.1261	0.51	Slope	0.1161	0.5487
Hist 1:5	0.1633	0.4064	Hist 1:5	0.3085	0.1102
Hist 6:10	0.3796	0.0463	Hist 6:10	0.2771	0.1535
Hist 11:15	0.2007	0.3058	Hist 11:15	0.2293	0.2405
Hist 16:20	0.2752	0.1564	Hist 16:20	0.1183	0.5486
Hist 21:25	0.3127	0.1052	Hist 21:25	0.0168	0.9326
Hist 26:30	0.03	0.8794	Hist 26:30	0.0279	0.88

<i>Phase_lag</i>			<i>Frequency</i>		
type	R	P	type	R	P
median	-0.4232	0.02	median	0.7785	1.00E-006
mean	-0.1979	0.3			
Hist 1:5	0.0241	0.9			
Hist 6:10	-0.1081	0.584			
Hist 11:15	-0.3152	0.1023			
Hist 16:20	0.0267	0.8926			
Hist 21:25	0.1406	0.4754			
Hist 26:30	-0.3884	0.0411			

Figure A.5.: Correlation analysis