
Student Names: … (please update)

Instructions: Update this file (or recreate a similar one, e.g. in Latex) to prepare your answers to the questions. Add text, equations and figures as needed. Please submit both the source file (.doc) and a pdf of your document, as well as all the used and updated Matlab functions in a single zipped file called lab7_name1_name2_name3.zip where name# are the team member's last names. Please submit **only one report per team**!*

10 Swimming with Salamandra Robotica – CPG Model

In this exercise you should implement a CPG based swimming controller for Salamandra Robotica. The Webots simulation is similar to the one of Lab 6, with slight changes in the script `matlabController.m` and several added functions.

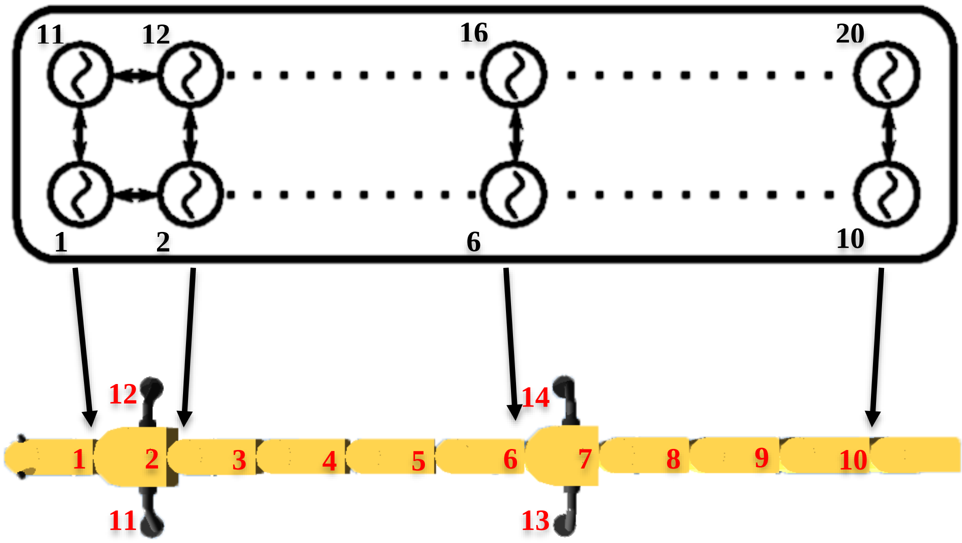


Figure 1 A double chain of oscillators controlling the robot's spine.

10.A Implement a double chain of oscillators.

Salamandra Robotica has 10 joints along its spine. Implement a double chain (dynamics described in equations 1 and 2) of oscillators in a way that lateral oscillator pairs are assigned to a single joint (see Figure 1). No need to implement leg oscillators for this exercise.

$$\dot{\theta}_i = 2\pi f + \sum_j r_j\, w_{ij} \sin(\theta_j - \theta_i - \phi_{ij}) \qquad (1)$$
$$\dot{r}_i = a(R_i - r_i) \qquad (2)$$
$$qs_i = r_i(1 + \cos(\theta_i)) - r_{i+10}(1 + \cos(\theta_{i+10})) \qquad (3)$$

θ_i : oscillator phase
f : frequency
w_{ij} : coupling weights
ϕ_{ij} : nominal phase lag
r_i : oscillator amplitude
R_i : nominal amplitude
a : convergence factor
qs_i : spinal joint angles

1) Complete `runCPGNetwork.m` (see function description). **Include your implemented code from this function in the report.** For the parameters check slide 22, lecture 6 (pay attention to different number of segments).

2) Use the output of your CPG network to generate the spinal joint angles according to equation 3. Implement this in the function `spineController.m`. Verify first in Matlab and then in Webots that your code works as expected. Use the function `loadLogs.m` to report your spinal joint angles qs_i.

10.B Effects of amplitude and phase lags on swimming performance

How does phase lag and oscillation amplitude influence the speed and energy? Use the provided grid search functions to explore the robot behavior for different combinations of amplitudes and phase lags. Implement your speed and energy measurement in the function `saveGridSearch`. Include 3D plots showing your grid search results and discuss them. How do your findings compare to the wavelengths observed in the salamander? Run the grid search twice, for frequencies of **1Hz** and **2Hz**.

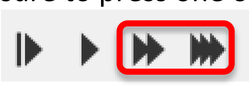
Hint 1: To use the grid search, in `matlabController.m` set the flag `GRID_SEARCH` to true. Provide the desired parameter vectors to the function `initializeGridSearch`. The function will repeatedly run simulations, each time with a different combination of parameters from the input vectors. The results are stored in the structure "`filename`"`.mat` which is located in your current Matlab folder. After the grid search finishes, the simulation will automatically pause itself.

Hint 2: An example how to load and visualize grid search results is shown in the script `loadGridSearch.m`. Pay attention to the name of the `.mat`-file you are loading. Before starting a new grid search, change the name of the `.mat`-file where the results will be stored. In case a grid search failed, delete the `.mat`-file before starting the new grid search.

Hint 3: Estimate how long it will take to finish the grid search. For example if you define both input parameter vectors as:

```
phi_total_vector=linspace(pi/4, 4*pi, 10);
R_vector=linspace(0.01, 1, 10);
```

will lead to 100 different simulations, each lasting `Trun` seconds to finish the grid search. Our suggestion is to choose wisely (not in this example) lower and upper limits of parameter vectors and limit their size to maximum 7 (which will result with 49 simulations). To speed-up a simulation, make sure to press one of those two buttons:



Hint 4: Energy can be estimated by integrating the product of instantaneous joint velocities and torques. Feel free to propose your own energy metrics, just make sure to include the justification. Check an example for distance metric in the function `saveGridSearch` to see how to implement your speed and energy metric.

10.C Amplitude gradient

1) So far we considered constant undulation amplitudes along the body for swimming. Implement a linear distribution of amplitudes along the spine, parametrized with two parameters: amplitudes of the first (`Rhead`) and last (`Rtail`) oscillator in the spine (corresponding to the first and last motor). To do so, modify the function `runCPGNetwork` that the input variable `R=[Rhead Rtail]` now has the size 2, and within the function you interpolate the amplitude gradient between values `Rhead` and `Rtail`.

2) Run a grid search over different values of parameters `Rhead` and `Rtail` (use the same range for both parameters). How does the amplitude gradient influence swimming performance (speed, energy)? Include 3D plots showing your grid search results. Do it once, for frequency **1Hz** and total phase lag of **2pi** along the spine.

3) How is the salamander moving (with respect to different body amplitudes)? How do your findings in **2)** compare to body deformations in the salamander? Based on your explorations, what could be possible explanations why the salamander moves the way it does?

10.D Turning and backwards swimming

1) How do you need to modulate the CPG network (`runCPGNetwork.m`) in order to induce turning? Implement this in the Webots model and plot example GPS trajectories and spine angles.

2) How could you let the robot swim backwards? Explain and plot example GPS trajectories and spine angles.