Marie Anselmet
Inès De Riedmatten
Héloïse Monnet

Student names: Marie Anselmet, Inès De Riedmatten, Héloïse Monnet

Instructions: Update this file (or recreate a similar one, e.g. in Word) to prepare your answers to the questions. Feel free to add text, equations and figures as needed. Hand-written notes, e.g. for the development of equations, can also be included e.g. as pictures (from your cell phone or from a scanner). ***This lab is graded.*** *and must be submitted before the* ***Deadline : 22-04-2020 23:59****. Please submit both the source file (*.doc/*.tex) and a pdf of your document, as well as all the used and updated Python functions in a single zipped file called lab6_name1_name2_name3.zip where name# are the team member's last names. Please submit only one report per team!*

The file `lab#.py` *is provided to run all exercises in Python. The list of exercises and their dependencies are shown in Figure 1. When a file is run, message logs will be printed to indicate information such as what is currently being run and and what is left to be implemented. All warning messages are only present to guide you in the implementation, and can be deleted whenever the corresponding code has been implemented correctly.*

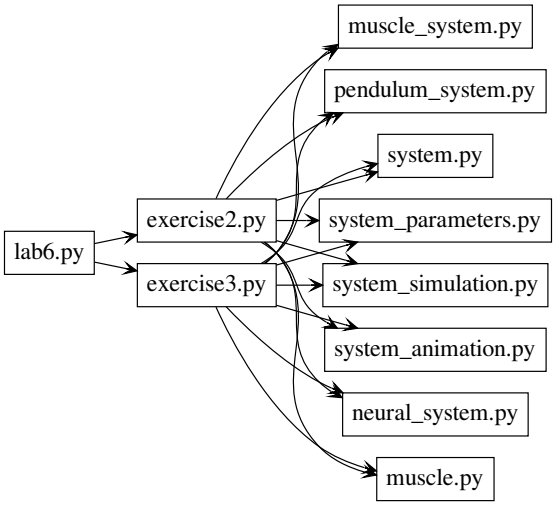

Figure 1: Exercise files dependencies. In this lab, you will be modifying `exercise2.py`, `exercise3.py` *and optionally* `pendulum_system.py`

Files to complete the exercises

- `lab6.py` : Main file
- `exercise2.py` : Main file to complete exercise 2
- `exercise3.py` : Main file to complete exercise 3
- `system_parameters.py` : Parameter class for Pendulum, Muscles and Neural Network (Create an instance and change properties using the instance. You do not have to modify the file)
- `muscle.py` : Muscle class (You do not have to modify the file)
- `system.py` : System class to combine different models like Pendulum, Muscles, Neural Network (You do not have to modify the file)
- `pendulum_system.py` : Contains the description of pendulum equation and Pendulum class. You can use the file to define perturbations in the pendulum.
- `muscle_system.py` : Class to combine two muscles (You do not have to modify the file)

Marie Anselmet
Inès De Riedmatten
Héloïse Monnet

- `neural_system.py` : Class to describe the neural network (You do not have to modify the file)
- `system_simulation.py` : Class to initialize all the systems, validate and to perform integration (You do not have to modify the file)
- `system_animation.py` : Class to produce animation of the systems after integration (You do not have to modify the file)

NOTE : '*You do not have to modify*' does not mean you should not, it means it is not necessary to complete the exercises. But, you are expected to look into each of these files and understand how everything works. You are free to explore and change any file if you feel so.

Exercise 2 : Pendulum model with Muscles

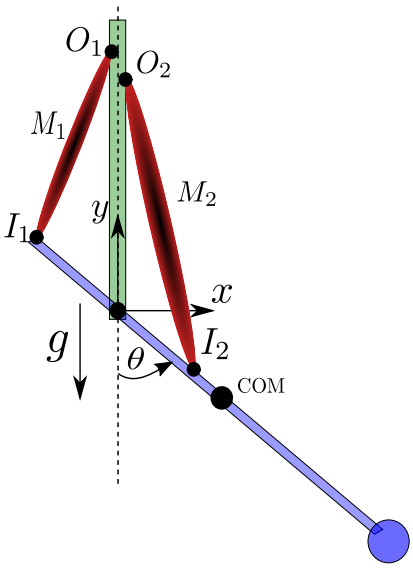

Figure 2: Pendulum with Antagonist Hill Muscles

The system is comprised of a physical pendulum described by equation 1 and a pair of antagonist muscles **M1** and **M2**. Muscle **M1** extends the pendulum (θ increases) and Muscle **M2** flexes the muscle (θ decreases).

Consider the system only for the pendulum range $\theta = [0, \pi]$

$$I\ddot{\theta} = -0.5 \cdot m \cdot g \cdot L \cdot sin(\theta) \tag{1}$$

Where,

- I - Pendulum inertia about the pendulum pivot joint $[kg \cdot m^2]$
- θ - Pendulum angular position with the vertical $[rad]$
- $\ddot{\theta}$ - Pendulum angular acceleration $[rad \cdot s^{-2}]$
- m - Pendulum mass $[kg]$
- g - System gravity $[m \cdot s^{-2}]$
- L - Length of the pendulum $[m]$

Each muscle is modelled using the Hill-type equations that you are now familiar with. Muscles have two attachment points, one at the origin and the other at the insertion point. The origin points are

Marie Anselmet
Inès De Riedmatten
Héloïse Monnet

denoted by $O_{1,2}$ and the insertion points by $I_{1,2}$. The two points of attachment dictate how the length of the muscle changes with respect to the change in position of the pendulum.

The active and passive forces produced by the muscle are transmitted to the pendulum via the tendons. In order to apply this force on to the pendulum, we need to compute the moment based on the attachments of the muscle.

Using the laws of sines and cosines, we can derive the length of muscle and moment arm as below. The reference to the paper can be found here Reference,

$$L_2 = \sqrt[2]{a_1^2 + a_2^2 + 2 \cdot a_1 \cdot a_2 \cdot \cos(\theta)} \tag{2}$$

$$h_2 = \frac{a_1 \cdot a_2 \cdot \sin(\theta)}{L_2} \tag{3}$$

Where,

- L_2 : Length of muscle 2
- a_1 : Distance between muscle 2 origin and pendulum origin ($|O_2C|$)
- a_2 : Distance between muscle 2 insertion and pendulum origin ($|I_2C|$)
- h_2 : Moment arm of the muscle

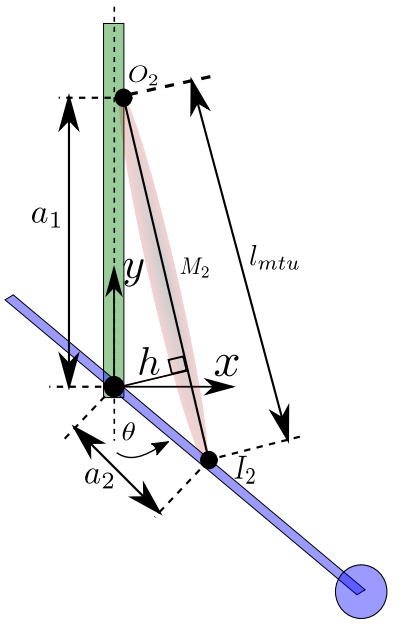

Figure 3: Computation of muscle length and moment arm

Equation 2 can be extended to the Muscle 2 in similar way. Thus, the final torque applied by the muscle on to the pendulum is given by,

$$\tau = F \cdot h \tag{4}$$

NOTE : τ can also be computed as the cross product of the force vector and radial vector. This method is used in the code to compute the torques and there by bypassing the moment arm computation.

Where,

Marie Anselmet
Inès De Riedmatten
Héloïse Monnet

- τ : Torque $[N \cdot m]$
- F : Muscle Tendon Force $[N]$
- h : Muscle Moment Arm $[m]$

In this exercise, the following states of the system are integrated over time,

$$X = \begin{bmatrix} \theta & \dot{\theta} & A_1 & l_{CE1} & A_2 & l_{CE2} \end{bmatrix} \tag{5}$$

Where,

- θ : Angular position of the pendulum [rad]
- $\dot{\theta}$: Angular velocity of the pendulum [rad/s]
- A_1 : Activation of muscle 1 with a range between [0.05, 1]. 0.05 corresponds to base line stimulation and 1 corresponds to maximal stimulation.
- l_{CE1} : Length of contractile element of muscle 1
- A_2 : Activation of muscle 2 with a range between [0.05, 1]. 0.05 corresponds to base line stimulation and 1 corresponds to maximal stimulation.
- l_{CE2} : Length of contractile element of muscle 2

To complete this exercise you will make use of the following files, `exercise2.py`, `system_parameters.py`, `muscle.py`, `system.py`, `pendulum_system.py`, `muscle_system.py`, `system_simulation.py`

Marie Anselmet
Inès De Riedmatten
Héloïse Monnet

2a. For the given default set of attachment points, compute and plot the muscle length and moment arm as a function of θ between $[pi/4, 3pi/4]$ using equations in eqn:2 and discuss how it influences the pendulum resting position and the torques muscles can apply at different joint angles. You are free to implement this code by yourself as it does not have any other dependencies.

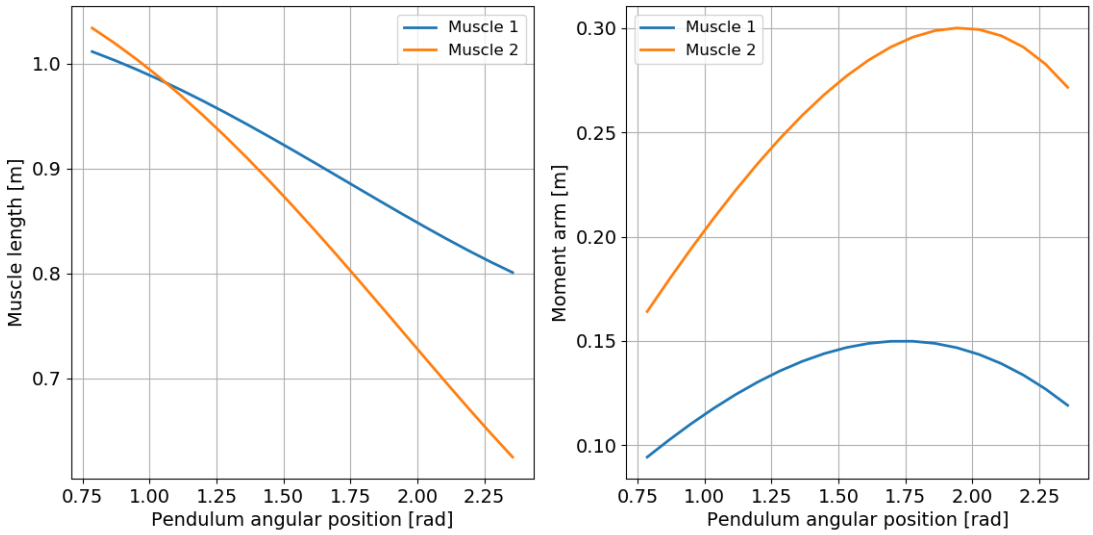

Figure 4: Muscle length and moment arm as a function of the pendulum angular position θ for muscles 1 and 2

First notice that given the default set of attachment points, muscles 1 and 2 should be considered as a pair of antagonist muscles as described in figure 2. But as in equations 2 and 3 there is no notion of direction (we only use the distance between the muscles attachment points and the pendulum origin), the following discussion does not take into account the flexor/extensor behavior of the muscles and so the direction of the torque applied by each muscle to the pendulum.

It can be seen in figure 4 that the muscle length decreases as θ increases for both muscles. Regarding the muscles moment arm, it increases for θ between (approximatively) $[\pi/4, 3\pi/5]$ and decreases a bit for θ between $[3\pi/5, 3\pi/4]$.

In this model, the torque applied by the muscles to the pendulum is determined by $\tau = F \cdot h$. So with a constant muscle force:

- if the muscles are long (i.e they loosen up), it creates a small muscle moment arm and thus a small torque is applied on the pendulum, that is the pendulum has a small pendulum angular position.
- if the muscles are short (i.e they contract), it creates a big moment arm and thus a big torque, that is the pendulum has a big pendulum angular position.

These plots also show that the distance between the muscles attachment points and the pendulum origin changes the muscles length and moment arm. In this case, muscle 2 attachment points are closer of the pendulum origin than those of muscle 1, inducing that muscle 2 moment arm is much bigger than muscle 1 moment arm. So muscle 2 applies a bigger torque to the pendulum than muscle 1.

Marie Anselmet
Inès De Riedmatten
Héloïse Monnet

2b. Using simple activation wave forms (example : sine or square waves) applied to muscles (use `system_simulation.py::add_muscle_stimulations` method in `exercise2.py`), try to obtain a limit cycle behavior for the pendulum. Use relevant plots to prove the limit cycle behavior. Explain and show the activations wave forms you used. Use `pendulum_system.py::PendulumSystem::pendulum_system` function to perturb the model.

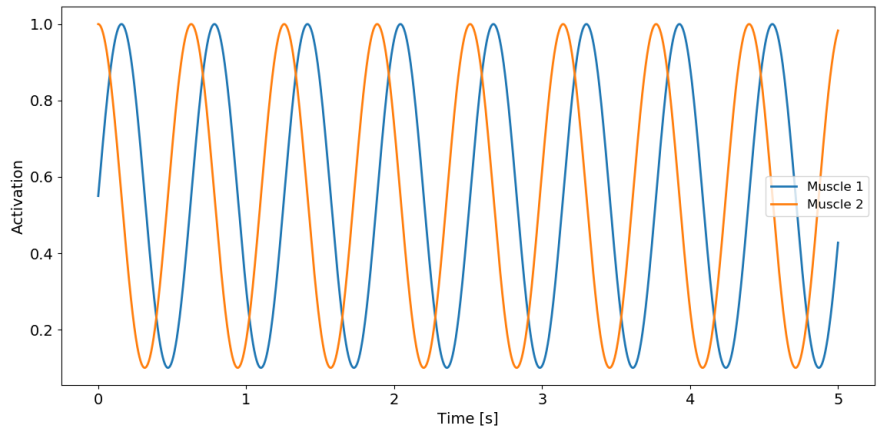

Figure 5: Activation wave forms used to stimulate muscles 1 and 2. Activation $= 0.45\sin(10t) + 0.55$ *for both muscles, with a phase lag of* $\pi/2$ *for muscle 2.*

The sine waves shown in figure 5 were used for muscles activation in order to obtain a limit cycle behavior for the pendulum. More precisely, the muscles activation followed the equation *Activation* $= a\sin(bt + c) + d$ with:

- $a = 0.45$ and $d = 0.55$ for the activation to be in range [0.05, 1]
- $b = 10$ to have several oscillations during the simulation time interval of 5s
- $c = 0$ for muscle 1 activation and $c = \pi/2$ for muscle 2 activation.

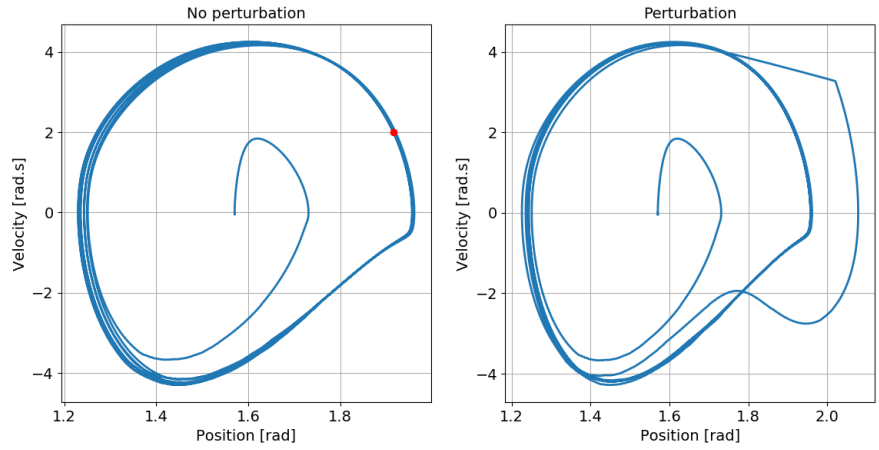

Figure 6: Phase portrait of the pendulum with and without perturbation. The perturbation consists in zeroing the torque applied to the pendulum during the time interval [1.2, 1.3]s. The red point corresponds to the Pointcaré section (see below).

As it can be seen in figure 6, the activation wave forms used enable to get a clear limit cycle behavior for the pendulum, with an unstable fixed point inside. Even after being perturbed, by setting to zero the torque applied to the pendulum between time interval [1.2, 1.3]s, the system converges to the stable limit cycle.

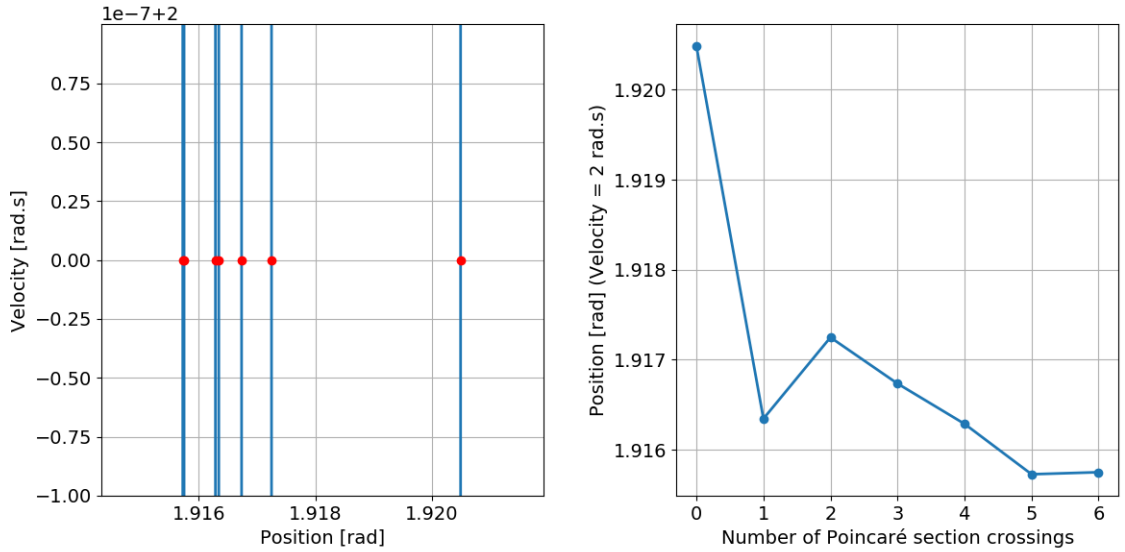

Figure 7: Crossings of the Poincaré section (left) lead to a Poincaré map (right).

In order to see more clearly the convergence to the limit cycle, a Poincaré map was plotted in figure 7. It is visible that, for a velocity fixed to 2 rad.s, the position of the pendulum rapidly converges to the almost constant value of 1.916 rad.

2c. Explore the relationship between stimulation frequency with the resulting pendulum's behavior. Report your inferences for a low and high frequency condition.

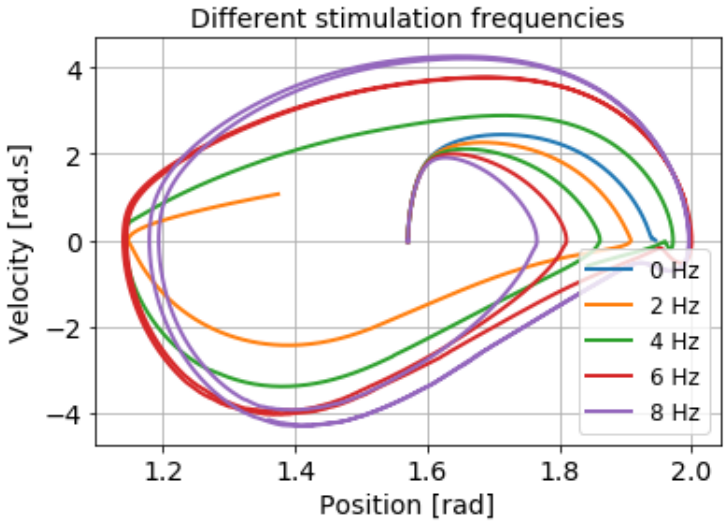

Figure 8: Phase portrait of the pendulum with different stimulation frequencies of the sinusoidal wave stimulation. Simulation time = 2 sec. Activation wave used to stimulate muscles 1 and 2 $= 0.45\sin(frequency * t) + 0.55$ *for both muscles, with a phase lag of* $\pi/2$ *for muscle 2.*

Marie Anselmet
Inès De Riedmatten
Héloïse Monnet

With increasing stimulation frequencies, we observe that the limit cycle becomes wider on the velocity axis, which means that higher velocities are reached in norm. In term of position, the range of θ reached by the pendulum is the same for frequencies between 0 and 6 Hz but it is a bit decreased for the 8 Hz stimulation. From 4 Hz, the limit cycle behavior is reached before 2 seconds of simulation, whereas for smaller frequencies, the limit cycle behavior is not reached. The stimulation is not strong enough. Indeed, with higher frequencies of stimulation, the temporal summation happens and the contraction of the muscle increases, which is translated into an increase of the velocity.

Exercise 3 : Neural network driven pendulum model with muscles

In this exercise, the goal is to drive the above system (Fig 2) with a symmetric four-neuron oscillator network. The network is based on Brown's half-center model with fatigue mechanism. Here we use the leaky-integrate and fire neurons for modelling the network. Figure 9 shows the network structure and the complete system.

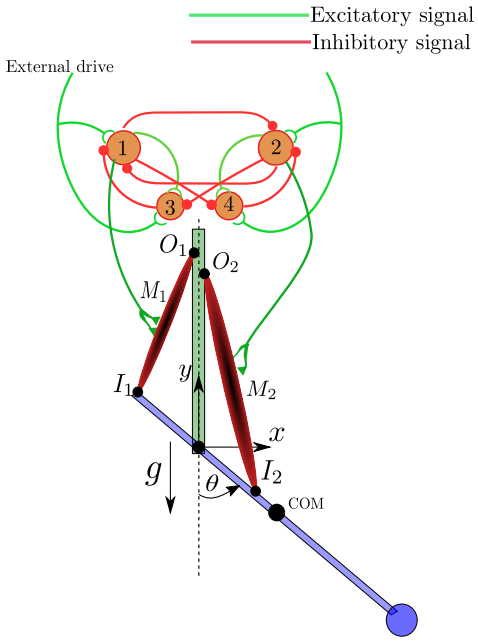

Figure 9: Pendulum with Antagonist Hill Muscles Driven Half Center Neural Network.

Since each leaky-integrate and fire neuron comprises of one first order differential equation, the states to be integrated now increases by four(one state per neuron). The states are,

$$X = \begin{bmatrix} \theta & \dot{\theta} & A_1 & l_{CE1} & A_2 & l_{CE2} & m_1 & m_2 & m_3 & m_4 \end{bmatrix} \tag{6}$$

Where,

- m_1 : Membrane potential of neuron 1
- m_2 : Membrane potential of neuron 2
- m_3 : Membrane potential of neuron 3
- m_4 : Membrane potential of neuron 4

To complete this exercise, additionally you will have to use `neural_system.py` and `exercise3.py`

Marie Anselmet
Inès De Riedmatten
Héloïse Monnet

3a. Find a set of weights and time constants for the neural network that produces oscillations to drive the pendulum into a limit cycle behavior. Plot the output of the network and the phase plot of the pendulum

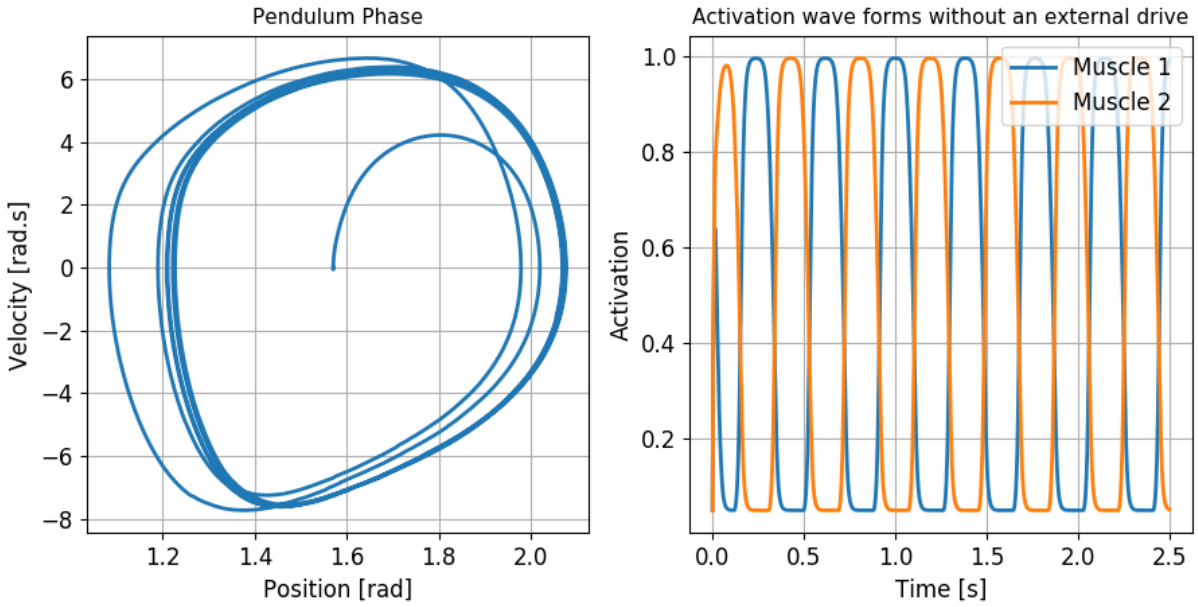

Figure 10: Phase portrait of the pendulum drived with a symmetric four-neuron oscillator network

In this system, neurons 1 and 2 are part of the half-center oscillators of Brown (1914). The idea of this model was to produce central oscillations. Neurons 1 and 2 reciprocally inhibit each other, which is responsible for the contralateral inhibition. To produce oscillations, neurons 3 and 4 are added to the network. They are called the slower neurons and represent the fatigue mechanism (the inhibition thanks to the negative feedback loop). If neuron 1 is activated, it will activate neuron 3, which will inhibit neuron 1. The same negative feedback mechanism happens with neurons 2 and 4. Without external drive, neuron 1 activates muscle 1 and neuron 2 activates muscle 2. The contralateral inhibition is useful so that the two antagonist muscles do not contract at the same time. Indeed, muscle 1 and 2 act in opposite ways. Concerning the negative feedback, it enables to end the contraction of the corresponding muscle, and to begin a new cycle. With the appropriate weights, the system exhibits a limit cycle behavior, as seen on figure 10.

The weights are given according to the symmetric four-neuron oscillator network seen in lecture 4, p.85:

$$\begin{bmatrix} w_{11} & w_{21} & w_{31} & w_{41} \\ w_{12} & w_{22} & w_{32} & w_{42} \\ w_{13} & w_{23} & w_{33} & w_{43} \\ w_{14} & w_{24} & w_{34} & w_{44} \end{bmatrix} = \begin{bmatrix} 0 & -5 & -5 & 0 \\ -5 & 0 & 0 & -5 \\ 5 & -5 & 0 & 0 \\ -5 & 5 & 0 & 0 \end{bmatrix}$$

Those weights are set in `exercise3.py::system_init`. The negative weights account for inhibition, whereas the positive weights correspond to activation. For the time constants τ, they are already set at [0.02, 0.02, 0.1, 0.1], as stated in class.

3b. As seen in the course, apply an external drive to the individual neurons and explain how the system is affected. Show plots for low [0] and high [1] external drives. To add external drive to the network you can use the method `system_simulation.py::add_external_inputs_to_network`

When there is no external drive applied to the individual neurons, we are in the same situation as in the previous question. Look at figure 11 for more detailed plots of the pendulum phase and state, and for a plot of the activation wave forms of both muscles when the external drive is null.

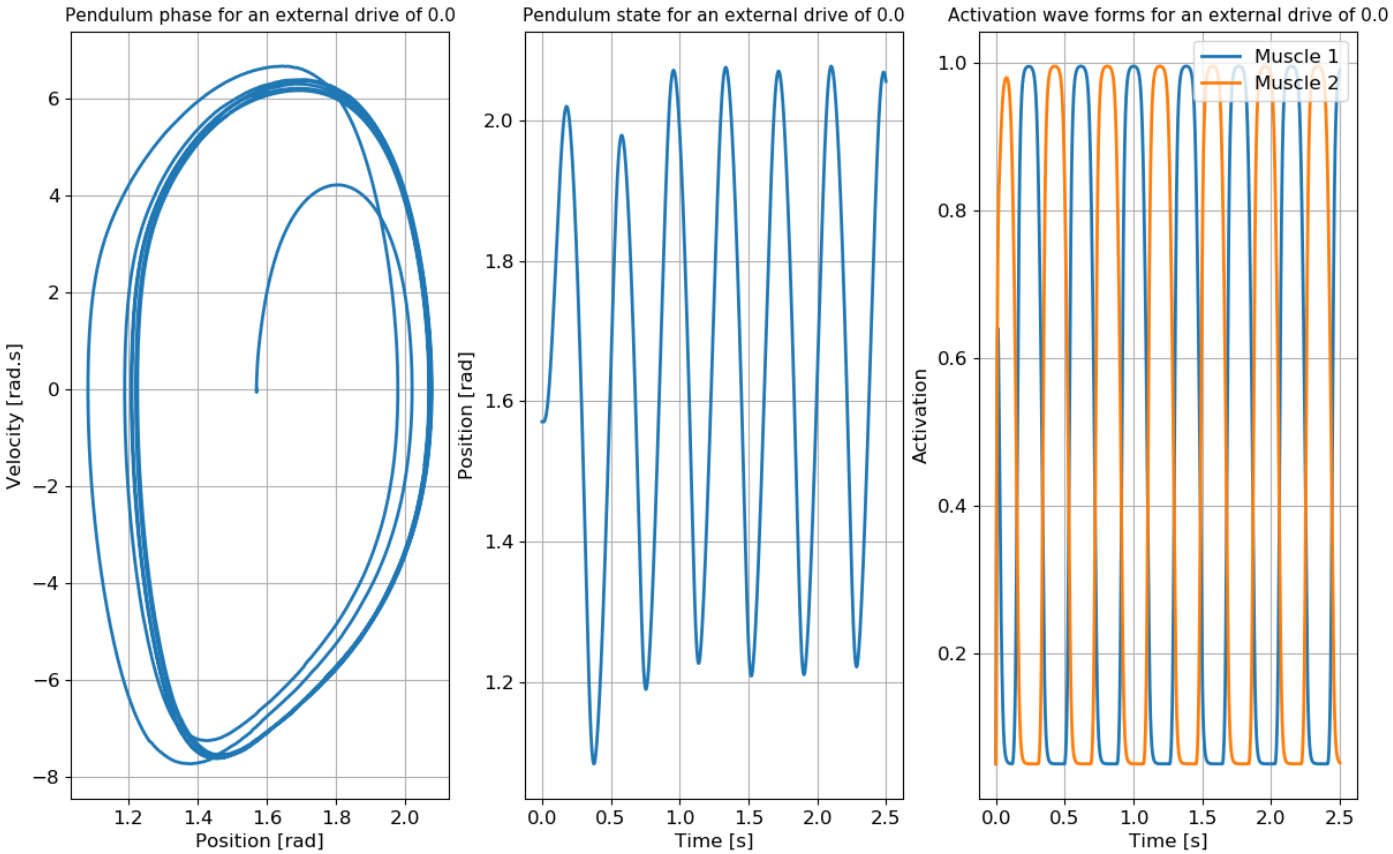

Figure 11: Pendulum phase, state, and activation wave forms of both muscles for a low external drive (0.0)

By comparing this with figure 12 where an high external drive of 1.0 is applied to the individual neurons of the network, we can derive the following points:

- The higher the drive is, the higher is the frequency of the oscillations of the pendulum system
- The higher the drive is, the smaller is the amplitude of the oscillations of the pendulum system

The increase in frequency of the oscillations as the external drive input increases is coherent. Indeed, when the neural network is more stimulated, this leads to a stronger output of the latter, giving rise to a higher frequency of muscle stimulation. The muscles contracting at higher rates, the pendulum system oscillates faster.
On the other hand, the angular velocity of the pendulum system does not seem to sufficiently increase to counter-balance the increased frequency. Indeed, we can extract from the plots that the higher is the external drive, the smaller is the amplitude of the oscillations of the pendulum system, as if the pendulum did not have enough time and velocity at each oscillation to reach a large angle of rotation.

Marie Anselmet
Inès De Riedmatten
Héloïse Monnet

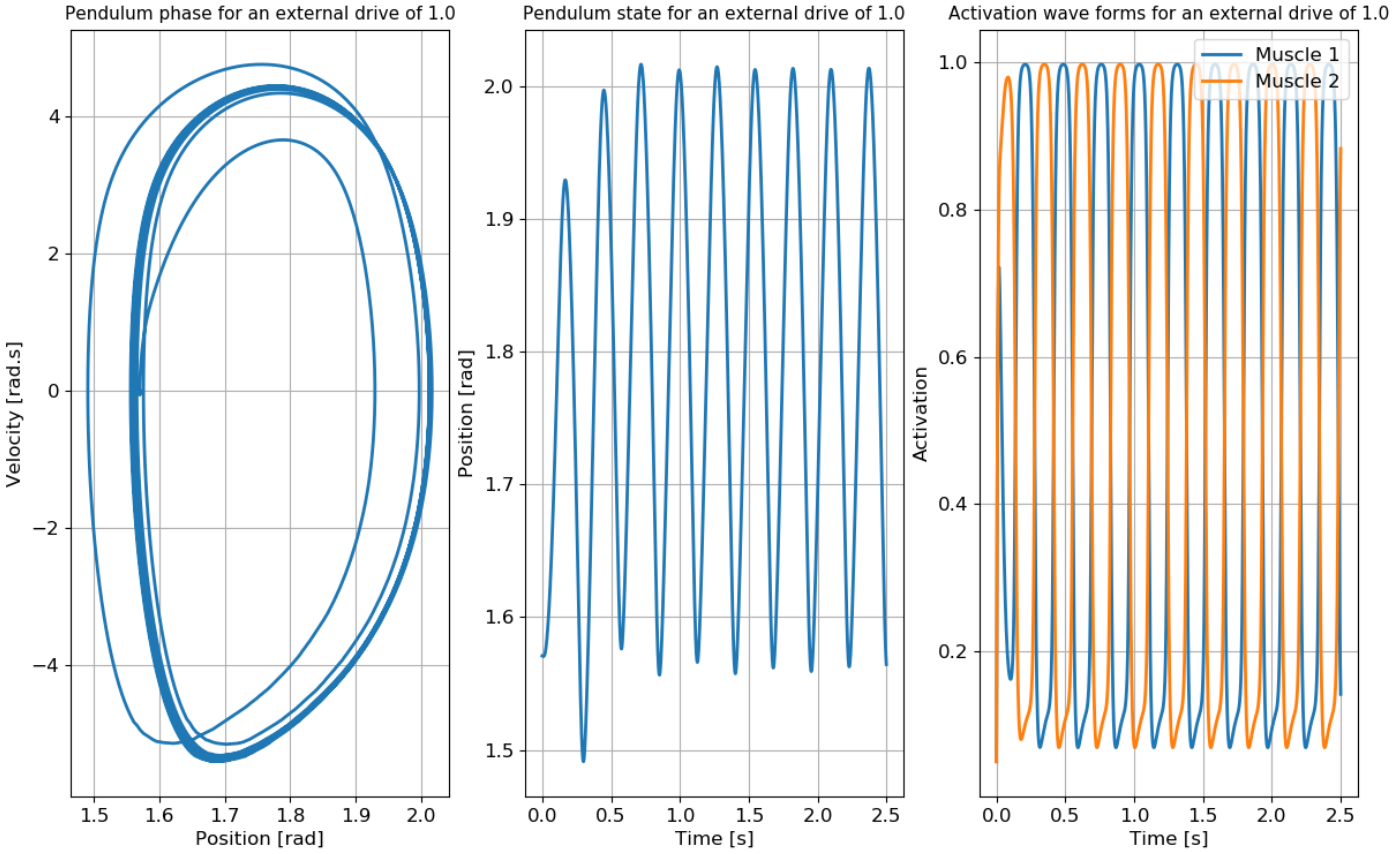

Figure 12: Pendulum phase, state, and activation wave forms of both muscles for a high external drive (1.0)

3c. [Open Question] What are the limitations of the half center model in producing alternating patterns to control the pendulum? What would be the effect of sensory feedback on this model? (No plots required)

The half center model is a quite simple model that enables to produce "basic" oscillations, like observed in this muscle-driven pendulum system. It enables also to model the fatigue mechanism by implementing a negative feedback loop with slower neurons. Nonetheless, real muscles are able to produce more complex oscillatory movements, like for example accelerating to a certain position and maintaining it in a stable equilibrium, which is not possible with this implementation of our pendulum system.
If a sensory feedback was added to this model, it would be possible to modulate the movement of the pendulum system according to sensory informations about the environment and the actual state of other muscles. These aspects are crucial to enable adaptation to the environment (especially to handle perturbations) and synchronization of muscle movements. Sensory feedback could even explain transitions between different gaits.

Marie Anselmet
Inès De Riedmatten
Héloïse Monnet

Appendix I : Hill muscle model equations

Name	Equation	Comment
F_{mtu}	$F_m = F_{se} = F_{ce} + F_{pe} - F_{be}$	Force of generated by the MTU
F_{se}	$F_{se} = \begin{cases} F_{max} \cdot (\frac{\varepsilon}{\varepsilon_{ref}})^2, & \text{if } \varepsilon > 0 \\ 0, & \text{otherwise} \end{cases}$	Force generated by the tendon
F_{pe}	$F_{pe} = \begin{cases} F_{max} \cdot (\frac{l_{ce}-l_{opt}}{l_{opt}w})^2 \cdot f_v(v_{ce}), & \text{if } l_{ce} > l_{opt} \\ 0, & \text{otherwise} \end{cases}$	Force of the parallel element preventing muscle overextension
F_{be}	$F_{be} = \begin{cases} F_{max} \cdot (\frac{l_{opt}-w-l_{ce}}{l_{opt}w/2})^2, & \text{if } l_{ce} \leq l_{opt} \cdot (1-w) \\ 0, & \text{otherwise} \end{cases}$	Force of the parallel element preventing muscle to collapse
F_{ce}	$F_{ce} = A \cdot F_{max} \cdot f_l(l_{ce}) \cdot f_v(v_{ce})$	Force generated by the muscle (ce)
A	$\frac{dA}{dt} = \tau \cdot (S(t) - A)$	Muscle activation is equal to the integral of the input signal I(t). Biologically, the input signal can be viewed as the normalized frequency of neuronal spike that reaches the muscle (i.e. restricted to the [0.05; 1] interval). The activation is thus also limited to this interval.
$f_l(l_{ce})$	$f_l(l_{ce}) = exp(c\lvert\frac{l_{ce}-l_{opt}}{l_{opt}w}\rvert^3)$	Muscle force - muscle length relationship function (function of the muscle length (lce)
$f_v(v_{ce})$	$f_v(v_{ce}) = \begin{cases} \frac{v_{max}-v_{ce}}{v_{max}+k \cdot v_{max}}, & \text{if } v_{ce} < 0 \\ N + (N-1)\frac{v_{max}-v_{ce}}{7.56 \cdot K \cdot v_{ce} - v_{max}}, & \text{otherwise} \end{cases}$	Muscle force - muscle velocity relationship function (function of the muscle velocity (vce)
$v_{ce}(f_v)$	$v_{ce}(f_v) = \begin{cases} v_{max} \cdot \frac{1-f_v}{1+f_v \cdot K}, & \text{if } f_v < 1.0 \\ v_{max} \cdot \frac{f_v - 1}{7.56 \cdot K \cdot (f_v - N) + 1 - N}, & \text{otherwise} \end{cases}$	Inverse of the muscle force - velocity function. This function is used for the resolution of the muscles equations.
l_{ce}	$\int v_{ce} \cdot dt$	Muscle (ce) length
l_{se}	$l_{se} = l_{mtu} - l_{ce}$	Tendon (se) length
l_{mtu}	$l_{mtu} = l_{opt} + l_{slack} + \delta l_{mtu}$	MTU length