# Neural Network Based Controller

Assignment 8 Submission Date: 23.06.2024
Name: Mateus Bonelli Salomão required time (hh:mm): 24:00

## Introduction

In this assignment, the task is to program a neural-network-based controller described in Tao, Bernd, and Florentin (2006) to move the Jena Fox robot in the provided multibody simulation model in Simulink. Before the controller can actually be implemented, it is necessary to adjust the given network and neuron signal equations to be consistent with the ISB convention used in the angle measurements of the Simulink model, which differs from the one used in the paper.

Firstly, the neural structure of the controller (in the author's original convention) will be briefly summarized.

The ISB convention will then be illustrated, and the adjusted network structure and equations used in the controller implementation will be presented.

Finally, the outputs of the implemented controller for the sample dataset given in Moodle will be shown in plots, as well as captured frames from the multibody simulation showcasing the robot's continuous walking gait.

## Summary - Neural Control

The sensor-driven walking controller of Tao, Bernd, and Florentin (2006) follows a hierarchical structure (see Figure 2 there). The bottom level represents the neuron modules local to the joints, including motor neurons and angle sensor neurons. The top level is a distributed neural network consisting of hip stretch receptors and ground contact sensor neurons, which modulate the motor neurons of the bottom level.

### Neural circuit - Top level

The ground contact sensor neuron of each leg has excitatory connections to the motor neurons of the hip flexor and knee extensor of the same leg as well as to the hip extensor and knee flexor of the other leg. The stretch receptor of each hip has excitatory (inhibitory) connections to motor neuron of the knee exten- sor (flexor) in the same leg. The function of stretch receptors AL/AR is to trigger the extensor motor neuron on the knee joint of a leg, upon reception of the AEA (Anterior Extreme Angle) signal of its hip joint.

### Neural circuit - Bottom level

The neuron module for each joint is composed of two angle sensor neurons and the motor neurons they contact (see Figure 2). Whenever its threshold is exceeded, the angle sen- sor neuron directly inhibits the corresponding motor neuron. In addition, each motor neuron also receives an excitatory synapse and an inhibitory synapse from the neurons of the top level.

## Problem Specification / Definitions

### ISB Convention and Definitions

Figure 1 shows a sketch of the robot with the ISB convention for the joint angles of the hips and knees, as well as the corresponding thresholds for extension and flexion. It must be noted that for the hip joints, a rotation in the clockwise direction corresponds to an extension of the hip, whilst a counterclockwise rotation corresponds to hip flexion. Conversely, a rotation in the clockwise direction of the knee joints corresponds to knee flexion, while counterclockwise rotations correspond to knee extension.

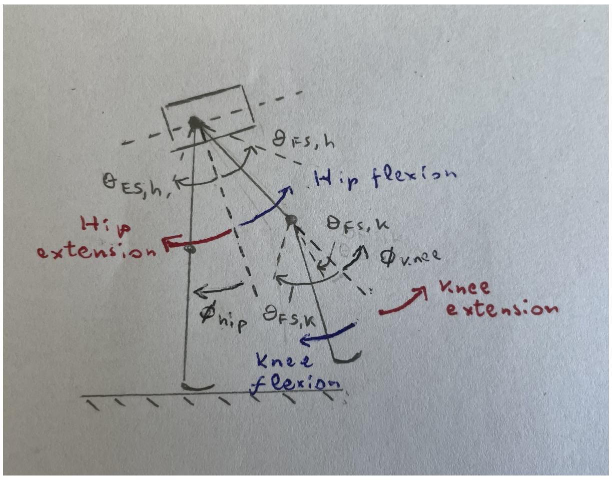


Figure 1: Problem sketch showing the ISB angle convention.

Table 1 gives an overview to all used symbols, constants, variables and initial conditions.

Table 1: Declaration of used symbols, constants, variables and initial conditions.

| Symbol | Property | Value | Unit |
|---|---|---|---|
| ES | Extensor sensor-neuron | - | - |
| FS | Flexor sensor-neuron | - | - |
| EM | Extensor motor-neuron | - | - |
| FM | Flexor motor-neuron | | |
| AL | Flexion receptor for anterior extreme angle of the left hip | - | - |
| AR | Flexion receptor for anterior extreme angle of the right hip | - | - |
| GL | Sensor neuron for ground contact of the left foot | - | - |
| GR | Sensor neuron for ground contact of the right foot | - | - |
| $\rho_X$ | Output signal of the sensor neuron / flexion receptor X | - | $[V]$ |
| $\phi$ | Joint angle measured in the ISB convention | - | o |
| $\Theta_{ES,h}$ | Threshold of the sensor neurons for extension of the hip | 5 | [o] |
| $\Theta_{FS,h}$ | Threshold of the sensor neurons for flexion of the hip | -10 | [o] |
| $\Theta_{ES,k}$ | Threshold of the sensor neurons for extension of the knee | -3 | [o] |
| $\Theta_{FS,k}$ | Threshold of the sensor neurons for flexion of the knee | -80 | [o] |
| y | mean membrane potential of the motor neuron | - | [V] |
| $\Delta t$ | sampling time of the multibody simulation | 1e-3 | [s] |
| $\tau$ | time constant of the motor neuron's cell membrane | 1e-2 | [s] |
| $u_{EM}$ | output of the extensor motor neuron of a joint | - | [V] |
| $u_{FM}$ | output of the flexor motor neuron of a joint | - | [V] |
| $\Theta_M$ | bias constant that controls the motor neurons' firing threshold | 1 | [V] |
| $M_{AMP}$ | magnitude of the servo amplifier | 3 | - |
| $G_{M,h}$ | output gain of the motor neurons in the hip joint | 1.5 | - |
| $G_{M,k}$ | output gain of the motor neurons in the knee joint | 1.35 | - |
| $\alpha$ | positive constant which modulates the response speed of the neurons | 2 | - |

## Controller architecture

The neuron architecture of the implemented controller is shown in Figure 2. THe only difference with respect to Tao, Bernd, and Florentin (2006) is the inversion of the flexor sensor and motor neurons with the extensor ones in the left and right hip blocks. As the provided paper uses the opposite convention for measuring the joint angles at the hip (clockwise rotations correspond to flexion and counterclockwise to extension), this modification needed in order to maintain the desired behaviour of the hip motor neurons when ground contacts of each leg are detected. There is one further adaptation, which will be addressed in the section below.

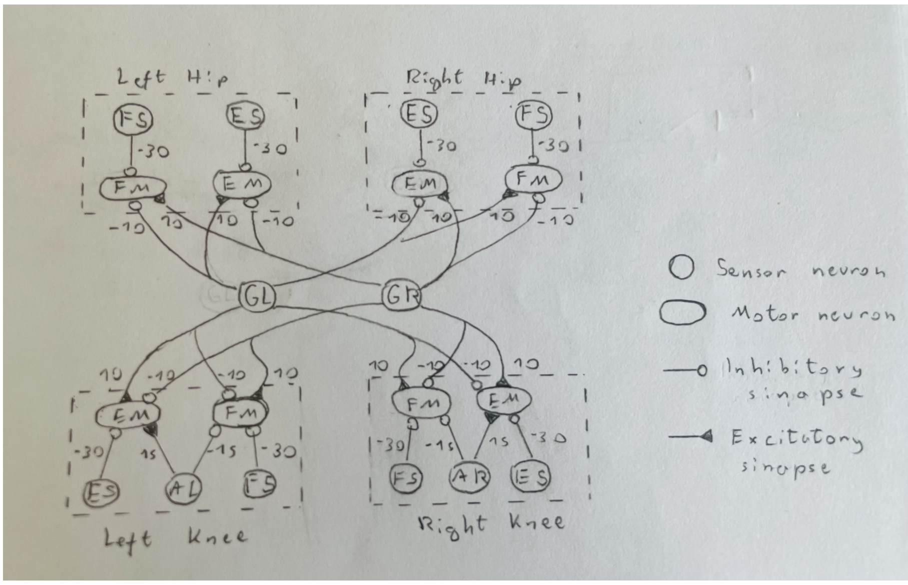


Figure 2: The neuron model of the implemented sensor-driven controller. The small numbers give the values of the connection weights.

# Approach and Implementation

For each one of the four joints, the corresponding motor voltage is given by equation 8 of Tao, Bernd, and Florentin (2006):

$$U = M_{AMP} G_M (s_{EM} u_{EM} + s_{FM} u_{FM}) \tag{1}$$

The output of the extensor/flexor motor neurons are given by

$$u = (1 + e^{\Theta_M - y})^{-1} \tag{2}$$

with membrane potentials

$$\tau \frac{dy}{dt} = -y + \sum \omega_x \rho_x \tag{3}$$

The outputs of the extensor and sensor flexor neurons ($\rho_{ES}, \rho_{FS}$) of each joint are related to the joint angles as described in equations 5 and 6 of Tao, Bernd, and Florentin (2006), respectively. THe output of the ground contact sensors $\rho_{GL}$ and $\rho_{GR}$ are given by equations 3 and 4

The outputs $\rho_{AL}, \rho_{AR}$, however, need to be adapted. As explained before, they detect (in the original convention) the extension of the hips beyond a certain threshold $\Theta_{ES,paper}$. This corresponds to a flexion below the threshold $\Theta_{FS,h}$ in the ISB convention, therefore the neuron outputs must be modified to comply with that

$$\rho_{AL} = (1 + e^{\alpha(\phi_{lh} - \Theta_{FS,h})})^{-1} \tag{4}$$

$$\rho_{AR} = (1 + e^{\alpha(\phi_{rh} - \Theta_{FS,h})})^{-1} \quad (5)$$

Equation 3 can be analyzed assuming that the membrane potential is represented by a capacitor and resistor in parallel. Then $\sum \rho_x \omega_x$ can be interpreted as a source voltage for the RC circuit, $u_s$. In the discrete time setting, we assume that the switch closes at each discrete time instant $t_k$ and remains closed (with $u_{s,k}$ constant) until $t_{k+1}$, and the process repeats itself. Solving the ODE for every time instant yields (with $\Delta t = t_{k+1} - t_k$):

$$y_{k+1} = y_k e^{-\Delta t/tau} + (\sum \omega_x \rho_x)_k$$

which can then be inserted in equation 2 and the result in equation 1 to compute the torque voltages.

### Implementation note

The implementation of the as described above uses persistent variables inside the controller block $rb\_control$. This generates an error in matlab: **joint voltage calculation uses constructs that are invalid when the block specifies or inherits a continuous sample time. Invalid constructs include the use of persistent or global variables, calls to exported functions or using coder.extrinsic.** The suggested action by Matlab must be taken: **Open the properties dialog of this block to set the sample time manually.**. This means that the update method of the function block must be set to **discrete** and the sample time to $\Delta t$.

## Results

Firslty, the joint angles and the ground contact signals of the sample dataset were fed into the controller and the voltages computed for each time instant, in order to check if the controller was correctly implemented. The results can be seen in figure 3.

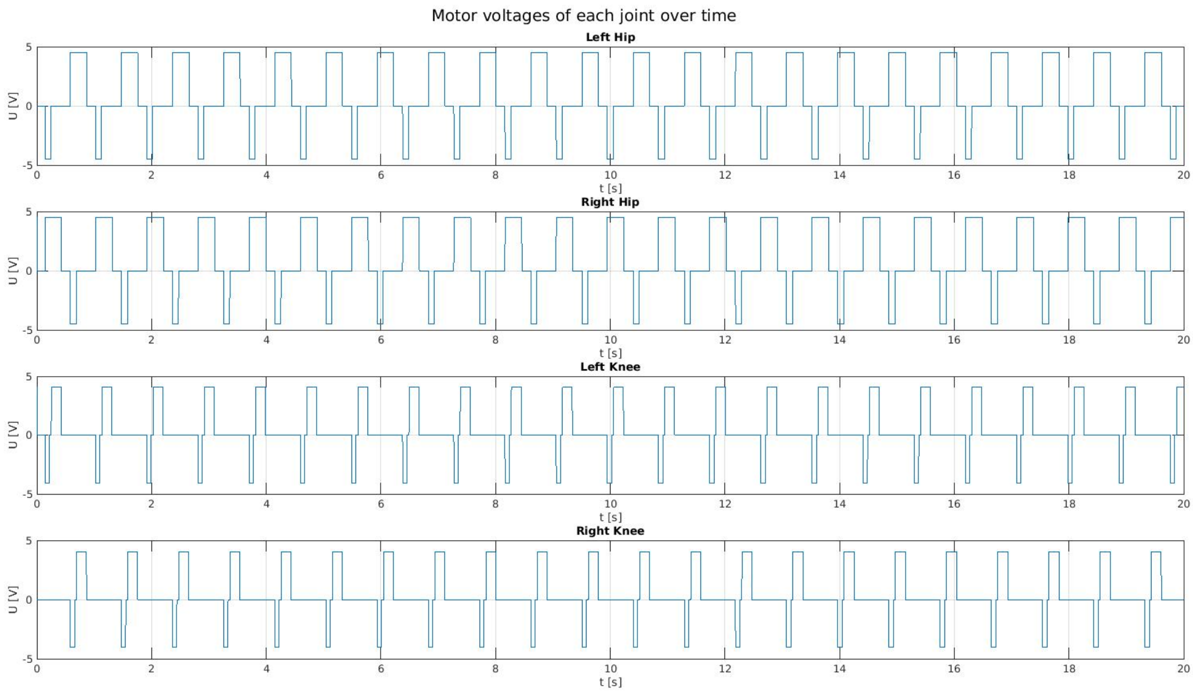


Figure 3: Controller outputs over time for the sample dataset

The model was then tested on the multibody simulation model in Simulink, and a sequence of frames were captured to illustrate its continuous motion:

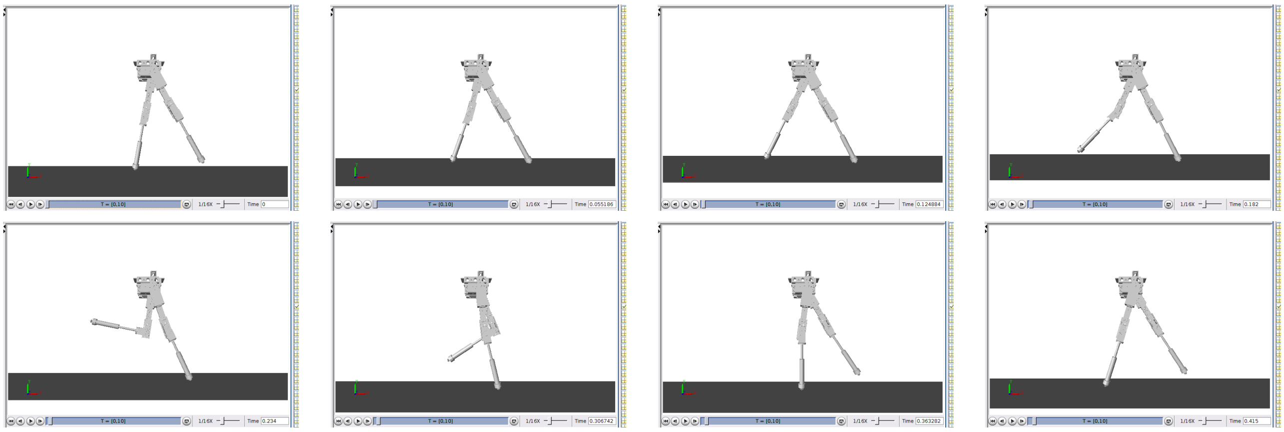

Figure 4: Sequence of captured frames of the walking gait of a multibody simulated robot.

## Discussion

It can be seen in Figure 3 that the implemented controller produces correct outputs when tested with the sample dataset.

Additionally, the walking gait is stable and continuous throughout the entire simulation for the parameters given in Table 1, and corresponds to repetitions of a gait period represented in Figure 4.

### Robot forward velocity

The walking speed of the robot is mainly influenced by five parameters: the gain of the motor neurons $G_{M,h}$ and the thresholds of the flexor and extensor sensor neurons of the hips and knees ($\Theta_{FS,h}$, $\Theta_{ES,h}$, $\Theta_{FS,k}$, $\Theta_{ES,k}$).

As for the first parameter, increasing the gain leads to higher walking speeds; and the values need to be in the range (1.5, 3.5) for the walking gait to be stable. If $G_{M,h} < 1.5$, the robot does not have enough torque to move forward and just stops after one step, while it jumps and falls to the ground due to very high joint torques for $G_{M,h} > 3.5$.

Regarding the second parameter, higher values of $\Theta_{FS,h}$ (smaller in magnitude) lead to slower walking as the leg is allowed to stretch less. Decreasing the value of this threshold (making it more negative) leads to an increase in walking speed since each step covers a higher distance. The gait becomes unstable and the robot falls for $\Theta_{FS,h} < -20°$, and it stops walking for $\Theta_{FS,h} > 0°$.

Regarding the third parameter, higher values of $\Theta_{ES,h}$ lead to higher amplitudes in the steps, which initially results in faster walking. However, at a certain point, the robot's legs become widely separated, which slows down the movement.

Changes to the fourth parameter seem to have less effect on the forward velocity. However, it seems that increasing $\Theta_{FS,k}$ slightly increases the speed.

Finally, an increase in the fifth parameter $\Theta_{ES,k}$ leads to slower walking, which is expected since it allows the legs to push the robot backwards when there is ground contact for values higher than 0, for example (which is also inconsistent with the walking gait of a human). Significantly decreasing the threshold causes the robot to stumble and diminishes the walking speed.

## References

Tao, G., Bernd, P., and Florentin, W. (2006). "Fast Biped Walking with a Sensor-driven Neuronal Controller and Real-time Online Learning". In: *The International Journal of Robotics Research*. DOI: 10.1177/0278364906063822.

---

## AI usage declaration