Task 2: Visuo motor coordination

In this task the robot Bioloid learns to control its arm and points at a red ball that moves in front of the robot. Self organizing map (SOM) is used to create a link between a visual information from a camera and an angle coordination of the robot´s arm.
As in the previous task the robot is connected to Matlab running on a computer and all the computation is made by a prepared script.
Because of very simple camera that the robot uses as an eyesight, three significant simplifications were established to make the task possible. Movement of the ball is constrained to only one dimension (left-right), whether the arm points at the ball is decided by an IR sensor and an image from the camera is preprocessed by a non-neural algorithm. The preprocessing algorithm computes a position of the ball in the picture as a distance from the left edge of the picture.
The following text is a description of the script and a guide to the task.

1. If the robot is already connected to Matlab skip this step. Connect the robot to a computer and find out which COM port number is assigned to the connection. Fill the number to the variable COM_port_number at the beginning of the script and execute the first cell.

2. Set the robot to the right position:
    a. Turn off the robot to disable torque and make the robot easy to handle.
    b. Set the robot to a pose shown in figure [1].
    c. Plug the planar manipulator to the connector on the robot‘s left arm.
    d. Turn on the robot and execute the second cell. The manipulator and the robot will move to the default position.
    e. Align the robot‘s arm and the manipulators effector as shown in figure [2].
    f. Close all figures.

3. Set all the parameters and execute the third cell.
   Parameters:
    a. *numberOfTrials*: number of trials for learning the map
    b. *somSize*: two dimensional size of the map
    c. *somShape*: a shape of the map (sheet, cylinder or toroid)
    d. *somInitValues*: initial values of neurons weights
    e. *somAlpha*: a learning rate
    f. *somRadius*: a radius of effect of a winning neuron
    g. *thresholdTest*: a threshold for IR senzor while it‘s testing whether the arm points at the ball (testing an angle from the map)
    h. *thresholdLearn*: a threshold for IR senzor while it‘s testing whether the arm points at the ball (learning new angle)

   Simplified description of the learning algorithm:
   For number of training steps do:
    a. move the ball to a random position (there are 5 discrete possible positions in range ±6,5 cm)

   b. acquire an image from the camera
   c. compute a position of the ball in the picture
   d. present the position to the map and find a winning neuron
   e. acquire an angle from the winning neuron
   f. use the angle to rotate the arm
   g. decide whether the arm points at the ball
      if it does not, randomly move the arm and try to locate the ball
      if it does, go to next step
   h. from the current arm‘s angle and the computed position create a feature vector
   i. use the feature vector to update the map

4. Execute the forth cell to show learned map, labeled neurons and U-matrix.

5. Set the number of testing trials and execute the fifth cell.

   The testing algorithm is similar to the learning one:
   For number of training steps do:
   a. move the ball to a random position (there is an infinite number of possible possition in range ±6,5 cm)
   b. acquire an image from the camera
   c. compute a position of the ball in the picture
   d. present the position to the map and find a winning neuron
   e. acquire an angle from the winning neuron
   f. use the angle to rotate the arm
   g. decide whether the arm points at the ball
      if it does not, increment an error counter

6. To terminate the connection between the robot and Matlab execute the sixth cell.

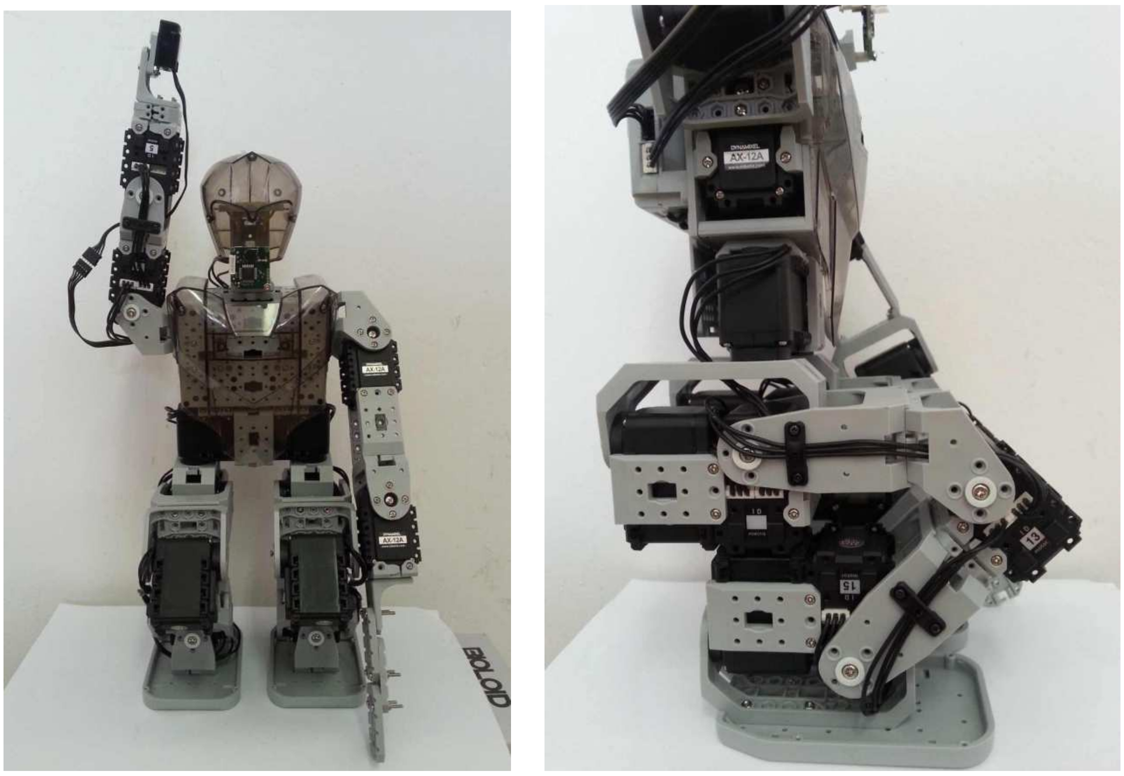
Fig. 1 – Robots pose.

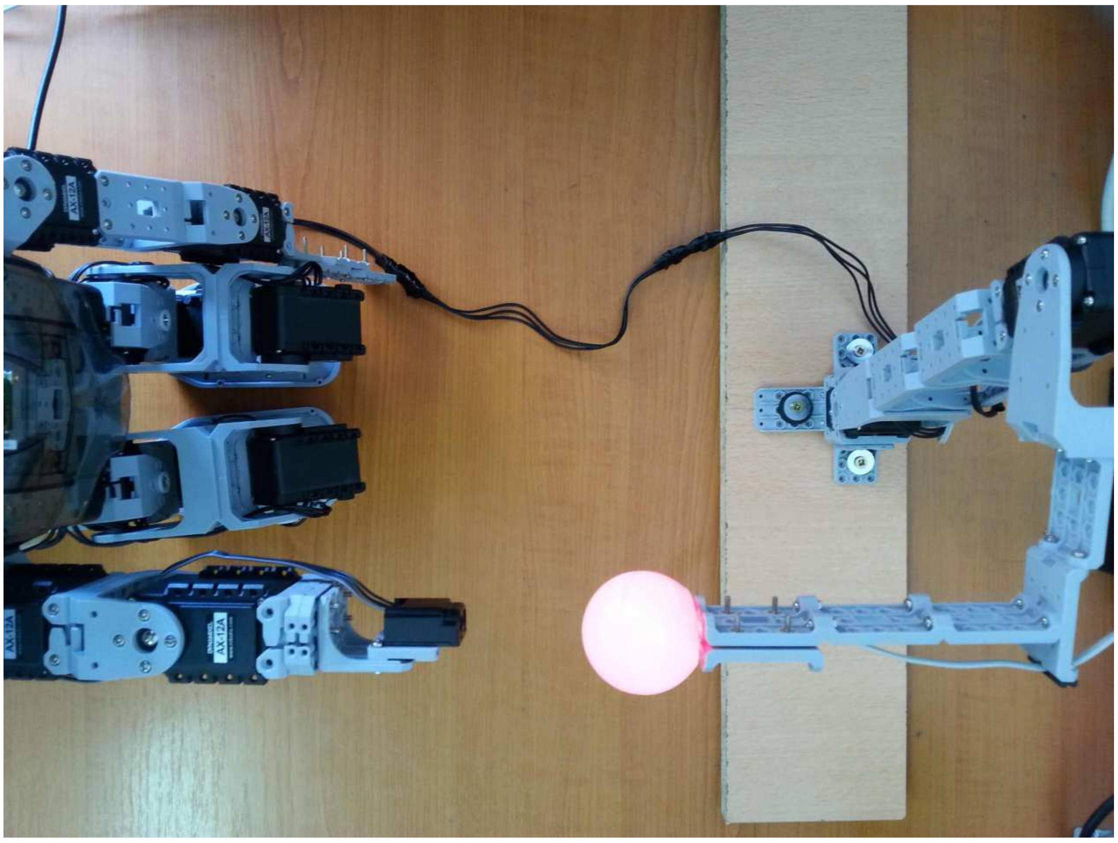
Fig. 2 – Alignment of the robot‘s arm and the manipulator.