HUMAN ROBOT INTERACTION

TUTORIAL 1: a first HRI prototype

Author: Prof. Pablo Lanillos | Department of Artificial Intelligence | Radboud University

Goal: The goal of this tutorial is to get familiar with the environment and program a first example of HRI with the humanoid robot Nao.

1. Webots environment and Robot Nao

Download and install webots from https://www.cyberbotics.com/
Download and extract the template code to your PC
Run the system and test it.

Note: A function to capture the camera from the webcam and show it in the webots display interface is provided.

2. Python controller

Port the nao_demo.c to your python controller. Provide the functionalities of generating predefined movements and printing information in the console. An example of the *gps* sensor (location) is given. The main function will be called run_keyboard. Minimum requirements:

- Implement a functionality in the *run_keyboard* function that reads the keyboard and when you press the left and the right arrow the head moves to the LEFT and RIGHT respectively with velocity 1.0 rad/second. The motors should be initialized in velocity control instead of position control.
- Add a functionality that when you press UP and DOWN arrow the robot walks forward and backward respectively.
- Add a functionality so when you press S (key) the head stops and the walking stops.

3. Face following

Develop a controller (with a new *run_face_follower* main function) that moves the head towards a detected face in the camera. As the robot cameras are not capturing the image, we cannot design a closed-loop controller. Thus, we will use the centre of the face as the reference to move the head. Use the two head angles: *Yaw* and *Pitch* to move left-right and up-down when the face moves in the image. For that purpose code a function with name *look_at* with input the 2D (pixel) location of the face centre.

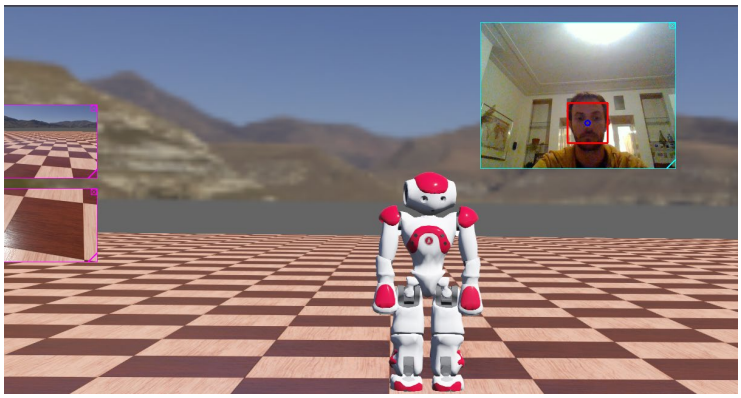

Figure 1: Face following scenario and the webcam face detection.

To detect the face use the opencv built-in function *detectMultiScale*. This algorithm uses the Haar descriptors that are stored in the file provided: *haarcascade_frontalface_default.xml*. These descriptors should be loaded in the initialization function of the controller (*__init__*)

4. Ball

Add to the environment a football ball (with radius 0.05) using the webots interface (click on the add button in the left panel) and develop a head controller that follows the ball (with a new *run_ball_follower* main function). **Use the bottom camera of the robot**. Here you can design a closed-loop controller to maintain the head centred in the ball. Run the program and move the ball by hand to test the head movement. *You need to check the head limits. In the real world sending out of range joint angles could be fatal for the robot.*

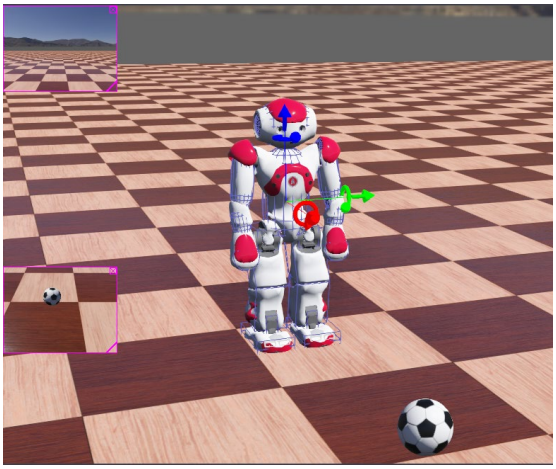

Figure 2: Football ball following scenario.

5. High-level behaviour to improve communication

Develop a set of behaviours to make the robot react when a face and the ball appears in the image. Write it in the new *run_hri* main function. *Tip: you can use predefined movements.*

6. Report, documentation and submission

Write a report of maximum two pages explaining the designing decisions for the controller and the high-level behaviour. Submit the pdf and the code in a compressed file with the full names of the team members. Only one person of the team has to submit it in Brightspace.