Artificial Intelligence of an hexapod robot.

Joffrey KRIEGEL

August 5, 2010

Abstract

This document is an abstract of the "Artificial Intelligence" I use on my hexapod robot.

Contents

1 Description 2

2 Conclusion 4

Bibliography 5

Chapter 1

Description

First of all, figure 1.1 shows you the current hardware of my hexapod. It is a simple architecture centered on the Gumstix overo [2] board. The communication with the servo motors is done by the SSC-32 [3] board. I can send commands to the robot with the Wifi (via SSH) and Bluetooth connexion. The speaker is used with vocal synthesis [4] to communicate the status of the robot and to say what it is doing and what it sees. The camera is only use to do some streaming to a laptop for the moment.

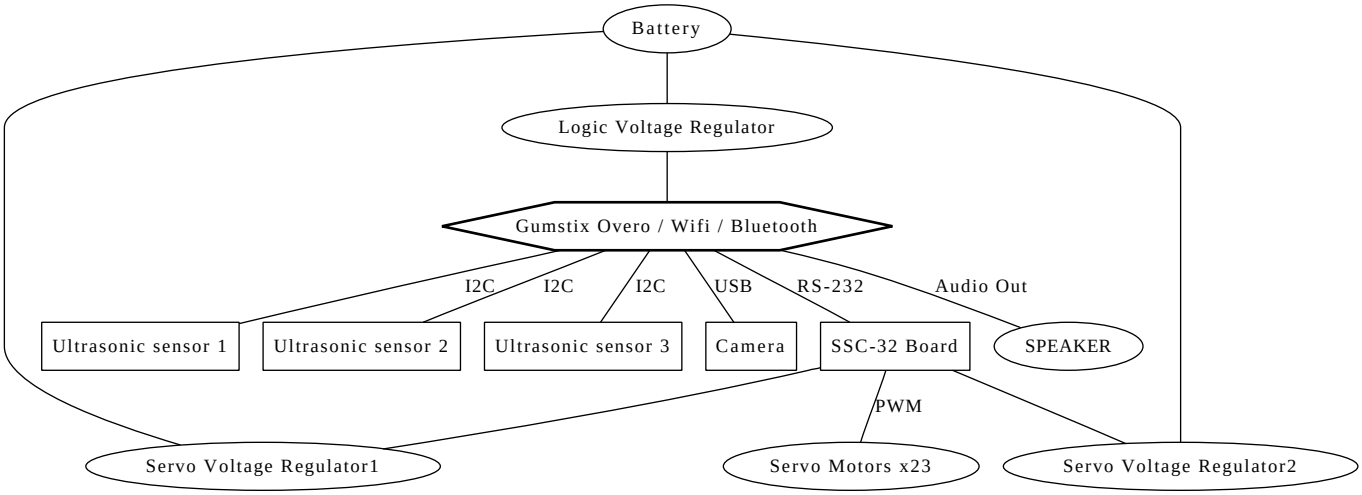

Figure 1.1: Current hardware implemented in the hexapod.

The software architecture is based on 4 threads named in the figure 1.2. The battery check thread use the ADC to check every one minute if the battery is OK. The ultrasound thread asks the status of one sensor every 100ms, so as I have 3 sensors, the total period of the thread is 300ms. And finally, the getInput thread look every 8ms on the bluetooth port if some messages are coming.

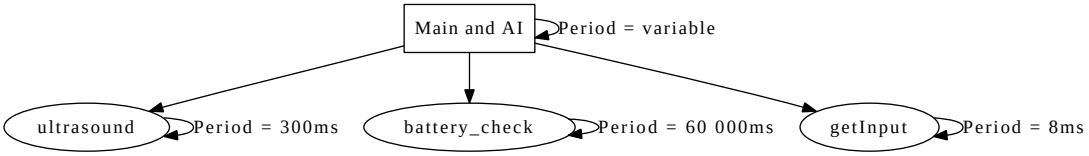

Figure 1.2: Differents threads implemented in the hexapod.

The first version of my hexapod's AI is quite simple. It's based on 3 ultrasonic range sensors which have ben put on its head. On sensor looks on the front, the second looks at 45 degres on the left and

the last looks at 45 degres on the right.

The AI processes every one second which is quite enought for the moment. The algorithm can be decomposed in 6 states described in the figure 1.3. At the beginning of each state, a function that looks the status of the 3 sensors is executed. Each possibility gives a different answer of the robot. The meaning of each possibility can be found in the table 1.1.

In the table, when you see a '0', it means that there is no object in front of this sensor. When there is a '1', the sensor detects something.

Front sensor	Left sensor	Right sensor	Meaning
0	0	0	No object present
0	0	1	Object on the right
0	1	0	Object on the left
0	1	1	Impossible to go in front
1	0	0	Object in front
1	0	1	Object in front and right
1	1	0	Object in front and left
1	1	1	Impossible to go in front (Wall or huge object)

Table 1.1: Different states of the 3 ultrasonic range sensors. — 0 = No-Object detected — 1 = Object detected

With this tab, we can found 6 different possible states. So, every second, the AI looks the sensors and decide in which state it has to go (or stay in the same state). The default direction choosen when avoiding an object is the left if it has the choice between right and left (eg. when sensors = 1-0-0).

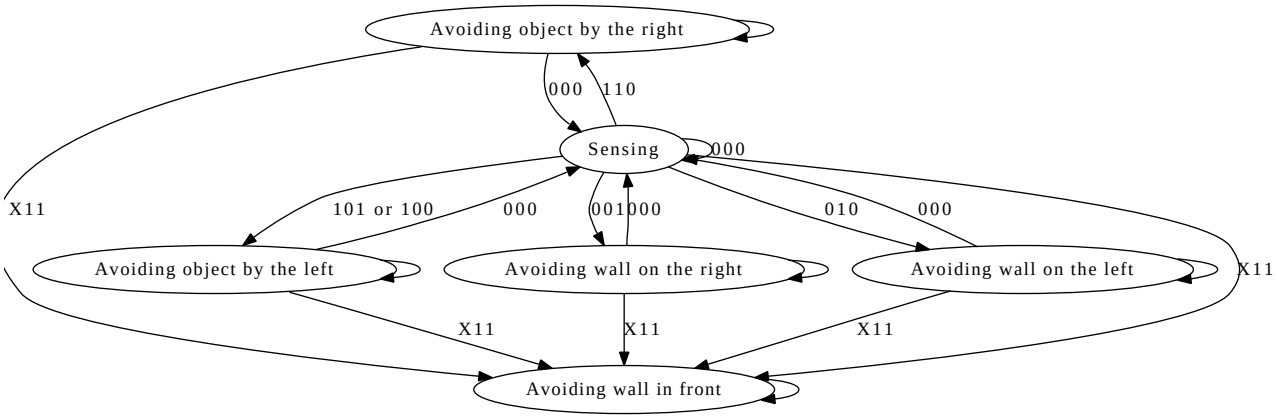

Figure 1.3: Different possible states

The main difference between an object and a wall avoidance is the way to avoid it. When detecting an object, the robot will straff on a side and continue in the same direction. But when it detects a wall, the robot will turn on itself to go in another direction.

Chapter 2

Conclusion

With this basic AI, the robot can now avoid objects wall in front of it. An improvment will be to have more distance sensors to check the sides and the back of the robot. Next I will try to add some vision algorithms using [5] and [6] .

Bibliography

[1] My Website : http://kriegel.joffrey.free.fr/.

[2] Gumstix : http://www.gumstix.com/

[3] SSC : http://www.lynxmotion.com/p-395-ssc-32-servo-controller.aspx

[4] espeak : http://espeak.sourceforge.net/

[5] OpenCV, Open Source Computer Vision : http://opencv.willowgarage.com/wiki/.

[6] Harpia, a visual way of programming image processing algorithms : http://sourceforge.net/projects/s2ilib/.