Overview of the Brain for the robot

October 1, 2015

Contents

1 Robot Brain **1**

2 Learning Algorithms **1**
2.1 RLPowerAlgorithm 1
2.1.1 Example 2
2.1.2 Ideas 2

3 Controllers **2**
3.1 RLPowerController 2

4 HAL **2**
4.1 Outputs 2
4.1.1 Servos 2
4.2 Inputs 3
4.2.1 PositionAsker 3

1 Robot Brain

The robot brain is the main class managing all the components of the robot. The main cycle of the robot is inside this class.

2 Learning Algorithms

There is a general interface that is for every learning algorithm. It consists of `next_evaluation()` function and is returning a new controller that needs to be swaped into *HAL*

2.1 RLPowerAlgorithm

The *RLPowerAlgorithm* is an implmenetation of the Learning Algorithm. It stands for ***R**einfoced **L**earning algorithm, **Po**licy learning by **W**eighting **E**xploration with the **R**eturn Algorithm.*

2.1.1 Example

In the example we have a ranking list composed by:

Parameters	P_1	P_2	P_3	P_4	P_5
Fitness	10	7	6	5	2

Total	30

If for P_c we intend the current parameters and for P_{c+1} we intend the next evaluation. To find the next evaluation, this is the formula:

$$\begin{aligned} P_{c+1} = P_c + \mathcal{N}\left(0, \sqrt{variance}\right) + \frac{(P_1 - P_c) \cdot 10}{30} + \\ + \frac{(P_2 - P_c) \cdot 7}{30} + \frac{(P_3 - P_c) \cdot 6}{30} + \frac{(P_4 - P_c) \cdot 5}{30} + \frac{(P_5 - P_c) \cdot 2}{30} \end{aligned} \tag{1}$$

2.1.2 Ideas

Here are some of the ideas for future improvements:

- It would be nice to have a decade rate for the elements in the rankings, so that the old good walking patterns at some point they get replaced by new ones. This is good in the event of an environment change (e.g. hitting the wall) in which case all the previous walking patterns are probably going to be useless.
- A good idea is to have an automatic way of suddenly increase the mutation rate in case of sudden environment change (e.g. hitting the wall). This should help escape bad situations.

3 Controllers

A controller object is interpreting some inputs and is giving some outputs. The object should be generated from the learning algorithm.

3.1 RLPowerController

TODO

4 HAL

HAL stands for Hardware Abstraction Layer. Is the module responsible for giving easy access to both inputs and outputs.

4.1 Outputs

4.1.1 Servos

This is taking the control input for the servo and sending it to the hardware.

4.2 Inputs

4.2.1 PositionAsker

PositionAsker is a class that asks for the position from a remote server which could be implemented with QRCode tracking or a more sophisticated system.