

dr12 robot simulation with VREP and python

Aly Shmahell
Bachelor of Computer Science
University of L'Aquila

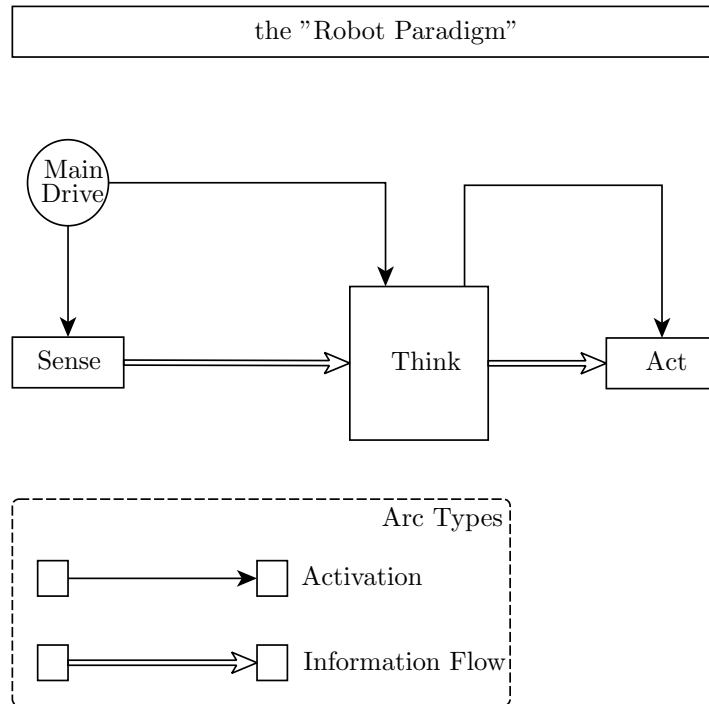
November 15, 2016

Abstract

This project describes the mental model of a dr12 robot designed with Python and Vrep API and simulated with the Vrep environment.

1 Introduction

The Modeling of dr12 behavior here is done using the Robot Paradigm, which is based on the consecutive actions: "Sense - Think - Act".



The main drive of the robot is to "sense and think" consecutively, this will dynamically reshape the "think" output as often as new data comes in from sensory devices, then comes the "act" process which is triggered by the "think" process.

2 Rundown of the implementation

Files and Directories:

VrepPythonAPI: houses most of the API files made by Coppelia Robotics GmbH.
Documentation: houses the tex file, pdf files, svg diagrams and the video Screen-cast (OGV format).

VrepAPIWrapper.py: my API Wrapper, contains the "Sense" and "Act" functions.

Brain.py: my Brain class, contains the "Think" function.

VrepAPIWrapper.py:

this file starts by importing the API as follows:

```
sys.path.insert(0, './VrepPythonAPI')
```

this is done in order to import the relative API files from "VrepPythonAPI".

then the following functions are implemented:

-) initFunction:

which closes all previous connections to vrep using "vrep.simxFinish(-1)" and starts a new connection with "vrep.simxStart" and assigns a clientID to the connection, and on failure it exits.

-) senseFunction:

connects to the proximity sensor with "simxGetObjectHandle" then reads the relative sensory info with "simxReadProximitySensor".

-) actuateFunction:

connects to the two motors with "simxGetObjectHandle" then sends speed information with "simxSetJointTargetVelocity".

-) finishFunction:

if there is a connection, it turns off the dr12 motors then terminates the connection.

Brain.py:

This class starts off by importing all functions from the wrapper that I wrote, then it calls the `initFunction`.

It also has the implementation of the `thinkFunction` as follows:

this function assigns a time variable that runs parallel in value to realtime, it reads sensory data using `senseFunction` and actuates the motors depending on the following scheme:

if there is a point detected nearby, it reads more sensory data and then offsets the time variable.

During this time offset, depending on the difference in coordinates value, we determine where the object is located relative to the robot.

If the object is close to the right side of the robot, the actuation turns the left motors off to steer the robot left, and if the object is on the left side, the right motors are turned off and the robot is steered right.

the main function: it calls the "Think" function, and if the Keyboard is interrupted (Ctrl+c) it calls the `finishFunction`.

3 Links:

The project is uploaded to my Github account under this link:

<https://github.com/AlyShmahell/AlyShmahell-UnivAq/tree/master/ComputerScience/IntelligentSystemsAndRobotics/Robotics>

also, the Screencast is uploaded to my youtube channel under this link:

<https://www.youtube.com/watch?v=q0Jg0ZeHSM4>