Reinforcement Learning with LEGO Mindstorms

Project Report

Per Steffen Czolbe (wjq874)

Handed in: December 10, 2018

1 Abstract

Contents

2 Introduction

scientific goal, educational goals, agenda by lego

2.1 The ev3 robot

Hardware

central component: the "brick" cpu: 32bit, 300MHz ARM9 Processor Memory: 64MB DDR RAM Storage: 16MB Flash up to 32gb SD-card

Communication: USB Bluetooth wifi

Periferals: up to 4 motors and 4 sensors.

src: LEGO MINDSTORMS EV3 Hardware Developer Kit (pdf) ©2013 The LEGO Group

Default Software

The ev3 is shipped with LEGOs own graphical programming software, based on the visual programming language LabVIEW.

Many community driven software packages exist. These support languages such as matlab, python, java, javascript, C++, C, GO, Ruby, Perl, R, Lue and may more (src: https://www.ev3dev.org/docs/programming-languages/). Direct control of the robot via command line is also possible.

3 Development Setup

- tutorial style description of how to get things to run with jupyter notebook.

3.1 New software: ev3dev

"ev3dev is a Debian Linux-based operating system that runs on several LEGO® MINDSTORMS compatible platforms including the LEGO® MINDSTORMS EV3 and Raspberry Pi-powered BrickPi." src: https://www.ev3dev.org/

Because of the limited storage available on the Brick, this 2GB linux image has to be added to the ev3 via an SD-Card.

3.2 External Computer for computations

While the brick runs a fully fletched linux system that can execute programms on it's own, the processing and memory contraints of the brick limit development of programms severely. Installing a new python package with pip often takes over 2 minutes. Installing deep learning frameworks like tensorflow, or performing image processing on the brick is expected to lead to severe sortages of processing power. Thus we want to run these more computationally expensive parts of the program on a seperate computer.

The setup of a main programm running on a computer and only interacting with the brick when nessesary to move motors or read sensor data has been employed by multiple community projects, such as the LEGO rubics cube solver. (src: https://www.ev3dev.org/projects/2014/05/09/Python-Rubiks-Cube-Solver/) We follow the same design approach implemented in these projects. The brick runs a remote procedure call server, implemented by the python3 libary 'rpyc'. This allows other computers to connect to the brick as clients and remotely interact with the python libary controlling the robot functions. All program logic is implemented on the computer, the brick is merely listening to commands. Next to the performance gains, another benefit is that the programm can be executed in a jupyter notebook on the computer, which allows better integration of documentation and visualizations within the programm.

4 Sensors and Motors

experiments, description of how to interact with modules

4.1 Time Delays

4.2 Motor Accuracy

4.3 Ultrasound Sensor Accuracy

4.4 Gyroscope accuracy

4.5 Need for calibration

- most sensors no calibration - motors and gyroscope have their 0-rotation initialized at system start. This makes reclibration nessesary every time the system is restarted. The robots should be designed with this in mind. For motors, automatic calibration can be achieved by having them hit against a push sensor first

4.6 General building techniques

- as simple and robust as possible -

5 Reinforcement Learning with LEGO Mindstorms

general issues arising when working with actual robots vs simulation

6 Colour Detection

6.1 Problem

what task solved

6.2 The Robot

description of the robot build

6.3 The Learning

implementation of learning, choices made in picking and implementing algorithm

6.4 Results

final results

6.5 Discussion

includes issues / future work

7 Crawl-Robot

7.1 Problem

what task solved

7.2 The Robot

description of the robot build

7.3 The Learning

implementation of learning, choices made in picking and implementing algorithm

7.4 Results

final results

7.5 Discussion

includes issues / future work

8 Swing-Robot

8.1 Problem

what task solved

8.2 The Robot

description of the robot build

8.3 The Learning

implementation of learning, choices made in picking and implementing algorithm

8.4 Results

final results

8.5 Discussion

includes issues / future work

9 Future Work

future robots/ multi actor learning