CourseWare Wiki / old / courses / b4m36uir / labs / resources B4M36UIR & BE4M36UIR ■B4M36UIR and BE4M36UIR - Artificial intelligence in robotics Classification Lectures **□ □** Laboratory Exercises Lab01 - Introduction to V-REP and Robot Locomotion Control Lab02 - Reactive-based Autonomous Robot Behavior Lab03 - Reactive-based Autonomous Robot Behavior contd. Lab04 - Grid and Graph-based Path Planning Methods Lab05 - Incremental Path Planning Lab06 - Randomized Sampling-based Motion Planning Lab07 - Randomized Sampling-based Motion Planning contd. Lab08 - Randomized Sampling-based Motion Planning contd. Lab09 - Data Collection Planning (TSP(N)) Lab10 - Data Collection Planning ((D)TSPN) Lab11 - Game Theory in Robotics Lab12 - Game Theory in Robotics contd. Lab13 - Game Theory in Robotics contd. Resources

Warning

Homeworks
Tutorials

This page is located in archive.

-Table of Contents

- Resources
 - o Hexapod Hardware Abstraction Layer (HAL)
 - V-REP hexapod robot model
 - V-REP scenes
 - Placing obstacles in the scene
 - o Importing a new maze into the V-REP
 - V-REP demos

Resources

During the labs we will be using V-REP simulator with a Python remote <u>API</u> and a real hexapod walking robot. This page provides resources which are common for all the labs.

Hexapod Hardware Abstraction Layer (HAL)

hexapod_vrep_v2.0

Update: Updated version of hexapod_vrep for dealing with noisy position, orientation and laser scanner data hexapod_vrep_v3.0

hexapod_real_v1.0

The HAL allows to use standardized function calls for interfacing both the simulated and the real robot regardless the particular implementation of the hardware layer. The particular interface is selected by choosing one of two possible includes in the main python script. For interfacing the simulated robot in the V-REP the main python should include in the header:

import hexapod_vrep.robot as rob

For interfacing the real robot the main python should include in the header:

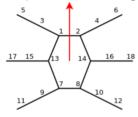
import hexapod_real.robot as rob

V-REP hexapod robot model

Hexapod model



Hexapod servos numbering:



V-REP scenes

scenes_pack_v1.0

Scenes are the simulation main playground. Obstacles, robots and other objects can be placed into the scene and controlled by the simulation. Besides, terrain maps can be imported from image files directly to the scene.

- Plain scene
 - o The very basic scene for testing locomotion principles.
- Simple obstacle
 - o Scene with only one obstacle to test the locomotion and basic obstacle detection and obstacle avoidance.
- Obstacle avoidance
 - o More advanced scene for testing reactive obstacle avoidance.
- Maze 1-3
 - o Scenes for testing of motion planning approaches with obstacles imported from the image file.

Placing obstacles in the scene

Obstacles in the scene can be made by adding a primitive shape into the scene by right clicking into the scene and selecting add » Primitive shape. Then we set up the dimensions of the shape by filling in the dialog.

To make the obstacle static (i.e. not influenced by gravity or collisions) but still exerting some collision responses on non-static respondable shapes it is necessary to disable Body is dynamic option in the shape dynamics properties accessed from the Hierarchy manager and Show dynamic

properties dialog as it is visualized in the following picture.



It is also necessary to make the obstacle Collidable, Measurable, Renderable and Detectable in the Common properties of the same dialog.

Importing a new maze into the V-REP

The maze creation is based on importing the height map into the V-REP simulator. As a source, gray scale .jpg or .png image is used. The maze can be imported into V-REP as follows:

- 1. Prepare the image of the maze preferably in a low resolution (approx. 200×200)
- 2. Go to File>Import>Heightfield and locate the image
- 3. Fill in the imported Heightfield dimensions where Z-scaling is the height difference between the black and white colors
- 4. Move the Heightfield in z-direction so the bottom of it lies under the ResizableFloor
- 5. Resize the ResizableFloor so it allows the robot to move freely through the whole maze
- 6. Set up the properties of the Heightfield according to the above described approach. Hence, make the object Collidable, Measurable, Renderable and Detectable.

V-REP demos

demo codes v1.0

V-REP demos intends to show the basics of controlling the hexapod model in the V-REP simulator.

• Joint position setting demo

• This demo shows how to precisely control the joint angles of the robot.

Simple walking demo

- o This demo shows a basic locomotion control of a hexapod robot using the Central Pattern Generator (CPG) neural network.
- o By adjusting variables 1 and r in the main function, the robot performs locomotion with a different turning radius and speed

• Distance sensor readout demo

• This demo shows how to interface the simulated distance sensor. Note, in the simulation the sensor is simulated as a ray type laser scanner whereas on the real robot an Asus Xtion Pro RGB-D camera is used to obtain the distance measurements.

Do not forget to symlink or include hexapod_vrep directory with appropriate version of the remoteApi library for your operating system to the demo directories

courses/b4m36uir/labs/resources.txt · Last modified: 2017/10/23 09:39 by cizekpe6