

Control of Mobile Robots: Sim.I.am

Last updated: 07/11/2014

Author Information

Jean-Pierre de la Croix

Georgia Institute of Technology

Professor Magnus Egerstedt

Georgia Institute of Technology

Course Details

Description

A MATLAB based educational bridge between theory and practice in robotics.

Sim.I.am is a robotics simulator that can be used to learn how to apply controls to mobile robots. The app is a demo of a multi-agent application where one robot follows the other robot through a cluttered environment. On-screen instructions are included. A manual is included to help learn how to work with the simulator. It also provides a set of programming exercises focused on how to implement the mobile robot navigation system (found in the app) from scratch.

Original Course Documents

Source file URL

Course Contents

Week 1: Learn About MATLAB and the Robot Simulator

Since the robot simulator is written in MATLAB, you should familiarize yourself with the MATLAB language and environment.

Familiarize yourself with the simulator by reading the manual and running this week's robot simulator.

- Manual

Run Sim.l.am simulation in MATLAB by executing launch.m from within the Week 1 directory.

launch.m

To help you get started with learning MATLAB, here is a selection of short video tutorials that cover concepts you will encounter in the assignments:

What is MATLAB?	The MATLAB Environment	MATLAB Variables	Mathematical Functions	MATLAB as a Calculator
Creating Vectors via Concatenation	Creating Uniformly Spaced Vectors (Colon Operator)	Creating Matrices	Concatenating Arrays	Array Creation Functions
Array Size and Length	Vector Arithmetic	Accessing Elements of a Vector	Accessing Elements of an Array	Matrix Multiplication
Using the MATLAB Editor	For Loop	Logical Operators	If-Else Statement	Calculating Eigenvalues and Eigenvectors
Line Plots	Working with Classes	Structure Arrays		

Week 2: Implement Components of the Simulator

Begin by unzipping the robot simulator for Week 2 below.

Before you can test controllers in the simulator, you will need to implement three components of the simulator, as described in the Week 2 section under Programming Assignments in the Manual.

- DifferentialDrive.m
- QBSupervisor.m (Odometry)
- QuickBot.m (Infrared (IR) Proximity Sensors)

These files are part of the code for the Week 2 version of the simulator. To run the simulation you will have to execute the launch.m file from within the Week 2 directory.

launch.m

Week 3: PID Regulator

This week you will be implementing the different parts of a PID regulator that steers the robot successfully to some goal location. Follow the instructions for the Week 3 Programming Assignment in the Manual to modify the following file.

- GoToGoal.m

This file is part of the code for the Week 3 version of the simulator. To run the simulation you will have to execute the launch.m file from within the Week 3 directory.

launch.m

Week 4: Obstacle Avoidance

This week you will be implementing the different parts of a controller that steers the robot successfully away from obstacles to avoid a collision. Follow the instructions for the Week 4 Programming Assignment in the Manual to modify the following file.

- AvoidObstacles.m

This file is part of the code for the Week 4 version of the simulator. To run the simulation you will have to execute the launch.m file from within the Week 4 directory.

launch.m

Week 5: Controller Arbitration

This week you will be adding small improvements to testing two arbitration mechanisms: blending and hard switches. Arbitration between the two controllers will allow the robot to drive to a goal, while not colliding with any obstacles on the way. Follow the instructions for the Week 5 Programming Assignment in the Manual to modify the following files.

- GoToGoal.m
- AvoidObstacles.m

- OAandGTG.m

These files are part of the code for the Week 5 version of the simulator. To run the simulation you will have to execute the launch.m file from within the Week 5 directory.

launch.m

Week 6: Wall Following

This week you will be implementing a wall following behavior that will aid the robot in navigating around obstacles. Follow the instructions for the Week 6 Programming Assignment in the Manual to modify the following file.

- FollowWall.m

This file is part of the code for the Week 6 version of the simulator. To run the simulation you will have to execute the launch.m file from within the Week 6 directory.

launch.m

Week 7: Full Navigation System

This week you will be combining the go-to-goal, avoid-obstacles, and follow-wall controllers into a full navigation system for the robot. Follow the instructions for the Week 7 Programming Assignment in the Manual to modify the following file.

- QBSupervisor.m

This file is part of the code for the Week 7 version of the simulator. To run the simulation you will have to execute the launch.m file from within the Week 7 directory.

launch.m

Resources

- MATLAB App installation file for the complete, fully functional Sim.I.am robot simulator:
simiam.mlappinstall

Links

- Coursera MOOC: Control of Mobile Robots
- MATLAB Central File Exchange post



This work is licensed under a Creative Commons Attribution-ShareAlike 3.0 Unported License.

Learn more about MathWorks academic resources:

[MATLAB Courseware](#)
[Tutorials](#) [Webinars](#)

[Hardware Resources](#)
[Technical Articles](#)

[Classroom Resources](#)

[MATLAB Examples](#)

[Books](#)