Introduction
ooo
ooo

Algorithms
oo
ooo
oo

Design Patterns
oo

Development Methodology
oooo

References

Obstacle Avoidance and Goal Detection Robot using RPi and LRF.

Atabak Hafeez   Maria Ficiu   Rubin Deliallisi   Siddharth Shukla

Jacobs University Bremen

February 23, 2016

Overview

Goal (Basic)
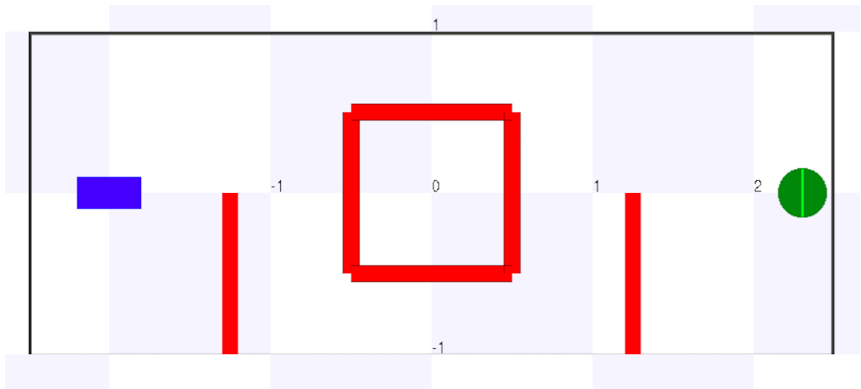
Given a goal and a robot:

- ▶ Turn the robot around
- ▶ Identify the goal object from LRF input
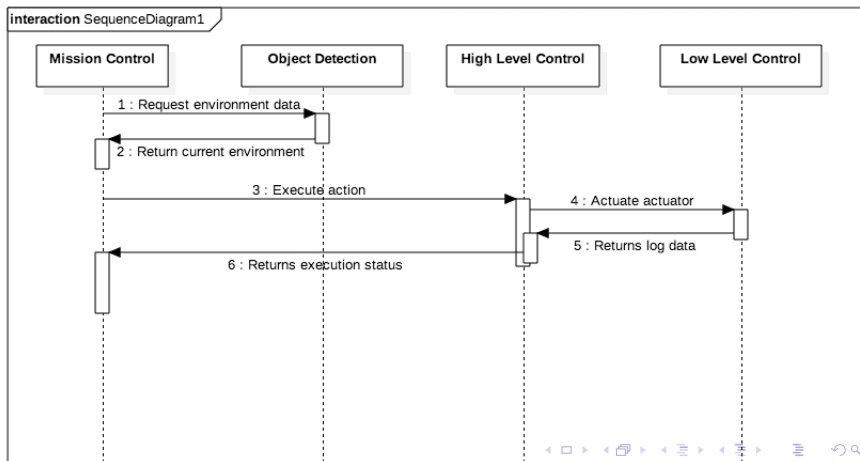- ▶ Move towards the goal

Goal (Advanced)

Given a goal, a robot, and a maze:

- ▶ Turn the robot around
- ▶ Follow the walls and avoid obstacles
- ▶ Identify the goal from LRF input
- ▶ Move towards the goal if nothing blocks

Introduction
○○●
○○○

Algorithms
○○
○○○
○○

Design Patterns
○○

Development Methodology
○○○○

References

Goal

Goal (Advanced)

Introduction	Algorithms	Design Patterns	Development Methodology	References
○○○	○○	○○	○○○○	
●○○	○○○			
	○○			

Architecture

Sequence Diagram

Component Diagram

Class Diagram

Introduction	Algorithms	Design Patterns	Development Methodology	References
○○○	●○	○○	○○○○	
○○○	○○○			
	○○			
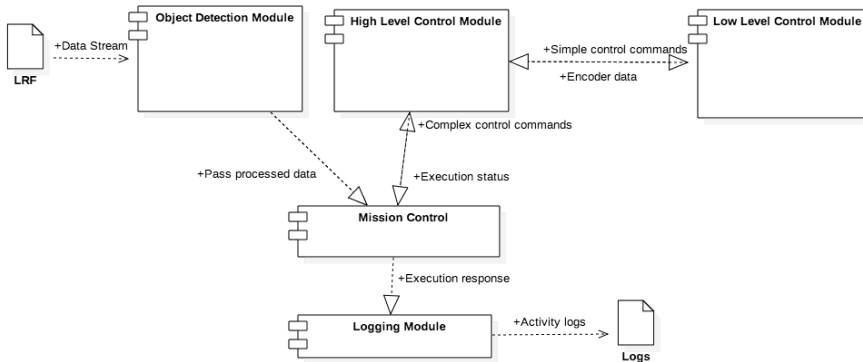
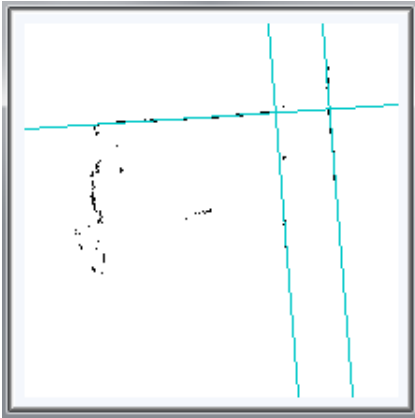Transforms

Hough Transform

- ▶ Algorithm to detect imperfect instances of geometric shapes
- ▶ Reduces the amount of data to be processes
- ▶ Usage
    - ▶ Detect outer walls and obstacles
    - ▶ Detect goal (Semi-Circle)
- ▶ Idea (line detection case)
    - ▶ Calculate the equation of the line through two consecutive points
    - ▶ Coefficients of the equation are added to a counter to record how many times the same equation is calculated
    - ▶ Coefficient calculated often equate to many points being roughly aligned

Hough Transform

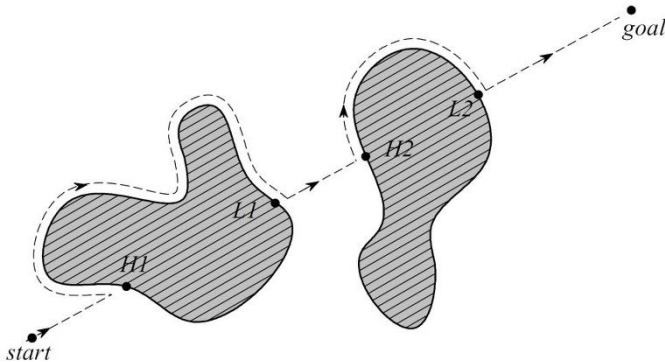| Introduction	Algorithms	Design Patterns	Development Methodology	References
000	00	00	0000	
000	●00			
	00			

Bug Algorithms

Bug algorithms

- ▶ Knowledge of local environment and a global goal
- ▶ Assumptions
    - ▶ Known direction and distance to goal
    - ▶ Obstacle detection and encoder data
    - ▶ Finitely many obstacles in finite area
- ▶ Idea
    - ▶ Head towards goal
    - ▶ Follow obstacles until you can head towards goal again
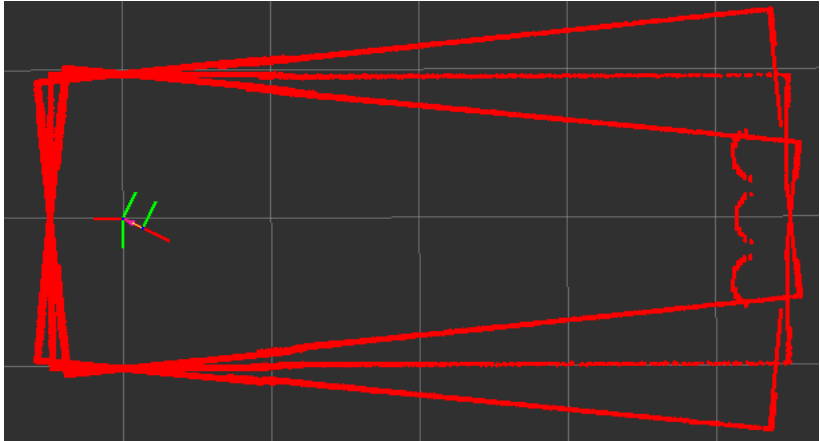   - ▶ Stop if there is no path to goal

Introduction
000
000

Algorithms
00
0●0
00

Design Patterns
00

Development Methodology
0000

References

Bug Algorithms

Bug Algorithms

Introduction	Algorithms	Design Patterns	Development Methodology	References
000	00	00	0000	
000	000●			
	00			

Bug Algorithms

(Modified) Bug algorithms

- ▶ Missing assumptions
    - ▶ Distance to goal is not known
    - ▶ Direction to goal might be imprecise(slippery surface, wrong encoder data)
- ▶ Additional assumptions
    - ▶ Environment has a rectangular shape
- ▶ Greedy approach
    - ▶ Head to the wall you are initially facing
    - ▶ Follow the walls until see the goal
    - ▶ Head to the goal

Introduction
○○○
○○○

Algorithms
○○
●○

Design Patterns
○○

Development Methodology
○○○○

References

Noise

Introduction	**Algorithms**	Design Patterns	Development Methodology	References
ooo	oo	oo	oooo	
ooo	ooo			
	o●			

Filtering and Mapping

Kalman Filtering and Mapping

- ► Kalman Filtering (Linear Quadratic Estimation)
    - ► Input
        - ► A series of measurements of a variable obtained over time
        - ► Observations contain statistical noise
    - ► Output
        - ► Estimates of the values of the variable
        - ► These estimates tend to be more precise than one time measurements
- ► Mapping
    - ► Create a dynamic map of the surrounding environment
    - ► Use this map to reach the goal faster after the first iteration
    - ► Needs Kalman Filtering to get precise position of gaps, walls and goal

Introduction	Algorithms	Design Patterns	Development Methodology	References
000	00	●○	0000	
000	000			
	00			

Patterns in Use

Singleton

- ▶ Comes from the mathematical 'Singleton (a.k.a. Unit set)'
- ▶ Restricts the instantiation of a class to one object
- ▶ Defines a 'getInstance' operation to expose the unique instance
- ▶ Useful when exactly one object needed to coordinate actions
- ▶ getInstance responsible for creating class unique instance

Introduction	Algorithms	Design Patterns	Development Methodology	References
000	00	0●	0000	
000	000			
	00			

Patterns in Use

Mediator

- ▶ Behavioural pattern - Can alter the program's running behaviour
- ▶ Communication between objects encapsulated with a mediator object
- ▶ Objects no longer communicate directly with each other
- ▶ Reduces the dependencies between communicating objects, thereby lowering the coupling

Introduction
000
000

Algorithms
00
000
00

Design Patterns
00

Development Methodology
●000

References

Stay Agile! Stay Alive!

- ▶ Biweekly code sprints
- ▶ Trello for task management and tracking backlog
- ▶ Daily standups to keep track of progress and blockers
- ▶ Weekly review and retrospective
- ▶ Coordinated Pair programming sessions
- ▶ End of sprint celebrations (Motivation!)

| Introduction	Algorithms	Design Patterns	Development Methodology	References
ooo	oo	oo	o●oo	
ooo	ooo			
	oo			

Workflow

Testing

- ▶ Test Driven Development (TDD)
    - ▶ Write tests before code
    - ▶ Helps clearly plan out program functionality
    - ▶ Reduces debug time drastically
- ▶ Focused on four different domains:
    - ▶ Unit Testing
    - ▶ Integration Testing
    - ▶ System Testing
    - ▶ Stress Testing

Introduction	Algorithms	Design Patterns	Development Methodology	References
○○○	○○	○○	○○●○	
○○○	○○○			
	○○			

Workflow

Version Control

- ▶ Git (using Github for remote)
- ▶ Divide tasks into issues
- ▶ Branching Model
    - ▶ Each issue to be a separate branch on the remote
    - ▶ To be merged back in to the master after completion of issue
- ▶ Pre commit hooks for code style guideline compliance

Workflow

Deliverables and Timeline

Gantt Chart for Timeline
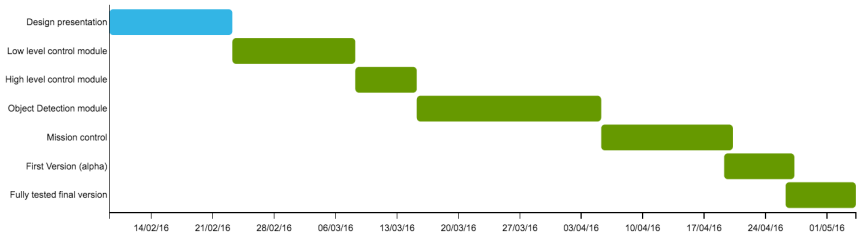
References

📄 Atabak Hafeez, Maria Ficiu, Rubin Deliallisi, Siddharth Shukla (2016)
Obstacle Avoidance and Goal Detection Robot
Jacobs University Bremen.

📄 Michael Misha Kazhdan(2004)
Seminar on Shape Analysis and Retrieval
Department of Computer Science, Johns Hopkins University, Baltimore, MD.

📄 Howie Choset
Robotic Motion Planning: Bug Algorithms
The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA.

📄 Greg Welch and Gary Bishop (2006)
An Introduction to the Kalman Filter
Department of Computer Science, University of North Carolina, Chapel Hill, NC.