Introduction to Robotics

Dr. David F. Barrero & Dra. Maria D. R-Moreno

Objectives

- Introduce main topics on Robotics

Bibliography

- Russell, S., Norvig, P. (2010). *Artificial Intelligence: A modern approach.* (Cap. 25). 3rd Ed. Prentice-Hall.
- ROS documentation (wiki)

Table of Contents

Introduction

Introduction to Robotics (I)

Robot definition

Active**, **artificial agent** whose environment is the **physical world

Russell and Norvig

- Active: It is not a rock
- Artificial: It is not an animal
- Physical world: It is not just software

Introduction

Introduction to Robotics (II)

Robotics also includes

- Robotic swarms (Video)
- Human-robot interfaces (Video), human-machine interfaces (Video), ...

Fields related to Robotics

- Mechanical engineering
- Electronics
- Artificial vision (Video)
- Machine Learning (Video)
- Learning (Video)
- Neural Networks (Video)
- Evolutionary Computation (Video)

Introduction

Introduction to Robotics (III)

Robots may be autonomous or teleoperated

- Greyscale between them
- We are interested in autonomous robots

Real-world is demanding

- **Inacessible**, the world is incompletely perceived by sensors
- **Non-deterministic**, the robot must deal with uncertainty
- **Non-episodic**, the effects of an action change over time
- **Dynamic**, the robot may need to deliberate or to act
- **Continous**, enumerating all the potential actions is inviable

Introduction

Applications

- Manufacturing and materials handling
- Survillance robots
- Hazardous environments
- Telepresence and virtual reality
- Augmentation of human abilities
- Prosthesis
- Space exploration
- Autonomous cars
- Security and defense

Introduction

Robot categories

Three big categories

- **Manipulators**
- **Mobile robots**
- **Humanoids**

This classification is not strict

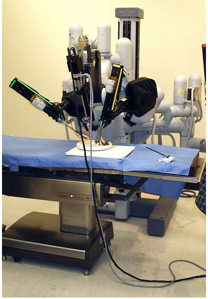
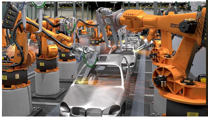
Introduction

Robot categories: Manipulators

Applications

- Manufacturing
- Robotically-assisted surgery
- Space
- Harzadous materials

(Video robotic arm)
(Video production line)


Introduction

Robot categories: Mobile robots

Applications

- Exploration
- Harzadous environments
- Defense and security
- Surveillance
- Telecare

Types: UAVs, UGVs, AUV, quads

(Video UAV)
(Video AUV)
(Video Fukushima)

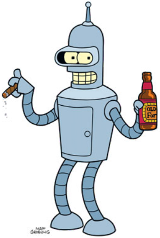
Introduction

Robot categories: Humanoids

Applications

- Research
- Enterteinment
- Telecare

(Video Nao) (Video Atlas) (Video DARPA) (Video DARPA bonus track)

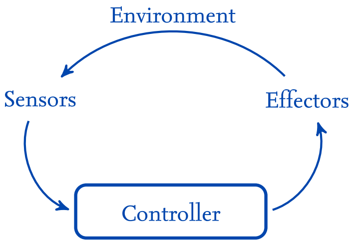




Introduction

Components

Any robot contains three components

- **Effectors** (actuator). Affects the environment
- **Sensors**. Provides knowledge about the environment
- **Controller**. Takes decisions



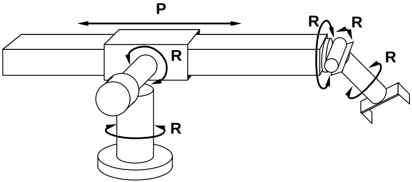
Effectors

Classification

- **Effector**: Device that affects the environment
  - In Robotics, it usually involves a motors or hydraulic devices
- Each motion provides a degree of freedom (DOF)
- Two ways to use effectors
  - Manipulation: Change the position of other objects
  - Locomotion: Change the position of the robot

Effectors

Manipulation (I)



- 6 is the minimum number required to position end-effector arbitrarily
- For dynamical systems, add velocity for each DOF
- **Terminal effector** (or end effector): Tool usually attached to a robotic arm

Effectors

Manipulation (II)

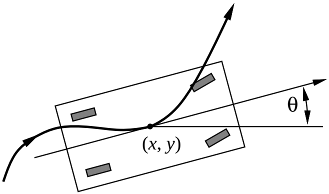
Big issue: Motion planning

- **Kinematics**: Given the pose, get the location
- **Inverse Kinematics**: Given the location, get the pose

(Source)

Effectors

Locomotion (I)



Two types of mobile robots

- **Holonomic**: Same DOF that control
- **Non-holonomic**: More DOF than controls

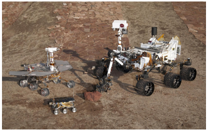
Control hardness depends on relation between DOF and control

- The larger is the gap, the harder is the control

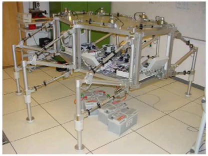
Effectors

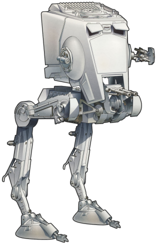
Locomotion (II)

Types of locomotion



Rover





Walker

Climber

Tracks

Sensors

Classification

Sensor: Device that gathers information

By energy emission:

- Passive
- Active

By function:

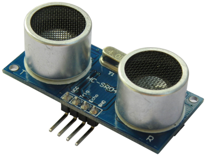
- Range sensors: Sonar, laser scanners, radar, tactile (like bumpers), ...
- Imaging sensors: Cameras (visual or infrared)
- Proprioceptive sensors: Encoders, inertial sensors, force sensors, torque sensors

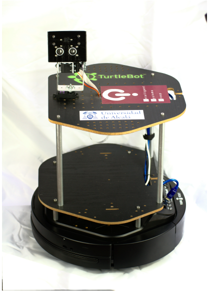
Sensors

Range sensors: Sonar

Provides a single distance measure

- Uses a sound pulse, usually ultrasound
- Short distance, usually for obstacle detection
- Not very precise
- Cheap ($\pm$ 2.5 euro)



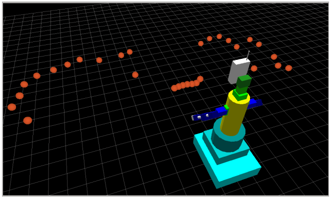


Sensors

Range sensors: Laser scanners

Provides precise ranges

- Several laser beams
- Extremely precise
- 1D, 2D and 3D laser scanners
- Very expensive (thousands)
- Useful for SLAM

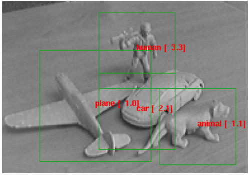


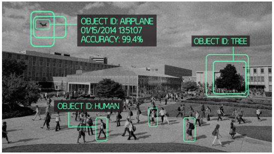
Sensors

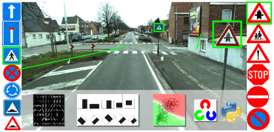
Imaging sensors: Cameras

Provides images of the environment

- Different wavelengths (visible or infrared)
- Very important in Robotics
- With proper algorithms, it is almost an universal sensor
- Need of computer vision
  - Object recognition, face recognition, obstacle detection, etc
- From cheap to expensive







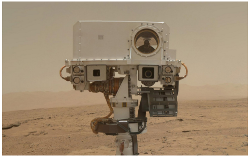
Sensors

Imaging sensors: Stereovision

Two parallel cameras

- Information fusion provides depth
- Similar to human vision
- The "poor man" alternative to laser scans
- Quite popular in Robotics



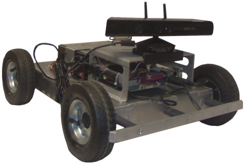


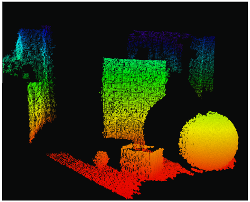
Sensors

Imaging sensors: Depth sensors

Provides image and depth

- Active depth sensor
- Kinect-type sensors
- Increasing popularity in Robotics





Sensors

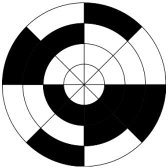
Proprioceptive sensors

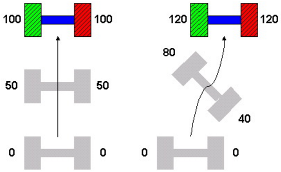
Sensors that provide information about the robot state

- Encoders, inertial sensors, force sensors, torque sensors, ...
- Needed for odometry

Odometry: Estimate of change in position over time

- Theoretical position differs from actual position
- Unbalanced motors, power unstability, sliding wheels, ...





Navigation

Perception

Mobile robots move around a physical environment

- Perception: Transform *noisy* sensor data into environment models

Perception must satisfy three conditions

- Complete
- Structured
- Reliable

Three problems

- Localization (unknown location)
- Mapping (unknown map)
- SLAM (unknown location and map)

Navigation

Localization

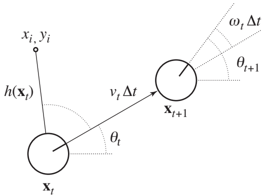
Localization: Where is the robot within the world?

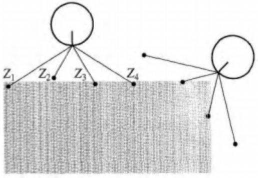
- Unknown location, known map
- Problem: Relate sensor readings to the world

Three problems

- Tracking, initial position known
- Global localization, initial position unknown
- Re-localization, incorrect known position (kidnapped robot problem)

Beacons: RFID tags, visual marks, QR codes, etc





Navigation

Localization: Monte-Carlo Localization

Monte-Carlo Localization (MCL)

- A type of particle filter
- Estimates robot location and orientation
- Uses a distribution of random particles that converge to robot position
- Prediction of sensor values after motion and compares with measures

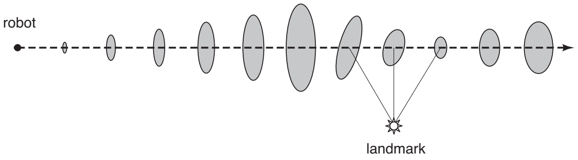
(Video particle filter)
(Video)

Navigation

Localization (II): Kalman filters

Kalman filters (or Extended Kalman Filters, EKF)

- Extended Kalman Filters, or EKF, is a popular extension
- Based on Bayesian Statistics
- Optimal for linear systems with Gaussian noise
    - Otherwise, MCL outperforms EKF
    - Needs less computational resources than MCL
- EKF used for normal operation, MCL used to solve ambigueties



Navigation

Mapping

Mapping: How is the wold around the robot?

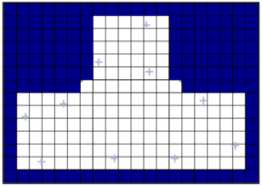
- Known location, unknown map
- Build map

Problem: Integrate sensor measurements to produce a map
(Video Underground Mine Mapping)

Navigation

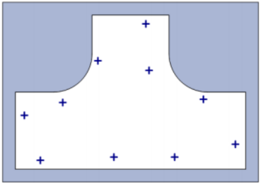
Mapping: Map models

Grid-based



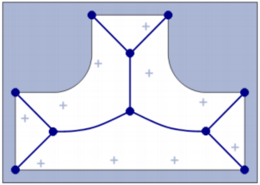
Collection of discretized pixels

Feature-based



Collection of landmark locations

Topological-based



Collection of nodes and links

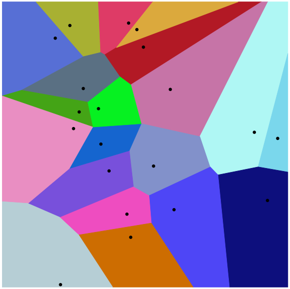
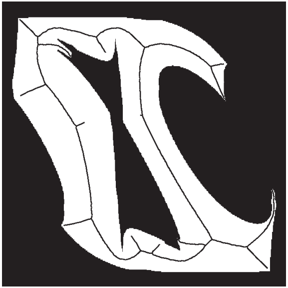
(Source)

Navigation

Mapping: Voronoi map

Voronoi diagram: Partition of a plane into regions based on distance points

- Regions boundaries maximize distance to obstacles
- Mapping of boundaries into graphs

(Source)

Navigation

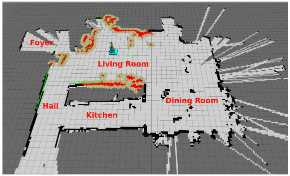
SLAM

SLAM: Map and location are unknown

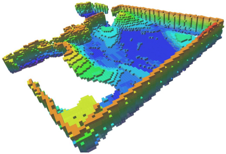
- Big, big problem in Robotics
- Huge number of applications

SLAM techniques

- Extended Kalman filters (EKF)
- Occupancy Grid Mapping



(Video SLAM)



(Video 3D SLAM)

Introduction (I)

Robotics platforms

- ROS
- Player
- Rock
- Pyro

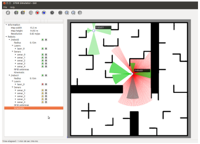
Problem: Testing

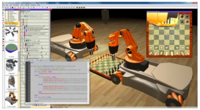
Introduction (II)

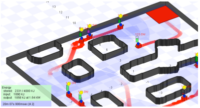
Simulation

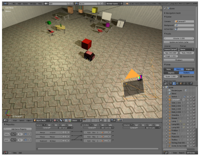
- Gazebo
- STDR
- V-Rep
- Stage
- MORSE
- Webots

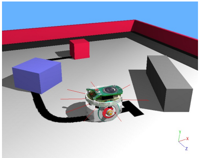
(Video)











Problem: Perception visualization

Introduction (III)

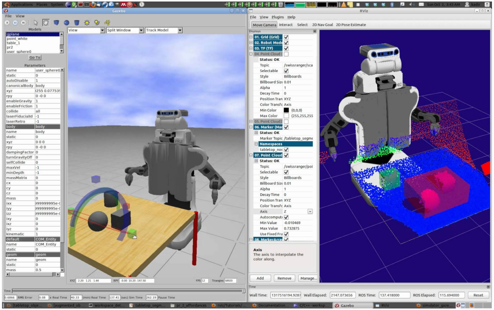
Sensor visualization

- RViz

Complex motion

- MoveIt!

(Video)



Introduction (IV)

The robot developer toolkit

- Robot
- Linux $\Rightarrow$ Operating system
- ROS $\Rightarrow$ Robotic platform
- Gazebo $\Rightarrow$ Simulator
- RViz $\Rightarrow$ Sensor visualization
- MoveIt! $\Rightarrow$ Motion operation

Robot Operating System (ROS) (I)

ROS.org

ROS is an "operating system" for robots

- Actually, despite its name, ROS is not an operating system
- Located on the operating system and under the applications
- ROS is a *robotic platform*

ROS was created in the Standford AI Lab, in 2007

- Continuous development since then
- Almost a standard in research
- Increasing popularity in the industry

Robot Operating System (ROS) (II)

ROS provides

- An execution environment
- A computation model (robot abstraction)
- Tools (like RViz and MoveIt!)
- Hardware independence

Features

- Robot drivers
- Advanced algorithms (SLAM, navigation, etc)
- Large number of supported sensors
- Pose estimation, localization, and navigation
- Standard robot messages

ROS versions

Version	Release	Logo	
Melodic Morenia	May 23rd, 2018		
Lunar Loggerhead	May 23rd, 2017		
Kinetic Kame	May 23rd, 2016		
Jade Turtle	May 23rd, 2015		
Indigo Igloo	July 22nd, 2014		Stable
Hydro Medusa	Sept. 4th, 2013		
Groovy Galapagos	Dec. 31, 2012		
Fuerte Turtle	April 23, 2012		
Electric Emys	August 30, 2011		

Significant differences among versions!

Installation

ROS runs over Linux

- Ubuntu recommended (official packages available)
- Ubuntu and ROS version strongly coupled
- Any Linux machine supports, in theory, ROS
- Warning: Installing ROS on weird platforms hurts!

Detailed installation instructions available

- `http://wiki.ros.org/jade/installation/`

ROSPlan: PDDL and ROS (yes, PDDL!!)

Documentation

- All the ROS documentation is available on its wiki
- ROS tutorials are a good entry point to ROS
- Be careful with ROS version
- Each robot is coded in a package, each package has its own documentation
- David F. Barrero's course