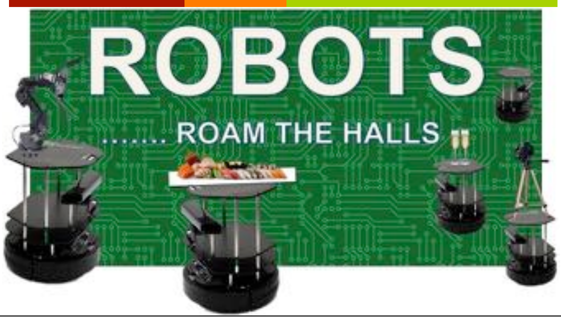
CS 189: Autonomous Robot Systems

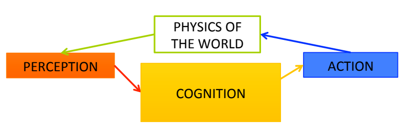
Spring 2018, Fridays 1-4pm, Pierce 301



Agenda

- Today's Agenda
  - Lecture: Architecture 1: Basics of Autonomy
  - Lab 1: Turtlebot Basics (movement and bump sensors)
- What happens next Friday?
  - **Pset 1: Robot Roomba. Due before & in class next Friday!**
  - Lab 2: Learn to use the camera.
- Reading this and next week:
  - PRR Chapters 1, 2, 3 (upto latched topics) and 6.

What Does it Mean to be Autonomous?



Basics of Autonomy

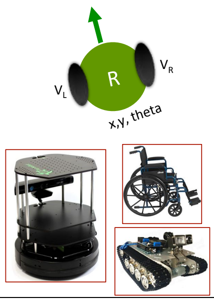
ACTION
- Action (Actuators)
  - Locomotion: Wheels (Differential Drives)
  - Also Legged, Aerial, Manipulation (Arms, Grippers)

PERCEPTION
- Perception (Sensors)
  - Proprioception (internal: IMU, encoders, etc)
  - Exteroception (external: Cameras! Color and Depth)

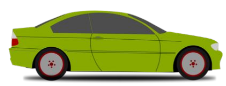
COGNITION
- Cognition (Control)
  - Today: Reactive Behaviors
  - Build up from there for the whole semester!

Robot Motion: Wheeled Robots

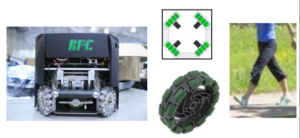
Differential Drive



Steered / Front-wheel Drive

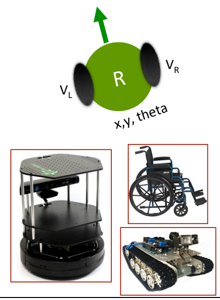


Holonomic



Robot Motion: Wheeled Robots

Differential Drive



Independently control speed of each wheel

- Move straight at different speeds $V_L = V_R$
- Turn on the Spot! $V_L = -V_R$
- PointA-to-PointB: Turn-Drive-Turn

Robot Motion: Wheeled Robots

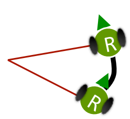
Differential Drive

Independently control speed of each wheel

- Move straight at different speeds $V_L = V_R$
- Turn on the Spot! $V_L = -V_R$
- PointA-to-PointB: Turn-Drive-Turn

Can do More

- Move in "curved lines"



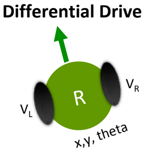
There's things you can't do!

- e.g. slide sideways

Because robot orientation
and movement are tied together
..... Non-holonomic

Robot Motion: Wheeled Robots

Differential Drive



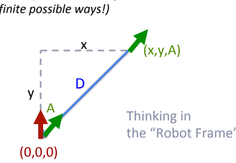
Motion Abstraction

Linear Velocity L
Angular Velocity W
About the center of the robot

INVERSE KINEMATICS
FORWARD KINEMATICS

Inverse Kinematics:

If I want to go from PointA to PointB
What inputs should I give?
(infinite possible ways!)



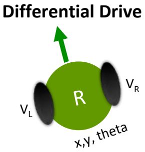
Popular Option: Turn-Move-Turn

Turn A = atan2 (x/y) = W x duration
Move D = sqrt (x^2 + y^2) = L x duration
(Turn again, to end in new orientation)

Caveat: Theory != Reality

Robot Motion: Wheeled Robots

Differential Drive

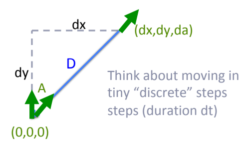


Motion Abstraction

Linear Velocity L
Angular Velocity W
About the center of the robot

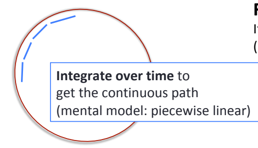
INVERSE KINEMATICS
FORWARD KINEMATICS

Forward Kinematics:
If I give inputs L and W, where will I end up?
(Curves = Continuous Motion)

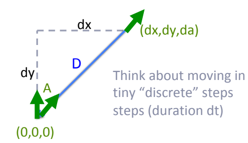


First turn A = Wdt Then move D = Ldt
dx/dt = DsinA = **L sinW**
dy/dt = DcosA = **L cosW**
da/dt = A = **W**

Robot Motion: Wheeled Robots

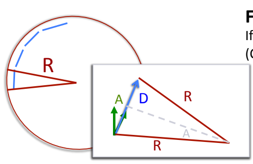


Forward Kinematics:
If I give inputs L and W, where will I end up?
(Curves = Continuous Motion)



First turn A = Wdt Then move D = Ldt
dx/dt = DsinA = **L sinW**
dy/dt = DcosA = **L cosW**
da/dt = A = **W**

Robot Motion: Wheeled Robots

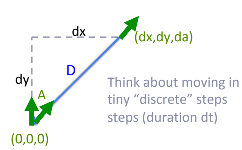


Trace a Circle
Given any fixed L and W, robot traces a circle

Radius R = (D/2) / sin(A/2)

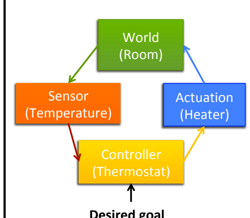
R = L / 2sinW/2
(control ratio of linear/angular velocities)

Forward Kinematics:
If I give inputs L and W, where will I end up?
(Curves = Continuous Motion)



First turn A = Wdt Then move D = Ldt
dx/dt = DsinA = **L sinW**
dy/dt = DcosA = **L cosW**
da/dt = A = **W**

What could possibly go wrong….



- **Open Loop Control**
  - Take a fixed sequence of actions, e.g. Driving in a square.
  - *When does this not work?*
- **Closed Loop Control**
  - Desired state (goal state, setpoint)
  - **Feedback** (i.e. compare measured state to desired)
  - *Classic Example: Thermostat*

Basics of Autonomy

ACTION
- Action (Actuators)
  - Locomotion: Wheels (Differential Drives)
  - Also Legged, Aerial, Manipulation (Arms, Grippers)

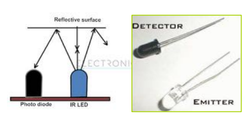
PERCEPTION
- **Perception (Sensors)**
  - Proprioception (internal: IMU, encoders, etc)
  - Exteroception (external: Cameras! Color and Depth)

COGNITION
- Cognition (Control)
  - Today: Reactive Behaviors
  - Build up from there for the whole semester!

Perception

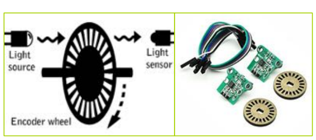
- Proprioception: Sense the Internal State of the Robot
  - Wheel encoders (detect skidding/slipping)
  - Inertial Measurement Unit (IMU)
  - Many others, e.g. wheeldrop, battery levels
- Exteroception: Sense the **external state of the environment**
  - Bump sensors!
  - Cameras: RGB and Depth
  - Many others, e.g. Sonar, LIDAR (self-driving cars)
    - *Key: Sensors measure physical qualities in the world (e.g. light or signal levels). They don't interpret the state of the world.*

Common Sensors

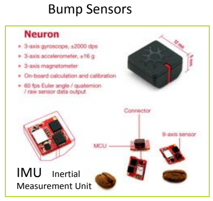


Infrared (IR) Proximity Sensors


Bump Sensors

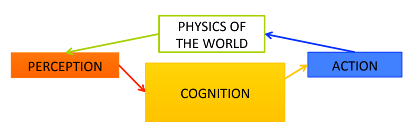


Wheel Encoders

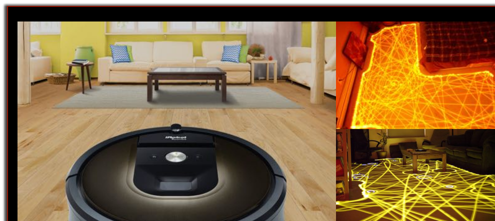


What is the world's simplest robot?

What could you do if you just had a simple differential drive and a bump sensor?



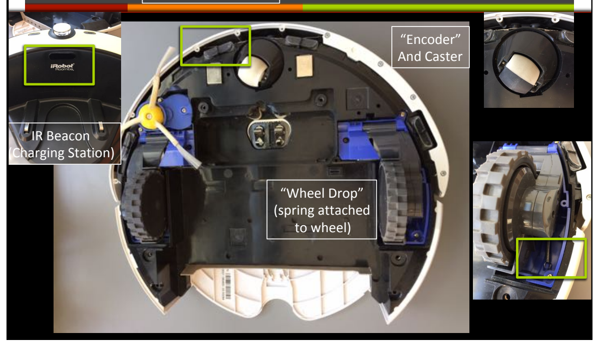
Roomba!



iRobot Founders --- Helen Greiner, Rodney Brooks, Colin Angle

Roomba!

BUMP SENSORS
(springy surface + IR)

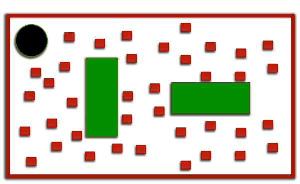


Reactive Programming

SIMPLE ROOMBA

If BUMP = TRUE
THEN Turn (random direction/amt)

ELSE MOVE-STRAIGHT



A simple roomba does a random walk to cover a complex unknown space.

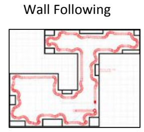
A more complex roomba, might have more bumpers (left/right) or might have more interesting default behaviors (zigzag, spirals).

But its still pretty simple.....

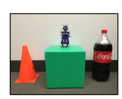
Lab 1: Motion and Bump Sensors
➔ Pset 1 your Roomba

Many Complex Behaviors are "Reactive"

Wall Following



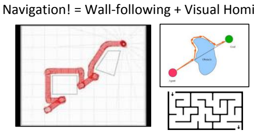
Visual Homing



Centering

Collision Avoidance

Navigation! = Wall-following + Visual Homing



Flocking = Neighbor Feedback

Perception: Beyond Bumpers

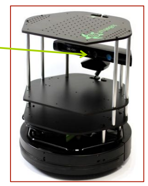
Stereo Camera

Sonar Ring

LIDAR

Pioneer robot with stereo cameras, sonar ring, and LIDAR

RGBD Camera



Can we measure obstacles at a distance?

Many Ways to Sense Depth

- IR Ranging
- LIDAR
- RADAR
- SONAR
- Stereo
- Depth Cameras

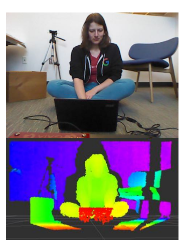
Amazing array of possible techniques by which to get a "depth" image of your surrounding.

- **Active systems** rely on sending light/radio/sound pulses and measuring something about the reflection (time of flight, intensity, distortion).
- **Passive systems** rely on optic properties (like stereo vision, depth from focus)

What technique you use depends on the environment. And one modality will probably not be enough….

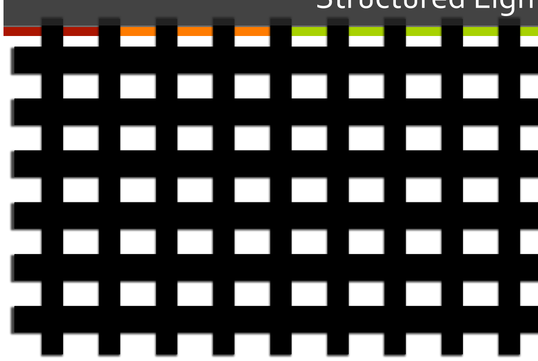
Modern Depth Cameras

- Depth Cameras
  - Kinect, Asus Xtion Pro, Intel Realsense
  - **2D Image of "depth" values**
    - Directly manipulate the distance values
    - Treat as grayscale image (OpenCV)
- When is Depth Better?
  - **Robot movement**
    - **Collision avoidance, navigation/mapping**
    - **We will use depth a lot! (Pset 2 onwards)**
- How do they work*:
  - (1) Structured Light (Active depth sensing)
  - (2) Depth from Blur and Stereo



**These slides are adapted from "How Kinect Works", talk by Prof. John MacCormick, Dickinson College.*

Structured Light



Kinect uses Structured Light

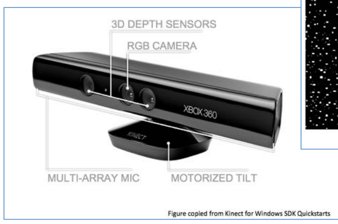
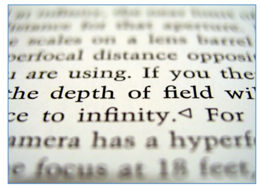


Figure copied from Kinect for Windows SDK Quickstarts

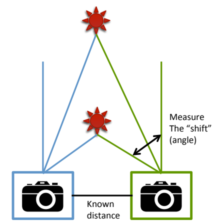
Shpunt et al, PrimeSense patent application US 2008/0106746

Project a speckle pattern of infrared laser light
Capture and analyze the result!
(PrimeSense patent, licensed by Microsoft and others, works at high frame rates!)

Kinect ALSO uses Blur and Stereo



From Wikimedia



Classic Computer Vision Techniques
(1) BLUR: Things further away look blurrier (astigmatic lens – best at your focal length)
(2) STEREO: Things viewed from two vantage points, look different based on distance

With so much smarts, what could go wrong!

Every sensor technology comes with its "failure modes"

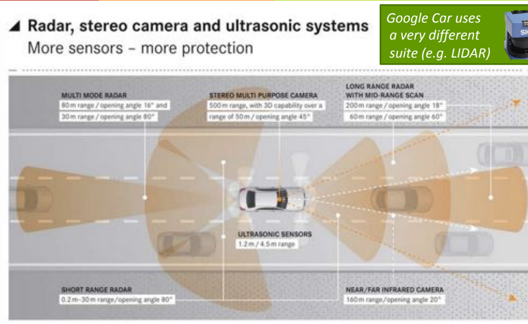
- E.g. Kinect-like depth cameras depend on projecting and observing infrared light.
  - May have lots of "Missing" parts ("NaNs")
  - Ambient IR can saturate your sensors (sunlight!)
  - Reflectivity and absorption depends on materials (glass, metal)
  - Limited range (energy) compared to lasers

  (Turned out to be perfect for indoor Gaming though!)

- RGB Cameras also have failure modes (more familiar)
  - Saturation (sunlight/dark), Artifacts (red eye, shadows)

- **ANSWER:** Many many sensors! And Many many experiments!

Example: Tesla



Google Car uses a very different suite (e.g. LIDAR)

Wrap-up and onto LAB

- Today's Agenda
  - Lecture: Architecture 1: Basics of Autonomy
  - **Lab 1: Turtlebot Basics (movement and bump sensors)**
- What happens next Friday?
  - **Pset 1: Robot Roomba. Due before & in class next Friday!**
  - Lab 2: Learn to use the camera.
- Reading this and next week:
  - PRR Chapters 1, 2, 3 (upto latched topics) and 6.

Lecture and Office Hour slides are available on canvas under **MODULES**
Reading is listed in lecture slides and on **SYLLABUS;** please read **all LAB materials**