Part I: Preliminaries
Agency, Motion and Anatomy

- the human race is improving their bodies and minds by technology
  - smartphones
  - prostheses
  - ...
- the future might be full of silicone-based beings possessing biological prostheses and vice versa
- the difference between humans and machines fades
  - today’s problem: If Google car causes an accident, who is responsible?
Comparison of Motion Efficiency

- **wheels**
  - very efficient on hard ground, not as much on soft ground
  - requires few and simple actuators

- **legs**
  - does not lose efficiency on soft ground
  - difficult to build for engineers
  - easy to build for nature
Accessibility vs Throughput
Legged Robots

Aldebaran Nao

Robonaut (NASA)
Legged Robots

Alpha dog (Boston Dynamics)

Genghis (MIT)
Wheeled Robots: Differential Drive

- wheels on both sides are separately and independently powered
Wheeled Robots: Ackerman Drive

- allows the inner turning wheel to turn at a larger angle than the outer turning wheel
- Ackerman is used in motor vehicles
Agency, Motion and Anatomy: Conclusion

- our fascination with biologically-inspired robots has led us to broad diversification
- there are robots using bird-like wings and robots which adapt to locomotion without knowledge of their own anatomy
Behaviors

- behavior-based robotics
  - interactions of a robot in an environment are likened to a finite set of distinct self-contained animal behaviors
  - animal behavior is innate and instinctive
- selection is responsible for developing breeds with unique behavior
- example of instinctive behavior
  - all cows, everywhere, face north or south while eating
  - animals leave an area when they sense an earthquake
Behaviors

● simple behavior may have complex causes
  ○ A Braitenberg “Alive” vehicle
    ■ two light sensors connected to the motors
    ■ with connection to the motor on the same side, the vehicle prefers dark places
    ■ with connection to the motor on the other side, the vehicle prefers light places
Common behavior types

- reflexes
  - fast and involuntary movements
- threshold-based behavior
- taxis
  - instinctive tendency to approach or avoid particular stimulus
- tropotaxis
  - taxic behavior in which sensory inputs are compared
- condition-based taxis
  - maintain a condition during a response
Common behavior types

● swarming
  ○ emergent, collective behavior of decentralized, self-organized systems, natural or artificial

A flock of auklets  Symbion project
Behaviors

- an issue that arises is the need for organization among competing behaviors
- possible solution: subsumption architecture

higher level subsumes lower levels
Architectures

- set of principles of robot behavior
- sense-think-act
  - traditional AI approach
  - borrows from human cognition
- subsumption architecture
  - replaced classic pipeline (horizontal) models
- 3T (three-tier) model
  - uses learning to migrate routine activities to higher levels
- RCS
  - under development in National Institute of Standards
  - perhaps the most complex
Architectures

- feedback-loop
  - usually used at low-level control
  - e.g., PID controller
Affect

- phenomenon that manifests itself in the form of feelings
- “To be alive is to have feelings.”
- robots have to understand human emotions in order to improve human-robot cooperation
- humans often use emotions for decision making when there is an absence of rational motivation
Uncanny valley

- human-like robots might seem “creepy”
- it is called the uncanny valley
Sensors

- **active**
  - emits signal to the environment
  - RADAR
  - SONAR

- **passive**
  - only receives signal already present in the environment
  - IR sensor for human body detection
  - accelerometer, gyroscope
Manipulators

- manipulators are composed of links and joints
- flexibility is measured in degrees of freedom (DOF)

Puma by Unimation, branch of Westinghouse (6 DOF)  Utah/MIT robot hand
Manipulators

● arm links can be seen as kinematic chains
● forward kinematic
  ○ known robot configuration → arm position
  ○ Denavit-Hartenberg (1955)
● backward kinematic
  ○ arm position → robot configuration (ambiguous)
Part II: Mobility
Potential fields

- one of the simplest methods to reach a goal position
- goal can also represent desired position of manipulator arm

\[(\Delta x, \Delta y) = \nabla P(x, y) = \left( \frac{\partial P}{\partial x}, \frac{\partial P}{\partial y} \right)\]
Potential fields

- obstacles can emit repulsive force
Roadmaps

- a robot might need a map of its environment
- roadmap is usually represented as a graph structure with nodes as places and edges as roads
- path is a set of connected roads
  - for path finding, the most common algorithm is A* and its variations
- vector map
  - graphs of nodes and edges to denote vertices of polygons
- raster map
  - the map is sampled to cells which might be occupied or free
Geometry-based Roadmaps

1. radial sweep from the start point
Geometry-based Roadmaps

2. remove invalid paths by eliminating segment colliding with an obstacle
Geometry-based Roadmaps

3. repeat using each obstacle vertex as new starting point
Geometry-based Roadmaps

3. repeat using each obstacle vertex as new starting point
Geometry-based Roadmaps

3. repeat using each obstacle vertex as new starting point
Geometry-based Roadmaps

4. convert to visibility graph of obstacle vertices
Geometry-based Roadmaps

- to avoid collisions, the obstacles might be dilated
- dilatation can, however, cause problems with narrow passages
Grids

- basic occupancy map
- resolution of the grid is an issue
  - large grids have a lower resolution and may result in blocking paths
  - higher resolution is computationally expensive
Voronoi Roadmaps (Grid-based Approach)

- offers maximum clearance between obstacles
Probabilistic Roadmaps

- created by randomly selecting free points and interconnecting them
- not necessarily lead to a solution as the graph might not be connected
Probabilistic Roadmaps - KD Trees

- probably the most popular probabilistic method
- constructed by alternatively subdividing cells with vertical and horizontal lines
- the tree is used to identify probable neighbors
- the goal is to have one point per cell
Rapidly Exploring Random Tree (RRT)

- can be built by constructing rooted graph at the start location and randomly going toward the goal

Parameters of the algorithm are:
- Number of nodes represented by \( n \)
- Edge Length (step size) of exploration represented by \( s \)

1. Let \( V \) contain the start vertex and let \( E \) be empty
2. Repeat
   a. Let \( q \) be a random valid robot configuration (i.e., a random point)
   b. Let \( v \) be the node of \( V \) that is closest to \( q \)
   c. Let \( p \) be the point along the ray from \( v \) to \( q \) that is at distance \( s \) from \( v \)
   d. If \((vp)\) is a valid edge) THEN add new node \( p \) to \( V \) with parent \( v \)
3. Until \( V \) has \( n \) vertices
Roadmaps: Conclusions

- roadmaps have been one of the most extensively explored area of robotics
- with reliable positioning, the need for outdoor map building disappears
- there exist a few commercial projects (such as at Google) that offer indoor maps for public use
- in the absence of available maps, robot has to build its own maps
Reactive Navigation

- study of planning and controlling course and position of the robot toward a desired location, where knowledge of objects in the environment is only known after the movement begins
- navigation is a set of reactive strategies
- often called bug algorithms
Bug 0 Strategy

1. Head toward the goal
2. Follow obstacles until you can head toward the goal again
3. Continue
Bug 1 Strategy

1. go around an obstacle in order to find the point closest to the goal
2. travel back to the closest point
3. resume moving toward the goal
Bug 2 Strategy

1. move toward the goal along the m-line
2. if an obstacle is encountered move along its perimeter until the m-line is reached
Tangent Bug Strategy

- considers a robot with 360° proximity sensor with a limited range
- robot chooses the discontinuity point minimizing the travel distance to the goal
Multi-Robot Mapping

● Ants algorithm
  ○ at the start, all cells are marked with a zero
  ○ at each steps, agent picks the least visited cell to move into
  ○ flexible and fault tolerant

● Multiple depth-first search
  ○ each agent builds its own DFS tree and stores its own ID in the cells
  ○ agents stay in their own trees, unless they exhaust unexplored cells

1. IF the current cell is unexplored, THEN mark it as explored
2. Annotate the cell with your ID
3. Annotate the cell with the direction of the previous cell (parent cell)
4. END IF
5. IF there are unexplored cell around, then go to one of them randomly
6. ELSE IF the current cell is marked with your ID, THEN mark it as visited
7. Go to the parent cell
8. ELSE go to one of the explored cells randomly
9. END IF
10. END IF
Fuzzy Control

- captures imprecision in human natural language

Example of a fuzzy membership for the concept of “tall”

Another fuzzy membership functions: people consider overlapping height average concepts rather than a single function