Introduction to Robot Motion Planning

Ron Parr
CPS 570

What is motion planning

• Basic problem in robotics
• Given:
  – Initial robot state
  – Environment
  – Physical laws
  – Goal state
• Produce:
  – Sequence of motions that will bring robot to goal state without violating physical laws or crashing into obstacles in the environment

Contrast with Classical Planning

• Robot state and environment state are often presented as distinct things
• Actions are physics-based, not logic based
• State is continuous
Why isn’t motion planning trivial?

- Obstacles
- Physical limits
- Complex goal criteria

What is an obstacle?

- Physical barrier (car hood, door jamb, etc.)
- Fragile environmental feature (window, person, etc.)
- Physical limit (speed limit, joint limit) – though note that treating physical limits as obstacles is just one way to deal with them

What is a robot?

- Robot arm
- Humanoid
- Spaceship
- Flexible needles
- Protein

- Anything with movable parts that we wish to maneuver into a desired configuration

Configuration Space (c-space)

- Describe a robot with a set of state variables
  - \(x, y, \theta\) for a planar robot (Roomba) (3-DOF)
  - \(x, y, z, \text{roll}, \text{pitch}, \text{yaw}\) for an aircraft (6-DOF)
  - Baxter has 7-DOF per arm
  - Aldebaran Nao has 25-DOF total

- Configuration space is the continuous space of these state variables – every point is a complete state

- Configuration space is not necessarily just DOF
  - May include velocities, higher derivatives
  - May include other variables that change, e.g., mass
What is an obstacle?

• Obstacles are forbidden regions of the configuration space
• Map physical (or virtual) obstacles into obstacles in the configuration space
• Motion planning reduces to finding a collision free path of a point through configuration space
• What’s missing?

Robots Occupy Space

Solution: Augment obstacles

Augmenting Obstacles

• Augmenting is easy if your robot is a circle
• What if your robot is complicating thing like Baxter?
  – Figuring out how to augment the obstacles can itself be a nasty problem
  – May consider approximate methods, e.g., putting a circle around the robot, if the shape of the robot is not critical to the tax
  – Might need to use approximate (incomplete) methods in general
Approaches to Planning

• Potential fields
• Discretization
• Sampling

Potential Fields

• Goal attracts
• Obstacles repel
• Simulate physics to find path to goal

Potential Field Example

Potential Field Challenges

• What is the strength of the fields?
• In previous example:
  – If obstacles are too repulsive, robot hides in the bottom left corner
  – If goal is too attractive, robot gets stuck near intersection of obstacles
• In general, can’t guarantee local optima will not trap robot
• May need to add random perturbation, other tricks to avoid local optima
Assumptions Behind Many Methods

- "Local planning" between two close points with no intervening obstacles is easy
- Given a point or small region, detecting of an obstacle is present is easy
- Determining the direction towards to goal is easy

Cell Decomposition

- Rough overview:
  - Cell decomposition is a trivial form of discretization
  - Break problem up into convex, polygonal regions
  - Plan to move between vertices or other selected points in/on the polygons

  - Nice approach but doesn’t scale to high dimensions

Uniform Discretization

- Divide configuration space up into cells
  - Cells could be hypercubes
  - Could be more carefully chosen shapes based upon geometric properties of the problem
  - Generate a coarse plan at the level of cells
- Produce a refined plan at level of continuous movements

Variations

- Many ways to conduct a search through a set of discrete cells
- Can trade off search complexity with c-space complexity, e.g.:
  - Adding velocity to the c-space increases the dimension of the c-space
  - Keeping track of a velocity constraint while searching a c-space without velocity complicates search
Costs and Guarantees

- Most formulations of motion planning problems are intractable – scale poorly with number of obstacles and/or dimension
- Cell decomposition methods will be complete up to the limits of discretization
- What’s the catch?
- Discretization is exponentially expensive in the dimension
- What’s the workaround?
- Hope that you can avoid generating all cells, i.e., avoid worst case

Non-uniform sampling

- Suppose you had a clever way of picking “interesting” points in configuration space
- Run a local planning to see which of these are easily reachable from each other
- Construct a graph, find shortest path, etc.

Sampling Methods

- Reason about points instead of cells
- Simplest version:
  - Cover configuration space with regularly spaced array of points – make these vertices of a graph
  - (note exponential cost in dimension)
  - Create edges between points that are reachable
  - Use shortest path algorithm on the graph
  - Optionally, refine motion plan
- Not that different from cell decomposition (yet)

Variations

- Sample randomly
- Cover the space by growing out from specific points
- Use human guidance to select where/how to pick points
- Guide any of the above with domain knowledge, heuristics, potential functions, etc.
**Pro:** Open spaces are easy

**Con:** Narrow passages are hard

What if we must traverse a narrow passage? May be difficult to generate samples in this region. Think about threading a needle.

**Sampling based methods - summary**

- Sampling methods do well in problems with lots of open space – no narrow passages
- Can lead to a different, and potentially helpful way of thinking about the problem – hardness comes from structure of problem, rather than dimension of space
- Interesting to compare w/GSAT

- Not complete
- Not a panacea, but another view on the problem

**Conclusions**

- Robot motion planning differs from classical planning in its use of continuous variables and physical actions/constraints
- Computationally difficult
- No silver bullet
- Different approaches have different trade offs

- Can be used to solve some difficult problems in practice!