Chapter 25

Outline

Robots, Effectors, and Sensors
Localization and Mapping
Motion Planning
Motor Control

Non-holonomic robots

A car has more DOF (3) than controls (2), so is non-holonomic; cannot generally transition between two infinitesimally close configurations

Mobile Robots

Sensors

Range finders: sonar (land, underwater), laser range finder, radar (aircraft), tactile sensors, GPS

Imaging sensors: cameras (visual, infrared)
Proprioceptive sensors: shaft decoders (joints, wheels), inertial sensors, force sensors, torque sensors

Manipulators

Configuration of robot specified by 6 numbers
\(\Rightarrow\) 6 degrees of freedom (DOF)

6 is the minimum number required to position end-effector arbitrarily. For dynamical systems, add velocity for each DOF.
Localization—Where Am I?

Compute current location and orientation (pose) given observations:

\[
A_{t-2} \rightarrow X_{t-1} \rightarrow X_t \rightarrow X_{t+1} \rightarrow Z_{t-1} \rightarrow Z_t \rightarrow Z_{t+1}
\]

Assume Gaussian noise in motion prediction, sensor range measurements.

Localization contd.

Can use particle filtering to produce approximate position estimate.

Mapping

Localization: given map and observed landmarks, update pose distribution.
Mapping: given pose and observed landmarks, update map distribution.
SLAM: given observed landmarks, update pose and map distribution.
Probabilistic formulation of SLAM:
add landmark locations \( L_1, \ldots, L_k \) to the state vector, proceed as for localization.

Mapping contd.
Motion Planning

Idea: plan in configuration space defined by the robot's DOFs

Solution is a point trajectory in free C-space

Configuration space planning

Basic problem: $\infty^3$ states! Convert to finite state space.

Cell decomposition:
divide up space into simple cells,
each of which can be traversed “easily” (e.g., convex)

Skeletonization:
identify finite number of easily connected points/lines
that form a graph such that any two points are connected
by a path on the graph

Skeletonization: Voronoi diagram

Voronoi diagram: locus of points equidistant from obstacles

Problem: doesn’t scale well to higher dimensions

Skeletonization: Probabilistic Roadmap

A probabilistic roadmap is generated by generating random points in C-space
and keeping those in freespace; create graph by joining pairs by straight lines

Problem: need to generate enough points to ensure that every start/goal
pair is connected through the graph
Motor control

Can view the motor control problem as a search problem in the dynamic rather than kinematic state space:
- state space defined by \( x_1, x_2, \ldots, x_1, x_2, \ldots \)
- continuous, high-dimensional (Sarcos humanoid: 162 dimensions)

Deterministic control: many problems are exactly solvable
esp. if linear, low-dimensional, exactly known, observable

Simple regulatory control laws are effective for specified motions

Stochastic optimal control: very few problems exactly solvable
\( \Rightarrow \) approximate/adaptive methods

---

Biological motor control

Motor control systems are characterized by massive redundancy

Infinitely many trajectories achieve any given task

E.g., 3-link arm moving in plane throwing at a target
    simple 12-parameter controller, one degree of freedom at target
    11-dimensional continuous space of optimal controllers

Idea: if the arm is noisy, only “one” optimal policy minimizes error at target

I.e., noise-tolerance might explain actual motor behaviour


---

Simple learning algorithm: Stochastic gradient

Minimize \( E_0[y^2] \) by gradient descent:

\[
\nabla_\theta E_0[y^2] = \nabla_\theta \left[ \frac{1}{N} \sum_{j=1}^{N} P_{\theta_0}(\theta_j) F(\theta_j)^2 \right] d\theta
\]

\[
= \frac{1}{N} \nabla_\theta \left[ \sum_{j=1}^{N} P_{\theta_0}(\theta_j) F(\theta_j)^2 \right] P_{\theta_0}(\theta_j) d\theta
\]

\[
= E_0 \left[ \frac{\nabla_\theta P_{\theta_0}(\theta)}{P_{\theta_0}(\theta)} y^2 \right]
\]

Given samples \((\theta_j, y_j), j = 1, \ldots, N\), we have

\[
\nabla_\theta E_0[y^2] = \frac{1}{N} \sum_{j=1}^{N} \frac{\nabla_\theta P_{\theta_0}(\theta_j)}{P_{\theta_0}(\theta_j)} y_j^2
\]

For Gaussian noise with covariance \( \Sigma \), i.e., \( P_{\theta_0}(\theta) = N(\theta_0, \Sigma) \), we obtain

\[
\nabla_\theta E_0[y^2] = \frac{1}{N} \sum_{j=1}^{N} \Sigma^{-1}(\theta_j - \theta_0)y_j^2
\]

---

Setup

Suppose a controller has “intended” control parameters \( \theta_0 \)
which are corrupted by noise, giving \( \theta \) drawn from \( P_{\theta_0} \)

Output (e.g., distance from target) \( y = F(\theta) \);

---

Results for 2-D controller

![Graph showing results for 2-D controller]
Summary

The rubber hits the road

Mobile robots and manipulators

Degrees of freedom to define robot configuration

Localization and mapping as probabilistic inference problems

(requires good sensor and motion models)

Motion planning in configuration space

requires some method for finitization