CMPT 882
Special Topics in Artificial Intelligence
Robotic Decision Making
Mo Chen
https://www.sfu.ca/~mochen
Course Outline

• Overview of algorithms used for robotic decision making
  • Theory-focused
  • Fundamentals for doing many areas of robotics research

• Dynamical systems
• Nonlinear optimization
• Optimal control
• Machine learning in robotics
• Localization and mapping
Logistics

• Academic Quadrangle 5030, MWF 12:30-13:20

• Office hour: Thursdays 15:00-16:00, TASC 1 8225

• Course website: https://coursys.sfu.ca/2019sp-cmpt-882-g1/pages/

• Contact: mochen@cs.sfu.ca
Caveats

• This class is in “experimental mode”

• Slight changes are expected

• Some things may not be super polished

• Please provide feedback and comments
Grading

• 40% Homework
  • 5 assignments, one on each unit

• 60% Project
Project suggestions

• Thoroughly understand and critically evaluate 3 to 5 papers in an area covered in this course

• Reproduce the results of 1 to 2 papers in an area covered in this course, and suggest or make improvements

• Implement an algorithm covered in or related to the class on a robot

• Mini Research project related to an area covered in this course
Project timeline

• Proposal (1-2 paragraphs)
  • Due Feb. 18

• Presentation
  • Last three classes of the semester

• Report (5 pages maximum)
  • Due Apr. 20
Recommended textbooks


Dynamical systems

• Mathematical models of robotic systems
  • Deterministic vs. stochastic
  • Continuous vs. discrete time

• Configuration of system described by the state, often denoted $x$
  • State changes, or evolves, according to the model
Dynamical systems

- Mathematical models of robotic systems
  - Deterministic vs. stochastic
  - Continuous vs. discrete time

- Configuration of system described by the state, often denoted $x$
  - State changes, or evolves, according to the model

- Deterministic, continuous time
  - $\frac{dx}{dt} = \dot{x}(t) = f(x, u)$
Dynamical systems

• Mathematical models of robotic systems
  • Deterministic vs. stochastic
  • Continuous vs. discrete time

• Configuration of system described by the state, often denoted $x$
  • State changes, or evolves, according to the model

• Deterministic, continuous time
  • $\frac{dx}{dt} = \dot{x}(t) = f(x, u)$

• Stochastic, discrete time
  • $x_{k+1}$ obtained from the probability distribution $p(x_{k+1} | x_k, u_k)$
System State

• Defined in terms of any variables of interest
  • Often denoted $x(t)$ or $x_k$

• Position
• Heading
System State

• Defined in terms of any variables of interest
  • Often denoted $x(t)$ or $x_k$

• Position
• Heading

• Velocity
• Angular velocity
System State

• Defined in terms of any variables of interest
  • Often denoted $x(t)$ or $x_k$

• Position
• Heading

• Velocity
• Angular velocity

• Voltages, concentrations of chemicals
• Human comfort, degree of trust
Control and disturbance

• Control/action: usually used to achieve a desired goal
  • Usually denoted $u(t)$ or $u_k$
  • Acceleration
  • Turn rate
Control and disturbance

• Control/action: usually used to achieve a desired goal
  • Usually denoted $u(t)$ or $u_k$
  • Acceleration
  • Turn rate

• Gas throttle
• Steering wheel angle
Control and disturbance

• Control/action: usually used to achieve a desired goal
  • Usually denoted $u(t)$ or $u_k$
  • Acceleration
  • Turn rate
  • Gas throttle
  • Steering wheel angle

• Disturbance
  • Usually denoted $d(t)$ or $d_k$
  • Bumps on the road
  • Input noise
Mathematical models of robotic systems

- Continuous state
  - Continuous time
    - Linear
      - Time-varying
      - Time-invariant
    - Nonlinear
      - Time-varying
      - Time-invariant
  - Discrete time
    - Discrete state
Mathematical models of robotic systems

- Continuous state
  - Continuous time
  - Linear
    - Time-varying
    - Time-invariant
  - Discrete time
    - Linear
      - Time-varying
      - Time-invariant
    - Nonlinear
      - Time-varying
      - Time-invariant

- Discrete state
Examples of Robotic Systems

Amazon.com Inc.

Google Inc.
Examples of Robotic Systems
Examples of Robotic Systems
Car models

\[
\begin{align*}
\dot{x} &= v \cos \theta \\
\dot{y} &= v \sin \theta \\
\dot{\theta} &= \omega
\end{align*}
\]

States: \((x, y, \theta)\); position and heading
Control: \(\omega\); turn rate (angular speed)
Car models

1. \[
\begin{align*}
\dot{x} &= v \cos \theta \\
\dot{y} &= v \sin \theta \\
\dot{\theta} &= \omega
\end{align*}
\]
States: \((x, y, \theta)\); position and heading
Control: \(\omega\); turn rate (angular speed)

2. \[
\begin{align*}
\dot{x} &= v \cos \theta \\
\dot{y} &= v \sin \theta \\
\dot{\theta} &= \omega \\
\dot{v} &= a \\
\dot{\omega} &= \alpha
\end{align*}
\]
States: \((x, y, \theta, v, \omega)\); position, heading, speed, turn rate
Control: \((a, \alpha)\); acceleration, angular acceleration
Car models

1. \[
\begin{align*}
\dot{x} &= v \cos \theta \\
\dot{y} &= v \sin \theta \\
\dot{\theta} &= \omega
\end{align*}
\]
   States: \((x, y, \theta)\); position and heading
   Control: \(\omega\); turn rate (angular speed)

2. \[
\begin{align*}
\dot{x} &= v \cos \theta \\
\dot{y} &= v \sin \theta \\
\dot{\theta} &= \omega \\
\dot{v} &= a \\
\dot{\omega} &= \alpha
\end{align*}
\]
   States: \((x, y, \theta, v, \omega)\);
   position, heading, speed, turn rate
   Control: \((a, \alpha)\);
   acceleration, angular acceleration

3. \[
\begin{align*}
\dot{x} &= v_x \\
\dot{y} &= v_y \\
\dot{v}_x &= \omega v_y + a_x \\
\dot{v}_y &= -\omega v_x + \frac{2}{m} (F_{c,f} \cos \delta_f + F_{c,r}) \\
\dot{\psi} &= \omega \\
\dot{\omega} &= \frac{2}{I_z} (l_f F_{c,f} - l_r F_{c,r}) \\
\dot{X} &= v_x \cos \psi - v_y \sin \psi \\
\dot{Y} &= v_x \sin \psi + v_y \cos \psi
\end{align*}
\]
Car models

1. \[ \dot{x} = v \cos \theta \]
   \[ \dot{y} = v \sin \theta \]
   \[ \dot{\theta} = \omega \]

States: \((x, y, \theta)\); position and heading
Control: \(\omega\); turn rate (angular speed)

2. \[ \dot{x} = v \cos \theta \]
   \[ \dot{y} = v \sin \theta \]
   \[ \dot{\theta} = \omega \]
   \[ \dot{v} = a \]
   \[ \dot{\omega} = \alpha \]

States: \((x, y, \theta, v, \omega)\);
Control: \((a, \alpha)\);

acceleration, angular acceleration

3. Bicycle model

\[ \dot{x} = v_x \]
\[ \dot{y} = v_y \]
\[ \dot{v}_x = \omega v_y + a_x \]
\[ \dot{v}_y = -\omega v_x + \frac{1}{m} \left( F_{c,f} \cos \delta_f + F_{c,r} \right) \]
\[ \dot{\psi} = \omega \]
\[ \dot{\omega} = \frac{2}{I_z} \left( l_f F_{c,f} - l_r F_{c,r} \right) \]
\[ \dot{X} = v_x \cos \psi - v_y \sin \psi \]
\[ \dot{Y} = v_x \sin \psi + v_y \cos \psi \]
Models

• All models are wrong; some are useful
Models

- All models are wrong; some are useful

- Definition of “useful” depends on situation
  - Simulation
  - Analysis and control
  - Verification
Models

• All models are wrong; some are useful

• Definition of “useful” depends on situation
  • Simulation
  • Analysis and control
  • Verification

• Considerations
  • Does the model capture the desired system behaviours
  • Is the model amenable to tractable computation
Nonlinear Optimization

• Choose $x$ to minimize some cost, subject to constraints
Nonlinear Optimization

• Choose \( x \) to minimize some cost, subject to constraints

\[
\begin{align*}
\text{minimize} & \quad f(x) \\
\text{subject to} & \quad g_i(x) \leq 0, \quad i = 1, \ldots, n \\
& \quad h_j(x) = 0, \quad j = 1, \ldots, m
\end{align*}
\]

Fuel cost, distance to obstacles, distance from goal, prediction error in machine learning

System dynamics, obstacle avoidance, goal reaching

• Equivalently, maximize \(- f(x)\): Maximize reward, maximize profit
Nonlinear Optimization

• Choose $x$ to minimize some cost, subject to constraints

\[
\begin{align*}
\text{minimize} & \quad f(x) \\
\text{subject to} & \quad g_i(x) \leq 0, \ i = 1, \ldots, n \\
& \quad h_j(x) = 0, \ j = 1, \ldots, m
\end{align*}
\]

Fuel cost, distance to obstacles, distance from goal, prediction error in machine learning

System dynamics, obstacle avoidance, goal reaching

• Equivalently, maximize $-f(x)$: Maximize reward, maximize profit

• Robotics spans many fields
  • Many conventions
  • Many notations clashes
Nonlinear Optimization

- A very difficult problem in general for $x \in \mathbb{R}^n$ where $n$ is large
  - Calculus facts: necessary and sufficient conditions
  - Rely on gradients (if possible)

- Sometimes, some components of $x$ may be integers

- Can we do better than brute force?

- Simpler cases

- Differentiable functions
  - Linear, convex, quasiconvex
  - Unconstrained problems
Nonlinear Optimization

• A very difficult problem in general for $x \in \mathbb{R}^n$ where $n$ is large
  • Calculus facts: necessary and sufficient conditions
  • Rely on gradients (if possible)

• Sometimes, some components of $x$ may be integers
  • Can we do better than brute force?
Nonlinear Optimization

• A very difficult problem in general for $x \in \mathbb{R}^n$ where $n$ is large
  • Calculus facts: necessary and sufficient conditions
  • Rely on gradients (if possible)

• Sometimes, some components of $x$ may be integers
  • Can we do better than brute force?

• Simpler cases
  • Differentiable functions
  • Linear, convex, quasiconvex
  • Unconstrained problems
Nonlinear Optimization

minimize \( f(x) \)
subject to \( g_i(x) \leq 0, i = 1, \ldots, n \)
\( h_j(x) = 0, j = 1, \ldots, m \)

- Nonlinear optimization:
  - Decision variable is \( x \in \mathbb{R}^n \)
Optimal Control

Nonlinear optimization:
- Decision variable is $x \in \mathbb{R}^n$

Optimal control:
- Decision variable is a function $u(\cdot)$

Cost functional, $J(x(\cdot), u(\cdot))$
- Final cost
- Running cost

Dynamic model
- Additional constraints
  - Eg. actuation limits

Dynamic model:
- $x(t) \in \mathbb{R}^n, u(t) \in \mathbb{R}^m, x(0) = x_0$
- $\dot{x}(t) = f(x(t), u(t))$
- $g(x(t), u(t)) \geq 0$

Minimize
- $l(x(t_f), t_f) + \int_0^{t_f} c(x(t), u(t), t) dt$

Graphical representation:
- Path of $x(t)$ and $u(t)$ over time
Robotic Safety

• Verification methods

  Assumptions → Control policy → Prove safety

• Considers all possible system behaviours, given assumptions

• Can be written as an optimal control problem
Reachability Analysis

- Model of robot
- Unsafe region

Reachable set

Unsafe region

Optimal control policy to avoid danger

Reachable set (States leading to danger)
Machine Learning

• Application of nonlinear optimization
  • Takes advantage of available data

• Supervised learning
  • Regression
  • Classification
Machine Learning

• Application of nonlinear optimization
  • Takes advantage of available data

• Supervised learning
  • Regression
  • Classification

• Unsupervised learning
  • Clustering
  • Reinforcement learning
Machine Learning

- Very scalable with additional data
- Requires a lot of data
Machine Learning

• Very scalable with additional data
• Requires a lot of data

• Computer vision
• Natural language processing
• Game playing
• Simulated robotics

• Physical robotics?
Localization and Mapping

• Localization
  • Given a map, figure out where the robot is (with respect to the map) using sensor information
  • Continuously do this while moving around in the environment
Localization and Mapping

• Localization
  • Given a map, figure out where the robot is (with respect to the map) using sensor information
  • Continuously do this while moving around in the environment

• Simultaneous localization and mapping
  • Figure out the map and localize at the same time
Localization and Mapping

• Localization
  • Given a map, figure out where the robot is (with respect to the map) using sensor information
  • Continuously do this while moving around in the environment

• Simultaneous localization and mapping
  • Figure out the map and localize at the same time

• Probabilistic models
  • of how the robot moves
  • of how the robot senses the environment
Sample of MARS Research

• [https://sfumars.com](https://sfumars.com)

• Control algorithms
• Computational complexity
• Reinforcement learning
• Human intent inference

• Theory
• Computation
• Experiments
Safety: A Crucial Perspective in Automation
Safety: A Crucial Perspective in Automation
Challenges in Safety-Critical Systems

• Account for all possible system behaviours
• Complex systems
• Complex environment
  • Weather conditions
  • Other robots
Reachability Analysis

- Model of robot
- Unsafe region

Reachable set (States leading to danger)
Optimal control policy to avoid danger
Human-controlled intruder

Autonomous quadrotors
Main Challenge:
Exponential Computational Complexity with DP

Computation time and RAM usage

1D: < 0.1s
negligible RAM

2D: seconds
negligible RAM

3D: minutes
tens of megabytes

4D: hours
hundreds of megabytes

5D: days
megabytes

6D: intractable!

\[ O(N^d) \] time and space complexity!
Research Directions

Dimensionality reduction
Self-Contained Subsystems

- Motivating example: Dubins Car

\[
\begin{align*}
\dot{x} &= v \cos \theta \\
\dot{y} &= v \sin \theta \\
\dot{\theta} &= \omega
\end{align*}
\]

- Subsystems are coupled through state and control

- Many systems have states that are not directly coupled to each other
  - Most common in vehicle dynamics
Research Directions

**Dimensionality reduction**

**Parallel computing**
Research Directions

Dimensionality reduction

Parallel computing

Perception systems
Research Directions

Human intent understanding

Campbell et al.
Proactive Human Intent Understanding

Is the red car
• A pursuer,
• Or a benign vehicle?

Robot car (green) *proactively* changes lanes to determine intent
Multi-Modal Human Intent Understanding

Motion
Emotion
Engagement
Audio
Research Directions

Human intent understanding

Robotic learning

Campbell et al.

Global Robots Ltd.
Curriculum Reinforcement Learning

Without curriculum

Distance-based reward shaping

Reachability-based curriculum

Random curriculum
Curriculum Reinforcement Learning

Task performance

- Reachability-based curriculum
- Random curriculum
- Without curriculum
- Distance-based reward shaping

Curriculum performance