CS 4649/7649
Robot Intelligence: Planning

Linear Optimal Control

Sungmoon Joo

School of Interactive Computing
College of Computing
Georgia Institute of Technology

*Slides based in part on Dr. Mike Stilman's slides

Linear Quadratic Regulator (LQR)

- Remember Gains: K_p and K_d
- LQR is an automated method for choosing **OPTIMAL** gains
- Optimal with respect to what?
  - Some(quadratic) function of state (e.g. Minimize distance to goal)
  - Some(quadratic) function of control input (e.g. Minimize torque)
  - **A relative weighting of both**

LQR Basis

- Linear Dynamic System:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t)$$

$$\begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{b}{J} \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{J} \end{bmatrix} \tau$$

$$J\ddot{\theta} + b\dot{\theta} = \tau$$

$$\ddot{\theta} = -\frac{b}{J}\dot{\theta} + \frac{1}{J}\tau$$

LQR Basis

- Linear Dynamic System: $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t)$
- Quadratic Performance Criterion: $C(\mathbf{x}, \mathbf{u}) = \frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} + \frac{1}{2}\mathbf{u}^T\mathbf{R}\mathbf{u}$

Symmetric Positive Definite (Typically Diagonal) Matrices

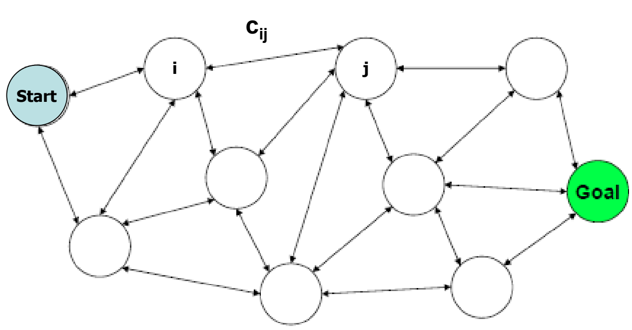
$$\mathbf{Q} = \begin{bmatrix} q_0 & & & \\ & q_1 & & \\ & & \cdots & \\ & & & q_n \end{bmatrix} \quad \mathbf{R} = \begin{bmatrix} r_0 & & & \\ & r_1 & & \\ & & \cdots & \\ & & & r_n \end{bmatrix}$$

LQR Basis

- Linear Dynamic System: $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t)$
- Quadratic Performance Criterion: $C(\mathbf{x}, \mathbf{u}) = \frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} + \frac{1}{2}\mathbf{u}^T\mathbf{R}\mathbf{u}$
- Gain Matrix: $\mathbf{K} = \mathbf{R}^{-1}\mathbf{B}^T\mathbf{P}$ $\quad \mathbf{u} = \mathbf{K}\mathbf{x} = \mathbf{R}^{-1}\mathbf{B}^T\mathbf{P}\mathbf{x}$
- Algebraic Riccati Equation: **P is a solution of ARE**

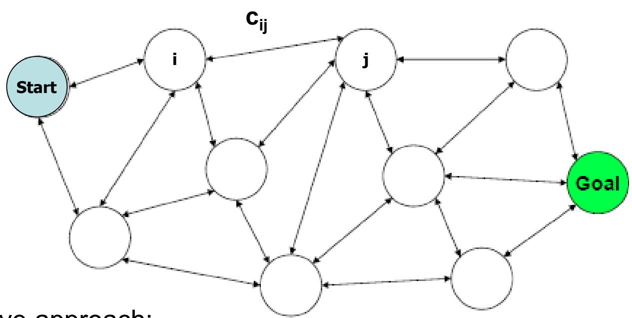
$$0 = \mathbf{Q} - \mathbf{P}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{P} + \mathbf{P}\mathbf{A} + \mathbf{A}^T\mathbf{P}$$

Detour: Shortest Path Search



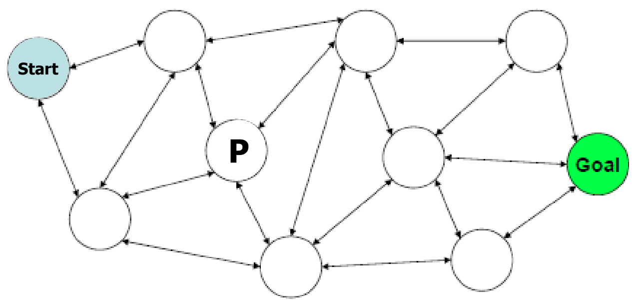
Consider a digraph G = (V,A) with nonnegative arc distance c_{ij}

Detour: Shortest Path Search



Naïve approach:
- Find all paths from s(start) to g(goal)
- Evaluate the length of each s ~ g path
- Find the minimum length
- The number of s ~ g paths increases exponentially. (Infeasible for large graph)

Detour: Shortest Path Search

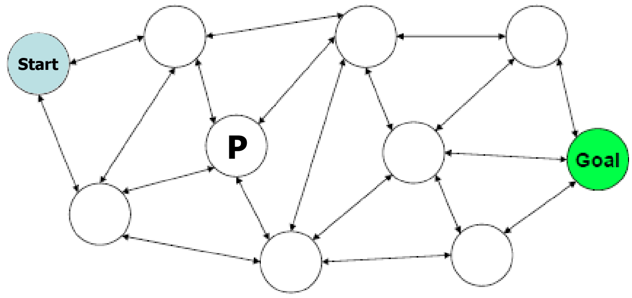


Observation:
- If the s ~ g shortest path passes by node p, the subpath (s, p) and (p, g) are shortest path from s to p and from p to g, respectively

$$d^*(g) = \text{length of shortest path from s to g}$$

$$d^*(g) = \min_{i \in V - g} \{d^*(i) + c_{ig}\}$$

Detour: Shortest Path Search



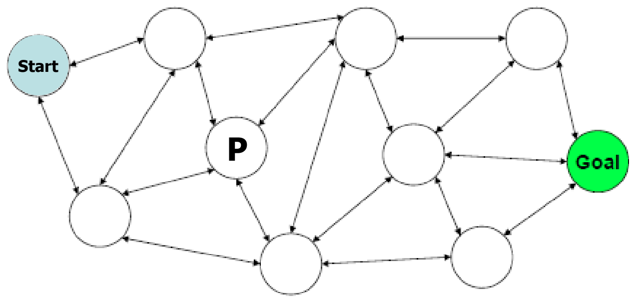
Forward Dynamic Programming

- Let j be a node at stage k, $D_k(j)$ be the shortest distance from s to j

$$D_k(j) = \min\{D_{k-1}(i) + c_{ij} | i \in \text{stage k -1}\}$$

$$k = 0, 1, \ldots$$

Detour: Shortest Path Search



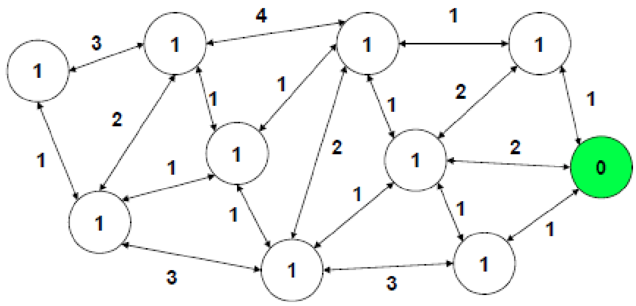
Backward Dynamic Programming

- Let $D_k(j)$ be the shortest distance from node j at stage k to g

$$D_k(j) = \min\{D_{k+1}(i) + c_{ji} | i \in \text{stage k +1}\}$$

$$k = N, N-1, \ldots$$

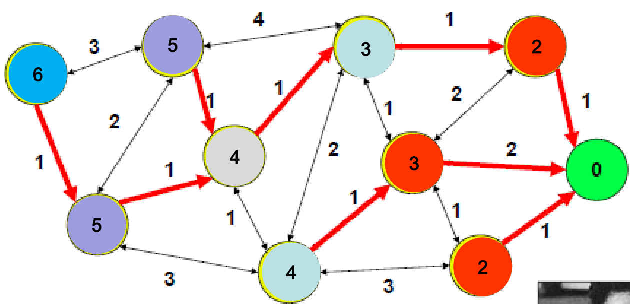
What is the 'Value' of a State?



MDP – Accumulated rewards, Want to maximize

Shortest distance problem – Cost-to-Go, Want to minimize

Value Iteration is Dynamic Programming



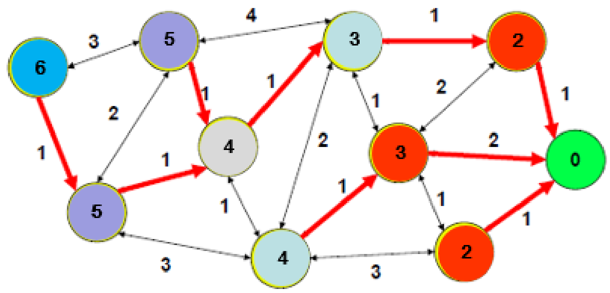
1953

What would Bellman do?
Dynamic Programming

$$J^*(\mathbf{x}_t, \mathbf{u}_t) = \min_u [C(\mathbf{x}_t, \mathbf{u}_t) + J^*(\mathbf{x}_{t+1}, \mathbf{u}_{t+1})]$$

Optimal Policy is obtained!!

Dijkstra '59



Dijkstra's Algorithm is a Special Case of Dynamic Programming!

: DP + greedy search with Cost-to-Come

: a graph search algorithm that solves the single-source shortest path problem for a graph with non-negative edge path costs

Dijkstra vs A

Back to LQR

- Linear Dynamic System: $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t)$
- Quadratic Performance Criterion: $C(\mathbf{x}, \mathbf{u}) = \frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} + \frac{1}{2}\mathbf{u}^T\mathbf{R}\mathbf{u}$
- Gain Matrix: $\mathbf{K} = \mathbf{R}^{-1}\mathbf{B}^T\mathbf{P}$ $\quad \mathbf{u} = \mathbf{K}\mathbf{x} = \mathbf{R}^{-1}\mathbf{B}^T\mathbf{P}\mathbf{x}$
- Algebraic Riccati Equation:

$$0 = \mathbf{Q} - \mathbf{P}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{P} + \mathbf{P}\mathbf{A} + \mathbf{A}^T\mathbf{P}$$

Define a Value Function

- **State Transitions:** $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t)$
- **Costs:** $C(\mathbf{x}_\tau, \mathbf{u}_\tau) = \frac{1}{2}\mathbf{x}^T(\tau)\mathbf{Q}\mathbf{x}(\tau) + \frac{1}{2}\mathbf{u}^T(\tau)\mathbf{R}\mathbf{u}(\tau)$
- **Minimize Cost For All Future Time**

$$J = \int_0^\infty C(x_\tau, u_\tau) d\tau$$

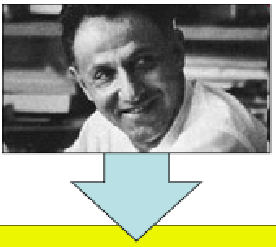
- **Minimize Cost After Time = t**

$$J(\mathbf{x}_t, \mathbf{u}_t) = \int_t^\infty C(x_\tau, u_\tau) d\tau$$

What would Bellman do?

Minimize Cost After Time = t

$$J^*(\mathbf{x}_t, \mathbf{u}_t) = \min_u \int_t^\infty C(x_\tau, u_\tau) d\tau$$



Critical Step!

$$J^*(\mathbf{x}_t, \mathbf{u}_t) = \min_u [C(\mathbf{x}_t, \mathbf{u}_t)\delta t + J^*(\mathbf{x}_{t+\delta t}, \mathbf{u}_{t+\delta t})]$$

Substitute for Value in Right Hand Side

$$J(\mathbf{x}_t, \mathbf{u}_t) = C(\mathbf{x}_t, \mathbf{u}_t)\delta t + J(\mathbf{x}_{t+\delta t}, \mathbf{u}_{t+\delta t})]$$

Assume V is smooth and does not depend explicitly on t:

$$\begin{aligned} J(\mathbf{x}_{t+\delta t}, \mathbf{u}_{t+\delta t}) &= J(\mathbf{x}_t, \mathbf{u}_t) + \frac{\delta J}{\delta x}\frac{\delta x}{\delta t}\delta t \\ &= J(\mathbf{x}_t, \mathbf{u}_t) + \frac{\delta J}{\delta x}(\mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t)\delta t \end{aligned}$$

$$J(\mathbf{x}_t, \mathbf{u}_t) = C(\mathbf{x}_t, \mathbf{u}_t)\delta t + J(\mathbf{x}_t, \mathbf{u}_t) + \frac{\delta J}{\delta x}(\mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t)\delta t$$

One More Fact/Assumption

Value function has the following form (P is symmetric, positive definite)

$$J(\mathbf{x}_t, \mathbf{u}_t) = \tfrac{1}{2}\mathbf{x}^T\mathbf{P}\mathbf{x}$$

One More Fact/Assumption

Value function has the following form (P is symmetric, positive definite)

$$J(\mathbf{x}_t, \mathbf{u}_t) = \frac{1}{2}\mathbf{x}^T\mathbf{P}\mathbf{x}$$

Why? It is an infinite sum of symmetric, positive definite matrices.

Linear Algebra Theorems:
- Sum of symmetric matrices = symmetric
- Sum of positive definite matrices = positive definite

Derivative with respect to x

Value function has the following form (P is symmetric, positive definite)

$$J(\mathbf{x}_t, \mathbf{u}_t) = \frac{1}{2}\mathbf{x}^T\mathbf{P}\mathbf{x}$$

$$\frac{\delta J}{\delta \mathbf{x}} = \mathbf{x}^T\mathbf{P}$$

Putting it All Together

$$J(\mathbf{x}_t, \mathbf{u}_t) = C(\mathbf{x}_t, \mathbf{u}_t)\delta t + J(\mathbf{x}_t, \mathbf{u}_t) + \frac{\delta J}{\delta x}(\mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t)\delta t$$

$$C(\mathbf{x}, \mathbf{u}) = \tfrac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} + \tfrac{1}{2}\mathbf{u}^T\mathbf{R}\mathbf{u}$$

$$\frac{\delta J}{\delta x} = \mathbf{x}^T\mathbf{P}$$

$$\begin{aligned} J(\mathbf{x}_t, \mathbf{u}_t) &= \tfrac{1}{2}\mathbf{x}_t^T\mathbf{Q}\mathbf{x}_t\delta t + \tfrac{1}{2}\mathbf{u}_t^T\mathbf{R}\mathbf{u}_t\delta t + \\ &\quad J(\mathbf{x}_t, \mathbf{u}_t) + \mathbf{x}_t^T\mathbf{P}(\mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t)\delta t \end{aligned}$$

How do we minimize this with respect to u?

$$\begin{aligned} J(\mathbf{x}_t, \mathbf{u}_t) &= \tfrac{1}{2}\mathbf{x}_t^T\mathbf{Q}\mathbf{x}_t\delta t + \tfrac{1}{2}\mathbf{u}_t^T\mathbf{R}\mathbf{u}_t\delta t + \\ &\quad J(\mathbf{x}_t, \mathbf{u}_t) + \mathbf{x}_t^T\mathbf{P}(\mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t)\delta t \end{aligned}$$

$$\frac{\delta J}{\delta \mathbf{u}} = 0$$

$$\cancel{\frac{\delta J}{\delta \mathbf{u}}} = 0 + \mathbf{u}_t^T\mathbf{R}\cancel{\delta t} + \cancel{\frac{\delta J}{\delta \mathbf{u}}} + \mathbf{x}_t^T\mathbf{P}\mathbf{B}\cancel{\delta t}$$

$$\mathbf{u}_t^T\mathbf{R} + \mathbf{x}_t^T\mathbf{P}\mathbf{B} = 0$$

Therefore Gain Matrix: K = ?

$$\mathrm{u}_t^T\mathrm{R} + \mathrm{x}_t^T\mathrm{PB} = 0$$

Nice Exercise # 1

$$\mathrm{u}_t = \mathrm{K}\mathrm{x}_t \quad \text{so} \quad \mathrm{K} = -\mathrm{R}^{-1}\mathrm{B}^T\mathrm{P}$$

The Rest is Simple Linear Algebra

Recall Bellman's Equation:

$$\begin{aligned} J(\mathrm{x}_t, \mathrm{u}_t) &= \tfrac{1}{2}\mathrm{x}_t^T\mathrm{Q}\mathrm{x}_t\delta t + \tfrac{1}{2}\mathrm{u}_t^T\mathrm{R}\mathrm{u}_t\delta t + \\ & J(\mathrm{x}_t, \mathrm{u}_t) + \mathrm{x}_t^T\mathrm{P}(\mathrm{A}\mathrm{x}_t + \mathrm{B}\mathrm{u}_t)\delta t \end{aligned}$$

Recall our solution for u_t:

$$\mathrm{u}_t = -\mathrm{R}^{-1}\mathrm{B}^T\mathrm{P}\mathrm{x}_t$$

Cancel some terms and replace every instance of u_t with the control of x_t

---

$$\begin{aligned} 0 &= \tfrac{1}{2}\mathrm{x}_t^T\mathrm{Q}\mathrm{x}_t + \tfrac{1}{2}(-\mathrm{R}^{-1}\mathrm{B}^T\mathrm{P}\mathrm{x}_t)^T\mathrm{R}(-\mathrm{R}^{-1}\mathrm{B}^T\mathrm{P}\mathrm{x}_t) \\ & +\mathrm{x}_t^T\mathrm{P}(\mathrm{A}\mathrm{x}_t + \mathrm{B}(-\mathrm{R}^{-1}\mathrm{B}^T\mathrm{P}\mathrm{x}_t)) \end{aligned}$$

The Rest is Simple Linear Algebra

$$0 = \tfrac{1}{2}x_t^T Q x_t + \tfrac{1}{2}(-R^{-1}B^T P x_t)^T R(-R^{-1}B^T P x_t) + x_t^T P A x_t - x_t^T B R^{-1} B^T P x_t$$

Exercise 2

$$0 = \tfrac{1}{2}x_t^T Q x_t - \tfrac{1}{2}x_t^T P B R^{-1} B^T P x_t + x_t^T P A x_t$$

The Rest is Simple Linear Algebra

$$0 = \tfrac{1}{2}x_t^T Q x_t - \tfrac{1}{2}x_t^T P B R^{-1} B^T P x_t + x_t^T P A x_t$$

Exercise 3

$$\boxed{0 = Q - PBR^{-1}B^T P + PA + A^T P}$$

There's that Riccati Equation!

- Linear Dynamic System: $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t)$
- Quadratic Performance Criterion: $C(\mathbf{x}, \mathbf{u}) = \frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} + \frac{1}{2}\mathbf{u}^T\mathbf{R}\mathbf{u}$
- Gain Matrix: $\mathbf{K} = \mathbf{R}^{-1}\mathbf{B}^T\mathbf{P}$ $\quad \mathbf{u} = \mathbf{K}\mathbf{x} = \mathbf{R}^{-1}\mathbf{B}^T\mathbf{P}\mathbf{x}$
- Algebraic Riccati Equation:

$$0 = \mathbf{Q} - \mathbf{P}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{P} + \mathbf{P}\mathbf{A} + \mathbf{A}^T\mathbf{P}$$

What did we just show?

LQR is a Special Case of Dynamic Programming!

- **Given a Linear System**
- **Quadratic Cost Critereon**