Special structures

What makes some sequential decision-making problems easy?

Cathy Wu

6.7920 Reinforcement Learning: Foundations and Methods

Readings

1. 6.231 Sp22 Lecture 3 notes, Section 2 [N3 §2]
2. DPOC vol 1, 3.1 (LQR), 3.3-3.4
3. (Optional) MIT Underactuated Robotics, Chapter 10 Trajectory Optimization [link]

Outline

1. Recap & roadmap
2. Template for structural DP arguments
3. Example: optimal stopping
4. Linear quadratic control (LQR)

Outline

1. **Recap & roadmap**
2. Template for structural DP arguments
3. Example: optimal stopping
4. Linear quadratic control (LQR)

So far: sequential decision making is hard

"Roadmap"

This time: What makes ***some*** sequential decision problems easy?

Next time [3x]: Why is there still ***hope*** of solving sequential decision problems?
(general solutions for ***small-state*** problems)

Next next time [8x]: Why is there still hope of solving ***large-state*** problems?

Outline

1. Recap & roadmap
2. **Template for structural DP arguments**
   a. Convexity, monotonicity
3. Example: optimal stopping
4. Linear quadratic control (LQR)

Template for Structural DP Arguments

1. Recognize that the **terminal** reward/cost-to-go function V_T^* has a **nice property** (**base case** in induction proof).
   - Example: convexity or monotonicity
2. Then, argue that this property implies that the **policy** $\boldsymbol{\pi^*_{T-1}}$ has some nice structure.
   - Example: a threshold policy is optimal
3. Extend this with an **induction step**: we show that if a reward-to-go function V satisfies the property, then the “next” reward-to-go function:

$$V^-(x) = \max_{a \in A(s)} \mathbb{E}\left[g(s,a,w) + V\big(f(s,a,w)\big)\right]$$

that is generated by a step of the DP algorithm will also satisfy this property.

Operations that Preserve Convexity

- Comes in handy in showing the convexity of reward-to-go functions.
- Non-negative weighted sums:
  - If $f_1, \dots, f_m : \mathcal{D} \to \mathbb{R}$ are convex and $w_1, \dots, w_m \geq 0$, then $w_1 f_1 + \ \dots + w_m f_m$ is convex.
  - For some $f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, the expectation $g : \mathcal{X} \to \mathbb{R}$ defined as
    $$g(x) = \int f(x, y) w(y) dy$$
    is convex if $w(y) \geq 0$ and the mapping $x \ \mapsto f(x, y)$ is convex for all $y \in \mathcal{Y}$.
- Composition with an affine map:
  - $g(x) \ = \ f(Ax \ + \ b)$ is convex if f is convex.
- Point-wise supremum:
  - $g(x) = \sup_{y \in \mathcal{Y}} f(x, y)$ is convex if $x \mapsto f(x, y)$ is convex for all $y \in \mathcal{Y}$.

Further reading: For a detailed treatment, please refer to the book Convex Optimization by Boyd and Vandenberghe.

Outline

1. Recap & roadmap
2. Template for structural DP arguments
3. **Example: optimal stopping**
4. Linear quadratic control (LQR)

Asset Selling With Irrevocable Decisions

- Discrete time setting, $t = \{0, 1, \dots, T-1\}$
- Problem: you have an asset to sell by time T.
  - At each epoch
    - You receive an offer w_t drawn independently from some distribution W (bounded).
    - You must either accept the offer and invest the money at a fixed interest rate $r > 0$ or reject and wait for the next offer.
  - Goal: maximize the expected final revenue.
- Notes:
  - Continuous state problem!
  - Assume that a rejected offer is lost.

Asset Selling With Irrevocable Decisions

- State s_t

$$s_{t+1} = \begin{cases} \text{sold} & \text{if } A_t = \text{Accept or } s_t = \text{sold} \\ w_t & o.w. \end{cases}$$

$\forall\{t = 0, \dots T - 1\}$.
  - Set $s_0 = 0$ as a dummy variable.
  - The state space is $S \subset \mathbb{R} \cup \{\text{sold}\}$.
- Action space:

$$A_t(s_t) = \begin{cases} \emptyset & \text{if } s_t = \text{sold} \\ \{\text{Accept, Reject}\} & o.w. \end{cases}$$

- The revenue for each period is defined as:

$$g_t(s_t, u_t, w_t) = \begin{cases} 0 & \text{if } u_t \neq \text{Accept} \\ (1+r)^{T-t} s_t & \text{if } u_t = \text{Accept} \end{cases}$$

with the revenue for the final state being:

$$g_T(s_T) = \begin{cases} 0 & \text{if } s_T = \text{sold} \\ s_T & o.w. \end{cases}$$

DP recursion & optimal policy

- Following the DP algorithm described in the previous section, set $V_T^*(s) = g_T(s)$.
  For $t = \{T-1, T-2, \dots, 0\}$, set:

$$V_t^*(s) = \begin{cases} \max\{(1+r)^{T-t}s, \mathbb{E}[V_{t+1}^*(w_t)]\} & \text{if } s \neq \text{sold} \\ 0 & \text{if } s = \text{sold} \end{cases}$$

It is the maximum of the termination value $(1+r)^{T-t}s$ and the continuation value $\mathbb{E}[V_{t+1}^*(w_t)]$

- Given the structure of the value-to-go functions, $V_t^*(s)$,
  the optimal policy can be easily computed as the following threshold policy:

$$\pi_t^*(s_t) | \, (s_t \neq \text{sold}) = \begin{cases} \text{Accept} & \text{if } s_t \geq \alpha_t \\ \text{Reject} & \text{if } s_t \leq \alpha_t \end{cases}$$

  where the thresholds, $\alpha_t = \frac{\mathbb{E}[V_{t+1}^*(w_t)]}{(1+r)^{T-t}}$, depend on time t.

- We remark that α_t's obey their own recursion. Since we must accept the last offer, $\alpha_T = -\infty$.
  For $t = \{T-1, \dots, 0\}$

$$\alpha_t = \frac{1}{1+r}\mathbb{E}[\max\{w_t, \alpha_{t+1}\}]$$

Proof: By induction.

- **Remark (infinite horizon version)**: With i.i.d. offers, the optimal policy is stationary and the optimal threshold α^* is the solution to the fixed point equation:

$$\alpha = \frac{1}{1+r}\mathbb{E}[\max\{w, \alpha\}]$$

In plain English: Accept the offer if it's better to invest now than wait for a slightly better offer in the future, which loses out on factor(s) of the interest rate.

Asset Selling With Offers Retained

- Now consider the setting:
  - The offers $w_0, \ldots, w_{T-1}$ are i.i.d., non-negative, bounded.
  - The rejected offers are not lost. At any period t, we can choose the highest offer received so far.
- To accommodate this setting, we define the state such that

$$s_{t+1} = \begin{cases} \text{sold} & \text{if } A_t = \text{Accept or } s_t = \text{sold} \\ \max\{s_t, w_t\} & o.w. \end{cases}$$

$\forall t = \{0, \ldots, T-1\}$.

- The action space and functions g_t's stay the same.

Optimal policy

Proposition

An optimal policy for asset selling with offers retained is a stationary policy $\pi^* = (\mu^*, \mu^*, \dots, \mu^*)$, where for $s \neq$ sold,

$$\pi_t^*(s) = \begin{cases} \text{Accept} & \text{if } s \geq \dfrac{1}{1+r} \mathbb{E}_w[\max\{s, w\}] \\ \text{Reject} & o.w. \end{cases}$$

Proof (Proposition)

1. **Monotonicity**: For $s \neq$ sold, we can set $V_T^*(s) = s$. For $t = T - 1$ and $s \neq$ sold,

$$V_{T-1}^*(s) = \max\{(1+r)s, \mathbb{E}[\max\{w_{T-1}, s\}]\}$$
$$\geq (1+r)s$$
$$= (1+r)V_T^*(s)$$

2. By **induction**, assume that $V_{t+1}^*(s) \geq (1+r)V_{t+2}^*(s)$. Then

$$V_t^*(s) = \max\{(1+r)^{T-t}s, \mathbb{E}[V_{t+1}^*(\max\{s, w_t\})]\}$$
$$\geq \max\{(1+r)^{T-t}s, (1+r)\mathbb{E}[V_{t+2}^*(\max\{s, w_t\})]\}$$
$$= (1+r)\max\{(1+r)^{T-(t+1)}s, \mathbb{E}[V_{t+2}^*(\max\{s, w_t\})]\}$$
$$= (1+r)V_{t+1}^*(s)$$

3. **Optimal stopping set**: $S_t^* := \{s \mid s \geq \alpha_t \coloneqq (1+r)^{-(T-t)}\mathbb{E}[V_{t+1}^*(\max\{s, w_t\})]\}$
4. **Convergence**: thresholds α_t converge (backwards) because:
   - Thresholds α_t are monotonically increasing (backwards)

$$\alpha_t \geq \alpha_{t+1} \rightarrow S_t^* \subseteq S_{t+1}^*$$

   - Thresholds α_t are bounded above (bounded offers)
   - Thresholds $\alpha_t \rightarrow \frac{1}{1+r}\mathbb{E}_w[\max\{s, w\}]$, since 1) $S_t^* \supseteq S_{t+1}^*$, 2) $\alpha_{T-1} = \frac{1}{1+r}\mathbb{E}_w[\max\{s, w\}]$

Outline

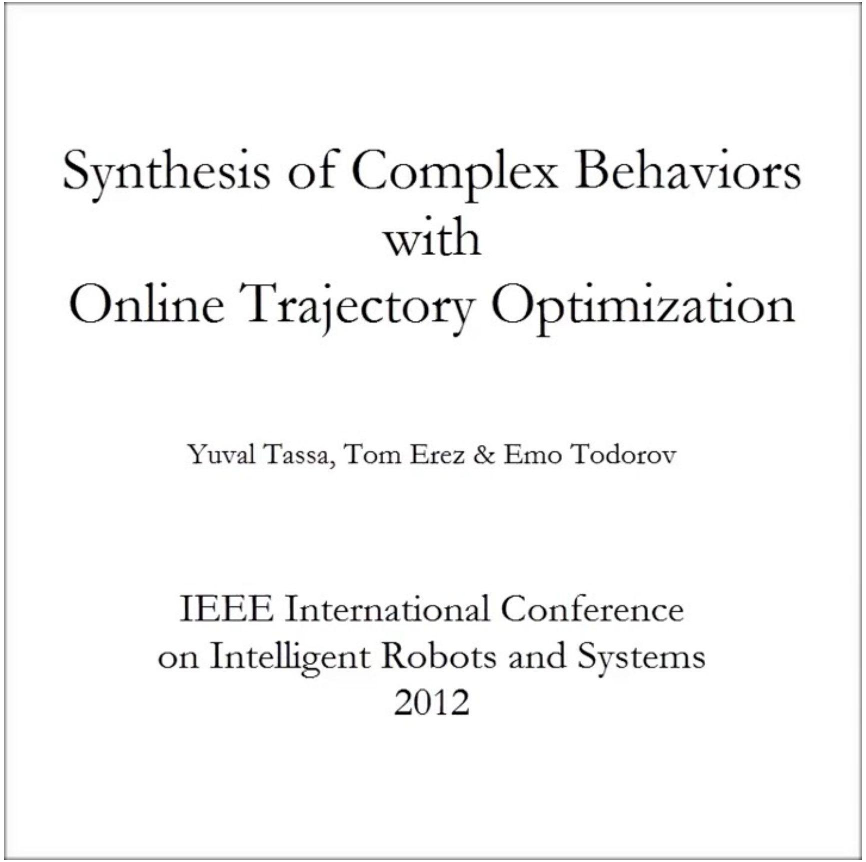
1. Recap & roadmap
2. Template for structural DP arguments
3. Example: optimal stopping
4. **Linear quadratic control (LQR)**
   a. Finite horizon LQR
   b. Linear quadratic Gaussian & Certainty equivalence
   c. Infinite horizon LQR & Algebraic Riccati Equations

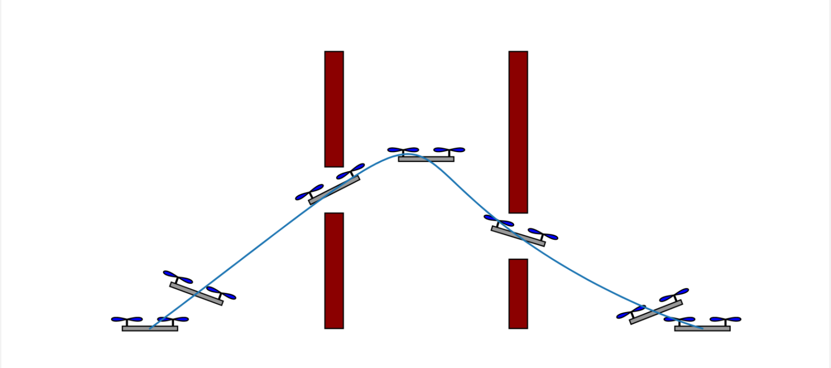
Notation "break"

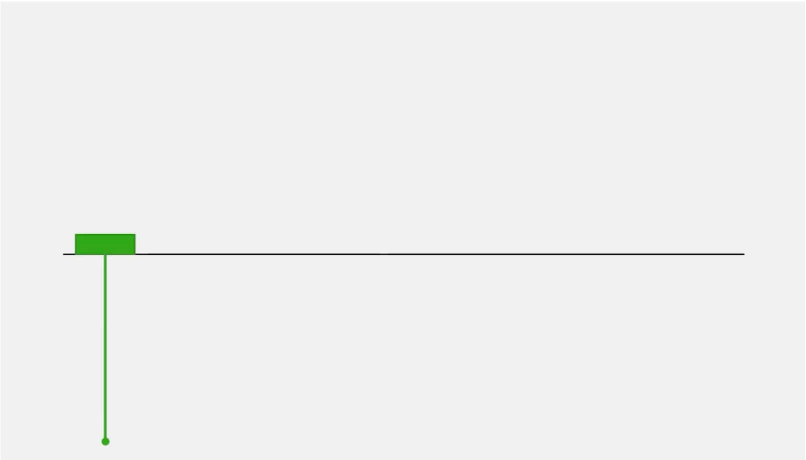
In the following section, and in deference to the rich tradition in control theory, we will be using standard control theory notation

- x and u, in place of s and a, to denote state and the control
- $c(x,u)$ in place of $r(s,a)$, to denote immediate cost or reward
- $r(s,a) = -c(x,u)$

Trajectory optimization




[Russ Tedrake]

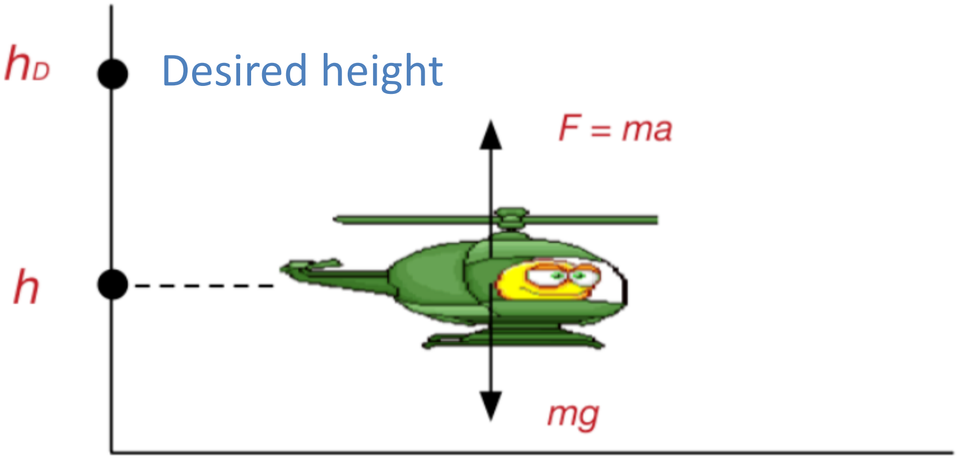

[Matthew Kelly]

Linear quadratic control

Assumptions: deterministic, finite horizon, discrete time

$\text{eig}(A) \leq 1 \rightarrow$ stable

Further reading:
Chen, Chi-Tsong. Linear system theory and design. 1984.



$$\underbrace{\begin{bmatrix} h_{t+1} \\ v_{t+1} \end{bmatrix}}_{x_{t+1}} = \underbrace{\begin{bmatrix} 1 & \Delta \\ 0 & 1 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} h_t \\ v_t \end{bmatrix}}_{x_t} + \underbrace{\begin{bmatrix} \frac{1}{2}\Delta^2 \\ \Delta \end{bmatrix}}_{B} \underbrace{(\alpha_t - g)}_{u_t}$$

State space form

$$x_{t+1} = f(x, u_t) = Ax_t + Bu_t$$

Linear time-invariant (LTI) system

The dynamics (discrete form) are governed by the equations of motion is:

$$h_{t+1} = h_t + \Delta v_t + \frac{1}{2}\Delta^2(\alpha_t - g)$$
$$v_{t+1} = v_t + \Delta(\alpha_t - g)$$

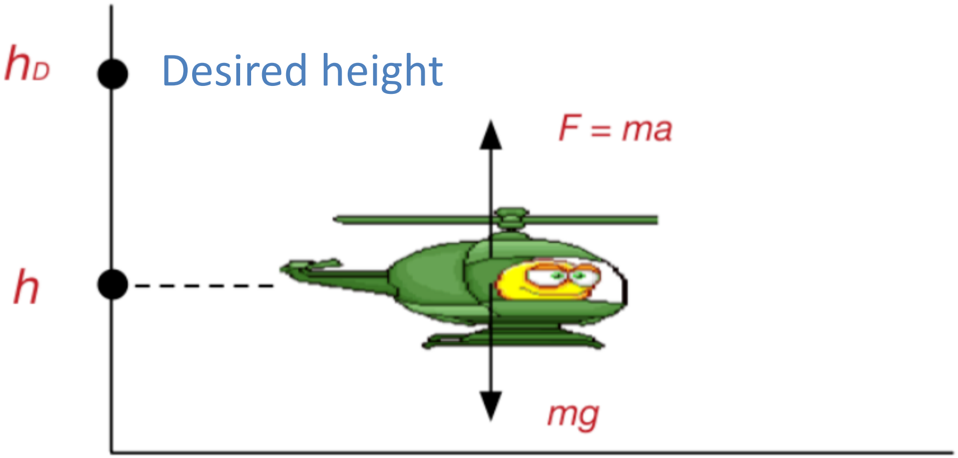
where Δ = time step (sec)

w.l.o.g.

$$x_t := \begin{bmatrix} h_t \\ v_t \end{bmatrix} - x_D$$
$$x_D := \begin{bmatrix} h_D \\ 0 \end{bmatrix}$$

Linear quadratic control

Assumptions: deterministic, finite horizon, discrete time



$$\underbrace{\begin{bmatrix} h_{t+1} \\ v_{t+1} \end{bmatrix}}_{x_{t+1}} = \underbrace{\begin{bmatrix} 1 & \Delta \\ 0 & 1 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} h_t \\ v_t \end{bmatrix}}_{x_t} + \underbrace{\begin{bmatrix} \frac{1}{2}\Delta^2 \\ \Delta \end{bmatrix}}_{B} \underbrace{(\alpha_t - g)}_{u_t}$$

(excuse the max/min…)

Goal: minimize

$$V(x_0; u) = \sum_{t=0}^{T-1} x_t^T Q x_t + u_t R u_t$$

$$Q \succcurlyeq 0, R \succ 0$$

Finite horizon LQR:

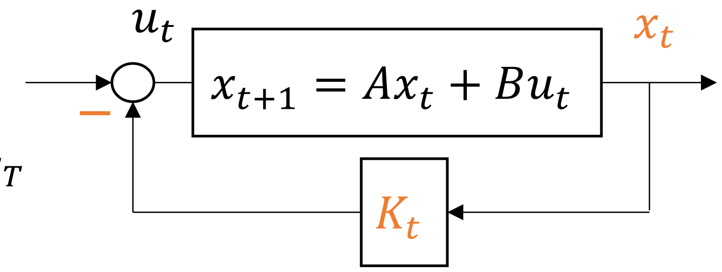
$$u = \min_{u_0, \ldots, u_{T-1}} V(x_0; u) = \sum_{t=0}^{T-1} x_t^T Q x_t + u_t^T R u_t + \underbrace{x_T^T Q_T x_T}_{\text{Terminal cost}}$$

$$\text{s.t.} \quad x_{t+1} = A x_t + B u_t, \qquad t = 0, 1, \ldots, T-1$$

Adapted from Kevin Jamieson

Linear quadratic control

$\mathrm{eig}(A - BK_t) \leq 1 \rightarrow$ stable



Finite horizon LQR

$$u = \min_{u_0,\ldots,u_{T-1}} V(x_0; u) = \sum_{t=0}^{T-1} x_t^T Q x_t + u_t^T R u_t + x_T^T Q_T x_T$$

$$\text{s.t.} \quad x_{t+1} = Ax_t + Bu_t, \qquad t = 0, 1, \ldots, T-1$$

Optimal control law is a linear feedback controller: $x_{t+1} = Ax_t + Bu_t = (A - BK_t)x_t$

Theorem (Finite horizon LQR)

The optimal cost-to-go and optimal control at time t are given by:

$$V^*(x_t) = x_t^T P_t x_t$$
$$u_t^* = -K_t x_t$$

where

$$P_t = Q + K_t^T R K_t + (A - BK_t)^T P_{t+1} (A - BK_t), \qquad P_T = Q_T$$
$$K_t = (R + B^T P_{t+1} B)^{-1} B^T P_{t+1} A, \qquad t \in \{0, \ldots, T-1\}$$

Proof (induction)

Finite horizon LQR:

$$u = \min_{u_0,\dots,u_{T-1}} V(x_0; u) = \sum_{t=0}^{T-1} x_t^T Q x_t + u_t^T R u_t + x_T^T Q_T x_T$$

s.t. $x_{t+1} = Ax_t + Bu_t, \quad t = 0, 1, \dots, T-1$

Theorem (Finite horizon LQR)

The optimal cost-to-go and optimal control at time t are given by:

$$V^*(x_t) = x_t^T P_t x_t$$
$$u_t^* = -K_t x_t$$

where

$P_t = Q + K_t^T R K_t + (A - BK_t)^T P_{t+1} (A - BK_k), \quad P_T = Q_T$

$K_t = (R + B^T P_{t+1} B)^{-1} B^T P_{t+1} A, \quad t \in \{0, \dots, T-1\}$

- **Base case (stage T)**:

$$V^*(x_t) = x_t^T P_t x_t$$
$$\Rightarrow P_T = Q_T$$

- **Special structure**: $V^*(x_T) = x_T^T Q_T x_T$ is convex.
- **Induction**: assume P_t holds & $V^*(x_t)$ convex, show for t-1

Recall: $r_t(x_t, u_t) \coloneqq x_t^T Q x_t + u_t R u_t$

$$V^*(x_{t-1}) = \min_{u_{t-1}} [x_{t-1}^T Q x_{t-1} + u_{t-1}^T R u_{t-1} + V^*(x_t)] \quad \text{(principle of optimality)}$$

$$= \min_{u_{t-1}} [x_{t-1}^T Q x_{t-1} + u_{t-1}^T R u_{t-1} + x_t^T P_t x_t] \quad \text{(induction hypothesis)}$$

$$= \min_{u_{t-1}} [x_{t-1}^T Q x_{t-1} + u_{t-1}^T R u_{t-1} + (Ax_{t-1} + Bu_{t-1})^T P_t (Ax_{t-1} + Bu_{t-1})] \quad \text{(system equations)}$$

$$\nabla_{u_{t-1}} V^*(x_{t-1}) = 2u_{t-1}^T R + 2(Ax_{t-1} + Bu_{t-1})^T P_t B = 0 \quad \text{(convexity)}$$

$$u_{t-1}^* = (R + B^T P_t B)^{-1} B^T P_t A x_{t-1} = -K_{t-1} x_{t-1} \quad (R \succ 0\text{, derives } K_t \text{ for any } t)$$

$$V^*(x_{t-1}) = x_{t-1}^T Q x_{t-1} + u_{t-1}^{*T} R u_{t-1}^* + (Ax_{t-1} + Bu_{t-1}^*)^T P_t (Ax_{t-1} + Bu_{t-1}^*)$$

$$= x_{t-1}^T \left(Q + K_{t-1}^T R K_{t-1} + (A - BK_{t-1})^T P_t (A - BK_{t-1})\right) x_{t-1} \quad \text{(derives } P_{t-1}\text{)}$$

$$= x_{t-1}^T P_{t-1} x_{t-1}$$

Linear quadratic control (stochastic)

- Assumptions: ~~deterministic~~, finite horizon, discrete time
- Gaussian noise → Linear quadratic Gaussian (LQG) problem

$$x_{t+1} = f(x_t, u_t, \epsilon_t) = Ax_t + Bu_t + \epsilon_t \quad \epsilon_t \sim \mathcal{N}(0, \Sigma)$$

- Revised optimization problem:

$$u = \min_{u_0,\ldots,u_{T-1}} V(x_0; u) = \mathbb{E}\left[\sum_{t=0}^{T-1} x_t^T Q x_t + u_t R u_t + x_T^T Q_f x_T\right]$$
$$\text{subject to} \quad x_{t+1} = Ax_t + Bu_t + \epsilon_t$$

Theorem (LQG)

The optimal cost-to-go and optimal control at time t are given by:

$$V^*(x_t) = x_t^T P_t x_t + \Sigma_t$$
$$u_t^* = -K_t x_t$$

certainty equivalence: control as if disturbances were known (deterministic)!

where

$$P_t = Q + K_t^T R K_t + (A - BK_t)^T P_{t+1}(A - BK_t), \qquad P_T = Q_f$$
$$K_t = (R + B^T P_{t+1} B)^{-1} B^T P_{t+1} A, \qquad \Sigma_{t-1} = Tr\,(\Sigma P_t) + \Sigma_t, \qquad \Sigma_T = 0$$
$$t \in \{0, \ldots, T-1\}$$

- **Intuition**: noise terms are independent of actions → optimal actions don't change.
- **Exercise**: complete the proof!

Linear quadratic control (towards infinite horizon)

- Assumptions: deterministic, ~~finite horizon~~, discrete time
- Revised optimization problem:

No "final step"

$$u^* = \min_{u_0,\ldots,u_{T-1}} V(x_0; u) = \lim_{T\to\infty} \sum_{t=0}^{T-1} x_t^T Q x_t + u_t R u_t$$

$$\text{subject to} \quad x_{t+1} = A x_t + B u_t$$

Later: infinite horizon problems

- Before (finite horizon): finite horizon → finite sum.
- Now, need some condition to keep sum finite.
  - System (A, B) is **controllable** if A is full rank & $\bar{A} := [B \quad AB \quad A^2B \quad \ldots \quad A^{n-1}B]$ is full rank (n).

Theorem (infinite horizon LQR)

If the system (A, B) is controllable, the optimal cost-to-go and optimal control converges to

$$V^*(x) = x^T P x$$
$$u^* = -Kx$$

where

Algebraic Riccati Equation (ARE)

$$P = Q + A^T P A - A^T P B (R + B^T P B)^{-1} B^T P A$$
$$K = (R + B^T P B)^{-1} B^T P A$$

- **Exercise**: show that the expression is equivalent to before (in the limit).

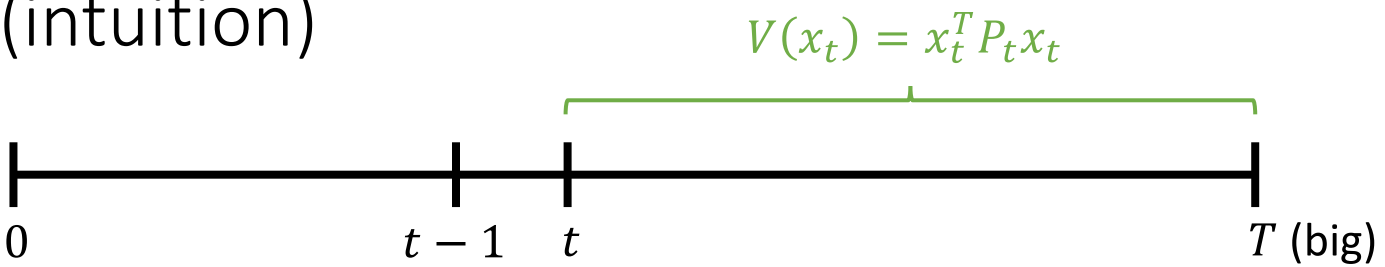
Controllability (for linear systems)

- System is **controllable** if A is full rank & $\bar{A} = [B \quad AB \quad A^2B \quad \dots \quad A^{n-1}B]$ is full rank (n).

- **Intuition**: Can s' be reached within n steps from any s?

$$\begin{aligned} x_{t+1} &= Ax_t + Bu_t \\ &= A(Ax_{t-1} + Bu_{t-1}) + Bu_t \\ &= A^2x_{t-1} + ABu_{t-1} + Bu_t \\ &= A^3x_{t-2} + A^2Bu_{t-2} + ABu_{t-1} + Bu_t \\ &\dots \end{aligned}$$

Proof (intuition)



For simplicity, take $P_T = Q_T = 0$

- $x^T P_t x \leq x^T P_{t-1} x$ (PSD)
- As $T \to \infty$, $x^T P_0 x$ must converge or go to infinity
- Controllability $\to$ For every x, there is a sequence $u_0, \dots, u_{n-1}$ (where $x \in \mathbb{R}^n$) that drives x to 0.
- After n steps, can set $u_k = 0$ for $k \geq n$.
- Controllability
  - $\to x^T P_0 x$ is bounded above, for any x
  - $\to P_0$ converges to finite limit.

LQR example (implementation)

$$x(k+1) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(k)$$

$$y(k) \qquad = [1 \quad 0] x(k)$$

Let $Q = C^T C = I, R = 0.3$. Infinite horizon. Solve the optimal control.

Solve DARE

$$S = A^T SA - A^T SB(R + B^T SB)^{-1} B^T SA + Q,\ K = -(R + B^T SB)^{-1} B^T SA$$

$$\Rightarrow S = \begin{bmatrix} 2.751.91 \\ 1.913.34 \end{bmatrix}$$

$$K = [-0.524, -1.44]$$

LQR example (implementation)

```
from __future__ import division, print_function
import numpy as np
import scipy.linalg
def dlqr(A,B,Q,R):
    """Solve the discrete time lqr controller.
    x[k+1] = A x[k] +S B u[k]
    cost = sum x[k].T*Q*x[k] + u[k].T*R*u[k]
    """
    #ref Bertsekas, p.151
    #first, try to solve the ricatti equation
    S = np.matrix(scipy.linalg.solve_discrete_are(A, B, Q, R))
    #compute the LQR gain
    K = -np.matrix(scipy.linalg.inv(B.T*S*B+R)*(B.T*S*A))
    eigVals, eigVecs = scipy.linalg.eig(A+B*K)
    return K, S, eigVals
```

```
A = np.array([[1,1],[0,1]])
B = np.array([[0],[1]])
Q = np.eye(2)
R = 0.3

K,S,_ = dlqr(A,B,Q,R)
print("S:", S)
print("K:", K)
```

```
S: [[2.75078485 1.90801622]
 [1.90801622 3.34052588]]
K: [[-0.52410456 -1.44169888]]
```

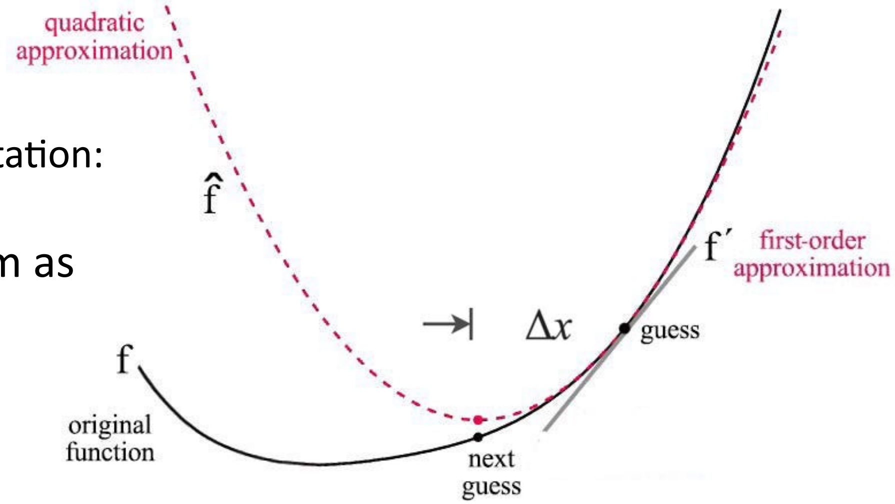
Courtesy Ding Zhao (CMU)

Further reading & extensions

- Further reading: 6.231 Sp22 Lecture 3 notes, Section 2 [N3 §2]
  - Excellent exposition, generally
  - Further discussion on observability
  - Stability
  - Loose ends (connections to other topics
- Lots of extensions
  - Continuous time (Callier & Desoer)
  - Model estimation, via LS & recursive LS
  - Adaptive control (Abbasi-Yadkori, 2011)
  - Unknown models, robust LQR (Dean, 2017)
  - Time Varying Regression with Hidden Linear Dynamics (Mania, 2022)

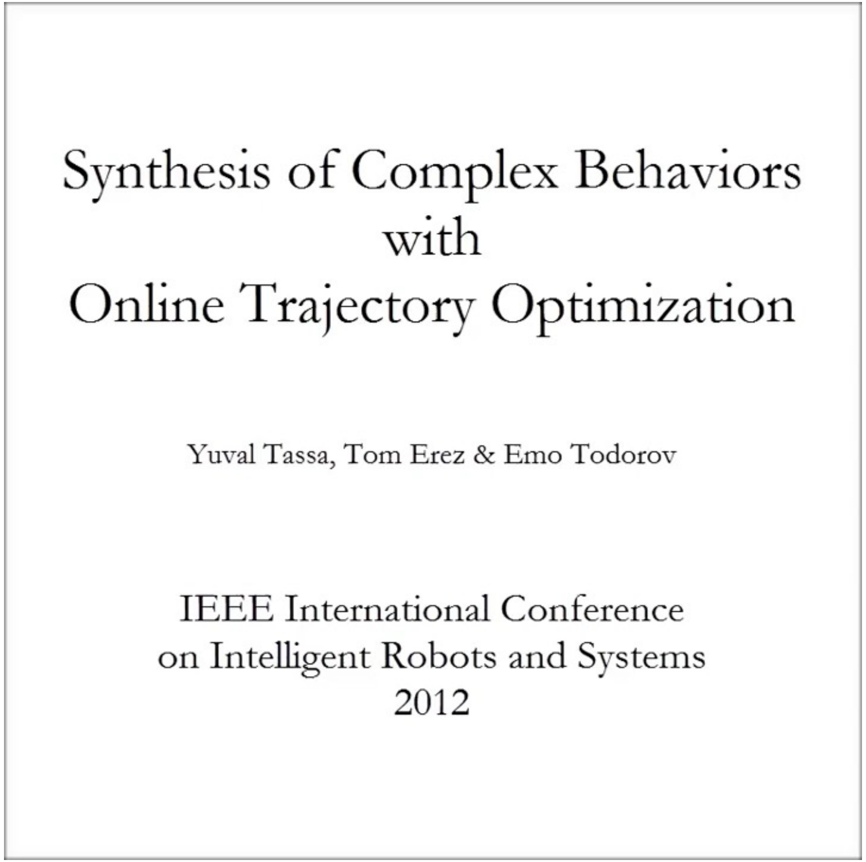
LQR – final notes

- Iterative LQR remains a powerful approach, e.g. in robotics.
- Extensions
  - Iterative LQR (iLQR) (full implementation: https://github.com/anassinator/ilqr)
    1. Approximate a nonlinear system as LQR using Taylor expansion
    2. Take a step or three
    3. Rinse and repeat to update the model & objective



[Jonathan Hui, 2018]

And there you have it - iLQR


Synthesis of Complex Behaviors
with
Online Trajectory Optimization

Yuval Tassa, Tom Erez & Emo Todorov

IEEE International Conference
on Intelligent Robots and Systems
2012

Summary & takeaways

- Certain DP problems admit closed form solutions, such as **optimal stopping** and **linear quadratic control** (LQR).
- DP for problems with special structures can be analyzed by **induction**, by showing that the special structure holds from one step to the previous, as well as for the terminal case. Special structures include **convexity** and **monotonicity**.
- LQR exhibits **certainty equivalence**: the optimal policy remains the same when random disturbances are replaced with their means (conditional expectation).

References

1. 6.231 Sp22 Lecture 3 notes, Section 2 [N3 §2]
2. DPOC vol 1, 3.1 (LQR), 3.3-3.4
3. Some material adapted from:
    - Daniel Russo (Columbia)