Reinforcement Learning

Lecture 12:
Linear Control

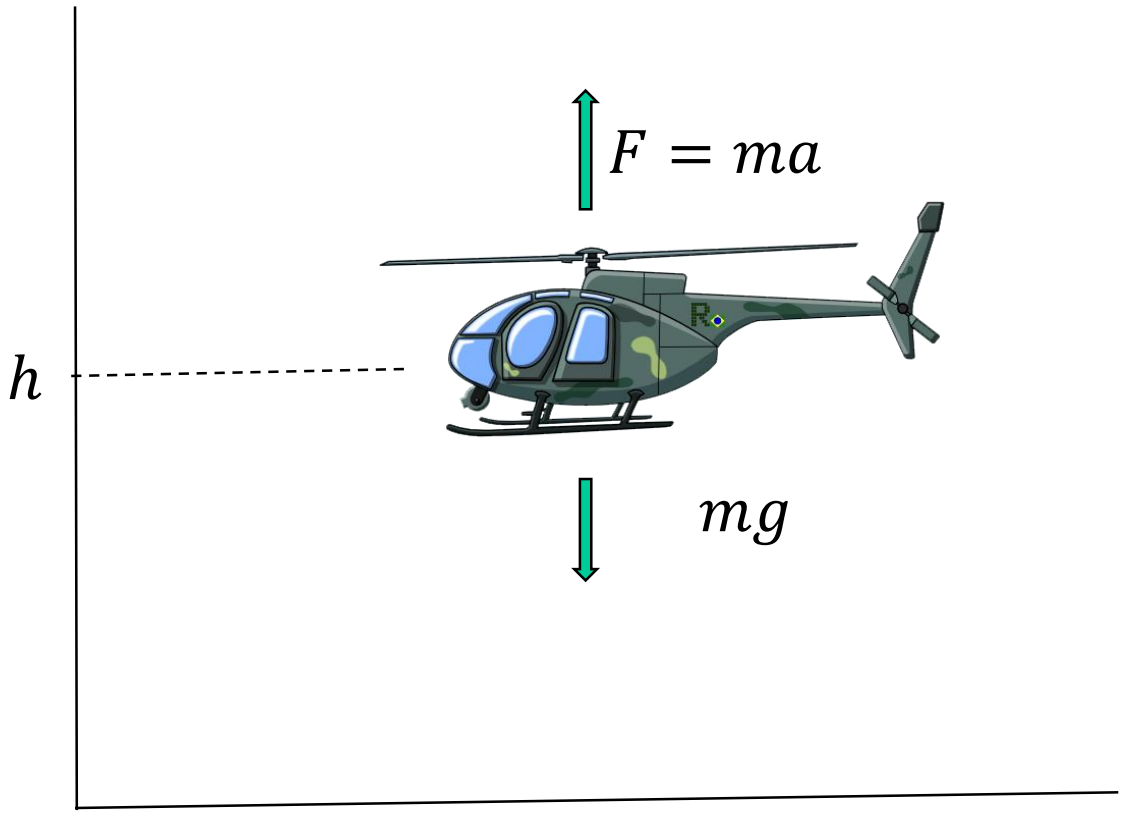
Yishay Mansour, Tel-Aviv University

Lecture 12: outline

- Linear Dynamics
- Linear Quadratic Regulator
  - LQR definition
  - Finite Horizon
  - Infinite Horizon
  - Controllability
- Extensions:
  - Affine system
  - Smoothness
  - Linear Quadratic Gaussian
  - Time changing
  - Non-linear
- Applications
  - Helicopter maneuvers

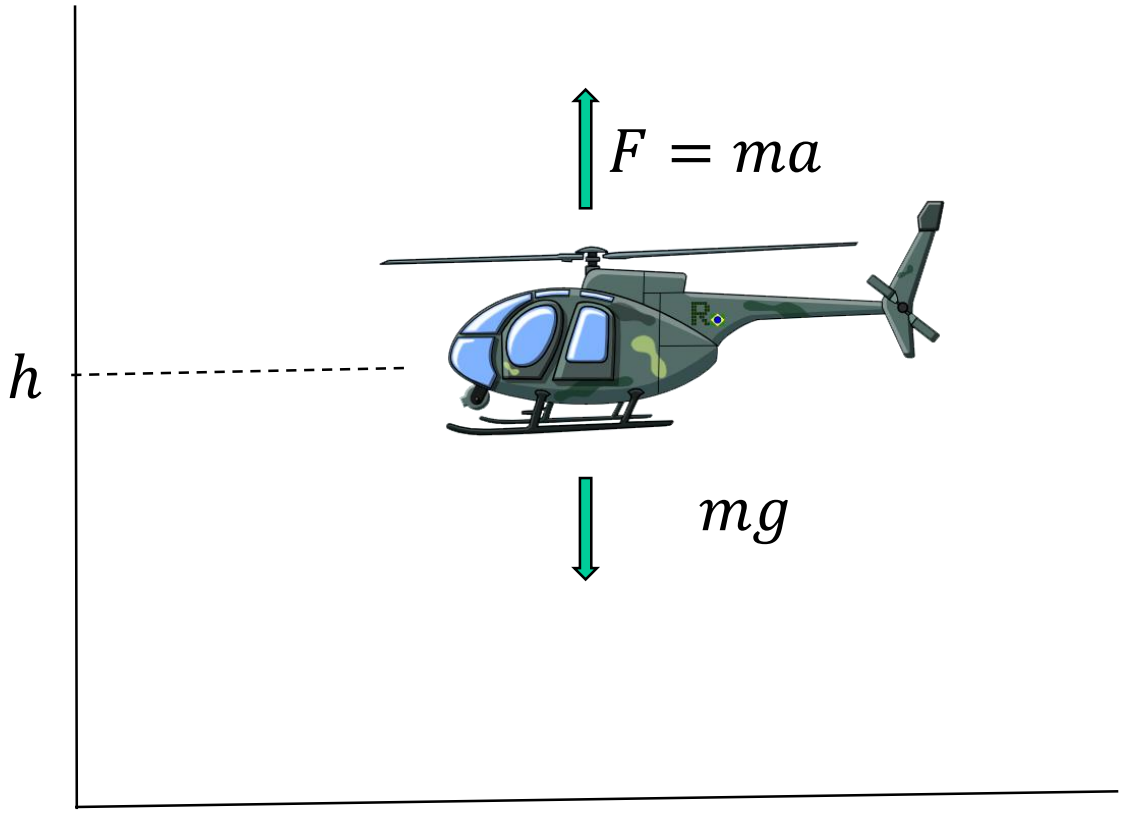
Linear System Dynamics

- Consider the helicopter
- Height set as a function of
  - Current position
  - Current acceleration
- Simple physical dynamics



Linear System Dynamics

- Dynamics:
  - $h_{t+1} = h_t + \tau v_t + \frac{1}{2}\tau^2(a_t - g)$
  - $v_{t+1} = v_t + \tau(a_t - g)$
- Parameters:
  - τ time step (sec)
  - h_t height (meter)
  - v_t velocity (m/sec)
  - a_t acceleration (m/s^2)
  - g gravitation



Linear System Dynamics

- Write in the following form
  - $\begin{bmatrix} h_{t+1} \\ v_{t+1} \end{bmatrix} = \begin{bmatrix} 1 & \tau \\ 0 & 1 \end{bmatrix} \begin{bmatrix} h_t \\ v_t \end{bmatrix} + \begin{bmatrix} \frac{1}{2}\tau^2 \\ \tau \end{bmatrix} (a_t - g)$
  - $x_{t+1} = A x_t + B u_t$
- Linear Time Invariant (LTI) system
  - $x_{t+1} = Ax_t + Bu_t$

Linear Dynamics System: Cost

- Would like to reach a desired state
  - Helicopter get to height h_D in T steps
- Would like to minimize some cost
- Quadratic objective
  - $J(u) = \sum_{t=1}^{T} x_t^\top Q x_t + u_t^\top R u_t^\top$
    - Metrices semi-positive-definite (cost non-negative): $Q, R \succ 0$
    - Symmetric: $Q = Q^\top; R = R^\top$

Example: Helicopter

- State and action:
  - Desired final state:
    - Height h_D
    - Velocity: 0
  - $x_t = \begin{bmatrix} h_t \\ v_t \end{bmatrix} - \begin{bmatrix} h_D \\ 0 \end{bmatrix}$
  - $u_t = (a_t - g)$

- Example
  - $h_0 = 0; v_0 = 0; \tau = 2; h_D = 4$
  - Dynamics:
    - $\begin{bmatrix} h_{t+1} \\ v_{t+1} \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} h_t \\ v_t \end{bmatrix} + \begin{bmatrix} 2 \\ 2 \end{bmatrix} u_u$
  - Set $u_1 = 1; u_2 = -1$
    - $\begin{bmatrix} h_1 \\ v_1 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 2 \\ 2 \end{bmatrix} 1 = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$
    - $\begin{bmatrix} h_2 \\ v_2 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 2 \end{bmatrix} - \begin{bmatrix} 2 \\ 2 \end{bmatrix} = \begin{bmatrix} 4 \\ 0 \end{bmatrix}$

Relating to MDP

- MDP:
  - States: s $\in S$
  - Actions: a $\in A$
  - Dynamics: $p('s|s,a)$

- Linear dynamics:
  - States: $x \in \mathbb{R}^n (= S)$
  - Action: $u \in \mathbb{R}^d \ (= A)$
  - Dynamics: $x' = Ax + Bu$

  - Alternatively:
    - $p(x'|x,u) = 1$

      If and only if
    - $x' = Ax + Bu$

Linear Quadratic Regulator: Discrete time finite horizon

- Minimize: over $u_0, \dots, u_{T-1}$

$$J(u_0, \dots, u_{T-1}, x_0) = \sum_{t=0}^{T-1} x_t^\top Q x_t + u_t^\top R u_t + x_T^\top Q_f x_T$$

- Such that
  - $x_{t+1} = Ax_t + Bu_t$
    - $t = 0, 1, \dots, T-1$

LQR parameters

- T time horizon (latter infinite horizon)
- $x_t^\top Q x_t$ state cost (deviation from desired state)
- $u_t^\top R u_t$ control input cost
- $x_T^\top Q_f x_T$ terminal state cost
- $J(u_0, \dots, u_{T-1}, x_0) = \sum_{t=0}^{T-1} x_t^\top Q x_t + u_t^\top R u_t + x_T^\top Q_f x_T$

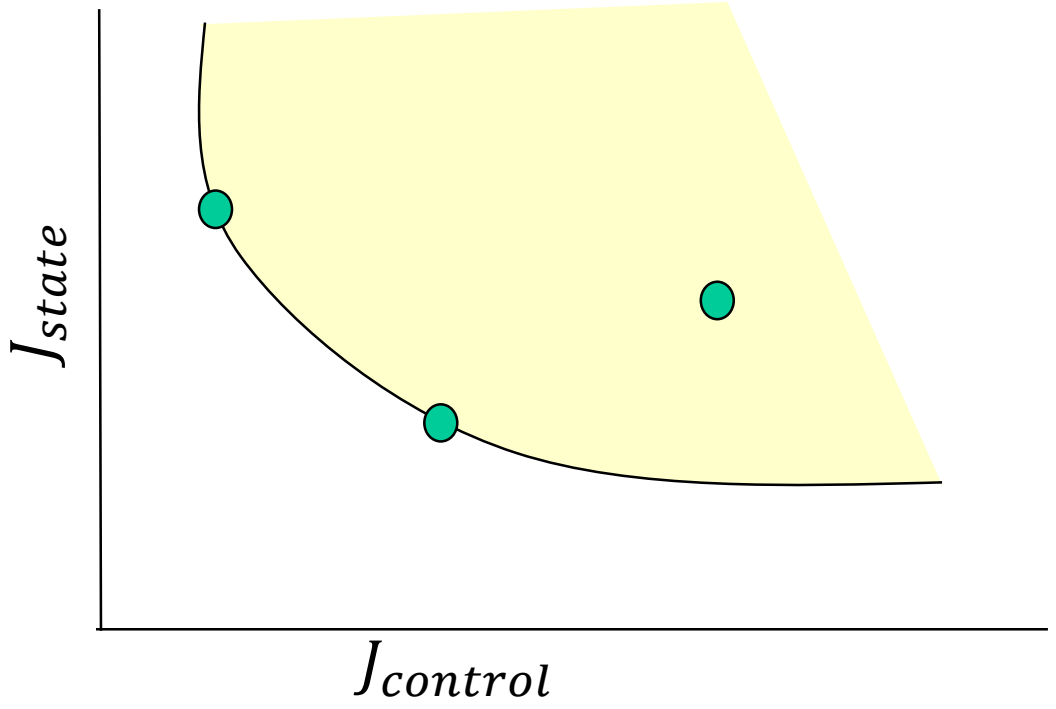
Motivating the costs

- Two parts to the cost:
  - State
  - Control
- Often “unrelated”
  - Energy (money)
  - Accuracy (meter)
- Actually: tradeoff

- Common setting
  - $R = \rho I \;\; ; \;\; Q = C^\top C$
  - $y_t = C x_t$
- Cost
  - $\sum_t \|y_t\|^2 + \rho \sum_t \|u_t\|^2$
  - ρ tradeoff parameter

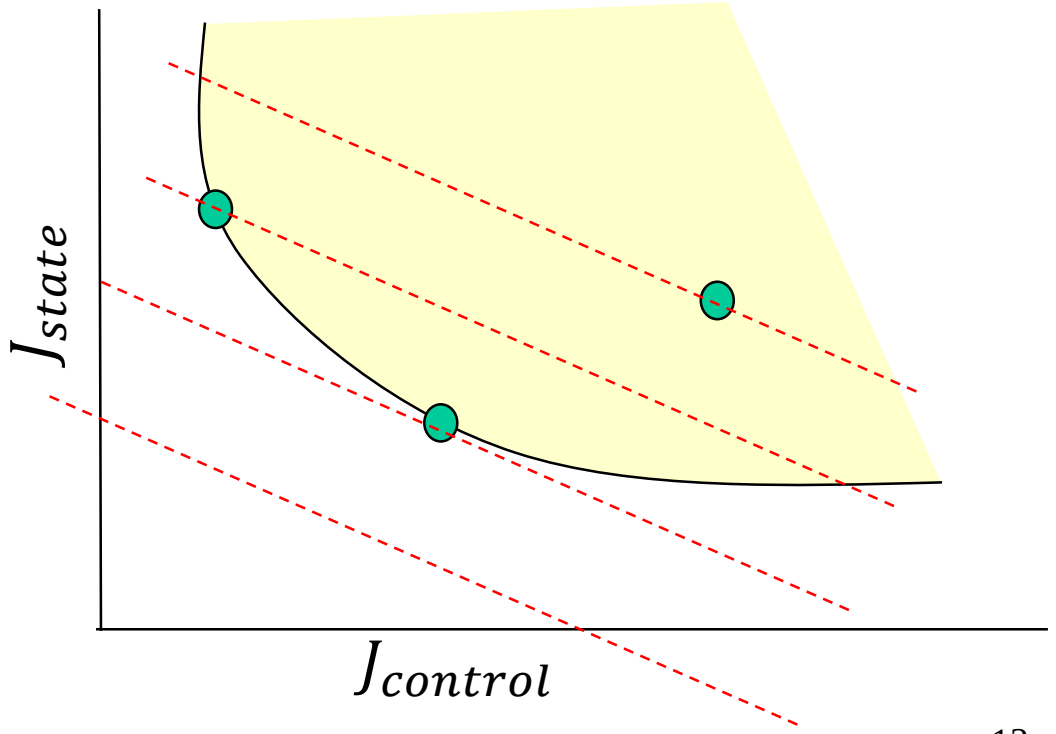
Tradeoff between state and control cost

- Set costs
  - $J_{state} = \sum_t \|y_t\|^2$
  - $J_{control} = \sum_t \|u_t\|^2$



Tradeoff between state and control cost

- Set costs
  - $J_{state} = \sum_t \|y_t\|^2$
  - $J_{control} = \sum_t \|u_t\|^2$
- Setting cost
  - Equal costs:
  - $J_{state} + \rho\, J_{control} = const$



Movement cost problem

- Assume there is no state cost
  - Only control cost
  - $J_{control} = \sum_t \|u_t\|^2$
- Very simple dynamics
  - $A = I; B = I$
  - $x_{t+1} = x_t + u_t$
- Given
  - x_0 and $x_T = x_{final}$
  - Set the controls
- Naïve solution
  - $u_1 = x_{final} - x_0; u_t = 0 \; for \; t \geq 2$
  - Cost $\|u_1\|^2 = \|x_{final} - x_0\|^2$
- Better
  - Move T steps on the line $x_0 \rightarrow x_T$
  - $u_t = \frac{x_{final} - x_0}{T}$
  - Cost: $T \left\| \frac{x_{final} - x_0}{T} \right\|^2 = \frac{1}{T} \|x_{final} - x_0\|^2$

Optimal LQR control

- Using dynamic programing
- For each state x_t define cost-to-go
  - $J(u_t, \dots, u_{T-1}, x_t) = \sum_{i=t}^{T-1} x_i^\top Q x_i + u_i^\top R u_i + x_T^\top Q_f x_T$
- An optimal solution for $[1, T]$
  - Is also optimal for any suffix $[t, T]$
- Define the optimal value function: $V_t^*(x_t)$

LQR: optimization problem

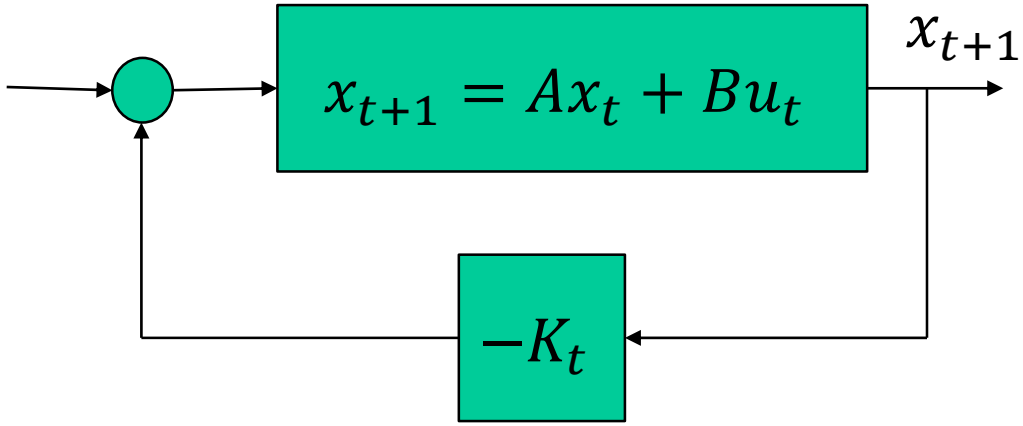
- Optimization problem
  - $V_0^*(x_0) = \min_{u_0,\dots,u_{T-1}} J(u_0, \dots, u_{T-1}, x_0)$
- At any time t
  - $V_t^*(x_t) = x_t^\top Q x_t + (u_t^*)^\top R u_t^* + V_{t+1}^*(Ax_t + Bu_t^*)$
  - $u_t^* = \arg\min_u x_t^\top Q x_t + u_t^\top R u_t + V_{t+1}^*(Ax_t + Bu_t)$
- Construct optimal control from $\mathrm{t} = T$ back to $\mathrm{t} = 0$

LQR: optimal control Theorem

- Theorem
  - The optimal cost-to-go and optimal control at time t
  - $V_t^*(x_t) = x_t^\top P_t x_t$ and $u_t^* = -K_t x_t$
  - Where
    - $P_t = Q + K_t^\top R K_t + (A - BK_t)^\top P_{t+1}(A - BK_t)\ ;\ P_T = Q_f$
      - P_t is symmetric and p.s.d., i.e., $P_t = P_t^\top$
    - $K_t = (R + B^\top P_{t+1} B)^{-1} B^\top P_{t+1} A$

LQR: optimal control Theorem

- What does the theorem say:
  - Optimal cost-to-go is a quadratic function of state
    - $x_t^\top P_t x_t$
  - The optimal control is linear in state
    - $u_t = -K_t x_t$



LQR: optimal control Theorem (Example)

- Consider
  - $A = \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix}; B = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$
  - $R = [1]$
  - $Q = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$
  - $P_T = Q_f = \begin{bmatrix} 100 & 0 \\ 0 & 100 \end{bmatrix}$

- $K_t = (R + B^\top P_{t+1} B)^{-1} B^\top P_{t+1} A$
  - $R + B^\top P_T B = 801$
  - $K_{T-1} = \frac{1}{801} B^\top P_T A$
    - $= \frac{[2 \quad 2]}{801} \begin{bmatrix} 100 & 0 \\ 0 & 100 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix}$
    - $= \begin{bmatrix} \frac{200}{801} & \frac{600}{801} \end{bmatrix}$
- $P_{T-1} = Q + K_{T-1}^\top R K_{T-1}$ $+(A - BK_{T-1})^\top P_T (A - BK_{T-1})$

LQR: optimal control Theorem (Example)

- Recall:

$$P_{T-1} = Q + K_{T-1}^{\top} R K_{T-1} + (A - BK_{T-1})^{\top} P_T (A - BK_{T-1})$$

- $A - BK_{T-1} = \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 2 \\ 2 \end{bmatrix} \begin{bmatrix} \frac{200}{801} & \frac{600}{801} \end{bmatrix} = \begin{bmatrix} \frac{401}{801} & \frac{402}{801} \\ \frac{-400}{801} & \frac{-399}{801} \end{bmatrix} \approx \begin{bmatrix} 1/2 & 1/2 \\ -1/2 & -1/2 \end{bmatrix}$

- $K_{T-1}^{\top} R K_{T-1} = \begin{bmatrix} \frac{200}{801} \\ \frac{600}{801} \end{bmatrix} [1] \begin{bmatrix} \frac{200}{801} & \frac{600}{801} \end{bmatrix} \approx \begin{bmatrix} 1/16 & 3/16 \\ 3/16 & 9/16 \end{bmatrix}$

LQR optimal control theorem: Proof

- Proof by induction:
  - From $t = T$ to $t = 0$
- Base: $t = T$
  - $V_T^*(x_T) = x_T^\top Q_f x_T$
    - $P_T = Q_f$
- Inductive step:
  - Assume it holds from t
  - Prove for $t - 1$

LQR optimal control theorem: Proof

- We have
  - $V_{t-1}^*(x_{t-1}) = \min_u x_{t-1}^\top Q x_{t-1} + u^\top R u + V_t^*(Ax_{t-1} + Bu)$
  - By the inductive hypothesis
  - $V_{t-1}^*(x_{t-1}) = \min_u x_{t-1}^\top Q x_{t-1} + u^\top R u + (Ax_{t-1} + Bu)^\top P_t (Ax_{t-1} + Bu)$
  - Solving for u
    - $\nabla_u V_{t-1}^*(x_{t-1}) = 2u^\top R + 2(Ax_{t-1} + Bu)^\top P_t B$
    - $u_{t-1}^* = -(R + B^\top P_t B)^{-1} B^\top P_t A x_{t-1} = -K_{t-1} x_{t-1}$
      - Since P_t is p.s.d we have that $R + B^\top P_t B$ is p.s.d. and invertable

LQR optimal control theorem: Proof

- Need to compute value function
  - $V_{t-1}^*(x_{t-1}) = x_{t-1}^\top Q x_{t-1} + {u_{t-1}^*}^\top R u_{t-1}^* + (Ax_{t-1} + Bu_{t-1}^*)^\top P_t (Ax_{t-1} + Bu_{t-1}^*)$
  - $= x_{t-1}^\top Q x_{t-1} + x_{t-1}^\top K_{t-1}^\top R K_{t-1} x_{t-1} + (Ax_{t-1} - BK_{t-1}x_{t-1})^\top P_t (Ax_{t-1} - BK_{t-1}x_{t-1})$
  - $V_{t-1}^*(x_{t-1}) = x_{t-1}^\top \left(Q + K_{t-1}^\top R K_{t-1} + (A - BK_{t-1})^\top P_t (A - BK_{t-1}) \right) x_{t-1}$
  - $V_{t-1}^*(x_{t-1}) = x_{t-1}^\top P_{t-1} x_{t-1}$
  - $P_{t-1} = Q + K_{t-1}^\top R K_{t-1} + (A - BK_{t-1})^\top P_t (A - BK_{t-1})$
    - P_{t-1} is the sum of p.s.d. matrices and therefore p.s.d.
- Q.E.D.

LQR Infinite Horizon

- Assume $T = \infty$
- Value function
  - $V_0^*(x_0) = \min_{u_0,\dots} \sum_{t=0}^{\infty} x_t^\top Q x_t + u_t^\top R u_t$
    - Where $x_{t+1} = Ax_t + Bu_t$
- Are the costs finite?
  - YES,
    - assuming we can reach $x_t = 0$

- Theorem
  - There exists matrices P and K,

  Such that at time t:
  - The optimal cost-to-go
    - $V^*(x_t) = x_t^\top P x_t$
  - and optimal action at time t
    - $u_t^* = -Kx_t$

LQR Infinite Horizon: computing

- Bellman's optimality equation:
  - $V^*(x) = \min_u x^\top Q x + u^\top R u + V^*(Ax + Bu)$
  - $= \min_u x^\top Q x + u^\top R u + (Ax + Bu)^\top P (Ax + Bu)$
- Minimizing over u
  - $\nabla_u V^*(x) = 2u^\top R + 2(Ax + Bu)^\top PB$
  - $u^* = -(R + B^\top PB)^{-1} B^\top PAx = -Kx$

LQR Infinite Horizon: Value function

- The value function:
  - $V^*(x) = x^\top P x$
  - $= x^\top Q x + u^{*\top} R u^* + (Ax + Bu^*)^\top P (Ax + Bu^*)$
  - $= x^\top (Q + A^\top P A - A^\top P B (R + B^\top P B)^{-1} B^\top P A) x$
- Holds for any x:
  - $P = Q + A^\top P A - A^\top P B (R + B^\top P B)^{-1} B^\top P A$
    - Algebraic Riccati Equation (ARE)

Controllability

- Solution to ARE
  - Exists, if system controllable
- Controllability.
  - From any x_0
  - Can reach any x_D
  - Using controls: u_0, ...

- Consider:
  - $x_{t+1} = Ax_t + Bu_t$
  - $= A(Ax_{t-1} + Bu_{t-1}) + Bu_t$
  - $= A^2x_{t-1} + ABu_{t-1} + Bu_t$
  - $= A^3x_{t-2} + A^2Bu_{t-2} + ABu_{t-1} + Bu_t$
  - $= A^{t+1}x_0 + \sum_{i=0}^{t} A^i\, Bu_{t-i}$

Controllability

- Matrix form:
  - $C = [B\ AB\ A^2B \cdots A^tB]$
  - $U = \begin{bmatrix} u_t^\top & u_{t-1}^\top \cdots u_0^\top \end{bmatrix}$
  - $x_{t+1} = A^{t+1}x_0 + CU^\top$
  - Sufficient that C is full rank
    - Can force any $x_{t+1} = x_D$
  - How large of a t we need?

- How far do we need to go?
  - Cayley-Hamilton theorem:
    - A^n linear depends on A^i, $i < n$
      - Consider the characteristic polynomial $p(\lambda) = \det(\lambda I - A)$
- Sufficient condition:
  - $C = [B\ AB\ A^2B \cdots A^{n-1}B]$
  - C is full rank
  - ARE has a solution

Controllability: Examples

- Helicopter example:
  - $A = \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix}; B = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$
  - $A^t = \begin{bmatrix} 1 & 2t \\ 0 & 1 \end{bmatrix}$
  - $C = \begin{bmatrix} 2 & 6 & 10 \\ 2 & 2 & 2 \end{bmatrix}$

- Un-controllable example
  - $A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}; B = \begin{bmatrix} 1 \\ 1 \end{bmatrix};$
  - $C = \begin{bmatrix} 1 & \cdots & 1 \\ 1 & \cdots & 1 \end{bmatrix}$
  - Let $x_0 = 0;\ U = \sum u_t$
  - $x_t = \begin{bmatrix} U \\ U \end{bmatrix}$
  - However, if we can reach x_t
    - We can reach it in x_1 and stay there

Controllability: Examples

- Another example
  - $A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}; B = \begin{bmatrix} 1 \\ 1 \end{bmatrix};$
  - $A^{2t} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}; A^{2t+1} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
  - $C = \begin{bmatrix} 1 & \cdots & 1 \\ 1 & \cdots & 1 \end{bmatrix}$
  - Let $x_0 = 0;\ U = \sum u_t$
  - $x_t = \begin{bmatrix} U \\ U \end{bmatrix}$

- Assume we can reach x_T
- Q: Are we guarantee in x_1
  - And stay there?
- A: NO
  - Let $x_0 = \begin{bmatrix} 1 \\ 0 \end{bmatrix};\ U = \sum u_t$
  - $x_{2t} = \begin{bmatrix} 1+U \\ U \end{bmatrix}; x_{2t+1} = \begin{bmatrix} U \\ 1+U \end{bmatrix}$

LQR: Stability

- P: solution to ARE
  - $K = (R + B^\top P B)^{-1} B^\top P A$
  - $u^* = -Kx$
- Optimal system dynamics
  - $x_{t+1} = Ax_t + Bu_t$
  - $= Ax_t - BKx_t$
  - $= (A - BK)x_t$
  - $= (A - BK)^{t+1} x_0$

- Eigenvalues of $A - BK$
  - $A - BK = M \Lambda M^{-1}$
    - Λ diagonal with $\lambda_1, \ldots, \lambda_n$
  - $x_t = M \Lambda^t M^{-1} x_0$
- If $\forall i: |\lambda_i| < 1$
  - System converges to zero
    - stable
- If $\exists i: |\lambda_i| > 1$
  - System divergence.
    - unstable

Lecture 12: outline

- Linear Dynamics
- Linear Quadratic Regulator
  - LQR definition
  - Finite Horizon
  - Infinite Horizon
  - Controllability
- Extensions:
  - Affine system
  - Smoothness
  - Linear Quadratic Gaussian
  - Time changing
  - Non-linear
- Applications
  - Helicopter maneuvers

Extensions of LQR

- Affine system
  - Add constant vector
- Smoothness
  - Penalize $u_t - u_{t-1}$
- Noise
  - Add Gaussian noise
  - Linear Quadratic Gaussian
- Linear time varying (LTV)
  - Matrices A_t, B_t
- Non-linear system
  - Linearize the system

LQR: Affine system

- New Dynamics
  - $x_{t+1} = Ax_t + Bu_t + \mathrm{c}$
  - $cost: x_t^\top Q x_t + u_t^\top R u_t$
- Optimal control remains linear!
  - Redo the math …
  - Do a reduction
- Define
  - $\begin{bmatrix} x_{t+1} \\ 1 \end{bmatrix} = \begin{bmatrix} A & c \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_t \\ 1 \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u_t$
  - $z_t = \begin{bmatrix} x_t \\ 1 \end{bmatrix}$
  - $z_{t+1} = A' z_t + B' u_t$

LQR: Smoothness

- Standard Dynamics
  - $x_{t+1} = Ax_t + Bu_t$
  - $x_t^\top Q x_t + u_t^\top R u_t + \Delta u_t^\top R' \Delta \mathrm{u_t}$
    - where $\Delta u_t = u_t - u_{t-1}$

- Solution:
  - Augment state x_t with u_{t-1}
  - Change action to Δu_t

- Define
  - $\begin{bmatrix} x_{t+1} \\ u_t \end{bmatrix} = \begin{bmatrix} A & B \\ 0 & I \end{bmatrix} \begin{bmatrix} x_t \\ u_{t-1} \end{bmatrix} + \begin{bmatrix} B \\ I \end{bmatrix} \Delta u_t$
  - $z_t = \begin{bmatrix} x_t \\ u_{t-1} \end{bmatrix}$
  - $z_{t+1} = A' z_t + B' u_t$
- Cost:
  - $Q' = \begin{bmatrix} Q & 0 \\ 0 & R \end{bmatrix}$,
  - $\sum_{t=0}^{\infty} z_t^\top Q' z_t + \Delta u_t^\top R' \Delta \mathrm{u_t}$

Linear Quadratic Gaussian (LQG)

- Assume stochastic dynamics
  - $x_{t+1} = Ax_t + Bu_t + w_t$
    - $w_t \sim N(0, W)$
- Similar objective
  - Now in expectation
  - Cost is non-zero
    - Due to noise

- Optimal value function
  - $V_t^*(x_t) = x_t^\top P_t x_t + m_t$

- Will show it by backward induction

LQG: Optimal value

- Value function
  - $V_{t-1}^*(x_{t-1}) = \min_{u_{t-1}} x_{t-1}^\top Q x_{t-1} + u_{t-1}^\top R u_{t-1} + E[V_t^*(x_t)]$
    - $x_t = Ax_{t-1} + Bu_{t-1} + w_t$
  - $E[V_t^*(x_t)] = E[(Ax_{t-1} + Bu_{t-1} + w_t)^\top P_t (Ax_{t-1} + Bu_{t-1} + w_t)] + m_t$
  - $V_{t-1}^*(x_{t-1}) = \min_{u_{t-1}} x_{t-1}^\top Q x_{t-1} + u_{t-1}^\top R u_{t-1} + (Ax_{t-1} + Bu_{t-1})^\top P_t (Ax_{t-1} + Bu_{t-1}) + m_{t-1}$
    - $m_{t-1} = m_t + Tr(WP_t)$
  - Identical solution to LQR
    - Gaussian Noise has NO EFECT on optimal action!

LQG: optimal control and value function

- Optimal value function and control
  - $\nabla_{u_{t-1}} V_{t-1}^*(x_{t-1}) = 2u_{t-1}R + 2(Ax_{t-1} + Bu_{t-1})^\top P_t B = 0$
  - $u_{t-1}^* = -(R + B^\top P_t B)^{-1} B^\top P_t A x_{t-1} = -K_{t-1} x_{t-1}$
  - $V_{t-1}^*(x_{t-1}) = x_{t-1}^\top P_t x_{t-1} + m_{t-1}$
    - $m_{t-1} = Tr(WP_t) + m_t; \quad m_T = 0$

- Why is the optimal control independent of W ?

Time varying system

- Simply:
  - $x_{t+1} = A_t x_t + B_t u_t$
  - Cost $x_t^\top Q x_t + u_t^\top R u_t$
- Similar math
  - $u_t^* = -K_t x_t$
  - $V_t^*(x_t) = x_t^\top P_t x_t$

Non-linear system

- Dynamics
  - $x_{t+1} = f(x_t, u_t)$
- Goal:
  - Stabilize around x^*
  - Needs u^* s.t.
    - $x^* = f(x^*, u^*)$
- Linearizing:
  - $x_{t+1} \approx f(x^*, u^*)$
  - $+A(x_t - x^*) + B(u_t - u^*)$
- The matrices
  - $A = \frac{\partial f(x^*, u^*)}{\partial x}$; $B = \frac{\partial f(x^*, u^*)}{\partial u}$
- Reduce to LQR:
  - $z_t = x_t - x^*$; $v_t = u_t - u^*$
  - $z_{t+1} = Az_t + Bu_t$
  - Cost: $z_t^\top Q z_t + v_t^\top R v_t$
  - $v_t = -Kz_t$
  - $u_{t+1} - u^* = -K(x_t - x^*)$
  - $u_{t+1} = u^* - K(x_t - x^*)$

Non-linear system

- When would it work
  - Need a good approximation
  - Both x^* and u^*
  - Mainly good for “final optimization”
  - Need some starting point
- Tracking a trajectory
  - Apprentice learning
  - Given:
    - $x_1^*, \cdots, x_T^*$
    - $u_1^*, \cdots, u_T^*$
  - Set the error:
    - $e_t = (x_t - x_t^*, u_t - u_T^*)$
  - Optimize performance:
    - Linearize the system
    - Solve for optimal control

System estimation

- So far, Planning
- What about Learning?
- Model based:
  - Estimate a system
    - A and B
  - Plan using the learned system

- Least Square methods
  - Ordinary
    - Offline
  - Recursive
    - online

System estimation: Ordinary least squares

- Assume we observe a trajectory
  - $(x_t, u_t)_{t=0}^{T}$
- System dynamics:
  - $x_{t+1} = Ax_t + Bu_t$
  - Unknowns: A, B
- Least squares
  - $\min_{A,B} \sum_t \|x_{t+1} - (Ax_t + Bu_t)\|_2^2$

- Solving regression:
  - $M = \begin{bmatrix} A \\ B \end{bmatrix}$; $z_t = [x_t \ u\_t]$
  - $Z = \begin{bmatrix} z_0 \\ \vdots \\ z_{T-1} \end{bmatrix}$; $X = \begin{bmatrix} x_1 \\ \vdots \\ x_T \end{bmatrix}$
  - $X \approx ZM$
- Least square solution:
  - $\widehat{M} = (Z^\top Z)^{-1} Z^\top X$
    - Assuming $Z^\top Z$ invertible

System estimation: Recursive least squares

- Keep updating the estimate
- At time t:
  - $\widehat{M}_t = \left(Z_t^\top Z_t\right)^{-1} Z_t^\top X_t = \Phi_t^{-1}\Psi_t$
    - $\Phi_t = Z_t^\top Z_t = \sum_i z_i^\top z_i$
    - $\Psi_t = Z_t^\top X_t = \sum_i z_i^\top x_i$

- Updating the parameters
  - $\Phi_{t+1}^{-1} = \left(\Phi_t + z_t^\top z_t\right)^{-1}$
  - $= \Phi_t^{-1} - \frac{\Phi_t^{-1} z_{t+1} z_{t+1}^\top \Phi_t^{-1}}{1 + z_{t+1}^\top \Phi_t^{-1} z_{t+1}}$
  - $\Psi_{t+1} = \Psi_t + z_{t+1}^\top x_{t+1}$
  - $\widehat{M}_{t+1} = \Phi_{t+1}^{-1}\Psi_{t+1}$
- Simple updates

Lecture 12: outline

- Linear Dynamics
- Linear Quadratic Regulator
  - LQR definition
  - Finite Horizon
  - Infinite Horizon
  - Controllability
- Extensions:
  - Affine system
  - Smoothness
  - Linear Quadratic Gaussian
  - Time changing
  - Non-linear
- Applications
  - Helicopter maneuvers

Applications: Helicopter

http://heli.stanford.edu/

What were the goals?

- Goals:
  - Learn wide-range maneuvers:
  - in-place flips and rolls,
  - loops,
  - auto-rotation landings,
  - and much more

- Results
  - For humans:
    - Most maneuver hard for to learn
  - Performance:
    - Outperform experts.

Flow of learning

- Build a Baseline Dynamics Model
  - Collect 20 minutes of data
  - Exploration: use all actions sufficiently
- Apprenticeship Learning:
  - Target Trajectory
    - Demonstrations for each maneuver
  - Refined Dynamics Model
    - For each specific maneuver
- Autonomous Flight Control
  - Learn reward function
    - penalizes for deviation from the inferred target trajectory.
  - Run LQR
    - For the non-linear model (offline)
  - Fly our helicopter autonomously
    - Good enough: Done
    - Else: add the new data and improve model

Helicopter state and controls

- State:
  - Position
  - Orientation
  - Velocity
  - Angular velocity
- 12 parameters
  - Normalized to helicopter coordinates

- Actions:
  - Four parameters:
  - Latitude cyclic pitch controls
    - Left-right
  - Longitude cyclic pitch controls
    - Front-back
  - Yaw rate
    - rotation rate of the helicopter about its vertical axis
  - Pitch angle
    - rotating the blades

Model Structure

- Approximating a linear model:
  - Inputs
  - (u, v, w): linear velocities
  - (p, q, r): angular velocities
  - (g_x, g_y, g_z): gravity
- Parameter Learning:
  - Record data from expert pilots
  - Estimated linear model from logs

$$
\begin{aligned}
\dot{u} &= vr - wq + A_x u + g_x + w_u, \\
\dot{v} &= wp - ur + A_y v + g_y + D_0 + w_v, \\
\dot{w} &= uq - vp + A_z w + g_z + C_4 u_4 + D_4 + w_w, \\
\dot{p} &= qr(I_{yy} - I_{zz})/I_{xx} + B_x p + C_1 u_1 + D_1 + w_p, \\
\dot{q} &= pr(I_{zz} - I_{xx})/I_{yy} + B_y q + C_2 u_2 + D_2 + w_q, \\
\dot{r} &= pq(I_{xx} - I_{yy})/I_{zz} + B_z r + C_3 u_3 + D_3 + w_r.
\end{aligned}
$$

Improved Helicopter Dynamics Model by Local Parameter Learning

- Inaccuracies
  - Dependency on previous states
  - Hidden parameters
    - Airflow
    - Rotor head speed
    - Actuator dynamics
  - Dynamics not linear!

- Solution 1:
  - Learn non-linear model
- Solution 2:
  - Concentrate on trajectory
  - Build trajectory specific models
  - Time exponential decaying

Learning a Reward Function from Multiple Demonstrations

- Given multiple trajectories
  - Generated by experts
- Reduce to a single trajectory
  - Best trajectory
- Dynamics
  - Linear: Average
  - Non-Linear: Main issue

- Uses:
  - Generative model
  - Compute ML solution
    - Using EM

Linear Quadratic Methods

- Uses an LQG model
  - $x_{t+1} = A_t x_t + B_t u_t + w_t$
  - Cost: $x_t^\top Q_t x_t + u_t^\top R u_t$
- Methodology
  - Compute linear approximation
  - Compute optimal policy
- Note:
  - Approximation only in the dynamics
- Design choices
  - Penalize change in control
  - Average cost
    - Add a parameter for average costs
    - Needed since optimal cost not zero
- Non-linearity
  - Horizon 2-seconds
  - Re-compute linearizationg

Lecture 12: outline

- Linear Dynamics
- Linear Quadratic Regulator
  - LQR definition
  - Finite Horizon
  - Infinite Horizon
  - Controllability
- Extensions:
  - Affine system
  - Smoothness
  - Linear Quadratic Gaussian
  - Time changing
  - Non-linear
- Applications
  - Helicopter maneuvers