Lecture 3. Optimal Control

Deterministic Discrete Finite Time

Controller Design Problem (Finite Horizon)

- Given a state space **X**, a control space **U**, an observation space **Y**, an initial state, noise models in stochastic case
- We need to design a control law that minimizes the objective function

	Deterministic Model	**Stochastic Model**
Continuous time	$\dot{x} = f(x,u)$ $y = h(x,u)$ $J = \int_{t=0}^{T} \delta^t l_t(x,u,\dot{x})dt$	$\dot{x} = f(x,u,w)$ $y = h(x,u,v)$ $J = \int_{t=0}^{T} \delta^t l_t(x,u,\dot{x})dt$
Discrete time	$\boxed{\begin{aligned} x_{k+1} &= f(x_k,u_k) \\ y_k &= h(x_k,u_k) \\ J &= \sum_{k=0}^{N} \delta^k l_k(x_k,u_k,x_{k+1}) \end{aligned}}$	$x_{k+1} = f(x_k,u_k,w_k)$ $y_k = h(x_k,u_k,v_k)$ $J = \sum_{k=0}^{N} \delta^k l_k(x_k,u_k,x_{k+1})$

Discrete Time Optimal Control

- Problem formulation and examples
- Dynamic programming
- Maximum principle
- Linear Quadratic Regulator

Problem Formulation

- Deterministic discrete time
- State feedback
- Finite horizon
- No discount
- Controllable

$$\min_{u_0, u_1, \ldots, u_{N-1}} J = \sum_{k=0}^{N-1} l_k(x_k, u_k) + l_N(x_N)$$

$$x_{k+1} = f(x_k, u_k)$$

Problem Formulation

- Decision variables: the control sequences $\{u_0, \ldots, u_{N-1}\}$
- Known parameters:
  - Initial state: x_0
  - Run-time costs: l_k
  - Terminal cost: l_N

$$\min_{u_0, u_1, \ldots, u_{N-1}} J = \sum_{k=0}^{N-1} l_k(x_k, u_k) + l_N(x_N)$$

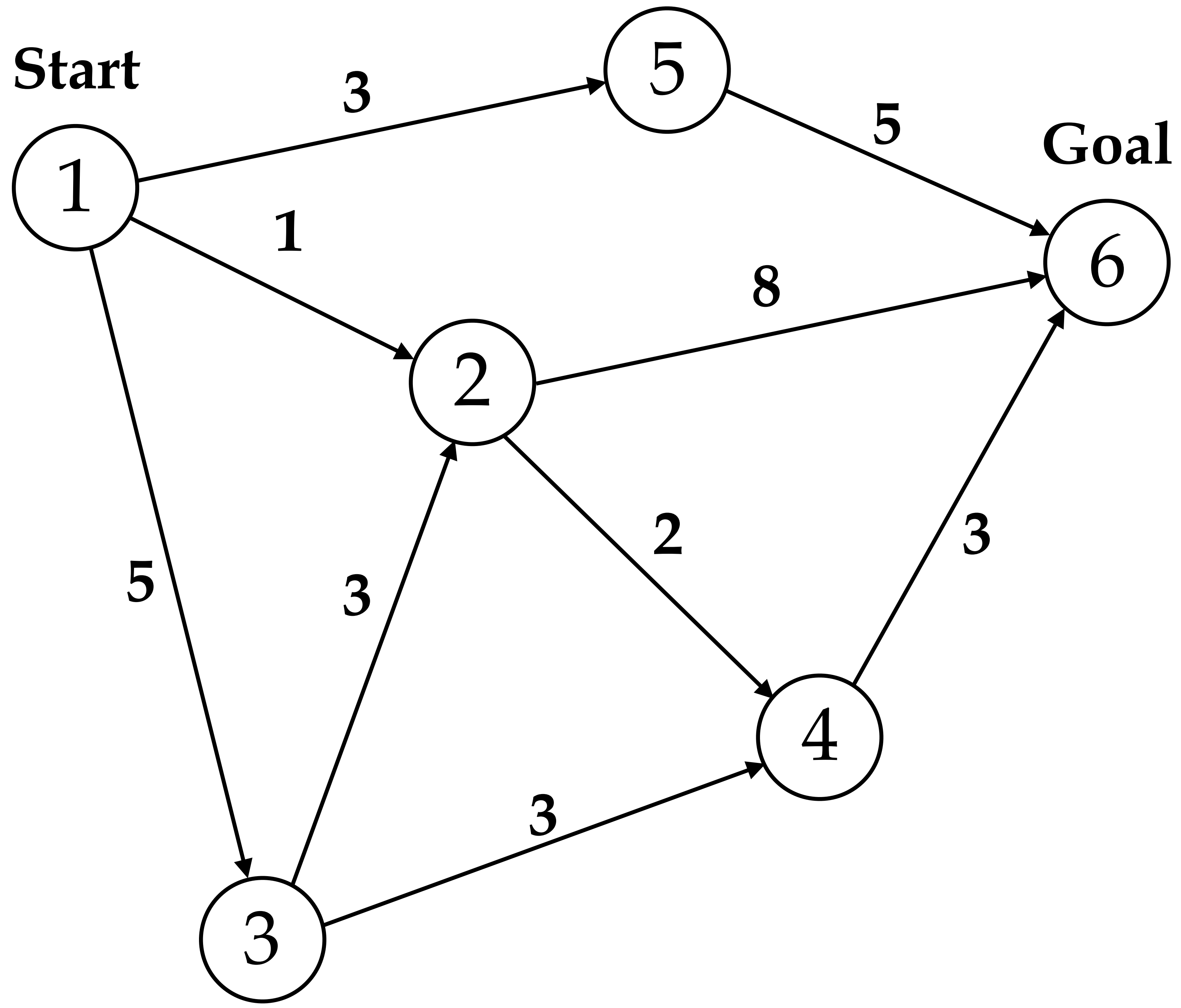
$$x_{k+1} = f(x_k, u_k)$$

Solutions

- Open-loop solution:
  - A sequence of future controls $\{u_0, \ldots, u_{N-1}\}$
- Closed-loop solution:
  - A control policy $u_k = q_k(x_k), \forall k$
  - More robust to disturbances

Example 1: Shortest Path Problem

- State space: nodes
- Action space: arcs
- Cost: distance traveled
- Termination: event triggered



Example 2: Regulation Problem

- Dynamic model (double integrator)

$$x_{k+1} = Ax_k + Bu_k$$

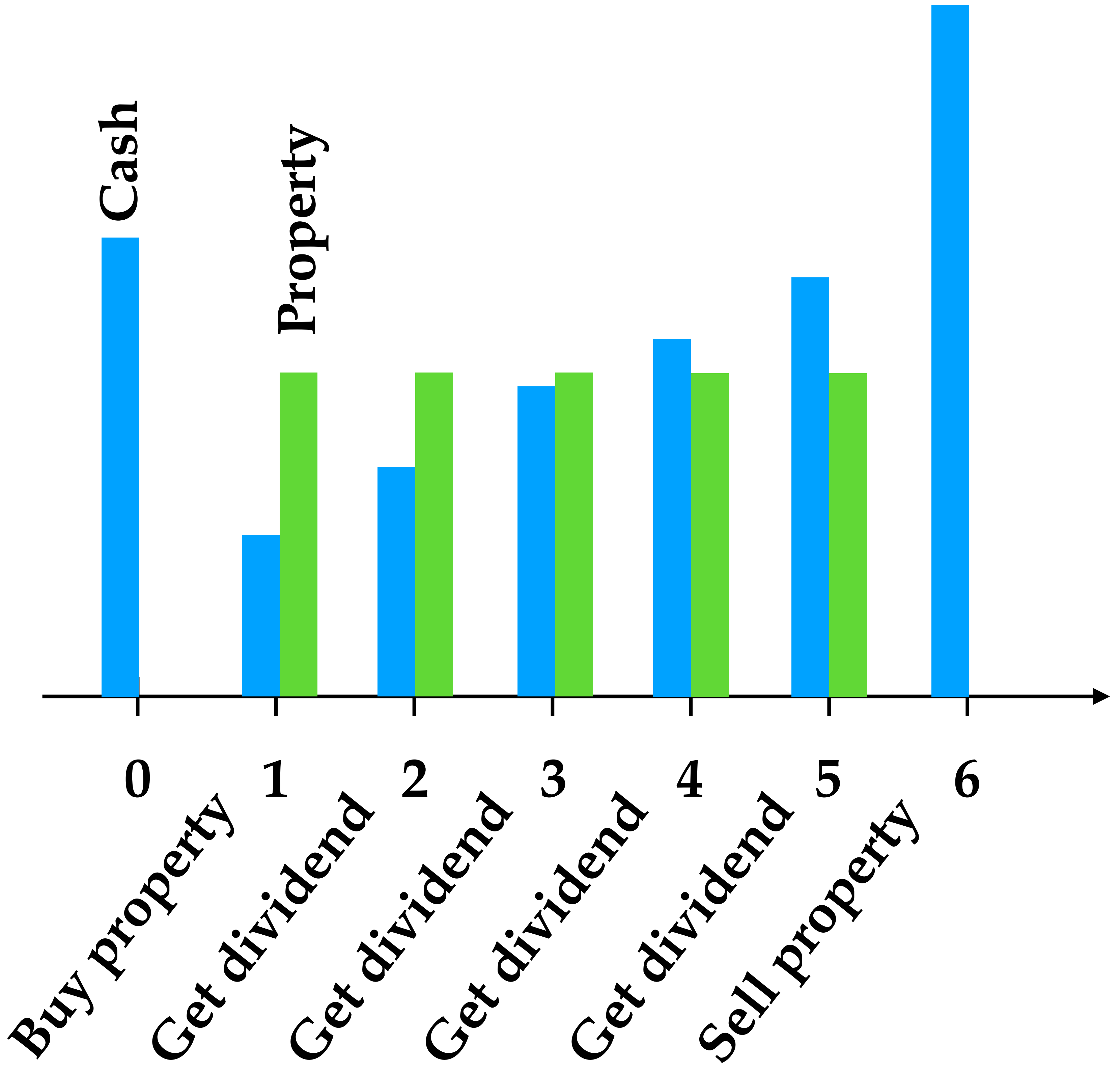
- Cost (quadratic)

$$J = \sum_{k=0}^{N-1} [x_k^T Q_k x_k + u_k^T R_k u_k] + x_N^T S_N x_N$$

- This is a linear quadratic regulation problem

Example 3: Investment Problem

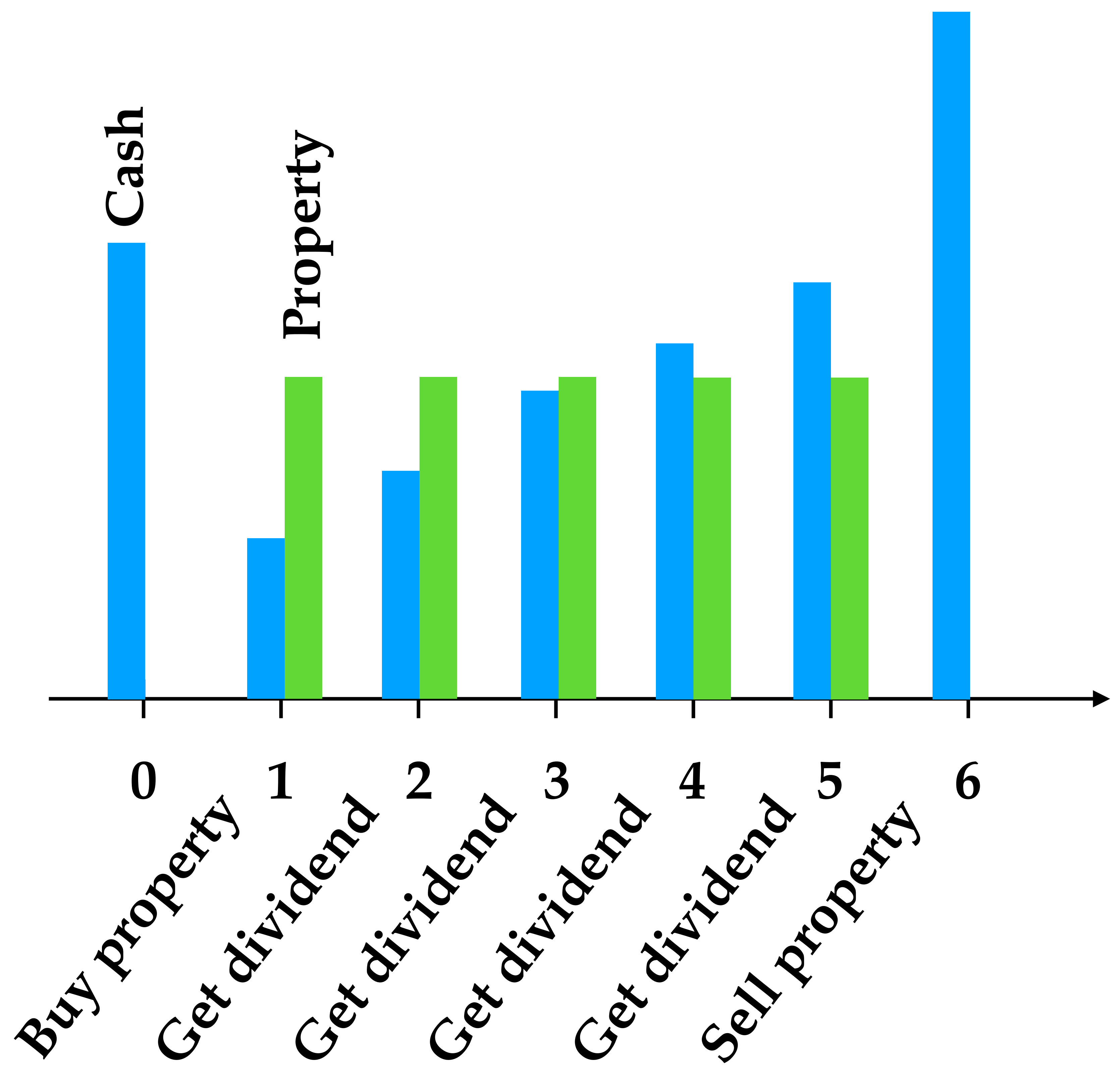
- I have \$m cash and want to maximize my cash 6 month later.
- There is a property A on the market with monthly dividend. A's price is floating and external to me (I cannot influence the price).
- I can choose to trade in/out property A at the beginning of every month.
- If I know the future prices of A, what is my trading strategy to maximize my cash 6 month later?



Example 3: Investment Problem

- State: the amount of cash and the amount of property $x \in \mathbb{R}^2$, with $x_0 = [m \quad 0]^T$
- Control: the amount of property bought $u \in [-1, 1]$
- Dynamics:

$$\begin{array}{ccc} & \text{Dividend rate at month k} & \text{Price at month k} \\ x_{k+1} = & \begin{bmatrix} 1 & r_k \\ 0 & 1 \end{bmatrix} x_k + & \begin{bmatrix} -p_k \\ 1 \end{bmatrix} u_k \end{array}$$



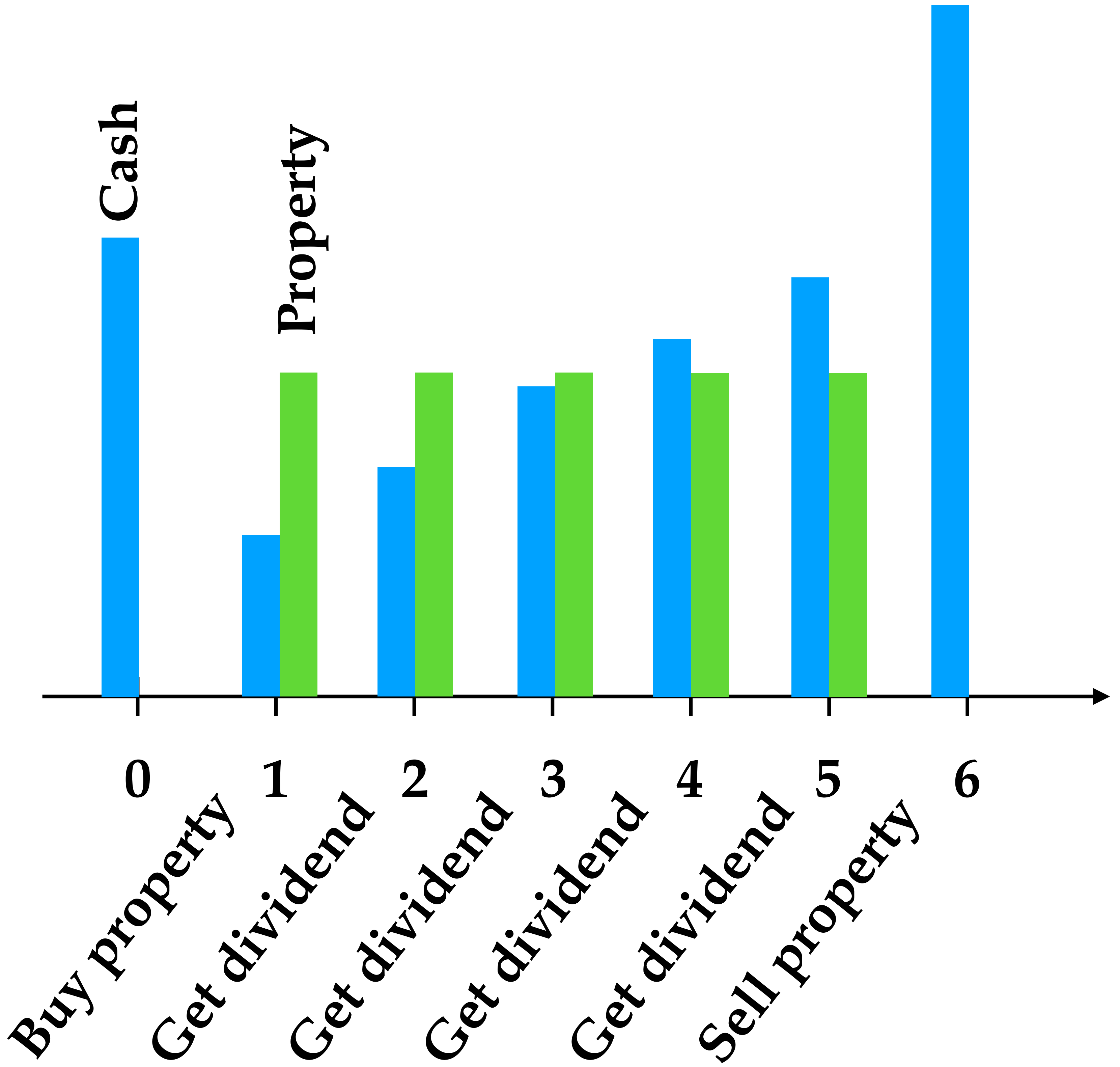
Example 3: Investment Problem

- No run-time cost
- We are minimizing

$$J = \begin{bmatrix} -1 & 0 \end{bmatrix} x_6$$

- Control constraints to ensure the problem is bounded

$$u \in [-1, 1]$$



Discrete Time Optimal Control

- Problem formulation and examples
- Dynamic programming
- Maximum principle
- Linear Quadratic Regulator

Optimal Solution

$$\min_{u_0,u_1,\ldots,u_{N-1}} J = \sum_{k=0}^{N-1} l_k(x_k,u_k) + l_N(x_N) \quad \longrightarrow \quad J_0(x_0;u_0,\ldots,u_{N-1})$$

$$x_{k+1} = f(x_k,u_k)$$

***We will not explicitly write this equality constraint if it is clear from the context.**

- Let $\{u_0^*,\ldots,u_{N-1}^*\}$ be an optimal control sequence
- Together with x_0, the optimal control sequence determines the corresponding state sequence $\{x_1^*,\ldots,x_N^*\}$
- The optimal cost corresponding to the optimal sequence is denoted $J^*(x_0)$

Principle of Optimality

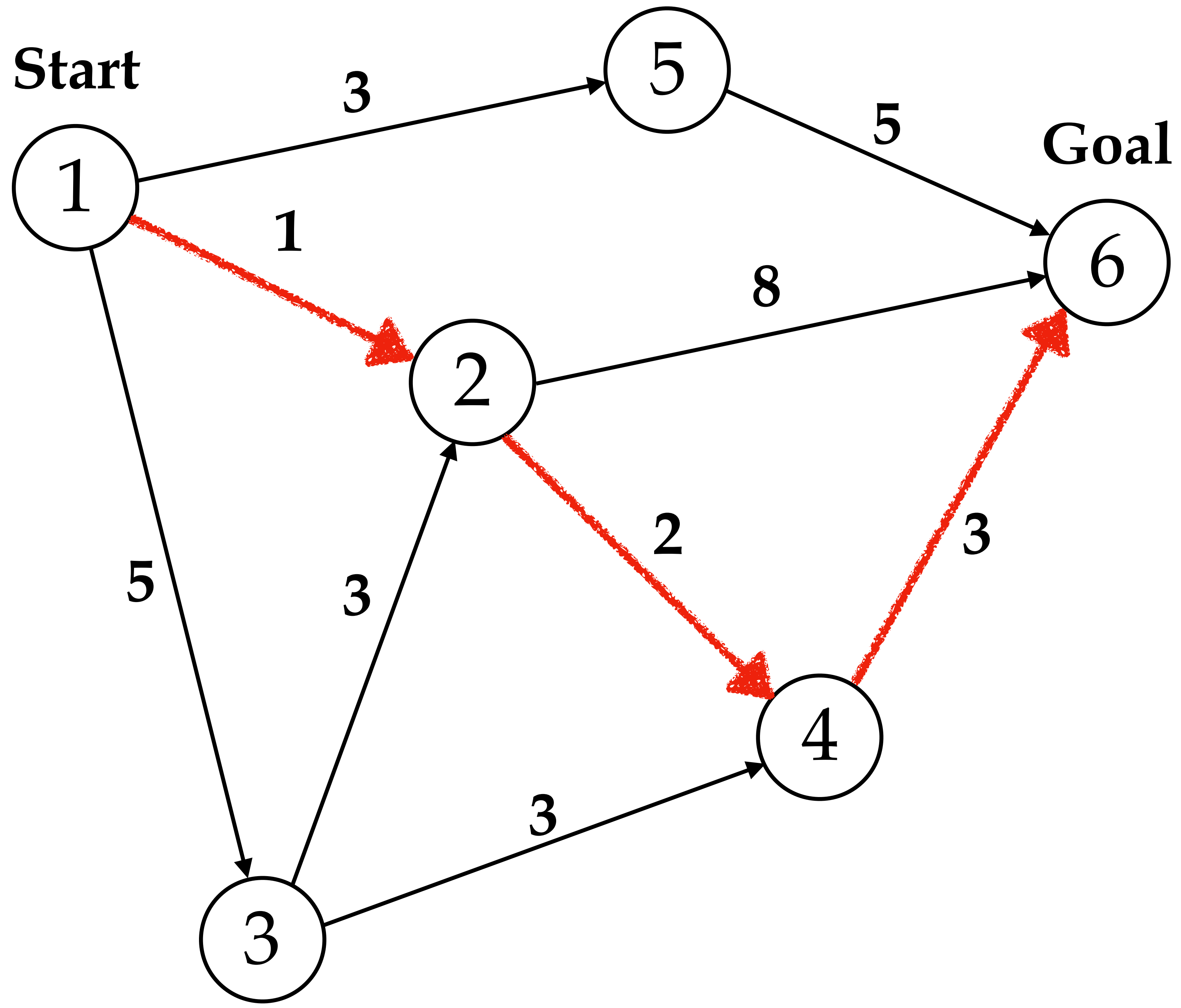
- Consider a subproblem: we start at x_k^* at time k and wish to minimize the "cost-to-go" from time k to time N over $\{u_k, \ldots, u_{N-1}\}$,

$$\min_{u_k,\ldots,u_{N-1}} l_k(x_k^*, u_k) + \sum_{m=k+1}^{N-1} l_m(x_m, u_m) + l_N(x_N) \quad \longrightarrow \quad J_k(x_k^*; u_k, \ldots, u_{N-1})$$

- Then the truncated optimal control sequence $\{u_k^*, \ldots, u_{N-1}^*\}$ is optimal for this subproblem.

Example: Shortest Path Problem

- 1-2-4-6 is the optimal path
- 2-4-6 is the optimal path from 2 to 6

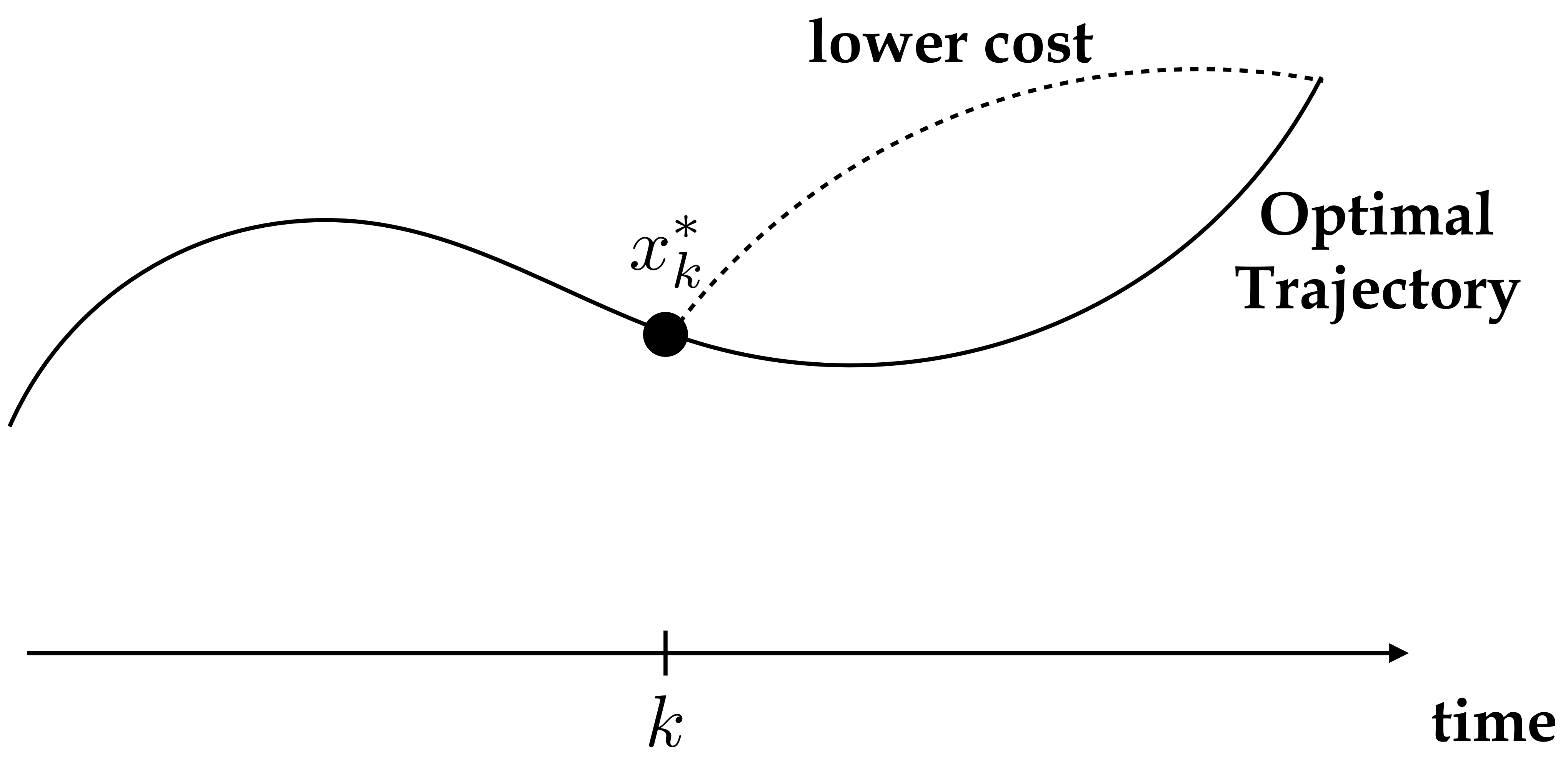


Proof by Contradiction

- Suppose the truncated control sequence $\{u_k^*, \ldots, u_{N-1}^*\}$ is not optimal for the subproblem.

- There exists another control sequence $\{u_k^o, \ldots, u_{N-1}^o\}$ that will result in lower cost of the subproblem.

$$J_k(x_k^*; u_k^*, \ldots, u_{N-1}^*) > J_k(x_k^*; u_k^o, \ldots, u_{N-1}^o)$$



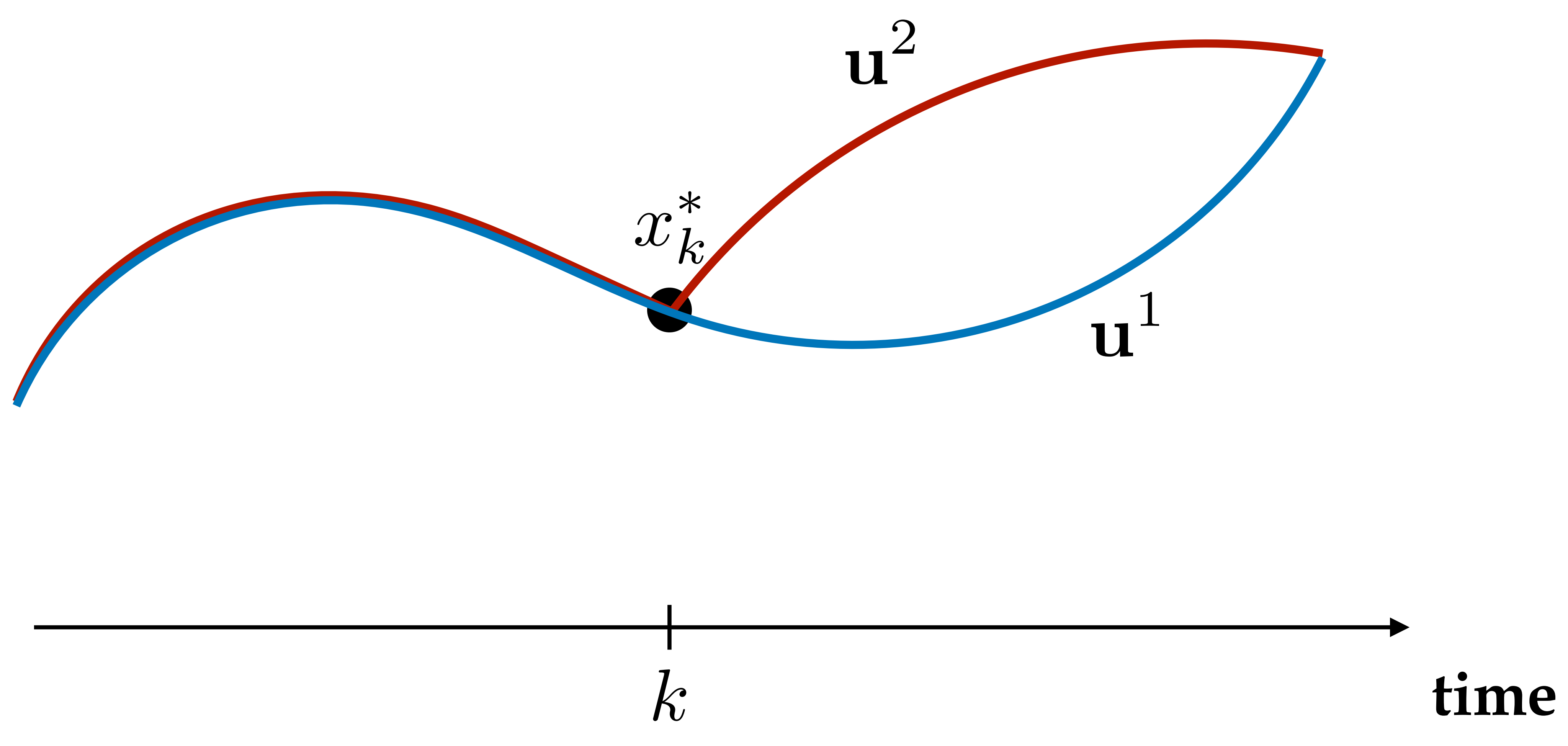
Proof by Contradiction

- Combining the optimal control sequence before k and the new control sequence starting at k: $\{u_0^*, \ldots, u_{k-1}^*, u_k^o, \ldots, u_{N-1}^o\}$

- Let's compare the total costs with respect to

  - $\mathbf{u}^2 = \{u_0^*, \ldots, u_{k-1}^*, u_k^o, \ldots, u_{N-1}^o\}$

  - $\mathbf{u}^1 = \{u_0^*, \ldots, u_{N-1}^*\}$



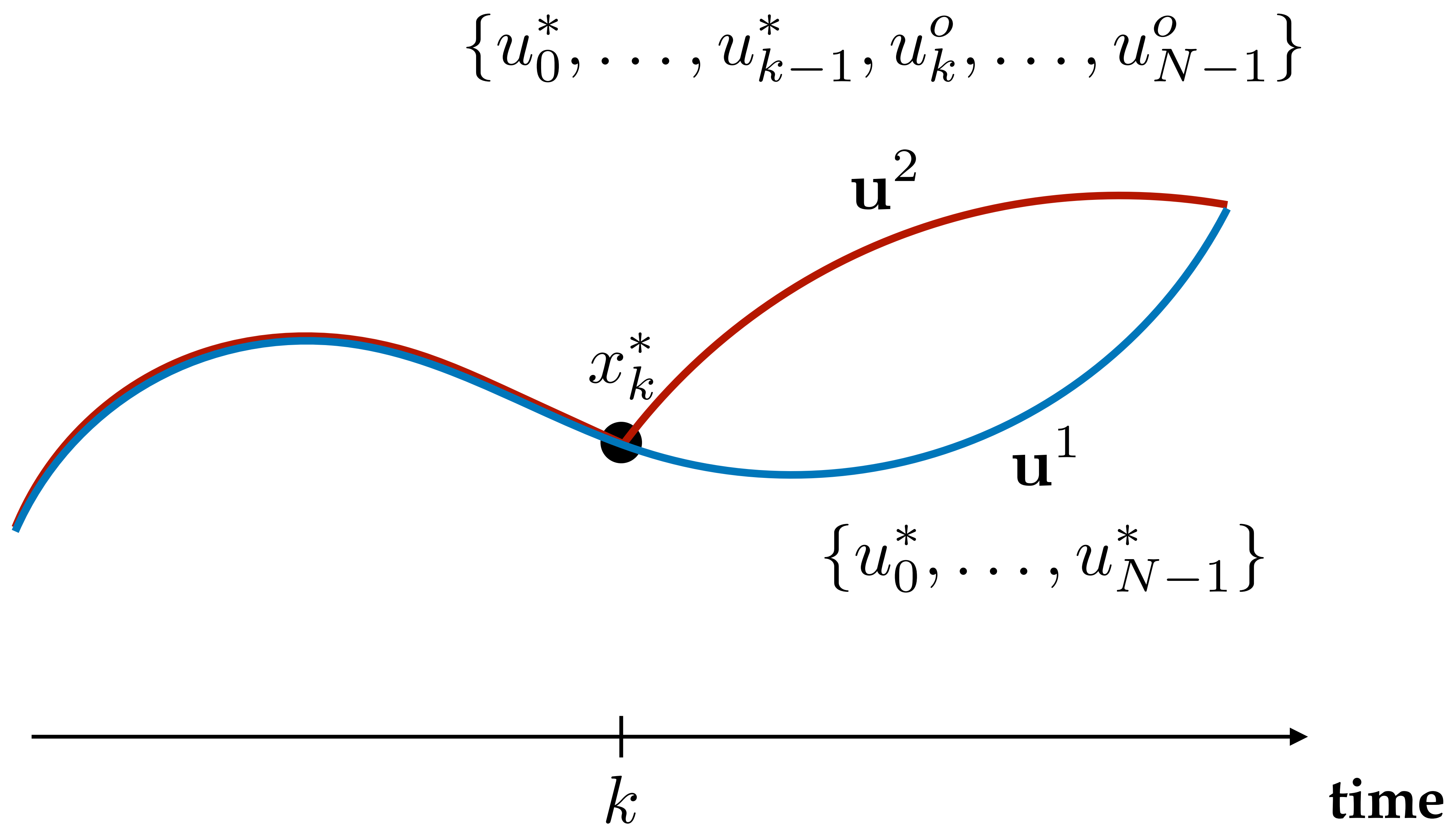
Proof by Contradiction

$$J_0(x_0; u_0, \ldots, u_{N-1}) = \sum_{m=0}^{k-1} l_m(x_m, u_m) + J_k(x_k; u_k, \ldots, u_{N-1})$$

$$J_0(x_0; \mathbf{u}^1) = \sum_{m=0}^{k-1} l_m(x_m^*, u_m^*) + J_k(x_k^*; u_k^*, \ldots, u_{N-1}^*)$$

$$> \sum_{m=0}^{k-1} l_m(x_m^*, u_m^*) + J_k(x_k^*; u_k^o, \ldots, u_{N-1}^o)$$

$$= J_0(x_0; \mathbf{u}^2)$$



Contradicts with the fact that $\mathbf{u}^1$ is an optimal control sequence!

Principle of Optimality

- Also known as Bellman Principle
- In plain word:
  - Any segment of an optimal trajectory is still optimal
  - An optimal trajectory won't have any suboptimal components
- The inverse is not true:
  - The concatenation of optimal trajectories may not be optimal

Utilizing Principle of Optimality

$$J_0(x_0; u_0, \ldots, u_{N-1}) = \sum_{m=0}^{k-1} l_m(x_m, u_m) + J_k(x_k; u_k, \ldots, u_{N-1})$$

Take minimization on both sides

$$\min_{u_0,\ldots,u_{N-1}} J_0(x_0; u_0, \ldots, u_{N-1}) = \min_{u_0,\ldots,u_{k-1}} \left\{ \sum_{m=0}^{k-1} l_m(x_m, u_m) + \min_{u_k,\ldots,u_{N-1}} J_k(\boxed{x_k}; u_k, \ldots, u_{N-1}) \right\}$$

This term depends on $\{u_0, \ldots, u_{k-1}\}$

These are not two independent minimization problems

But they can be solved sequentially

Utilizing Principle of Optimality

$$\underbrace{\min_{u_0,\ldots,u_{N-1}} J_0(x_0; u_0, \ldots, u_{N-1})}_{J_0^*(x_0)} = \min_{u_0,\ldots,u_{k-1}} \left\{ \sum_{m=0}^{k-1} l_m(x_m, u_m) + \underbrace{\min_{u_k,\ldots,u_{N-1}} J_k(x_k; u_k, \ldots, u_{N-1})}_{J_k^*(x_k)\ \text{optimal cost-to-go}} \right\}$$

Let $k = 1$

$$J_0^*(x_0) = \min_{u_0} \{l_0(x_0, u_0) + J_1^*(x_1)\}$$

By induction

$$J_k^*(x_k) = \min_{u_k} \{l_k(x_k, u_k) + J_{k+1}^*(x_{k+1})\} \qquad \forall k = 0, 1, \ldots, N-1$$

$$J_N^*(x_N) = l_N(x_N)$$

Cost-to-Go and Value Function

Cost-to-go at time k: a function that depends on the current state and all future control inputs

$$J_k(x_k; u_k, \ldots, u_{N-1}) = \sum_{m=k}^{N-1} l_m(x_m, u_m) + l_N(x_N)$$

Value function (optimal cost-to-go) at time k: a function that only depends on the current state (assuming optimal actions will be taken in the future)

$$J_k^*(x_k) = \min_{u_k, \ldots, u_{N-1}} J_k(x_k; u_k, \ldots, u_{N-1})$$

Bellman Equation

$$J_k^*(x_k) = \min_{u_k} \left\{ l_k(x_k, u_k) + J_{k+1}^*(x_{k+1}) \right\}$$

- Bellman equation is an update rule for value functions under the principle of optimality.
- General form:
  - Optimize over action
  - Optimization objective: run-time cost plus the value at next state
- We will see many variations of the Bellman equations in different scenarios.

Dynamic Programing

- Solving the optimal control sequence recursively using the principle of optimality
- At time N, we have

$$J_N^*(x_N) = l_N(x_N)$$

- For $k = N-1, N-2, \ldots, 1, 0$, we solve the following optimization

$$J_k^*(x_k) = \min_{u_k} \left\{ l_k(x_k, u_k) + J_{k+1}^*(x_{k+1}) \right\}$$

and obtain the optimal control policy

$$q_k^*(x_k) = \arg\min_{u_k} \left\{ l_k(x_k, u_k) + J_{k+1}^*(x_{k+1}) \right\}$$

Dynamic Programing

- Through this backward pass, we can only obtain the control policy as a mapping from the state to the control.

- The optimal control sequence needs to be obtained in a forward pass considering the initial state x_0

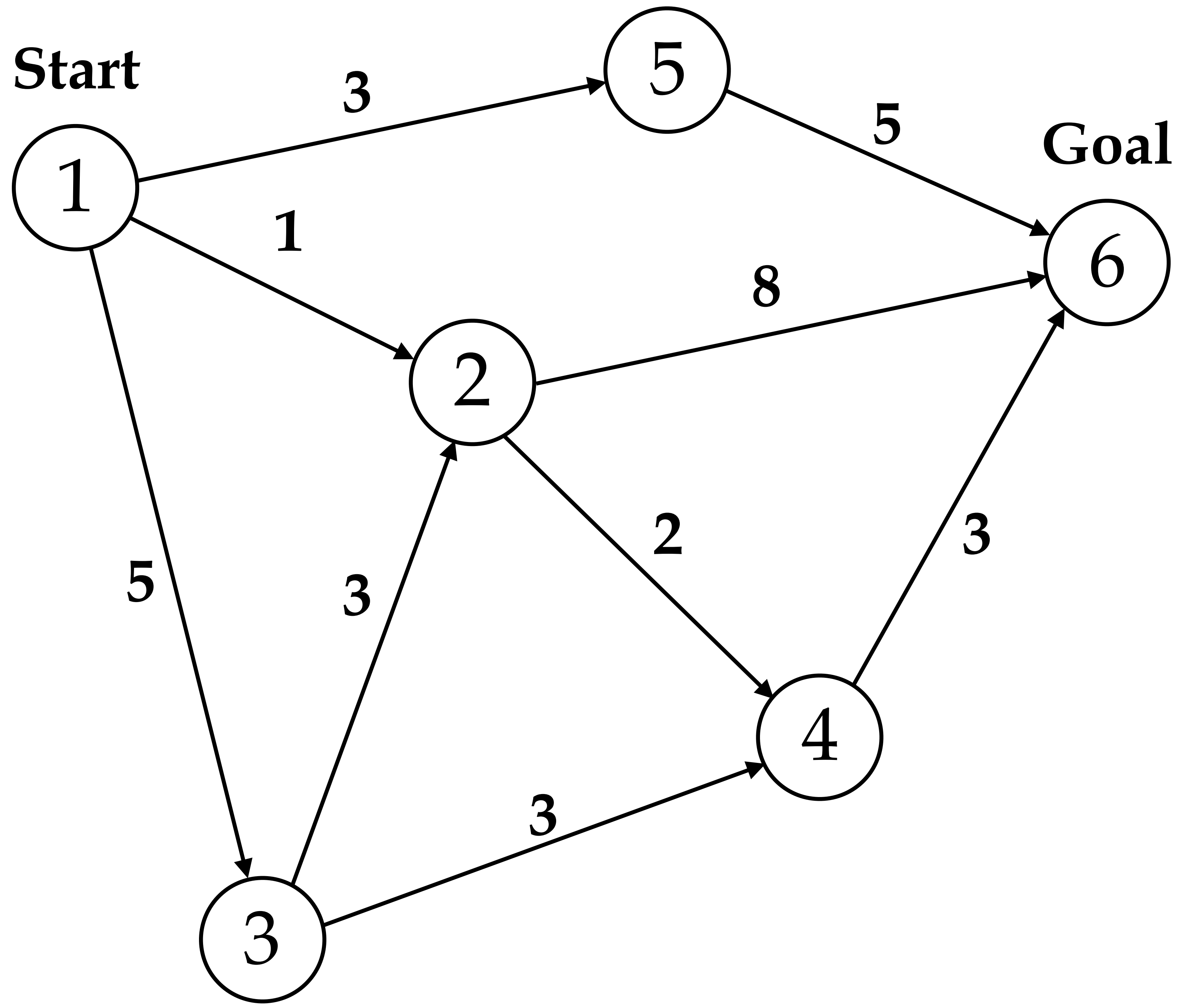
$$
\begin{aligned}
x_0^* &= x_0 \\
u_k^* &= q_k^*(x_k^*), \forall k = 0, 1, \ldots, N-1 \\
x_{k+1}^* &= f(x_k^*, u_k^*), \forall k = 0, 1, \ldots, N-1
\end{aligned}
$$

Example: Shortest Path Problem

Backward pass

Optimal cost-to-go	Optimal action
$J^*(6) = 0$	
$J^*(5) = 5 + J^*(6) = 5$	5->6
$J^*(4) = 3 + J^*(6) = 3$	4->6
$J^*(2) = \min\{8 + J^*(6), 2 + J^*(4)\} = 5$	2->4
$J^*(3) = \min\{3 + J^*(2), 3 + J^*(4)\} = 6$	3->4
$J^*(1) = \min\{1 + J^*(2), 5 + J^*(3)\} = 6$	1->2

Forward pass



Discrete Time Optimal Control

- Problem formulation and examples
- Dynamic programming
- Maximum principle
- Linear Quadratic Regulator

Parameterization of the Value Function

- Bellman equation:

$$J_k^*(x_k) = \min_{u_k} \left\{ l_k(x_k, u_k) + J_{k+1}^*(x_{k+1}) \right\}$$

- Linear parameterization with co-state λ_k (gradient)

$$J_k^*(x_k) = \lambda_k^T x_k + \gamma_k$$

- Relationship between co-states

$$\lambda_k^T x_k + \gamma_k = \min_{u_k} \left\{ l_k(x_k, u_k) + \lambda_{k+1}^T f(x_k, u_k) + \gamma_{k+1} \right\}$$

Hamiltonian

$$\lambda_k^T x_k + \gamma_k = \min_{u_k} \{\underbrace{l_k(x_k, u_k) + \lambda_{k+1}^T f(x_k, u_k)}_{H_k(x_k, u_k, \lambda_{k+1})} + \gamma_{k+1}\}$$

Hamiltonian

- Hamiltonian is equivalent to cost-to-go.

$$J_k(x_k; u_k, \underline{u_{k+1} \ldots, u_{N-1}}) = l_k(x_k, u_k) + \underline{\sum_{m=k+1}^{N-1} l_m(x_m, u_m) + l_N(x_N)}$$

$$H_k(x_k, u_k, \underline{\lambda_{k+1}}) = l_k(x_k, u_k) + \underline{\lambda_{k+1}^T f(x_k, u_k)}$$

- The future decision variables are considered by the co-state.

Dynamic Programming with Co-States

$$\lambda_k^T x_k + \gamma_k = \min_{u_k} \{ \underbrace{l_k(x_k, u_k) + \lambda_{k+1}^T f(x_k, u_k)}_{H_k(x_k, u_k, \lambda_{k+1})} + \gamma_{k+1} \}$$

Hamiltonian

- Assume all functions are smooth
- The optimal solution of the bellman equation above should satisfy

$$\lambda_k = \frac{\partial}{\partial x} H_k(x_k, u_k, \lambda_{k+1}) = \frac{\partial}{\partial x} l_k(x_k, u_k) + \left[\frac{\partial}{\partial x} f(x_k, u_k) \right]^T \lambda_{k+1}$$

$$u_k = \arg\min_u H_k(x_k, u, \lambda_{k+1})$$

Lemma on Partial Derivatives*

- Lemma (Partial Derivative): Consider the following optimization

$$V^*(a) = \min_b V(a, b)$$

then the partial derivatives satisfy

$$\frac{d}{da} V^*(a) = \frac{\partial}{\partial a} V(a, b^*)$$

where $b^* = \arg\min V(a, b)$.

- Proof: $\frac{d}{da} V^*(a) = \frac{d}{da} V(a, b^*) = \frac{\partial}{\partial a} V(a, b^*) + \frac{\partial}{\partial b} V(a, b^*) \frac{db^*}{da}$ and $\frac{\partial}{\partial b} V(a, b^*) \frac{db^*}{da} = 0$

Dynamic Programming with Co-States

- At time N, we find a co-state function $\lambda_N(x_N)$ and a constant γ_N that satisfies

$$\lambda_N(x_N) + \gamma_N = l_N(x_N)$$

- For $k = N-1, N-2, \ldots, 1, 0$, we solve the following equations

$$\lambda_k = \frac{\partial}{\partial x} H_k(x_k, u_k, \lambda_{k+1}) = \frac{\partial}{\partial x} l_k(x_k, u_k) + \left[\frac{\partial}{\partial x} f(x_k, u_k)\right]^T \lambda_{k+1}$$

maybe a function on $x_{k+1} = f(x_k, u_k)$

$$u_k = \arg\min_u H_k(x_k, u, \lambda_{k+1})$$

to obtain the optimal control policy $u_k = q_k^*(x_k)$ and co-state function $\lambda_k(x_k)$. The constant term should also be computed from the Bellman equation.

Dynamic Programing with Co-States

- The optimal control sequence and co-states can be obtained in a forward pass considering the initial state x_0

$$\begin{aligned} x_0^* &= x_0 \\ u_k^* &= q_k^*(x_k^*), \forall k = 0, 1, \ldots, N-1 \\ \lambda_k^* &= \lambda_k(x_k^*), \forall k = 0, 1, \ldots, N \\ x_{k+1}^* &= f(x_k^*, u_k^*), \forall k = 0, 1, \ldots, N-1 \end{aligned}$$

How to Interpret the Co-State?

- We introduced co-state as one way to parameterize the value function and define it as the gradient of the value function.
- It can also be interpreted as:
  - Shadow price
  - Lagrange multiplier

Co-State and Shadow Price

$$J_k^*(x_k) = \lambda_k^T x_k + \gamma_k$$

- Shadow price
  - The "unit cost" of the current state value
  - Will be illustrated later in the investment problem

Co-State and Lagrange Multiplier*

- The optimization problem (primal):

$$\min_{u_k, x_{k+1}=f(x_k,u_k)} \{l_k(x_k, u_k) + J^*_{k+1}(x_{k+1})\}$$

- Lagrangian function with multiplier

$$L(x_k, u_k, x_{k+1}, \lambda_{k+1}) = l_k(x_k, u_k) + J^*_{k+1}(x_{k+1}) - \lambda^T_{k+1}(x_{k+1} - f(x_k, u_k))$$

- Lagrangian dual function

$$L^*(x_k, \lambda_{k+1}) = \min_{u_k, x_{k+1}} L(x_k, u_k, x_{k+1}, \lambda_{k+1})$$

Co-State and Lagrange Multiplier*

- The maximization of the dual problem provides a lower bound to the solution of the primal problem

$$\max_{\lambda_{k+1}} \underbrace{\min_{u_k, x_{k+1}} \{l_k(x_k, u_k) + J^*_{k+1}(x_{k+1}) - \lambda^T_{k+1}(x_{k+1} - f(x_k, u_k))\}}_{L^*(x_k, \lambda_{k+1})} \leq \min_{u_k, x_{k+1} = f(x_k, u_k)} \{l_k(x_k, u_k) + J^*_{k+1}(x_{k+1})\}$$

- The difference is called the duality gap.

- Intuitively, the problem is relaxed in its dual form. Hence it would be easier to find a smaller value.

Co-State and Lagrange Multiplier*

$$\max_{\lambda_{k+1}} \min_{u_k, x_{k+1}} \{l_k(x_k, u_k) + J^*_{k+1}(x_{k+1}) - \lambda^T_{k+1}(x_{k+1} - f(x_k, u_k))\} \leq \min_{u_k, x_{k+1} = f(x_k, u_k)} \{l_k(x_k, u_k) + J^*_{k+1}(x_{k+1})\}$$

$$\max_{\lambda_{k+1}} \left\{ \min_{u_k} \{\underbrace{l_k(x_k, u_k) + \lambda^T_{k+1} f(x_k, u_k)}_{\text{Hamiltonian}}\} + \min_{x_{k+1}} \underline{\{J^*_{k+1}(x_{k+1}) - \lambda^T_{k+1} x_{k+1}\}} \right\}$$

According to the first-order optimality condition, the optimal solution $x^*_{k+1}, \lambda^*_{k+1}$ should satisfy

$$\frac{\partial}{\partial x} J^*_{k+1}(x^*_{k+1}) = \lambda^*_{k+1}$$

This justifies the linear parameterization

Maximum Principle

- Theorem:
  - If $\{u_0^*, \ldots, u_{N-1}^*\}$ and $\{x_1^*, \ldots, x_N^*\}$ are the optimal state and control trajectories starting at x_0, then there exists a co-state trajectory $\{\lambda_1^*, \ldots, \lambda_N^*\}$ with $\lambda_N^* = \frac{\partial}{\partial x_N} J_N^*(x_N^*)$ satisfying

$$x_{k+1}^* = \frac{\partial}{\partial \lambda} H_k(x_k^*, u_k^*, \lambda_{k+1}^*) = f(x_k^*, u_k^*)$$

$$\lambda_k^* = \frac{\partial}{\partial x} H_k(x_k^*, u_k^*, \lambda_{k+1}^*) = \frac{\partial}{\partial x} l_k(x_k^*, u_k^*) + \left[\frac{\partial}{\partial x} f(x_k^*, u_k^*)\right]^T \lambda_{k+1}^*$$

$$u_k^* = \arg\min_u H_k(x_k^*, u, \lambda_{k+1}^*)$$

Maximum Principle

- It is a necessary condition, not a sufficient condition.
- The solutions we obtained using “dynamic programming with co-states” always satisfy the maximum principle.

Example: Investment Problem

- State: the amount of cash and the amount of property $x \in \mathbb{R}^2$, with $x_0 = [m \quad 0]^T$

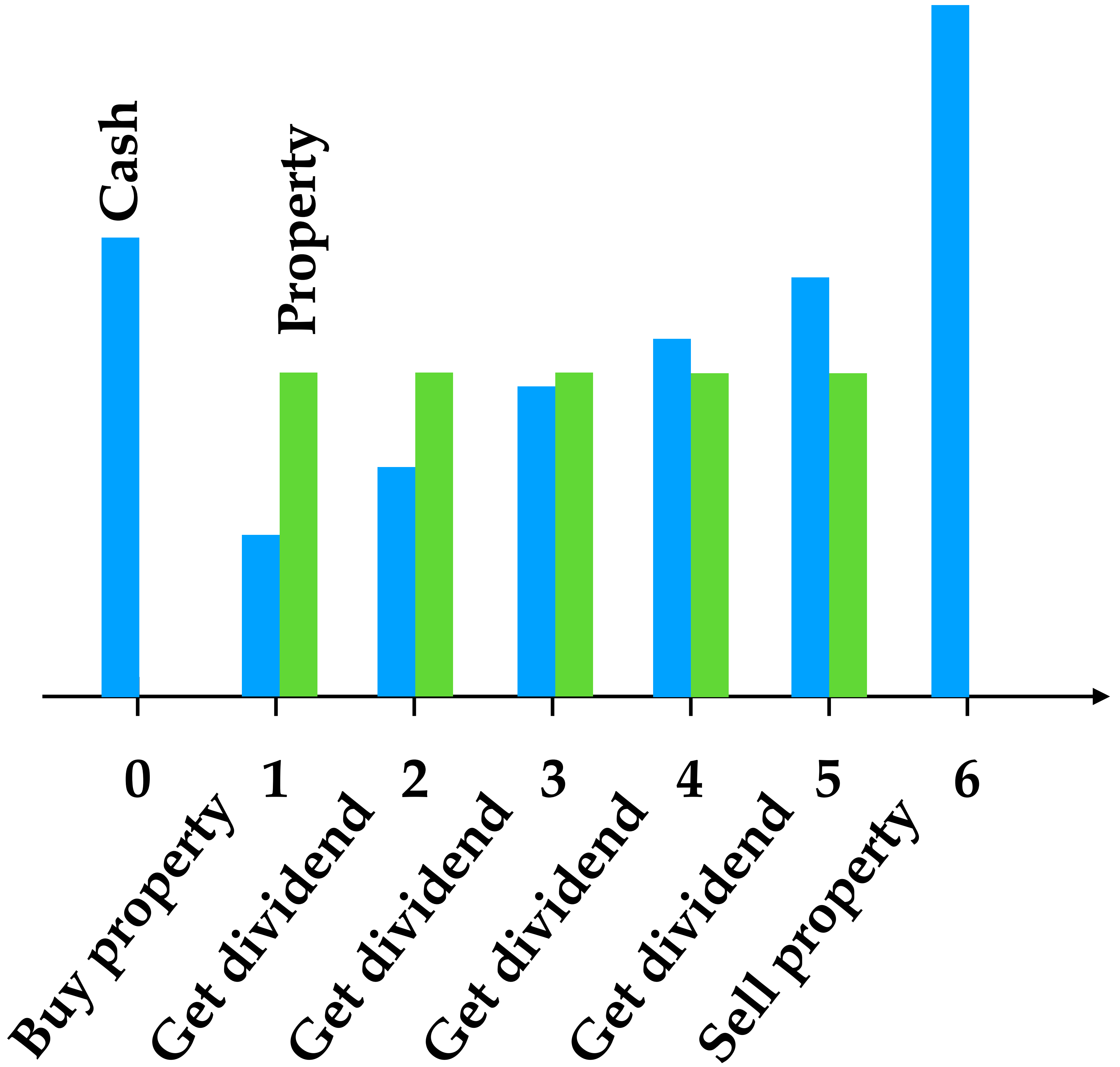
- Control: the amount of property bought $u \in [-1, 1]$

- Dynamics:

$$x_{k+1} = \begin{bmatrix} 1 & r_k \\ 0 & 1 \end{bmatrix} x_k + \begin{bmatrix} -p_k \\ 1 \end{bmatrix} u_k$$

- Objective

$$J = \begin{bmatrix} -1 & 0 \end{bmatrix} x_6$$



Example: Investment Problem

- Value function: $J_k^*(x_k) = \lambda_k^T x_k + \gamma_k$
- Since there is no run-time cost, $J_k^*(x_k) = J_6^*(x_6)$
- Shadow price: How much does cash/property **worth** at month k with respect to the objective function?
- For example, $\lambda_6 = [-1 \ 0]^T$ means that property worths nothing at month 6.
- We need to solve for the shadow price at every stage.

Example: Investment Problem

- Hamiltonian

State =
[amount of cash;
amount of property]

$$H_k = \lambda_{k+1}^T \left\{ \begin{bmatrix} 1 & r_k \\ 0 & 1 \end{bmatrix} x_k + \begin{bmatrix} -p_k \\ 1 \end{bmatrix} u_k \right\}$$

Co-state =
[shadow price of cash;
shadow price of property]

The shadow price of cash does not change throughout time

$$\lambda_k = \frac{\partial}{\partial x} H_k(x_k, u_k, \lambda_{k+1}) \quad \Rightarrow \quad \lambda_k = \begin{bmatrix} 1 & 0 \\ r_k & 1 \end{bmatrix} \lambda_{k+1}$$

The shadow price of property is higher going backward in time, due to dividend payoff in the future.

Example: Investment Problem

- Hamiltonian

State = [amount of cash; amount of property]

Co-state = [shadow price of cash; shadow price of property]

$$H_k = \lambda_{k+1}^T \left\{ \begin{bmatrix} 1 & r_k \\ 0 & 1 \end{bmatrix} x_k + \begin{bmatrix} -p_k \\ 1 \end{bmatrix} u_k \right\}$$

$$u_k = \arg\min_u H_k(x_k, u, \lambda_{k+1}) \Rightarrow u_k = \begin{cases} -1 & [-p_k \;\; 1]\lambda_{k+1} \geq 0 \\ 1 & [-p_k \;\; 1]\lambda_{k+1} < 0 \end{cases}$$

The payoff in the future if we buy one unit of property:
we will get one more unit of property, hence gain $\lambda_{k+1,2}$ cost
we will use p_k amount of cash, hence reduce $p_k\lambda_{k+1,1}$ cost

Control strategy: buy property if it costs less, otherwise sell

Example: Investment Problem

- Control strategy:

$$u_k = \begin{cases} -1 & [-p_k \;\; 1]\lambda_{k+1} \geq 0 \\ 1 & [-p_k \;\; 1]\lambda_{k+1} < 0 \end{cases}$$

- Buy property if it costs less, otherwise sell
- This is a bang-bang control
- The switch depends on the future perspective of the payoff, i.e., the shadow price

Example: Investment Problem

$$\lambda_k = \begin{bmatrix} 1 & 0 \\ r_k & 1 \end{bmatrix} \lambda_{k+1} \qquad u_k = \begin{cases} -1 & [-p_k \;\; 1]\lambda_{k+1} \geq 0 \\ 1 & [-p_k \;\; 1]\lambda_{k+1} < 0 \end{cases}$$

Month	Dividend rate	Property price	Co-state (shadow price)	Action
6			[-1, 0]	
5	1	1	[-1, -1]	-1
4	0	2	[-1, -1]	-1
3	1	3	[-1, -2]	-1
2	0	3	[-1, -2]	-1
1	1	2	[-1, -3]	1
0	0	1	[-1, -3]	1

Example: Investment Problem

- This is a constrained linear time-varying problem.
- Solutions to linear optimal control problems are usually bang-bang control.
- In linear optimal control problems, the co-state usually does not depend on the state. But in linear quadratic problems, the co-state depends on the state.
- It is assumed that we know the future dynamics, i.e., future price and dividend rate. In the case that the dynamics are unknown, the problem becomes stochastic, which requires adaptive strategies.

Discrete Time Optimal Control

- Problem formulation and examples
- Dynamic programming
- Maximum principle
- Linear Quadratic Regulator

Linear Quadratic Regulator

$$J = \frac{1}{2}\sum_{k=0}^{N-1}[x_k^T Q_k x_k + u_k^T R_k u_k] + \frac{1}{2}x_N^T S_N x_N$$

$$x_{k+1} = A_k x_k + B_k u_k$$

$Q_0, \ldots, Q_{N-1}\ \ S_N$	symmetric positive semi-definite
$R_0, \ldots, R_{N-1}$	symmetric positive definite
A_k, B_k	controllable

Linear Quadratic Regulator

$$J = \frac{1}{2}\sum_{k=0}^{N-1}[x_k^T Q_k x_k + u_k^T R_k u_k] + \frac{1}{2}x_N^T S_N x_N$$

$$x_{k+1} = A_k x_k + B_k u_k$$

Hamiltonian

$$H_k(x_k, u_k, \lambda_{k+1}) = \frac{1}{2}x_k^T Q_k x_k + \frac{1}{2}u_k^T R_k u_k + \lambda_{k+1}^T (A_k x_k + B_k u_k)$$

Maximum principle

$$\lambda_k = \frac{\partial}{\partial x} H_k(x_k, u_k, \lambda_{k+1}) = Q_k x_k + A_k^T \lambda_{k+1}$$
$$0 = \frac{\partial}{\partial u} H_k(x_k, u_k, \lambda_{k+1}) = R_k u_k + B_k^T \lambda_{k+1}$$

Linear Quadratic Regulator

$$J = \frac{1}{2}\sum_{k=0}^{N-1}[x_k^T Q_k x_k + u_k^T R_k u_k] + \frac{1}{2}x_N^T S_N x_N$$

$$x_{k+1} = A_k x_k + B_k u_k$$

Boundary condition

$$J_N^*(x_N) = \frac{1}{2}x_N^T S_N x_N \qquad \lambda_N = \frac{\partial}{\partial x}J_N^*(x_N) = S_N x_N \qquad \gamma_N = -\frac{1}{2}x_N^T S_N x_N$$

From N to N-1 step

$$\lambda_{N-1} = Q_{N-1}x_{N-1} + A_{N-1}^T\lambda_N = Q_{N-1}x_{N-1} + A_{N-1}^T S_N A_{N-1}x_{N-1} + A_{N-1}^T S_N B_{N-1}u_{N-1}$$
$$0 = R_{N-1}u_{N-1} + B_{N-1}^T\lambda_N = R_{N-1}u_{N-1} + B_{N-1}^T S_N A_{N-1}x_{N-1} + B_{N-1}^T S_N B_{N-1}u_{N-1}$$

From N to N-1: Control Law

$$\lambda_{N-1} = Q_{N-1}x_{N-1} + A_{N-1}^T\lambda_N = Q_{N-1}x_{N-1} + A_{N-1}^T S_N A_{N-1}x_{N-1} + A_{N-1}^T S_N B_{N-1}u_{N-1}$$

$$0 = R_{N-1}u_{N-1} + B_{N-1}^T\lambda_N = \underline{R_{N-1}u_{N-1} + B_{N-1}^T S_N A_{N-1}x_{N-1} + B_{N-1}^T S_N B_{N-1}u_{N-1}}$$

$$0 = [B_{N-1}^T S_N B_{N-1} + R_{N-1}]u_{N-1} + B_{N-1}^T S_N A_{N-1}x_{N-1}$$

$$u_{N-1} = -\underbrace{[B_{N-1}^T S_N B_{N-1} + R_{N-1}]^{-1} B_{N-1}^T S_N A_{N-1}}_{K_{N-1}} x_{N-1}$$

*** We can take the inverse due to positive definiteness assumptions**

From N to N-1: Co-State

$$\lambda_{N-1} = Q_{N-1}x_{N-1} + A_{N-1}^T\lambda_N = \underline{Q_{N-1}x_{N-1} + A_{N-1}^T S_N A_{N-1}x_{N-1} + A_{N-1}^T S_N B_{N-1}u_{N-1}}$$

$$u_{N-1} = -\underbrace{[B_{N-1}^T S_N B_{N-1} + R_{N-1}]^{-1} B_{N-1}^T S_N A_{N-1}}_{K_{N-1}} x_{N-1}$$

$$\lambda_{N-1} = Q_{N-1}x_{N-1} + A_{N-1}^T S_N A_{N-1}x_{N-1} - A_{N-1}^T S_N B_{N-1}K_{N-1}x_{N-1}$$

$$\lambda_{N-1} = \underbrace{[Q_{N-1} + A_{N-1}^T S_N A_{N-1} - A_{N-1}^T S_N B_{N-1}K_{N-1}]}_{P_{N-1}} x_{N-1}$$

$$P_{N-1} = Q_{N-1} + A_{N-1}^T P_N A_{N-1} - A_{N-1}^T P_N B_{N-1}[B_{N-1}^T P_N B_{N-1} + R_{N-1}]^{-1} B_{N-1}^T P_N A_{N-1}$$

$$P_N := S_N$$

From N to N-1: Value Function

$$J^*_{N-1}(x_{N-1}) = \frac{1}{2}x^T_{N-1}Q_{N-1}x_{N-1} + \frac{1}{2}u^T_{N-1}R_{N-1}u_{N-1} + \frac{1}{2}x^T_N P_N x_N$$

$$\Downarrow$$

$$\begin{aligned} J^*_{N-1}(x_{N-1}) =& \frac{1}{2}x^T_{N-1}\underline{Q_{N-1}}x_{N-1} + \frac{1}{2}x^T_{N-1}\underline{K^T_{N-1}R_{N-1}K_{N-1}}x_{N-1} \\ &+ \frac{1}{2}x^T_{N-1}\underline{[A_{N-1} - B_{N-1}K_{N-1}]^T P_N [A_{N-1} - B_{N-1}K_{N-1}]}x_{N-1} \end{aligned}$$

We are going to show

$$P_{N-1} = Q_{N-1} + K^T_{N-1}R_{N-1}K_{N-1} + [A_{N-1} - B_{N-1}K_{N-1}]^T P_N [A_{N-1} - B_{N-1}K_{N-1}]$$

By definition

$$P_{N-1} = Q_{N-1} + A^T_{N-1}P_N A_{N-1} - A^T_{N-1}P_N B_{N-1}[B^T_{N-1}P_N B_{N-1} + R_{N-1}]^{-1}B^T_{N-1}P_N A_{N-1}$$

From N to N-1: Value Function

Target Equation

$$P_{N-1} = \cancel{Q_{N-1}} + K_{N-1}^T R_{N-1} K_{N-1} + [A_{N-1} - B_{N-1}K_{N-1}]^T P_N [A_{N-1} - B_{N-1}K_{N-1}]$$

$$= A_{N-1}^T \cancel{P_N} A_{N-1} - 2A_{N-1}^T P_N B_{N-1} K_{N-1} + K_{N-1}^T B_{N-1}^T P_N B_{N-1} K_{N-1}$$

$$K_{N-1}^T [R_{N-1} + B_{N-1}^T P_N B_{N-1}] K_{N-1} \qquad = A_{N-1}^T P_N B_{N-1} K_{N-1}$$

By definition

$$P_{N-1} = \cancel{Q_{N-1}} + A_{N-1}^T \cancel{P_N} A_{N-1} - A_{N-1}^T P_N B_{N-1} \underbrace{[B_{N-1}^T P_N B_{N-1} + R_{N-1}]^{-1} B_{N-1}^T P_N A_{N-1}}_{K_{N-1}}$$

From N to N-1: Value Function

- Hence we verified that

$$J^*_{N-1}(x_{N-1}) = \frac{1}{2} x^T_{N-1} P_{N-1} x_{N-1}$$

- Then we can conclude

$$\gamma_{N-1} = J^*_{N-1}(x_{N-1}) - \lambda^T_{N-1} x_{N-1} = -\frac{1}{2} x^T_{N-1} P_{N-1} x_{N-1}$$

- Co-state can only provide first-order information (i.e. gradient). To fully describe the value function, we still need to compute the zero-order term.

From N to N-1: Summary

$$u_{N-1} = -\underbrace{[B_{N-1}^T P_N B_{N-1} + R_{N-1}]^{-1} B_{N-1}^T P_N A_{N-1}}_{K_{N-1}} x_{N-1}$$

$$P_{N-1} = Q_{N-1} + A_{N-1}^T P_N A_{N-1} - A_{N-1}^T P_N B_{N-1} [B_{N-1}^T P_N B_{N-1} + R_{N-1}]^{-1} B_{N-1}^T P_N A_{N-1}$$

$$\lambda_{N-1} = P_{N-1} x_{N-1}$$

$$\gamma_{N-1} = -\frac{1}{2} x_{N-1}^T P_{N-1} x_{N-1}$$

$$J_{N-1}^*(x_{N-1}) = \frac{1}{2} x_{N-1}^T P_{N-1} x_{N-1}$$

Induction

At time k+1

$$J^*_{k+1}(x_{k+1}) = \frac{1}{2}x^T_{k+1}P_{k+1}x_{k+1} \qquad \lambda_{k+1} = P_{k+1}x_{k+1}, \gamma_{k+1} = -\frac{1}{2}x^T_{k+1}P_{k+1}x_{k+1}$$

Analogous to the case from N to N-1,

Solve the equations from the maximum principle

$$H_k(x_k, u_k, \lambda_{k+1}) = \frac{1}{2}x_k^T Q_k x_k + \frac{1}{2}u_k^T R_k u_k + \lambda^T_{k+1}(A_k x_k + B_k u_k)$$

$$\lambda_k = \frac{\partial}{\partial x}H_k(x_k, u_k, \lambda_{k+1}) = Q_k x_k + A_k^T \lambda_{k+1}$$

$$0 = \frac{\partial}{\partial u}H_k(x_k, u_k, \lambda_{k+1}) = R_k u_k + B_k^T \lambda_{k+1}$$

At time k

$$J^*_k(x_k) = \frac{1}{2}x_k^T P_k x_k \qquad \lambda_k = P_k x_k, \gamma_k = -\frac{1}{2}x_k^T P_k x_k$$

$$P_k = Q_k + A_k^T P_{k+1} A_k - A_k^T P_{k+1} B_k [R_k + B_k^T P_{k+1} B_k]^{-1} B_k^T P_{k+1} A_k$$

$$u_k = -\underbrace{[R_k + B_k^T P_{k+1} B_k]^{-1} B_k^T P_{k+1} A_k}_{K_k} x_k$$

Discrete Algebraic Riccati Equation (DARE)

$$P_k = Q_k + A_k^T P_{k+1} A_k - A_k^T P_{k+1} B_k [R_k + B_k^T P_{k+1} B_k]^{-1} B_k^T P_{k+1} A_k$$

- It describes the dynamics of the co-state
- Can be viewed as another form of Bellman update
- There is a unique solution of all P_k if we compute backward in time from the boundary condition $P_N = S_N$
- The solution does not depend on the system state, hence can be computed offline.

Implementation of LQR

- Obtain system dynamic model $x_{k+1} = A_k x_k + B_k u_k$

- Specify the matrices $Q_0, \ldots, Q_{N-1}$, S_N, $R_0, \ldots, R_{N-1}$ in the objective function

$$J = \frac{1}{2} \sum_{k=0}^{N-1} [x_k^T Q_k x_k + u_k^T R_k u_k] + \frac{1}{2} x_N^T S_N x_N$$

- Solve for $P_0, \ldots, P_N$ in the DARE backward in time (offline)

- Real-time control: at every time step, once we obtain the state measurement x_k, then we apply the following optimal control

$$u_k = -\underbrace{[R_k + B_k^T P_{k+1} B_k]^{-1} B_k^T P_{k+1} A_k}_{K_k} x_k$$

Example: Regulation of a Double Integrator

- Dynamics: one dimensional double integrator with sampling time T

$$x_{k+1} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} x_k + \begin{bmatrix} \frac{T^2}{2} \\ T \end{bmatrix} u_k$$

- Cost function

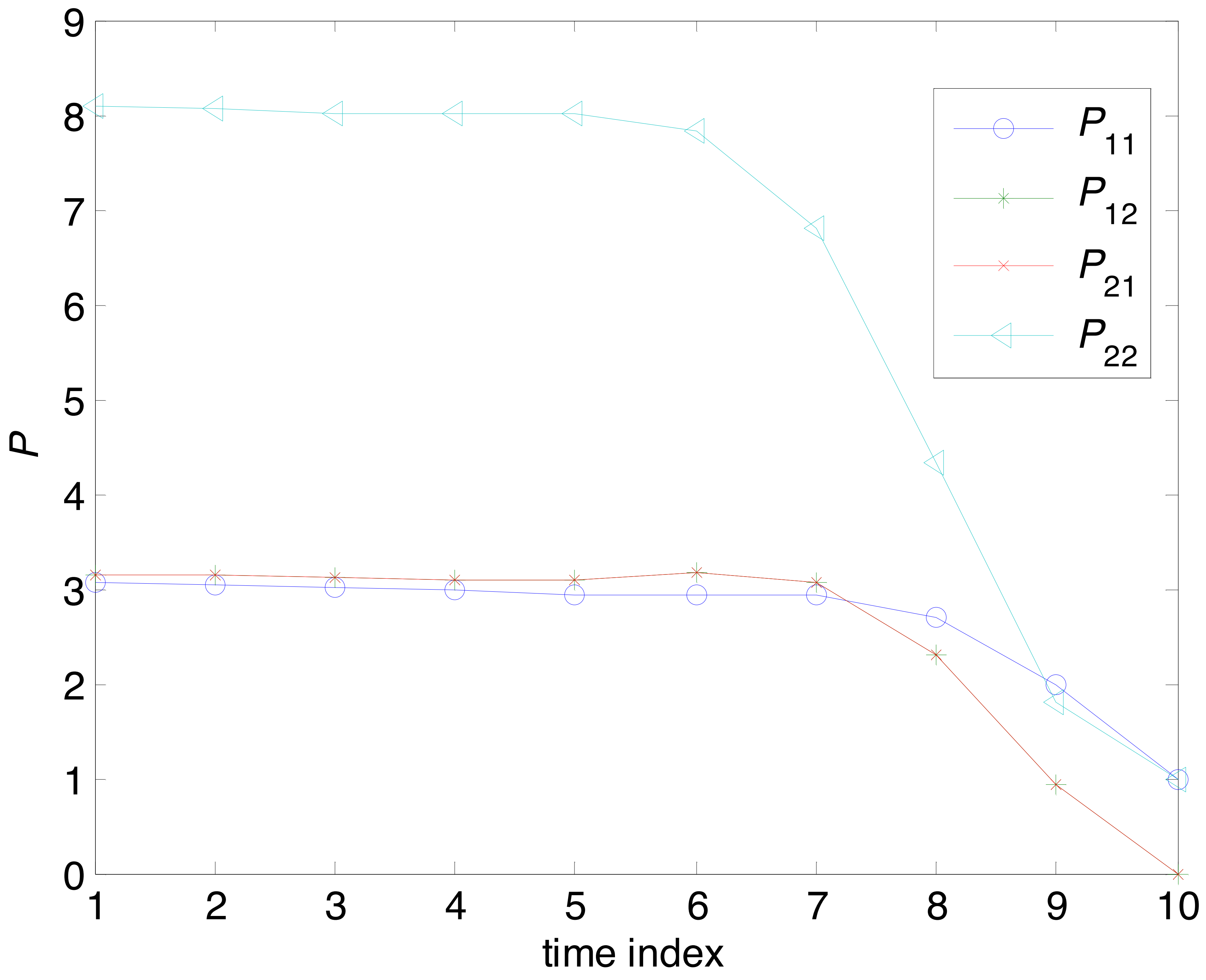
$$J = \frac{1}{2} \sum_{k=0}^{N-1} [x_k^T Q x_k + u_k^T R u_k] + \frac{1}{2} x_N^T S_N x_N$$

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, R = 1$$

Example: Regulation of a Double Integrator

- DARE

$$P_k = Q + A^T P_{k+1} A - A^T P_{k+1} B[R + B^T P_{k+1} B]^{-1} B^T P_{k+1} A$$

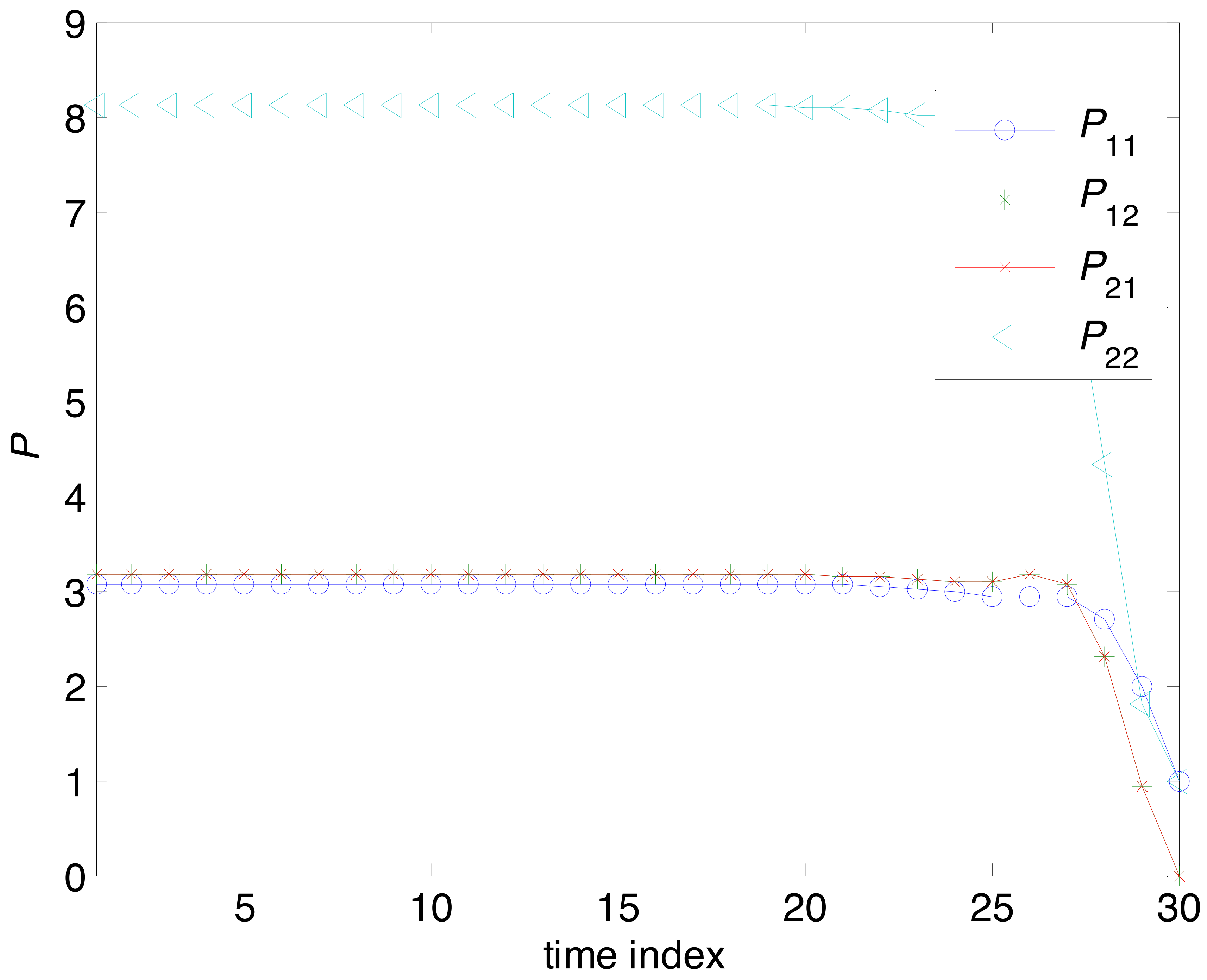


$$N = 10, P_N = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Example: Regulation of a Double Integrator

- DARE

$$P_k = Q + A^T P_{k+1} A - A^T P_{k+1} B[R + B^T P_{k+1} B]^{-1} B^T P_{k+1} A$$



$$N = 30, P_N = \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix}$$

Example: Regulation of a Double Integrator

- Observations
  - P_k is always symmetric
  - Regardless of the boundary condition P_N, the solution of the DARE always converges to the same steady state P_s
  - In this way, the control law converges (backward in time) to

$$u_k = -\underbrace{[R + B^T P_s B]^{-1} B^T P_s A}_{K_s} x_k$$

Example: Regulation of a Double Integrator

- The converged control law is indeed the solution to a **infinite horizon** LQR problem

$$u_k = -\underbrace{[R + B^T P_s B]^{-1} B^T P_s A}_{K_s} x_k$$

- There are two problems (to be answered later):
  - Under which condition the DARE (hence the control law) will converge?
  - Will the converged control law stabilize the system?

Question

- A time-invariant nonlinear optimal control problem:

$$x_{k+1} = f(x_k) + Bu_k$$
$$l(x, u) = q(x) + \frac{1}{2}u^T Ru$$
$$l_N(x) = q(x)$$

  - What is the Hamilton?
  - What is the optimal control law?
  - Dynamics of the co-state?

Discrete Time Optimal Control

- Problem formulation and examples
- Dynamic programming
- Maximum principle
- Linear Quadratic Regulator