Hybrid Systems Course:
Switched Linear Quadratic Regulation
Discrete-Time Optimal Control Problem

A discrete-time controlled dynamical system

\[ x_{k+1} = f(x_k, u_k), \quad k = 0, 1, \ldots, \text{ initial condition } x_0 \]
Discrete-Time Optimal Control Problem

A discrete-time controlled dynamical system

\[ x_{k+1} = f(x_k, u_k), \quad k = 0, 1, \ldots, \text{initial condition } x_0 \]

**Problem:** Given a time horizon \([0, N]\), find the optimal input sequence \(u = (u_0, \ldots, u_{N-1})\) that minimizes

\[
J(u) = \sum_{k=0}^{N-1} \ell(x_k, u_k) + \phi(x_N)
\]

- Running cost \(\ell(x_k, u_k) \geq 0\)
- Terminal cost \(\phi(x_N) \geq 0\)
Discrete-Time Optimal Control Problem

A discrete-time controlled dynamical system

\[ x_{k+1} = f(x_k, u_k), \quad k = 0, 1, \ldots, \text{ initial condition } x_0 \]

**Problem:** Given a time horizon \([0, N]\), find the optimal input sequence \(u = (u_0, \ldots, u_{N-1})\) that minimizes

\[
J(u) = \sum_{k=0}^{N-1} \ell(x_k, u_k) + \phi(x_N)
\]

- Running cost \(\ell(x_k, u_k) \geq 0\)
- Terminal cost \(\phi(x_N) \geq 0\)

Extension to discrete-time hybrid system

\[ x_{k+1} = f(x_k, u_k, \sigma_k), \quad k = 0, 1, \ldots \]
Linear Quadratic Regulation (LQR) Problem

A discrete-time linear system with given initial condition $x_0$:

$$x_{k+1} = Ax_k + Bu_k$$
Linear Quadratic Regulation (LQR) Problem

A discrete-time linear system with given initial condition $x_0$:

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

**Problem:** find optimal input sequence $u = (u_0, \ldots, u_{N-1})$ that minimizes

$$J(u) = \sum_{k=0}^{N-1} \left( x_k^T Q x_k + u_k^T R u_k \right) + x_N^T Q_f x_N$$

running cost \hspace{2cm} terminal cost
Linear Quadratic Regulation (LQR) Problem

A discrete-time linear system with given initial condition $x_0$:

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

**Problem:** find optimal input sequence $u = (u_0, \ldots, u_{N-1})$ that minimizes

$$J(u) = \sum_{k=0}^{N-1} \left( x_k^T Q x_k + u_k^T R u_k \right) + x_N^T Q_f x_N$$

- Running cost:
- Terminal cost:

- State weight matrix $Q = Q^T \succeq 0$
- Control weight matrix $R = R^T \succ 0$ (no free control)
- Final state weight matrix $Q_f = Q_f^T \succeq 0$
LQR Problem: Motivation

A discrete-time linear system with given initial condition $x_0$:

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

**Problem:** find optimal input sequence $u = (u_0, \ldots, u_{N-1})$ that minimizes

$$J(u) = \sum_{k=0}^{N-1} \left( x_k^T Q x_k + u_k^T R u_k \right) + x_N^T Q_f x_N$$

Compromises between the conflicting goals:

- minimize overall control effort
- minimize overall state deviation from 0

Larger control input can drive the state to zero faster
LQR Problem: Special Cases

Energy efficient stabilization: \( Q = Q_f = \alpha I, \ R = \beta I \)

\[
J(u) = \alpha \sum_{k=0}^{N} \|x_k\|^2 + \beta \sum_{k=0}^{N-1} \|u_k\|^2
\]
LQR Problem: Special Cases

Energy efficient stabilization: \( Q = Q_f = \alpha I, \ R = \beta I \)

\[
J(u) = \alpha \sum_{k=0}^{N} \|x_k\|^2 + \beta \sum_{k=0}^{N-1} \|u_k\|^2
\]

- Weights \( \alpha, \beta > 0 \) determine the emphasis between two objectives:
  (i) state stays close to 0;
  (ii) use less control energy
LQR Problem: Special Cases

**Problem:** find the control sequence $\mathbf{u} = (u_0, \ldots, u_{N-1})$ with the least energy that can steer the system state from $x_0$ to $x_N = 0$. 
LQR Problem: Special Cases

Problem: find the control sequence $u = (u_0, \ldots, u_{N-1})$ with the least energy that can steer the system state from $x_0$ to $x_N = 0$

- Set $Q = 0$ since we do not care about deviation from 0 of states at times $0, 1, \ldots, N - 1$
- Choose a very large $\alpha$ since the final state $x_N$ needs to be 0 in optimal solution
LQR Problem: Special Cases

**Problem:** find the control sequence \( u = (u_0, \ldots, u_{N-1}) \) with the least energy that can steer the system state from \( x_0 \) to \( x_N = 0 \)

- Set \( Q = 0 \) since we do not care about deviation from \( o \) of states at times \( 0, 1, \ldots, N - 1 \)
- Choose a very large \( \alpha \) since the final state \( x_N \) needs to be 0 in optimal solution

**Minimum energy steering to 0:** \( Q = 0, \quad Q_f = \alpha I, \quad R = I \)

\[
J(u) = \alpha \|x_N\|^2 + \sum_{k=0}^{N-1} \|u_k\|^2
\]

13 / 117
LQR Problem: Special Cases

A discrete-time linear system with output and given initial condition $x_0$:

\[ x_{k+1} = Ax_k + Bu_k \]
\[ y_k = Cx_k \]

Problem: find optimal input sequence $u = (u_0, \ldots, u_{N-1})$ that minimizes

\[ J(u) = \alpha \sum_{k=0}^{N} \|y_k\|^2 + \beta \sum_{k=0}^{N-1} \|u_k\|^2 \quad (\alpha > 0, \beta > 0) \]
LQR Problem: Special Cases

A discrete-time linear system with output and given initial condition $x_0$:

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

**Problem:** find optimal input sequence $u = (u_0, \ldots, u_{N-1})$ that minimizes

$$J(u) = \alpha \sum_{k=0}^{N} \|y_k\|^2 + \beta \sum_{k=0}^{N-1} \|u_k\|^2 \quad (\alpha > 0, \beta > 0)$$

As an LQR problem

- State weight matrix $Q = \alpha C^T C \succeq 0$
- Control weight matrix $R = \beta I \succ 0$
- Final state weight matrix $Q_f = Q = \alpha C^T C \succeq 0$
LQR Problem: Special Cases

A discrete-time linear system with given initial condition $x_0$:

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

**Problem:** track a reference state trajectory $x_0^r, x_1^r, \ldots, x_N^r$ with efficient control:
LQR Problem: Special Cases

A discrete-time linear system with given initial condition \( x_0 \):

\[
x_{k+1} = Ax_k + Bu_k
\]

**Problem:** track a reference state trajectory \( x_0^r, x_1^r, \ldots, x_N^r \) with efficient control:

\[
J(u) = \alpha \sum_{k=0}^{N} \|x_k - x_k^r\|^2 + \beta \sum_{k=0}^{N-1} \|u_k\|^2
\]

- \( \|x_k - x_k^r\|^2 \): tracking error penalty
- \( \|u_k\|^2 \): control energy
LQR Problem: Special Cases

A discrete-time linear system with given initial condition $x_0$:

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

**Problem:** track a reference state trajectory $x'_0, x'_1, \ldots, x'_N$ with efficient control:

$$J(u) = \alpha \sum_{k=0}^{N} \|x_k - x'_k\|^2 + \beta \sum_{k=0}^{N-1} \|u_k\|^2$$

Can be formulated as a (time-varying) LQR problem
LQR Problem: Special Cases

A discrete-time linear system with given initial condition $x_0$:

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

**Problem:** track a reference state trajectory $x_0^r, x_1^r, \ldots, x_N^r$ with efficient control:

$$J(u) = \alpha \sum_{k=0}^{N} \| x_k - x_k^r \|^2 + \beta \sum_{k=0}^{N-1} \| u_k \|^2$$

Can be formulated as a (time-varying) LQR problem

- Augment the state $x$ to $\tilde{x} = \begin{bmatrix} x \\ z \end{bmatrix}$ with $z \in \mathbb{R}$; let $\tilde{x}_0 = \begin{bmatrix} x_0 \\ 1 \end{bmatrix}$
LQR Problem: Special Cases

A discrete-time linear system with given initial condition $x_0$:

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

**Problem:** track a reference state trajectory $x^r_0, x^r_1, \ldots, x^r_N$ with efficient control:

$$J(u) = \alpha \sum_{k=0}^{N} \|x_k - x^r_k\|^2 + \beta \sum_{k=0}^{N-1} \|u_k\|^2$$

Can be formulated as a (time-varying) LQR problem

- Augment the state $x$ to $\tilde{x} = \begin{bmatrix} x \\ z \end{bmatrix}$ with $z \in \mathbb{R}$; let $\tilde{x}_0 = \begin{bmatrix} x_0 \\ 1 \end{bmatrix}$

- Augmented state dynamics to $\tilde{x}_{k+1} = \begin{bmatrix} A & 0 \\ 0 & 1 \end{bmatrix} \tilde{x}_k + \begin{bmatrix} B \\ 0 \end{bmatrix} u_k$
LQR Problem: Special Cases

A discrete-time linear system with given initial condition $x_0$:

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

**Problem:** track a reference state trajectory $x_0^r, x_1^r, \ldots, x_N^r$ with efficient control:

$$J(u) = \alpha \sum_{k=0}^{N} \|x_k - x_k^r\|^2 + \beta \sum_{k=0}^{N-1} \|u_k\|^2$$

Can be formulated as a (time-varying) LQR problem

- Augment the state $x$ to $\tilde{x} = \begin{bmatrix} x \\ z \end{bmatrix}$ with $z \in \mathbb{R}$; let $\tilde{x}_0 = \begin{bmatrix} x_0 \\ 1 \end{bmatrix}$

- Augmented state dynamics to $\tilde{x}_{k+1} = \begin{bmatrix} A & 0 \\ 0 & 1 \end{bmatrix} \tilde{x}_k + \begin{bmatrix} B \\ 0 \end{bmatrix} u_k$

- Choose $\tilde{Q}_k = \alpha \begin{bmatrix} I & -(x_k^r)^T \\ -(x_k^r) & I \end{bmatrix} \begin{bmatrix} I & -x_k^r \end{bmatrix}$, $\tilde{R} = \beta I$, $\tilde{Q}_f = \tilde{Q}_N$
Switched LQR Problem

A discrete-time switched linear system with given initial condition $x_0$:

$$x_{k+1} = A_{\sigma_k} x_k + B_{\sigma_k} u_k,$$

- continuous state: $x_k \in \mathbb{R}^n$
- discrete state (mode): $\sigma_k \in \Sigma = \{1, 2, \ldots, M\}$
Switched LQR Problem

A discrete-time switched linear system with given initial condition $x_0$:

$$x_{k+1} = A_{\sigma_k} x_k + B_{\sigma_k} u_k,$$

**Problem:** Find the optimal input sequence $(u_0, \ldots, u_{N-1})$ and mode sequence $(\sigma_0, \ldots, \sigma_{N-1})$ that minimize the cost function

$$\sum_{k=0}^{N-1} \left( x_k^T Q_{\sigma_k} x_k + u_k^T R_{\sigma_k} u_k \right) + x_N^T Q_f x_N$$

- State weight and control weight matrices mode dependent
Switched LQR Problem

A discrete-time switched linear system with given initial condition \(x_0\):

\[ x_{k+1} = A_{\sigma_k} x_k + B_{\sigma_k} u_k, \]

**Problem:** Find the optimal input sequence \((u_0, \ldots, u_{N-1})\) and mode sequence \((\sigma_0, \ldots, \sigma_{N-1})\) that minimize the cost function

\[
\sum_{k=0}^{N-1} \left( x_k^T Q_{\sigma_k} x_k + u_k^T R_{\sigma_k} u_k \right) + x_N^T Q_f x_N
\]

- State weight and control weight matrices mode dependent

**Observations:**

- In different modes, both dynamics and running costs are different
- If mode sequence is given, becomes a time-varying LQR problem
- Main challenge is determining the mode sequence
Example

Building cooling system:
- Multiple building zones
- Air Handling Units (AHUs)

State variables:
- Zone temperatures, humidity

Controls:
- AHU damper open/close
- Fan powers

Objectives:
- Maintain comfort
- Reduce energy usage

Courtesy of Jianghai Hu, Purdue University
Outline

- Solve LQR problem using dynamic programming method
- Extend the method to solve SLQR problem
- Complexity reduction techniques
Outline

- Solve LQR problem using dynamic programming method
- Extend the method to solve SLQR problem
- Complexity reduction techniques

We first look at the LQR problem:

\[
\text{Minimize } \sum_{k=0}^{N-1} \left( x_k^T Q x_k + u_k^T R u_k \right) + x_N^T Q_f x_N
\]

subject to \( x_{k+1} = A x_k + B u_k, \quad k = 0, \ldots, N - 1 \)

\( x_0 \) fixed
Direct Approach: LQR via Least-squares

The state along the time horizon \([0, N]\) is a linear function of \(u\) and \(x_0\):

\[
\begin{bmatrix}
  x_1 \\
  x_2 \\
  \vdots \\
  x_N
\end{bmatrix} = \begin{bmatrix}
  B & 0 & \cdots & 0 \\
  AB & B & 0 & \cdots \\
  \vdots & \vdots & \ddots & \vdots \\
  A^{N-1}B & A^{N-2}B & \cdots & B
\end{bmatrix} \begin{bmatrix}
  u_0 \\
  u_1 \\
  \vdots \\
  u_{N-1}
\end{bmatrix} + \begin{bmatrix}
  A \\
  A^2 \\
  \vdots \\
  A^N
\end{bmatrix} x_0
\]
Direct Approach: LQR via Least-squares

The state along the time horizon $[0, N]$ is a linear function of $u$ and $x_0$:

$$
\begin{bmatrix}
  x_1 \\
  x_2 \\
  \vdots \\
  x_N
\end{bmatrix} = 
\begin{bmatrix}
  B & 0 & \cdots & 0 \\
  AB & B & 0 & \cdots \\
  \vdots & \vdots & \ddots & \vdots \\
  A^{N-1}B & A^{N-2}B & \cdots & B
\end{bmatrix}
\begin{bmatrix}
  u_0 \\
  u_1 \\
  \vdots \\
  u_{N-1}
\end{bmatrix}
+ 
\begin{bmatrix}
  A \\
  A^2 \\
  \vdots \\
  A^N
\end{bmatrix} x_0
$$

Minimize the function:

$$
J(u) = x^T
\begin{bmatrix}
  Q & Q & \cdots & Q_f \\
  Q & \ddots & \ddots & \vdots \\
  \vdots & \ddots & Q & \cdots \\
  Q_f & \cdots & Q & R
\end{bmatrix}
+ u^T
\begin{bmatrix}
  R & R & \cdots \\
  R & \ddots & \ddots \\
  \vdots & \ddots & R & \cdots \\
  R & \cdots & R
\end{bmatrix}
$$
Direct Approach: LQR via Least-squares

The state along the time horizon \([0, N]\) is a linear function of \(u\) and \(x_0\):

\[
x = Gu + Hx_0
\]

Minimize the function:

\[
J(u) = x^TQx + u^TRu
\]

\[
= (Gu + Hx_0)^TQ(Gu + Hx_0) + u^TRu
\]
Direct Approach: LQR via Least-squares

The state along the time horizon \([0, N]\) is a linear function of \(u\) and \(x_0\):

\[
x = Gu + Hx_0
\]

Minimize the function:

\[
J(u) = x^T Q x + u^T R u \\
= (Gu + Hx_0)^T Q (Gu + Hx_0) + u^T R u \\
= \|Q^{\frac{1}{2}} (Gu + Hx_0)\|^2 + \|R^{\frac{1}{2}} u\|^2
\]

This is a least-squares problem.
Direct Approach: LQR via Least-squares

The state along the time horizon \([0, N]\) is a linear function of \(u\) and \(x_0\):

\[
x = Gu + Hx_0
\]

Minimize the function:

\[
J(u) = x^T Q x + u^T R u
\]

\[
= (Gu + Hx_0)^T Q (Gu + Hx_0) + u^T R u
\]

\[
= \|Q^{\frac{1}{2}}(Gu + Hx_0)\|^2 + \|R^{\frac{1}{2}} u\|^2
\]

This is a least-squares problem.

The optimal control is

\[
u^* = -(R + G^T Q G)^{-1} G^T Q Hx_0
\]
Direct Approach: LQR via Least-squares

Limitations of Direct Approach:

- Matrix inversion needed to find optimal control
- Problem (matrices) dimension increases with time horizon $N$
- Impractical for large $N$ let alone infinite horizon case
- Sensitivity of solutions to numerical errors
Direct Approach: LQR via Least-squares

Limitations of Direct Approach:

- Matrix inversion needed to find optimal control
- Problem (matrices) dimension increases with time horizon $N$
- Impractical for large $N$ let alone infinite horizon case
- Sensitivity of solutions to numerical errors

Observations:

- Problem easier to solve for shorter time horizon $N$
- $(N + 1)$-horizon solution related to $N$-horizon solution
- Exploit this relation to design an iterative solution procedure
Direct Approach: LQR via Least-squares

Limitations of Direct Approach:

- Matrix inversion needed to find optimal control
- Problem (matrices) dimension increases with time horizon $N$
- Impractical for large $N$ let alone infinite horizon case
- Sensitivity of solutions to numerical errors

Observations:

- Problem easier to solve for shorter time horizon $N$
- $(N+1)$-horizon solution related to $N$-horizon solution
- Exploit this relation to design an iterative solution procedure

Dynamic programming: an iterative approach that can

- Re-use results for smaller $N$ to solve for larger $N$ case
- In each iteration only need to deal with a problem of fixed size
Dynamic Programming Approach

**Idea:** Solve a sequence of optimal control problems over time horizons \([t, N]\), for decreasing \(t, = N, N - 1, \ldots, 0\)
Dynamic Programming Approach

**Idea:** Solve a sequence of optimal control problems over time horizons $[t, N]$, for decreasing $t, = N, N - 1, \ldots, 0$

- **Value function** at time $t$ is the optimal cost over $[t, N]$: 

$$V_t(x) = \min_{u_t, u_{t+1}, \ldots, u_{N-1}} \sum_{k=t}^{N-1} \left( x_k^T Q x_k + u_k^T R u_k \right) + x_N^T Q_f x_N$$

with the initial condition $x_t = x$
Dynamic Programming Approach

**Idea:** Solve a sequence of optimal control problems over time horizons \([t, N]\), for decreasing \(t = N, N - 1, \ldots, 0\)

- **Value function** at time \(t\) is the optimal cost over \([t, N]\):

\[
V_t(x) = \min_{u_t, u_{t+1}, \ldots, u_{N-1}} \sum_{k=t}^{N-1} \left( x_k^T Q x_k + u_k^T R u_k \right) + x_N^T Q_f x_N
\]

with the initial condition \(x_t = x\)

- **Value function backward iteration:**
  \(V_{t-1}(\cdot)\) can be computed based on \(V_t(\cdot)\)
Dynamic Programming Approach

**Idea:** Solve a sequence of optimal control problems over time horizons \([t, N]\), for decreasing \(t, = N, N - 1, \ldots, 0\)

- **Value function** at time \(t\) is the optimal cost over \([t, N]\):

\[
V_t(x) = \min_{u_t, u_{t+1}, \ldots, u_{N-1}} \sum_{k=t}^{N-1} \left( x_k^T Q x_k + u_k^T R u_k \right) + x_N^T Q_f x_N
\]

with the initial condition \(x_t = x\)

- Value function backward iteration:
  \(V_{t-1}(\cdot)\) can be computed based on \(V_t(\cdot)\)

- Optimal cost of original problem is \(V_0(x_0)\)

- Optimal input sequence can be recovered from value functions
Motivating Example

- Start from point A
- Try to reach point B
- Each step only move right ($\rightarrow N = 6$)
- Cost labeled on each edge

**Problem:** Path from A to B with the least cost?
Motivating Example

- Start from point A
- Try to reach point B
- Each step only move right ($\rightarrow N = 6$)
- Cost labeled on each edge

**Problem:** Path from A to B with the least cost?

- For $\ell$-by-$\ell$ grid, the total number of legal paths is $\frac{(2\ell)!}{(\ell!)^2}$, which grows fast with $\ell$. In our case $\ell = 3$, hence total number of legal path is 20.
Value Functions

Value function at $z$ is the least possible cost to reach $B$ from $z$. 
Value Functions

**Value function** at $z$ is the least possible cost to reach $B$ from $z$

**Principle of Optimality:** If a least-cost path from $A$ to $B$ is

$$x_0^* = A \rightarrow x_1^* \rightarrow x_2^* \rightarrow \cdots \rightarrow x_6^* = B,$$

then any truncation of it:

$$x_t^* \rightarrow x_{t+1}^* \rightarrow \cdots \rightarrow x_6^* = B$$

is also a least-cost path from $x_t^*$ to $B$
Value Functions

**Value function** at $z$ is the least possible cost to reach $B$ from $z$

**Principle of Optimality:** If a least-cost path from $A$ to $B$ is

$$x_0^* = A \rightarrow x_1^* \rightarrow x_2^* \rightarrow \cdots \rightarrow x_6^* = B,$$

then any truncation of it:

$$x_t^* \rightarrow x_{t+1}^* \rightarrow \cdots \rightarrow x_6^* = B$$

is also a least-cost path from $x_t^*$ to $B$

Value function at any point $z$ reached at time $t$ satisfies

$$V_t(z) = \min\{w_u + V_{t+1}(z_u'), w_d + V_{t+1}(z_d')\}$$

Optimal action when $x_t = z$ is the one providing the minimum argument and can be recovered from $V_{t+1}(\cdot)$
Value Function Iteration: Results
Value Function Iteration: Results

Diagram of a value function iteration process with values assigned to different points. The diagram illustrates the progression of values through iterations, with significant points marked and numbers indicating the values at each step. The process likely involves determining the optimal value function through successive approximations, with each iteration improving the accuracy of the solution. The diagram serves as a visual representation of the algorithm's progression, highlighting how the values converge to an optimal solution.
Value Function Iteration: Some Observations

Reduced computational complexity: for $\ell$-by-$\ell$ grid

- Only need to compute $\ell^2$ value functions
- No need to enumerate $\frac{(2\ell)!}{(\ell!)^2}$ paths
- Solve an optimization problem of fixed size in each iteration
Value Function Iteration: Some Observations

Reduced computational complexity: for $\ell$-by-$\ell$ grid

- Only need to compute $\ell^2$ value functions
- No need to enumerate $\frac{(2\ell)!}{(\ell!)^2}$ paths
- Solve an optimization problem of fixed size in each iteration

Provide solutions to a family of optimal control problems

- Even if starting from a different initial position, there is no need for re-computation
- The input is determined as a function of the current state (state feedback static policy)
Value Function Iteration: Some Observations

Reduced computational complexity: for $\ell$-by-$\ell$ grid

- Only need to compute $\ell^2$ value functions
- No need to enumerate $\frac{(2\ell)!}{(\ell!)^2}$ paths
- Solve an optimization problem of fixed size in each iteration

Provide solutions to a family of optimal control problems

- Even if starting from a different initial position, there is no need for re-computation
- The input is determined as a function of the current state (state feedback static policy)

Particularly suitable for multi-stage decision problems when the number of control choices is small at each stage
Value Functions of LQR Problem

The value function at time $t \in \{0, 1, \ldots, N\}$ and state $x \in \mathbb{R}^n$ is

$$V_t(x) = \min_{u_t, u_{t+1}, \ldots, u_{N-1}} \sum_{k=t}^{N-1} \left( x_k^T Q x_k + u_k^T R u_k \right) + x_N^T Q_f x_N$$
Value Functions of LQR Problem

The value function at time $t \in \{0, 1, \ldots, N\}$ and state $x \in \mathbb{R}^n$ is

$$V_t(x) = \min_{u_t, u_{t+1}, \ldots, u_{N-1}} \sum_{k=t}^{N-1} \left( x_k^T Q x_k + u_k^T R u_k \right) + x_N^T Q_f x_N$$

- $V_t(x)$ is the optimal cost of the LQR problem within a shorter time horizon (from time $t$ to $N$), starting from the initial condition $x_t = x$.
- $V_0(x_0)$ is the optimal cost of the original LQR problem.
LQR problem: Dynamic Programming Solution

Bellman equation:

\[
V_t(x) = \min_{u_t=v} \left[ x^T Q x + v^T R v + V_{t+1}(A x + B v) \right]
\]

cost-to-go for the current state

current running cost

cost-to-go for the next state

\[
= x^T Q x + \min_{u_t=v} \left[ v^T R v + V_{t+1}(A x + B v) \right]
\]
LQR problem: Dynamic Programming Solution

Bellman equation:

\[
V_t(x) = \min_{u_t = v} \left[ x^T Q x + v^T R v + V_{t+1}(A x + B v) \right]
\]

(cost-to-go for the current state)

\[
= x^T Q x + \min_{u_t = v} \left[ V_{t+1}(A x + B v) \right]
\]

(current running cost)

Optimal control:

can be recovered from the value functions as follows

\[
u_t^*(x) = \arg \min_v \left[ v^T R v + V_{t+1}(A x + B v) \right]
\]

(optimal control at time \( t \) when \( x_t = x \))
LQR problem: Dynamic Programming Solution

- Value function at time $N$ is quadratic: $V_N(x) = x^T Q_f x$
LQR problem: Dynamic Programming Solution

- Value function at time $N$ is quadratic: $V_N(x) = x^T Q_f x$
- Suppose $V_{t+1}(x) = x^T P_{t+1} x$ is quadratic, then

$$V_t(x) = \min_v \left[ x^T Q x + v^T R v + V_{t+1}(A x + B v) \right] = x^T P_t x$$
LQR problem: Dynamic Programming Solution

By setting \( V_{t+1}(x) = x^T P_{t+1} x \) in the expression of \( V_t(x) \), we get

\[
V_t(x) = \min_v \left[ x^T Q x + v^T R v + V_{t+1}(Ax + Bv) \right]
\]

\[
= \min_v \left[ x^T Q x + v^T R v + (Ax + Bv)^T P_{t+1} (Ax + Bv) \right]
\]

\[
= \min_v \left[ x^T Q x + v^T (R + B^T P_{t+1} B) v + 2v^T B^T P_{t+1} A x \\
+ x^T A^T P_{t+1} A x \right]
\]
LQR problem: Dynamic Programming Solution

By setting $V_{t+1}(x) = x^T P_{t+1} x$ in the expression of $V_t(x)$, we get

$$V_t(x) = \min_v \left[ x^T Q x + v^T R v + V_{t+1}(A x + B v) \right]$$

$$= \min_v \left[ x^T Q x + v^T R v + (A x + B v)^T P_{t+1}(A x + B v) \right]$$

$$= \min_v \left[ x^T Q x + v^T (R + B^T P_{t+1} B) v + 2 v^T B^T P_{t+1} A x + x^T A^T P_{t+1} A x \right]$$

Minimizer is given by

$$u_t^* = -(R + B^T P_{t+1} B)^{-1} B^T P_{t+1} A x$$
LQR problem: Dynamic Programming Solution

By setting \( V_{t+1}(x) = x^T P_{t+1} x \) in the expression of \( V_t(x) \), we get

\[
V_t(x) = \min_v \left[ x^T Q x + v^T R v + V_{t+1}(A x + B v) \right]
\]

\[
= \min_v \left[ x^T Q x + v^T R v + (A x + B v)^T P_{t+1} (A x + B v) \right]
\]

\[
= \min_v \left[ x^T Q x + v^T (R + B^T P_{t+1} B) v + 2v^T B^T P_{t+1} A x \\
+ x^T A^T P_{t+1} A x \right]
\]

Minimizer is given by

\[
u^*_t = -(R + B^T P_{t+1} B)^{-1} B^T P_{t+1} A x
\]

If we plug it back into the expression of \( V_t(x) \), we obtain

\[
V_t(x) = x^T \left( Q + A^T P_{t+1} A - A^T P_{t+1} B (R + B^T P_{t+1} B)^{-1} B^T P_{t+1} A \right) x
\]
LQR problem: Dynamic Programming Solution

- Value function at time $N$ is quadratic: $V_N(x) = x^T Q_f x$
- Suppose $V_{t+1}(x) = x^T P_{t+1} x$ is quadratic, then

$$V_t(x) = \min_v \left[ x^T Q x + v^T R v + V_{t+1}(Ax + Bv) \right] = x^T P_t x$$

is quadratic with $P_t$ obtained from $P_{t+1}$ by Riccati mapping:

$$P_t := Q + A^T P_{t+1} A - A^T P_{t+1} B (R + B^T P_{t+1} B)^{-1} B^T P_{t+1} A$$

which is achieved by the linear state feedback control

$$u_t^* = - \underbrace{(R + B^T P_{t+1} B)^{-1} B^T P_{t+1} A}_{\text{Kalman gain}} x = -K_t x$$
LQR problem: Dynamic Programming Solution

- Value function at any time is quadratic (easy numeric representation)
- Optimal control is of linear state feedback form with time-varying gains
- Yields the optimal solutions for all initial conditions $x_0$ and all initial times $t_0 \in \{0, 1, \ldots, N\}$ simultaneously
- Easily extended to time-varying dynamics and costs cases
LQR Solution Algorithm

Set $P_N = Q_f$

for $t = N - 1, N - 2, \ldots, 0$ do

Compute the value functions backward in time:

$$P_t := Q + A^T P_{t+1} A - A^T P_{t+1} B (R + B^T P_{t+1} B)^{-1} B^T P_{t+1} A$$

end for

Return $V_0(x_0)$ as the optimal cost

Set $x_0^* = x_0$

for $t = 0, 1, \ldots, N - 1$ do

Recover the optimal control and state trajectory forward in time:

$$u_t^* = -(R + B^T P_{t+1} B)^{-1} B^T P_{t+1} A x_t^*$$

$$x_{t+1}^* = A x_t^* + B u_t^*$$

end for

Return $u_t^*$ and $x_t^*$ as the optimal control and state sequences
Example

\[ x_{k+1} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x_k + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_k, \quad y_k = \begin{bmatrix} 1 & 0 \end{bmatrix} x_k = C x_k, \quad x_0 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \]

Cost function: \( J(U) = \sum_{k=0}^{N-1} \| u_k \|^2 + \rho \sum_{k=0}^{N} \| y_k \|^2 \) (\( N = 20 \))

\[ * : \rho = 0.3 \]
\[ \circ : \rho = 10 \]
Example

Optimal control is of the form

\[ u_t^* = [a_t \ b_t] \ x_t^*, \ t = 0, 1, \ldots, 19 \]

The Kalman gains \( a_t \) and \( b_t \) rapidly converge to some constant values.
Convergence of Riccati Recursion

**Theorem**

> a If \((A, B)\) is stabilizable, then Riccati recursion will converge to a solution \(P_{ss}\) of the **Algebraic Riccati Equation (ARE)**:

\[
P_{ss} = Q + A^T P_{ss} A - A^T P_{ss} B (R + B^T P_{ss} B)^{-1} B^T P_{ss} A
\]

If further \(Q = C^T C\) for some \(C\) such that \((C, A)\) is detectable, then \(P_{ss}\) is unique, and under the steady-state optimal control gain

\[
K_{ss} = (R + B^T P_{ss} B)^{-1} B^T P_{ss} A,
\]

the closed-loop system \(A_{cl} = A - BK_{ss}\) is stable

---

Important Properties of Riccati Mapping

The Riccati mapping \( P_t = \rho(P_{t+1}) \) defined by

\[
P_t = Q + A^T P_{t+1} A - A^T P_{t+1} B (R + B^T P_{t+1} B)^{-1} B^T P_{t+1} A
\]

is a mapping \( \rho : \mathbb{S}_+ \to \mathbb{S}_+ \) between set of positive semidefinite matrices.

---

**Proposition (Monotonicity)**

For \( P, P' \in \mathbb{S}_+ \) with \( P \preceq P' \), we have \( \rho(P) \preceq \rho(P') \)

**Proposition (Concavity)**

For \( P, P' \in \mathbb{S}_+ \) and \( \theta \in [0, 1] \), \( \rho(\theta P + (1 - \theta)P') \succeq \theta \rho(P) + (1 - \theta) \rho(P') \)

Back to Switched LQR Problem

A discrete-time switched linear system with given initial condition $x_0$:

$$x_{k+1} = A_{\sigma_k} x_k + B_{\sigma_k} u_k,$$

- continuous state: $x_k \in \mathbb{R}^n$
- discrete state (mode): $\sigma_k \in \Sigma = \{1, 2, \ldots, M\}$

**Problem:** Find the optimal input sequence $(u_0, \ldots, u_{N-1})$ and mode sequence $(\sigma_0, \ldots, \sigma_{N-1})$ that minimize the cost function

$$\sum_{k=0}^{N-1} \left( x_k^T Q_{\sigma_k} x_k + u_k^T R_{\sigma_k} u_k \right) + x_N^T Q_f x_N$$
Back to Switched LQR Problem

Find control sequence $u_0, \ldots, u_{N-1}$ and mode sequence $\sigma_0, \ldots, \sigma_{N-1}$ to

$$\minimize \sum_{k=0}^{N-1} \left( x_k^T Q_{\sigma_k} x_k + u_k^T R_{\sigma_k} u_k \right) + x_N^T Q_f x_N$$

subject to $x_{k+1} = A_{\sigma_k} x_k + B_{\sigma_k} u_k, \ k = 0, \ldots, N-1$

$x_0$ fixed

Value function at each $t = 0, 1, \ldots, N$ and $x$ is the optimal cost over horizon $[t, N]$ assuming $x_t = x$

$$V_t(x) = \min_{\sigma_t, \ldots, \sigma_{N-1}, u_t, \ldots, u_{N-1}} \sum_{k=t}^{N-1} \left( x_k^T Q_{\sigma_k} x_k^T + u_k^T R_{\sigma_k} u_k \right) + x_N^T Q_f x_N$$
Back to Switched LQR Problem

Observations:

- Solution strategy: dynamic programming
- In each step, need to determine both the optimal $u$ and $\sigma$
- Value function no longer quadratic
- $V_0(x_0)$ is the optimal cost of the original problem
- Value function $V_t(x)$ does not depend on mode $\sigma_{t-1}$
  - Hint: no switching cost

Robust optimal control: assume $\sigma$ is not controllable

$$\inf_u \sup_{\sigma} J(u, \sigma)$$

is a convex problem!
Bellman Equation of SLQR Problem

Value functions at different times are related by

\[ V_t(x) = \min_{\sigma_t=\sigma, u_t=v} \left[ x^T Q_{\sigma} x + v^T R_{\sigma} v + V_{t+1}(A_{\sigma} x + B_{\sigma} v) \right] \]

Optimal control and mode are the ones achieving minimum above:

- Optimal state-dependent switching policy \( \sigma_t^*(x) \)
- Optimal state feedback controller \( u_t^*(x) \)

Bad news: value functions are in general not quadratic

- \( V_N(x) = x^T Q_f x \) is quadratic
- However, for \( t = N - 1, N - 2, \ldots \)
$t = N - 1$ Case

$V_{N-1}(x) = \min_{\sigma} \min_v \left[ x^T Q_{\sigma} x + v^T R_{\sigma} v + V_N(A_{\sigma} x + B_{\sigma} v) \right]$

Bellman equation for the LQR problem of $\sigma$-th subsystem

$= \min_{\sigma} \left[ x^T \rho_{\sigma}(Q_f)x \right]$

$\rho_{\sigma}$ is Riccati mapping of subsystem $(A_{\sigma}, B_{\sigma})$ with weights $Q_{\sigma}, R_{\sigma}$
\( t = N - 1 \) Case

\( V_{N-1}(x) \) is pointwise minimum of a number of quadratic functions
\( \rightarrow \) piecewise quadratic

\[
V_{N-1}(x) = \min_{P \in \mathcal{P}_{N-1}} x^T P x
\]

where \( \mathcal{P}_{N-1} = \{ \rho_1(Q_f), \ldots, \rho_M(Q_f) \} := \rho_M(Q_f) \)

- State space partitioned into cones (radially invariant minimizer)
- One optimal mode for each cone
- One optimal linear state feedback controller for each cone
**$t = N - 2$ Case**

\[
V_{N-2}(x) = \min_{\sigma} \min_v \left[ x^T Q_{\sigma} x + v^T R_{\sigma} v + V_{N-1}(A_{\sigma} x + B_{\sigma} v) \right]
\]

\[
= \min_{\sigma} \min_v \min_{P \in \mathcal{P}_{N-1}} \left[ x^T Q_{\sigma} x + v^T R_{\sigma} v + (A_{\sigma} x + B_{\sigma} v)^T P (A_{\sigma} x + B_{\sigma} v) \right]
\]

Bellman equation for the LQR problem of $\sigma$-th subsystem

\[
= \min_{\sigma} \min_{P \in \mathcal{P}_{N-1}} \min_v \left[ x^T Q_{\sigma} x + v^T R_{\sigma} v + (A_{\sigma} x + B_{\sigma} v)^T P (A_{\sigma} x + B_{\sigma} v) \right]
\]

\[
= \min_{\sigma} \min_{P \in \mathcal{P}_{N-1}} x^T \rho_{\sigma}(P) x
\]

**Conclusion:** value function $V_{N-2}(x)$ is the pointwise minimum of $M^2$ quadratic functions:

\[
V_{N-2}(x) = \min_{P \in \mathcal{P}_{N-2}} x^T P x
\]

where $\mathcal{P}_{N-2} = \rho_{\Sigma}(\mathcal{P}_{N-1}) = \{ \rho_{\sigma}(P) : P \in \mathcal{P}_{N-1}, \sigma \in \Sigma \}$. 

72 / 117
General $t$ Case

If at $t + 1$, $V_{t+1}(x) = \min_{P \in \mathcal{P}_{t+1}} x^T P x$ for a set $\mathcal{P}_{t+1}$ of p.s.d. matrices, then at time $t$, the value function is given by

$$V_t(x) = \min_{P \in \mathcal{P}_t} x^T P x$$

where $\mathcal{P}_t$ is obtained from $\mathcal{P}_{t+1}$ by switched Riccati recursion:

$$\mathcal{P}_t = \rho \Sigma (\mathcal{P}_{t+1}) := \bigcup_{\sigma \in \Sigma} \rho_{\sigma} (\mathcal{P}_{t+1})$$

Size of $\mathcal{P}_t$ is bigger than $\mathcal{P}_{t+1}$: $|\mathcal{P}_t| = M \cdot |\mathcal{P}_{t+1}|$
SLQR Solution Algorithm

Set $P_N = \{Q_f\}$

for $t = N-1, N-2, \ldots, 0$ do
  Compute the set of p.s.d. matrices:

  $$P_t = \rho M(P_{t+1})$$

end for

Return $V_0(x_0) = \min_{P \in P_0} x_0^T P x_0$ as the optimal cost

Set $x_0^* = x_0$

for $t = 0, 1, \ldots, N-1$ do
  Recover the optimal mode $\sigma_t^*$ and the optimal control $u_t^*$ from

  $$(\sigma_t^*, u_t^*) = \arg \min_{\sigma, v} \left[ (x_t^*)^T Q_\sigma x_t^* + v^T R_\sigma v + V_{t+1}(A_\sigma x_t^* + B_\sigma v) \right]$$

  Let $x_{t+1}^* = A_{\sigma_t^*} x_t^* + B_{\sigma_t^*} u_t^*$

end for

Return $\sigma_t^*$ and $u_t^*$ as the optimal mode and control sequences
Complexity Reduction

**Issue:** Number of matrices in $P_t$ grows exponentially
Complexity Reduction

**Issue:** Number of matrices in $\mathcal{P}_t$ grows exponentially

In the set $\mathcal{P}_t$ defining the value function $V_t(x) = \min_{P \in \mathcal{P}_t} x^T P x$

- Matrix $P \in \mathcal{P}_t$ is called **effective** if for at least one $x \neq 0$

  $$x^T P x < x^T P' x, \quad \forall P' \in \mathcal{P}_t \setminus \{P\}$$

- Otherwise $P$ is called **redundant**
Complexity Reduction

**Issue:** Number of matrices in $\mathcal{P}_t$ grows exponentially

In the set $\mathcal{P}_t$ defining the value function $V_t(x) = \min_{P \in \mathcal{P}_t} x^T P x$

- Matrix $P \in \mathcal{P}_t$ is called **effective** if for at least one $x \neq 0$

  $$x^T P x < x^T P' x, \quad \forall P' \in \mathcal{P}_t \setminus \{P\}$$

- Otherwise $P$ is called **redundant**

Redundant matrices can be discarded without affecting optimal solution because of the monotonicity of Riccati mapping
Complexity Reduction

**Issue:** Number of matrices in $\mathcal{P}_t$ grows exponentially

In the set $\mathcal{P}_t$ defining the value function $V_t(x) = \min_{P \in \mathcal{P}_t} x^T P x$

- Matrix $P \in \mathcal{P}_t$ is called **effective** if for at least one $x \neq 0$

  $$x^T P x < x^T P' x, \quad \forall P' \in \mathcal{P}_t \setminus \{P\}$$

- Otherwise $P$ is called **redundant**

Redundant matrices can be discarded without affecting optimal solution because of the monotonicity of Riccati mapping

**Sufficient condition** for $P \in \mathcal{P}_t$ to be redundant:

$$P \succeq \text{a convex combination of } P' \in \mathcal{P}_t \setminus \{P\}$$
Complexity Reduction

**Issue:** Number of matrices in $P_t$ grows exponentially

In the set $P_t$ defining the value function $V_t(x) = \min_{P \in P_t} x^T P x$

- Matrix $P \in P_t$ is called **effective** if for at least one $x \neq 0$

  \[ x^T P x < x^T P' x, \quad \forall P' \in P_t \setminus \{P\} \]

- Otherwise $P$ is called **redundant**

Redundant matrices can be discarded without affecting optimal solution because of the monotonicity of Riccati mapping

**Sufficient condition** for $P \in P_t$ to be redundant:

\[ P \succeq \text{a convex combination of } P' \in P_t \setminus \{P\} \]

Proof:

\[ x^T P x \geq \sum_{P_i \in P_t \setminus \{P\}} \alpha_i x^T P_i x \geq x^T P_j x, \quad \text{for some } P_j \in P_t \setminus \{P\} \]
Complexity Reduction

**Issue:** Number of matrices in $\mathcal{P}_t$ grows exponentially

In the set $\mathcal{P}_t$ defining the value function $V_t(x) = \min_{P \in \mathcal{P}_t} x^T Px$

- Matrix $P \in \mathcal{P}_t$ is called **effective** if for at least one $x \neq 0$
  $$x^T Px < x^T P' x, \quad \forall P' \in \mathcal{P}_t \setminus \{P\}$$

- Otherwise $P$ is called **redundant**

Redundant matrices can be discarded without affecting optimal solution because of the monotonicity of Riccati mapping

**Sufficient condition** for $P \in \mathcal{P}_t$ to be redundant:

$$P \succeq \text{a convex combination of } P' \in \mathcal{P}_t \setminus \{P\} \quad (1)$$

Proof:

$$x^T Px \geq \sum_{P_i \in \mathcal{P}_t \setminus \{P\}} \alpha_i x^T P_i x \geq x^T P_j x, \quad \text{for some } P_j \in \mathcal{P}_t \setminus \{P\}$$

LMI feasibility condition to test.
Example of Ineffective Matrices

A switched LQR problem specified by

\[
A_1 = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}, \quad B_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 1.5 & 1 \\ 0 & 1.5 \end{bmatrix}, \quad B_2 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}
\]

\[Q_\sigma = I, \quad R_\sigma = 1, \quad N = 20\]

16 matrices in \( P_{16} \) evaluated along half of the unit circle because of the radial invariance
Decision Tree Pruning
Further Reduction by Relaxation

Remove more matrices by relaxing Condition (1) to

\[ P \succeq \sum_{i \in \mathcal{I}} \alpha_i P_i - \varepsilon I \]  \hspace{1cm} (2)

- \( \varepsilon > 0 \) is a small constant specifying approximation quality
- Even a small \( \varepsilon \) could result in significant reduction in complexity


Example

\[ A_1 = \begin{bmatrix} 2 & 1 \\ 0 & 1 \end{bmatrix}, \quad B_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 2 & 1 \\ 0 & 0.5 \end{bmatrix}, \quad B_2 = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \]

\[ Q_1 = Q_2 = I, \quad R_1 = R_2 = 1, \quad Q_f = I, \quad N = 100 \]

| \( k \) | \( |\mathcal{P}_{N-k}| \) |
|-------|-------------|
| 1     | 2           |
| 2     | 4           |
| 3     | 5           |
| 4     | 5           |
| 5     | 5           |
| 6     | 5           |
Example (cont.)

Optimal switching policy (Gray region: mode 1 optimal; Black region: mode 2 optimal)
Another Example

\[
A_1 = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}, \quad B_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}
\]

\[
A_2 = \begin{bmatrix} 1.5 & 1 \\ 0 & 1.5 \end{bmatrix}, \quad B_2 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}
\]

\[Q_\sigma = I, \quad R_\sigma = 1, \quad N = 20\]

- Without any reduction, complexity grows exponentially
- With reduction, complexity saturates at 360 matrices
- With relaxation \((\varepsilon = 10^{-3})\), complexity saturates at 14 matrices
SLQR Problem with Switching Cost

Cost function to be minimized:

\[
\sum_{k=0}^{N-1} \left( x_k^T Q_{\sigma_k} x_k + u_k^T R_{\sigma_k} u_k + w(\sigma_k, \sigma_{k+1})(x_k) \right) + x_N^T Q_f x_N
\]
SLQR Problem with Switching Cost

Cost function to be minimized:

\[
\sum_{k=0}^{N-1} \left( x_k^T Q_{\sigma_k} x_k + u_k^T R_{\sigma_k} u_k + w(\sigma_k, \sigma_{k+1})(x_k) \right) + x_N^T Q_f x_N
\]

**Value function** $V_t(\sigma, x)$ is the optimal cost-to-go starting from $x_t = x$ with previous mode being $\sigma_{t-1} = \sigma$

**Bellman equation:**

\[
V_t(\sigma, x) = \min_{\sigma_t=\sigma', u_t=v} \left[ x^T Q_{\sigma'} x + v^T R_{\sigma'} v + w(\sigma, \sigma')(x) + V_{t+1}(\sigma', A_{\sigma'} x + B_{\sigma'} v) \right]
\]
SLQR Problem with Switching Cost

Cost function to be minimized:

\[
\sum_{k=0}^{N-1} \left( x_k^T Q_{\sigma_k} x_k + u_k^T R_{\sigma_k} u_k + w(\sigma_k, \sigma_{k+1})(x_k) \right) + x_N^T Q_f x_N
\]

Value function \( V_t(\sigma, x) \) is the optimal cost-to-go starting from \( x_t = x \) with previous mode being \( \sigma_{t-1} = \sigma \)

Bellman equation:

\[
V_t(\sigma, x) = \min_{\sigma_t = \sigma', \sigma' \neq \sigma} \left[ x^T Q_{\sigma'} x + v^T R_{\sigma'} v + w(\sigma, \sigma')(x) + V_{t+1}(\sigma', A_{\sigma'} x + B_{\sigma'} v) \right]
\]

- Optimal switching policy \( \sigma^*_t(\sigma, x) \) and optimal control \( u^*_t(\sigma, x) \)
- Both depend on previous mode \( \sigma \) and current state \( x \)
- Same technique with piecewise quadratic value functions if \( w(\sigma, \sigma')(\cdot) \) is quadratic, linear, or constant.
Continuous-Time LQR Problem

A continuous-time linear system with given initial condition $x_0$:

$$\dot{x} = Ax + Bu$$

**Problem:** find the optimal control input $u(t)$ over the time horizon $[0, t_f]$ that minimizes

$$J = \int_0^{t_f} \left( x^T Q x + u^T R u \right) \, dt + x(t_f)^T Q_f x(t_f)$$

- State running weight $Q = Q^T \succeq 0$
- Control running weight $R = R^T \succ 0$
- Final state weight $Q_f = Q_f^T \succeq 0$
Continuous-Time LQR Problem

Value function at time $t \in [0, t_f]$:

$$V_t(x) = \min_{u(s), s \in [t, t_f]} \int_t^{t_f} \left[ x(s)^T Q x(s) + u(s)^T R u(s) \right] ds + x(t_f)^T Q_f x(t_f)$$

- $V_t(x)$ is the optimal cost-to-go at time $t$ from state $x$
- optimal cost of the original LQR problem is given by $V_0(x_0)$
- at time $t_f$, the value function is quadratic $V_{t_f}(x) = x^T Q_f x$
Continuous-Time LQR Problem

Value function at time $t \in [0, t_f]$: 

$$V_t(x) = \min_{u(s), s \in [t, t_f]} \int_t^{t_f} \left[ x(s)^T Q x(s) + u(s)^T R u(s) \right] \, ds + x(t_f)^T Q_f x(t_f)$$

- $V_t(x)$ is the optimal cost-to-go at time $t$ from state $x$
- optimal cost of the original LQR problem is given by $V_0(x_0)$
- at time $t_f$, the value function is quadratic $V_{t_f}(x) = x^T Q_f x$

As in the discrete-time case, the value function can be shown to be quadratic at any time: 

$$V_t(x) = x^T P(t) x, \ t \in [0, t_f]$$
A Heuristic Derivation of Value Functions

• Assume that the system starts from $x$ at time $t$

\[ x(t) = x, \quad t \in [0, t_f), \quad x \in \mathbb{R}^n \]

• Assume that the control input is kept constant for a brief $\delta$-length time horizon

\[ u(s) = w, \quad s \in [t, t + \delta] \]

• Assume that the value function is quadratic at any time:

\[ V_t(x) = x^T P(t)x, \quad t \in [0, t_f] \]
A Heuristic Derivation of Value Functions

Bellman equation:

\[
V_t(x) \simeq \min_{u(\cdot) = w} \left[ \delta(x^T Q x + w^T R w) + V_{t+\delta}(x + \delta(Ax + Bw)) \right]
\]

- \(V_t(x)\): cost-to-go at time \(t\)
- \(u(\cdot) = w\): control input
- \(\delta(x^T Q x + w^T R w)\): cost during \([t, t + \delta]\)
- \(V_{t+\delta}(x + \delta(Ax + Bw))\): cost-to-go from time \(t + \delta\)
A Heuristic Derivation of Value Functions

Bellman equation:

\[ V_t(x) \simeq \min_{u(\cdot) \equiv w} \left[ \delta(x^T Qx + w^T R w) + V_{t+\delta}(x + \delta(Ax + Bw)) \right] \]

\[ V_{t+\delta}(x + \delta(Ax + Bw)) \]
\[ = [x + \delta(Ax + Bw)]^T P(t + \delta)[x + \delta(Ax + Bw)] \]
\[ \simeq [x + \delta(Ax + Bw)]^T [P(t) + \delta \dot{P}(t)][x + \delta(Ax + Bw)] \]
\[ \simeq x^T P(t)x + \delta \left[ x^T P(t)(Ax + Bw) + (Ax + Bw)^T P(t)x + x^T \dot{P}(t)x \right] \]
\[ = V_t(x) + \delta \left[ x^T P(t)(Ax + Bw) + (Ax + Bw)^T P(t)x + x^T \dot{P}(t)x \right] \]
A Heuristic Derivation of Value Functions

Bellman equation:

\[
V_t(x) \simeq \min_{u(\cdot) \equiv w} \left[ \delta(x^TQx + w^TRw) + V_{t+\delta}(x + \delta(Ax + Bw)) \right]
\]

\[
V_{t+\delta}(x + \delta(Ax + Bw)) \\
\simeq V_t(x) + \delta \left[ x^TP(t)(Ax + Bw) + (Ax + Bw)^TP(t)x + x^T\dot{P}(t)x \right]
\]
A Heuristic Derivation of Value Functions

Bellman equation:

\[ V_t(x) \simeq \min_{u(\cdot) \equiv w} \left[ \delta(x^T Qx + w^T R w) + V_{t+\delta}(x + \delta(Ax + Bw)) \right] \]

\[ V_{t+\delta}(x + \delta(Ax + Bw)) \]
\[ \simeq V_t(x) + \delta \left[ x^T P(t)(Ax + Bw) + (Ax + Bw)^T P(t)x + x^T \dot{P}(t)x \right] \]

As \( \delta \to 0 \), Bellman equation becomes asymptotically:

\[ 0 = \min_{u(t) = w} \left\{ x^T Qx + w^T R w + x^T P(t)(Ax + Bw) + (Ax + Bw)^T P(t)x \right. \]
\[ + \left. x^T \dot{P}(t)x \right\} \]
The optimal control law at time $t$ is then:

$$u^*(t) = \arg\min_w \left\{ x^T Qx + w^T R w + x^T P(t)(Ax + Bw) 
+ (Ax + Bw)^T P(t)x + x^T \dot{P}(t)x \right\}$$

$$= -R^{-1}B^T P(t) x$$

$K(t)$: Kalman gain
A Heuristic Derivation of Value Functions

The optimal control law at time $t$ is then:

$$u^*(t) = \arg \min_w \left\{ x^T Qx + w^T R w + x^T P(t) (Ax + Bw) + (Ax + Bw)^T P(t) x + x^T \dot{P}(t) x \right\}$$

$$= -R^{-1} B^T P(t) x$$

$K(t):$ Kalman gain

Plug this back into the asymptotic version of the Bellman equation:

$$0 = \left\{ x^T Qx + w^T R w + x^T P(t) (Ax + Bw) + (Ax + Bw)^T P(t) x + x^T \dot{P}(t) x \right\}_{w=u^*_t}$$

$$\rightarrow 0 = x^T \left\{ Q + P(t) A + A^T P(t) - P(t) BR^{-1} B^T P(t) + \dot{P}(t) \right\} x, \quad \forall x$$

$$\rightarrow -\dot{P}(t) = Q + P(t) A + A^T P(t) - P(t) BR^{-1} B^T P(t)$$

Initial condition: $P(t_f) = Q_f$ and integrated backward in time till time 0
LQR problem: Dynamic Programming Solution

The value functions are quadratic

\[ V_t(x) = x^T P(t) x \]

with \( P(t) \) satisfying the **Riccati (matrix) differential equation**:

\[
-\dot{P}(t) = Q + P(t)A + A^TP(t) - P(t)BR^{-1}B^TP(t), \quad P(t_f) = Q_f
\]

The optimal control is a **linear state feedback controller**:

\[
u^*(t) = -R^{-1}B^TP(t)x
\]
LQR Solution Algorithm

Set $P(t_f) = Q_f$

Solve the matrix Riccati equation backward in time:

$$-\dot{P}(t) = Q + P(t)A + A^T P(t) - P(t)BR^{-1}B^T P(t)$$

Return $V_0(x_0) = x_0^T P(0)x_0$ as the optimal cost

Set $x^*(0) = x_0$

Recover the optimal control and trajectory forward in time

$$\begin{aligned}
\dot{x}^*(t) &= Ax^*(t) + Bu^*(t) \\
u^*(t) &= K(t)x^*(t)
\end{aligned}$$

$t \in [0, t_f]$

where $K(t)$ is the Kalman gain computed by

$$K(t) = -R^{-1}B^T P(t)$$
Switched LQR Problem

A continuous-time switched linear system with given initial condition $x_0$:

$$\dot{x} = A_\sigma x + B_\sigma u$$

- continuous state: $x(t) \in \mathbb{R}^n$
- discrete state (mode): $\sigma(t) \in \Sigma = \{1, 2, \ldots, M\}$
Switched LQR Problem

A continuous-time switched linear system with given initial condition $x_0$:

$$\dot{x} = A_\sigma x + B_\sigma u$$

**Problem:** Find the optimal mode $\sigma(t) \in \Sigma$ and input $u(t)$ over the time horizon $[0, t_f]$ that minimize the cost function

$$\int_0^{t_f} \left( x^T Q_\sigma x + u^T R_\sigma u \right) \, dt + x(t_f)^T Q_f x(t_f)$$

- State running weight $Q_\sigma = Q_\sigma^T \succeq 0$, $\sigma \in \Sigma$
- Control running weight $R_\sigma = R_\sigma^T \succ 0$, $\sigma \in \Sigma$
- Final state weight $Q_f = Q_f^T \succeq 0$
- No switching cost
Switched LQR Problem

A continuous-time switched linear system with given initial condition $x_0$:

$$\dot{x} = A_\sigma x + B_\sigma u$$

**Problem:** Find the optimal mode $\sigma(t) \in \Sigma$ and input $u(t)$ over the time horizon $[0, t_f]$ that minimize the cost function

$$\int_0^{t_f} \left(x^T Q_\sigma x + u^T R_\sigma u\right) dt + x(t_f)^T Q_f x(t_f)$$

- State running weight $Q_\sigma = Q_{\sigma}^T \succeq 0$, $\sigma \in \Sigma$
- Control running weight $R_\sigma = R_{\sigma}^T \succ 0$, $\sigma \in \Sigma$
- Final state weight $Q_f = Q_{f}^T \succeq 0$
- No switching cost

**Observations:**

- In different modes, both dynamics and running costs are different
- If mode sequence is given, becomes a time-varying LQR problem
- Main challenge is determining the mode sequence
Continuous-Time SLQR Problem

Value function at time $t \in [0, t_f]$: 

$$V_t(x) = \min_{u(s), \sigma(s), s \in [t, t_f]} \left\{ \int_t^{t_f} \left[ x(s)^T Q_{\sigma(s)} x(s) + u(s)^T R_{\sigma(s)} u(s) \right] ds + x(t_f)^T Q_f x(t_f) \right\}$$ 

- $V_t(x)$ is the optimal cost-to-go at time $t$ from state $x$
- value function independent of $\sigma$ due to the absence of switching cost
- optimal cost of the original SQLR problem is given by $V_0(x_0)$
- at time $t_f$, the value function is quadratic: $V_{t_f}(x) = x^T Q_f x$
Continuous-Time SLQR Problem

**Value function** at time $t \in [0, t_f]$:

$$V_t(x) = \min_{u(s), \sigma(s), s \in [t, t_f]} \left\{ \int_t^{t_f} \left[ x(s)^T Q_{\sigma(s)} x(s) + u(s)^T R_{\sigma(s)} u(s) \right] ds + x(t_f)^T Q_f x(t_f) \right\}$$

- $V_t(x)$ is the optimal cost-to-go at time $t$ from state $x$
- value function independent of $\sigma$ due to the absence of switching cost
- optimal cost of the original SQLR problem is given by $V_0(x_0)$
- at time $t_f$, the value function is quadratic: $V_{t_f}(x) = x^T Q_f x$

As in the discrete-time case, the value function at any time is the minimum of a (time-varying) set of quadratic functions:

$$V_t(x) = \inf_{P \in \mathcal{P}(t)} x^T P x$$
Derivation of Value Functions

To obtain a more tractable optimal control problem:

• embed the switched system in the larger family

\[ \dot{x} = A_\lambda x + B_\lambda u, \quad x(0) = x_0 \]

where \( A_\lambda = \sum_{i=1}^{M} \lambda_i A_i \) and \( B_\lambda = \sum_{i=1}^{M} \lambda_i B_i \) are parameterized by

\[ \lambda = (\lambda_1, \ldots, \lambda_M) \text{ with } \lambda_i \geq 0, i = 1, \ldots, M, \sum_{i=1}^{M} \lambda_i = 1 \]

\[ \rightarrow \lambda \in S, \text{ } S \text{ being a simplex.} \]

When \( \lambda \) takes value in a vertex of \( S \), we get one of the dynamical systems among which switching occurs. For instance, if \( \lambda_i = 1 \), then,

\[ \dot{x} = A_i x + B_i u \]
Derivation of Value Functions

To obtain a more tractable optimal control problem:

- reformulate the optimal control problem as follows:

Find \( u(t) \) and \( \lambda(t) \), \( t \in [0, t_f] \), to

\[
\begin{align*}
\text{minimize} & \quad \int_0^{t_f} \left( x^T Q_\lambda x + u^T R_\lambda u \right) \, dt + x(t_f)^T Q_f x(t_f) \\
\text{subject to} & \quad \dot{x} = A_\lambda x + B_\lambda u, \quad t \in [0, t_f] \\
& \quad x_0 \text{ fixed}
\end{align*}
\]

where \( Q_\lambda = \sum_{i=1}^{M} \lambda_i Q_i \) and \( R_\lambda = \sum_{i=1}^{M} \lambda_i R_i \)
Derivation of Value Functions

To obtain a more tractable optimal control problem:

- reformulate the optimal control problem as follows:

  Find \( u(t) \) and \( \lambda(t) \), \( t \in [0, t_f] \), to

\[
\text{minimize } \int_0^{t_f} \left( x^T Q_{\lambda} x + u^T R_{\lambda} u \right) dt + x(t_f)^T Q_f x(t_f)
\]

subject to

\[
\dot{x} = A_{\lambda} x + B_{\lambda} u, \quad t \in [0, t_f]
\]

\( x_0 \) fixed

where \( Q_{\lambda} = \sum_{i=1}^{M} \lambda_i Q_i \) and \( R_{\lambda} = \sum_{i=1}^{M} \lambda_i R_i \)

If the optimal \( \lambda(t) \), \( t \in [0, t_f] \), takes values in the vertices of the simplex \( S \), then, the solution is also optimal for the original switched problem, otherwise only a suboptimal solution can be determined.
Derivation of Value Functions

• Assume that the system starts from \( x \) at time \( t \)

\[
    x(t) = x, \quad t \in [0, t_f), \quad x \in \mathbb{R}^n
\]

• Assume that the control input is kept constant for a brief \( \delta \)-length time horizon

\[
    u(s) = w, \quad s \in [t, t + \delta]
\]

• Assume that the value function is the minimum of a (time-varying) set of quadratic functions:

\[
    V_t(x) = \inf_{P \in \mathcal{P}(t)} x^T P x
\]
Derivation of Value Functions

Bellman equation:

\[
V_t(x) \simeq \min_{w, \lambda} \left[ \delta(x^T Q \lambda x + w^T R \lambda w) + V_{t+\delta}(x + \delta(A \lambda x + B \lambda w)) \right]
\]

- \(V_t(x)\): cost-to-go at \(t\)
- \(\min_{w, \lambda}\): cost during \([t, t + \delta]\)
- \(V_{t+\delta}(x + \delta(A \lambda x + B \lambda w))\): cost-to-go from time \(t + \delta\)
Derivation of Value Functions

Bellman equation:

\[ V_t(x) \simeq \min_{w, \lambda} \left[ \delta(x^T Q_\lambda x + w^T R_\lambda w) + V_{t+\delta}(x + \delta(A_\lambda x + B_\lambda w)) \right] \]

Note that since

\[ V_{t+\delta}(x) = \inf_{P(t+\delta) \in P(t+\delta)} x^T P(t + \delta)x \]

we get

\[ V_t(x) \simeq \min_{w, \lambda, P(t+\delta) \in P(t+\delta)} \left[ \delta(x^T Q_\lambda x + w^T R_\lambda w) + (x + \delta(A_\lambda x + B_\lambda w))^T P(t + \delta)(x + \delta(A_\lambda x + B_\lambda w)) \right] \]
Derivation of Value Functions

Expand $P(t + \delta) \in \mathcal{P}(t + \delta)$ as $P(t + \delta) \simeq P(t) + \delta \dot{P}(t)$ for some $P(t) \in \mathcal{P}(t)$
Derivation of Value Functions

Expand $P(t + \delta) \in \mathcal{P}(t + \delta)$ as $P(t + \delta) \simeq P(t) + \delta \dot{P}(t)$ for some $P(t) \in \mathcal{P}(t)$

Let $\delta \to 0$. The Bellman equation becomes asymptotically:

$$0 = \min_{w, \lambda, P(t) \in \mathcal{P}(t)} \left\{ x^T Q_\lambda x + w^T R_\lambda w + x^T P(t)(A_\lambda x + B_\lambda w) + (A_\lambda x + B_\lambda w)^T P(t)x + x^T \dot{P}(t)x \right\}$$
Derivation of Value Functions

Expand $P(t + \delta) \in \mathcal{P}(t + \delta)$ as $P(t + \delta) \simeq P(t) + \delta \dot{P}(t)$ for some $P(t) \in \mathcal{P}(t)$

Let $\delta \to 0$. The Bellman equation becomes asymptotically:

$$0 = \min_{w, \lambda, P(t) \in \mathcal{P}(t)} \left\{ x^T Q_\lambda x + w^T R_\lambda w + x^T P(t)(A_\lambda x + B_\lambda w) \\ + (A_\lambda x + B_\lambda w)^T P(t)x + x^T \dot{P}(t)x \right\}$$

Using the optimal control, the value function is of the form

$$V_t(x) = \inf_{P \in \mathcal{P}(t)} x^T Px$$

where the set $\mathcal{P}(t)$ satisfies

$$-\dot{P}(t) \in \{ Q_\lambda + P(t)A_\lambda + A_\lambda^T P(t) - P(t)B_\lambda R_\lambda^{-1} B_\lambda^T P(t) : \lambda \in S \}$$

$\forall P(t) \in \mathcal{P}(t)$. 
Value Functions of C.-T. SLQR Problem

The value function $V_t(x)$ is still of the form

$$V_t(x) = \inf_{P \in \mathcal{P}(t)} x^T P x$$

$\mathcal{P}(t)$ can be computed from the Riccati differential inclusion

$$-\dot{P}(t) \in \{ Q_\lambda + P(t)A_\lambda + A_\lambda^T P(t) - P(t)B_\lambda R_\lambda^{-1} B_\lambda^T P(t) : \lambda \in S \}$$

where $A_\lambda, B_\lambda, Q_\lambda, R_\lambda$ is any convex combination of $A_i, B_i, Q_i, R_i$ for $i \in \Sigma$.
Value Functions of C.-T. SLQR Problem

The value function $V_t(x)$ is still of the form

$$V_t(x) = \inf_{P \in \mathcal{P}(t)} x^T P x$$

$\mathcal{P}(t)$ can be computed from the Riccati differential inclusion

$$-\dot{P}(t) \in \{ Q_\lambda + P(t)A_\lambda + A_\lambda^T P(t) - P(t)B_\lambda R_\lambda^{-1} B_\lambda^T P(t) : \lambda \in S \}$$

where $A_\lambda, B_\lambda, Q_\lambda, R_\lambda$ is any convex combination of $A_i, B_i, Q_i, R_i$ for $i \in \Sigma$

In general, $\mathcal{P}(t)$ is very difficult to compute analytically and numerically

- Discretize the C.-T. SLS into D.-T. SLS