# 02_Unconstrained MPC: LQR



## Summary

- LQR is easily to **solved**, so it is preferred by us
- Two methods can be used when horizon is finite: **DP and FRH**
- Finite horizon will face the problem of **instability**. But if with infinite horizon, a controllable system will always **keep stable** and the solution of FRH will **converge** to the static solution

# 1. Model and Motivation of LQR

## Model

**Cost of State :**

$$I(x,u) = x^T Q x + u^T R u$$

**Cost of a trajectory :**

$$V\left(x_0, \mathbf{u}\right) = \sum_{i=0}^{N} x_i^T Q x_i + u_i^T R u_i$$

Motivation

- Can adjust the weight between input and state
- Can be easily and quickly solve in Embedded Controller

# 2. Solution of Finite-Horizon LQR

## Dynamic Programming Method (Bellman Recursion)

**Basic Idea:**

This problem has the **optimal sub-structure** property: For the given optimal input sequence, no matter $x_n$, the input from n to N should make the sub-structure is also the optimal, otherwise, we can build a better solution from $x_n$ which made the total value smaller than current optimal solution. Which also indicates:

- no matter how we arrive at a state $x_n$, the optimal solution from $x_n$ only depend on $x_n$, not related to how we arrive at $x_n$
- Start from tail and iteration forward, we can find the best solution

**Process**

In the finite-horizon case, we explicitly knows that the final state is a state with quadratic cost $x^T H x$. Then we can use mathematical induction to prove $V^*(x)$ is quadratic. Assume for timestamp $i+1$, it holds, then put it into timestamp $i$.

$$\begin{aligned} V_i\left(x_i\right) &= \min_{u_i} x_i^T Q x_i + u_i^\top R u_i + V_{i+1}\left(A x_i + B u_i\right) \\ &= \min_{u_i} \left( x_i^\top Q x_i + u_i^\top R u_i + \left(A x_i + B u_i\right)^\top H_{i+1}\left(A x_i + B u_i\right) \right) \end{aligned}$$

Then we can take the derivative to zero and get the $V_i^*(x)$

1. start from the last step N and compute

$$\begin{aligned} V_N^\star\left(x_N\right) &:= \min_{u_N} x_N^T Q x_N + u_N^T R u_N \\ &= x_N^\top Q x_N \quad \text{(Quardratic)} \\ H_N &:= Q \end{aligned}$$

2. Iterate backwards for $i = N-1, \cdots, 0$, do DP iterations

$$V_i^\star\left(x_i\right) := \min_{u_i} x_i^\top Q x_i + u_i^\top R u_i + V_{i+1}^\star\left(A x_i + B u_i\right)$$

A closed-form equation can be calculated:

$$\begin{aligned} u_i^\star\left(x_i\right) &= K_i x_i \quad K_i = -\left(R + B^\top H_{i+1} B\right)^{-1} B^\top H_{i+1} A \\ V_i^\star\left(x_i\right) &= x_i^T H_i x_i \quad H_i = Q + K_i^T R K_i + \left(A + B K_i\right)^\top H_{i+1}\left(A + B K_i\right) \end{aligned}$$

3. Finally, we will get $V^*(x_0) = V_0^*(x_0)$ and the optimal controller $u_0^*(x_0)$

**Properties**

- $V_i^*(x)$ is **quadratic** (can be proved by mathematical induction method)
- $V_i^*(x)$ is **positive definite** (can be proved by mathematical induction method)
- Optimizer $u_0^*(x)$ is **linear** of current state

## Receding Horizon Method

Receding Horizon Method use **optimization perspective** to solve this problem. It first represent the problem to a model with single variable.

**Model**

$$V^{\star}\left(x_0\right) := \min_{\mathbf{u}} \quad \mathbf{x}^{\top}\mathcal{Q}\mathbf{x} + \mathbf{u}^{\top}\mathcal{R}\mathbf{u} \quad \text{s.t.} \quad \mathcal{A}\mathbf{x} + \mathcal{B}\mathbf{u} = \mathcal{C}x_0$$

where

$$\mathbf{x} = \begin{bmatrix} x_1^T & \cdots & x_N^T \end{bmatrix}^T, \mathbf{u} = \begin{bmatrix} u_0^T & \cdots & u_{N-1}^T \end{bmatrix}^T$$

$$\mathcal{A} := \begin{bmatrix} -1 & 0 & \cdots & \cdots & \cdots & 0 \\ A & -1 & 0 & \cdots & \cdots & 0 \\ 0 & A & -1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & A & -1 \end{bmatrix}$$

$$\mathcal{B} := \begin{bmatrix} B & 0 & \cdots & 0 \\ 0 & B & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & B \end{bmatrix} \quad \mathcal{C} := \begin{bmatrix} -A \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$\mathcal{Q} := \operatorname{diag}(Q, \ldots, Q) \quad \mathcal{R} := \operatorname{diag}(R, \ldots, R)$$

**Solution**

$$\mathbf{u} = \mathcal{K}x_0 = \begin{bmatrix} K_0 \\ \vdots \\ K_{N-1} \end{bmatrix} x_0 \quad \mathcal{K} = -\left(\mathcal{R} + F^T\mathcal{Q}F\right)^{-1} F^T\mathcal{Q}G$$

$$F = -\mathcal{A}^{-1}\mathcal{B} \quad G = \mathcal{A}^{-1}\mathcal{C}$$

## Comparison

- Although the $u$ with different equation, but they are the same in several simulation in MATLAB
- Because their $u$ based on different reference, so the $K(\mathrm{DB})$ may not equal to $K(\mathrm{FRH})$

**Dynamic Programming**

**Pros:**

- Leads to elegant **closed-form solution** for LQR
- Provides a solution when $N \to \infty$

**Cons:**

- Virtually no problems have simple, closed-form solutions (except LQR)

**Optimization / Least-squares**

**Pros:**

- Can **extend** to nonlinear, constrained systems with complex cost-functions

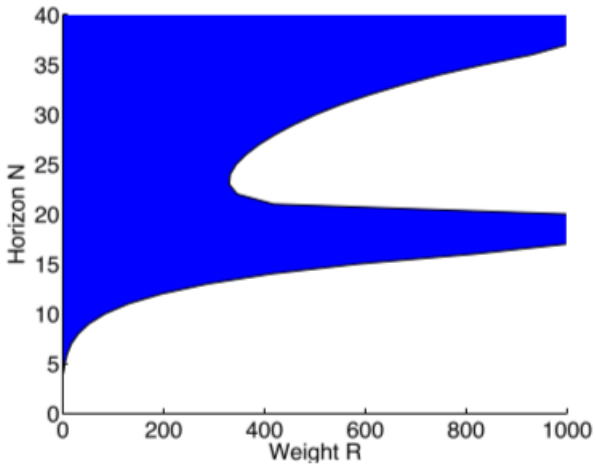
**Cons:**

- Finite-horizon only
- More computationally intense

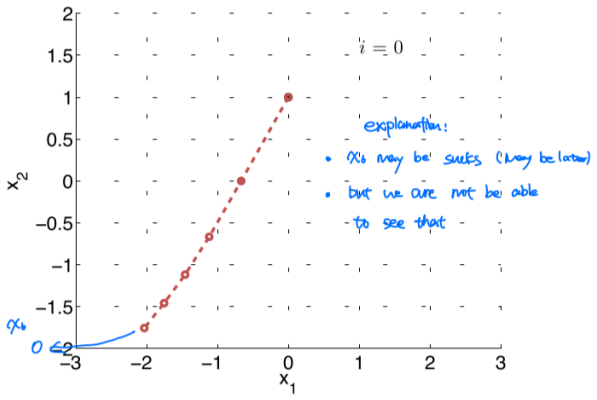
# 3. Stability of LQR Controllers

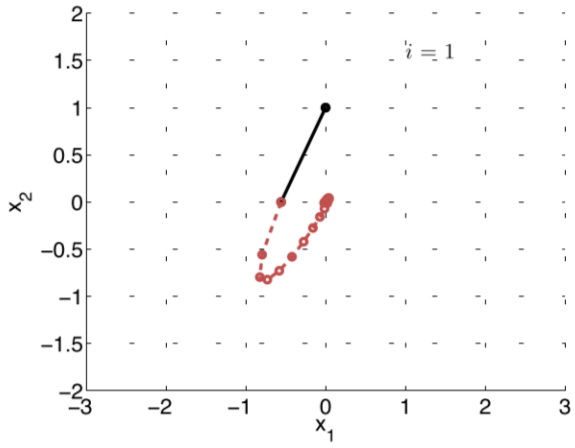
## Problem of Finite Horizon Controller

Because using finite horizon, some early stage decision may be harmful to later stages optimization. One example of stability graph for system $x^+ = (A + BK_{R,N})\,x$ in the following figure.




Another example for effect of length of the horizon:






## Extending to Infinite Horizon

**Assumption**

Assume system is **controllable**, i.e. have input sequence that generates a bounded cost. This is only possible when $x$, $u$ finally goes to zero.

**Solution:**

First, the optimal $V$ should meet: $V^\star(x) := \min_u I(x,u) + V^\star(Ax+Bu)$. Second, there is a fact (can be proved by some complex mathematical theory): $V^*(x)$ is a quadratic function with form $V^*(x) = x^T P x$ with $P$ positive definite. Then we can prove :

**Theorem (Lyapunov Function for LQR)**

The optimal value function $V^\star(x) = x^\top P x$ is a **Lyapunov function** for the system $x^+ = (A+BK)x$ where $K = -\left(R + B^T P B\right)^{-1} B^T P A$ and $P$ solves

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

for some $Q \succeq 0, R \succ 0$

**Conclusion**

- Finite horizon LQR converges to static solution as $N \to \infty$
- Infinite-horizon LQR is nominally stabilizing

# Summary

- The **LQR controller**, we widely used it because it is **solvable**.
- For a **Finite-Horizon LQR controller**, we can solve it by **Dynamic Programming** or **Optimization Method (Receding Horizon Method)**. One big problem of Finite-Horizon LQR Controller is the finite-horizon **may cause the system unstable**.

- Theoretically, an **Infinite-Horizon LQR Controller** can always **stable**, and a Finite-Horizon LQR Controller will **converge** to the Infinite-Horizon Controller as the horizon increase.
- We can also get the **closed-loop solution** for Infinite-horizon LQR Controller.