

1. 最优化控制和基本概念

Single Input Single Output System

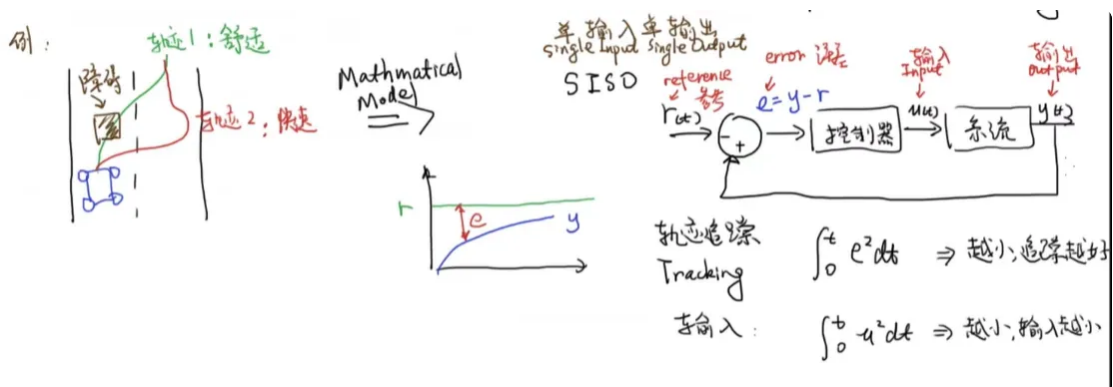
Multiple Input Multiple Output (MIMO)

MPC

概念：

MPC的三个步骤：

Single Input Single Output System



代价 / 目标函数 (Cost / Object Function) :

$$J = \int_0^t qe^2 dt + ru^2 dt$$

前一项越小, 说明误差越小, 效果越好; 后一项越小, 说明控制输入量越小, 系统的能耗也越小;

最终要确定调节参数 q, r 的值, 使得:

$$\min J$$

这里的 q, r 更像是一种权重。

Multiple Input Multiple Output (MIMO)

Multiple Input Multiple Output
MIMO
多输入多输出.

$$\frac{dx}{dt} = Ax + Bu$$

↑ 状态变量

$$y = Cx$$

$$J = \int_0^{\infty} E^T Q E + U^T R U dt$$

例:

$$\frac{d}{dt} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = A \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + B \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad K = \begin{bmatrix} r_1 \\ r_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Rightarrow E = \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} = \begin{bmatrix} y_1 - r_1 \\ y_2 - r_2 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

各节

$$E^T Q E = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^T \begin{bmatrix} q_1 & 0 \\ 0 & q_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = q_1 x_1^2 + q_2 x_2^2$$

$$E^T Q E = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^T \begin{bmatrix} q_1 & 0 \\ 0 & q_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = q_1 x_1^2 + q_2 x_2^2$$

$$U^T R U = r_1 u_1^2 + r_2 u_2^2$$

Q, R: 调节系数. q_1, q_2, r_1, r_2 不重复系数.

MPC

概念:

通过模型来预测系统在某一未来时间段的表现来进行优化控制。

常用离散型状态空间表达:

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

MPC的三个步骤:

Step 1: 在 k 时刻, 测量/估计当前系统状态;

Step 2: 基于 $u_k, u_{k+1}, \dots, u_{k+N-1}$ 来进行最优化控制;

$$J = \sum_k^{N-1} E_k^T Q E_k + U_k^T R U_k + E_N^T F E_N$$

$E_N^T F E_N$ 表示预测区间最后时刻的代价。

Step 3: 在 k 时刻, 只实施 u_k 。滚动优化控制 (Receding Horizon Control)

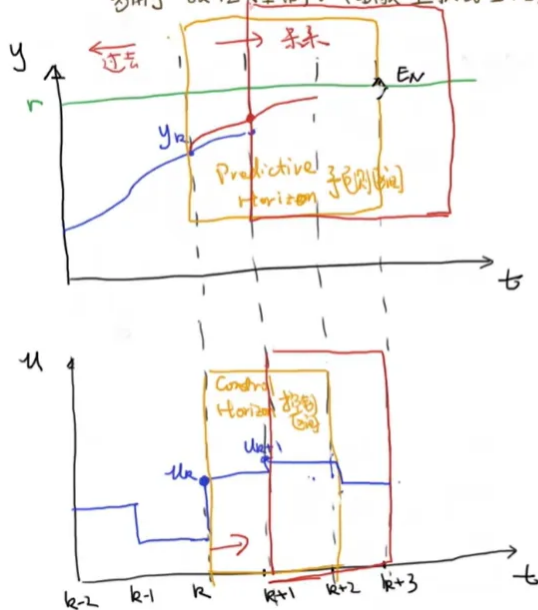
从上可以看出, **MPC**每一步都需要求解最优化的控制, 因此需要消耗大量计算资源。

MPC: 模型预测控制

通过模型来预测系统在某一采样时间内的表现并进行优化控制。

多用于 数位 控制。离散型状态空间表示

$$X_{k+1} = AX_k + BU_k$$



3 steps:

在 k 时刻:

Step 1: 估计/测量读取当前系统状态。

Step 2: 基于 $u_k, u_{k+1}, \dots, u_{k+N}$ 来进行最优化。

$$J = \sum_{k=0}^{N-1} E_k^T Q E_k + u_k^T R u_k + \underbrace{E_N^T F E_N}_{\text{最终 Terminal}}$$

★ step 3: 只取 u_k

Receding Horizon Control
滚动优化控制。

→ 考虑 Constraints 约束



为什么要引入MPC控制? 最优控制中的代价函数需要计算从0时刻到正无穷时刻的积分, 这是一种很贪婪的行为, 需要消耗大量算力; 同时, 系统如果是一个时变系统, 或者面临扰动的话, 前一时刻得到的最优并不一定是下一时刻的最优值。