2. 最优化数学建模推导

MPC的三个步骤：

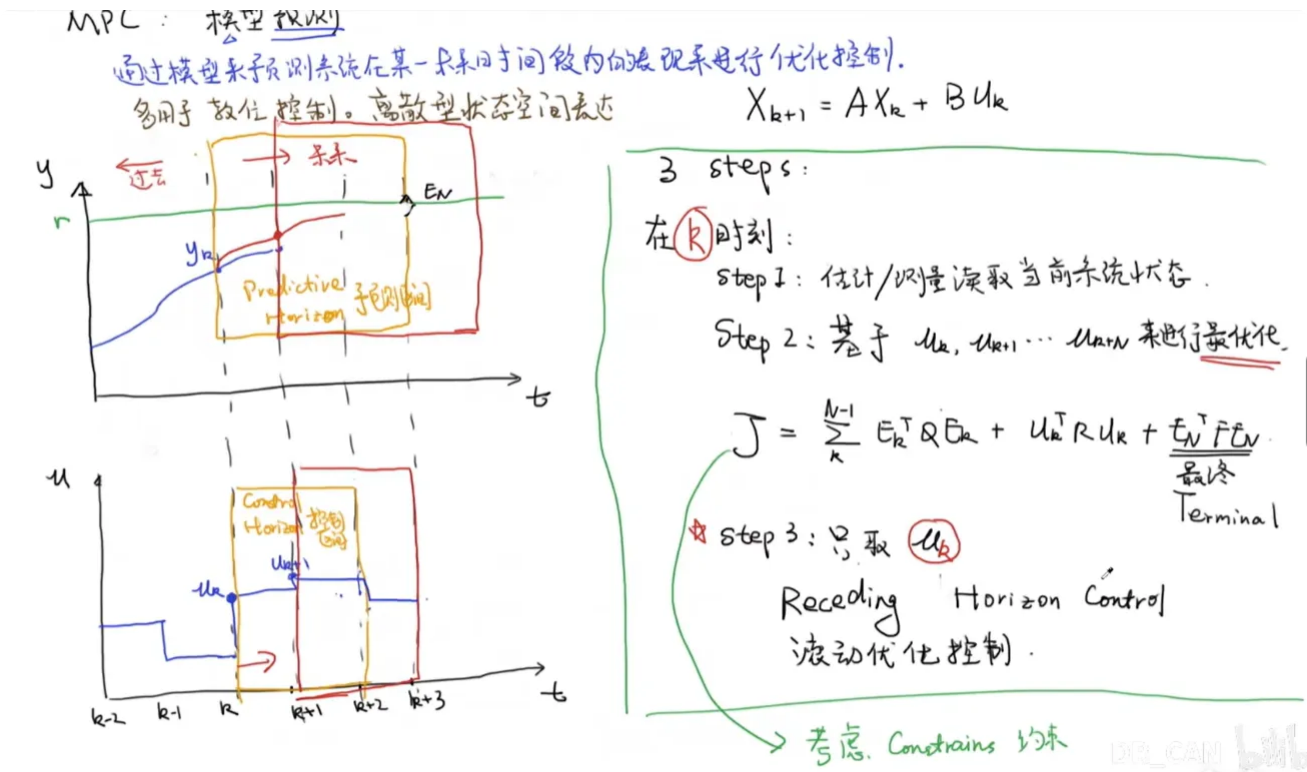
Step 1: 在 k 时刻，测量/估计当前系统状态；

Step 2: 基于 $\boldsymbol{u}_k, \boldsymbol{u}_{k+1}, \ldots, \boldsymbol{u}_{k+N-1}$ 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 $\boldsymbol{u}_k$ 。滚动优化控制（Receding Horizon Control）



常用的最优化策略

常用的最优化策略：二次规划

一般形式：

$$\min Z^T QZ + C^T Z \qquad (2)$$

二次规划的求解器现在有比较成熟的库/包（Matlab、Python等），我们要做的主要是**建立模型并化成一般形式**，然后用求解器求解。

MPC代价函数推导

对于一个系统的状态方程：

$$\boldsymbol{x}(k+1) = \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}\boldsymbol{u}(k) \qquad (3)$$

其中， $\boldsymbol{x} = [x_1, x_2, \ldots, x_n]^T_{n\times 1}$ 表示系统的状态向量， $\boldsymbol{u} = [u_1, u_2, \ldots, u_n]^T_{n\times 1}$ 表示系统的控制输入向量， $\boldsymbol{A}_{\boldsymbol{n}\times\boldsymbol{n}}$ 是系统的状态矩阵， $\boldsymbol{B}_{\boldsymbol{n}\times\boldsymbol{n}}$ 是系统的输入矩阵。

> 在 k 时刻， $\boldsymbol{u}_{(k+N-1|k)}$ 表示在 k 时刻预测到 $k+N-1$ 时刻的控制输入， N 表示预测区间； $\boldsymbol{x}_{(k+N|k)}$ 表示在 k 时刻预测到 $k+N$ 时刻的系统状态。

为了简化推导过程，作出如下假设：

1. 系统的输出 $\boldsymbol{y}(k) = \boldsymbol{x}(k)$ ；
2. 系统的参考值（期望值） $\boldsymbol{R}(\text{Reference}) = 0$ 。

那么误差 $E = Y - R = X - 0 = X$

代价函数：

$$J = \sum_{i=0}^{N-1}(\boldsymbol{x}^T_{(k+i|k)}\boldsymbol{Q}\boldsymbol{x}_{(k+i|k)} + \boldsymbol{u}^T_{(k+i|k)}\boldsymbol{R}\boldsymbol{u}_{(k+i|k)}) + \boldsymbol{x}^T_{(k+N|k)}\boldsymbol{F}\boldsymbol{x}_{(k+N|k)} \qquad (4)$$

其中， $\sum_{i=0}^{N-1}\boldsymbol{x}^T_{(k+i|k)}\boldsymbol{Q}\boldsymbol{x}_{(k+i|k)}$ 表示**误差加权和**， $\sum_{i=0}^{N-1}\boldsymbol{u}^T_{(k+i|k)}\boldsymbol{R}\boldsymbol{u}_{(k+i|k)}$ 表示**输入加权和**， $\boldsymbol{x}^T_{(k+N|k)}\boldsymbol{F}\boldsymbol{x}_{(k+N|k)}$ 表示系统在预测最后时刻的误差，称为**终端误差。**

注意，实际情况下我们可能还需要考虑模型的约束。

初始状态下， $\boldsymbol{x}_{(k|k)} = \boldsymbol{x}_k$ ， $\boldsymbol{x}_k$ 是我们的初始条件。根据系统的状态方程，有：

$$
\begin{aligned}
\boldsymbol{x}_{(k+1|k)} &= \boldsymbol{A}\boldsymbol{x}_{(k|k)} + \boldsymbol{B}\boldsymbol{u}_{(k|k)} \\
&= \boldsymbol{A}\boldsymbol{x}_{k} + \boldsymbol{B}\boldsymbol{u}_{(k|k)}
\end{aligned} \tag{5a}
$$

$$
\begin{aligned}
\boldsymbol{x}_{(k+2|k)} &= \boldsymbol{A}\boldsymbol{x}_{(k+1|k)} + \boldsymbol{B}\boldsymbol{u}_{(k+1|k)} \\
&= \boldsymbol{A}(\boldsymbol{A}\boldsymbol{x}_{k} + \boldsymbol{B}\boldsymbol{u}_{(k|k)}) + \boldsymbol{B}\boldsymbol{u}_{(k+1|k)} \\
&= \boldsymbol{A}^2\boldsymbol{x}_{k} + \boldsymbol{A}\boldsymbol{B}\boldsymbol{u}_{(k|k)} + \boldsymbol{B}\boldsymbol{u}_{(k+1|k)}
\end{aligned} \tag{5b}
$$

$$
\begin{aligned}
\boldsymbol{x}_{(k+3|k)} &= \boldsymbol{A}\boldsymbol{x}_{(k+2|k)} + \boldsymbol{B}\boldsymbol{u}_{(k+2|k)} \\
&= \boldsymbol{A}(\boldsymbol{A}^2\boldsymbol{x}_{k} + \boldsymbol{A}\boldsymbol{B}\boldsymbol{u}_{(k|k)} + \boldsymbol{B}\boldsymbol{u}_{(k+1|k)}) + \boldsymbol{B}\boldsymbol{u}_{(k+2|k)} \\
&= \boldsymbol{A}^3\boldsymbol{x}_{k} + \boldsymbol{A}^2\boldsymbol{B}\boldsymbol{u}_{(k|k)} + \boldsymbol{A}\boldsymbol{B}\boldsymbol{u}_{(k+1|k)} + \boldsymbol{B}\boldsymbol{u}_{(k+2|k)}
\end{aligned} \tag{5c}
$$

一直迭代下去，我们可以得到：

$$
\boldsymbol{x}_{(k+N|k)} = \boldsymbol{A}^N\boldsymbol{x}_{k} + \boldsymbol{A}^{N-1}\boldsymbol{B}\boldsymbol{u}_{(k|k)} + \cdots + \boldsymbol{B}\boldsymbol{u}_{(k+N-1|k)} \tag{5d}
$$

写成矩阵形式：

$$
X_k = \underset{(n+1)\times 1}{\begin{bmatrix} x_{(k|k)} \\ x_{(k+1|k)} \\ x_{(k+2|k)} \\ x_{(k+3|k)} \\ \vdots \\ x_{(k+n|k)} \end{bmatrix}} = \underbrace{\begin{bmatrix} I \\ A \\ A^2 \\ A^3 \\ \vdots \\ A^n \end{bmatrix}}_{M} x_k + \underbrace{\begin{bmatrix} 0 & 0 & \cdots & 0 \\ A^0B & 0 & \cdots & 0 \\ AB & B & \cdots & 0 \\ A^2B & AB & \ddots & \vdots \\ \vdots & \vdots & & \\ A^{n-1}B & A^{n-2}B & \cdots & B \end{bmatrix}}_{C} \underbrace{\begin{bmatrix} u_{(k|k)} \\ u_{(k+1|k)} \\ u_{(k+2|k)} \\ u_{(k+3|k)} \\ \vdots \\ u_{(k+n-1|k)} \end{bmatrix}}_{V_k} \; n\times 1 \tag{6a}
$$

$$
X_k = M x_k + C V_k \tag{6b}
$$

初始向量

展开Eq.(4)，我们得到：

$$
\begin{aligned}
J =\ & x^T_{(k|k)} Q x_{(k|k)} + x^T_{(k+1|k)} Q x_{(k+1|k)} + \cdots + x^T_{(k+n-1|k)} Q x_{(k+n-1|k)} \\
& + x^T_{(k+n|k)} F x_{(k+n|k)} + \sum_{i=0}^{n-1} u^T_{(k+i|k)} R u_{(k+i|k)} \\
=\ & \begin{bmatrix} x_{(k|k)} \\ x_{(k+1|k)} \\ \vdots \\ x_{(k+n|k)} \end{bmatrix}^T \underbrace{\begin{bmatrix} Q & & \\ & Q & \\ & & \ddots & \\ & & & F \end{bmatrix}}_{\bar{Q}} \begin{bmatrix} x_{(k|k)} \\ x_{(k+1|k)} \\ \vdots \\ x_{(k+n|k)} \end{bmatrix} + \sum_{i=0}^{n-1} u^T_{(k+i|k)} R u_{(k+i|k)} \\
=\ & X_k^T \bar{Q} X_k + \begin{bmatrix} u_{(k|k)} \\ u_{(k+1|k)} \\ \vdots \\ u_{(k+n-1|k)} \end{bmatrix}^T \begin{bmatrix} R & & & \\ & R & & \\ & & \ddots & \\ & & & R \end{bmatrix} \begin{bmatrix} u_{(k|k)} \\ u_{(k+1|k)} \\ \vdots \\ u_{(k+n-1|k)} \end{bmatrix} \\
=\ & X_k^T \bar{Q} X_k + U_k^T \bar{R} U_k
\end{aligned}
$$

也即：

$$
\begin{aligned}
J &= \boldsymbol{X}_k^T \overline{\boldsymbol{Q}} \boldsymbol{X}_k + \sum_{i=0}^{n-1} \boldsymbol{u}^T_{(k+1|k)} \boldsymbol{R} \boldsymbol{u}_{(k+1|k)} \\
&= \boldsymbol{X}_k^T \overline{\boldsymbol{Q}} \boldsymbol{X}_k + \boldsymbol{U}_k^T \overline{\boldsymbol{R}} \boldsymbol{U}_k
\end{aligned} \tag{7}
$$

联立Eq.(6b)和Eq.(7)，得到：

$$
J = (\boldsymbol{M}\boldsymbol{x}_k + \boldsymbol{C}\boldsymbol{U}_k)^T \overline{\boldsymbol{Q}} (\boldsymbol{M}\boldsymbol{x}_k + \boldsymbol{C}\boldsymbol{U}_k) + \boldsymbol{U}_k^T \overline{\boldsymbol{R}} \boldsymbol{U}_k \tag{8}
$$

整理公式，可以得到：

$$
\begin{aligned}
J &= x_k^T \bar{Q} x_k + U_k^T \bar{R} U_k \\
&= (\cancel{M x_k} + C U_k)^T \bar{Q} (M x_k + C U_k) + U_k^T \bar{R} U_k \\
&\quad\; x_k^T M^T \quad + U_k^T C^T \\
&= x_k^T M^T \bar{Q} M x_k + \boxed{x_k^T M^T \bar{Q} C U_k}^T + \boxed{U_k^T C^T \bar{Q} M x_k} + U_k^T C^T \bar{Q} C U_k + U_k^T \bar{R} U_k
\end{aligned} \tag{9}
$$

J 是一个数（标量），而等式右边的第二项和第三项也是一个数，且二者互为转置，因此等式右边的第二项和第三项相等。

令 $\boldsymbol{M}^T\overline{\boldsymbol{Q}}\boldsymbol{M} = \boldsymbol{G}, \boldsymbol{M}^T\overline{\boldsymbol{Q}}\boldsymbol{C} = \boldsymbol{E}, \boldsymbol{C}^T\overline{\boldsymbol{Q}}\boldsymbol{C} + \overline{\boldsymbol{R}} = \boldsymbol{H}$ ，可以将代价函数写成：

$$J = \boldsymbol{x}_k^T \boldsymbol{G} \boldsymbol{x}_k + 2\boldsymbol{x}_k^T \boldsymbol{E} \boldsymbol{U}_k + \boldsymbol{U}_k^T \boldsymbol{H} \boldsymbol{U}_k \qquad (10)$$

最终，我们将代价函数转化成了二次规划的一般形式，如Eq.(2)所示。

x_k 是系统已知的初始状态。