

$$\mathbf{u}_{[k|k]}^* = -\mathbf{k}_{\text{mpc}} \mathbf{x}_{[k|k]} \quad (5.3.18)$$

与 LQR 的控制策略一致,这是一个负反馈系统,实际上它的结果也与 LQR 所得到的结果式(4.4.23a)保持一致,具体请参考下一节的例子。

### 5.3.4 一维案例分析——与 LQR 的比较

在本小节中,我们将使用 MPC 来分析一个简单的一维调节系统,该系统在第 4 章例 4.3.1 中曾经使用 LQR 方法进行过分析和求解,现在我们将比较这两种不同的方法。系统的状态空间方程为

$$\mathbf{x}_{[k+1]} = \mathbf{A}\mathbf{x}_{[k]} + \mathbf{B}\mathbf{u}_{[k]} \quad (5.3.19a)$$

这是一个调节问题,因此性能指标可以定义为

$$J = \frac{1}{2} \mathbf{x}_{[k+N|k]}^T \mathbf{S} \mathbf{x}_{[k+N|k]} + \frac{1}{2} \sum_{i=0}^{N_p-1} [\mathbf{x}_{[k+i|k]}^T \mathbf{Q} \mathbf{x}_{[k+i|k]} + \mathbf{u}_{[k+i|k]}^T \mathbf{R} \mathbf{u}_{[k+i|k]}] \quad (5.3.19b)$$

其中,  $\mathbf{A}=\mathbf{B}=\mathbf{S}=\mathbf{R}=\mathbf{Q}=[1]$ 。定义预测区间为  $N_p=2$ ,初始状态为  $\mathbf{x}_{[k]}=[1]$ 。首先需要计算矩阵  $\Phi$ 、 $\Gamma$ 、 $\Omega$ 、 $\Psi$ 、 $\mathbf{H}$ 、 $\mathbf{F}$ ,可得

$$\Phi = \begin{bmatrix} \mathbf{A} \\ \mathbf{A}^2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad (5.3.20a)$$

$$\Gamma = \begin{bmatrix} \mathbf{B} & \mathbf{0} \\ \mathbf{A}\mathbf{B} & \mathbf{B} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \quad (5.3.20b)$$

$$\Omega = \begin{bmatrix} \mathbf{Q} & \mathbf{0} \\ \mathbf{0} & \mathbf{S} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \Psi = \begin{bmatrix} \mathbf{R} & \mathbf{0} \\ \mathbf{0} & \mathbf{R} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (5.3.20c)$$

代入式(5.3.13)可得

$$\mathbf{H} = \Gamma^T \Omega \Gamma + \Psi = \begin{bmatrix} 3 & 1 \\ 1 & 2 \end{bmatrix} \quad \mathbf{F} = \Gamma^T \Omega \Phi = \begin{bmatrix} 2 \\ 1 \end{bmatrix} \quad (5.3.20d)$$

代入式(5.3.16)可得

$$\mathbf{U}_{[k]}^* = -\mathbf{H}^{-1} \mathbf{F} \mathbf{x}_{[k|k]} = - \begin{bmatrix} 3 & 1 \\ 1 & 2 \end{bmatrix}^{-1} \begin{bmatrix} 2 \\ 1 \end{bmatrix} \times [1] = - \begin{bmatrix} 0.6 \\ 0.2 \end{bmatrix} \quad (5.3.21)$$

请读者把这一结果与 4.4.2 节中的结果进行比较,可以发现最优控制序列  $\mathbf{U}_{[k]}^*$  与使用 LQR 的结果是一致的,只是在表达形式上有所不同。因为这两种方法求解的是同一个最优化问题。LQR 使用贝尔曼最优化理论从末端向前递归求解;而 MPC 则将未来的控制序列作为求解项,预测未来的情况并使用二次规划进行求解。

同时需要注意的是,通过 LQR 求解得到的是一个反馈矩阵,控制量则需要使用反馈矩阵乘以当前状态变量得到;而通过 MPC 求解得到的最优控制序列则是当前预测区间内的一系列最优控制量。

表 5.3.1 展示了在不同预测区间  $N_p$  的情况下  $k$  时刻的最优控制序列,可以发现当预测区间  $N_p > 6$  时,最优控制序列的第一项将收敛至  $[-0.61803]$ 。这也与 4.4.2 节的结果相同。表 5.3.1 中加粗的数字是为了强调取值来源,不代表矩阵。

表 5.3.1 不同预测区间的最优控制序列

预测区间	$U_{[k]}^*$	$u_{[k k]}^*$
1	$-[0.5]^T$	$[-0.5]$
2	$-[0.6 \quad 0.2]^T$	$[-0.6]$
3	$-[0.61538 \quad 0.23077 \quad 0.07692]^T$	$[-0.61538]$
4	$-[0.61765 \quad 0.23529 \quad 0.08823 \quad 0.02941]^T$	$[-0.61765]$
5	$-[0.61798 \quad 0.23595 \quad 0.08989 \quad 0.03371 \quad 0.01124]^T$	$[-0.61798]$
6	$-[0.61803 \quad 0.23605 \quad 0.09013 \quad 0.03434 \quad 0.01288 \quad 0.00429]^T$	$[-0.61803]$
7	$-[0.61803 \quad 0.23607 \quad 0.09016 \quad 0.03443 \quad 0.01312 \quad 0.00492 \quad 0.00164]^T$	$[-0.61803]$

### 5.3.5 一维案例分析——MPC 控制器的反馈特性

在 MPC 的实际应用中,只会将求解出的最优控制序列  $U_{[k]}^*$  的第一项  $u_{[k|k]}^*$  施加到系统中,然后在下一时刻重新计算新的控制序列,这就需要控制器有足够的运算资源和存储空间。那么,为什么要“舍掉”  $U_{[k]}^*$  中后面的结果而不是直接按照最优控制序列来控制系统?为什么不通过离线计算的方式算好后面每一时刻的最优控制量再按照时间序列输入到系统当中?

本小节将结合前面的一维案例进行分析说明。

以预测空间  $N_p = 5$  为例,图 5.3.2(a) 的上半部分是  $k=0$  时刻计算得到的控制序列  $U_{[0]}^*$ ,下半部分是在线计算在每个时刻的最优控制序列  $U_{[k]}^*$ 。可以发现  $U_{[0]}^*$  序列里的各个元素与  $U_{[i]}^*$  中的第一项之间的差距可以忽略不计。图 5.3.2(b) 显示了直接使用  $U_{[0]}^*$  与使用  $U_{[i]}^*$  控制序列中第一项这两种算法时系统的表现,离线和在线的控制效果几乎是一致的。图 5.3.2 中,加粗的数字是为了强调取值来源,不代表矩阵。

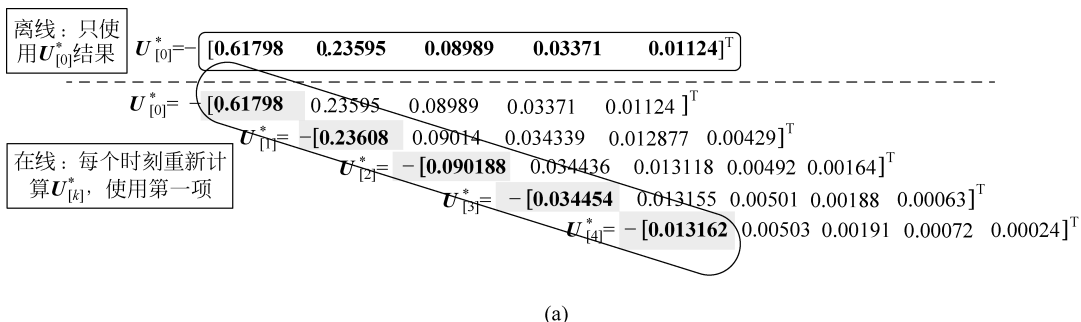


图 5.3.2 “完美”系统在线与离线算法比较