

第四章 模型预测控制

模型预测控制 (MPC) 起源于七十年代末, 最早是过程工业中基于最优控制理论、针对实际困难发展起来的一类控制设计方法, 目前已经广泛用于机器人、自动驾驶等各类系统。模型预测控制的基本思想是通过预测模型在不断向前滚动的有限时间窗口内对未来输出进行预测, 基于预测和优化获得下一步的控制信号, 使系统能够长期稳定运行并适应系统和环境的变化。各种 MPC 算法具有基本相同的结构, 仅在预测模型和成本函数的选择有所不同。

模型预测控制的优势主要体现如下:

- 类似于 PID 调节器, 概念直观, 设计过程中能够较方便的处理约束, 调节相对容易, 对控制知识有限的工程人员非常友好。
- 可用于控制具有延时、不稳定、多变量系统、外扰等复杂特性的系统, 设计过程中可以自然与前馈控制结合。

MPC 在应用中的主要局限在于对实时计算具有较高的要求。一方面, 如果系统的动态是时变的, 需要在每个采样时间通过辨识调整预测模型并计算控制量; 另一方面, 当考虑非线性和约束时, 所需的计算量随着问题的复杂度升高。对于当今飞速发展的算力, 这个问题在高端应用中已经不是本质困难, 但在许多工业过程控制中的确存在, 而且需要分配于控制算法本身以外的目的 (通信、与操作员的对话、警报、记录命令等)。

§1 基本原理

MPC 的策略与驾驶汽车时使用的控制策略非常相似。驾驶员根据目测获取所需的参考轨迹, 并结合汽车自身动力特性采取控制动作 (油门、制动、转向) 操纵汽车遵循参考轨迹。每个时刻只采取第一个控制动作, 并以滚动窗口方式重复该过程以进行下一次控制。作为对比, 经典控制方案 (如 PID) 根据过去的控制误差执行控制操作, 而没有利用参考轨迹的信息 (即只有反馈而没有前馈控制)。

由于需要在线计算，MPC 通常基于离散时间模型进行，该模型能够预测系统在已知控制输入下未来的输出值。考虑轨迹跟踪问题，假设待跟踪的参考输入轨迹为 $w(k)$ 。MPC 的基本实现过程如下（图 1）：

1. 选择一个固定长度的预测时间窗口宽度 P 。在每个时刻 t ，结合当前实测数据 $y_m(t)$ 、当前控制和预测模型预测其后 N 步的输出 $y_P(t+k|t)$ ， $k = 1, \dots, P$ ，并根据上一步的预测与实测偏差对本次预测进行修正，降低模型误差对控制性能的影响。
2. 选择期望优化的控制窗口宽度 L （通常小于预测窗口 P ）。计算最优的控制策略 $u(t|t), \dots, u(t+L-1|t)$ ，使得该控制序列下模型预测输出 $y(t+1|t), \dots, y(t+L|t)$ 与参考轨迹 $w(t+1), \dots, w(t+L)$ 偏差最小。
3. 发送优化得到第一个控制信号 $u(t|t)$ 到执行机构，其它则摒弃。
4. 进入下个采样周期 $t \rightarrow t+1$ ，并重复上述步骤。

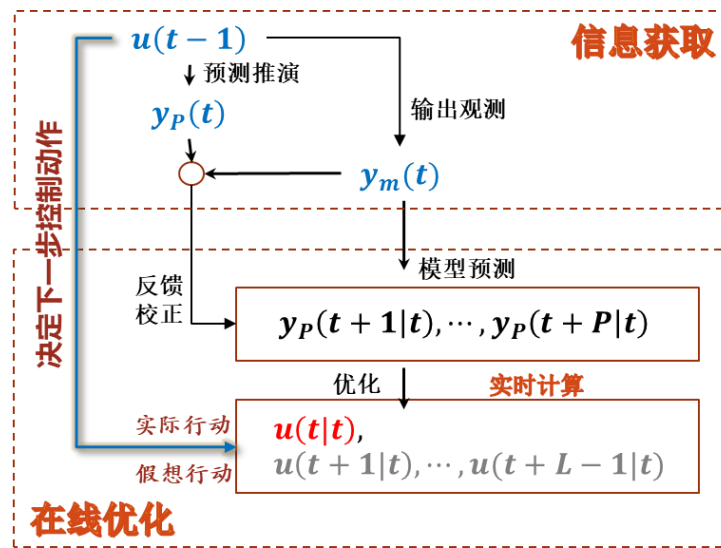


图 1. 模型预测控制基本原理

上述过程中预测模型起着决定性的作用。预测模型必须能够捕获对象的过程动态，以精确预测未来输出，并且易于实施和理解。业内最受欢迎的预测模型是脉冲响应模型和阶跃响应模型，该模型非常直观且容易获得，也可以用于多变量过程，但只能用于开环稳定的控制对象。状态空间模型在学术研究界更为普遍，因为控制器的推导非常简单，即使对象是复杂的多变量系统，而且

更容易对稳定性和鲁棒性进行评估。

优化器是预测控制中的另一个基本部分，优化问题的大小取决于变量的数量和所选的预测范围，如果存在约束或者对优化解的鲁棒性有所要求，则优化所需时间可能会高出多个数量级，使得 MPC 适用的对象带宽大大降低，即只能用于运动较慢、采样周期较长的过程。

MPC 的性能与所提供的预测模型有关，尽管针对预测控制的稳定性和鲁棒性还缺乏足够的理论结果，实践表明 MPC 对模型误差具有一定的鲁棒性，是一种合理的控制策略。本章将针对两种常用的预测模型进行介绍。

§2 基于阶跃响应的动态矩阵法

动态矩阵法是基于非参数化预测模型的预测控制算法。这里参数化模型指的是由一定结构和相应结构参数确定的模型，例如传递函数模型具有有理分式的结构，其参数为分子和分母多项式的系数；状态空间模型具有一阶微分方程组的结构，其参数为矩阵 (A, B, C, D) 。非参数化模型没有明显的结构，但在一定先验假设下可以抽象简化为参数化模型。例如一个系统的阶跃响应或脉冲响应序列直接反映了它的动态特性，如果假设系统是线性定常系统，则可以从响应序列中辨识出传递函数或状态空间模型。

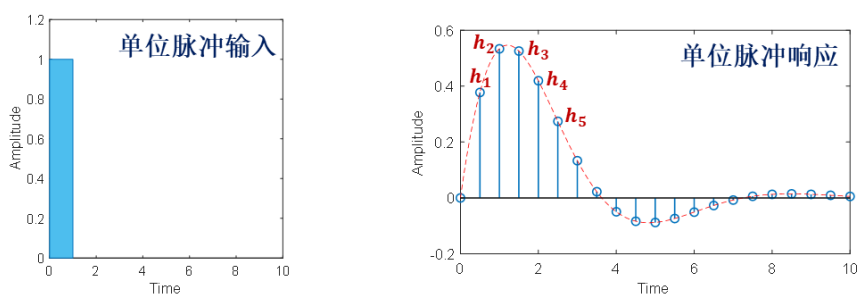


图 2. 离散时间系统的单位脉冲输入和单位脉冲响应

动态矩阵法以阶跃响应为其预测模型。在介绍动态响应法之前，需要介绍一下基于阶跃响应预测输出的基本原理，它与连续时间系统基于脉冲响应的卷积形式有相似之处。假设控制对象为线性定常系统，在单位脉冲输入 $\delta(k)$ 下（即 $\delta(k) = 1$ 仅当 $k = 0$ ）下响应为

$$y(k) = \begin{cases} h_k, & k \geq 0 \\ 0, & k < 0 \end{cases} \quad (2.1)$$

由线性定常系统的时不变特性，在延迟的单位脉冲输入 $\delta(k - k_0)$ 下，响应为

$$y(k) = \begin{cases} h_{k-k_0}, & k \geq k_0 \\ 0, & k < k_0 \end{cases} \quad (2.2)$$

因此对于任意控制序列 $u(k) = \sum_{j=0}^{\infty} u(j)\delta(k - j)$ ，根据线性叠加原理，可得到以如下卷积形式表示的输入输出关系：

$$y(k) = \sum_{j=0}^k h_{k-j}u(j) = \sum_{j=0}^k h_j u(k - j). \quad (2.3)$$

上述关系也可以等价的用阶跃响应表述，其中阶跃响应可由脉冲响应计算：

$$s_k = \sum_{j=0}^k h_j, \quad k = 0, 1, 2, \dots \quad (2.4)$$

其物理意义对应于连续时间系统中阶跃响应为脉冲响应的时域积分。从上式可得

$h_k = s_k - s_{k-1}$ ，代入 (2.3) 得到：

$$y(k) = \sum_{j=0}^k h_j u(k - j) = \sum_{j=1}^k (s_j - s_{j-1})u(k - j) = \sum_{j=1}^k s_j \Delta u(k - j), \quad (2.5)$$

其中 $\Delta u(j) = u(j) - u(j - 1)$ 是控制变量在第 j 个采样周期的增量（见图 3）。

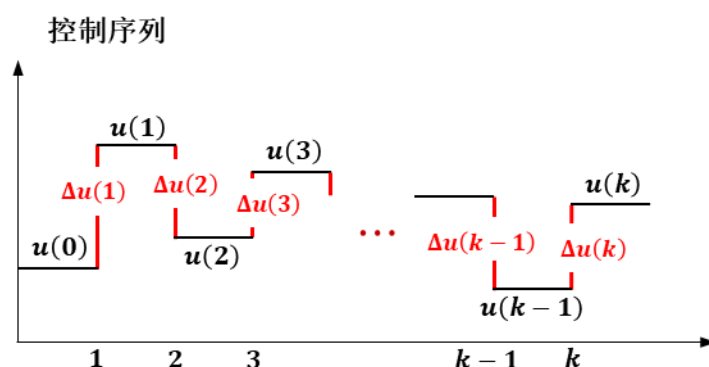


图 3. 控制序列与控制增量序列

这表明系统输出可以表示为阶跃响应序列与控制增量序列的卷积，其连续时间类比为阶跃响应与输入函数时间导数的卷积：

$$y(t) = \int_0^t h(t-\tau)u(\tau)d\tau = \int_0^t s(t-\tau)\dot{u}(\tau)d\tau.$$

由于实际过程中通常希望控制动作尽量平缓, 即控制增量尽量小, 因此基于阶跃响应与控制增量的卷积进行设计更为方便。

下面分析预测控制的设计。假设当前时刻为 k , 则之后第 i 步系统输出可根据(2.5)进行预测, 并可以分解为如下两部分的叠加:

$$y(k+i) = \sum_{j=1}^i s_j \Delta u(k+i-j) + \sum_{j=i+1}^{k+i} s_j \Delta u(k+i-j), \quad (2.6)$$

其中第一部分是 k 时刻之后控制输入的影响, 而第二部分是 k 时刻之前控制输入的影响。

上述预测只包含了由输入产生的强迫响应, 而与初值相关的响应并没有包含在内, 因此是不准确的, 此外模型失配、外部扰动和噪声也会带来预测的误差。这些信息无法从预测模型获得, 但是都间接反映在实测数据中。因此我们在预测(2.6)的基础上, 利用 k 时刻对输出的实测值 $y_m(k)$ 和预测值 $y(k)$ 之间的偏差进行反馈校正:

$$y(k+i|k) \leftarrow y(k+i) + \alpha_i [y_m(k) - \sum_{j=1}^k s_j \Delta u(k-j)], \quad (2.7)$$

其中记号 $y(k+i|k)$ 表示根据 k 时刻的实测值得到的预测值。容易看到, 如果在 k 时刻实测值与预测值正好吻合, 则不需反馈修正。反馈系数需根据经验选取, 为简单起见我们选取 $\alpha_i \equiv 1$ (即假设 k 时刻以后的偏差一直保持不变)。

进一步, 我们根据 (2.6) 的分解将式 (2.7) 整理如下:

$$y(k+i|k) = \sum_{j=1}^i s_j \Delta u(k+i-j) + y_0(k+i|k), \quad (2.8)$$

其中第一部分是 k 时刻以后 (待选) 控制输入 $\Delta u(k), \dots, \Delta u(k+i-1)$ 对输出预测 $y(k+i|k)$ 的影响, 这些输入是后面需要通过优化确定的变量。 k 时刻以前已经发生的控制动作以及反馈校正产生的综合影响包含在第二部分里 (设 $\alpha_i = 1$):

$$\begin{aligned}
y_0(k+i|k) &= \sum_{j=i+1}^{k+i} s_j \Delta u(k+i-j) + \left[y_m(k) - \sum_{j=1}^k s_j \Delta u(k-j) \right] \\
&= y_m(k) + \sum_{j=1}^k (s_{j+i} - s_j) \Delta u(k-j). \tag{2.9}
\end{aligned}$$

综上，如果我们希望预测时间窗口内的 P 步输出 $y_0(k+i|k)$ ，可以用下面公式，基于阶跃响应序列 $\{s_j\}$ 、从初始时刻到当前时刻完成的控制动作 $\{\Delta u(j)\}$ 以及当前的输出测量值 $y_m(k)$ 进行在线计算：

$$\begin{bmatrix} y_0(k+1|k) \\ \vdots \\ y_0(k+P|k) \end{bmatrix} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} y_m(k) + \begin{bmatrix} s_{k+1} - s_k & s_2 - s_1 \\ \vdots & \vdots \\ s_{k+P} - s_k & s_{1+P} - s_1 \end{bmatrix} \begin{bmatrix} \Delta u(0) \\ \vdots \\ \Delta u(k-1) \end{bmatrix}. \tag{2.10}$$

但上式的一个显著问题是，随着时间 k 的增长，第二项里控制增量序列的长度会趋于无穷长，系数矩阵也趋于无穷大，这显然是在线计算无法完成的。

为解决这个问题，我们必须将序列和系数矩阵进行合理截断。注意到系统渐进稳定时阶跃响应会逐渐趋于稳定，因此 $\lim_{k \rightarrow \infty} (s_{k+i} - s_k) = 0$ 。所以可以只保留 $\Delta u(k-1)$ 前若干步控制增量（之前控制对 k 时刻后的输入影响可以忽略不计）。保留步数可以取阶跃响应序列的长度 N （因为这个长度能够保证阶跃响应已经基本达到稳定），这样我们得到实用的计算公式：

$$\begin{bmatrix} y_0(k+1|k) \\ \vdots \\ y_0(k+P|k) \end{bmatrix} \approx \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} y_m(k) + \begin{bmatrix} s_{N+1} - s_N & \cdots & s_2 - s_1 \\ \vdots & \ddots & \vdots \\ s_{N+P} - s_N & \cdots & s_{1+P} - s_1 \end{bmatrix} \begin{bmatrix} \Delta u(k-N) \\ \vdots \\ \Delta u(k-1) \end{bmatrix} \tag{2.11}$$

为方便起见，上式简写为

$$\vec{y}_{P0}(k) \approx y_m(k) \vec{e} + A_{N0} \Delta \vec{u}_N(k), \tag{2.12}$$

它清楚地描述了当前测量值 $y_m(k)$ 和已完成控制动作 $\Delta \vec{u}_N(k)$ 对未来输出预测值的影响。

现在计入未来 L 步控制输入 $\Delta \vec{u}_L(k) = [\Delta u(k), \dots, \Delta u(k+L-1)]^T$ 的影响，预测窗口内的 P 步输出预测值 $\vec{y}_{PL}(k) = [y(k+1|k), \dots, y(k+P|k)]^T$ 可根据式 (2.8) 计算如下：

$$\vec{y}_{PL}(k) = \vec{y}_{P0}(k) + A\Delta\vec{u}_L(k), \quad (2.13)$$

其中 $P \times L$ 维矩阵

$$A = \begin{bmatrix} s_1 & & & \\ s_2 & s_1 & & \\ \vdots & & \ddots & \\ s_{P-1} & s_{P-2} & \cdots & s_{P-L} \\ s_P & s_{P-1} & \cdots & s_{P-L+1} \end{bmatrix}$$

可由系统的阶跃响应通过离线计算获得, 称为动态矩阵, 它的大小由预测窗口长度 P 和待选控制步数 L 决定。

为跟踪某预设轨迹 $w(k)$, 我们希望选择最佳的控制动作 $\Delta\vec{u}_L(k)$ 以尽量减小预测轨迹与预设轨迹的距离, 同时控制动作变化幅度尽量平缓, 因此设定优化目标为

$$\min_{\Delta\vec{u}_L(k)} \|\vec{y}_{PL}(k) - \vec{w}_P(k)\|_Q^2 + \|\Delta\vec{u}_L(k)\|_R^2, \quad (2.14)$$

其中 $\|x\|_P^2 \triangleq x^T P x$, $\vec{w}_P(k) = [w(k+1), \dots, w(k+P)]^T$.

根据预测公式 (2.13), $\vec{y}_{PL}(k)$ 是 $\Delta\vec{u}_L(k)$ 的线性函数, 因此这是一个典型的无约束的二次型优化问题

$$\min_{\Delta\vec{u}_L(k)} \|A\Delta\vec{u}_L(k) + \vec{y}_{P0}(k) - \vec{w}_P(k)\|_Q^2 + \|\Delta\vec{u}_L(k)\|_R^2,$$

其中包含反馈校正信息的 $\vec{y}_{P0}(k)$ 需要在线计算获得, 参考跟踪轨迹 $\vec{w}_P(k)$ 需要事先给定或者在线实时规划, 它们在优化问题中均视为常量。因此, 根据极值条件 $\frac{\partial J}{\partial \Delta\vec{u}_L(k)} = 0$, 容易计算获得最优控制动作:

$$\Delta\vec{u}_L(k) = (A^T Q A + R)^{-1} A^T Q [\vec{w}_P(k) - \vec{y}_{P0}(k)] \quad (2.15).$$

在实际执行过程中, 只采用第一步控制动作, 因此可进一步化简

$$\Delta u(k) = d^T [\vec{w}_P(k) - \vec{y}_{P0}(k)] \quad (2.16)$$

其中 $d^T = c^T (A^T Q A + R)^{-1} A^T Q$, $c^T = [1 \ 0 \ \cdots \ 0]$.

综上, 基于动态矩阵法实现预测控制的流程如图 4 所示。由于 (2.11) 式计算需要历史 N 个控制输入数据, 但在开始阶段没有足够多的历史控制动作,

因此不足部分统一按零计算。若问题存在其它约束，则需要根据具体情况求解。这通常会增加优化求解的时间，甚至出现没有可行解的情况，对控制的实时性提出更高的要求。

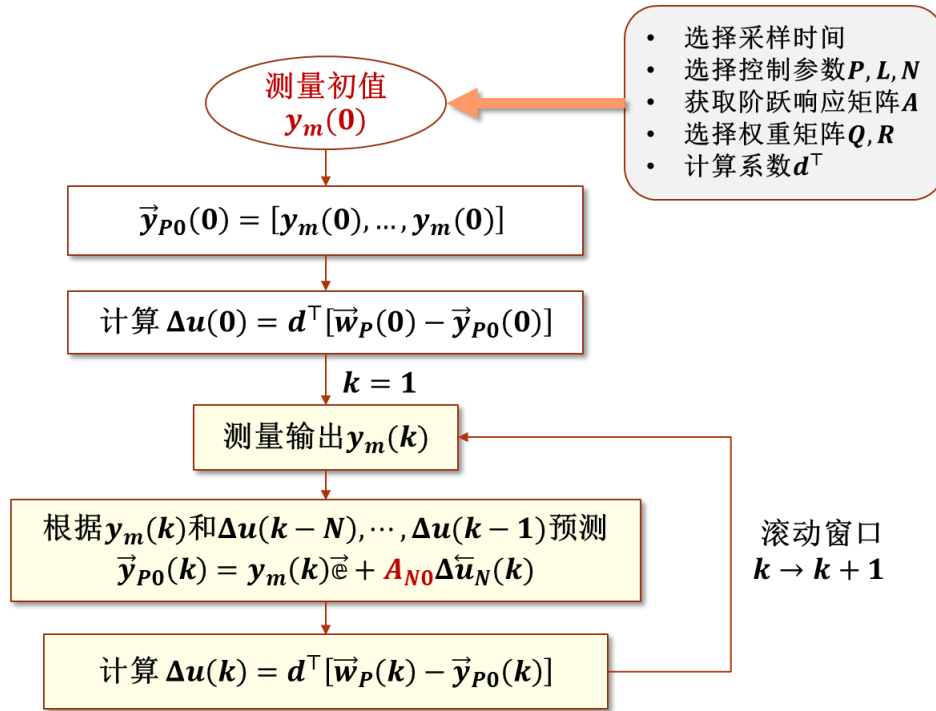


图 4. 动态矩阵法算法流程

从上述分析过程中可以看到，预测控制的核心思想是通过滚动有限时间窗口的优化（receding horizon）实现系统在无限时间区间内的稳定运行，以综合有限时间窗口和无限时间窗口最优控制的优点。由于优化是反复在线进行，被控过程中各种不确定性，如模型的失配、系统的干扰等，均可以反馈形式抑制其对系统的影响，起到实际的鲁棒和抗干扰控制效果。

基于动态矩阵法设计的预测控制性能与多个典型参数的选择有关系，一般应遵循如下原则：

(1) 为精确描述系统运动的细节特征，采样时间应该足够短，但太短时会增加在线计算负担。建议选取适当采样周期使得开环系统响应上升时间段（10%-90%）内有 10-20 个采样。

(2) 预测窗口应足够宽，以描述完整的暂态过程，建议在过渡过程时间内至少设置 20-30 个采样，采样周期按照（1）方案选取。

(3) 控制范围(control horizon)不需要很宽, 一方面会增加计算负担, 另方面对整体性能影响不大, 建议选择预测范围的 20%-30%, 至少 2-3 步。

(4) 权重矩阵 Q, R 根据对跟踪误差和控制动作的要求确定。

例 1: 已知某对象的脉冲响应序列为

$$\mathbf{h} = (0.15, 0.25, 0.2, 0.18, 0.15, 0.08),$$

目标跟踪轨迹为恒值 $w(k) \equiv 10$ 。选择预测步长 $P = 3$, 控制步长 $L = 2$, $Q = I$, $R = 0$, 反馈校正系数 $\bar{\alpha}_3 = [1 \ 1 \ 1]^T$ 。已知对象前三个采样周期的实际输出 $y_m(0) = 9.0$, $y_m(1) = 9.5$, $y_m(3) = 10$ 。利用动态矩阵法计算前三步控制动作 $\Delta u(0)$, $\Delta u(1)$, $\Delta u(2)$ 。

解: 根据脉冲响应容易计算阶跃响应为

$$\mathbf{s} = (0.15, 0.4, 0.6, 0.78, 0.93, 1.01).$$

由此可计算式(2.11)和 (2.13) 中变换矩阵

$$A_{60} = \begin{bmatrix} 0 & 0.08 & 0.15 & 0.18 & 0.20 & 0.25 \\ 0 & 0.08 & 0.23 & 0.33 & 0.38 & 0.45 \\ 0 & 0.08 & 0.23 & 0.41 & 0.53 & 0.63 \end{bmatrix}, \quad A = \begin{bmatrix} 0.15 & 0 \\ 0.4 & 0.15 \\ 0.6 & 0.4 \end{bmatrix},$$

以及计算控制动作的系数矩阵

$$\mathbf{d}^T = \mathbf{c}^T (\mathbf{A}^T \mathbf{Q} \mathbf{A} + \mathbf{R})^{-1} \mathbf{A}^T \mathbf{Q} = [3.04 \quad 3.11 \quad -1.17].$$

第一步 $k = 0$: 此时没有历史控制动作, $\hat{u}_6(0)$ 中 6 个元素全部设为 0, 因此得到预测 $\bar{\mathbf{y}}_{30}(0) = \bar{\mathbf{e}}\mathbf{y}_m(0) = [1 \ 1 \ 1]^T \times 9 = [9 \ 9 \ 9]^T$ 。从而

$$\Delta \mathbf{u}(0) = \mathbf{d}^T [\bar{\mathbf{w}}_3(0) - \bar{\mathbf{y}}_{30}(0)] = 4.983.$$

第二步 $k = 1$: 此时已经有了一个历史控制动作, $\hat{u}_6(1)$ 其余 5 个元素设为 0, 预测 $\bar{\mathbf{y}}_{30}(1) = \bar{\mathbf{e}}\mathbf{y}_m(1) + A_{60}[0 \ 0 \ 0 \ 0 \ 0 \ \Delta \mathbf{u}(0)] = \begin{bmatrix} 10.745 \\ 11.741 \\ 12.637 \end{bmatrix}$, 计算:

$$\Delta \mathbf{u}(1) = \mathbf{d}^T [\bar{\mathbf{w}}_3(1) - \bar{\mathbf{y}}_{30}(1)] = -4.594.$$

第三步 $k = 2$: $\bar{\mathbf{y}}_{30}(2) = \bar{\mathbf{e}}\mathbf{y}_m(2) + A_{60}[0 \ 0 \ 0 \ 0 \ \Delta \mathbf{u}(0) \ \Delta \mathbf{u}(1)] = \begin{bmatrix} 9.848 \\ 9.825 \\ 9.745 \end{bmatrix}$.

$$\Delta \mathbf{u}(2) = \mathbf{d}^T [\vec{\mathbf{w}}_3(2) - \vec{\mathbf{y}}_{30}(2)] = 0.709.$$

§3 基于状态空间模型的设计方法

动态矩阵法概念非常直观并易于实现，但对于较复杂的控制对象需要较强的算力支持，因为需要较长的阶跃响应序列作为预测模型，从而增加内存和在线计算负担（例如矩阵 \mathbf{A}_{N0} 的维数与序列长度有关）。此外，基于输出的反馈所获信息有限，无法利用可测量的状态信息。

本节介绍利用状态空间模型作为预测模型的设计方法，相比于动态矩阵法，它的参数相对较少，并可以充分利用对象的可测信息（包括状态变量及干扰变量），有利于控制系统性能的提高，并可直接用于多变量系统。特别的，这种方法不需要开环系统稳定，因此适用于机器人控制等应用场景。

假设被测对象的状态空间模型为

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k), \quad \mathbf{y}(k) = \mathbf{C}\mathbf{x}(k), \quad (3.1)$$

我们将它作为预测模型。由第三章（3-6）式可知从时刻 k 之后的状态解为

$$\mathbf{x}(k+j) = \mathbf{A}^j \mathbf{x}(k) + \sum_{i=1}^j \mathbf{A}^{j-i} \mathbf{B} \mathbf{u}(k+i-1). \quad (3.2)$$

假设所有状态均可测量（如果不能可引入状态观测器），与动态矩阵法类似，我们通过模型预测 P 步内的状态：

$$\mathbf{X}(k) = \begin{bmatrix} \mathbf{A} \\ \vdots \\ \mathbf{A}^P \end{bmatrix} \hat{\mathbf{x}}(k) + \begin{bmatrix} \mathbf{B} & & \\ \vdots & \ddots & \\ \mathbf{A}^{L-1} \mathbf{B} & \cdots & \mathbf{B} \\ \vdots & \ddots & \vdots \\ \mathbf{A}^{P-1} \mathbf{B} & \cdots & \sum_{j=1}^{P-L} \mathbf{A}^j \mathbf{B} \end{bmatrix} \triangleq \mathbf{F} \hat{\mathbf{x}}(k) + \mathbf{G} \mathbf{U}(k), \quad (3.3)$$

其中 $\hat{\mathbf{x}}(k)$ 为状态 $\mathbf{x}(k)$ 的测量值，

$$\mathbf{X}(k) = \begin{bmatrix} \mathbf{x}(k+1) \\ \cdots \\ \mathbf{x}(k+P) \end{bmatrix}, \quad \mathbf{U}(k) = \begin{bmatrix} \mathbf{u}(k) \\ \cdots \\ \mathbf{u}(k+L-1) \end{bmatrix}.$$

考虑调节器问题，优化目标为：

$$\min_{U(k)} \|X(k)\|_Q + \|U(k)\|_R, \quad (3.4)$$

则根据第三章 (3-8) 式可知最优解为

$$U(k) = -(G^T Q G + R)^{-1} G^T Q F \hat{x}(k). \quad (3.5)$$

只取第一个控制 $u(k)$ 作为下一步动作, 可得到最终的反馈控制律:

$$u(k) = -d^T \hat{x}(k), \quad (3.6)$$

其中 $d^T = e^T (G^T Q G + R)^{-1} G^T Q F$ 可离线计算得到 $e^T = [1, 0, \dots, 0]$ 。

若优化目标选取为输出轨迹跟踪:

$$\min_{U(k)} \|Y(k) - W(k)\|_Q + \|U(k)\|_R, \quad (3.7)$$

则同理根据输出预测序列

$$Y(k) = \begin{bmatrix} CA \\ \vdots \\ CA^P \end{bmatrix} \hat{x}(k) + \begin{bmatrix} CB & & \\ \vdots & \ddots & \\ CA^{L-1}B & \ddots & CB \\ \vdots & \ddots & \vdots \\ CA^{P-1}B & \cdots & \sum_{j=1}^{P-L} CA^j B \end{bmatrix} \triangleq F_o \hat{x}(k) + G_o U(k), \quad (3.8)$$

可求得最优解为

$$u(k) = f^T W(k) - d_o^T \hat{x}(k), \quad (3.9)$$

其中

$$d_o^T = e^T (G_o^T Q G_o + R)^{-1} G_o^T Q_o F_o,$$

$$f^T = e^T (G_o^T Q G_o + R)^{-1} G_o^T Q.$$

从 (3.9) 式可见, 模型预测控制的设计结果自然形成了一个状态反馈与 (按输入) 前馈结合的控制结构, 其中反馈系数和前馈系数分别为 d_o 和 f 。如果系统中有可测量的扰动信号, MPC 中也将包含相应的前馈控制部分以抑制扰动对系统的影响。

若状态不可测, 则需要通过观测器(或滤波器)重构状态:

$$\begin{aligned} \hat{x}(k) &= \hat{x}(k|k-1) + K[y_m(k) - \hat{y}(k|k-1)] \\ &= \hat{x}(k|k-1) + K[y_m(k) - C\hat{x}(k|k-1)], \end{aligned}$$

上述二次型优化指标中引入的惩罚项是针对控制限幅设置的，但应用中常常希望对控制的变化幅度（等价于信号带宽）进行限制，这对应于对其增量的约束：

$$\min_{U(k)} \|Y(k) - W(k)\|_Q + \|\Delta U(k)\|_R, \quad (3.10)$$

其中 $\Delta U(k) = [\Delta u(k), \dots, \Delta u(k+L-1)]$.

为了求解这个问题，我们首先将状态空间模型变换为以控制增量 $\Delta u(k)$ 为输入的模式。注意到：

$$x(k+1) = Ax(k) + B[u(k-1) + \Delta u(k)],$$

可以通过将 $u(k)$ 作为新的状态变量，改写状态方程如下：

$$\begin{bmatrix} x(k+1) \\ u(k) \end{bmatrix} = \begin{bmatrix} A & B \\ 0 & I \end{bmatrix} \begin{bmatrix} x(k) \\ u(k-1) \end{bmatrix} + \begin{bmatrix} B \\ I \end{bmatrix} \Delta u(k)$$

定义扩展状态向量为 $\bar{x}(k) = [x^T(k), u^T(k-1)]^T$ ，则上式可以简写为

$$\bar{x}(k+1) = \bar{A}\bar{x}(k) + \bar{B}\Delta u(k),$$

相应的，输出方程为 $y(k) = [C^T \ 0]\bar{x}(k) = \bar{C}^T\bar{x}(k)$ 。同理可仿照 (3.7) - (3.9)

推导预测控制律，得到如下形式的控制律：

$$u(k) = f^T W(k) - d^T \hat{x}(k) - p^T u(k-1), \quad (3.10)$$

其中除了 (3.9) 式包含的前馈与反馈项以外，还包含了上一步控制动作的影响。

§4 预测控制系统的闭环稳定性

预测控制系统的闭环稳定性分析是个非常困难的问题，并没有 LQR 那样比较简洁的结果。容易看到，在预测控制器中如果令预测窗口趋近于无穷大，即 $P \rightarrow \infty$, $L \rightarrow \infty$ ，则优化问题变为标准的二次型最优控制问题，而二次型最优控制当 $(A, Q^{\frac{1}{2}})$ 能观时一定是渐进稳定的。

但当预测窗口宽度有限时，闭环系统的渐进稳定性并不显然，而且参数选择不当时系统甚至可能失稳，相关条件分析较为复杂。通常在一定约束条件下可得到充分条件，这方面有很多的研究，在此不再展开。