

第四章 模型预测控制

本章内容

— automatic control — automatic control — automatic control — automatic control —

1. 基本思想
2. 动态矩阵法
3. 基于状态空间的方法
4. 应用举例

基本思想

控制系统的设计

— automatic control — automatic control — automatic control — automatic control —

传统控制系统设计：采集**数据**建立**模型**，根据模型设计开环/闭环控制策略。

- 有限拍控制：无法持续工作
- 无限拍控制：系统/环境变化影响；难以处理约束

解决思路：鲁棒控制；抗干扰控制；自适应控制；…

模型预测控制：基于**模型**短期预测 + 结合**数据**在线优化

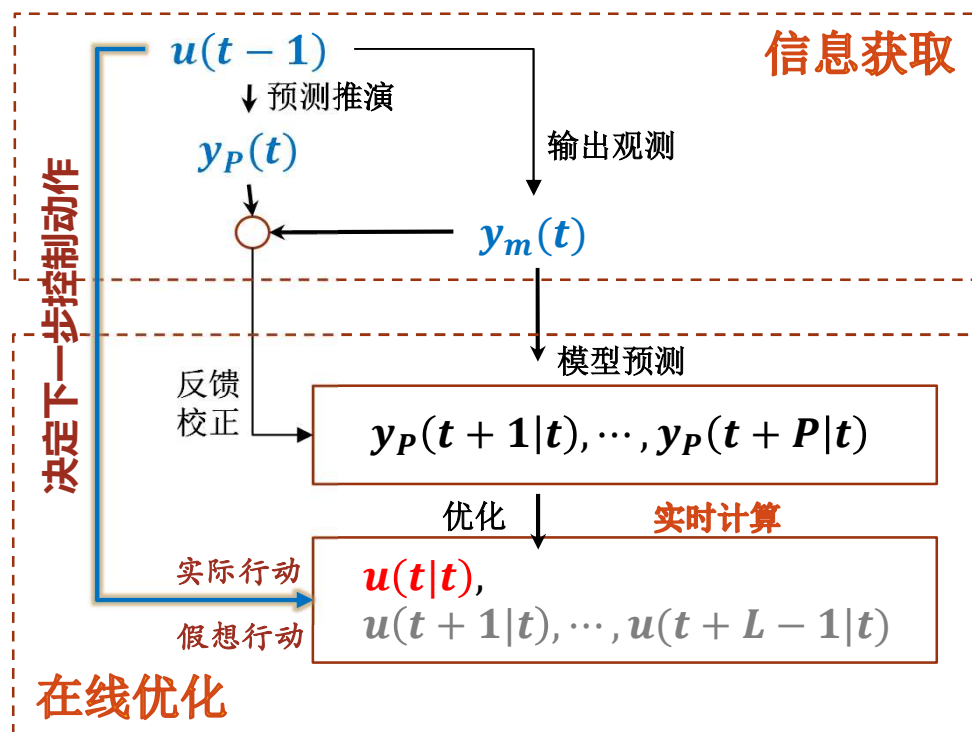
应用：化工过程 → 航空、交通、机器人、…

模型预测控制的基本思想

— automatic control — automatic control — automatic control — automatic control —



目标: $y_p(t+k|t) = w(t+k)$; 策略: $u(t|t), \dots, u(t|t+L-1)$



下棋

- 根据规则（模型）
局势（反馈）预测
- 看三步（优化）
走一步（滚动）



基本结构

— automatic control — automatic control — automatic control — automatic control —

- 根据模型预测设定
时间窗口内的输出

$$\Delta u(k) \xrightarrow{\text{模型}} y_p(k+1), \dots, y_p(k+P)$$

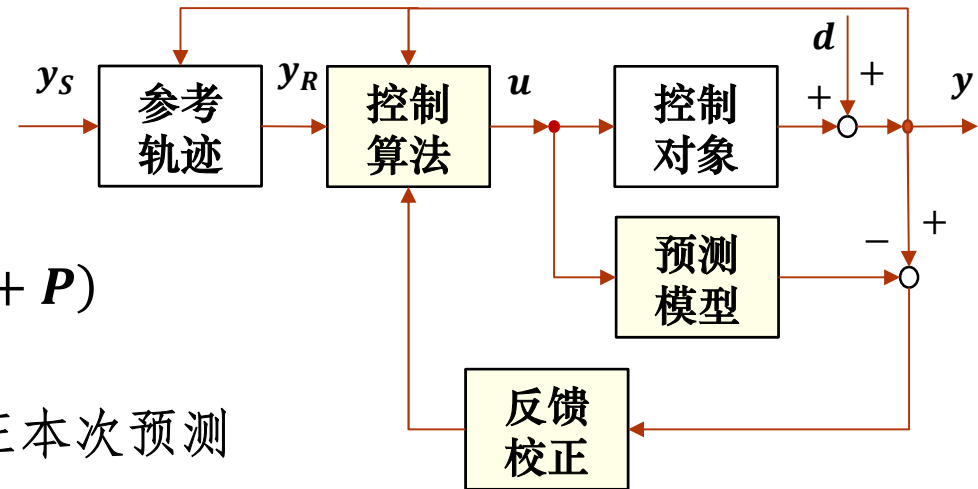
- 根据上步预测与实测数据修正本次预测

$$y(k+i|k) \xrightarrow{\text{校正}} y_p(k+i|k) + \alpha_i [y_m(k) - y_p(k)]$$

- 根据预设轨迹对控制进行开环优化

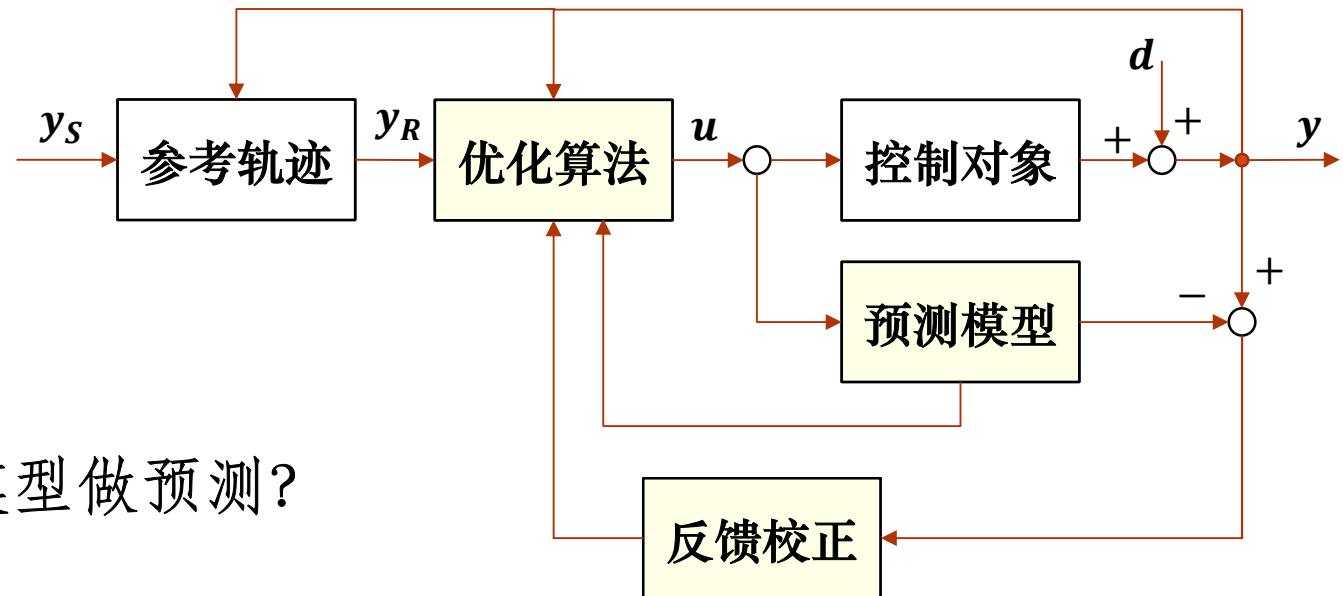
$$\{\Delta u(k), \dots, \Delta u(k+L)\} = \arg \min_{\Delta u(k), \dots, \Delta u(k+L)} \sum_i |y(k+i|k) - y_R(k+i)|^2$$

- 实施控制动作 $\Delta u(k)$



需要解决的问题

— automatic control — automatic control — automatic control — automatic control —



- 用什么模型做预测？
- 怎么预测？如何校正？
- 如何根据预测数据优化控制序列？
- 如何保障闭环系统的性能？

预测模型

— automatic control — automatic control — automatic control — automatic control —

1. 参数化模型 【从数据中辨识】

- 状态空间模型
$$\begin{cases} \mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}u(k) \\ \mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) + \mathbf{D}u(k) \end{cases}$$
- 输入输出模型
$$\sum_{j=0}^n \mathbf{a}_j \mathbf{y}(k-j) = \sum_{i=0}^m \mathbf{b}_i u(k-i)$$

2. 非参数化模型 【直接取自数据】

- 脉冲响应
$$y_m(k) = \sum_{j=1}^N \mathbf{h}_j u(k-j)$$
- 阶跃响应
$$y_m(k) = \sum_{j=1}^N \mathbf{s}_j \Delta u(k-j)$$

优化与分析

— automatic control — automatic control — automatic control — automatic control —

1. 优化

- 通过最小化跟踪误差得到最优控制序列
- 只在有限时间窗口内优化，得到开环解
- 可处理控制受约束情形，对复杂问题实时性要求高

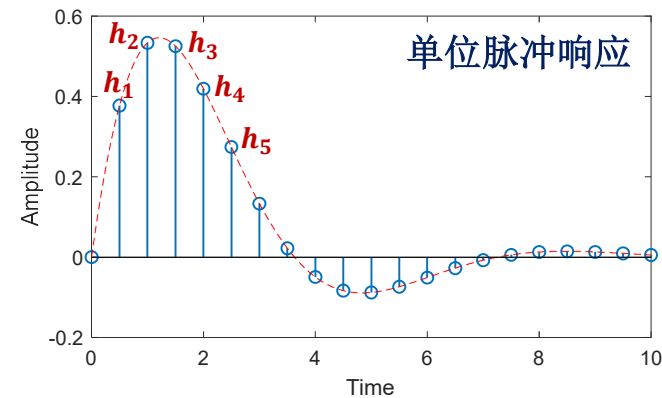
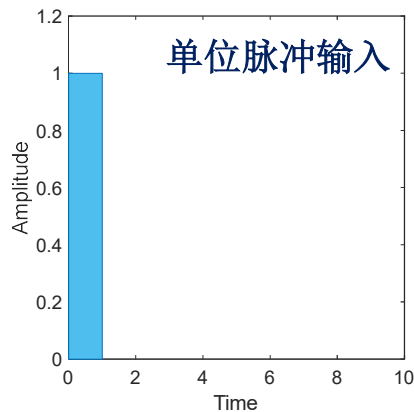
2. 分析

- 闭环稳定性较难分析
- 常常要求控制对象是稳定的，或者先设计校正使之稳定。

动态矩阵法（非参数模型）

单位脉冲响应

— automatic control — automatic control — automatic control — automatic control —



- 单位脉冲输入 $\delta(k)$: $\delta(k) = 1$ 仅当 $k = 0$ ，单位脉冲响应为

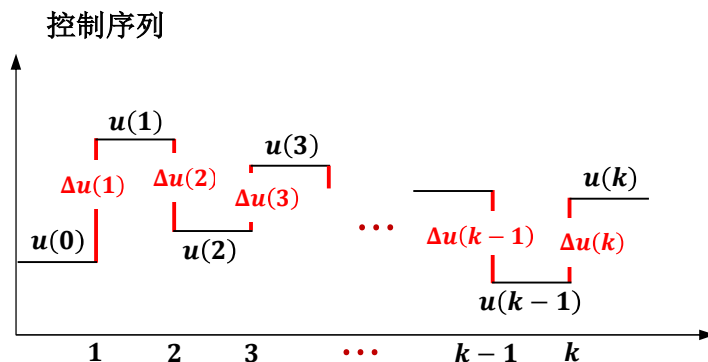
$$y(k) = \begin{cases} h_k, & k \geq 0 \\ 0, & k < 0 \end{cases}$$

- 对于延迟的单位脉冲输入 $\delta(k - k_0)$ ，其响应为

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

基于单位脉冲响应的卷积模型

— automatic control — automatic control — automatic control — automatic control —



$$u_{\text{ZOH}}(t) = u(k), \quad kT < t < (k+1)T$$

$$\begin{aligned} \Delta u(k) &= u(k) - u(k-1), & k \geq 1 \\ \Delta u(k) &= u(0), & k = 0 \end{aligned}$$

- 控制序列可展开为：

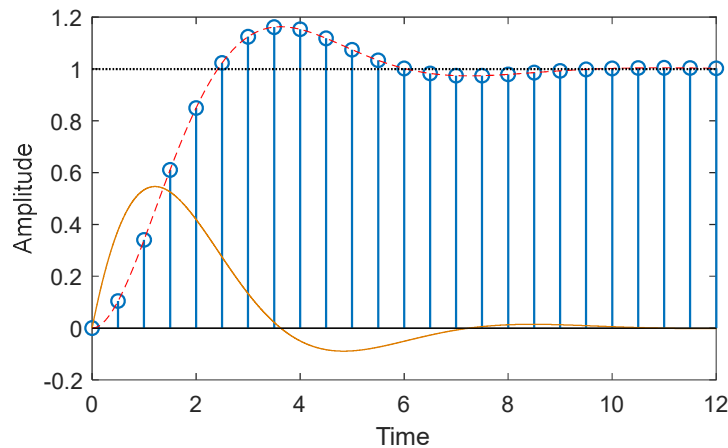
$$u(k) = \sum_{j=0}^{\infty} u(j) \delta(k-j)$$

- 根据线性叠加原理和单位脉冲响应，可得如下卷积关系

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

基于阶跃响应的卷积模型

— automatic control — automatic control — automatic control — automatic control —



阶跃响应是脉冲响应的积分

$$s_k = \sum_{j=0}^k h_j,$$
$$h_k = s_k - s_{k-1}$$

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

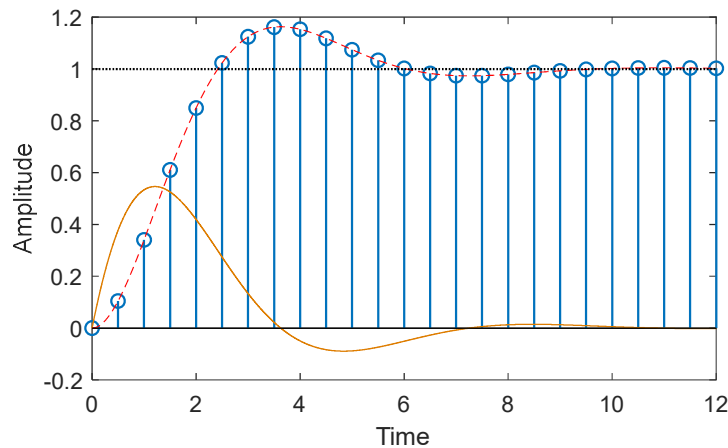
输出可以表示为阶跃响应与控制增量函数的卷积

【对比连续时间形式： $y(t) = \int_0^t h(t-\tau)u(\tau)d\tau = \int_0^t s(t-\tau)\dot{u}(\tau)d\tau$ 】

采用控制增量模型相当于引入了积分环节，有利于消除静差。

基于阶跃响应的卷积模型

— automatic control — automatic control — automatic control — automatic control —

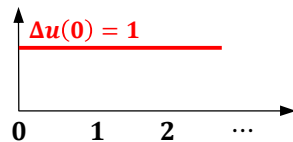


阶跃响应是脉冲响应的“积分”

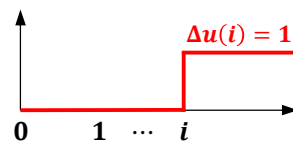
$$s_k = \sum_{j=0}^k h_j,$$

$$h_k = s_k - s_{k-1}$$

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



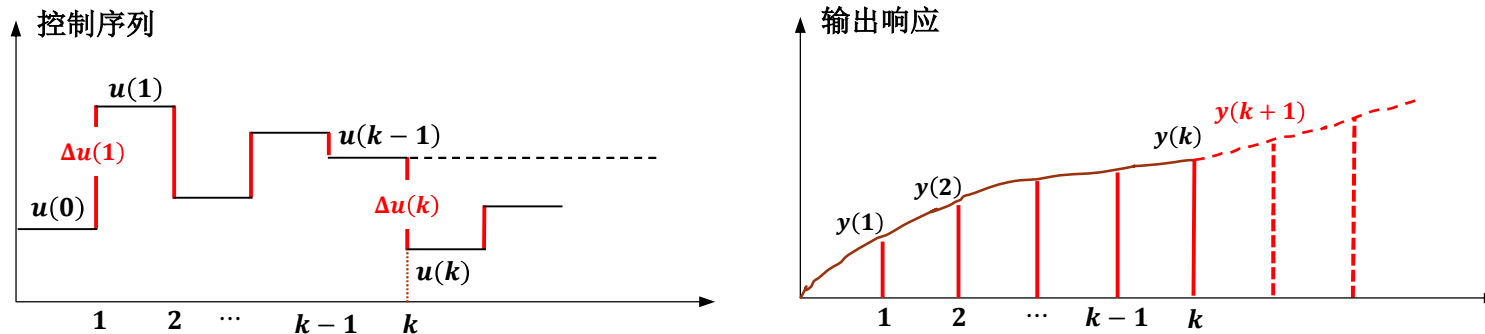
— 单位阶跃输入 $\Delta u(0) = 1$, $y(k) = s_k$



— 单位阶跃输入 $\Delta u(i) = 1$, $y(k) = s_{k-i}$

基于单位脉冲响应的预测

— automatic control — automatic control — automatic control — automatic control —



考察 **k** 时刻起输出与控制输入的关系:

$$\begin{aligned}
 y(k+i) &= \sum_{j=1}^{k+i} s_j \Delta u(k+i-j) \\
 &= \sum_{j=1}^i s_j \Delta u(k+i-j) + \sum_{j=i+1}^{k+i} s_j \Delta u(k+i-j)
 \end{aligned}$$

(待定) 未来控制动作的影响

过去控制动作的影响

反馈校正

— automatic control — automatic control — automatic control — automatic control —

预测与实际输出的偏差:

- 初值产生的自由响应（对象稳定会衰减掉）
- 模型失配产生的偏离（无法利用模型纠正）
- 扰动和噪声产生误差（无法利用模型纠正）

策略：利用测量数据估计偏差 $\mathbf{e}(k) = \mathbf{y}_m(k) - \mathbf{y}(k)$ 校正

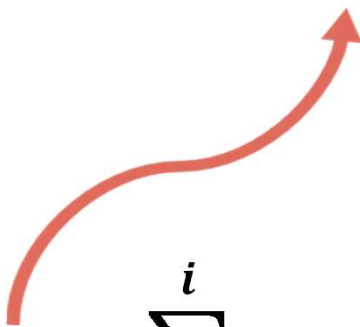
$$\mathbf{y}(k+i|k) \leftarrow \mathbf{y}(k+i) + \alpha_i \left[\mathbf{y}_m(k) - \sum_{j=1}^k \mathbf{s}_j \Delta \mathbf{u}(k-j) \right]$$

其中反馈系数需根据经验选取.

$\alpha_i \equiv 1$ 意味着误差不变 【根据误差先验设置 \leftarrow 内模原理】

反馈校正

— automatic control — automatic control — automatic control — automatic control —

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

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

(待定) 未来控制动作的影响 过去控制动作的影响

校正后的预测

— automatic control — automatic control — automatic control — automatic control —

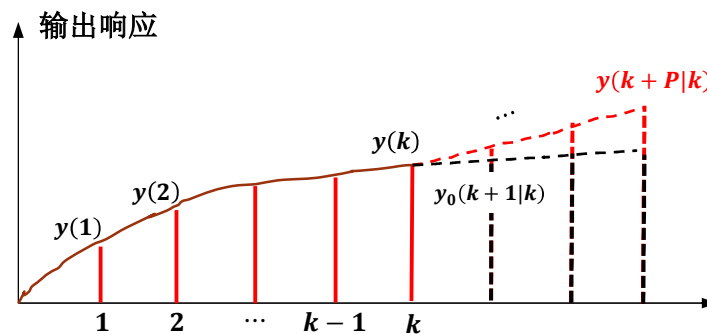
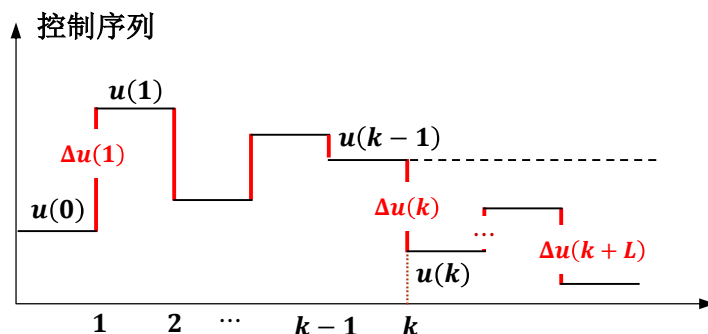
$$\mathbf{y}(k+i|k) = \sum_{j=1}^i s_j \Delta \mathbf{u}(k+i-j) + \mathbf{y}_0(k+i|k)$$

其中 $\mathbf{y}_0(k+i|k)$ 代表如果控制保持不变(即 $\Delta \mathbf{u}(k+i) \equiv \mathbf{0}$) 时经过反馈校正后的预测:

$$\begin{aligned} \mathbf{y}_0(k+i|k) &= \sum_{j=i+1}^{k+i} s_j \Delta \mathbf{u}(k+i-j) + \alpha_i \left[\mathbf{y}_m(k) - \sum_{j=1}^k s_j \Delta \mathbf{u}(k-j) \right] \\ &= \alpha_i \mathbf{y}_m(k) + \sum_{j=1}^k (s_{j+i} - \alpha_i s_j) \Delta \mathbf{u}(k-j) \end{aligned}$$

动态矩阵

— automatic control — automatic control — automatic control — automatic control —



预测 L 步控制动作下 $P(>L)$ 步输出 $y(k+1|k), \dots, y(k+P|k)$:

$$\begin{bmatrix} y(k+1|k) \\ y(k+2|k) \\ \vdots \\ y(k+P-1|k) \\ y(k+P|k) \end{bmatrix} = \begin{bmatrix} y_0(k+1|k) \\ y_0(k+2|k) \\ \vdots \\ y_0(k+P-1|k) \\ y_0(k+P|k) \end{bmatrix} + \begin{bmatrix} s_1 & & & \\ s_2 & s_1 & & \\ \vdots & & \ddots & \\ s_{P-1} & s_{P-2} & \cdots & s_{P-L-1} \\ s_P & s_{P-1} & \cdots & s_{P-L} \end{bmatrix} \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \vdots \\ \Delta u(k+L) \end{bmatrix}$$

简写: $\vec{y}_{PL}(k) = \vec{y}_{P0}(k) + \mathbf{A} \Delta \vec{u}_L(k)$

基于 L 步控制对 P 步输出的预测

动态矩阵
(Dynamical Matrix)

校正后的预测

— automatic control — automatic control — automatic control — automatic control —

$$y_0(k+i|k) = \alpha_i y_m(k) + \sum_{j=1}^k (s_{j+i} - \alpha_i s_j) \Delta u(k-j)$$

$$\begin{bmatrix} y_0(k+1|k) \\ y_0(k+2|k) \\ \vdots \\ y_0(k+P-1|k) \\ y_0(k+P|k) \end{bmatrix}_{P \times 1} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{P-1} \\ \alpha_P \end{bmatrix} y_m(k)$$

计算量随 k 增长！

$$+ \begin{bmatrix} s_{k+1} - \alpha_1 s_k & \cdots & s_3 - \alpha_1 s_2 & s_2 - \alpha_1 s_1 \\ s_{k+2} - \alpha_2 s_k & \cdots & s_4 - \alpha_2 s_2 & s_3 - \alpha_2 s_1 \\ \vdots & \cdots & \vdots & \vdots \\ s_{k+P-1} - \alpha_{P-1} s_k & \cdots & s_{1+P} - \alpha_{P-1} s_2 & s_P - \alpha_{P-1} s_1 \\ s_{k+P} - \alpha_P s_k & \cdots & s_{2+P} - \alpha_P s_2 & s_{1+P} - \alpha_P s_1 \end{bmatrix}_{P \times k} \begin{bmatrix} \Delta u(0) \\ \vdots \\ \Delta u(k-N) \\ \vdots \\ \Delta u(k-1) \end{bmatrix}_{k \times 1}$$

简写： $\vec{y}_{P0}(k) = y_m(k) \vec{\alpha} + \mathbf{A}_0 \Delta \bar{u}(k)$

过去的
控制动作

校正后的预测

— automatic control — automatic control — automatic control — automatic control —

若系统渐进稳定，则足够大时 $\mathbf{s}_{k+i} \approx \mathbf{s}_k$ 。

此时若取 $\alpha_i \equiv 1$ ，则可做如下截断：

$$\begin{aligned} \mathbf{y}_0(k+i|k) &= \mathbf{y}_m(k) + \sum_{j=1}^k (\mathbf{s}_{j+i} - \mathbf{s}_j) \Delta \mathbf{u}(k-j) \\ &\approx \mathbf{y}_m(k) + \sum_{j=1}^N (\mathbf{s}_{j+i} - \mathbf{s}_j) \Delta \mathbf{u}(k-j) \end{aligned}$$

$$\begin{bmatrix} \mathbf{y}_0(k+1|k) \\ \mathbf{y}_0(k+2|k) \\ \vdots \\ \mathbf{y}_0(k+P-1|k) \\ \mathbf{y}_0(k+P|k) \end{bmatrix}_{P \times 1} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \\ 1 \end{bmatrix} \mathbf{y}_m(k) + \begin{bmatrix} \mathbf{s}_{N+1} - \mathbf{s}_N & \cdots & \mathbf{s}_2 - \mathbf{s}_1 \\ \vdots & \ddots & \vdots \\ \mathbf{s}_{N+P} - \mathbf{s}_N & \cdots & \mathbf{s}_{1+P} - \mathbf{s}_1 \end{bmatrix}_{P \times N} \begin{bmatrix} \Delta \mathbf{u}(k-N) \\ \vdots \\ \Delta \mathbf{u}(k-1) \end{bmatrix}_{N \times 1}$$

历史控制的影响最多回溯到前 N 个

简写： $\vec{\mathbf{y}}_{P0}(k) \approx \mathbf{y}_m(k) \vec{\mathbf{e}} + \mathbf{A}_{N0} \Delta \vec{\mathbf{u}}_N(k)$

滚动优化

— automatic control — automatic control — automatic control — automatic control —

为跟踪某预设轨迹 $\mathbf{w}(k)$ ，希望选择最佳的控制动作

$$\Delta \vec{\mathbf{u}}_L = [\Delta \mathbf{u}(k), \dots, \Delta \mathbf{u}(k+L)]^\top$$

以尽量减小预测轨迹与预设轨迹的距离, 并避免控制动作过于激烈, 因此目标设定为

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

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

根据预测公式可知 $\vec{\mathbf{y}}_{PM}(k) = \vec{\mathbf{y}}_{P0}(k) + \mathbf{A} \Delta \vec{\mathbf{u}}_L(k)$, 因此这是一个典型的二次型优化问题。

滚动优化

— automatic control — automatic control — automatic control — automatic control —

无约束的二次型优化问题

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

的解可以由极值条件 $\frac{\partial J}{\partial \Delta \vec{u}_L(k)} = \mathbf{0}$ 获得：

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

实际只采用第一步控制动作，因此得到反馈控制：

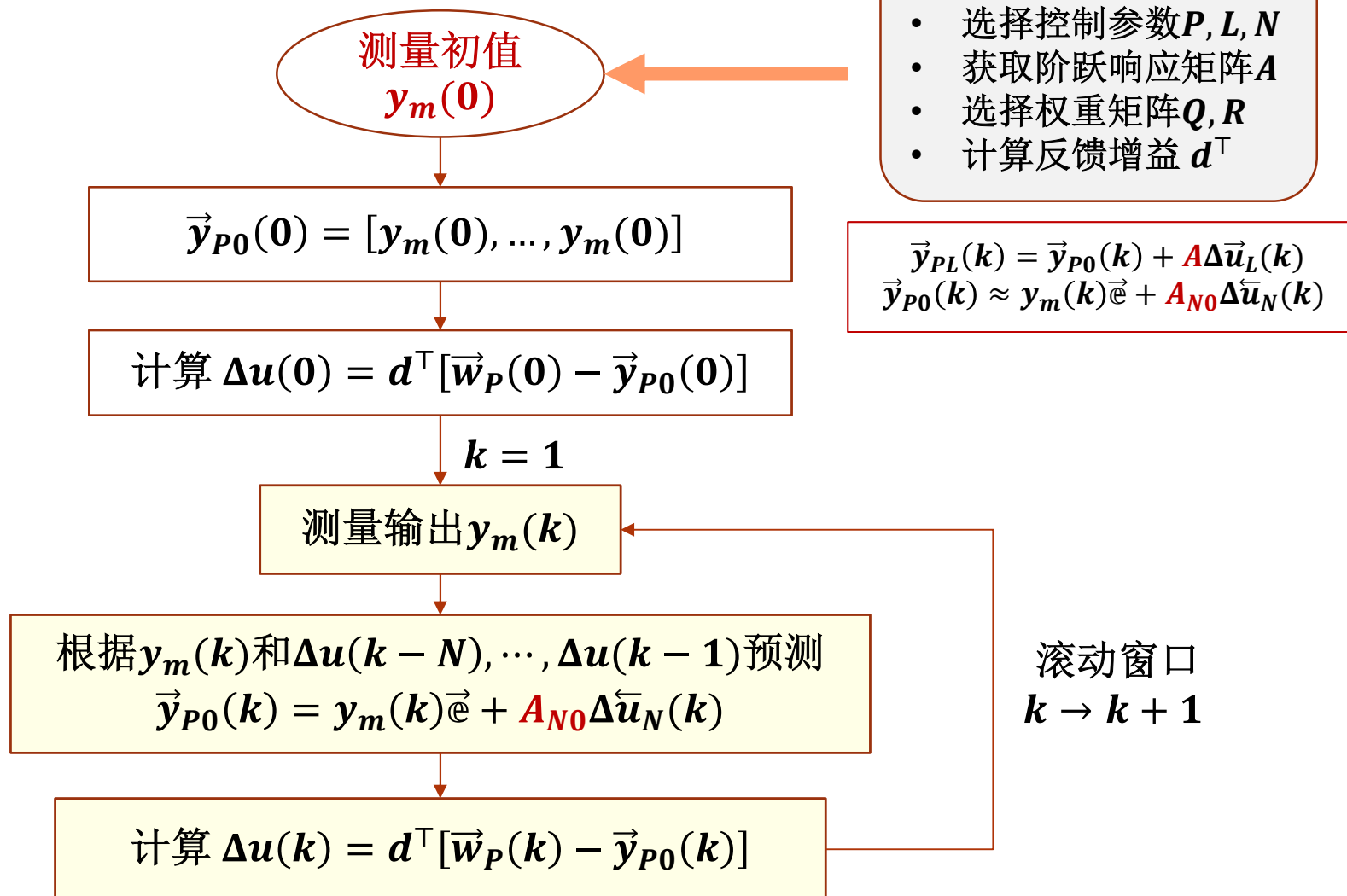
$$\Delta u(k) = d^\top [\vec{w}_P(k) - \vec{y}_{P0}(k)]$$

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

若问题存在约束，则需要根据具体情况求解。

算法流程

— automatic control — automatic control — automatic control — automatic control —



滚动优化

— automatic control — automatic control — automatic control — automatic control —

- 滚动优化不是针对不变的全局优化目标，目标随时间滚动变化且总在有限时域内优化。
- 优化过程不是一次离线进行，而是反复在线进行；
- 被控过程中各种不确定性，如模型的失配、系统的干扰等，均可以反馈形式进行校正并被优化，起到实际的鲁棒和抗干扰控制效果。

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

Diagram illustrating the components of the rolling optimization equation:

- $\Delta \mathbf{u}(k)$ is associated with the matrix \mathbf{D} (labeled "前馈" / Feedforward).
- $\bar{\mathbf{w}}_P(k)$ is associated with the matrix \mathbf{P} (labeled "反馈" / Feedback).
- $\bar{\mathbf{y}}_{P0}(k)$ is associated with the matrix \mathbf{I} (labeled "反馈" / Feedback).

示例1

— automatic control — automatic control — automatic control — automatic control —

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 $\mathbf{w}(k) \equiv 10$ 。

选择预测步长 $P = 3$ ，控制步长 $L = 2$ ， $Q = I$ ， $R = 0$ ，
反馈校正系数 $\vec{e} = [1 \ 1 \ 1]^T$ 。

已知对象实际输出

$$y_m(0) = 9.0, y_m(1) = 9.5, y_m(3) = 10$$

利用动态矩阵控制计算前两步控制动作 $\Delta \mathbf{u}(0)$ 和 $\Delta \mathbf{u}(1)$ 。

示例1

— automatic control — automatic control — automatic control — automatic control —

$$N = 6, P = 3, L = 2$$

$$h = (0.15, 0.25, 0.2, 0.18, 0.15, 0.08) \Rightarrow$$
$$\Rightarrow s = (0.15, 0.4, 0.6, 0.78, 0.93, 1.01)$$

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

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

$$\vec{e} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

示例1

— automatic control — automatic control — automatic control — automatic control —

第一步：

$$\text{预测 } \vec{y}_{30}(\mathbf{0}) = \vec{e} y_m(\mathbf{0}) = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \times 9 = \begin{bmatrix} 9 \\ 9 \\ 9 \end{bmatrix}$$

$$\text{计算: } \Delta u(\mathbf{0}) = \mathbf{d}^\top [\vec{w}_3(\mathbf{0}) - \vec{y}_{30}(\mathbf{0})]$$

$$= [3.04 \quad 3.11 \quad -1.17] \times \left(\begin{bmatrix} 10 \\ 10 \\ 10 \end{bmatrix} - \begin{bmatrix} 9 \\ 9 \\ 9 \end{bmatrix} \right) = 4.983$$

示例1

— automatic control — automatic control — automatic control — automatic control —

第二步 【 $\Delta u(0) = 4.983$ 】：

$$\text{预测 } \vec{y}_{30}(1) = \vec{e}y_m(1) + A_{60} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \Delta u(0) \end{bmatrix} = \begin{bmatrix} 10.746 \\ 11.742 \\ 12.639 \end{bmatrix}$$

$$\text{计算: } \Delta u(1) = d^T [\vec{w}_3(1) - \vec{y}_{30}(1)]$$

$$\begin{aligned} &= [3.04 \quad 3.11 \quad -1.17] \times \left(\begin{bmatrix} 10 \\ 10 \\ 12.637 \end{bmatrix} - \begin{bmatrix} 10.746 \\ 11.742 \\ 12.639 \end{bmatrix} \right) \\ &= -4.606 \end{aligned}$$

第二步控制I

示例1

— automatic control — automatic control — automatic control — automatic control —

第三步 【 $\Delta u(0) = 4.983$, $\Delta u(1) = -4.594$ 】 :

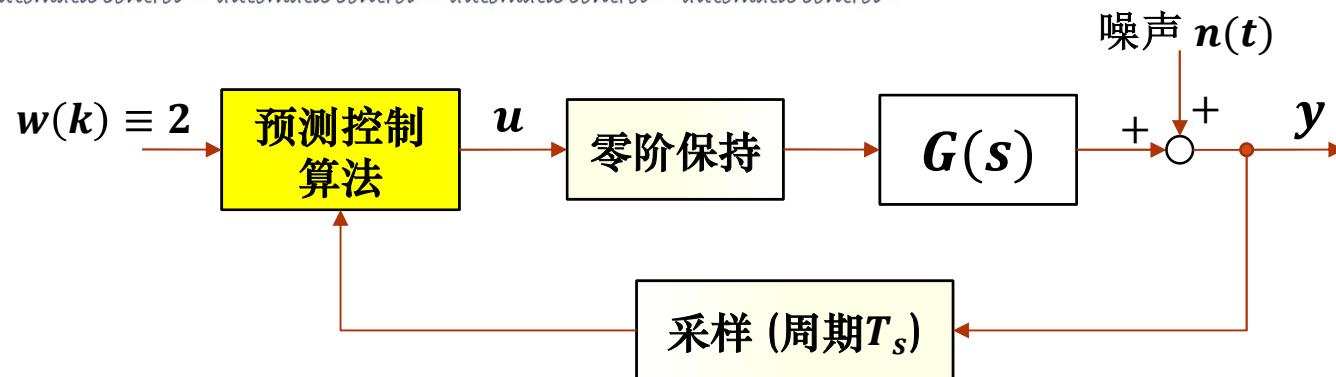
$$\vec{y}_{30}(2) = \vec{e} y_m(2) + A_{60} \begin{bmatrix} 0 \\ \vdots \\ \Delta u(0) \\ \Delta u(1) \end{bmatrix} = \begin{bmatrix} 9.845 \\ 9.821 \\ 9.739 \end{bmatrix}$$

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

$$= [3.04 \quad 3.11 \quad -1.17] \times \left(\begin{bmatrix} 10 \\ 10 \\ 10 \end{bmatrix} - \begin{bmatrix} 9.845 \\ 9.821 \\ 9.739 \end{bmatrix} \right) = 0.724$$

示例2

— automatic control — automatic control — automatic control — automatic control —



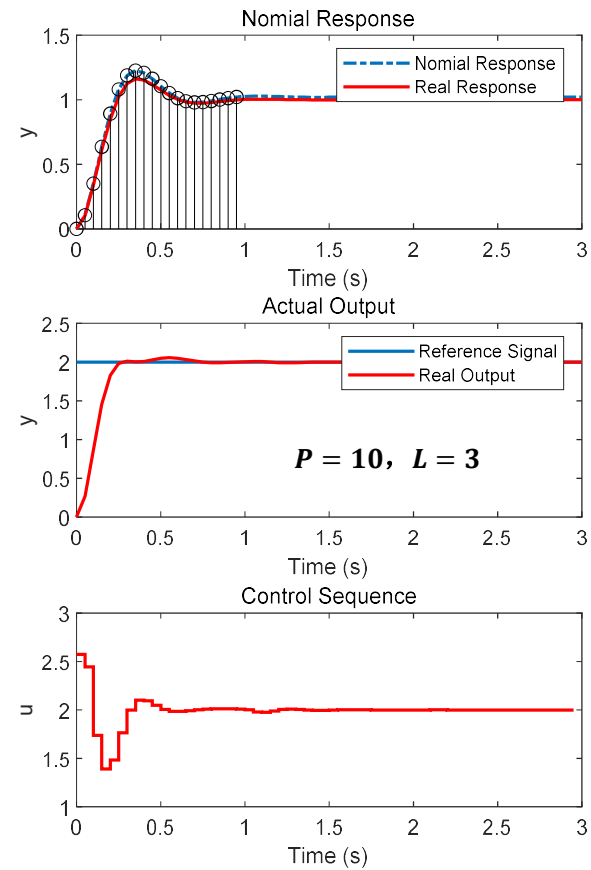
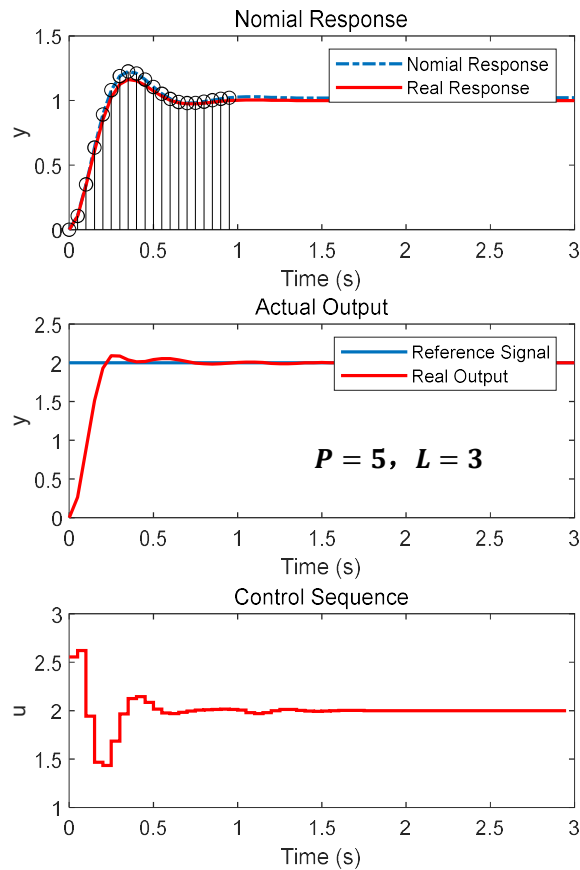
控制对象的标称传递函数为 $\bar{G}(s) = \frac{100}{s^2 + 10s + 100}$. $Q = \mathbb{I}_P$,

$R = r\mathbb{I}_L$ 。

观察控制性能随预测步长 P 和模型失配的影响。

示例2：预测窗口的影响

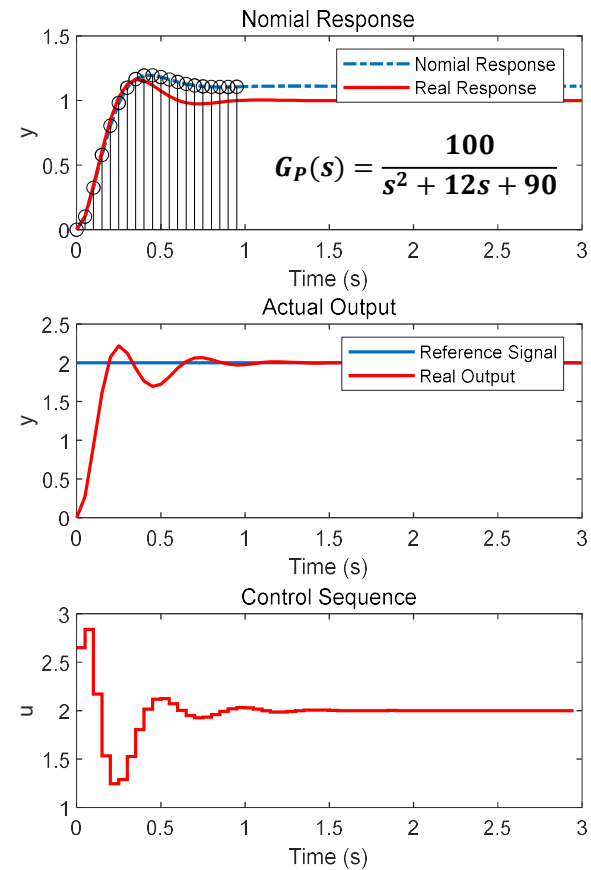
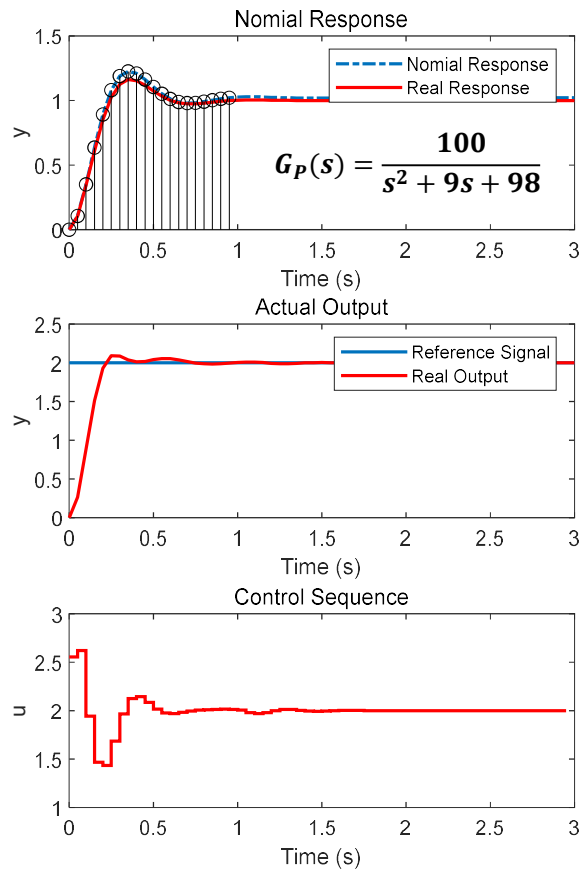
— automatic control — automatic control — automatic control — automatic control —



$$G_P(s) = \frac{100}{s^2 + 9s + 98}, \quad T_s = 0.05, \quad N = 20$$

示例2：模型失配的影响

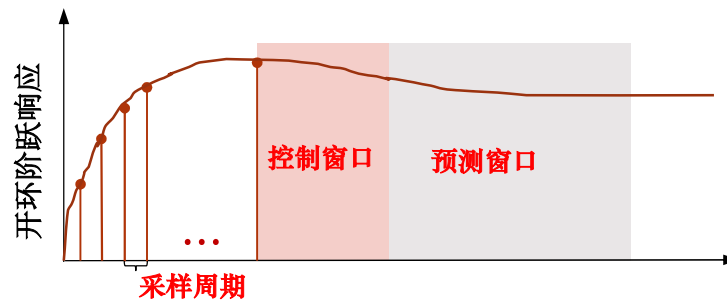
— automatic control — automatic control — automatic control — automatic control —



$T_s = 0.05$, $N = 20$, 预测步长 $P = 5$, 控制步长 $L = 3$

参数选择

— automatic control — automatic control — automatic control — automatic control —



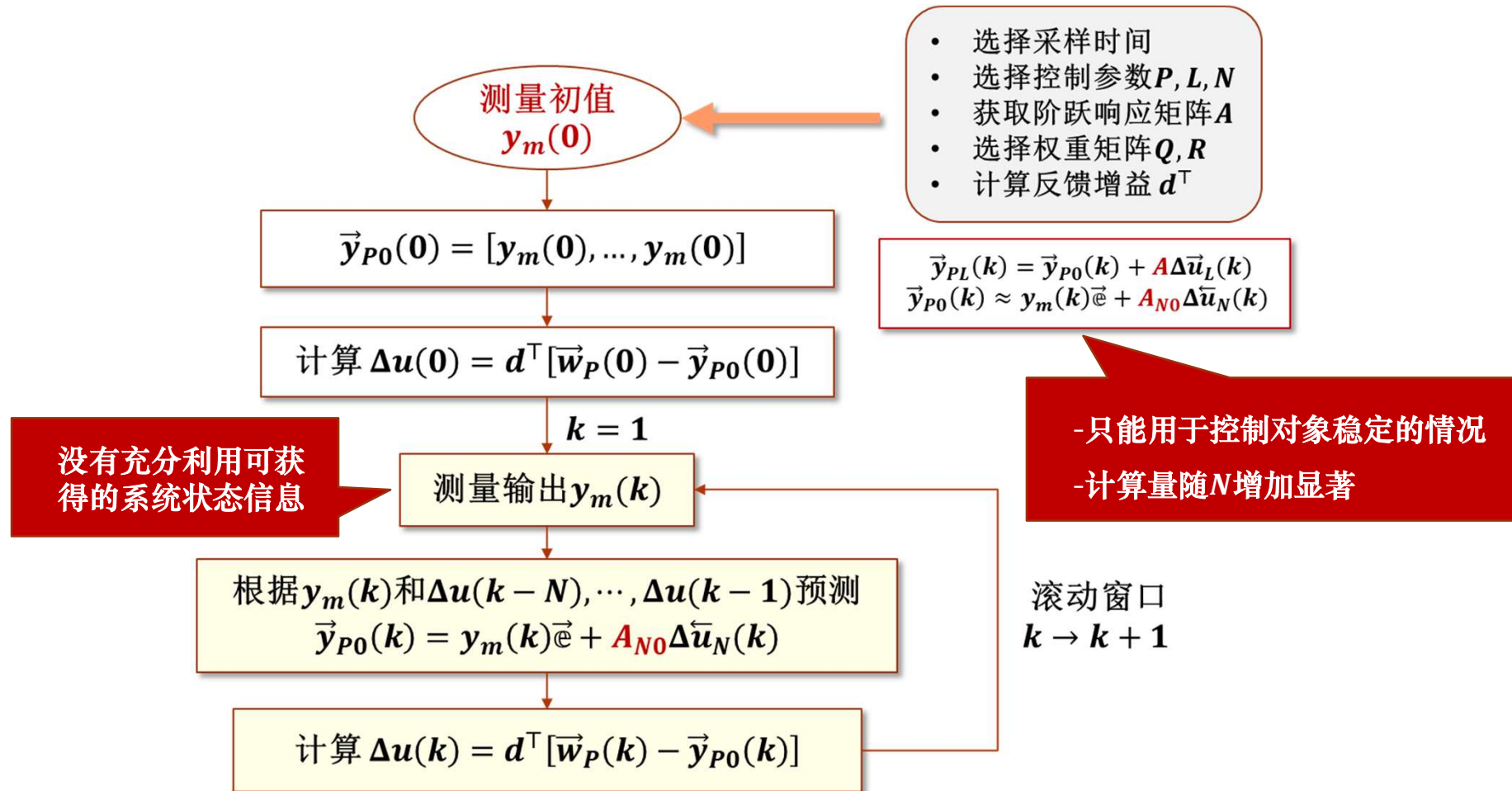
- 选择采样时间
- 选择控制参数 P, L, N
- 获取动态矩阵 A, A_0
- 选择权重矩阵 Q, R
- 计算系数 d^T

- 采样时间：建议在开环系统上升时间内设置10-20个采样
- 采样长度：尽量覆盖整个暂态过程
- 预测窗口(prediction horizon)：20-30个采样
- 控制窗口(control horizon)：预测范围的20%-30%，至少2-3步
- 满足性能要求前提下，参数应尽可能小，以减轻计算负担

基于状态空间模型的方法

动态矩阵法的局限

— automatic control — automatic control — automatic control — automatic control —



基于状态空间模型的预测控制

— automatic control — automatic control — automatic control — automatic control —

- 利用参数模型可以减少计算量和内存占用，且可用于对象开环不稳定的情况
- 充分利用过程的可测信息（输出变量、状态变量及干扰变量），有利于控制系统性能的提高。
- 可以综合预测控制和状态反馈控制的优点，能够更好地跟踪目标和抑制噪声，并方便用于多变量系统。

基于状态空间模型的预测控制

— automatic control — automatic control — automatic control — automatic control —

预测模型：

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{b}u(k), \quad \mathbf{y}(k) = \mathbf{c}^\top \mathbf{x}(k)$$

设计思路：

- 1) 选择预测步长为 P 和控制步长 L
- 2) 利用模型和当前状态测量值 $\hat{\mathbf{x}}(k)$ 预测未来 P 步的输出
- 3) 根据跟踪误差最小原则优化 $u(k), \dots, u(k+L-1)$
- 4) 执行控制动作 $u(k)$
- 5) 滚动执行 $k \rightarrow k+1$

基于状态空间模型的预测

— automatic control — automatic control — automatic control — automatic control —

预测模型：

$$\begin{aligned}\vec{y}_{PL}(k) &= \vec{y}_{P0}(k) + \mathbf{A}\Delta\vec{u}_L(k) \\ \vec{y}_{P0}(k) &\approx y_m(k)\vec{e} + \mathbf{A}_{N0}\Delta\vec{u}_N(k)\end{aligned}$$

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{b}u(k), \quad \mathbf{y}(k) = \mathbf{c}^\top \mathbf{x}(k)$$

假设预测步长为 P ，控制步长 L (L 步以后控制保持不变)

记 $\hat{\mathbf{x}}(k)$ 是 k 时刻状态测量值，则运动轨迹 ($j < L$)：

$$\mathbf{y}(k+j) = \mathbf{c}^\top \mathbf{A}^j \hat{\mathbf{x}}(k) + \sum_{i=1}^j \mathbf{c}^\top \mathbf{A}^{i-1} \mathbf{b}u(k+j-i) \quad \mathbf{【} j < L \mathbf{】}$$

$$\mathbf{y}(k+j) = \mathbf{c}^\top \mathbf{A}^j \hat{\mathbf{x}}(k) + \sum_{i=1}^{L-1} \mathbf{c}^\top \mathbf{A}^{i-1} \mathbf{b}u(k+j-i)$$

k 时刻以前的输出
对预测有影响吗？

$$+ \sum_{i=L}^j \mathbf{c}^\top \mathbf{A}^{i-1} \mathbf{b}u(k+L-1) \quad \mathbf{【} j \geq L \mathbf{】}$$

基于状态空间模型的预测

— automatic control — automatic control — automatic control — automatic control —

$$\text{记 } Y(k) = \begin{bmatrix} y(k+1) \\ \vdots \\ y(k+P) \end{bmatrix}, \quad U(k) = \begin{bmatrix} u(k) \\ \vdots \\ u(k+L-1) \end{bmatrix}$$

则有 $Y(k) = F\hat{x}(k) + GU(k)$, 其中

$$F = \begin{bmatrix} c^T A \\ \vdots \\ c^T A^P \end{bmatrix}, \quad G = \begin{bmatrix} c^T b & & \\ \vdots & \ddots & \\ c^T A^{L-1} b & \dots & c^T b \\ c^T A^L b & & c^T (A + I) b \\ \vdots & \ddots & \vdots \\ c^T A^{P-1} b & \dots & c^T (A^{P-L} + \dots + I) b \end{bmatrix}_{P \times L}$$

控制优化

— automatic control — automatic control — automatic control — automatic control —

若优化目标选取为轨迹跟踪：

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

则易求最优解为

$$U(k) = -(G^T Q G + R)^{-1} G^T Q [F \hat{x}(k) - W(k)]$$

取第一个控制

$$u(k) = f^T W(k) - d^T \hat{x}(k)$$

其中 $d^T = e^T (G^T Q G + R)^{-1} G^T Q F$, **$e^T = [1 \ 0 \ \cdots \ 0]$**

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

反馈校正

— automatic control — automatic control — automatic control — automatic control —

优化控制方案

$$\mathbf{u}(k) = \mathbf{r}(k) - \mathbf{d}_o^\top \hat{\mathbf{x}}(k)$$

包含了基于状态 $\mathbf{x}(k)$ 实时测量的反馈校正。

- 若状态不可测，则需要通过观测器(或滤波器)重构状态：

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

- 如果模型中包含可测扰动，可引入相应前馈控制

基于控制增量的预测与优化

— automatic control — automatic control — automatic control — automatic control —

有时希望限制控制的变化幅度，这对应于其增量约束：

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

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

此时需要重新表述状态空间模型：

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

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

$$\Rightarrow \bar{x}(k + 1) = \bar{A}\bar{x}(k) + \bar{b}\Delta u(k)$$

$$\bar{y}(k) = [c^\top \quad 0]\bar{x}(k) = \bar{c}^\top \bar{x}(k)$$

基于控制增量的预测与优化

— automatic control — automatic control — automatic control — automatic control —

同理可根据估计的扩展状态 $\hat{\mathbf{x}}(k)$ 对状态演化进行预测：

$$\bar{\mathbf{x}}(k+j) = \bar{\mathbf{A}}^j \hat{\mathbf{x}}(k) + \sum_{i=1}^j \bar{\mathbf{A}}^{i-1} \bar{\mathbf{b}} \Delta u(k+j-i)$$

$$\bar{\mathbf{Y}}(k) = \begin{bmatrix} \bar{\mathbf{c}}^\top \bar{\mathbf{A}} \\ \vdots \\ \mathbf{c}^\top \bar{\mathbf{A}}^P \end{bmatrix} \hat{\mathbf{x}}(k) + \begin{bmatrix} \bar{\mathbf{c}}^\top \bar{\mathbf{b}} & & \\ \vdots & \ddots & \\ \bar{\mathbf{c}}^\top \bar{\mathbf{A}}^{M-1} \bar{\mathbf{b}} & \dots & \bar{\mathbf{c}}^\top \bar{\mathbf{b}} \\ \vdots & \ddots & \vdots \\ \bar{\mathbf{c}}^\top \bar{\mathbf{A}}^{P-1} \bar{\mathbf{b}} & \dots & \bar{\mathbf{c}}^\top \bar{\mathbf{A}}^{P-M} \bar{\mathbf{b}} \end{bmatrix} \begin{bmatrix} \Delta u(k) \\ \vdots \\ \Delta u(k+L-1) \end{bmatrix}$$

$\hat{\mathbf{x}}(k) = \begin{bmatrix} \hat{\mathbf{x}}(k) \\ u(k-1) \end{bmatrix}$

$$= \begin{bmatrix} \mathbf{c}^\top \mathbf{A} \\ \vdots \\ \mathbf{c}^\top \mathbf{A}^P \end{bmatrix} \hat{\mathbf{x}}(k) + \begin{bmatrix} \mathbf{c}^\top \mathbf{b} \\ \vdots \\ \sum_{i=0}^{P-1} \mathbf{c}^\top \mathbf{A}^i \mathbf{b} \end{bmatrix} u(k-1) + \begin{bmatrix} \mathbf{c}^\top \mathbf{b} & & \\ \mathbf{c}^\top (\mathbf{A}\mathbf{b} + \mathbf{b}) & \mathbf{c}^\top \mathbf{b} & \\ \vdots & \ddots & \ddots \\ \sum_{i=0}^{P-1} \mathbf{c}^\top \mathbf{A}^i \mathbf{b} & \dots & \dots & \sum_{i=0}^{P-M} \mathbf{c}^\top \mathbf{A}^i \mathbf{b} \end{bmatrix} \Delta \mathbf{U}(k)$$

现在
过去
将来

$$\triangleq \mathbf{F}_o \hat{\mathbf{x}}(k) + \mathbf{B}_o u(k-1) + \mathbf{D}_o \Delta \mathbf{U}(k)$$

基于控制增量的预测与优化

— automatic control — automatic control — automatic control — automatic control —

针对优化目标： $\min_{U(k)} \|Y(k) - W(k)\|_Q + \|\Delta U(k)\|_R$

易求最优解为

$$\Delta U(k) = -(D_o^\top Q D_o + R)^{-1} D_o^\top Q [F_o \hat{x}(k) + B_o u(k-1) - W(k)]$$

取第一个控制

$$\Delta u(k) = f^\top W(k) - p^\top u(k-1) - d^\top \hat{x}(k)$$

其中 $f^\top = e^\top (D_o^\top Q D_o + R)^{-1} D_o^\top Q$,

$$d^\top = e^\top (D_o^\top Q D_o + R)^{-1} D_o^\top Q F_o, \quad p^\top = (D_o^\top Q D_o + R)^{-1} D_o^\top Q B_o$$

闭环系统的稳定性

— automatic control — automatic control — automatic control — automatic control — automatic control —

预测控制系统的闭环稳定性没有简洁的结果。

如果令预测/控制窗口趋近于无穷大，即 $P \rightarrow \infty$, $L \rightarrow \infty$ ，则优化问题变为标准的LQR问题，而LQR当 $(A, Q^{\frac{1}{2}})$ 能观时一定是渐进稳定的。

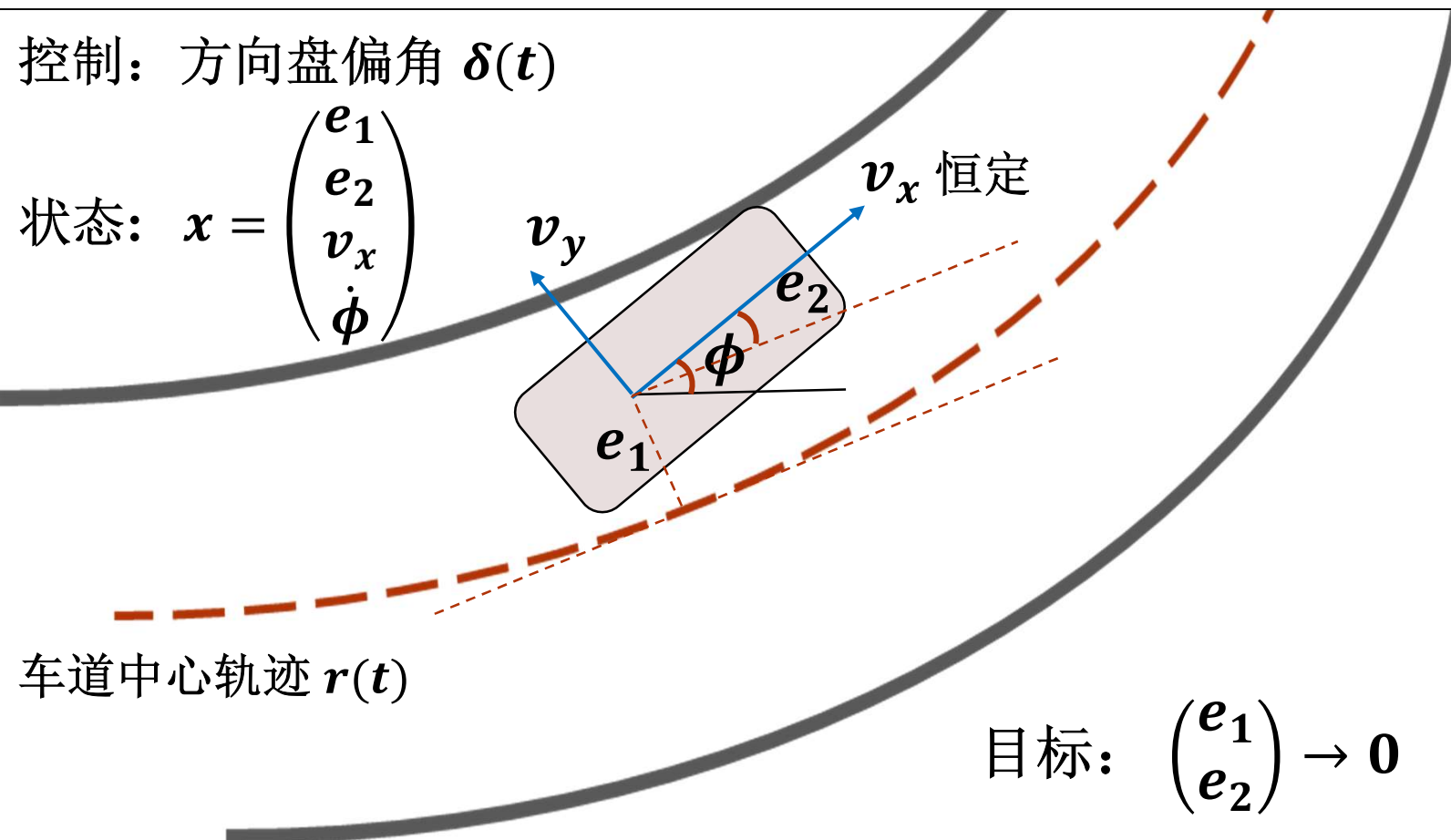
当预测窗口有限大时，闭环系统的渐进稳定性并不显然，而且参数选择不当时系统甚至可能失稳，相关条件分析较为复杂。通常在一定约束条件下可得到充分条件，这方面有很多的研究，在此不再赘述。

应用示例：车道辅助保持控制

— automatic control — automatic control — automatic control — automatic control — automatic control —

控制：方向盘偏角 $\delta(t)$

状态： $x = \begin{pmatrix} e_1 \\ e_2 \\ v_x \\ \dot{\phi} \end{pmatrix}$



车道保持控制模型

— automatic control — automatic control — automatic control — automatic control —

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\delta(t) + \mathbf{E}r(t), \quad \mathbf{y}(t) = \mathbf{C}\mathbf{x}(t)$$

$$\mathbf{A} = \begin{bmatrix} 0 & v_x & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\frac{2C_f+2C_r}{mv_x} & -v_x - \frac{2C_f l_f - 2C_r l_r}{mv_x} \\ 0 & 0 & -\frac{2C_f l_f - 2C_r l_r}{I_z v_x} & -\frac{2C_f l_f^2 + 2C_r l_r^2}{I_z v_x} \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ \frac{2C_f}{m} \\ \frac{2C_f l_f}{I_z} \end{bmatrix}, \quad \mathbf{E} = \begin{bmatrix} 0 \\ -v_x \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

m : 汽车质量 (kg).

I_z : 转动惯量 ($\text{kg} \cdot \text{m}^2$).

l_f : 质心与前轮径向距离 (m).

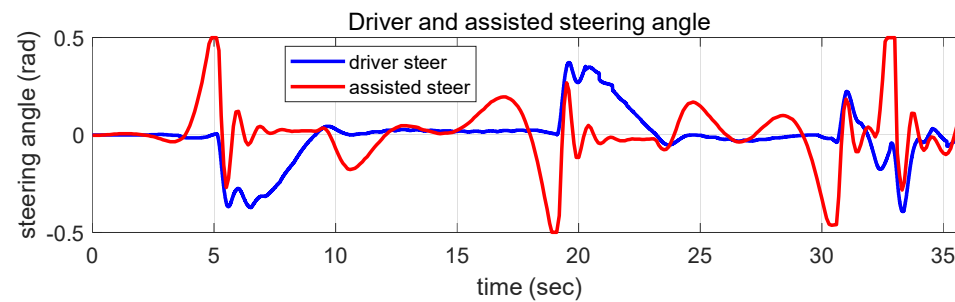
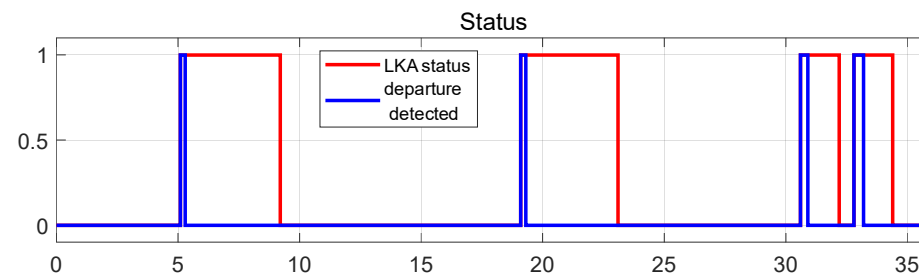
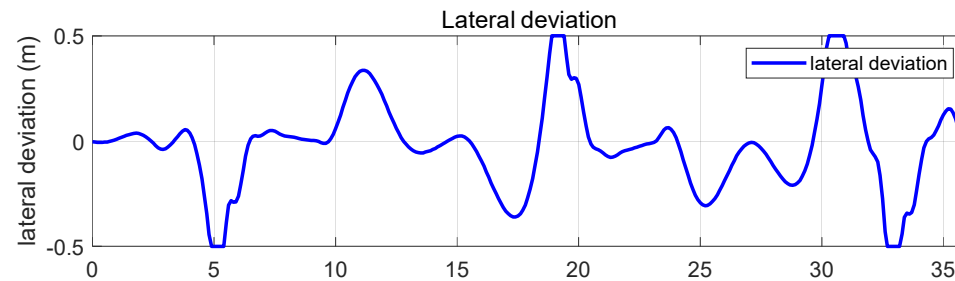
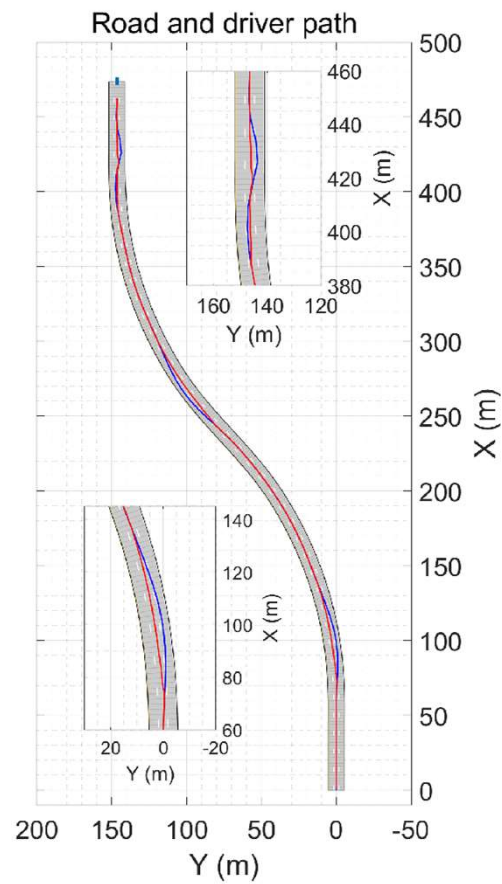
l_r : 质心与后轮径向距离 (m).

C_f : 前轮侧偏刚度 (N/rad).

C_r : 后轮侧偏刚度 (N/rad).

车道保持控制

— automatic control — automatic control — automatic control — automatic control — automatic control —



LKATestBenchExample.mdl

总结

— automatic control — automatic control — automatic control — automatic control — automatic control —

模型预测控制是一个一般的设计框架：

- 反馈、前馈、PID的思想均有所体现。
- 可以根据需要选择不同的预测模型，而且预测模型可在线更新或切换，保证系统的鲁棒性和容错性。
- 可以考虑非线性和约束的影响，由于每步优化都是有限维变量的优化，因此原则可解，但计算量会显著增加，对硬件的实时性要求高。
- 缺乏严格的理论保证，但具有很强的实用性。