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；…

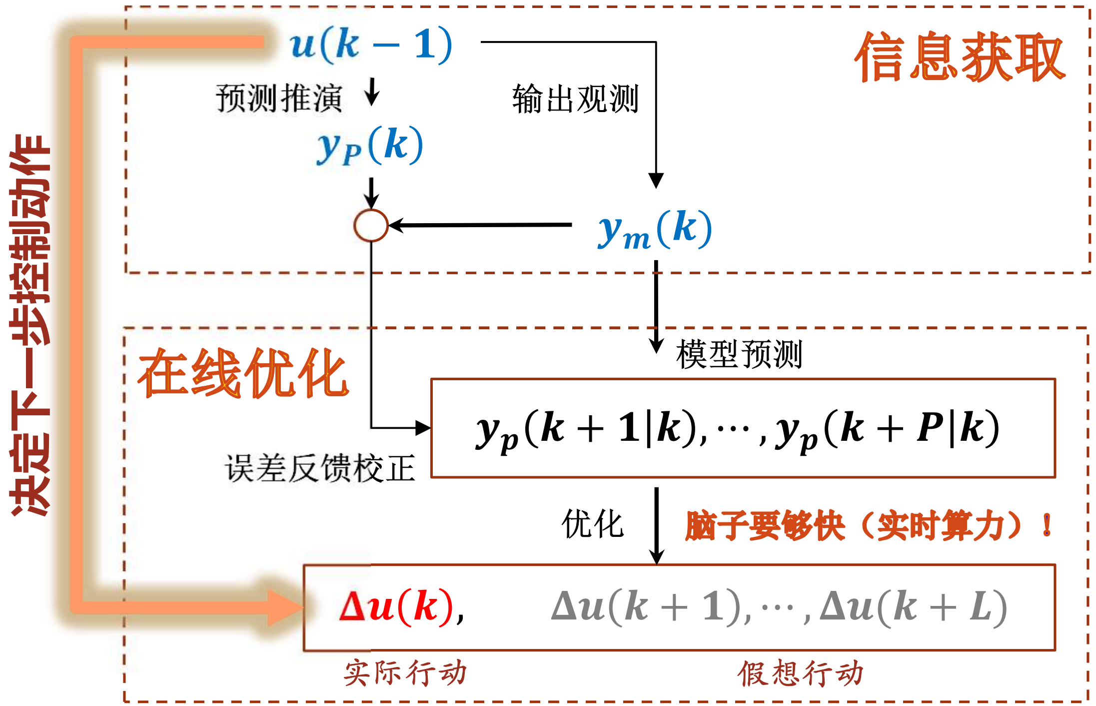
模型预测控制：基于模型短期预测＋结合数据在线优化

应用：化工过程 → 航空、交通、机器人、…

模型预测控制的基本思想

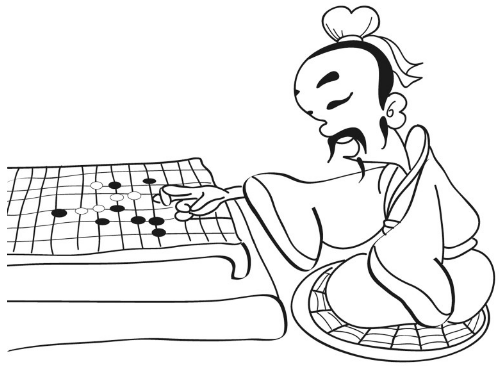
— automatic control — automatic control — automatic control — automatic control —

目标：$y(k+1) = y_R(k+1)$；　策略：$\Delta u(k)$



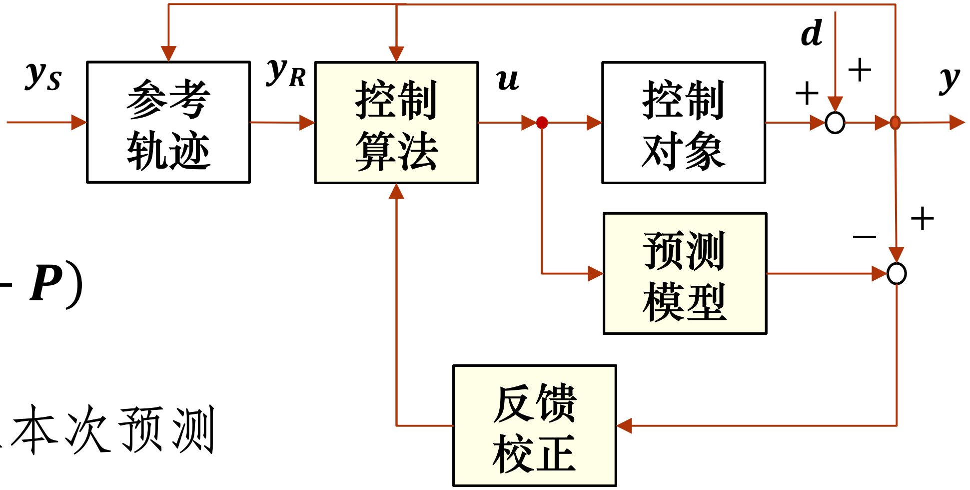
下 棋

- 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

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

- 根据上步预测与实测数据修正本次预测

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

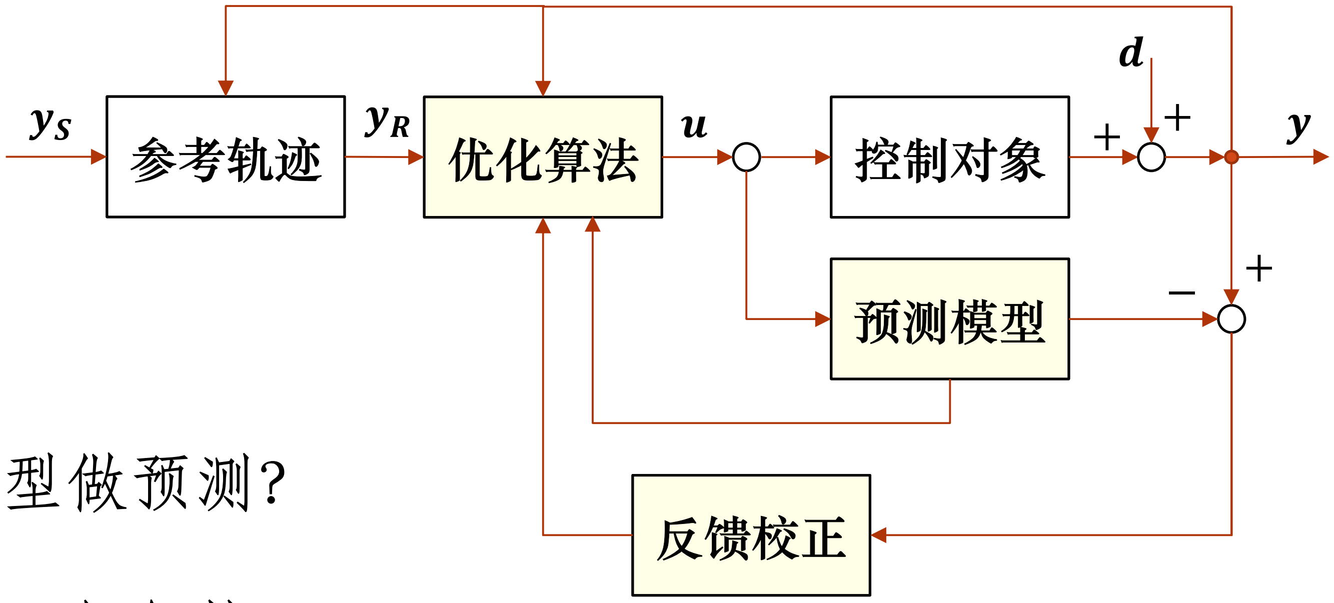
- 根据预设轨迹对控制进行开环优化

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

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

2. 非参数化模型【直接取自数据】

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

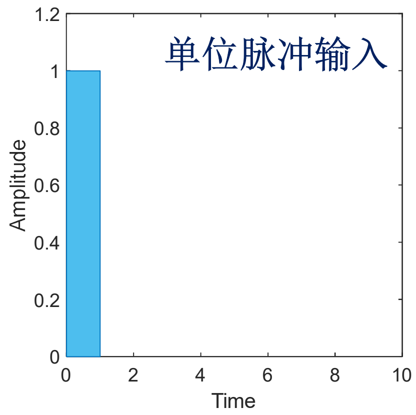
优化与分析

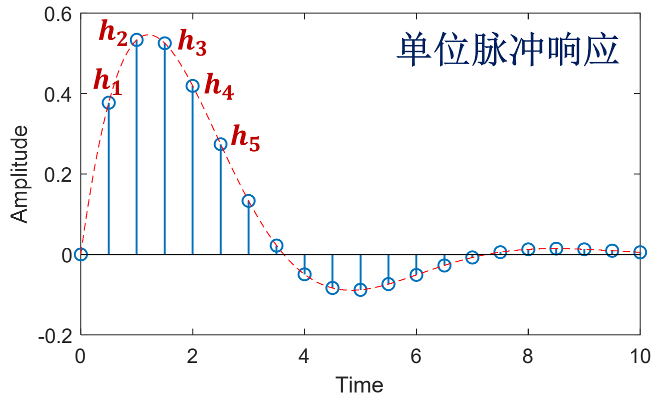
1. 优化
   - 通过最小化跟踪误差得到最优控制序列
   - 只在有限时间窗口内优化，得到开环解
   - 可处理控制受约束情形，对复杂问题实时性要求高
2. 分析
   - 闭环稳定性较难分析
   - 常常要求控制对象是稳定的，或者先设计校正使之稳定。

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$\boldsymbol{\delta(k)}$： $\boldsymbol{\delta(k) = 1}$ 仅当$\boldsymbol{k = 0}$，单位脉冲响应为

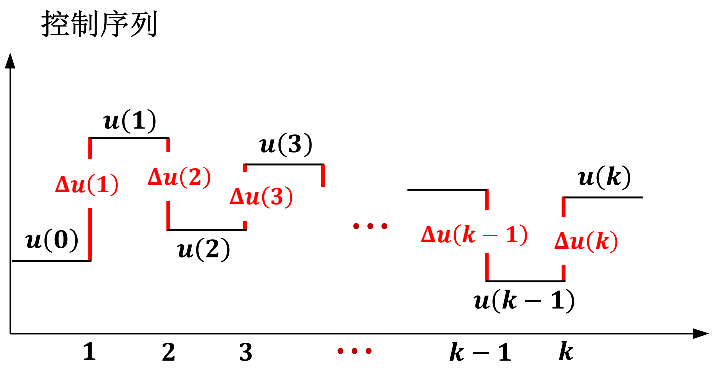
$$\boldsymbol{y(k) = \begin{cases} h_k, & k \geq 0 \\ 0, & k < 0 \end{cases}}$$

- 对于延迟的单位脉冲输入$\boldsymbol{\delta(k - k_0)}$, 其响应为

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

基于单位脉冲响应的卷积模型

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



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

$$\Delta u(k) = u(k) - u(k-1), \qquad k \geq 1$$
$$\Delta u(k) = u(0), \qquad k = 0$$

- 控制序列可展开为：

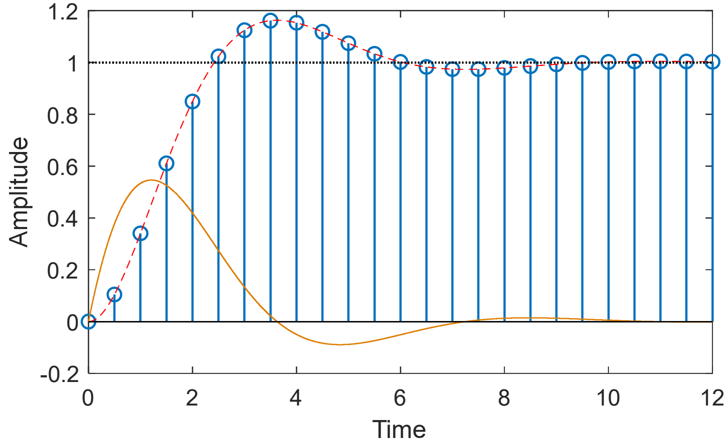
$$u(k) = \sum_{j=0}^{\infty} u(j)\delta(k-j)$$

- 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

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

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

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

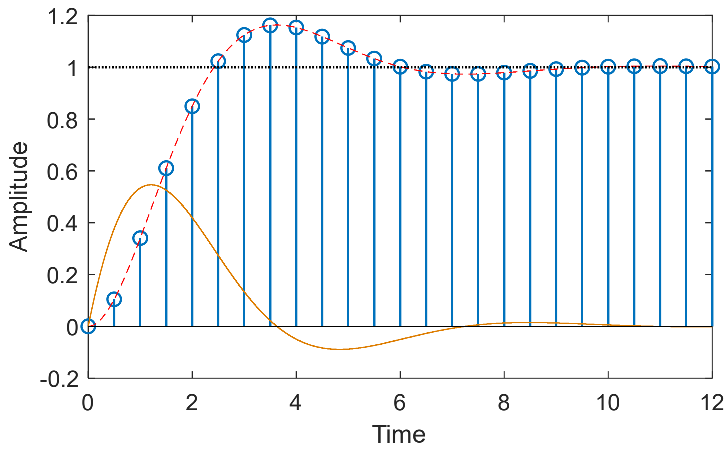
输出可以表示为阶跃响应与控制增量函数的卷积

【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 —

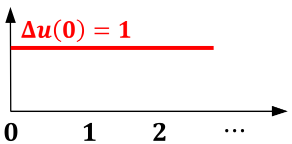


阶跃响应是脉冲响应的“积分”

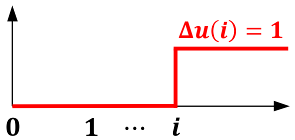
$$s_k = \sum_{j=0}^{k} h_j ,$$

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

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



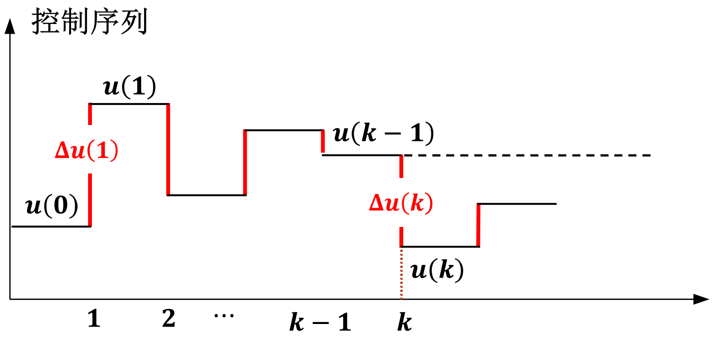
- 单位阶跃输入 $\Delta u(0) = 1$，$y(k) = s_k$

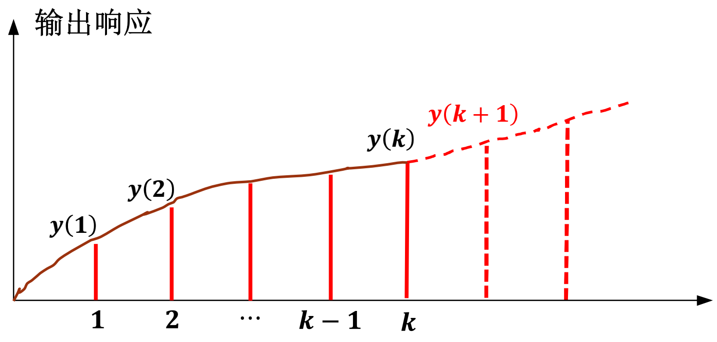


- 单位阶跃输入 $\Delta u(i) = 1$，$y(k) = s_{k-i}$

基于单位脉冲响应的预测

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





考察$\boldsymbol{k}$时刻起输出与控制输入的关系:

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

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

（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$\boldsymbol{k}$时刻测量数据$\boldsymbol{y_m(k)}$估计偏差并反馈校正

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

其中反馈系数需根据经验选取. $\boldsymbol{\alpha_i \equiv 1}$意味着误差不变。

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 —

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

其中$y_0(k+i|k)$代表如果控制保持不变(即 $\Delta \mathrm{u}(\mathrm{k}+\mathrm{i})\equiv 0$)时

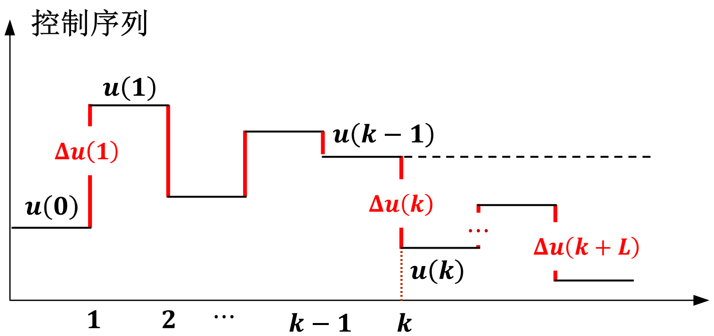
经过反馈校正后的预测：

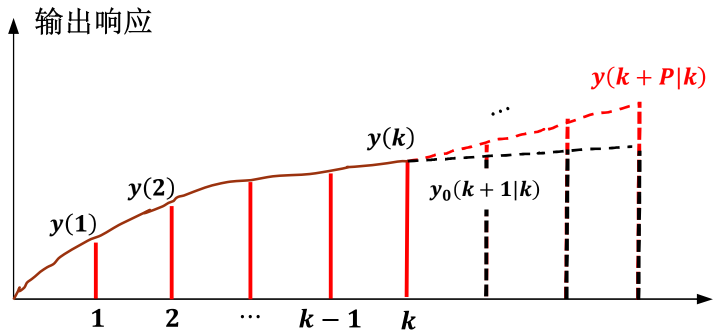
$$y_0(k+i|k)=\sum_{j=i+1}^{k+i}s_j\Delta u(k+i-j)+\alpha_i\left[y_m(k)-\sum_{j=1}^{k}s_j\Delta u(k-j)\right]$$

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

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 $y(k+1|k),\cdots,y(k+P|k)$:

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

动态矩阵
（Dynamical Matrix）

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

基于L步控制对P步输出的预测

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！

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

$$\boxed{\text{简写：}\ \vec{y}_{P0}(k)=y_m(k)\vec{\alpha}+A_0\Delta\overleftarrow{u}(k)}$$

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 $s_{k+i} \approx s_k$。

此时若取$\alpha_i \equiv 1$, 则可做如下截断：

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

$$\approx y_m(k) + \sum_{j=1}^{N}\left(s_{j+i} - s_j\right)\Delta u(k-j)$$

$$\begin{bmatrix} y_0(k+1|k) \\ y_0(k+2|k) \\ \vdots \\ y_0(k+P-1|k) \\ y_0(k+P|k) \end{bmatrix}_{P\times 1} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \\ 1 \end{bmatrix} y_m(k) + \begin{bmatrix} s_{N+1} - s_N & \cdots & s_2 - s_1 \\ \vdots & \ddots & \vdots \\ s_{N+P} - s_N & \cdots & s_{1+P} - s_1 \end{bmatrix}_{P\times N} \begin{bmatrix} \Delta u(k-N) \\ \vdots \\ \Delta u(k-1) \end{bmatrix}_{N\times 1}$$

历史控制的影响最多回溯到前N个

$$\boxed{\text{简写：}\ \vec{y}_{P0}(k) \approx y_m(k)\vec{e} + A_{N0}\Delta\overleftarrow{u}_N(k)}$$

滚动优化

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

为跟踪某预设轨迹$\boldsymbol{w}(\boldsymbol{k})$，希望选择最佳的控制动作

$$\boldsymbol{\Delta}\vec{\boldsymbol{u}}_{\boldsymbol{L}} = [\boldsymbol{\Delta u}(\boldsymbol{k}), \cdots, \boldsymbol{\Delta u}(\boldsymbol{k}+\boldsymbol{L})]^{\top}$$

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

$$\min_{\boldsymbol{\Delta}\vec{\boldsymbol{u}}_{\boldsymbol{L}}} \|\vec{\boldsymbol{y}}_{\boldsymbol{PL}}(\boldsymbol{k}) - \vec{\boldsymbol{w}}_{\boldsymbol{P}}(\boldsymbol{k})\|_{\boldsymbol{Q}}^{2} + \|\boldsymbol{\Delta}\vec{\boldsymbol{u}}_{\boldsymbol{L}}\|_{\boldsymbol{R}}^{2}$$

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

根据预测公式可知$\vec{\boldsymbol{y}}_{\boldsymbol{PM}}(\boldsymbol{k}) = \vec{\boldsymbol{y}}_{\boldsymbol{P0}}(\boldsymbol{k}) + \boldsymbol{A}\boldsymbol{\Delta}\vec{\boldsymbol{u}}_{\boldsymbol{L}}(\boldsymbol{k})$，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 $\dfrac{\partial J}{\partial \Delta\vec{u}_L(k)} = 0$ 获得：

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

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

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 —

- 选择采样时间
- 选择控制参数 P, L, N
- 获取阶跃响应矩阵 A
- 选择权重矩阵 Q, R
- 计算反馈增益 $d^\top$

$$\vec{y}_{PL}(k) = \vec{y}_{P0}(k) + A\Delta\vec{u}_L(k)$$
$$\vec{y}_{P0}(k) \approx y_m(k)\vec{\mathbb{e}} + A_{N0}\Delta\overleftarrow{u}_N(k)$$

测量初值 $y_m(0)$

$$\vec{y}_{P0}(0) = [y_m(0), \dots, y_m(0)]$$

计算 $\Delta u(0) = d^\top[\vec{w}_P(0) - \vec{y}_{P0}(0)]$

$k = 1$

测量输出 $y_m(k)$

根据 $y_m(k)$ 和 $\Delta u(k-N), \cdots, \Delta u(k-1)$ 预测

$$\vec{y}_{P0}(k) = y_m(k)\vec{\mathbb{e}} + A_{N0}\Delta\overleftarrow{u}_N(k)$$

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

滚动窗口 $k \to k+1$

滚动优化

- 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。

$$\underbrace{\Delta \boldsymbol{u}}_{I}(\boldsymbol{k}) = \underbrace{\boldsymbol{d}^{\top}}_{P}[\vec{\boldsymbol{w}}_{\boldsymbol{P}}(\boldsymbol{k}) - \underbrace{\vec{\boldsymbol{y}}_{\boldsymbol{P0}}(\boldsymbol{k})}_{D}]$$

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$\boldsymbol{w}(\boldsymbol{k}) \equiv \mathbf{10}$。

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

已知对象实际输出

$$\boldsymbol{y_m}(\mathbf{0}) = \mathbf{9.0}, \boldsymbol{y_m}(\mathbf{1}) = \mathbf{9.5}, \boldsymbol{y_m}(\mathbf{3}) = \mathbf{10}$$

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

示例1

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

$$\boldsymbol{N=6, P=3, L=2}$$

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

$$\Rightarrow \boldsymbol{s}=(0.15, 0.4, 0.6, 0.78, 0.93, 1.01)$$

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

$$\begin{aligned} \boldsymbol{d}^{\top} &= \boldsymbol{c}^{\top}(\boldsymbol{A}^{\top}\boldsymbol{Q}\boldsymbol{A}+\boldsymbol{R})^{-1}\boldsymbol{A}^{\top}\boldsymbol{Q} \\ &= [3.04 \quad 3.11 \quad -1.17] \end{aligned}$$

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

示例1

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

第一步：

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

计算：$\boldsymbol{\Delta u}(\mathbf{0}) = \boldsymbol{d}^{\top}[\vec{\boldsymbol{w}}_{3}(\mathbf{0}) - \vec{\boldsymbol{y}}_{30}(\mathbf{0})]$

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

示例1

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

第二步【$\Delta u(0) = 4.983$】：

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

第二步控制

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

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

$$= -4.606$$

示例1

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

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

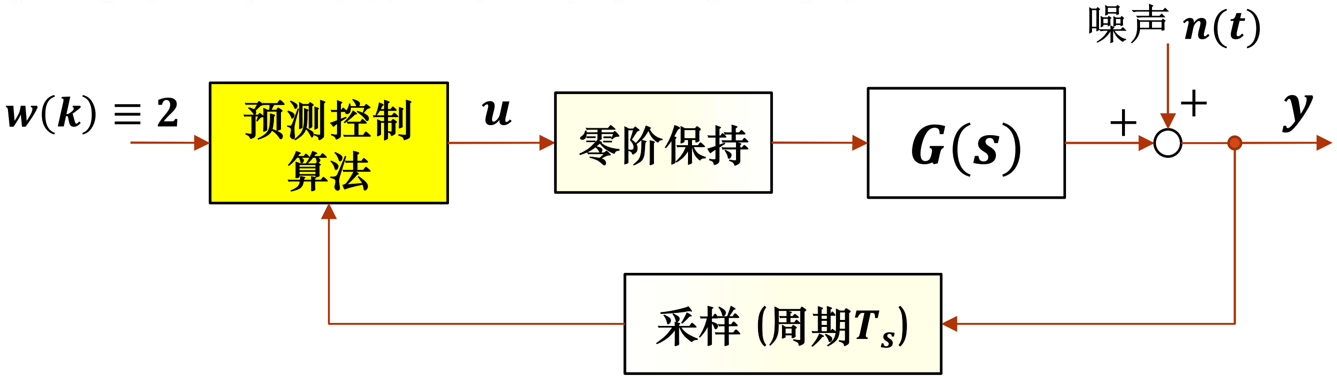
$$\vec{y}_{30}(2)=\vec{e}y_m(2)+A_{60}\begin{bmatrix}0\\ \vdots\\ \Delta u(0)\\ \Delta u(1)\end{bmatrix}=\begin{bmatrix}9.845\\ 9.821\\ 9.739\end{bmatrix}$$

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

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

示例2

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

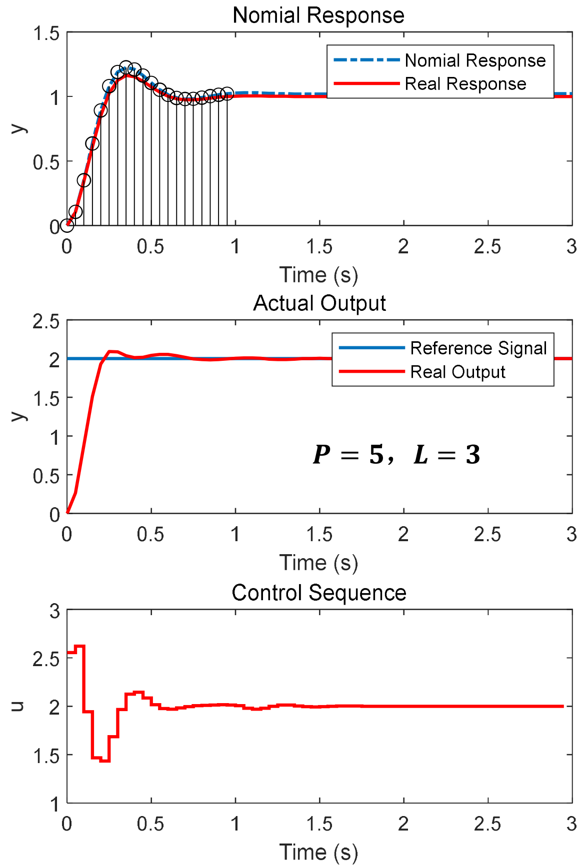


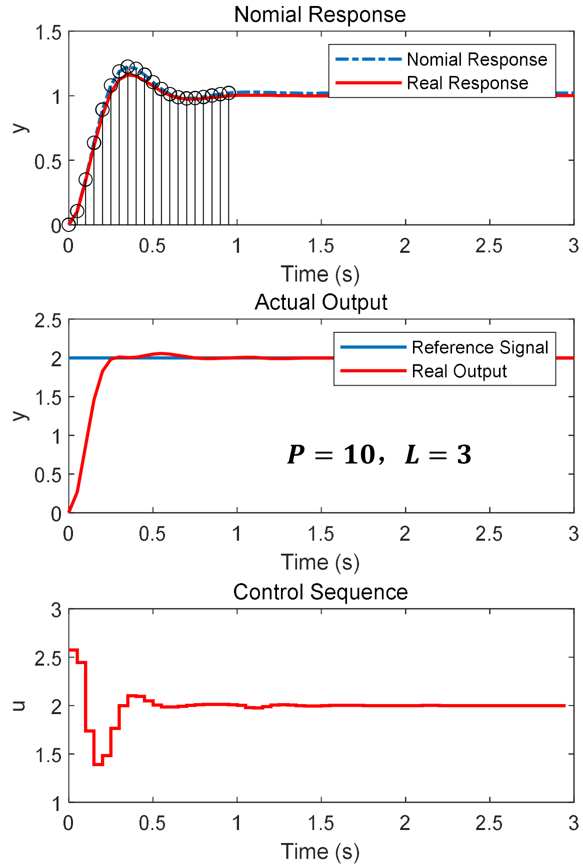
控制对象的标称传递函数为 $\overline{G}(s) = \frac{100}{s^2+10s+100}$. $Q = \mathbb{I}_P$, $R = r\mathbb{I}_L$。

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 —

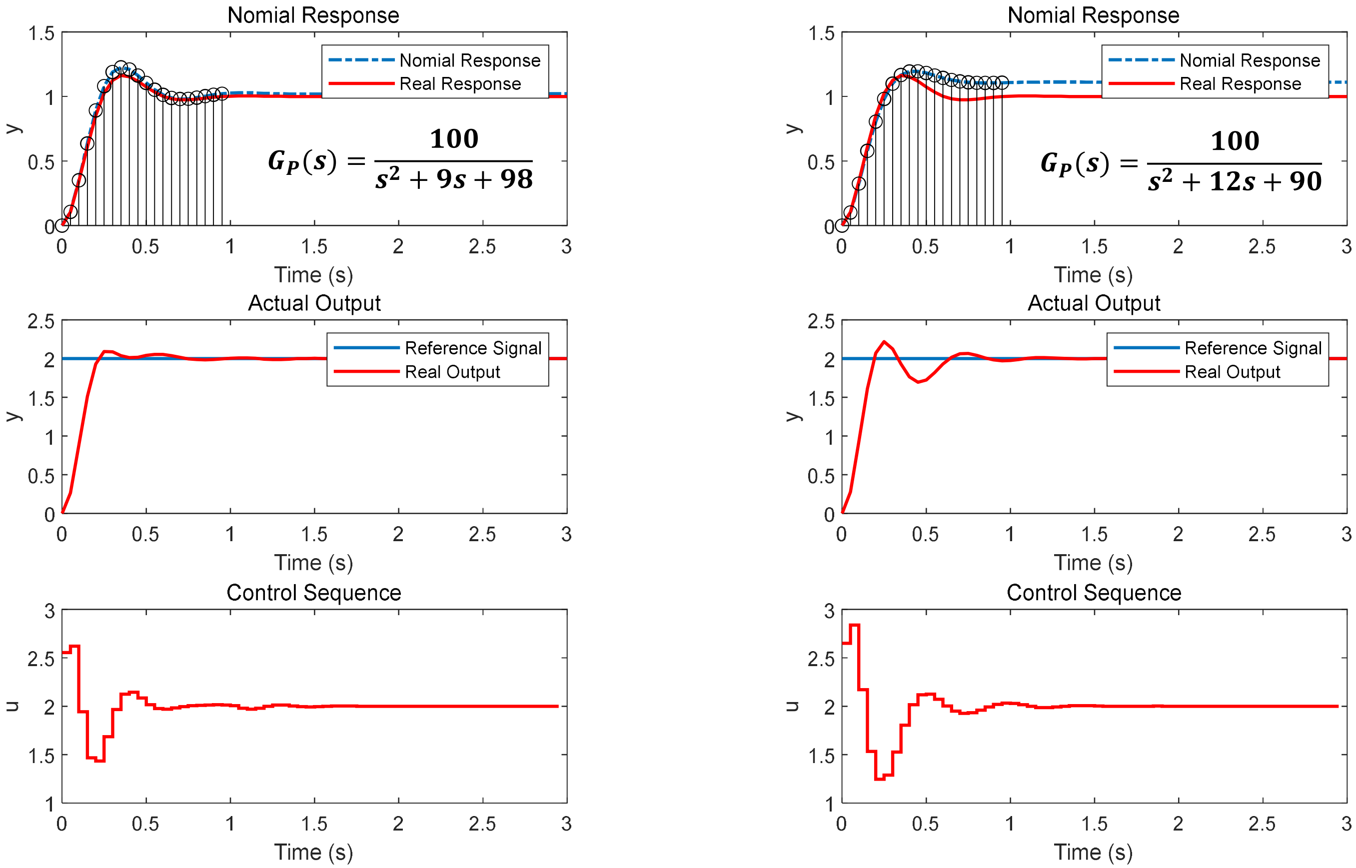




$$G_P(s) = \frac{100}{s^2+9s+98},\ T_s = 0.05,\quad N = 20$$

示例2：模型失配的影响

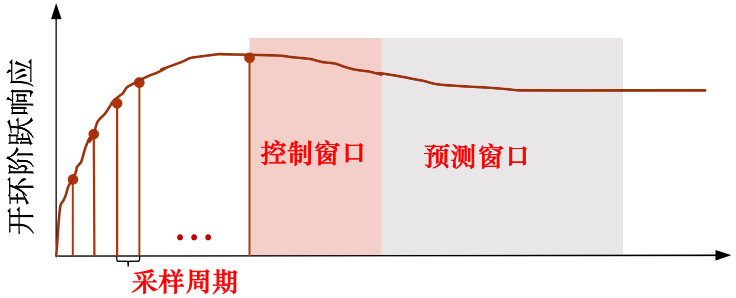
— automatic control — automatic control — automatic control — automatic control —



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

参数选择

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



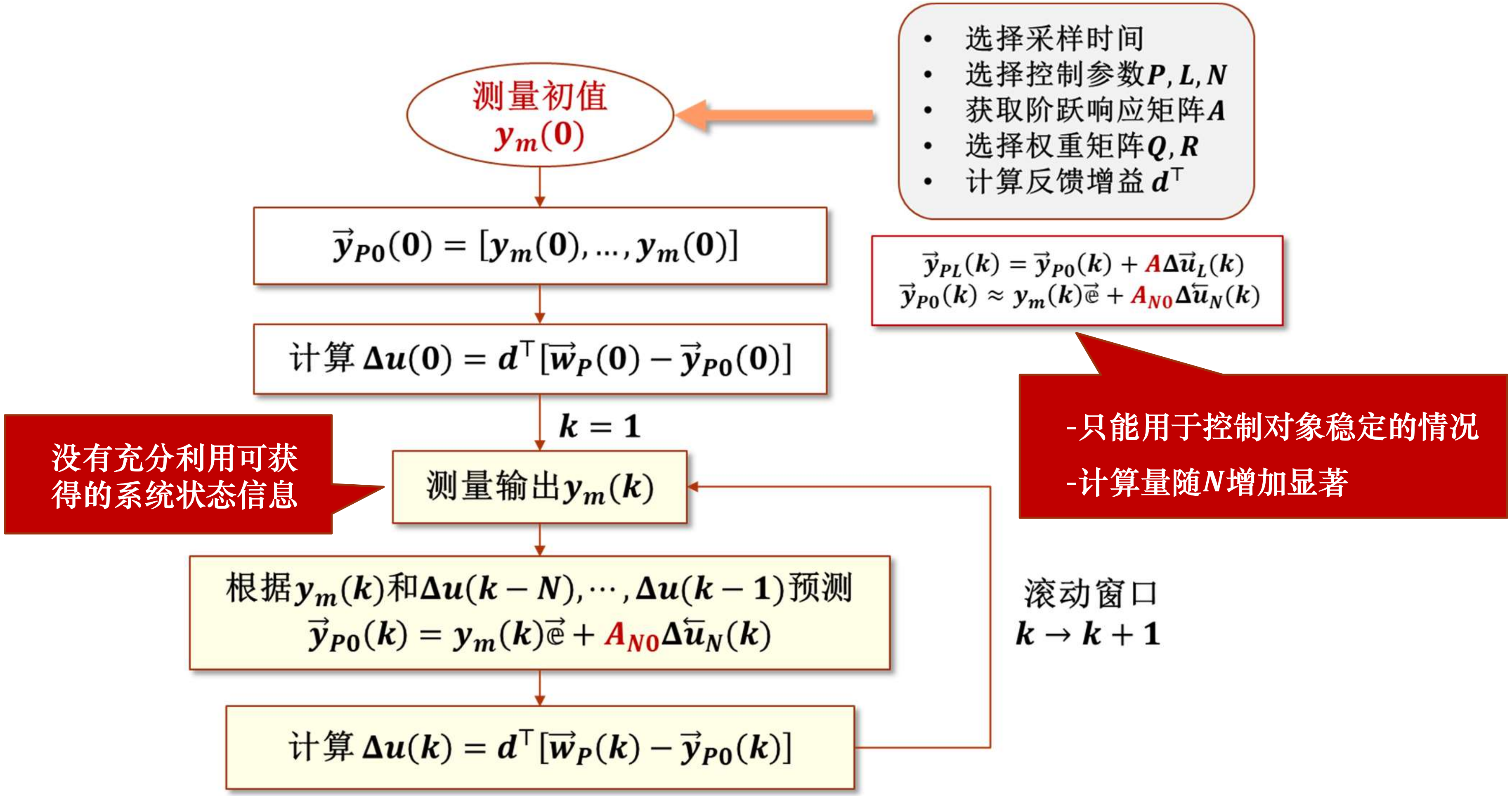
- 选择采样时间
- 选择控制参数$\boldsymbol{P}, \boldsymbol{L}, \boldsymbol{N}$
- 获取动态矩阵$\boldsymbol{A}, \boldsymbol{A}_0$
- 选择权重矩阵$\boldsymbol{Q}, \boldsymbol{R}$
- 计算系数$\boldsymbol{d}^{\top}$

- 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：

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

设计思路：

1）选择预测步长为 $\boldsymbol{P}$ 和控制步长 $\boldsymbol{L}$

2）利用模型和当前状态测量值$\hat{\boldsymbol{x}}(\boldsymbol{k})$预测未来$\boldsymbol{P}$步的输出

3）根据跟踪误差最小原则优化 $\boldsymbol{u}(\boldsymbol{k}),\cdots,\boldsymbol{u}(\boldsymbol{k}+\boldsymbol{L}-\mathbf{1})$

4）执行控制动作 $\boldsymbol{u}(\boldsymbol{k})$

5）滚动执行$\boldsymbol{k}\rightarrow\boldsymbol{k}+\mathbf{1}$

基于状态空间模型的预测

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

$$\vec{y}_{PL}(k) = \vec{y}_{P0}(k) + A\Delta\vec{u}_L(k)$$
$$\vec{y}_{P0}(k) \approx y_m(k)\vec{e} + A_{N0}\Delta\overleftarrow{u}_N(k)$$

预测模型：

$$x(k+1) = Ax(k) + bu(k), \quad y(k) = c^\top x(k)$$

假设预测步长为 P，控制步长 L（L步以后控制保持不变）

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

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

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

$$+ \sum_{i=L}^{j} c^\top A^{i-1} bu(k+L-1) \quad 【j \ge L】$$

k时刻以前的输出对预测有影响吗？

基于状态空间模型的预测

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

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

则有 $\boldsymbol{Y}(k)=\boldsymbol{F}\hat{\boldsymbol{x}}(k)+\boldsymbol{G}\boldsymbol{U}(k)$, 其中

$$\boldsymbol{F}=\begin{bmatrix} \boldsymbol{c}^\top \boldsymbol{A} \\ \vdots \\ \boldsymbol{c}^\top \boldsymbol{A}^P \end{bmatrix}, \qquad \boldsymbol{G}=\begin{bmatrix} \boldsymbol{c}^\top \boldsymbol{b} & & \\ \vdots & \ddots & \\ \boldsymbol{c}^\top \boldsymbol{A}^{L-1}\boldsymbol{b} & \cdots & \boldsymbol{c}^\top \boldsymbol{b} \\ \boldsymbol{c}^\top \boldsymbol{A}^{L}\boldsymbol{b} & & \boldsymbol{c}^\top (\boldsymbol{A}+\boldsymbol{I})\boldsymbol{b} \\ \vdots & \ddots & \vdots \\ \boldsymbol{c}^\top \boldsymbol{A}^{P-1}\boldsymbol{b} & \cdots & \boldsymbol{c}^\top \left(\boldsymbol{A}^{P-L}+\cdots+\boldsymbol{I}\right)\boldsymbol{b} \end{bmatrix}_{P\times L}$$

控制优化

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

若优化目标选取为轨迹跟踪:

$$\min_{\boldsymbol{U}(\boldsymbol{k})} \|\boldsymbol{Y}(\boldsymbol{k}) - \boldsymbol{W}(\boldsymbol{k})\|_{\boldsymbol{Q}} + \|\boldsymbol{U}(\boldsymbol{k})\|_{\boldsymbol{R}}$$

则易求最优解为

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

取第一个控制

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

其中 $\boldsymbol{d}^\top = \boldsymbol{e}^\top (\boldsymbol{G}^\top \boldsymbol{Q} \boldsymbol{G} + \boldsymbol{R})^{-1} \boldsymbol{G}^\top \boldsymbol{Q} \boldsymbol{F}$, 【$\boldsymbol{e}^\top = [\mathbf{1}\ \mathbf{0} \cdots \mathbf{0}]$】

$$\boldsymbol{r}(\boldsymbol{k}) = \boldsymbol{e}^\top (\boldsymbol{G}^\top \boldsymbol{Q} \boldsymbol{G} + \boldsymbol{R})^{-1} \boldsymbol{G}^\top \boldsymbol{Q} \boldsymbol{W}(\boldsymbol{k}).$$

反馈校正

由于优化控制

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

包含了对状态 $\boldsymbol{x}(\boldsymbol{k})$ 的实时测量，因此该方案自动包含了反馈校正，不需额外的校正措施.

若状态不可测，则需要通过观测器(或滤波器)重构状态:

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

基于控制增量的预测与优化

有时希望限制控制的变化幅度，这对应于其增量约束：

$$\min_{\boldsymbol{U}(\boldsymbol{k})} \|\boldsymbol{Y}(\boldsymbol{k}) - \boldsymbol{W}(\boldsymbol{k})\|_{\boldsymbol{Q}} + \|\boldsymbol{\Delta U}(\boldsymbol{k})\|_{\boldsymbol{R}}$$

其中 $\boldsymbol{\Delta U}(\boldsymbol{k}) = [\boldsymbol{\Delta u}(\boldsymbol{k}), \cdots, \boldsymbol{\Delta u}(\boldsymbol{k} + \boldsymbol{M} - \mathbf{1})]$.

此时需要重新表述状态空间模型：

$$\boldsymbol{x}(\boldsymbol{k}+\mathbf{1}) = \boldsymbol{A}\boldsymbol{x}(\boldsymbol{k}) + \boldsymbol{b}[\boldsymbol{u}(\boldsymbol{k}-\mathbf{1}) + \boldsymbol{\Delta u}(\boldsymbol{k})]$$

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

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

$$\overline{\boldsymbol{y}}(\boldsymbol{k}) = [\boldsymbol{c}^\top \ \ \mathbf{0}]\overline{\boldsymbol{x}}(\boldsymbol{k}) = \overline{\boldsymbol{c}}^\top \overline{\boldsymbol{x}}(\boldsymbol{k})$$

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$\widehat{\bar{x}}(k)$对状态演化进行预测：

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

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

$$\boxed{\widehat{\bar{x}}(k)=\begin{bmatrix}\widehat{x}(k)\\ u(k-1)\end{bmatrix}}$$

$$=\underset{\text{现在}}{\begin{bmatrix}c^\top A\\ \vdots\\ c^\top A^P\end{bmatrix}\widehat{x}(k)}+\underset{\text{过去}}{\begin{bmatrix}c^\top b\\ \vdots\\ \sum_{i=0}^{P-1}c^\top A^i b\end{bmatrix}u(k-1)}+\underset{\text{将来}}{\begin{bmatrix}c^\top b & & & \\ c^\top(Ab+b) & c^\top b & & \\ \vdots & \ddots & \ddots & \\ \sum_{i=0}^{P-1}c^\top A^i b & \cdots & \cdots & \sum_{i=0}^{P-M}c^\top A^i b\end{bmatrix}\Delta U(k)}$$

$$\triangleq F_o\widehat{x}(k)+B_o u(k-1)+D_o\Delta U(k)$$

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：$\min\limits_{\boldsymbol{U}(\boldsymbol{k})} \|\boldsymbol{Y}(\boldsymbol{k}) - \boldsymbol{W}(\boldsymbol{k})\|_{\boldsymbol{Q}} + \|\boldsymbol{\Delta U}(\boldsymbol{k})\|_{\boldsymbol{R}}$

易求最优解为

$$\boldsymbol{\Delta U}(\boldsymbol{k}) = -(\boldsymbol{D}_{\boldsymbol{o}}^{\top}\boldsymbol{Q}\boldsymbol{D}_{\boldsymbol{o}} + \boldsymbol{R})^{-1}\boldsymbol{D}_{\boldsymbol{o}}^{\top}\boldsymbol{Q}[\boldsymbol{F}_{\boldsymbol{o}}\widehat{\boldsymbol{x}}(\boldsymbol{k}) + \boldsymbol{B}_{\boldsymbol{o}}\boldsymbol{u}(\boldsymbol{k}-\mathbf{1}) - \boldsymbol{W}(\boldsymbol{k})]$$

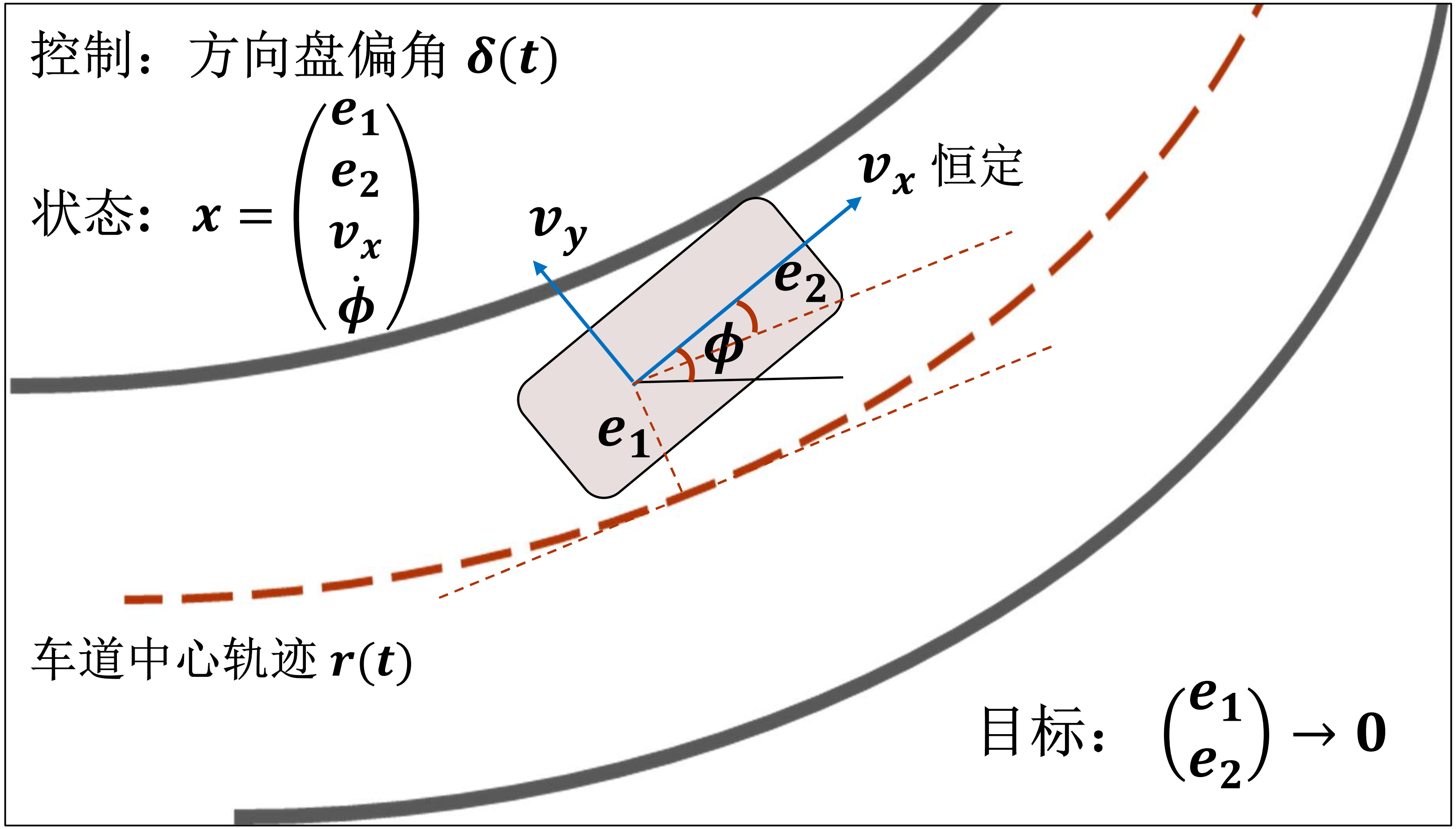
取第一个控制

$$\boldsymbol{\Delta u}(\boldsymbol{k}) = \boldsymbol{r}(\boldsymbol{k}) - \boldsymbol{p}_{\boldsymbol{o}}^{\top}\boldsymbol{u}(\boldsymbol{k}-\mathbf{1}) - \boldsymbol{d}_{\boldsymbol{o}}^{\top}\widehat{\boldsymbol{x}}(\boldsymbol{k})$$

其中 $\boldsymbol{r}(\boldsymbol{k}) = \boldsymbol{e}^{\top}(\boldsymbol{D}_{\boldsymbol{o}}^{\top}\boldsymbol{Q}\boldsymbol{D}_{\boldsymbol{o}} + \boldsymbol{R})^{-1}\boldsymbol{D}_{\boldsymbol{o}}^{\top}\boldsymbol{Q}\boldsymbol{W}(\boldsymbol{k})$,

$\boldsymbol{d}_{\boldsymbol{o}}^{\top} = \boldsymbol{e}^{\top}(\boldsymbol{D}_{\boldsymbol{o}}^{\top}\boldsymbol{Q}\boldsymbol{D}_{\boldsymbol{o}} + \boldsymbol{R})^{-1}\boldsymbol{D}_{\boldsymbol{o}}^{\top}\boldsymbol{Q}\boldsymbol{F}_{\boldsymbol{o}}$, $\boldsymbol{p}_{\boldsymbol{o}}^{\top} = (\boldsymbol{D}_{\boldsymbol{o}}^{\top}\boldsymbol{Q}\boldsymbol{D}_{\boldsymbol{o}} + \boldsymbol{R})^{-1}\boldsymbol{D}_{\boldsymbol{o}}^{\top}\boldsymbol{Q}\boldsymbol{B}_{\boldsymbol{o}}$

应用示例：车道辅助保持控制


控制：方向盘偏角 $\boldsymbol{\delta(t)}$
状态：$\boldsymbol{x} = \begin{pmatrix} \boldsymbol{e_1} \\ \boldsymbol{e_2} \\ \boldsymbol{v_x} \\ \boldsymbol{\dot{\phi}} \end{pmatrix}$
$\boldsymbol{v_y}$
$\boldsymbol{v_x}$ 恒定
$\boldsymbol{e_2}$
$\boldsymbol{\phi}$
$\boldsymbol{e_1}$
车道中心轨迹 $\boldsymbol{r(t)}$
目标：$\begin{pmatrix} \boldsymbol{e_1} \\ \boldsymbol{e_2} \end{pmatrix} \rightarrow \boldsymbol{0}$

车道保持控制模型

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

$$\dot{x}(t) = Ax(t) + B\delta(t) + Er(t), \qquad y(t) = Cx(t)$$

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

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

m: 汽车质量 (kg).

I_z: 转动惯量 ($\mathrm{kg} \cdot m^2$).

l_f: 质心与前轮径向距离 (m).

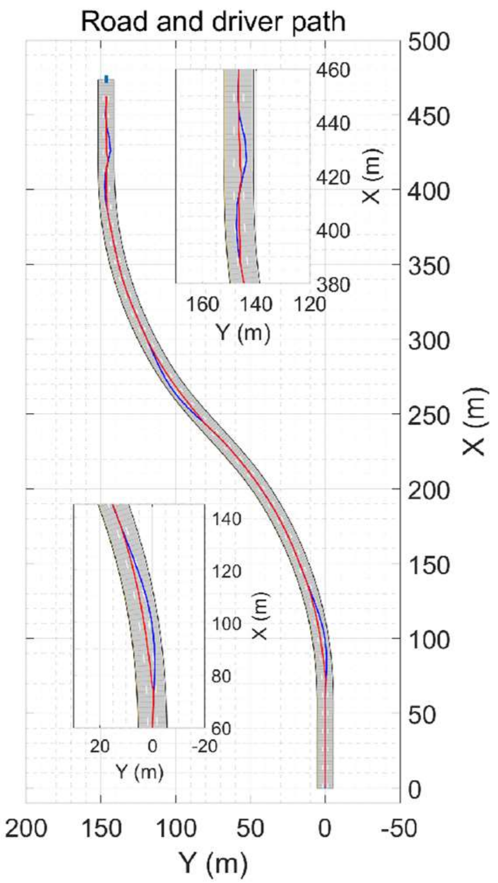
l_r: 质心与后轮径向距离 (m).

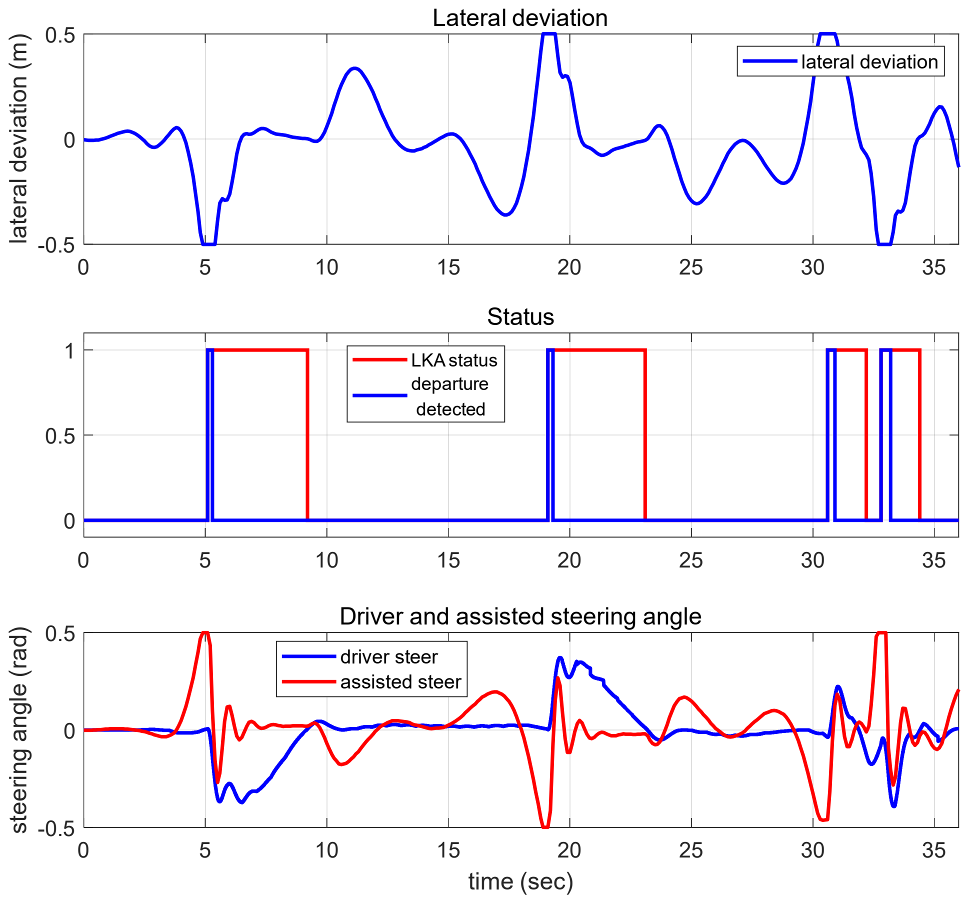
C_f: 前轮侧偏刚度 (N/rad).

C_r: 后轮侧偏刚度 (N/rad).

车道保持控制

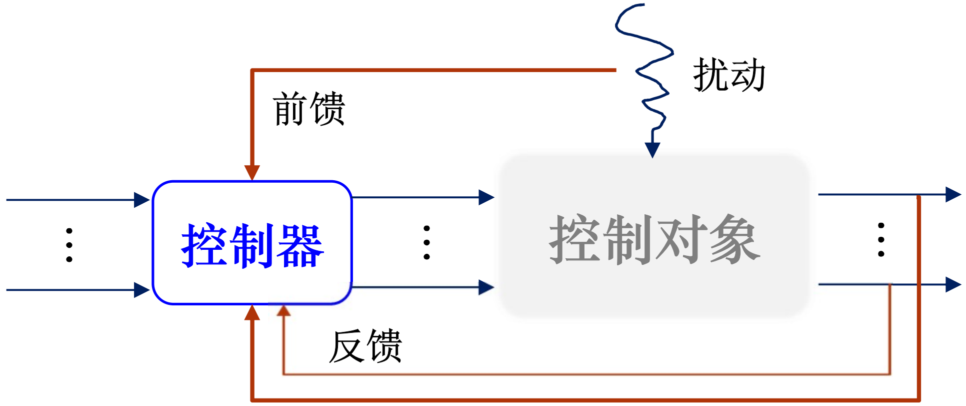
— automatic control — automatic control — automatic control — automatic control — automatic control —





LKATestBenchExample.mdl

课程总结



- 解耦控制：化繁为简
- 抗扰控制：知己知彼（反馈+前馈）
- 最优控制：目标与困境
- 预测控制：前瞻与决策