

自动控制实践B——2022年春季学期

第五章 Anti-Windup设计 (1)

授课教师：董广忠 (Assoc. Prof.)

哈尔滨工业大学（深圳），HITsz
机电工程与自动化学院 SMEA





课程 安排

1

执行器约束问题

2

Anti-Windup设计





5.1 执行器的约束问题



5.1.1

执行器约束问题的提出

5.1.2

执行器约束的描述方法

5.1.3

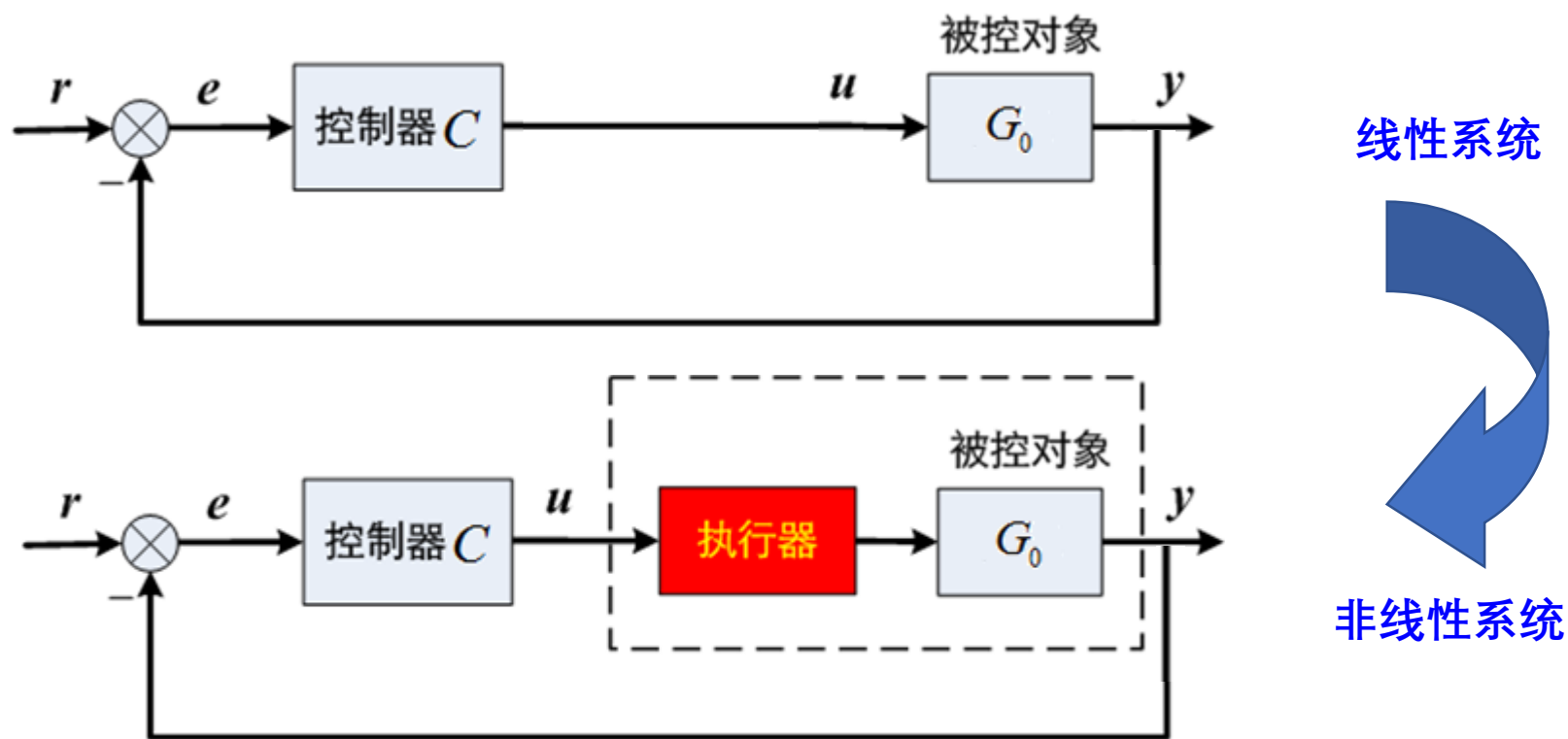
不稳定对象的局部稳定性

5.1.4

积分器的Windup问题



控制系统的许多方面都可以用线性模型来描述，然而，有些非线性情况却需要考虑。执行器存在一些典型的限制：如电机的最大转速限制、峰值力矩限制；阀门的开合不能超过全开或全闭；等。

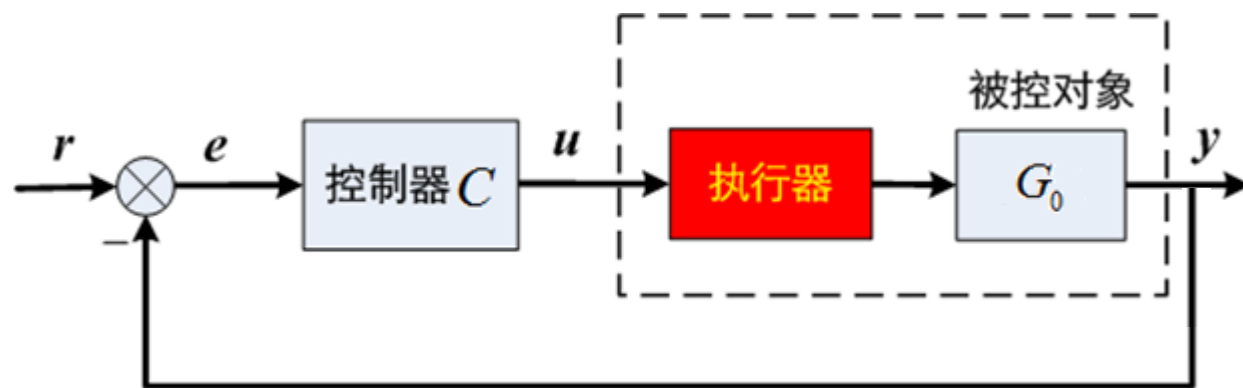
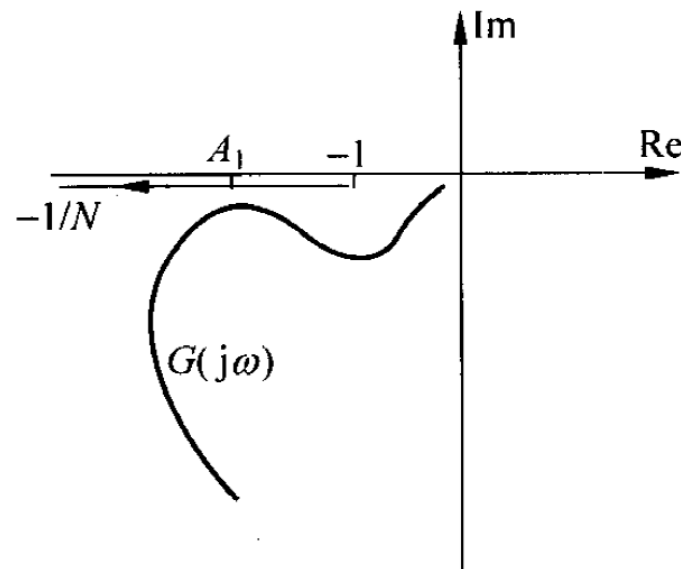
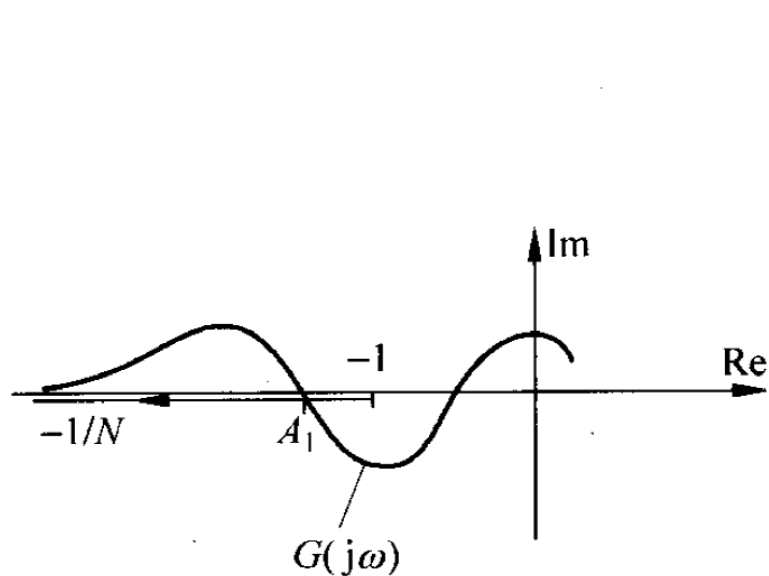




5.1.1 执行器约束问题的提出



执行器饱和的影响：

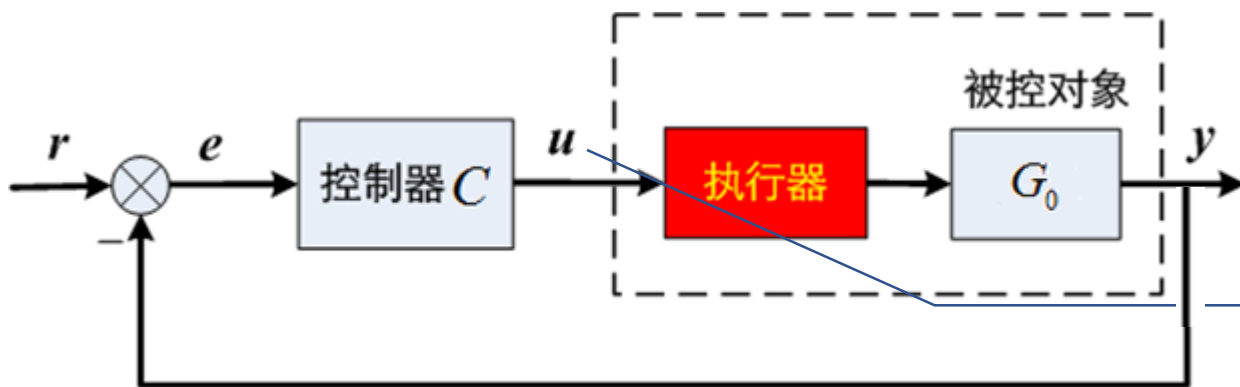


非线性系统

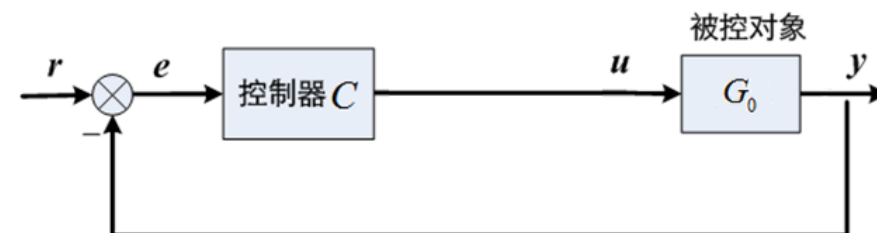


对于在比较宽范围内运行的控制系统，**控制量有可能达到执行器的极限(幅值极限或变化速率极限)**。当这种情况发生时，**反馈回路失效**，系统将运行于**开环状态**，即只要执行器处于饱和，执行器输出就停留在其极限状态，系统的输出相当于开环响应，对于不稳定的被控对象来说，这是非常危险的。

限于在线性系统理论的范围内讨论！



导致控制变量增大的原因是什么？

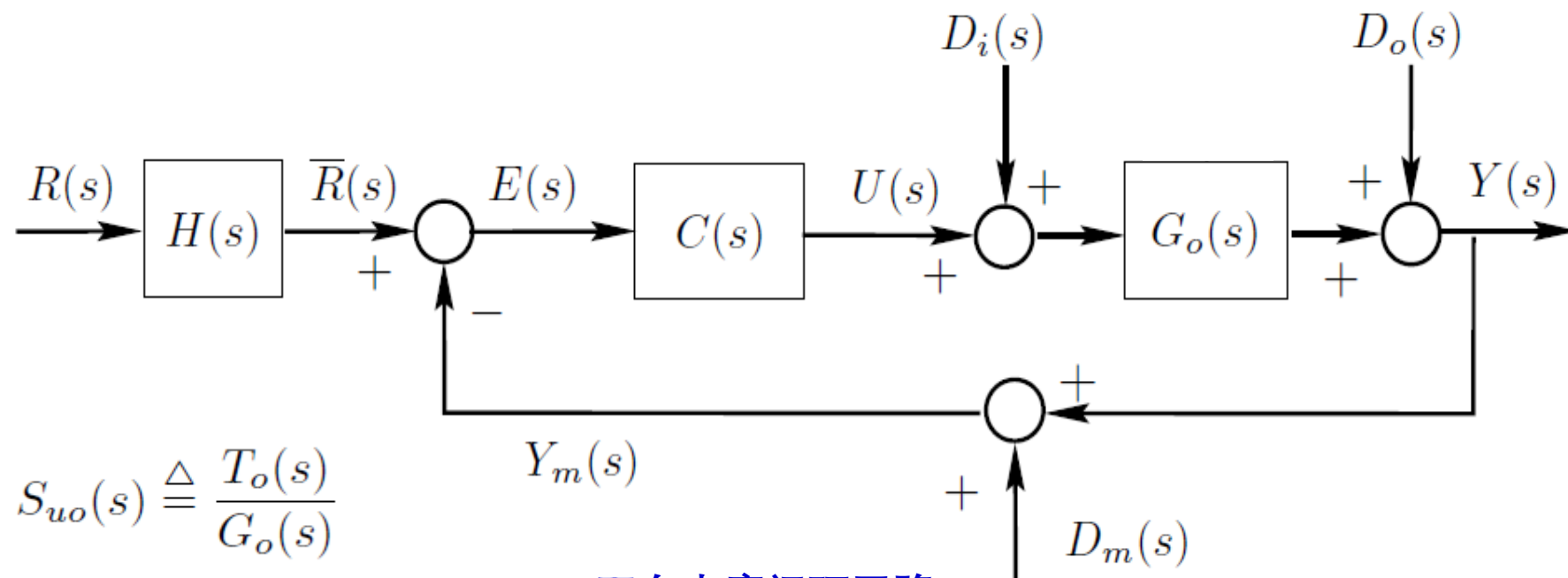




5.1.1 执行器约束问题的提出



◆ 控制变量增大的原因分析



双自由度闭环回路

$$T_o \triangleq \frac{G_o C}{1 + G_o C}$$

$$S_o \triangleq \frac{1}{1 + G_o C}$$

$$S_{uo} \triangleq \frac{C}{1 + G_o C}$$

$$S_{io} \triangleq \frac{G_o}{1 + G_o C}$$



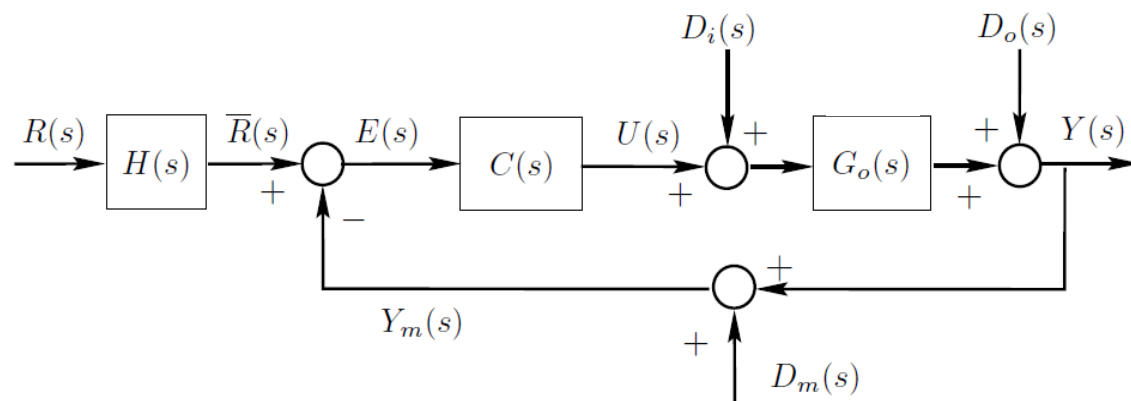


5.1.1 执行器约束问题的提出



◆ 控制变量增大的原因分析

输入与输出表达式关系如下：



$$Y(s) = T_o(s)(H(s)R(s) - D_m(s)) + S_o(s)D_o(s) + S_{io}(s)D_i(s)$$

$$U(s) = S_{uo}(s) \left(H(s)R(s) - D_m(s) - D_o(s) - G_o(s)D_i(s) \right)$$

$$T_o \triangleq \frac{G_o C}{1 + G_o C}$$

$$S_o \triangleq \frac{1}{1 + G_o C}$$

$$S_{uo} \triangleq \frac{C}{1 + G_o C}$$

$$S_{io} \triangleq \frac{G_o}{1 + G_o C}$$



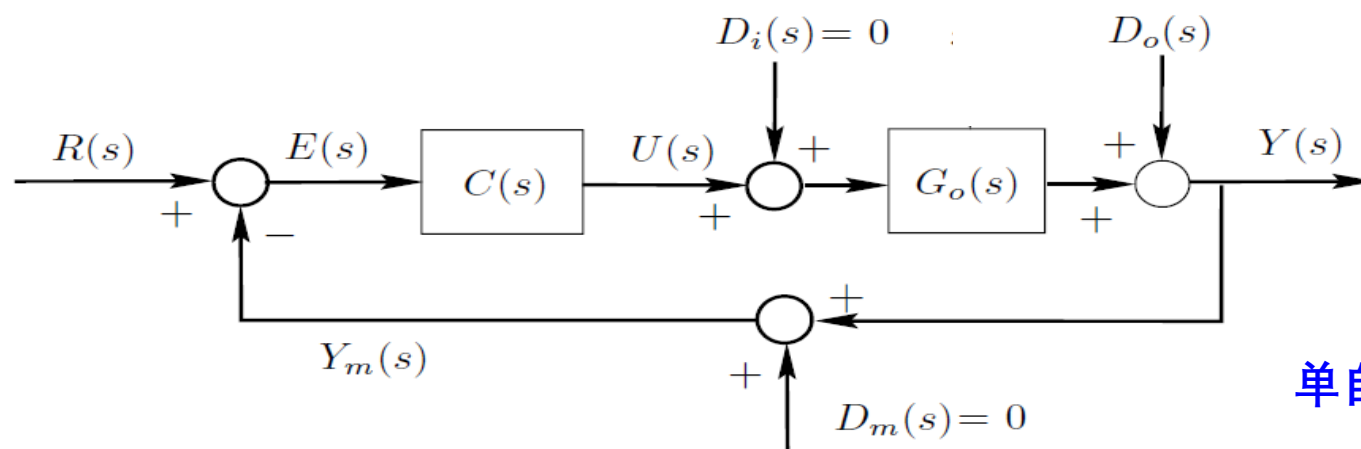


5.1.1 执行器约束问题的提出



◆ 控制变量增大的原因分析

参考信号 $R(s)$ 或者输出扰动信号 $D_o(s)$ 存在大的快速变化时，通常会引起控制器输出信号较大的峰值和变化速率，而输入扰动的高频变化通常已经由对象特性滤掉。



单自由度闭环回路

$$U(s) = S_{uo}(s) \left(R(s) - D_m(s) - D_o(s) - G_o(s)D_i(s) \right)$$

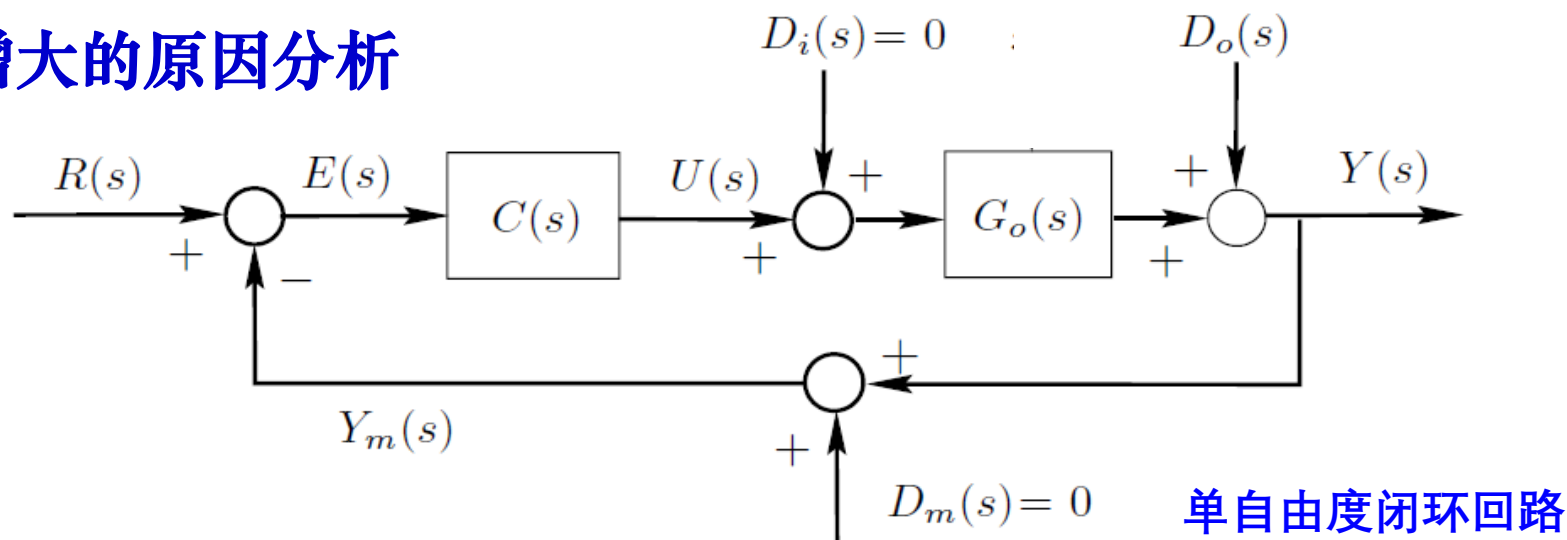




5.1.1 执行器约束问题的提出



◆ 控制变量增大的原因分析



$$U(s) = S_{uo}(s) \left(R(s) - D_m(s) - D_o(s) - G_o(s)D_i(s) \right)$$

$$U(s) = S_{uo}(s)(R(s) - D_o(s))$$

$$S_{uo}(s) \triangleq \frac{T_o(s)}{G_o(s)}$$

$$sU(s) = S_{uo}(s)[sR(s) - sD_o(s)] = \frac{T_o(s)}{G_o(s)}[sR(s) - sD_o(s)]$$



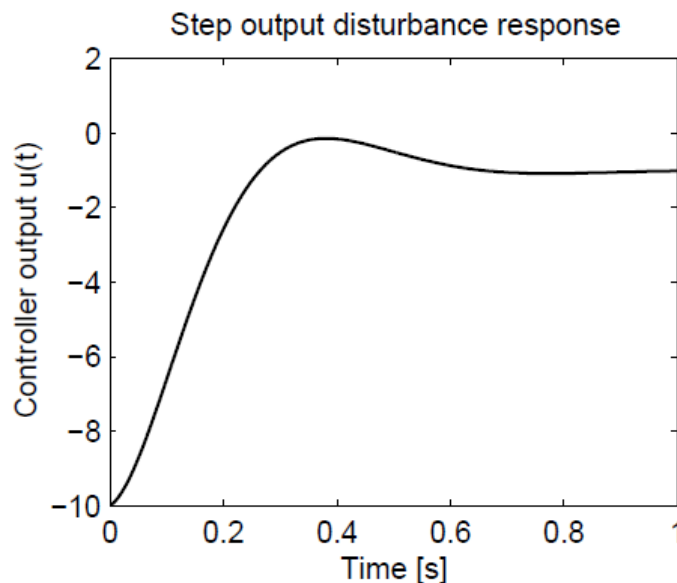
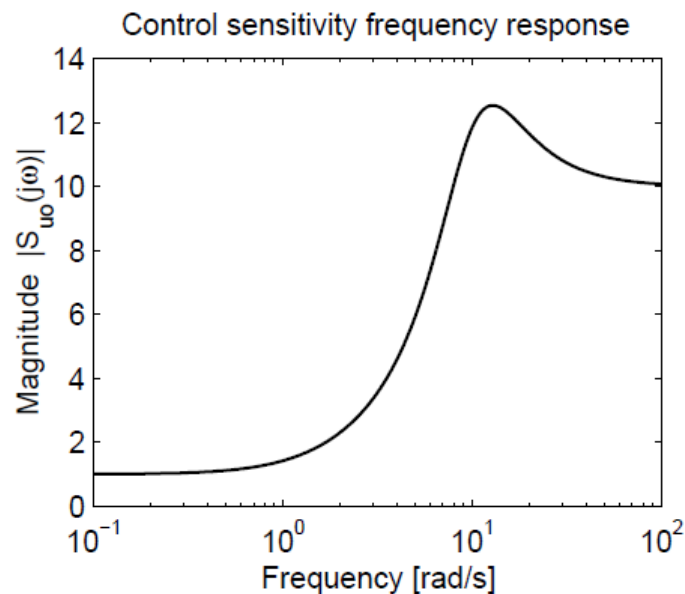


5.1.1 执行器约束问题的提出

闭环带宽近似10倍于对象带宽

◆ 控制变量增大的原因分析—例1

$$G_o(s) = \frac{10}{(s+10)(s+1)} \quad T_o(s) = \frac{100}{s^2 + 12s + 100}$$



如果闭环带宽远大于开环模型的带宽，则 S_{uo} 将显著地放大参考信号和输出扰动中的高频成分。

$$U(s) = S_{uo}(s)(R(s) - D_o(s))$$

$$sU(s) = S_{uo}(s)[sR(s) - sD_o(s)]$$





5.1.1 执行器约束问题的提出



◆ 问题的提出——执行器约束带来的设计约束

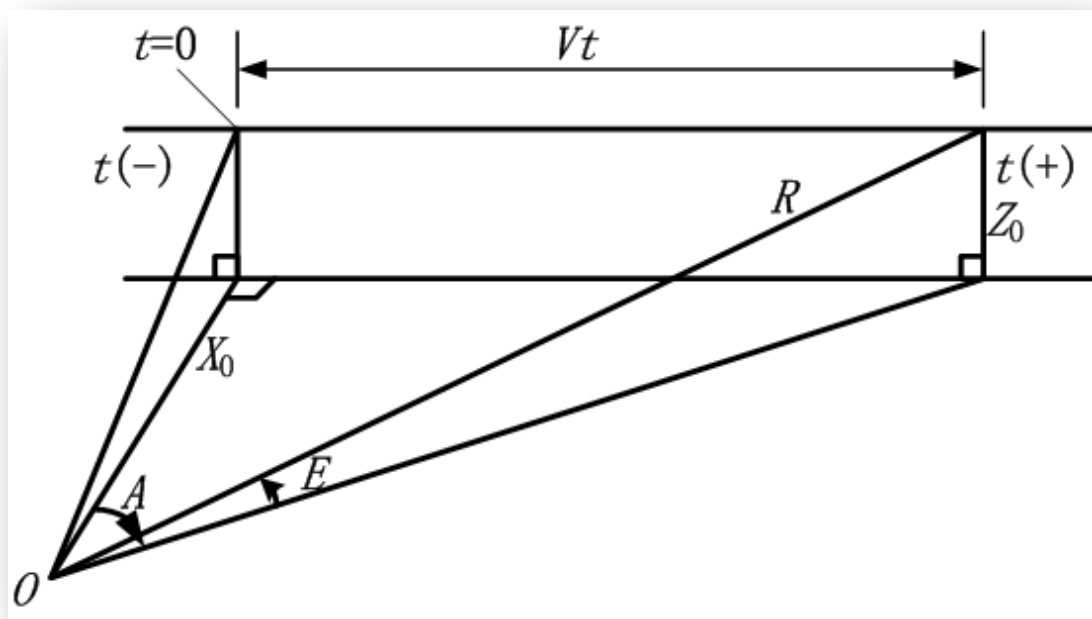
对于实际的执行器，其能力都是有限的，存在幅值和变化速率的约束（分别称为**饱和与转换速率限制**）。如果忽视这些约束，可能带来严重的性能下降，甚至使系统失稳。



为了避免执行器饱和或转换速率限制引起的问题，通常有必要对闭环带宽设置一个上界！

◆ 问题的提出——执行器约束带来的设计约束

在设计流程的方案设计阶段，需要根据性能指标要求、信号分析等确定执行器的能力并完成选型。



3.1.1 输入信号的分析



例1 跟踪直线飞行目标时间伺服系统的输入

3

部件选择

- 驱动电机的额定速度和力矩；
- 传感器的量程和其它参数（如最大速度）等。

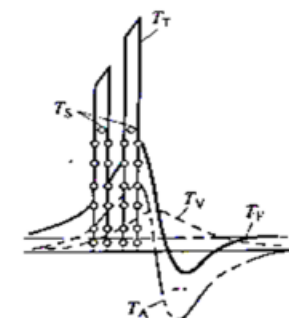


图 3-5 跟踪过程的力矩分量
 T_A —加速度力矩； T_V —速度力矩； T_F —摩擦力矩； T_S —冲击力矩； T_T —总负载力矩

负载力矩：加速度力矩、速度力矩、摩擦力矩、冲击力矩





5.1.1 执行器约束问题的提出

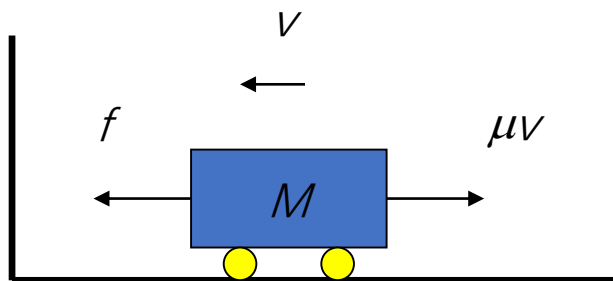


◆ 问题的提出——执行器约束带来的设计约束

对于有些控制问题，在问题描述时就需要明确执行器的约束，并在设计时加以考虑；否则可能会导致问题本身失去意义。

$x(t_f) = x_f$ 更一般地，还可以考虑状态的约束。

$$\begin{aligned} L(t, x(t), u(t)) &= 1 \\ K(t_f, x_f) &= 0 \quad |u| \leq \bar{u} \end{aligned}$$



$$\dot{x} = f(t, x, u), \quad x(t_0) = x_0$$

$$u \in U \subset \mathbb{R}^m$$

$$J(u) := \int_{t_0}^{t_f} L(t, x(t), u(t)) dt + K(t_f, x_f)$$

Bolza问题





5.1 执行器的约束问题



5.1.1

执行器约束问题的提出

5.1.2

执行器约束的描述方法

5.1.3

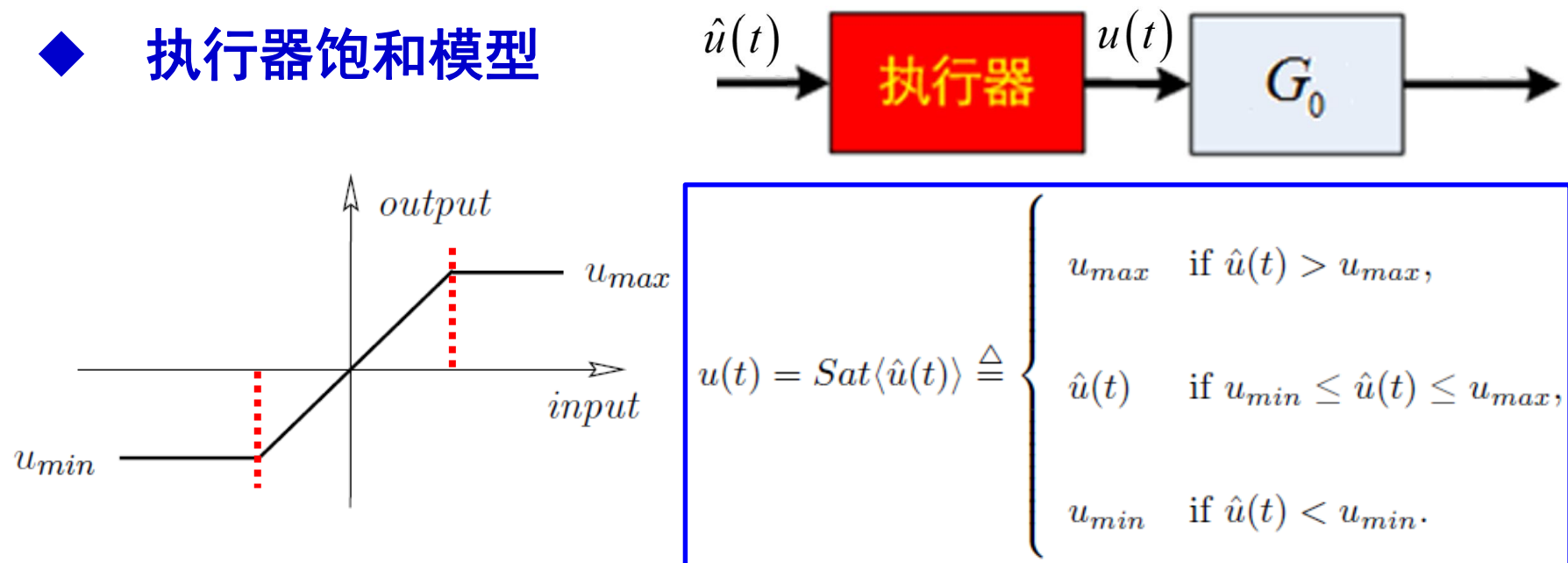
不稳定对象的局部稳定性

5.1.4

积分器的Windup问题



◆ 执行器饱和模型

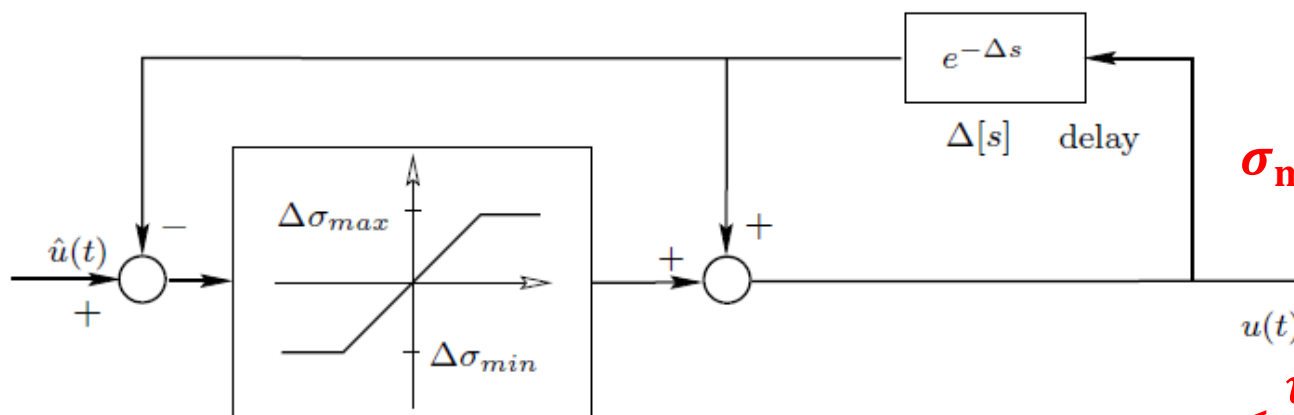


u_{max} ——执行器输出的最大幅值； $u(t)$ ——执行器实际输出；

u_{min} ——执行器输出的最小幅值； $\hat{u}(t)$ ——执行器期望输出；

$Sat\langle \bullet \rangle$ ——饱和函数。

◆ 执行器转换速率限制模型



$$\sigma_{\min} \leq \frac{du(t)}{dt} \leq \sigma_{\max}$$

$$\sigma_{\min} \leq \frac{u(t) - u(t - \Delta)}{\Delta} \leq \sigma_{\max}$$

$$\dot{u}(t) = \text{Sat}(\dot{\hat{u}}(t)) \triangleq \begin{cases} \sigma_{\max} & \text{if } \dot{\hat{u}}(t) > \sigma_{\max}, \\ \dot{\hat{u}}(t) & \text{if } \sigma_{\min} \leq \dot{\hat{u}}(t) \leq \sigma_{\max}, \\ \sigma_{\min} & \text{if } \dot{\hat{u}}(t) < \sigma_{\min}. \end{cases}$$

σ_{\max} —— 执行器速率最大值;

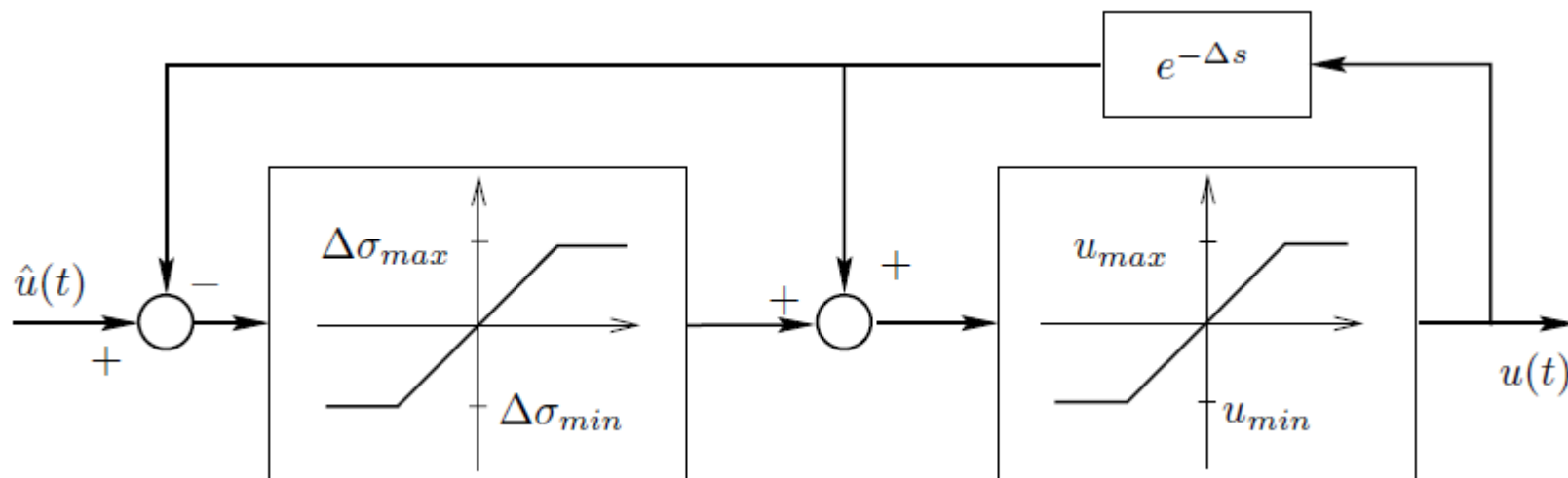
σ_{\min} —— 执行器速率最小值;

$\dot{u}(t)$ —— 执行器实际输出速率;

$\dot{\hat{u}}(t)$ —— 执行器期望输出速率;

$\text{Sat}(\cdot)$ —— 饱和函数。

◆ 执行器同时具有饱和与转换速率限制的模型



u_{max} ——执行器输出的最大幅值；

$u(t)$ ——执行器实际输出；

u_{min} ——执行器输出的最小幅值；

$\hat{u}(t)$ ——执行器期望输出；

$Sat\langle\bullet\rangle$ ——饱和函数。

σ_{max} ——执行器速率最大值；

σ_{min} ——执行器速率最小值；

$\dot{u}(t)$ ——执行器实际输出速率；

$\dot{\hat{u}}(t)$ ——执行器期望输出速率；

$Sat\langle\bullet\rangle$ ——饱和函数。



5.1 执行器的约束问题



5.1.1

执行器约束问题的提出

5.1.2

执行器约束的描述方法

5.1.3

不稳定对象的局部稳定性

5.1.4

积分器的Windup问题

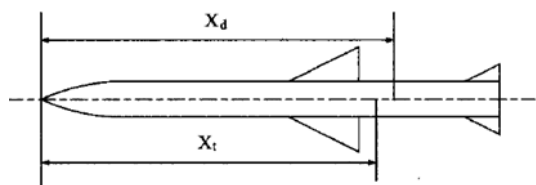


从控制的角度来说，具有自平衡特性的被控对象是稳定的被控对象，而不具有自平衡特性的被控对象就是不稳定的被控对象。

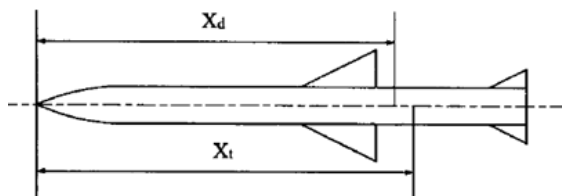
$$l\ddot{\theta} - g\theta = -\ddot{y} = -a(t)$$

$$\frac{\Theta(s)}{A(s)} = -\frac{1/l}{s^2 - g/l}$$

$$G(s) = \frac{\omega_0^2}{s^2 - \omega_0^2}$$

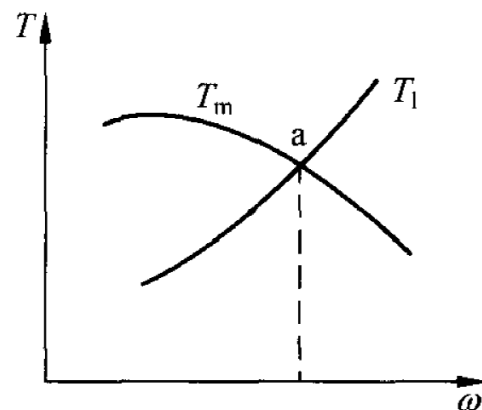


静稳定导弹



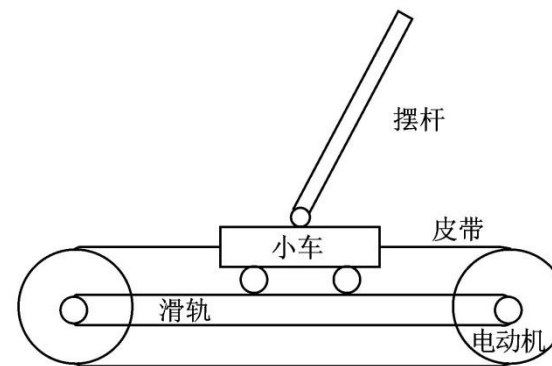
静不稳定导弹

X_d : 压心
 X_t : 质心



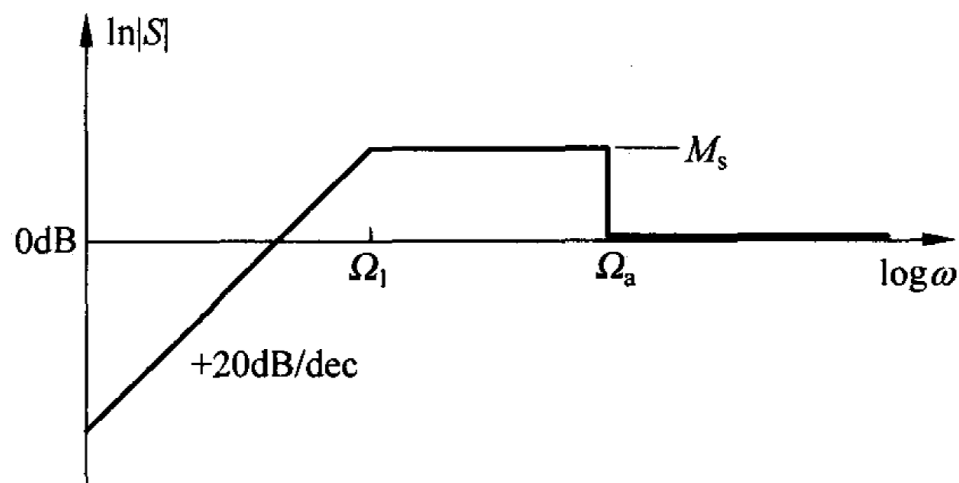
发动机的力矩特性

典型的不稳定被控对象



对于不稳定的被控对象，Bode积分约束对控制设计的影响更为突出。

$$S = \frac{1}{1 + KG} \quad \int_0^\infty \ln |S(j\omega)| d\omega = \pi \sum_{i=1}^N \operatorname{Re}(p_i), \quad \nu > 1$$

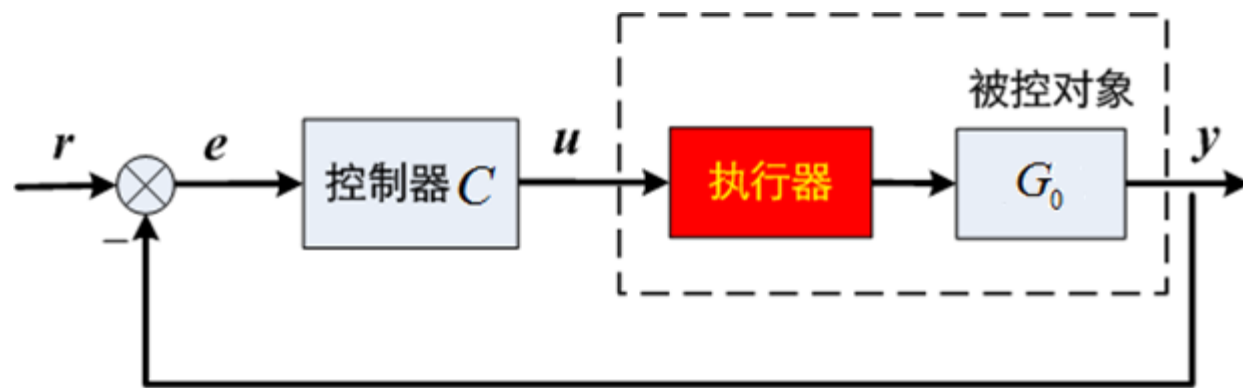


灵敏度函数的一种近似

除了考虑不稳定的极点对性能指标 M_s 的影响，还要认识到不稳定对象的控制系统都只是局部稳定的，这是由执行机构的限制造成的。



执行器存在一些典型的限制：电机最大转速限制、峰值力矩限制；
阀门的开合不能超过全开或全闭等。



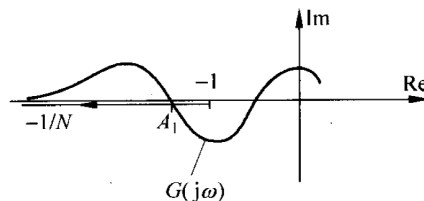
当控制量达到执行器的极限(幅值极限或变化速率极限)时，
反馈回路失效，系统将运行于开环状态。

控制量变大的原因:

$$U(s) = S_{uo}(s)(R(s) - D_o(s))$$

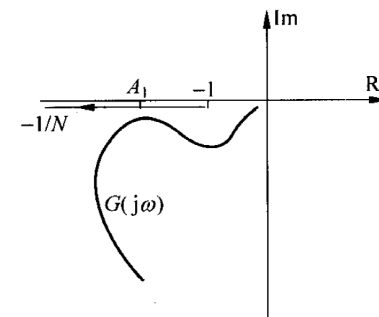
$$sU(s) = S_{uo}(s)[sR(s) - sD_o(s)]$$

$$S_{uo}(s) \triangleq \frac{T_o(s)}{G_o(s)}$$

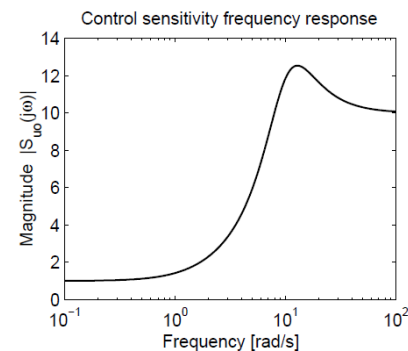


$$G_o(s) = \frac{10}{(s+10)(s+1)}$$

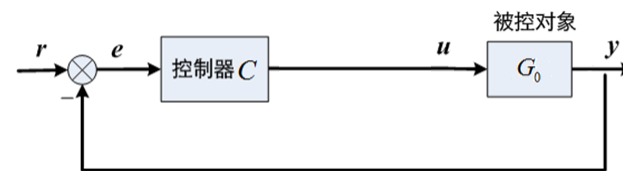
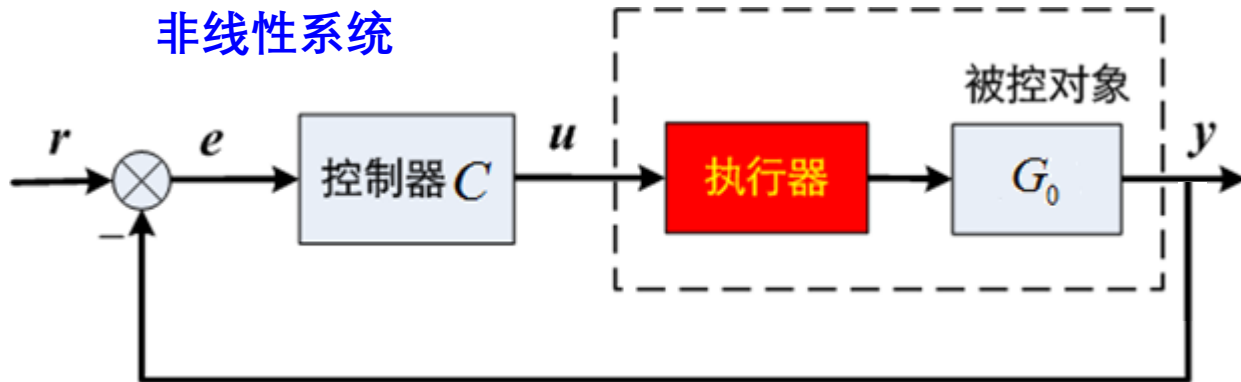
执行器饱和的影响:



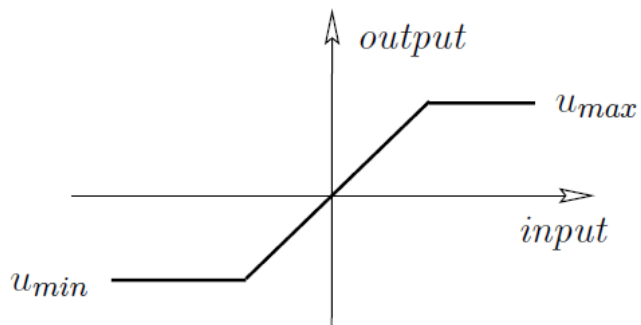
$$T_o(s) = \frac{100}{s^2 + 12s + 100}$$



非线性系统

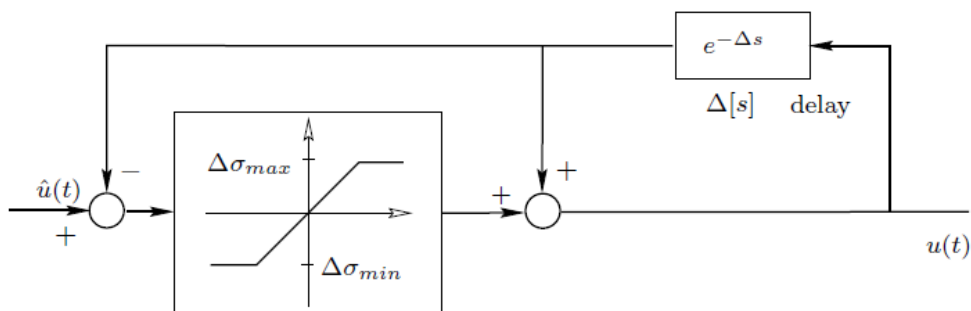


• 执行器饱和模型



$$u(t) = Sat\langle \hat{u}(t) \rangle \triangleq \begin{cases} u_{max} & \text{if } \hat{u}(t) > u_{max}, \\ \hat{u}(t) & \text{if } u_{min} \leq \hat{u}(t) \leq u_{max}, \\ u_{min} & \text{if } \hat{u}(t) < u_{min}. \end{cases}$$

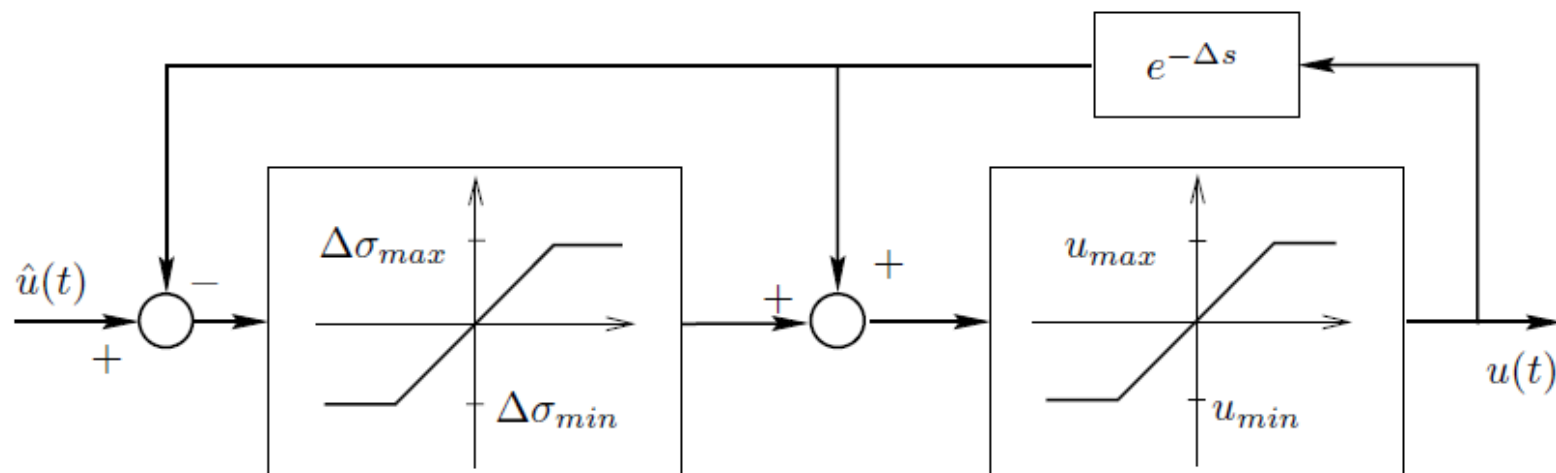
• 执行器转换速率限制模型



$$\dot{u}(t) = Sat\langle \dot{\hat{u}}(t) \rangle \triangleq \begin{cases} \sigma_{max} & \text{if } \dot{\hat{u}}(t) > \sigma_{max}, \\ \dot{\hat{u}}(t) & \text{if } \sigma_{min} \leq \dot{\hat{u}}(t) \leq \sigma_{max}, \\ \sigma_{min} & \text{if } \dot{\hat{u}}(t) < \sigma_{min}. \end{cases}$$

◆ 执行器约束的描述方法

- 执行器同时具有饱和与转换速率限制的模型



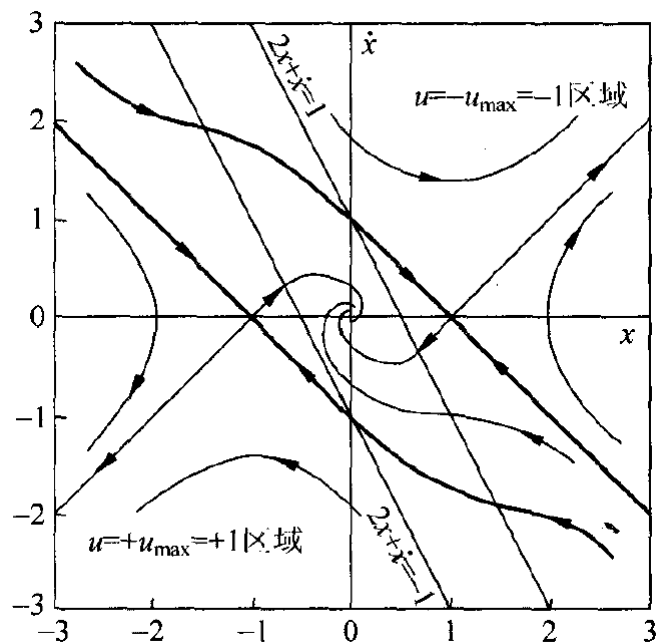
$$u(t) = Sat\langle \hat{u}(t) \rangle \triangleq \begin{cases} u_{max} & \text{if } \hat{u}(t) > u_{max}, \\ \hat{u}(t) & \text{if } u_{min} \leq \hat{u}(t) \leq u_{max}, \\ u_{min} & \text{if } \hat{u}(t) < u_{min}. \end{cases} \quad \dot{u}(t) = Sat\langle \dot{\hat{u}}(t) \rangle \triangleq \begin{cases} \sigma_{max} & \text{if } \dot{\hat{u}}(t) > \sigma_{max}, \\ \dot{\hat{u}}(t) & \text{if } \sigma_{min} \leq \dot{\hat{u}}(t) \leq \sigma_{max}, \\ \sigma_{min} & \text{if } \dot{\hat{u}}(t) < \sigma_{min}. \end{cases}$$

◆ 不稳定对象的局部稳定性

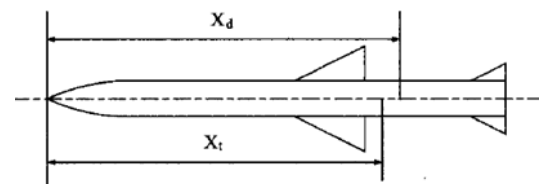
$$u = -2x - \dot{x} = -1 \quad \ddot{x} = x - 1$$

$$u = -2x - \dot{x} = +1 \quad \ddot{x} = x + 1$$

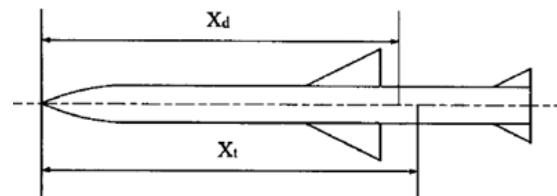
$$\ddot{x} + \dot{x} + x = 0$$



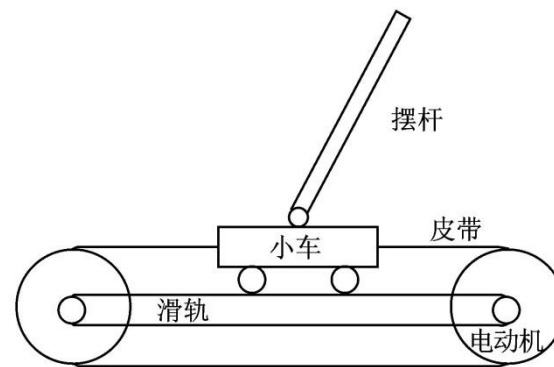
$$G(s) = \frac{\omega_0^2}{s^2 - \omega_0^2}$$



静稳定导弹



静不稳定导弹



考虑典型的不稳定被控对象及PD控制器，则平衡点是稳定的焦点，控制为0时则为鞍点：

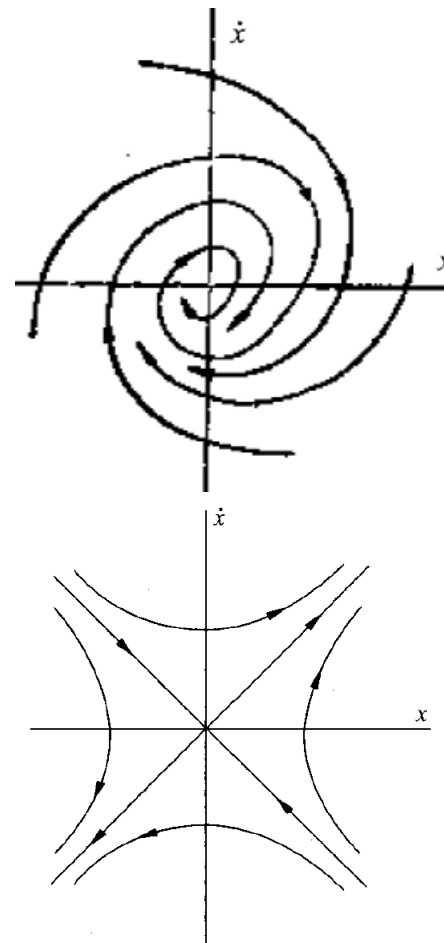
$$G(s) = \frac{\omega_0^2}{s^2 - \omega_0^2}$$

$$\ddot{x} = \omega_0^2 x + \omega_0^2 u$$

$$u = -2x - \frac{1}{\omega_0} \dot{x} \Rightarrow \ddot{x} + \omega_0 \dot{x} + \omega_0^2 x = 0$$

$$u=0 \Rightarrow \ddot{x} = x$$

$$\omega_0 = 1$$





5.1.3 不稳定对象的局部稳定性



发生饱和时:

$$\ddot{x} = \omega_0^2 x + \omega_0^2 u$$

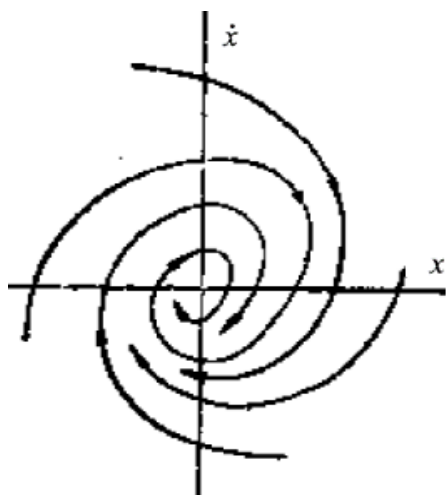
$$\omega_0 = 1$$

$$u = -2x - \dot{x} = -1$$

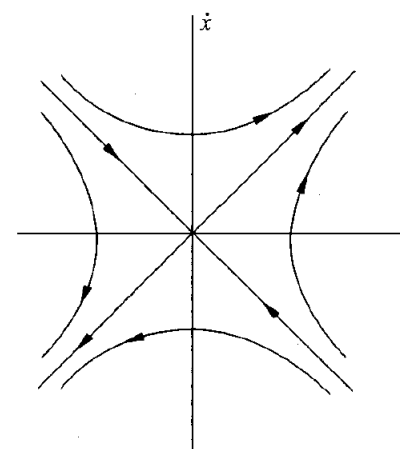
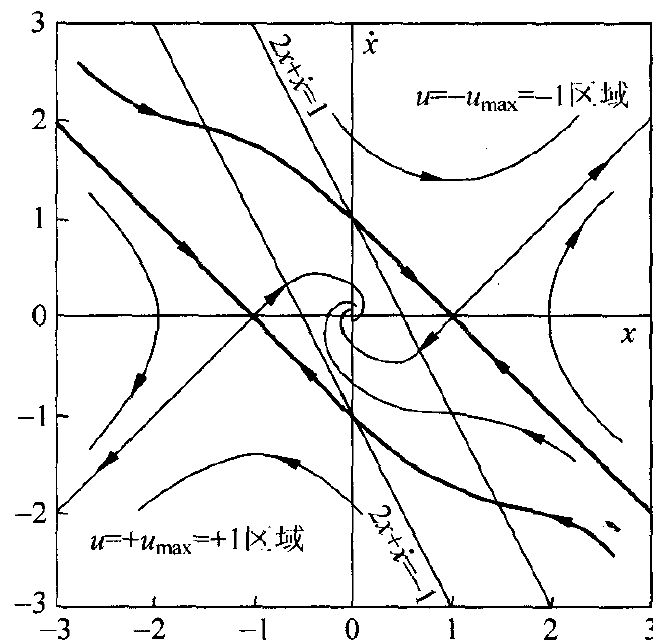
$$\ddot{x} = x - 1$$

$$u = -2x - \dot{x} = +1$$

$$\ddot{x} = x + 1$$



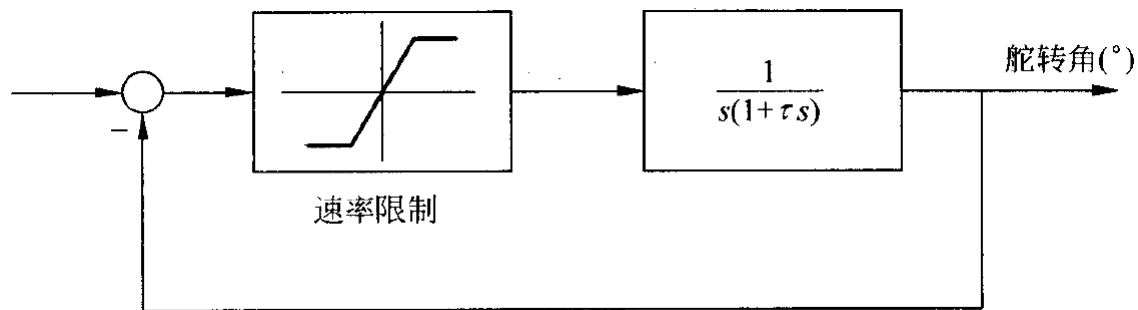
$$\ddot{x} + \omega_0 \dot{x} + \omega_0^2 x = 0$$



$$\ddot{x} = x$$

轮船、飞行器等对象普遍存在执行机构速率限制，核电站中核燃料的进料速率也存在限制。

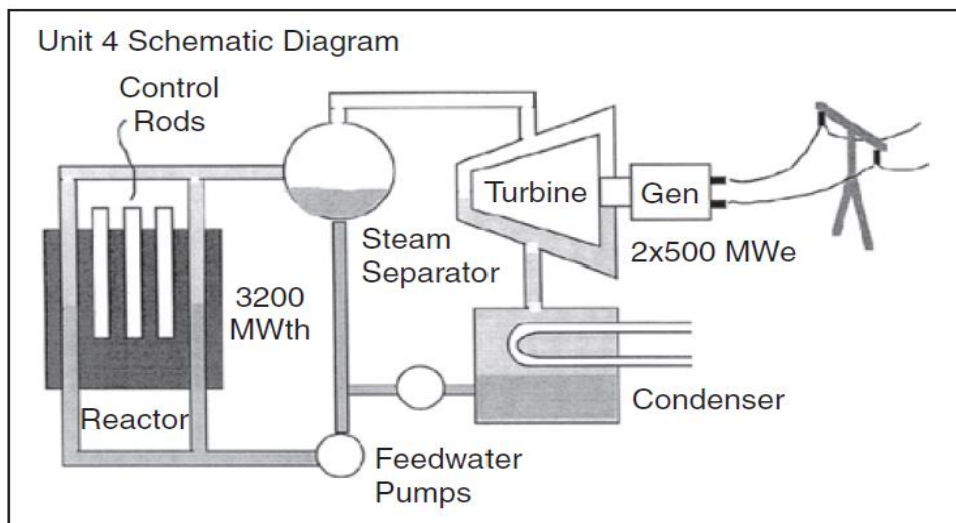
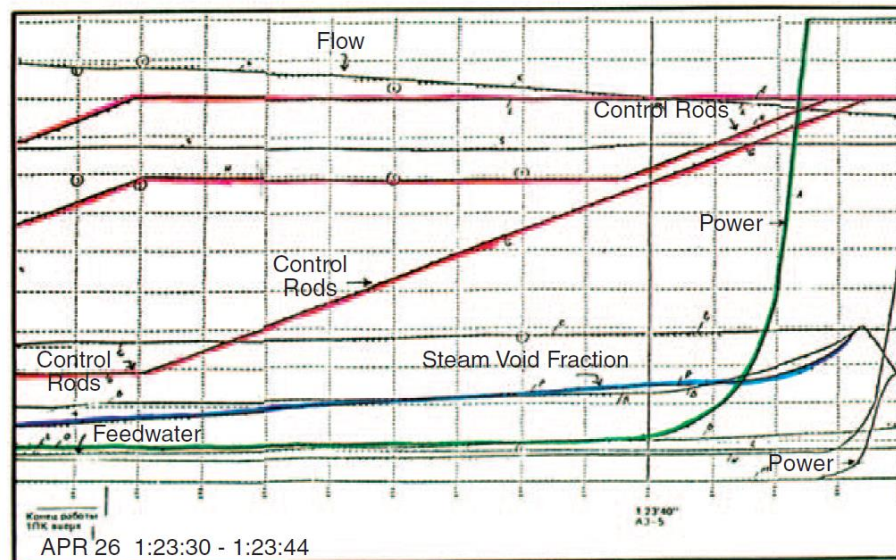
速率和幅值饱和存在的问题是一样的，对于不稳定对象的控制系统来说，进入饱和后，相当于系统开环，系统的运行由对象本身的不稳定特性决定，一旦脱离吸引区就会发散。



Chernobyl nuclear power plant shortly after the accident on 26 April 1986.

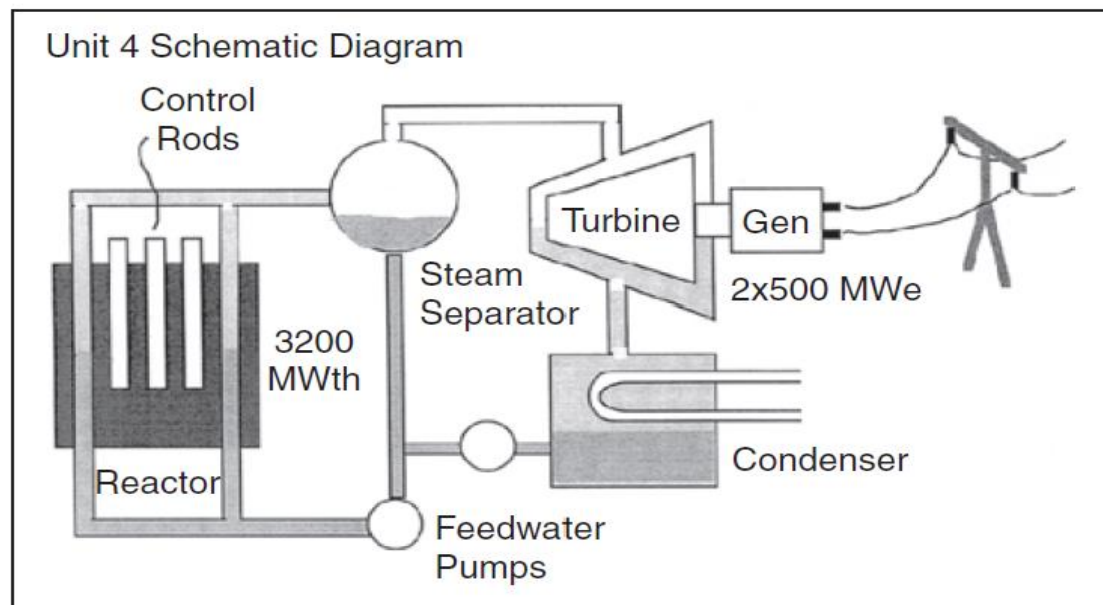
运行记录：
燃料棒
功率

We will consider unstable to be
synonymous with dangerous.



Chernobyl nuclear power plant shortly after the accident on 26 April 1986.

由于操作失误，核反应堆燃料棒的进料速率已经达到最大值，使整个系统的状态脱离吸引区，沿不稳定的相轨迹迅速发散，10s内将核反应堆的功率升至300000MW，超出额定功率100倍，摧毁了反应堆。最后，高压蒸汽顶开了1000t重的防护盖板，造成重大的核污染事故。



Chernobyl nuclear power plant shortly after the accident on 26 April 1986.



5.1 执行器的约束问题



5.1.1

执行器约束问题的提出

5.1.2

执行器约束的描述方法

5.1.3

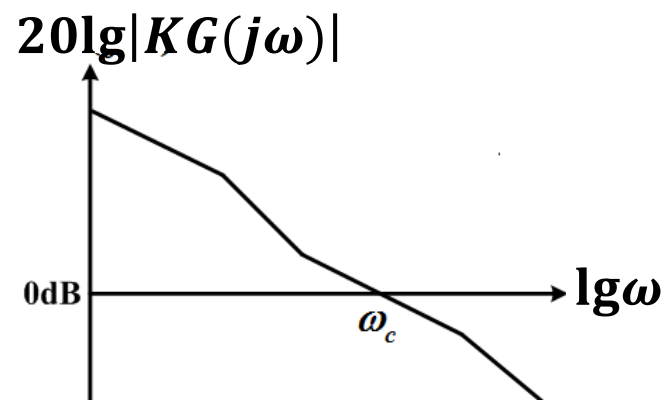
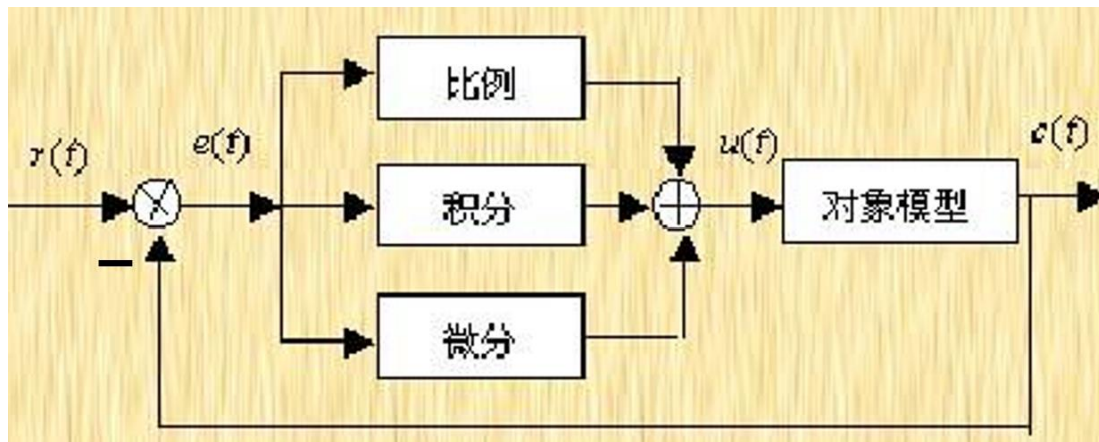
不稳定对象的局部稳定性

5.1.4

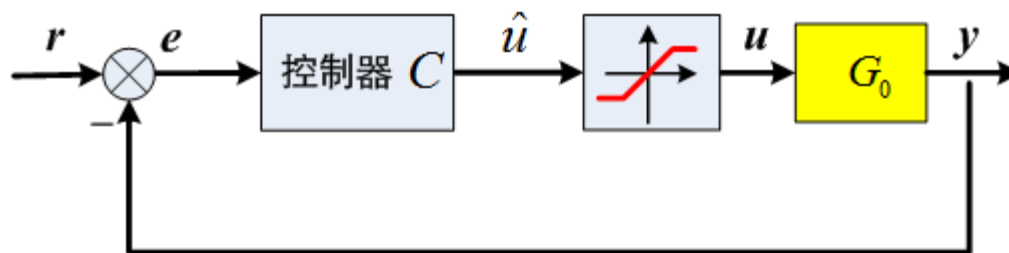
积分器的Windup问题



控制中经常使用积分器来消除稳态误差，其前提是控制回路工作在线性范围内。



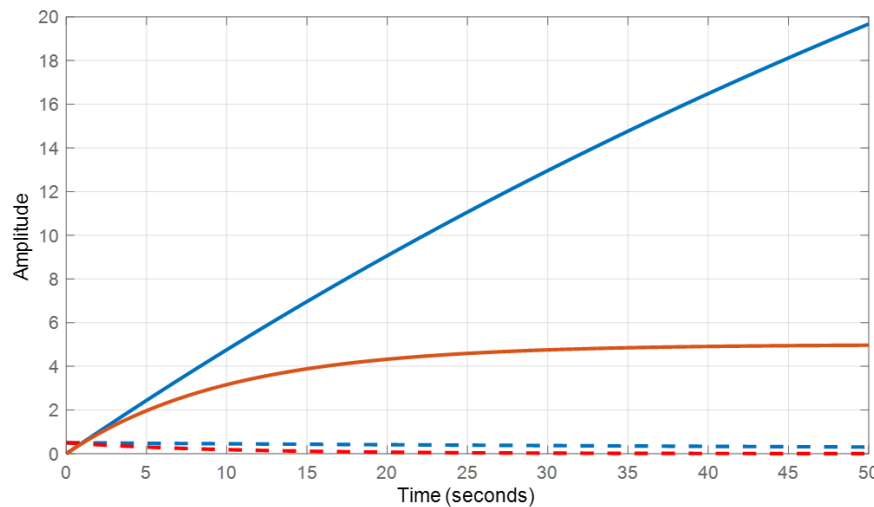
$$C(s) = \frac{1}{s}$$



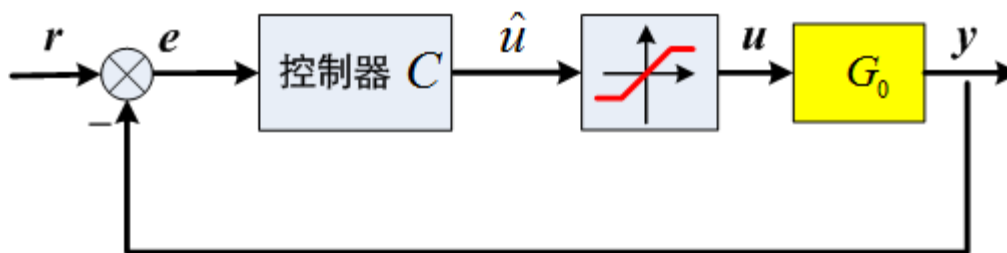
当执行器达到其约束边界而进入饱和后，在系统误差的作用下，积分器的输出不断累积，导致整个控制器的输出 \hat{u} 可能会累积到很大的值，但执行器由于处于饱和状态而无法响应。

$$e(t) = 0.5e^{-0.1t}$$

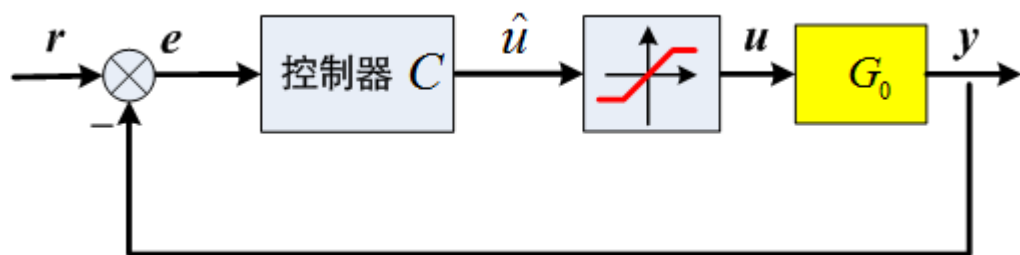
$$e(t) = 0.5e^{-0.01t}$$



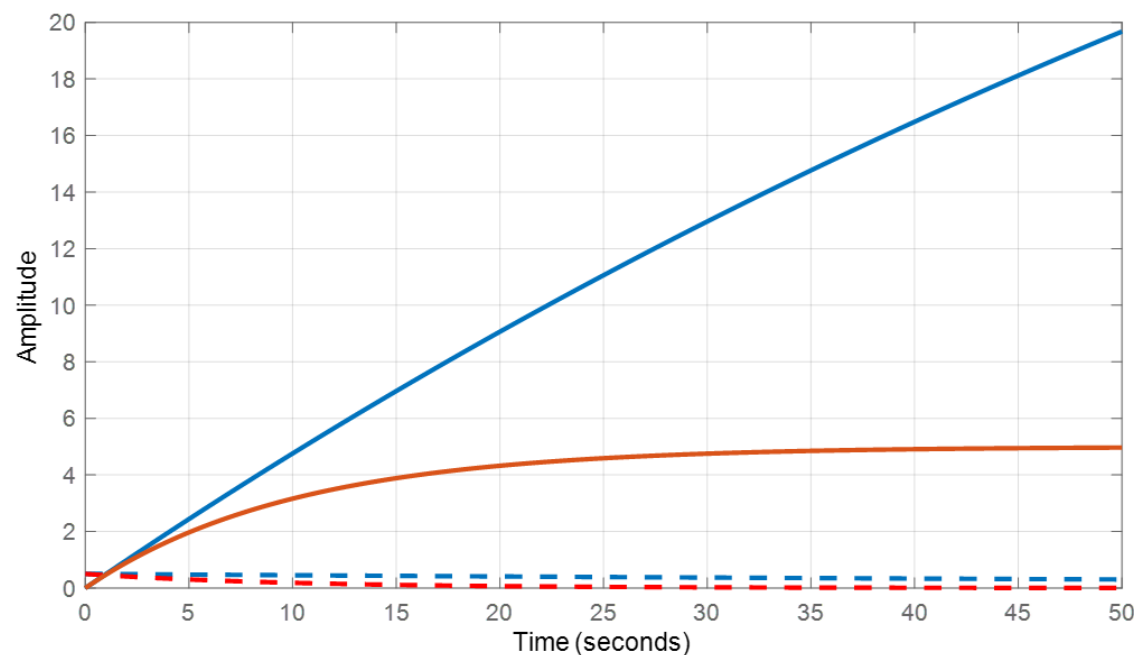
$$C(s) = \frac{1}{s} \Rightarrow \dot{\hat{u}} = e(t)$$



积分器的累积值过大，必然会导致执行器退出饱和时间变长，而且控制器的输出回到饱和边界之内(线性范围)后，这一积分值作为初始条件，会导致很大的暂态响应，这一现象称为**Windup**，会严重影响控制系统的性能，甚至使得系统失稳。



$$C(s) = \frac{1}{s} \Rightarrow \dot{\hat{u}} = e(t)$$



◆例2——积分器的Windup问题

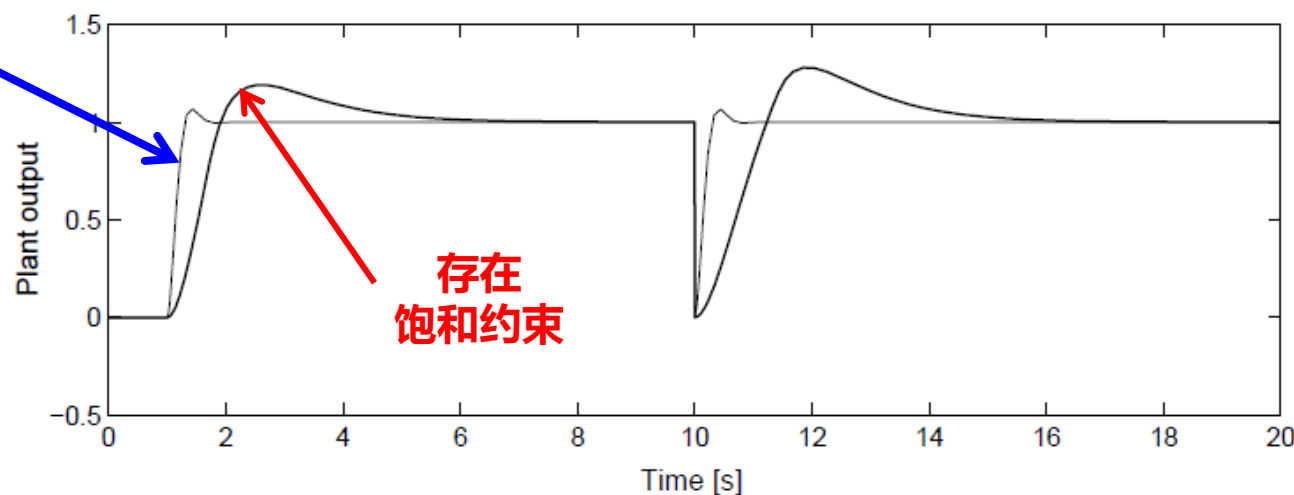
$$S_{uo}(s) \triangleq \frac{T_o(s)}{G_o(s)}$$

$$G_o(s) = \frac{2}{(s+1)(s+2)}$$

$$T_o(s) = \frac{100}{s^2 + 13s + 100}$$

$$C(s) = \frac{50(s+1)(s+2)}{s(s+13)}$$

不存在
饱和约束



在1秒时施加单位阶跃输入信号，在10秒存在一个负向单位阶跃输出扰动信号。执行器的线性工作范围是[-3,3]。



5.1.4 积分器的Windup问题



积分器的Windup现象：很大的暂态响应，会严重影响系统的性能，甚至使系统失稳。

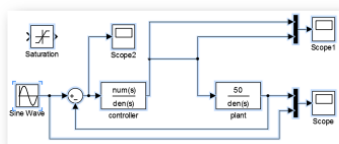


3.1.1 输入信号的分析



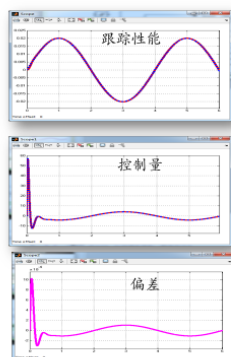
例：探究饱和和非线性环节对控制系统性能的影响

无饱和环节



指令频率0.25Hz, 幅值0.02

最大控制量<60, 最大偏差0.001

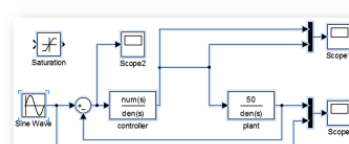


3.1.1 输入信号的分析



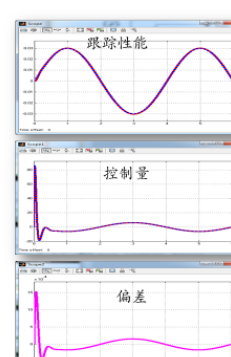
例：探究饱和和非线性环节对控制系统性能的影响

无饱和环节



指令频率0.25Hz, 幅值0.03

最大控制量>80, 最大偏差0.0015

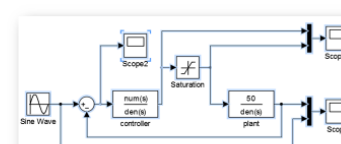


3.1.1 输入信号的分析



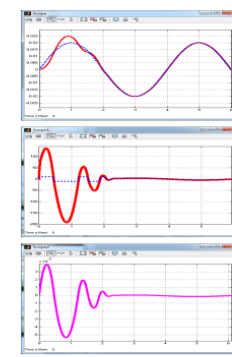
例：探究饱和和非线性环节对控制系统性能的影响

有饱和环节



指令频率0.25Hz, 幅值0.02

最大控制量200, 最大偏差0.005

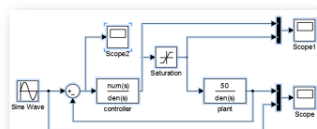


3.1.1 输入信号的分析



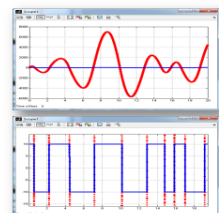
例：探究饱和和非线性环节对控制系统性能的影响

有饱和环节

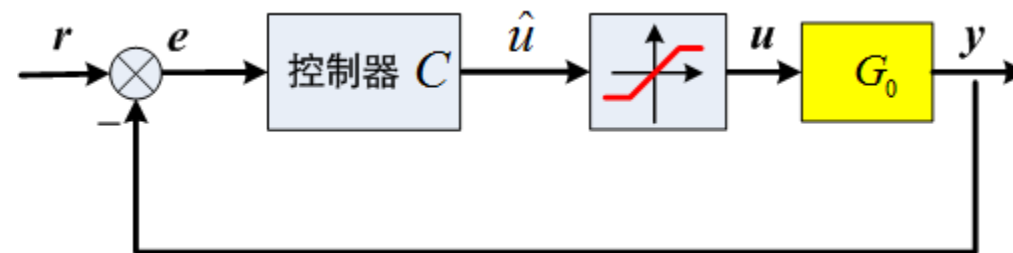


指令频率0.25Hz, 幅值0.03

最大控制量7000, 不稳定



如何解决这一问题?





◆ 执行器约束的解决办法

(1) 保证执行器不会超出范围。

——对执行器要求过高，或导致性能不必要的下降。

(2) 修改设计，应对约束。

——对于一定程度（如100%）的超出约束，可获得满意的效果。对于存在更严重的超出约束的情况，可能是由于针对指定的性能指标执行器选型不当。

当执行器达到约束的边界时切断积分作用！





5.2 Anti-Windup设计



5.2.1

Anti-Windup设计策略

5.2.2

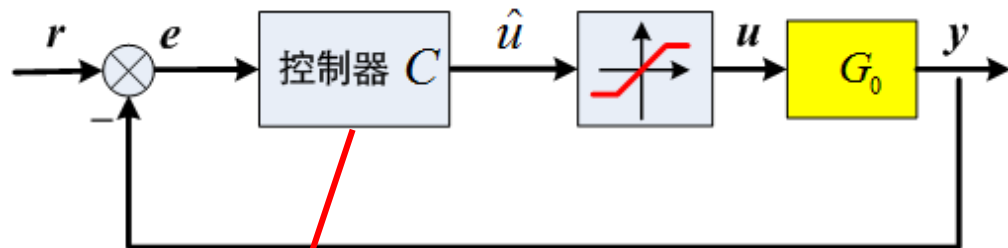
Anti-Windup设计原理分析

5.2.3

Anti-Windup的多种实现形式



◆ 带有Anti-Windup的PI控制器

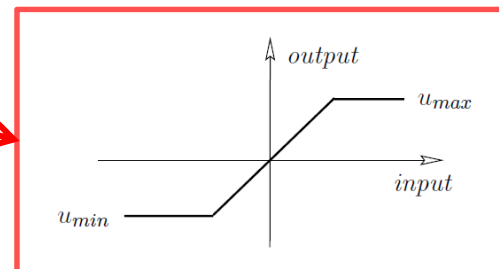
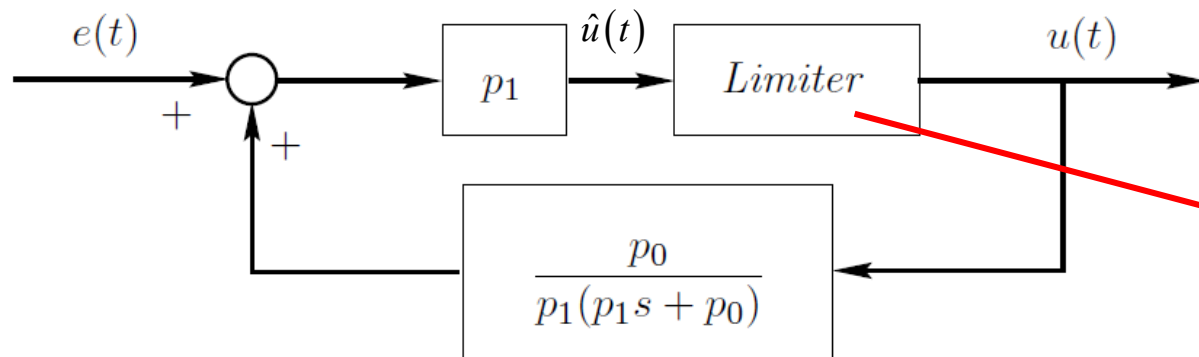


线性工作范围内的传递函数:

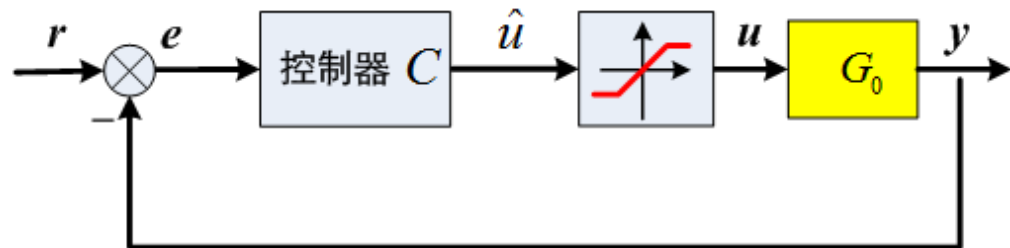
$$\frac{U(s)}{E(s)} = \frac{p_1 s + p_0}{s}$$

$$C(s) = \frac{p_1 s + p_0}{s}, p_0 > 0, p_1 > 0$$

$$\frac{U(s)}{E(s)} = \frac{p_1}{1 - \frac{p_1 p_0}{p_1(p_1 s + p_0)}} = \frac{p_1(p_1 s + p_0)}{(p_1 s + p_0) - p_0} = \frac{p_1 s + p_0}{s}$$



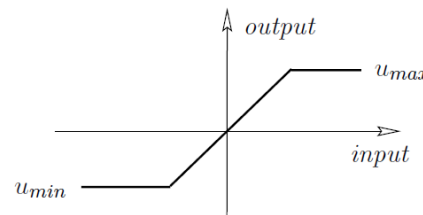
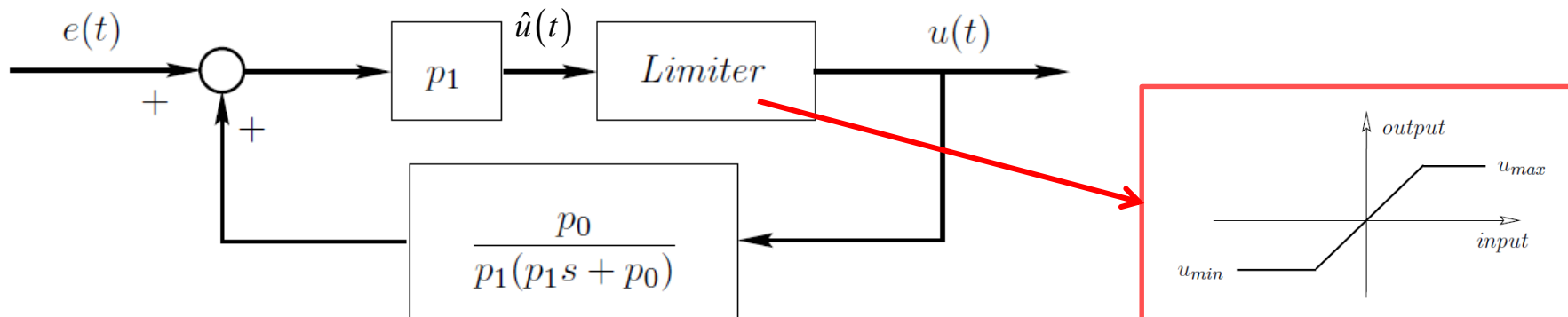
◆ 带有Anti-Windup的PI控制器



线性工作范围内的传递函数:

$$\frac{U(s)}{E(s)} = \frac{p_1 s + p_0}{s}$$

在执行器线性工作范围内，控制器实现比例+积分的作用，而执行器达到约束边界时，控制器切除了积分作用。

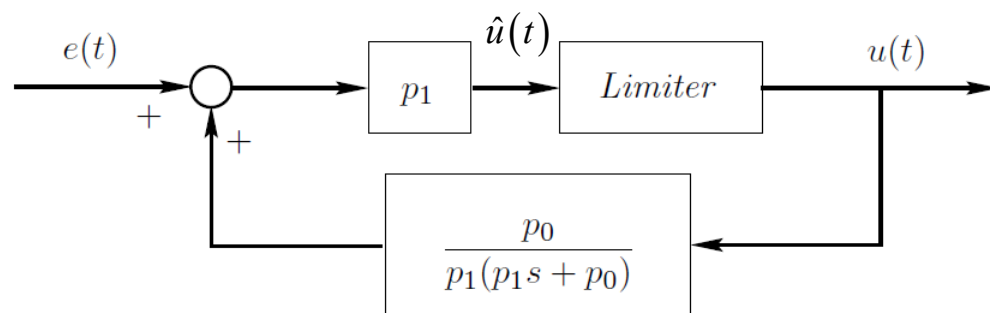




◆ 带有Anti-Windup的PI控制器

——总结Anti-Windup设计策略

- (1) 控制器的动态由对象的实际输入信号（执行器的实际输出）来驱动（误差直接驱动比例环节）；
- (2) 由对象的实际输入信号驱动时，控制器的动态是稳定的。



执行器达到约束边界时，控制器切除了积分作用！



5.2.1 Anti-Windup的设计策略



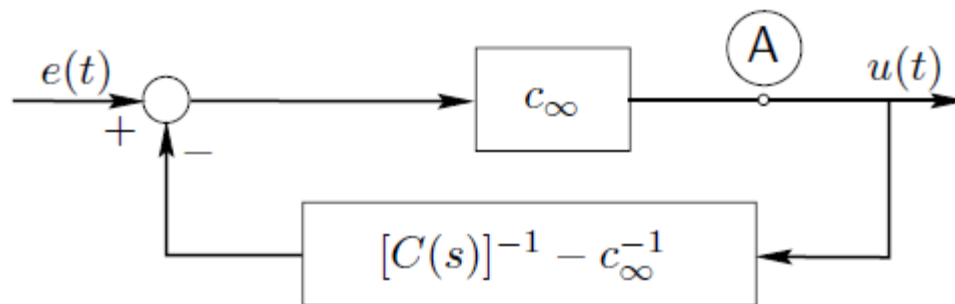
◆ Anti-Windup的基本结构

假设控制器是双正则的最小相位系统，将其分解为

比例项和严格正则项：

$$C(s) = c_{\infty} + \bar{C}(s)$$

$$\begin{aligned}\frac{U(s)}{E(s)} &= \frac{c_{\infty}}{1 + ([C(s)]^{-1} - c_{\infty}^{-1})c_{\infty}} \\ &= \frac{c_{\infty}}{[C(s)]^{-1}c_{\infty}} \\ &= C(s)\end{aligned}$$



设计策略：

严格正则的最小相位控制器可通过适当添加远离虚轴的最小相位零点变为双正则的形式。

- (1) 控制器的动态由对象的实际输入信号来驱动；
- (2) 由对象的实际输入信号驱动时，控制器的动态是稳定的。



◆ Anti-Windup的基本结构

假设控制器是双正则的最小相位系统，将其分解为

比例项和严格正则项：

$$C(s) = c_{\infty} + \bar{C}(s)$$

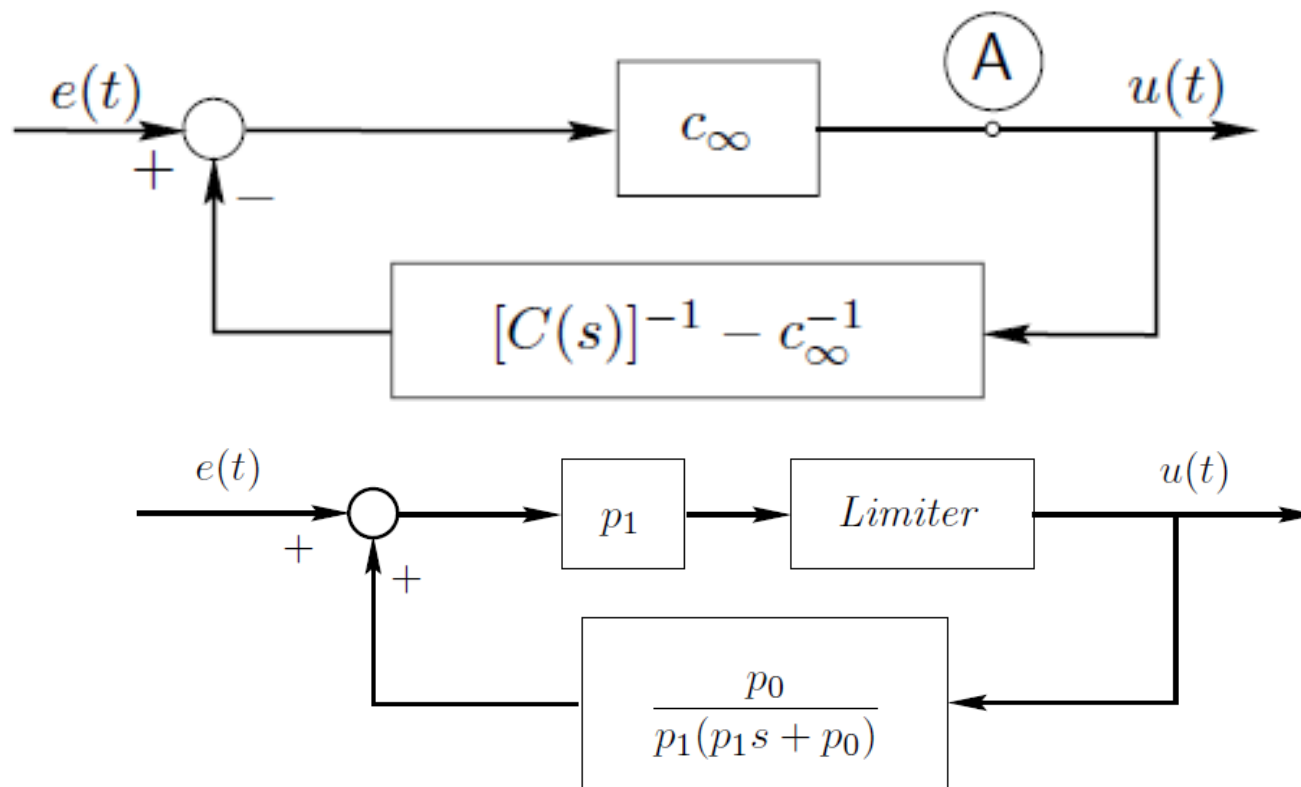
PI控制器：

$$C(s) = \frac{p_1 s + p_0}{s}, p_0 > 0, p_1 > 0$$

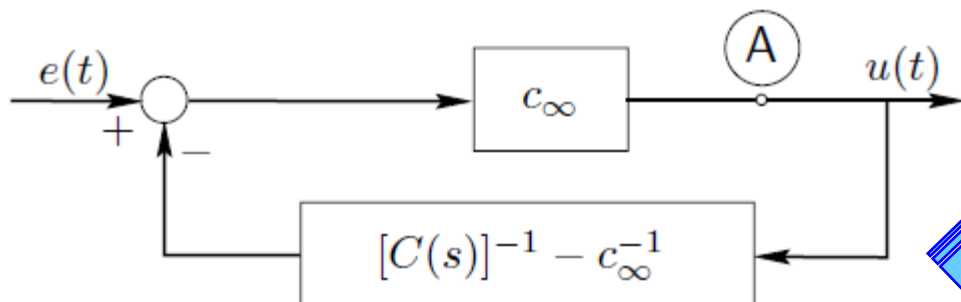
$$C(s) = p_1 + \frac{p_0}{s}, p_0 > 0, p_1 > 0$$

$$c_{\infty} = p_1, \bar{C}(s) = \frac{p_0}{s}$$

$$\begin{aligned} [C(s)]^{-1} - c_{\infty}^{-1} &= \frac{s}{p_1 s + p_0} - \frac{1}{p_1} \\ &= \frac{p_1 s - p_1 s - p_0}{p_1(p_1 s + p_0)} \\ &= \frac{-p_0}{p_1(p_1 s + p_0)} \end{aligned}$$

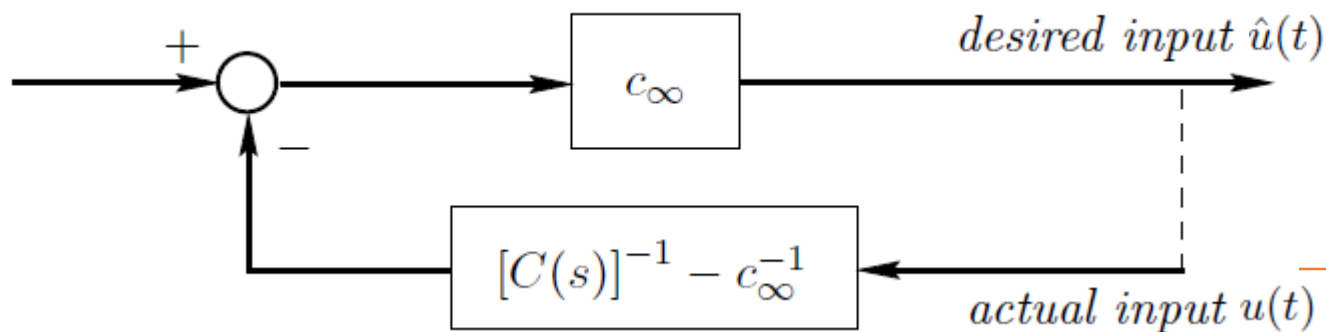


◆ Anti-Windup的基本结构



双正则控制器:

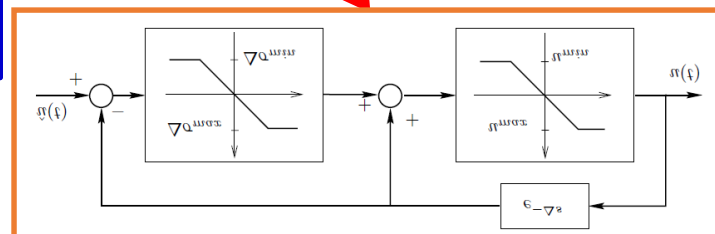
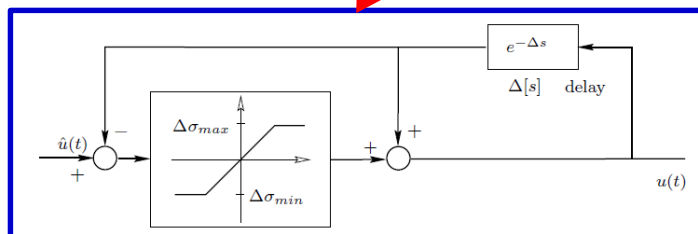
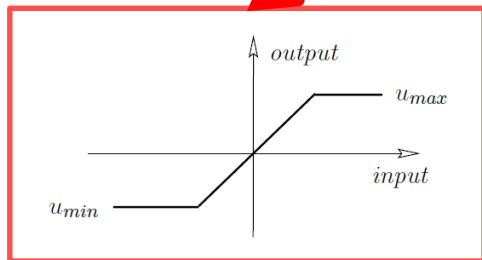
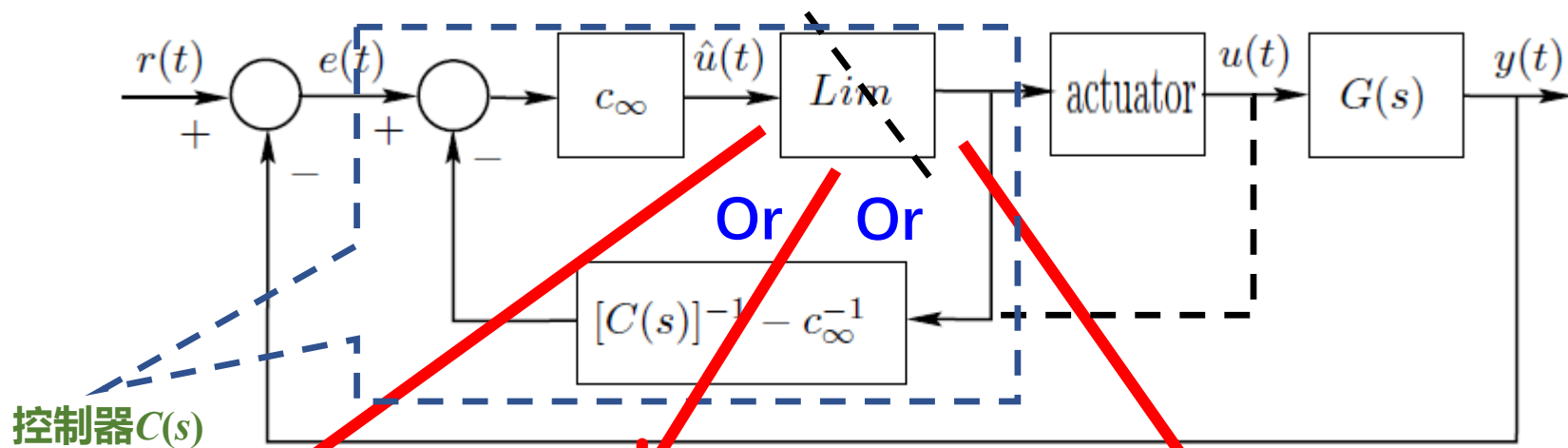
$$C(s) = c_\infty + \bar{C}(s)$$



设计策略:

- (1) 控制器的动态由对象的实际输入信号来驱动;
- (2) 由对象的实际输入信号驱动时, 控制器的动态是稳定的。

◆ Anti-Windup控制回路——含有执行器约束





5.2.1 Anti-Windup的设计策略



● 例2 (续) 积分器Anti-Windup设计

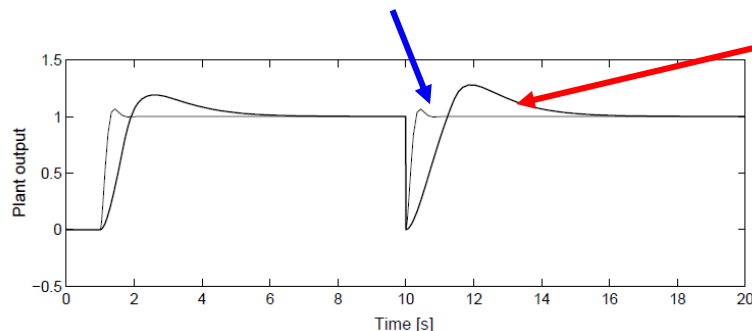
——含有执行器幅值约束

$$G_o(s) = \frac{2}{(s+1)(s+2)}$$

$$T_o(s) = \frac{100}{s^2 + 13s + 100}$$

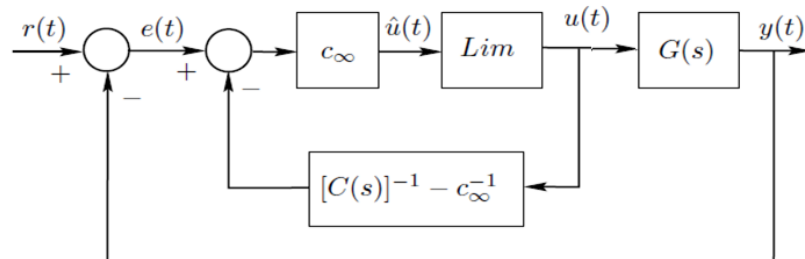
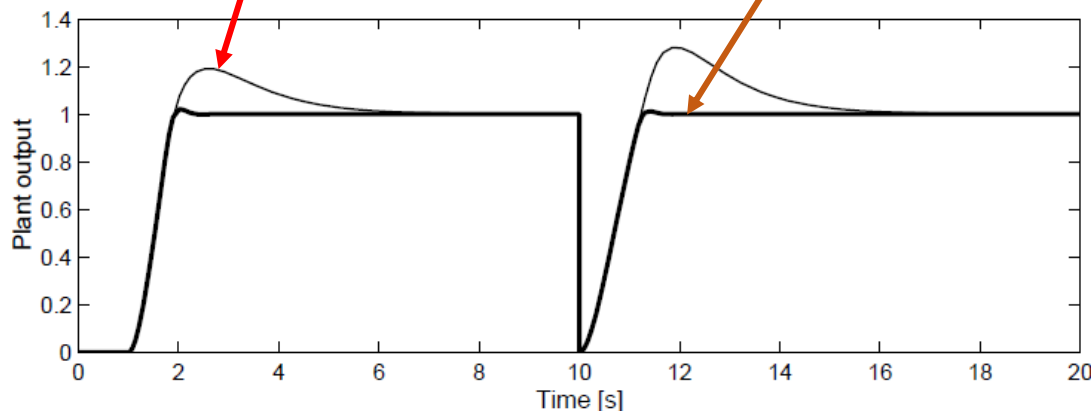
$$C(s) = \frac{50(s+1)(s+2)}{s(s+13)}$$

不存在饱和约束



存在饱和约束
非Anti-Windup设计

存在饱和约束
Anti-Windup设计



● 例3 Anti-Windup设计

——含有执行器转换速率约束

$$C_{PI}(s) = K_p \left(1 + \frac{1}{T_r s} \right)$$

$$K_P = 0.5 \quad T_r = 1.5[s]$$

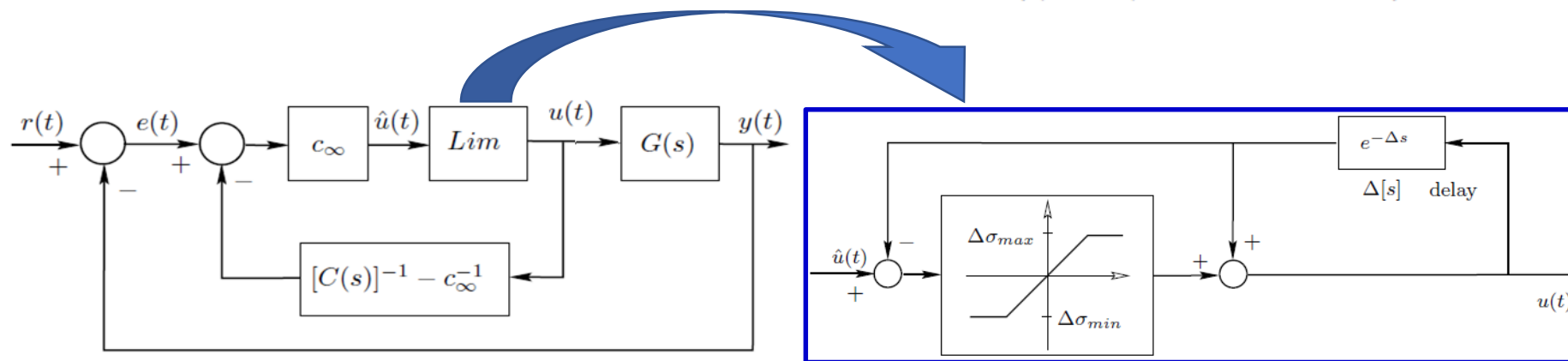


$$c_\infty = K_p = 0.5$$

$$[C(s)]^{-1} - c_\infty^{-1} = -\frac{1}{K_p(T_r s + 1)} = -\frac{2}{(1.5s + 1)}$$

假设执行器的转换速率小于等于 $0.2s^{-1}$ 。

$$Y(s) = e^{-s} \left(\frac{1}{(s + 1)^2} U(s) + D_g(s) \right)$$

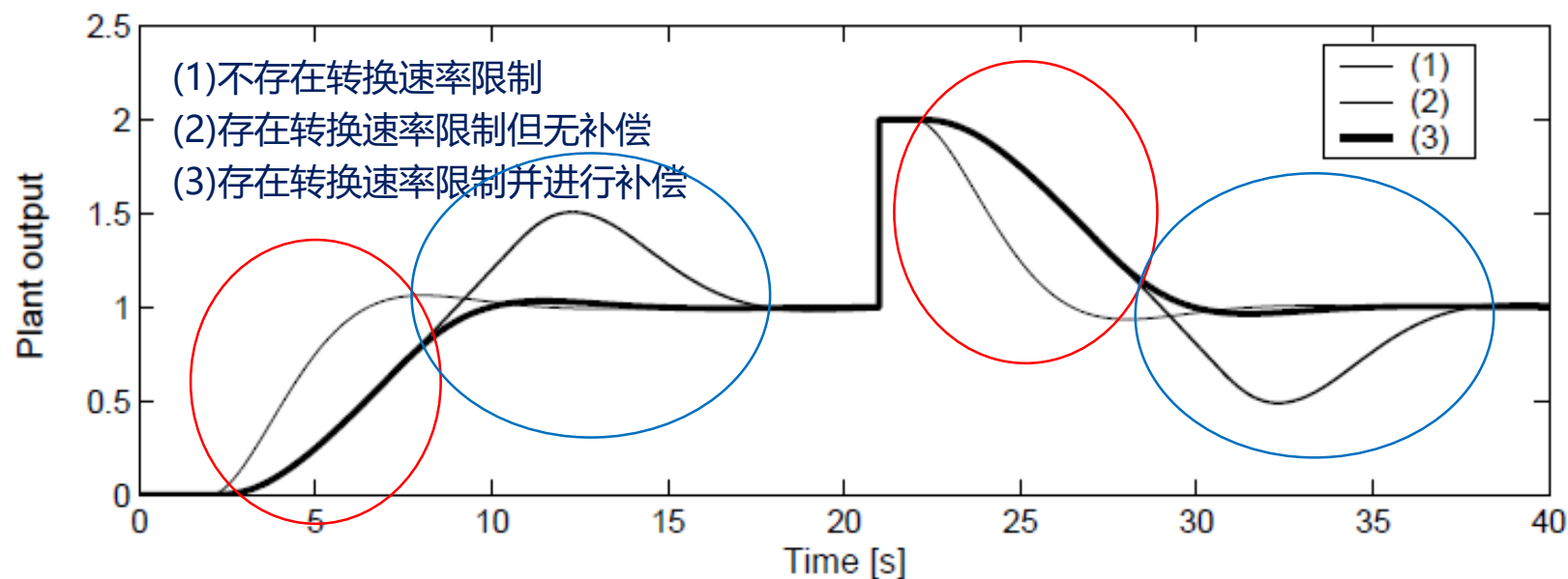


● 例3 Anti-Windup设计

——含有执行器转换速率约束

$$Y(s) = e^{-s} \left(\frac{1}{(s+1)^2} U(s) + D_g(s) \right) \quad c_\infty = K_p = 0.5 \quad [C(s)]^{-1} - c_\infty^{-1} = -\frac{2}{(1.5s+1)}$$

在1s时输入单位阶跃参考信号，在20s时存在单位阶跃扰动输入。

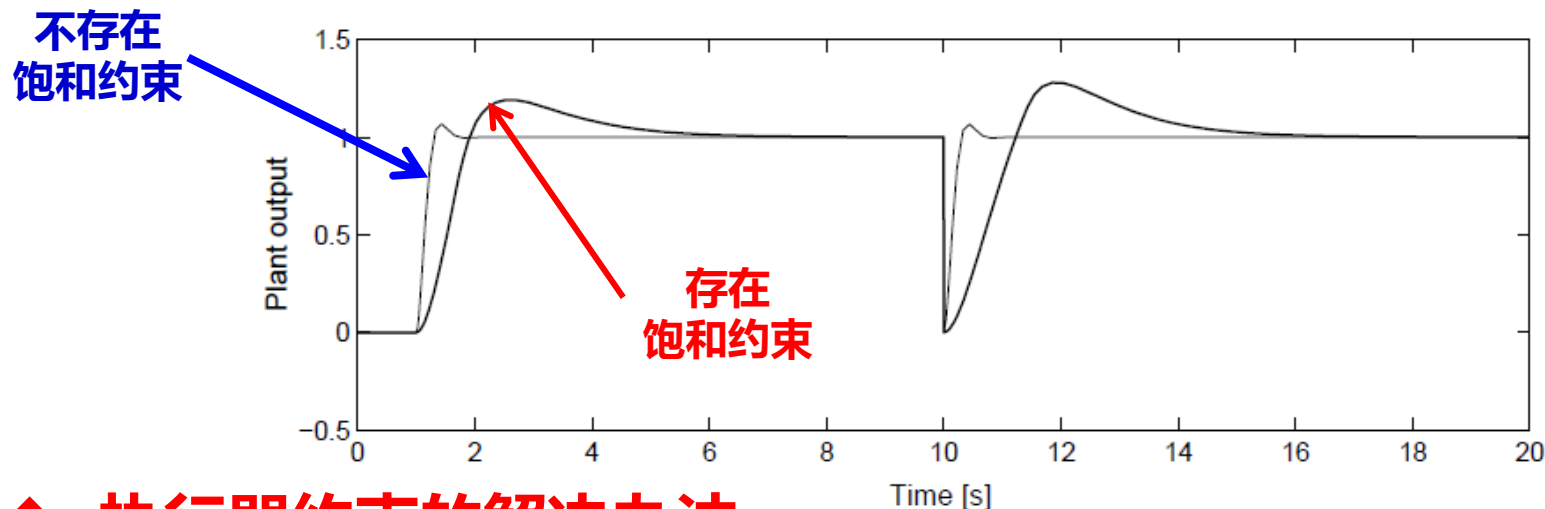




Thanks for your attention!

◆ 积分器的Windup问题

$$S_{uo}(s) \triangleq \frac{T_o(s)}{G_o(s)}$$



◆ 执行器约束的解决办法

- (1) 保证执行器不会超出范围
- (2) 修改设计，应对约束

$$G_o(s) = \frac{2}{(s+1)(s+2)}$$

$$C(s) = \frac{50(s+1)(s+2)}{s(s+13)}$$

当执行器达到约束的边界时切断积分作用！