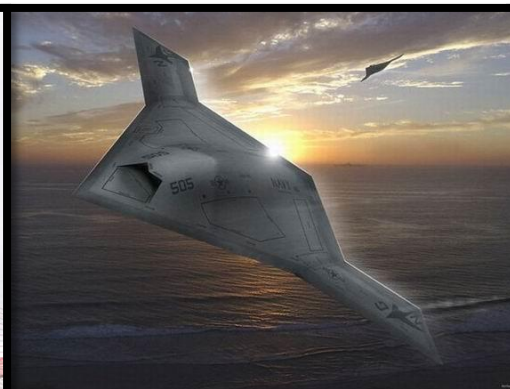


机器人智能控制

易建强 蒲志强 袁如意

中国科学院自动化研究所

2020年秋季



第十讲 自抗扰控制

蒲志强 高级工程师

zhiqiang.pu@ia.ac.cn



本讲的主要内容

一、基本介绍

二、跟踪微分器

三、非线性反馈

四、扩张状态观测器

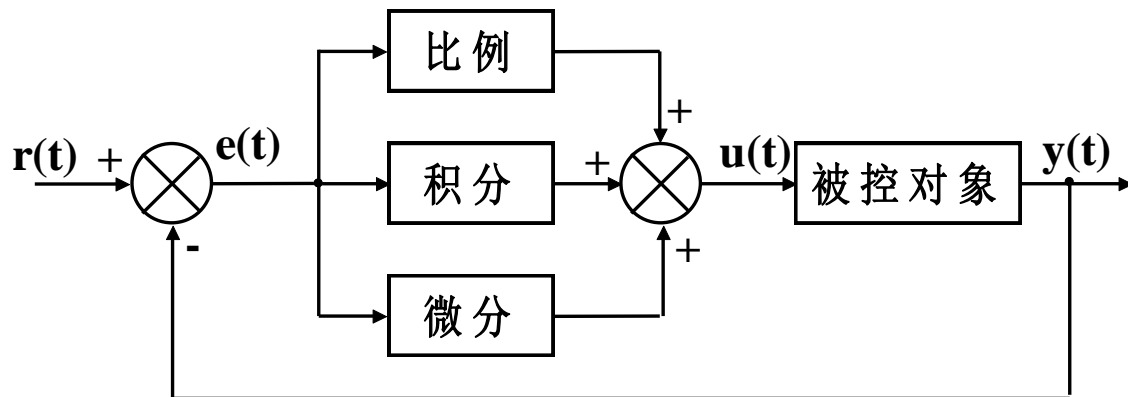
五、应用举例

六、一些思考

一、基本介绍

1. 自抗扰控制的产生
2. 自抗扰控制的基本结构
3. 自抗扰控制的应用

■ 从PID说起



$$u = K_P e + K_I \int e + K_D \frac{de}{dt}$$

PID的优势

- ★ 基于误差消除误差 (VS 模型论)
- ★ 原理简单，易于实现
- ★ 通用控制器

90%以上实际控制系统采用PID

□ PID的缺陷

请思考？

- ✓ 直接以 $e=v-y$ 的方式产生调节误差并不完全合理，可能产生“快速性”和“超调”间的矛盾。
- ✓ 产生误差信号的微分没有太好的办法，只能近似实现
- ✓ 具有好的动态品质的PID控制裕度不大，闭环动态品质对PID增益的变化敏感。
- ✓ 误差比例、积分、微分的线性组合并非最好方式
- ✓ 误差的积分对抑制常值扰动有效果，但对于其他广泛的扰动形式效果甚微。

- 例1

考虑被控对象

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = f(x_1, x_2, t) + w(t) + u \\ y = x_1 \end{cases}$$

其中， $f(x_1, x_2, t)$ 为未知加速度项， $w(t)$ 为外界干扰。仿真取

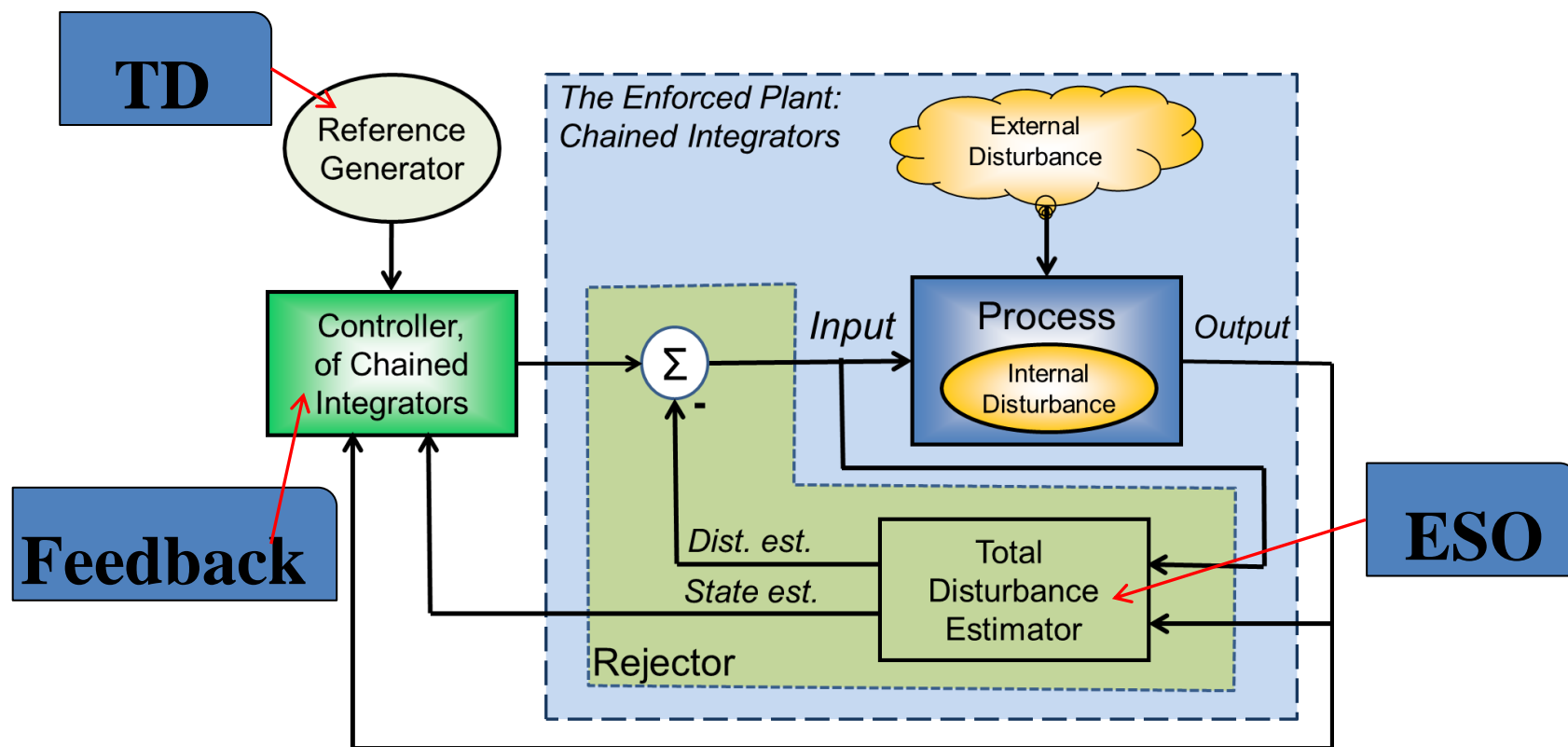
$$f(x_1, x_2, t) = \gamma_1 \cos(\omega_1 t) x_1 + \gamma_2 \cos(\omega_2 t) x_2 + \omega(t)$$

$$\gamma_1 = \gamma_2 = 1, \omega_1 = 0.6, \omega_2 = 0.7, \omega(t) = 1,$$

$$w(t) = 0.5 \text{sign}(\sin(\omega t))$$

初始时刻 $x_1(0)=x_2(0)=0$ ，试设计PID控制器，跟踪阶跃指令。

- 自抗扰控制 (Active Disturbance Rejection Control, ADRC)

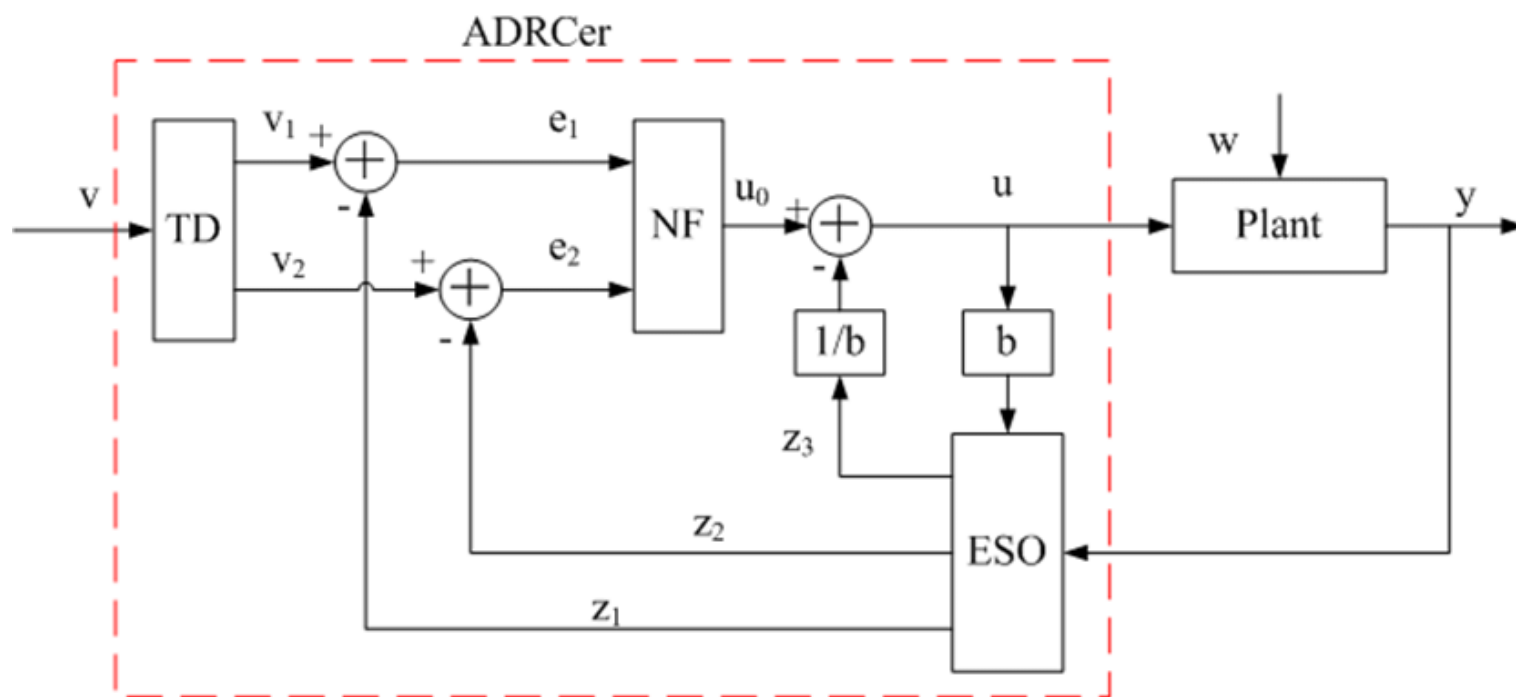


跟踪微分器 (Tracking Differentiator, TD)

非线性反馈 (Nonlinear Feedback, NF)

扩张状态观测器 (Extended State Observer, ESO)

- 典型二阶被控系统的例子



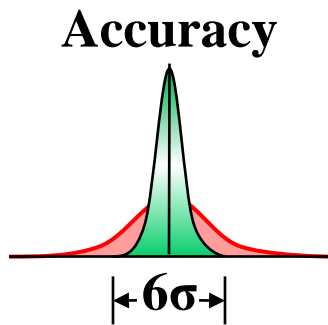
跟踪微分器 (Tracking Differentiator, TD)

非线性反馈 (Nonlinear Feedback, NF)

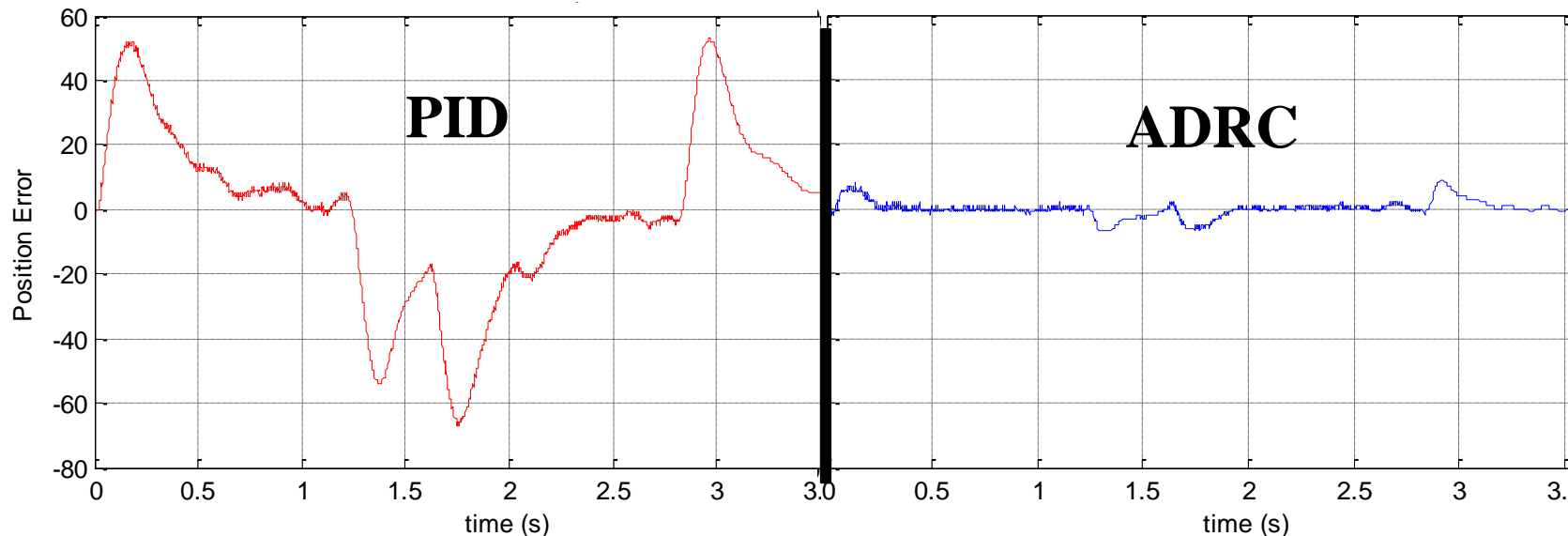
扩张状态观测器 (Extended State Observer, ESO)

工业伺服驱动器测试（北美财富100强企业，2009）

控制精度



测试方法是在原有平台中用ADRC算法替换PID算法。工程师花费45分钟来调节PID参数，相反他花费2分钟来调节ADRC参数。



位置误差减少81%

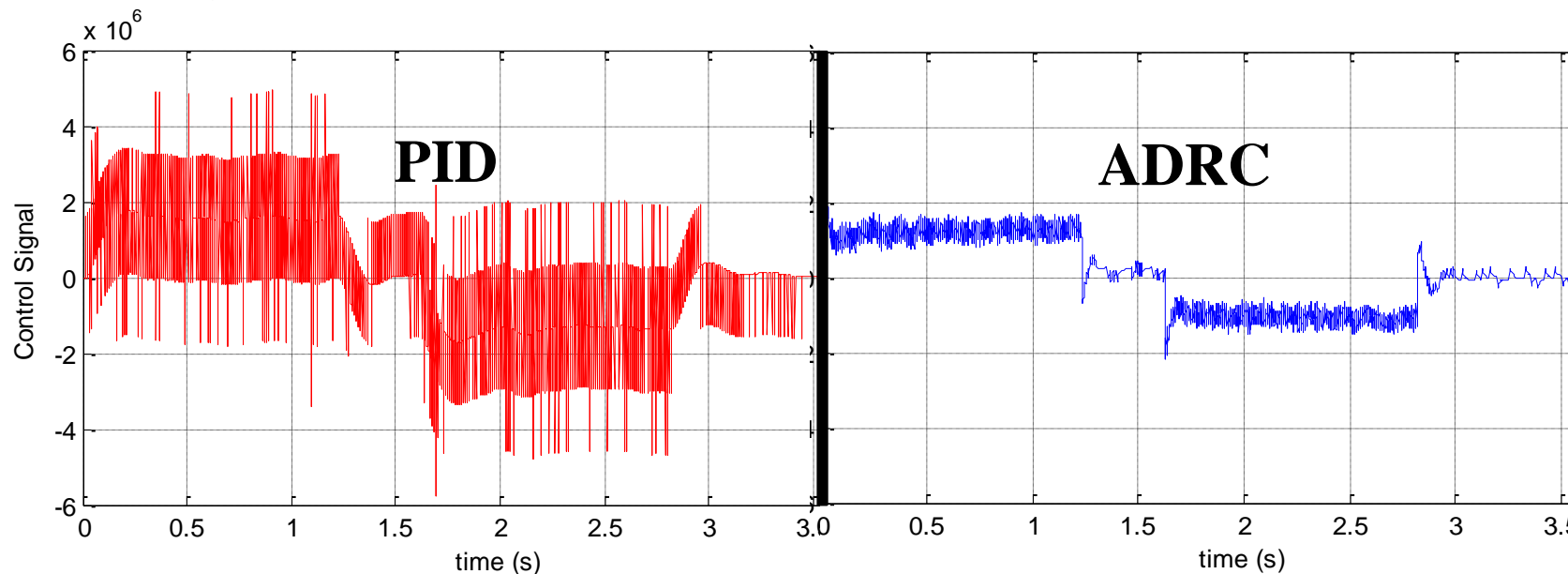
□ 工业伺服驱动器测试（北美财富100强企业，2009）

➤ 能量消耗

Energy Savings



通过最小化超调量、振动、响应延迟等达到节省能量的目的。



能量节省41%（力矩均方根）

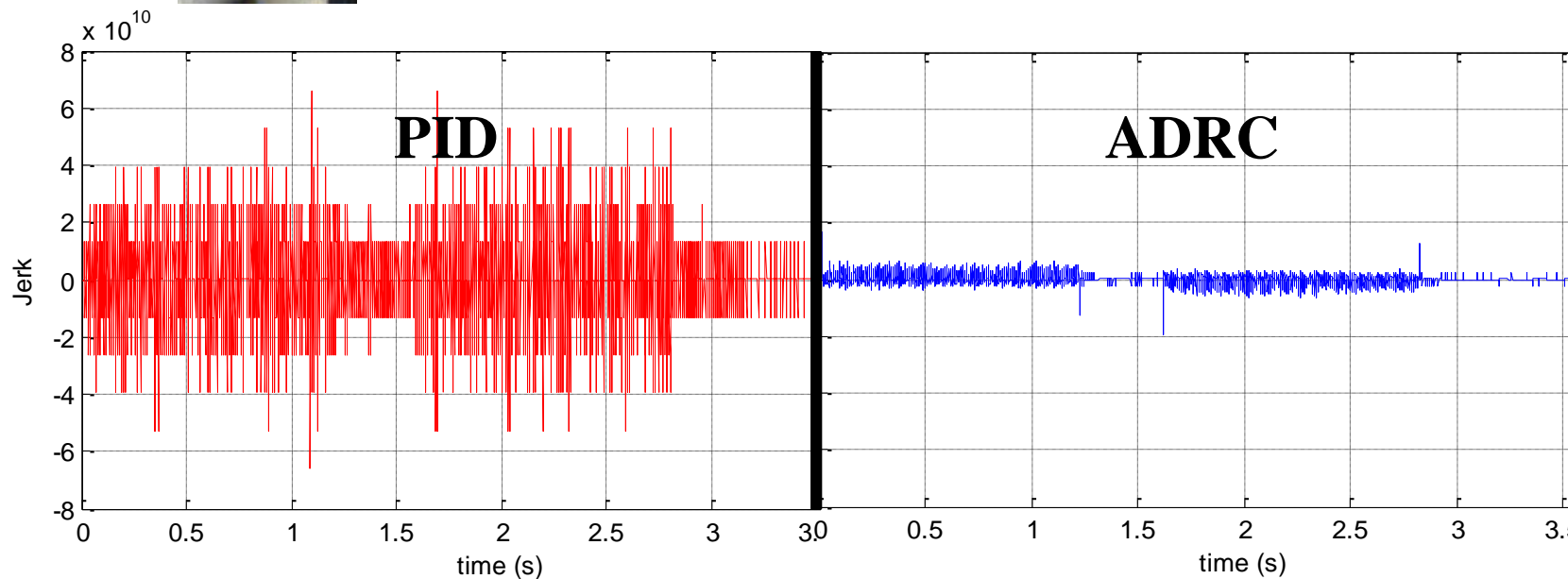
工业伺服驱动器测试（北美财富100强企业，2009）

机械寿命

Machine Life



减少机械磨损，缩短宕机时间，延长机械寿命。



颠簸程度（Jerk）减小71%

□ 工业挤压机测试（Parker, Ohio, 2010）

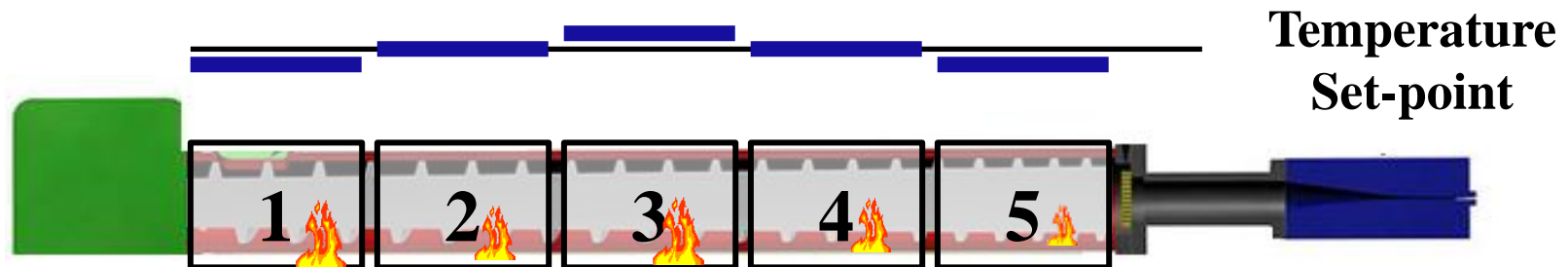


Implemented in PLC across ten extrusion lines, with an average of 57% energy savings

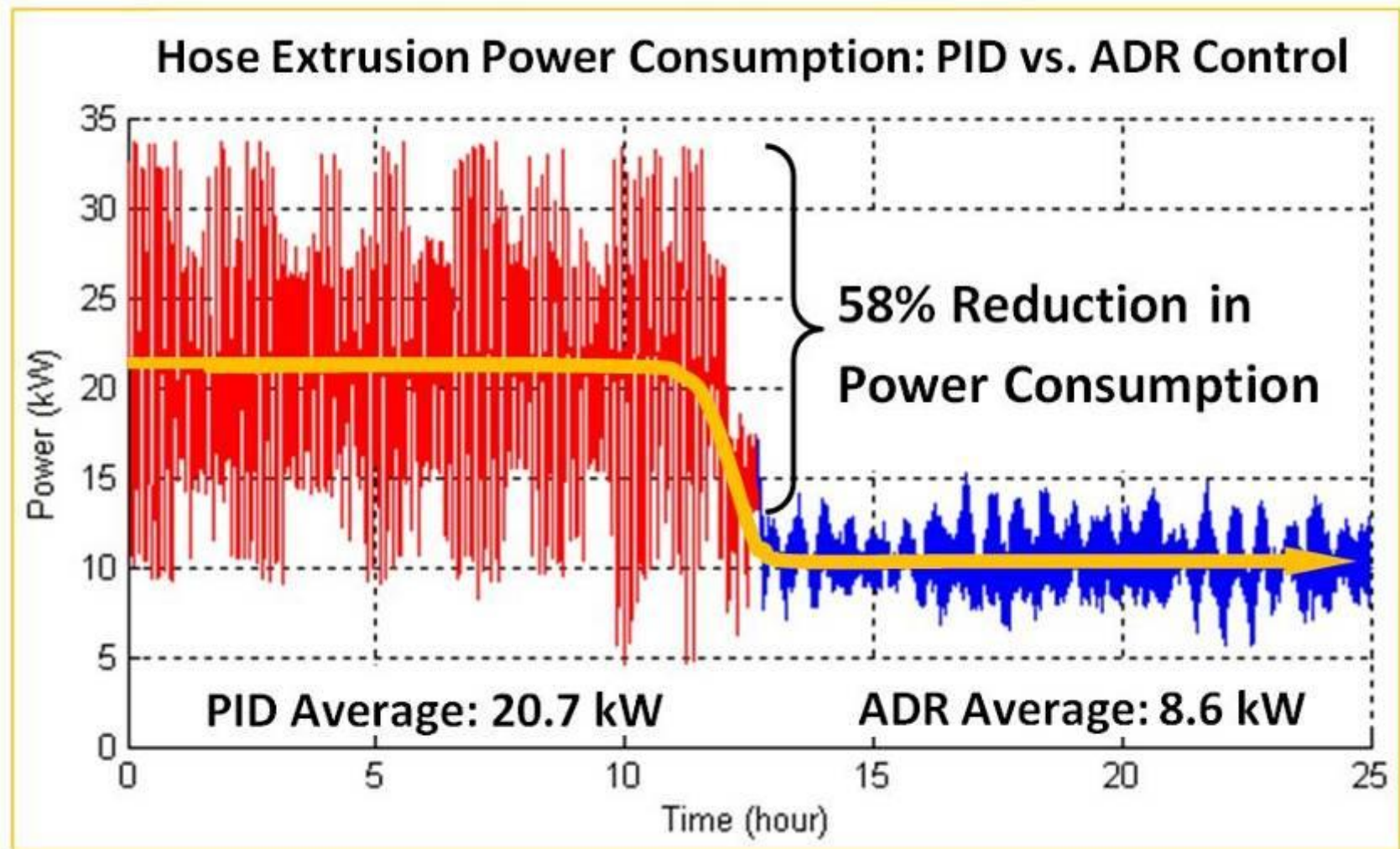
□ Process Control: Temperature Control



PID passively responds to disturbance, causing zone oscillation



□ Power Reduction by 58%!



□ 芯片级应用: Freescale Kinetis



Tools and Software

Kinetis Motor Suite

Freescale motor control development software for Kinetis MCUs based on ARM® Cortex® cores

Increased Efficiency

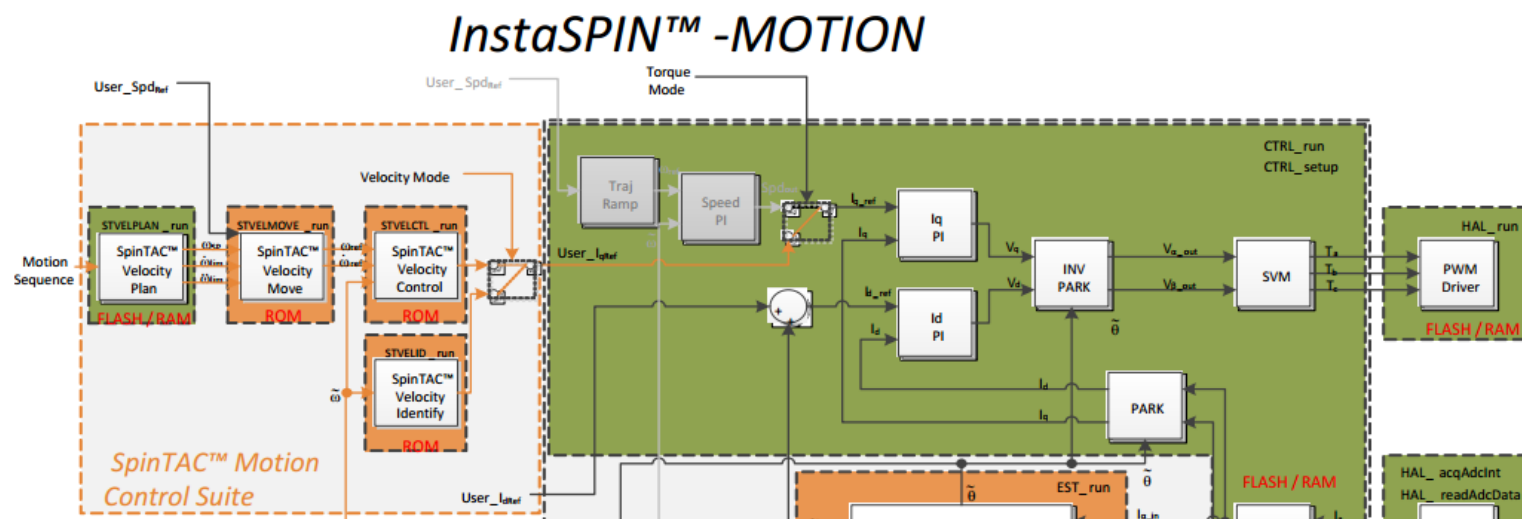
To increase your motor efficiency while further reducing time to market, the Kinetis motor suite streamlines your PID loop tuning. Traditionally, PID loop tuning is trial and error based and requires in-depth motor control expertise, resulting in non-optimal system performance that can degrade over time. Kinetis motor suite addresses these issues by including advanced Active Disturbance Rejection Control (ADRC) from LineStream Technologies™.



ACIM motors or sensored ACIM motors. Software is pre-programmed into the MCU at the factory and enables simplified tuning of control loops and regulation of estimation blocks. Kinetis motor observer provides features that regulate motor and inverter efficiency, power factor, current and torque ripple and noise.

- **Kinetis motor manager:** A graphical user interface that provides a real-time, non-intrusive display of system variables and MCU memory map. The Kinetis motor manager includes a software application

□ 芯片级应用: TI InstaSPIN-MOTION



CONTROL

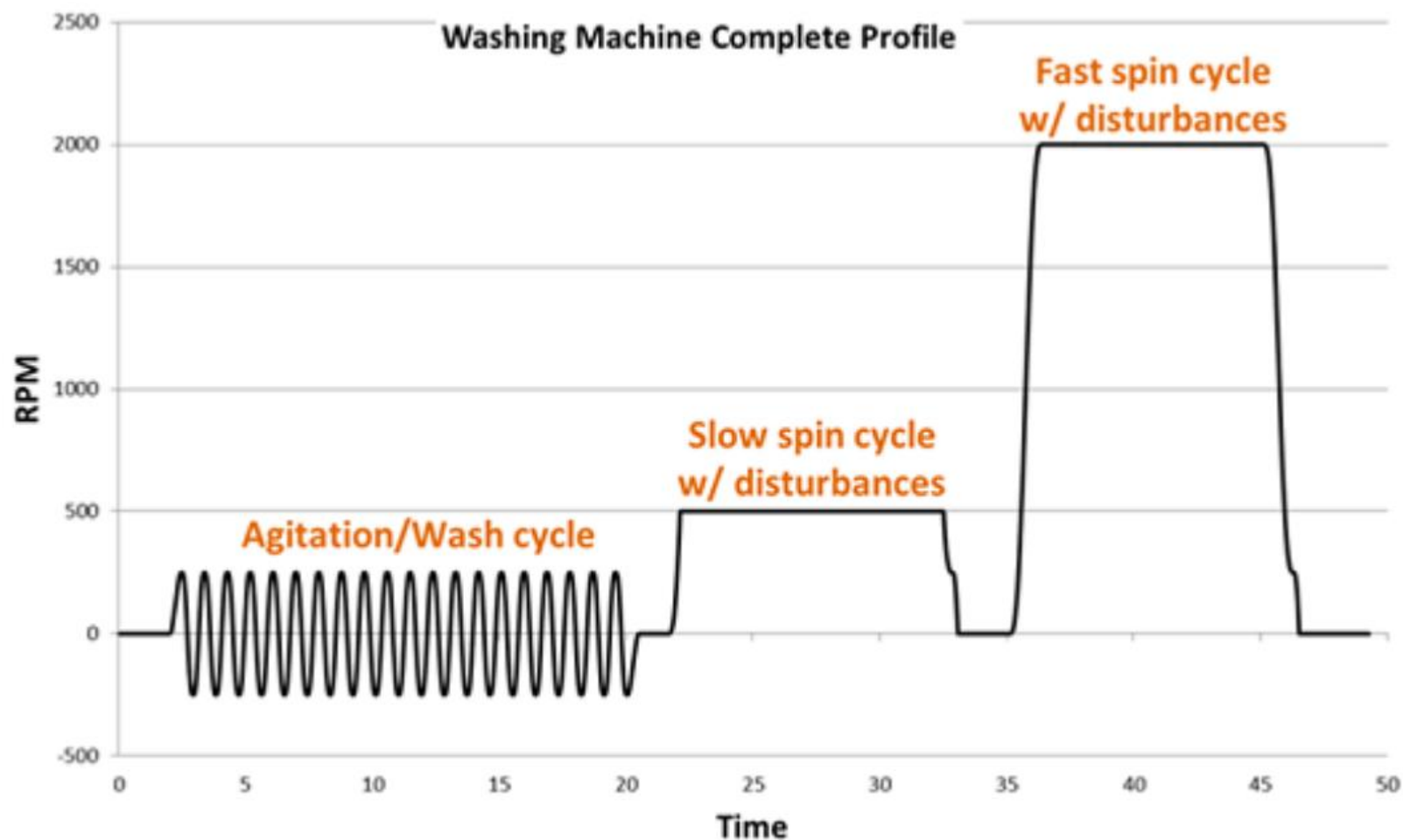
SpinTAC Control is an advanced speed and position controller featuring Active Disturbance Rejection Control (ADRC), which proactively estimates and compensates for system disturbance, in real-time. SpinTAC automatically compensates for undesired system behavior caused by:

- Uncertainties (for example, resonant mode)
- Nonlinear friction
- Changing loads
- Environmental changes.

SpinTAC Control presents better disturbance rejection and trajectory tracking performance than a PI controller, and can tolerate a wide range of inertia change. This means that SpinTAC improves accuracy and system performance, and minimizes mechanical system duress.

❑ 芯片级应用: TI InstaSPIN-MOTION

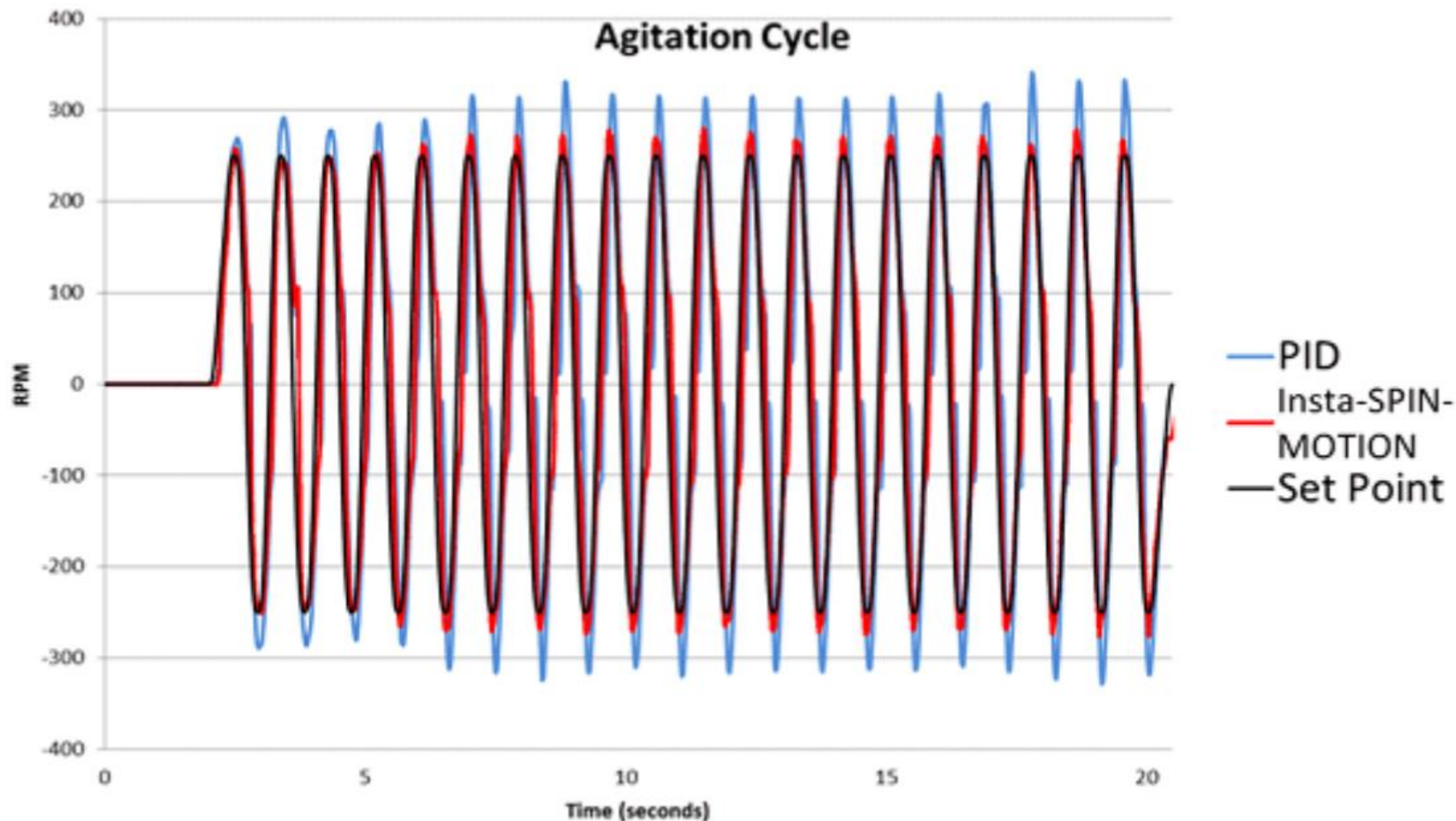
➤ 洗衣机控制



参考指令配置文件

❑ 芯片级应用: TI InstaSPIN-MOTION

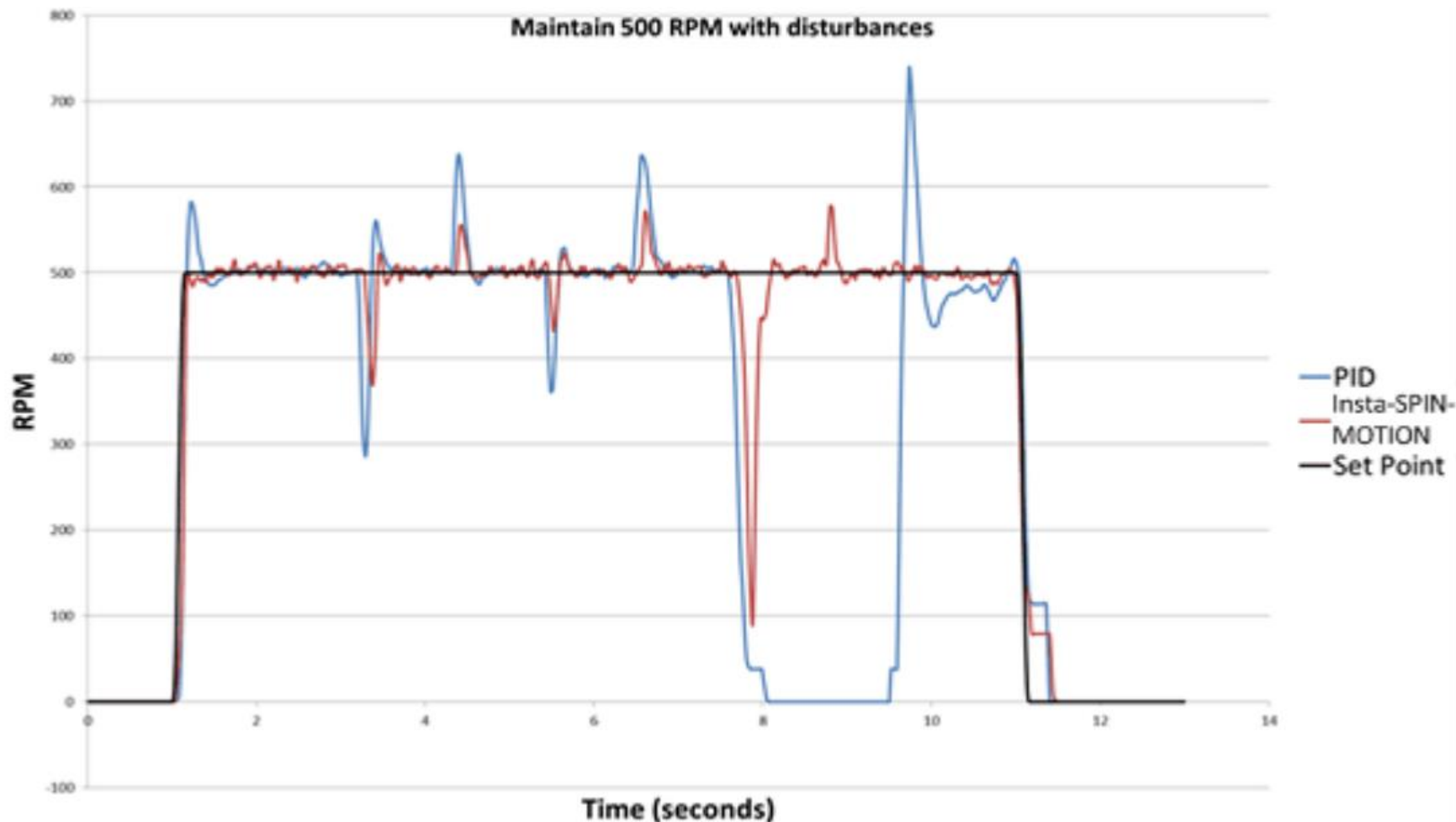
➤ 洗衣机控制



搅拌阶段—更高的控制精度（最大误差：PID 91rpm，ADRC 30rpm）

❑ 芯片级应用: TI InstaSPIN-MOTION

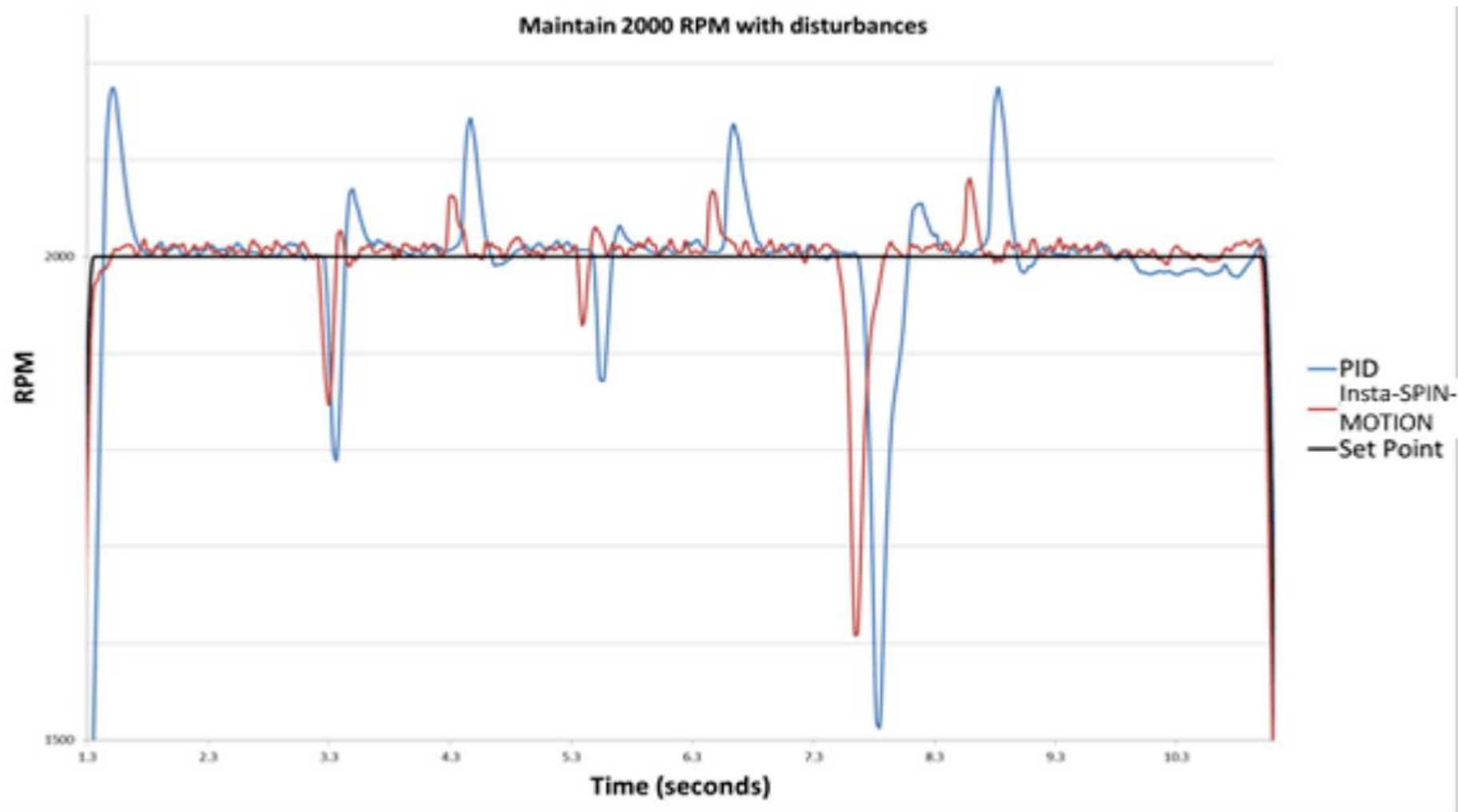
➤ 洗衣机控制



慢旋阶段—更小的超调和振荡，更大的抗扰能力

❑ 芯片级应用: TI InstaSPIN-MOTION

➤ 洗衣机控制



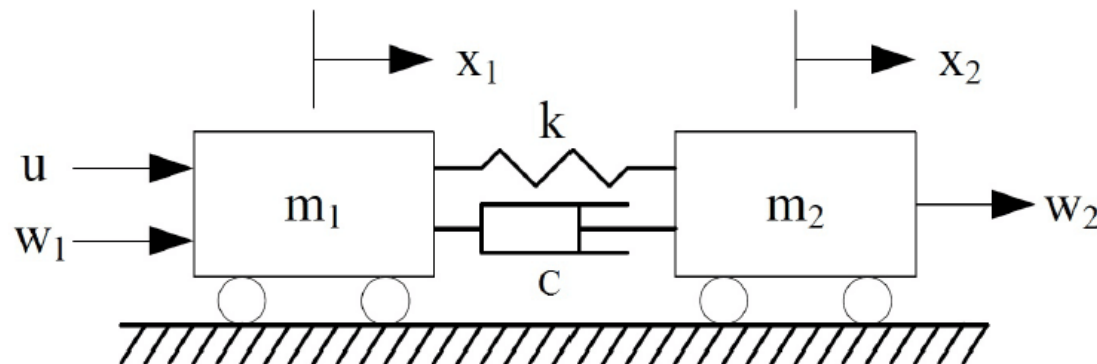
快旋阶段—更小的超调和振荡，更大的抗扰能力

□ Benchmark

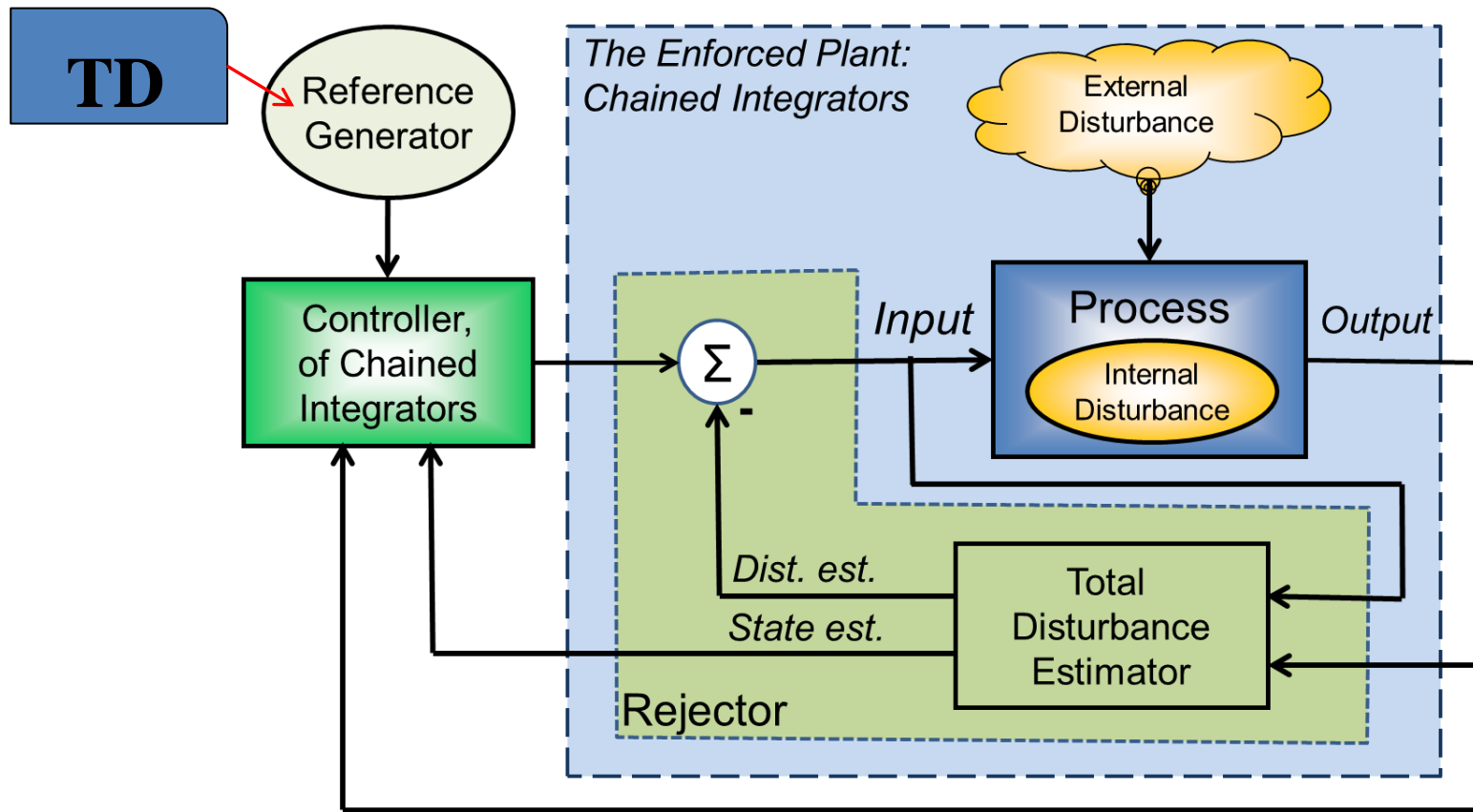
➤ Load disturbances

- PID – four loads
- ADRC – four loads
- PID – six loads
- ADRC – six loads

➤ Two-mass system

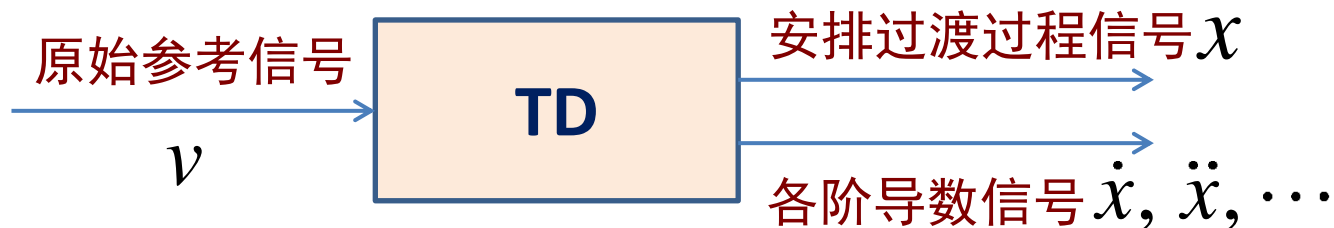


二、跟踪微分器



■跟踪微分器的作用？

- 合理产生微分信号
- 合理安排过渡过程
- 其他附带的好处



■如何产生具有因果关系的微分信号？

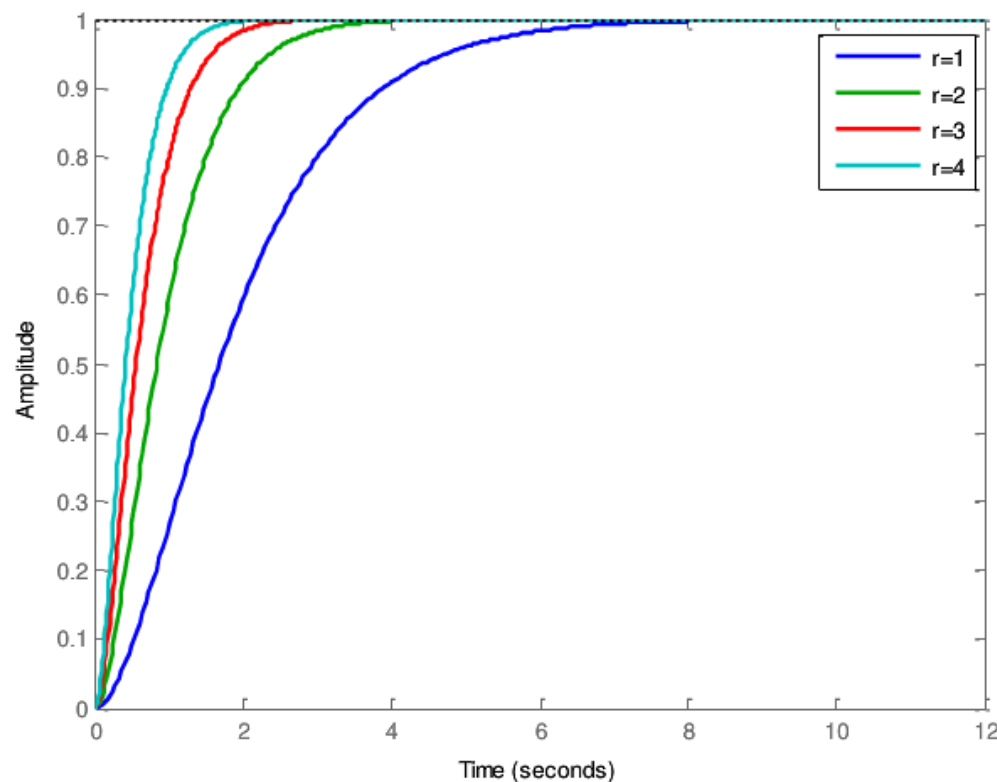
Casual differentiation

由

$$W(s) = \frac{r^2}{s^2 + 2rs + r^2}$$

可以无超调达到系统
设定值。

其中 r 为速度因子。



**思想：通过某环节尽可能快地跟踪输入信号，
同时给出近似微分信号**

■思考

既然目的是要让 v 尽快地跟踪上输入信号 r ，那么这本身就可以转化为一个控制问题：

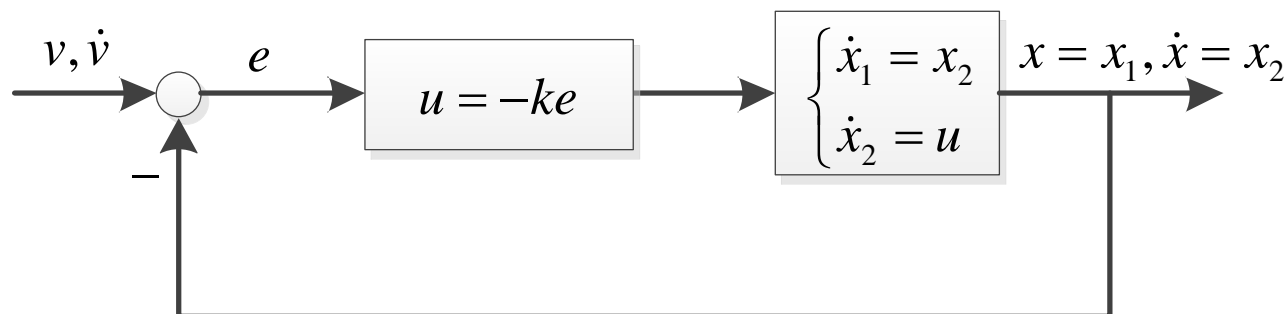
参考指令： v

输出： x 及各阶导数

被控对象是什么？

控制律是什么？

$$x^{(n)} = u$$



二阶对象为例的系统框图

□ 如何设计控制律，使跟踪微分器收敛？

令 $e = \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} = \begin{bmatrix} v - x_1 \\ \dot{v} - x_2 \end{bmatrix} \quad \begin{cases} \dot{e}_1 = e_2 \\ \dot{e}_2 = \ddot{v} - u \end{cases}$

假设参考指令为一个慢变（相对于跟踪微分器动态）过程，有 $\begin{cases} \dot{v} \approx 0 \\ \ddot{v} \approx 0 \end{cases}$

取 $u = k_1 e_1 + k_2 e_2$

有闭环系统 $\dot{e} = \begin{bmatrix} 0 & 1 \\ -k_1 & -k_2 \end{bmatrix} e = Ae$

特征方程为： $sI - A = 0 \Rightarrow s^2 + k_2 s + k_1 = 0$

收敛的充要条件： $k_1, k_2 > 0$

□ 微分方程实现

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = k_1(v - x_1) - k_2 x_2 \end{cases}$$

若取 $k_1 = r^2, k_2 = 2r$

得
$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = r^2(v - x_1) - 2rx_2 \end{cases}$$

对比
$$W(s) = \frac{r^2}{s^2 + 2rs + r^2}$$

所以，这只是其中一种特殊的实现形式而已，此时特征方程为

$$s^2 + 2rs + r^2 = (s + r)^2 = 0$$

系统有两根相等的负实根（无超调）

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = r^2(v - x_1) - 2rx_2 \end{cases}$$

□ 离散实现

$$\begin{cases} x_1(k+1) = x_1(k) + hx_2(k) \\ x_2(k+1) = x_2(k) + h \left[r^2(v(k) - x_1(k)) - 2rx_2(k) \right] \end{cases}$$

上面的实现形式只是给出了一类特殊的跟踪微分器设计方法（控制律为线性，且为无超调跟踪），但不是唯一的！

若设计为最速控制律，则可实现输出最快地跟踪输入信号。

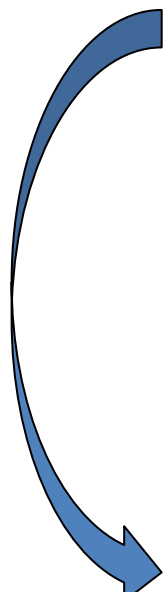
□ 最速控制实现

针对二阶积分器串联对象

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = u, |u| \leq r \end{cases}$$

以原点为终点的快速最优控制综合函数为

$$u(x_1, x_2) = -r \operatorname{sign} \left(x_1 + \frac{x_2 |x_2|}{2r} \right)$$


$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -r \operatorname{sign} \left(x_1 - v + \frac{x_2 |x_2|}{2r} \right) \end{cases}$$

连续形式

□ 最速控制实现

离散形式
$$\begin{cases} fh = fhan(x_1(k) - u(k), x_2(k), r, h_0) & h_0 = nh \\ x_1(k+1) = x_1(k) + hx_2(k) & \text{快速因子 } r \\ x_2(k+2) = x_2(k) + fh & \text{滤波因子 } h_0 \end{cases}$$

$$u = fhan(x_1, x_2, r, h)$$

$$\begin{cases} d = rh \\ d_0 = hd \\ y = x_1 + hx_2 \\ a_0 = \sqrt{d^2 + 8r|y|} \\ a = \begin{cases} x_2 + \frac{(a_0 - d)}{2} \text{sign}(y) & , |y| > d_0 \\ x_2 + \frac{y}{h} & , |y| \leq d_0 \end{cases} \\ fhan = - \begin{cases} r \text{sign}(a), |a| > d \\ r \frac{a}{d} & , |a| \leq d \end{cases} \end{cases}$$



$$fsg(x, d) = (\text{sign}(x + d) - \text{sign}(x - d)) / 2$$

$$\begin{cases} d = rh^2 \\ a_0 = hx_2 \\ y = x_1 + a_0 \\ a_1 = \sqrt{d(d + 8|y|)} \\ a_2 = a_0 + \text{sign}(y)(a_1 - d) / 2 \\ a = (a_0 + y) fsg(y, d) + a_2(1 - fsg(y, d)) \\ fhan = -r \left(\frac{a}{d} \right) fsg(a, d) - r \text{sign}(a)(1 - fsg(a, d)) \end{cases}$$

■ 为什么要安排过渡过程？

直接以 $e=v-y$ 的方式产生调节误差并不完全合理，可能产生“快速性”和“超调”间的矛盾。

PID控制的精髓是基于误差消除误差。在初始时刻误差很大，很容易产生系统超调，因此有必要为参考输入“安排”一个合理的过渡过程。

Initial Peaking

举例：快速性和超调间的矛盾



■ 如何实现安排过渡过程？

解决思路：降低系统初始误差，则可以在不改变系统阻尼的情况下用较大的增益来加快过渡过程而又不引起过大的超调。

函数发生器 (Function Generator)

跟踪微分器 (Tracking Differentiator)

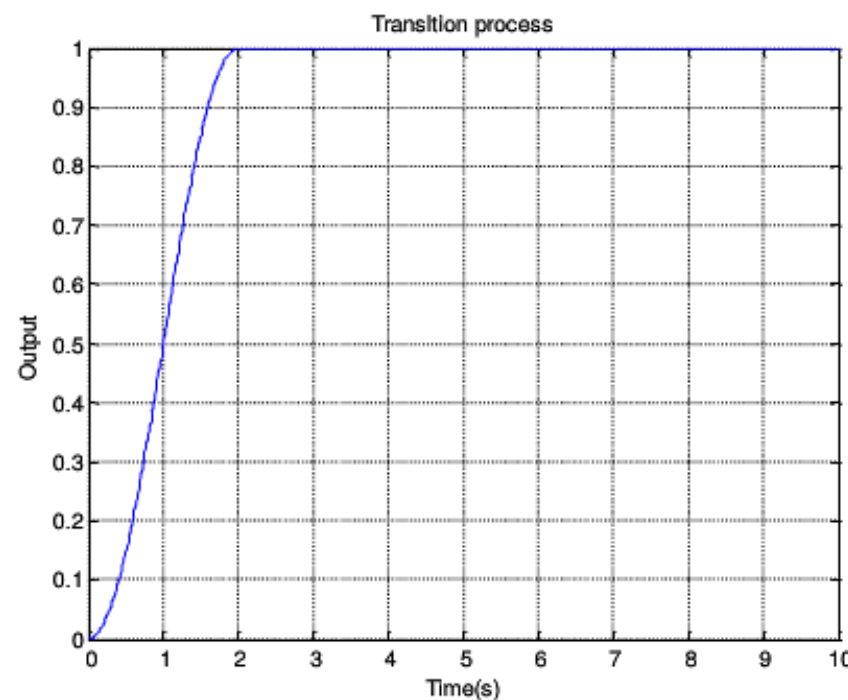
□ 函数发生器1：正弦

$$trns(T_0, t) = \begin{cases} \frac{1}{2} \left\{ 1 + \sin \left[\pi \left(\frac{t}{T_0} - \frac{1}{2} \right) \right] \right\}, & t \leq T_0 \\ 1, & t > T_0 \end{cases}$$
$$dtrns(T_0, t) = \begin{cases} \frac{\pi}{2T_0} \cos \left(\pi \left(\frac{t}{T_0} - \frac{1}{2} \right) \right), & t \leq T_0 \\ 0, & t > T_0 \end{cases}$$

$trns(T_0, t)$ 为安排的过渡过程

$dtrns(T_0, t)$ 为其微分

T_0 为过渡时间



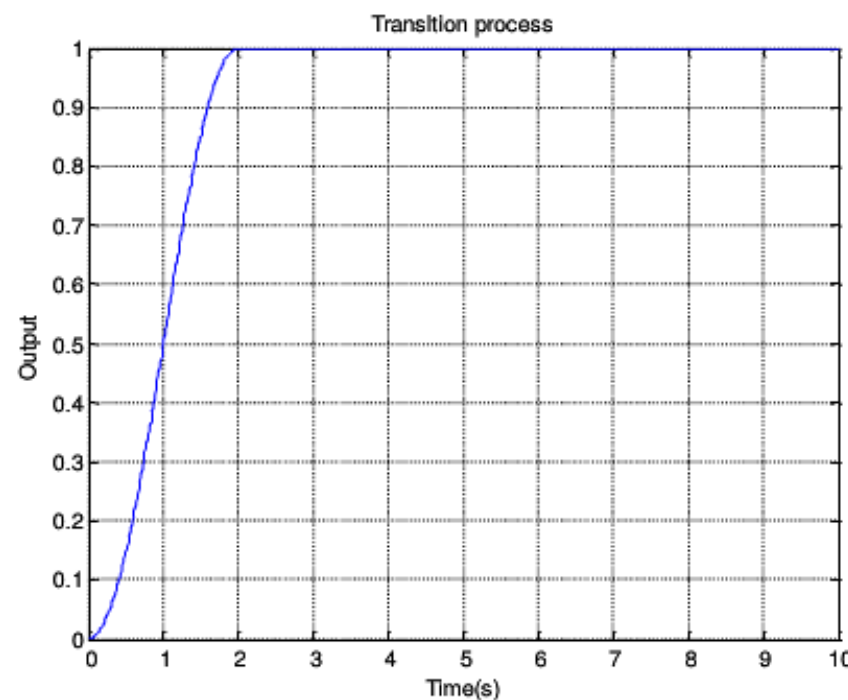
□ 函数发生器1：正弦

$$trns(T_0, t) = \begin{cases} \frac{1}{2} \left\{ 1 + \sin \left[\pi \left(\frac{t}{T_0} - \frac{1}{2} \right) \right] \right\}, & t \leq T_0 \\ 1, & t > T_0 \end{cases}$$
$$dtrns(T_0, t) = \begin{cases} \frac{\pi}{2T_0} \cos \left(\pi \left(\frac{t}{T_0} - \frac{1}{2} \right) \right), & t \leq T_0 \\ 0, & t > T_0 \end{cases}$$

$trns(T_0, t)$ 为安排的过渡过程

$dtrns(T_0, t)$ 为其微分

T_0 为过渡时间



□ 函数发生器2：加减速函数

$$r(t) = \begin{cases} 2\frac{r_0}{T^2}t^2, & t \leq \frac{T}{2} \\ r_0(-2\frac{t^2}{T^2} + 4\frac{t}{T} - 1), & \frac{T}{2} < t \leq T \\ r_0, & t > T \end{cases}$$

$r(t)$ 为安排的过渡过程

易求得其速度和加速度

T 为过渡时间

前半程加速、后半程减速，符合一般物理规律

□ 跟踪微分器

$$W(s) = \frac{r^2}{s^2 + 2rs + r^2}$$

$$\begin{cases} fh = fhan(x_1(k) - u(k), x_2(k), r, h_0) & h_0 = nh \\ x_1(k+1) = x_1(k) + hx_2(k) \\ x_2(k+2) = x_2(k) + hfh \end{cases}$$

- 高阶形式

$$W(s) = \frac{r^n}{(s + r)^n}$$

原则：根据被控对象的响应特性合理选择阶次

□ 配置系统零点

本质：求取系统的逆，用TD求微分

原系统

$$\ddot{y} - f(y, \dot{y}) = u$$

逆系统

$$u = \ddot{v} - f(v, \dot{v})$$

优势：传统的微分器对噪声很敏感，TD对噪声有很好的滤波作用

□ 求取函数极值


本质：用TD求微分

对于函数 $y = f(x)$

求极值的典型数值算法为 $x_{k+1} = x_k - \lambda \left. \frac{dy}{dx} \right|_{x=x_k}, \lambda > 0$

假设：函数的表达式未知，只能实时获取输入-输出量，无法直接获取导数信息，则不能直接利用上述迭代公式。

$$\frac{dy}{dx} = \frac{dy}{dt} / \frac{dx}{dt} = \frac{\dot{y}}{\dot{x}}$$

$$x(k+1) = x(k) - \lambda \frac{y_2(k)}{x_2(k)}$$


$$\begin{cases} x_1(k+1) = x_1(k) + hx_2(k) \\ x_2(k+1) = x_2(k) + hfhan(x_1(k) - x(k), x_2(k), r, h) \end{cases}$$

□ 数字整流

本质：用TD跟踪期望信号，跟踪速度可调

原信号

$$y = a \sin(50 \times 2\pi t)$$

绝对值信号

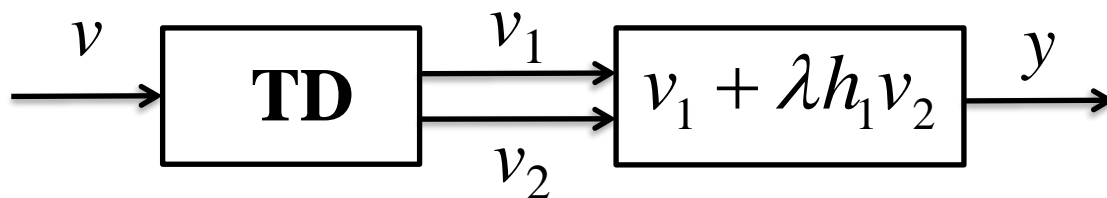
$$|y| = a |\sin(50 \times 2\pi t)|$$

将绝对值信号送入TD，适当调整TD的速度因子和滤波因子，就能得到很好的直流信号

□ 相位超前功能

本质：TD跟踪期望信号，TD产生微分信号

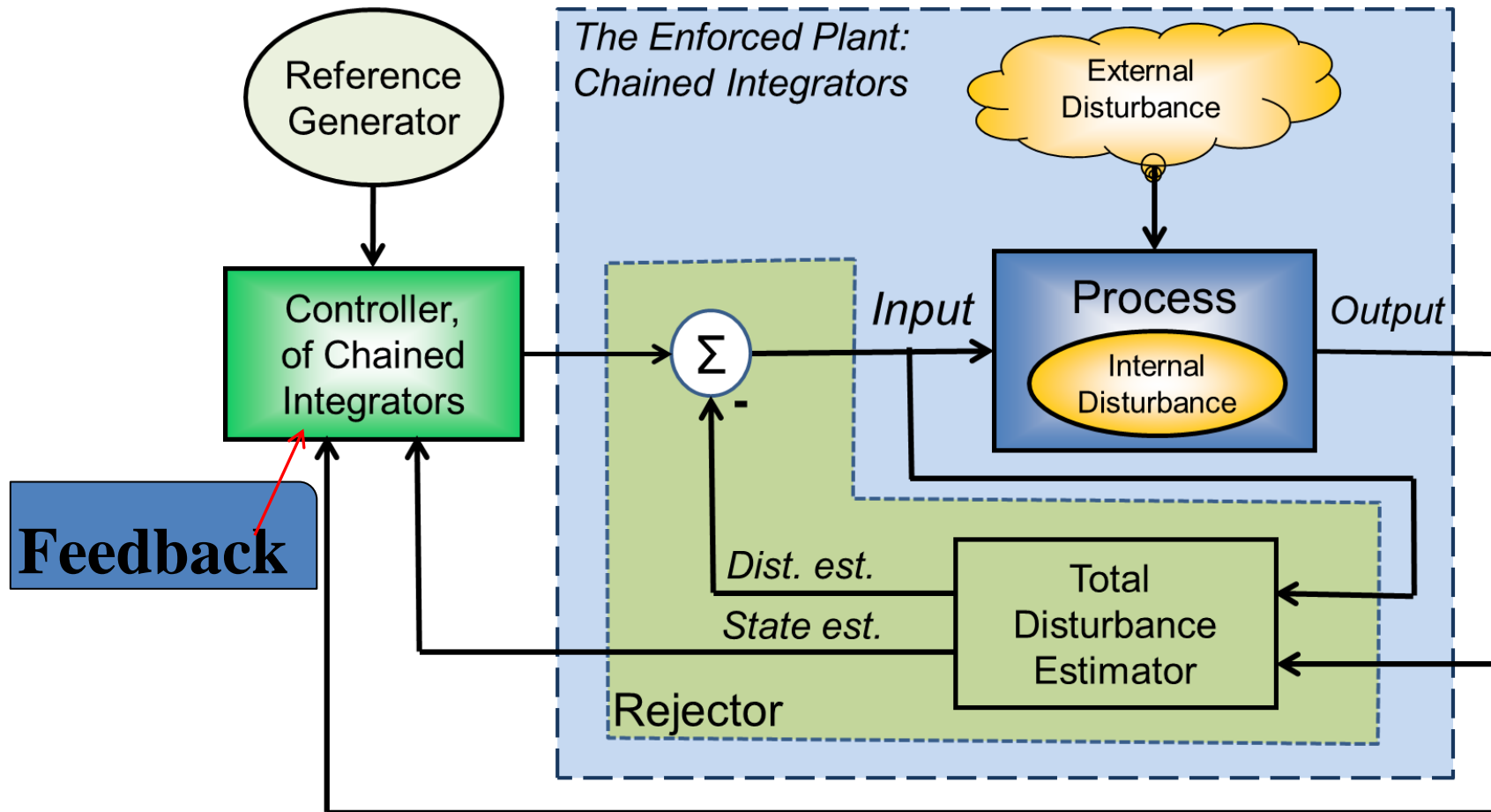
实现框图



$$\begin{cases} v_1(k+1) = v_1(k) + hv_2(k) \\ v_2(k+1) = v_2(k) + hfhan(v_1(k) - v(k), v_2(k), r, h) \end{cases}$$

$$y(k) = \frac{1}{\gamma} (v_1(k) + \lambda h_1 v_2(k)), \quad \gamma > 1$$

三、非线性反馈



■ 回到PID

经典PID控制律

$$u = K_P e + K_I \int e + K_D \frac{de}{dt}$$

PID：误差的线性组合

有没有更好的形式？

■ 非线性反馈

误差的非线性组合？

非线性反馈拥有比线性反馈更高的效率

$$e_1 = e, e_2 = \int e, e_3 = \frac{de}{dt}$$

$$fal(e, \alpha, \delta) = \begin{cases} |e|^\alpha \operatorname{sign}(e), & |e| > \delta \\ \frac{e}{\delta^{1-\alpha}}, & |e| \leq \delta \end{cases}$$

$$u = \beta_1 fal(e_1, \alpha_1, \delta) + \beta_2 fal(e_2, \alpha_2, \delta) + \beta_3 fal(e_3, \alpha_3, \delta)$$

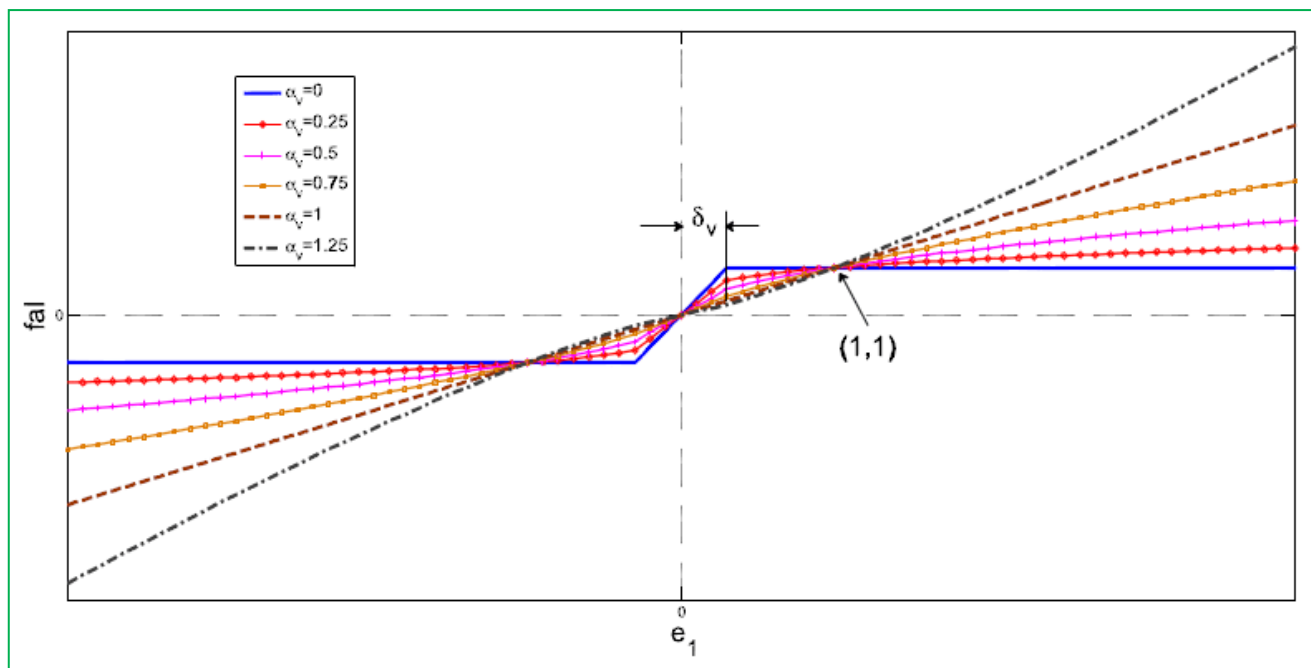
$$\alpha_0 < 0 < \alpha_1 < 1 < \alpha_2, \text{ or } 0 < \alpha_0 < \alpha_1 < 1 < \alpha_2$$

■ 非线性函数（举例）

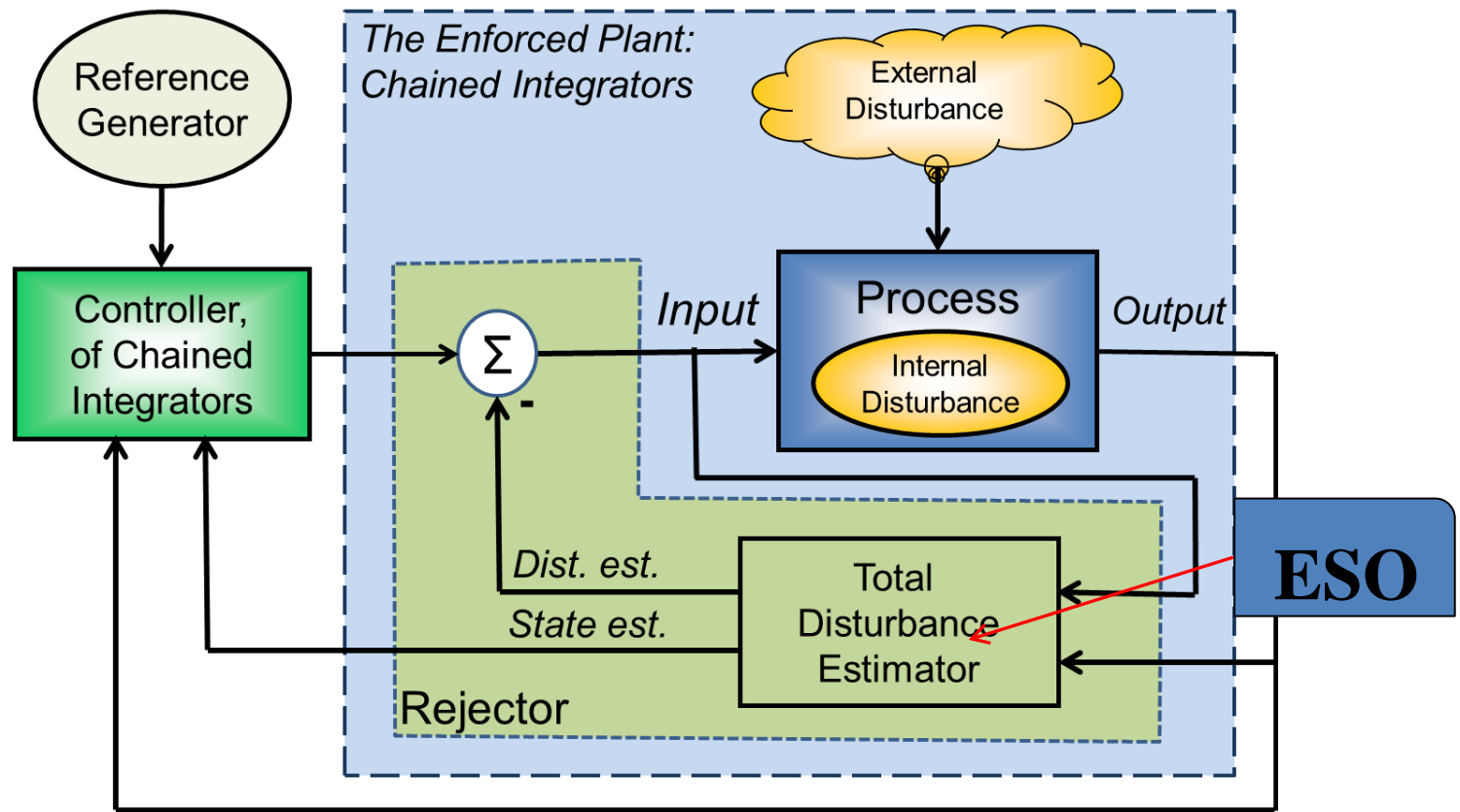
$$fal(e, \alpha, \delta) = \begin{cases} |e|^\alpha \operatorname{sign}(e), & |e| > \delta \\ \frac{e}{\delta^{1-\alpha}}, & |e| \leq \delta \end{cases}$$

大误差小增益

小误差大增益



四、扩张状态观测器



■ 什么是状态观测器？

根据测量到的系统输入（控制量）和系统输出（测量量）来确定系统部分或所有内部状态信息的装置，就是状态观测器。

举例：无人机

控制量：舵偏角、油门开度

测量量：欧拉角/角速率、位置、速度

估计量：气流角

■ 观测器设计举例

对于二阶非线性系统

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = f(x_1, x_2, t) + bu \\ y = x_1 \end{cases}$$

当函数 f 和 b 已知时，可以建立如下状态观测器

$$\begin{cases} e_1 = \hat{x}_1 - y \\ \dot{\hat{x}}_1 = \hat{x}_2 - l_1 e_1 \\ \dot{\hat{x}}_2 = f(\hat{x}_1, \hat{x}_2, t) + bu - l_2 e_1 \end{cases}$$

合理选择观测器增益 l_1 和 l_2 ，则观测器可重构出系统状态。

■ 观测器设计举例

对于上述问题的两点疑问

$$\begin{cases} e_1 = \hat{x}_1 - y \\ \dot{\hat{x}}_1 = \hat{x}_2 - l_1 e_1 \\ \dot{\hat{x}}_2 = f(\hat{x}_1, \hat{x}_2, t) + bu - l_2 e_1 \end{cases}$$

1、 f 未知或系统有干扰时怎么办？

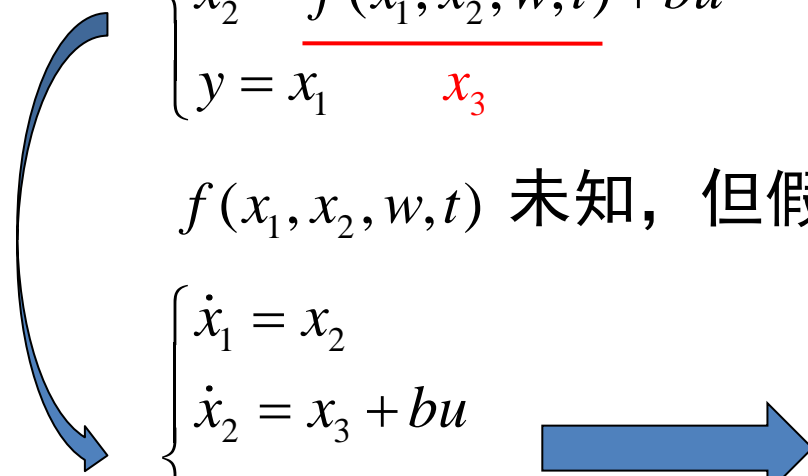
2、 $l_1 e_1$ 和 $l_2 e_1$ 为线性形式，能否替换为非线性形式？

■ 扩张状态观测器

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = \underline{f(x_1, x_2, w, t) + bu} \\ y = x_1 \end{cases}$$

扩张状态

$f(x_1, x_2, w, t)$ 未知，但假设 $\dot{f} = h(t)$



A curved blue arrow on the left points from the first system to the second. A straight blue arrow in the middle points from the second system to the third.

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = x_3 + bu \\ \dot{x}_3 = h(t) \\ y = x_1 \end{cases} \longrightarrow \begin{cases} \dot{\hat{x}}_1 = \hat{x}_2 - g_1(e_1) \\ \dot{\hat{x}}_2 = \hat{x}_3 + bu - g_2(e_1) \\ \dot{\hat{x}}_3 = -g_3(e_1) \\ e_1 = \hat{x}_1 - y \end{cases}$$

思想：把所有的扰动和不确定性都打包为一个扩张状态，进行统一的估计；特别地，可把系统简化为一个积分串联型的标准形式。

■ 线性扩张状态观测器

收敛性分析

$$\left\{ \begin{array}{l} \dot{\hat{x}}_1 = \hat{x}_2 - g_1(e_1) \\ \dot{\hat{x}}_2 = \hat{x}_3 + bu - g_2(e_1) \\ \dot{\hat{x}}_3 = -g_3(e_1) \\ e_1 = \hat{x}_1 - y \end{array} \right. \longrightarrow \left\{ \begin{array}{l} \dot{\hat{x}}_1 = \hat{x}_2 - l_1 e_1 \\ \dot{\hat{x}}_2 = \hat{x}_3 + bu - l_2 e_1 \\ \dot{\hat{x}}_3 = -l_3 e_1 \\ e_1 = \hat{x}_1 - y \end{array} \right.$$

线性扩张状态观测器

(Linear Extended State Observer, LESO)

■ 非线性扩张状态观测器

$$fal(e, \alpha, \delta) = \begin{cases} |e|^\alpha \operatorname{sign}(e), & |e| > \delta \\ \frac{e}{\delta^{1-\alpha}}, & |e| \leq \delta \end{cases}, 0 < \alpha \leq 1$$

$$\begin{cases} \dot{\hat{x}}_1 = \hat{x}_2 - g_1(e_1) \\ \dot{\hat{x}}_2 = \hat{x}_3 + bu - g_2(e_1) \\ \dot{\hat{x}}_3 = -g_3(e_1) \\ e_1 = \hat{x}_1 - y \end{cases} \quad \longrightarrow \quad \begin{cases} \dot{\hat{x}}_1 = \hat{x}_2 - l_1 fal(e_1, \alpha_1, \delta) \\ \dot{\hat{x}}_2 = \hat{x}_3 + bu - l_2 fal(e_1, \alpha_2, \delta) \\ \dot{\hat{x}}_3 = -l_3 fal(e_1, \alpha_3, \delta) \\ e_1 = \hat{x}_1 - y \end{cases}$$

通常, $\alpha_1 = 1, \alpha_2 = 0.5, \alpha_3 = 0.25$

收敛性分析

■ 线性时变扩张状态观测器

$$\left\{ \begin{array}{l} \dot{\hat{x}}_1 = \hat{x}_2 - g_1(e_1) \\ \dot{\hat{x}}_2 = \hat{x}_3 + bu - g_2(e_1) \\ \dot{\hat{x}}_3 = -g_3(e_1) \\ e_1 = \hat{x}_1 - y \end{array} \right. \quad \longrightarrow \quad \left\{ \begin{array}{l} \dot{\hat{x}}_1 = \hat{x}_2 - l_1(t)e_1 \\ \dot{\hat{x}}_2 = \hat{x}_3 + bu - l_2(t)e_2 \\ \dot{\hat{x}}_3 = -l_3(t)e_3 \\ e_1 = \hat{x}_1 - y \end{array} \right.$$

$$\dot{e} = A(t)e + b(-h(x, w)) \quad A(t) = \begin{bmatrix} -l_1(t) & 1 & 0 & \cdots & 0 \\ -l_2(t) & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ -l_n(t) & 0 & \cdots & 0 & 1 \\ -l_{n+1}(t) & 0 & \cdots & 0 & 0 \end{bmatrix} \quad b = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}$$

[Ref.] Zhiqiang Pu et. al., A class of Adaptive extended state observers for nonlinear disturbed systems, ***IEEE Trans. Ind. Electron.***, 2015.

■ 扩张状态观测器的参数整定

• LESO

对于一阶、二阶、三阶对象，分别配置特征多项式为

$$(s + \omega_0)^2, (s + \omega_0)^3, (s + \omega_0)^4$$

则LESO增益为特征多项式对应系数。例如二阶对象：

$$\begin{cases} l_1 = 3\omega_0 \\ l_2 = 3\omega_0^2 \\ l_3 = \omega_0^3 \end{cases} \quad \omega_0 \text{为观测器带宽。}$$

[Ref.] Zhiqiang Gao, Scaling and bandwidth-parameterization based controller tuning, ACC 2003.

■ 扩张状态观测器的参数整定

• NESO

一些经验公式（斐波那契数列）

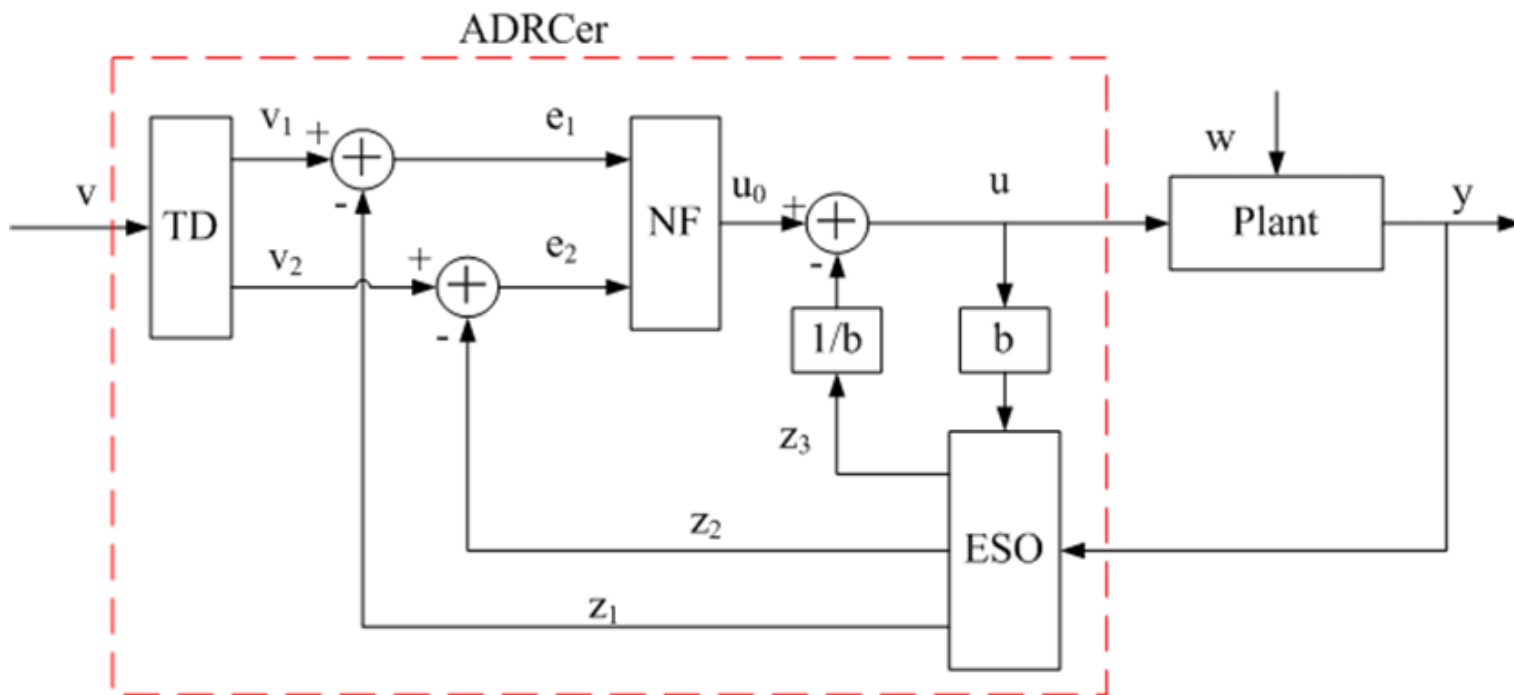
$$f_0 = 0, f_1 = 1, f_{n+1} = f_n + f_{n-1}, n = 1, 2, 3, \dots$$

1	1	2	5	8	13	34	55
1	3	8	13	21	34	55	89

$$l_1 = \frac{1}{h}, l_2 = \frac{1}{3h^2}, l_3 = \frac{2}{8^2 h^3}, l_4 = \frac{5}{13^3 h^4}, \dots$$

五、应用举例

■ 自抗扰控制器设计的分离性原理



TD、ESO、NF为三个独立的部分，可灵活组合使用，不必拘泥于形式

■ 二阶系统的标准算法

被控对象
$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = f(x_1, x_2, w, t) + bu \\ y = x_1 \end{cases}$$

- Step 1: TD

目的：安排过渡过程并获取微分信号

$$\begin{cases} v_1(k+1) = v_1(k) + hv_2(k) \\ v_2(k+1) = v_2(k) + hfhan(v_1(k) - v(k), v_2(k), r, h) \end{cases}$$

■ 二阶系统的标准算法

- Step 2: ESO

目的：估计总扰动（内部不确定性和外部扰动）

$f(x_1, x_2, w, t)$ 未知，但假设 $\dot{f} = h(t)$

$$\left\{ \begin{array}{l} \dot{x}_1 = x_2 \\ \dot{x}_2 = x_3 + bu \\ \dot{x}_3 = h(t) \\ y = x_1 \end{array} \right. \quad \longrightarrow \quad \left\{ \begin{array}{l} \dot{z}_1 = z_2 - g_1(e_1) \\ \dot{z}_2 = z_3 + bu - g_2(e_1) \\ \dot{z}_3 = -g_3(e_1) \\ e_1 = z_1 - y \end{array} \right.$$

g_i 取不同的函数形式，则ESO衍化成NESO、LESO、TESO等不同的形式

■ 二阶系统的标准算法

- Step 3: NF

目的：非线性反馈形成控制律

$$u_0 = \beta_1 \text{fal}(e_1, \alpha_1, \delta) + \beta_2 \text{fal}(e_2, \alpha_2, \delta)$$

或：

$$u_0 = -\text{fhan}(e_1, ce_2, r, h_1)$$

- Step 4: 动态补偿

目的：补偿总扰动

$$u = u_0 - \frac{z_3}{b}$$

■ 如何实现动态补偿线性化？

被控对象 $\ddot{x} = f(x, \dot{x}, w, t) + bu, b \neq 0$

我们期望闭环系统行为具有如下动态特性

$$\ddot{x} = g(x, \dot{x})$$

$f(x_1, x_2, w, t)$ 未知

设计ESO, 估计未知项 $\tilde{f} \approx f$

构造控制律 $u = (u_0 - \tilde{f}) / b$

积分串联标准型

则闭环系统为 $\ddot{x} = u_0 + (f - \tilde{f}) \approx u_0$

■ 多变量系统的解耦控制

对于MIMO系统

$$\begin{cases} \ddot{x}_1 = f_1(x_1, \dot{x}_1, \dots, x_m, \dot{x}_m) + b_{11}u_1 + \dots + b_{1m}u_m \\ \ddot{x}_2 = f_2(x_1, \dot{x}_1, \dots, x_m, \dot{x}_m) + b_{21}u_1 + \dots + b_{2m}u_m \\ \vdots \\ \ddot{x}_m = f_m(x_1, \dot{x}_1, \dots, x_m, \dot{x}_m) + b_{m1}u_1 + \dots + b_{mm}u_m \\ y_1 = x_1, \dots, y_m = x_m \end{cases}$$

设置虚拟控制量进行解耦，转化成 m 个二阶系统，设计 m 个自抗扰控制器进行控制

■ 应用练习：针对例1设计自抗扰控制器

六、一些思考

