

2024-2025 学年度春季



课程名称：《自动控制原理（一）》 第10讲 控制系统的动态性能-Part 2

课程学时：共56学时

课程性质：专业基础课

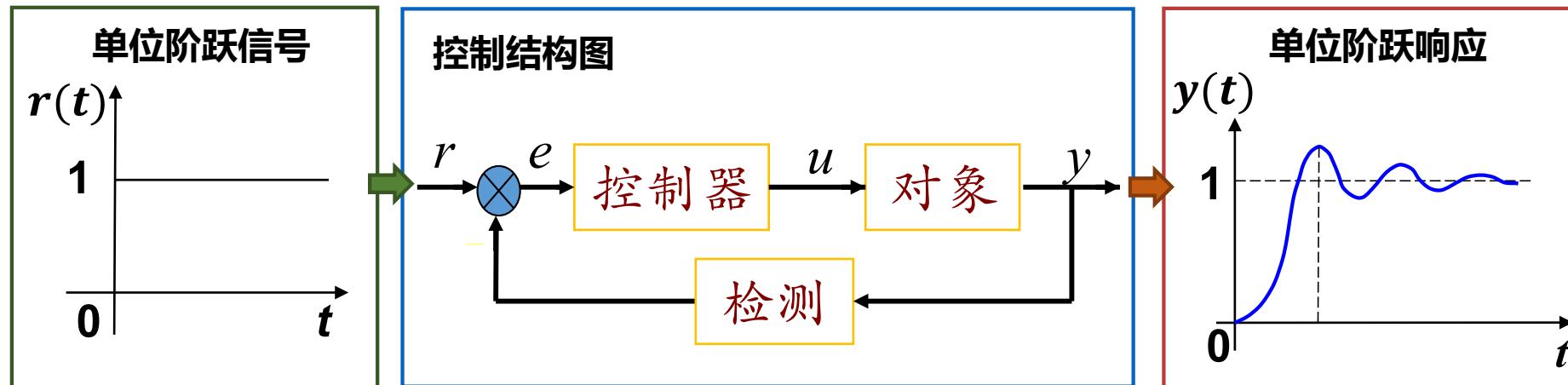
学生对象：自动化2305班
(32人)

授课教师：刘骁康

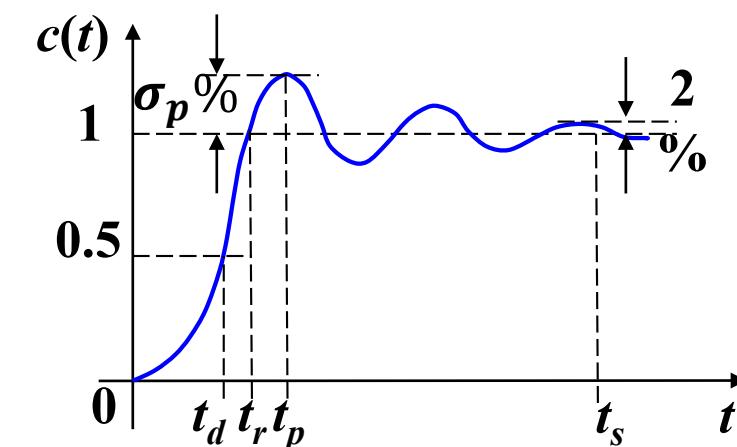
课程目标：掌握自动控制的基本原理、控制系统的建模、性能分析和综合设计方法

■ 控制系统的动态性能

动态性能分析: 零初始条件下对系统的**单位阶跃响应**的动态过程进行分析。



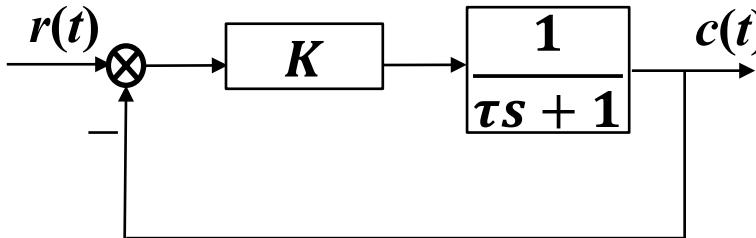
- 延迟时间 t_d
- 上升时间 t_r
- 峰值时间 t_p
- 调整时间 (调节时间) t_s
- 最大超调量 σ_p
- 振荡次数
- 衰减比



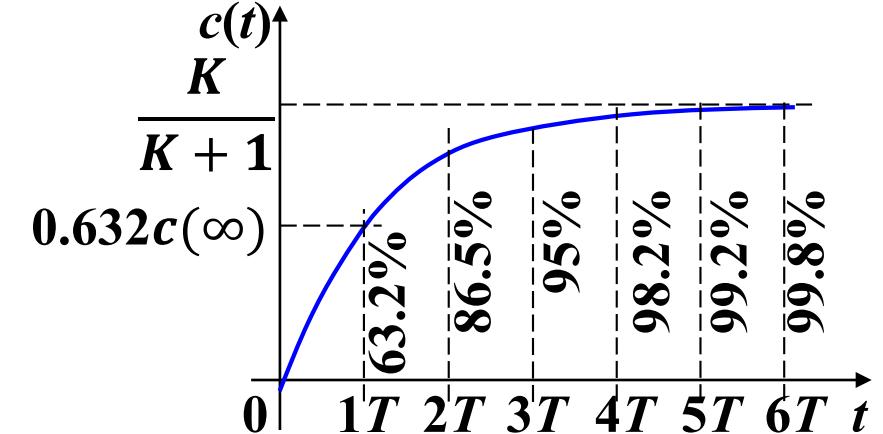
回顾-第9讲

■ 控制系统的动态性能

□ 一阶系统：



该一阶系统的闭环传递函数为 $G_B(s) = \frac{K_B}{Ts+1}$



- ✓ 延迟时间： $t_d = 0.69T$ 。
- ✓ 上升时间： $t_r = 2.2T$ 。
- ✓ 调整时间： $t_s = \begin{cases} 3T, \Delta = 5\%, \\ 4T, \Delta = 2\%. \end{cases}$

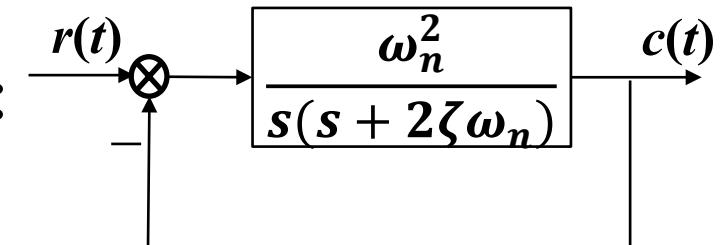
□ 结论：闭环时间常数 T 完全反映了一阶系统的动态性能。
 T 反映系统惯性， T 减小，惯性减小，响应速度加快。

■ 控制系统的动态性能

□ 二阶系统性能指标分析

$$\frac{d^2c(t)}{dt^2} + 2\zeta\omega_n \frac{dc(t)}{dt} + \omega_n^2 c(t) = \omega_n^2 r(t)$$

◆ 对 $s_{1,2} = -\zeta\omega_n \pm \omega_n\sqrt{\zeta^2 - 1}$ 进行分析可判断系统的稳定性：

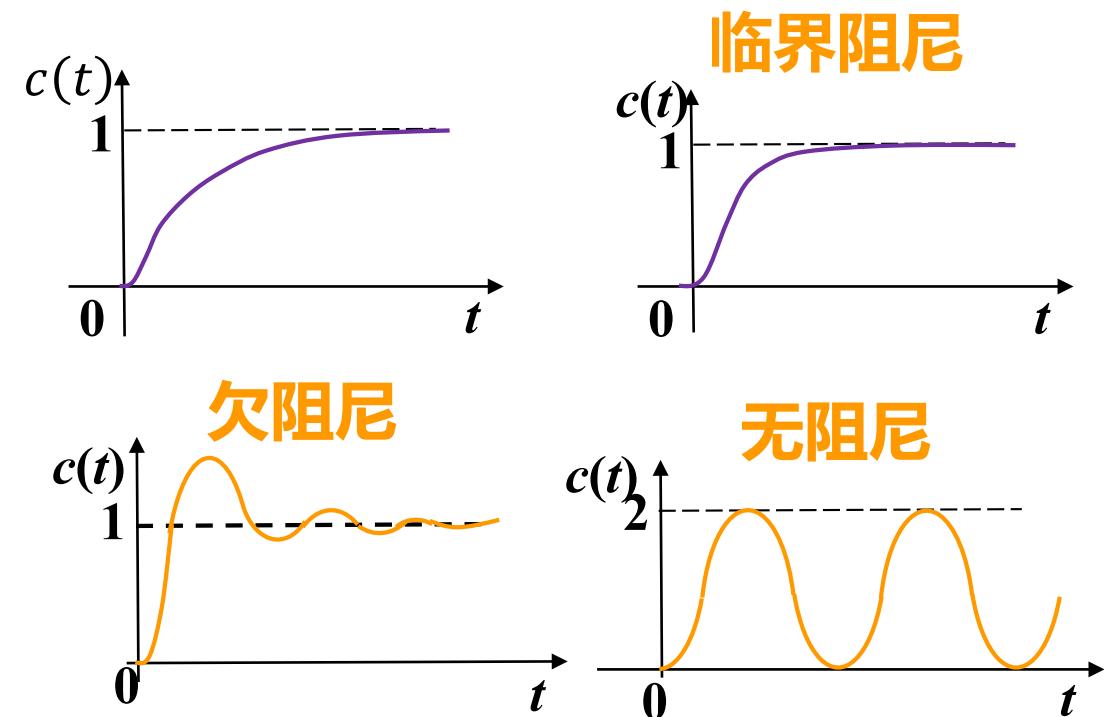


- $\zeta < 0$ 为**负阻尼**, 系统不稳定。
- $\zeta = 0$ 为**无阻尼**, 特征根为共轭纯虚根 $s_{1,2} = \pm j\omega_n$, 临界稳定。
- $0 < \zeta < 1$ 为**欠阻尼**, 系统稳定。
- $\zeta = 1$ 为**临界阻尼**, 特征根为**实数重根** $s_{1,2} = -\omega_n$, 稳定。
- $\zeta > 1$ 为**过阻尼**, 系统稳定。

回顾-第9讲

【总结】

- 从振荡程度看， ζ 越小振荡越厉害； ζ 增大到一定程度时单调上升。
- 从过渡过程时间看，无振荡时，以临界阻尼时过渡过程时间最短。欠阻尼状态的过渡过程时间比临界阻尼时更短。

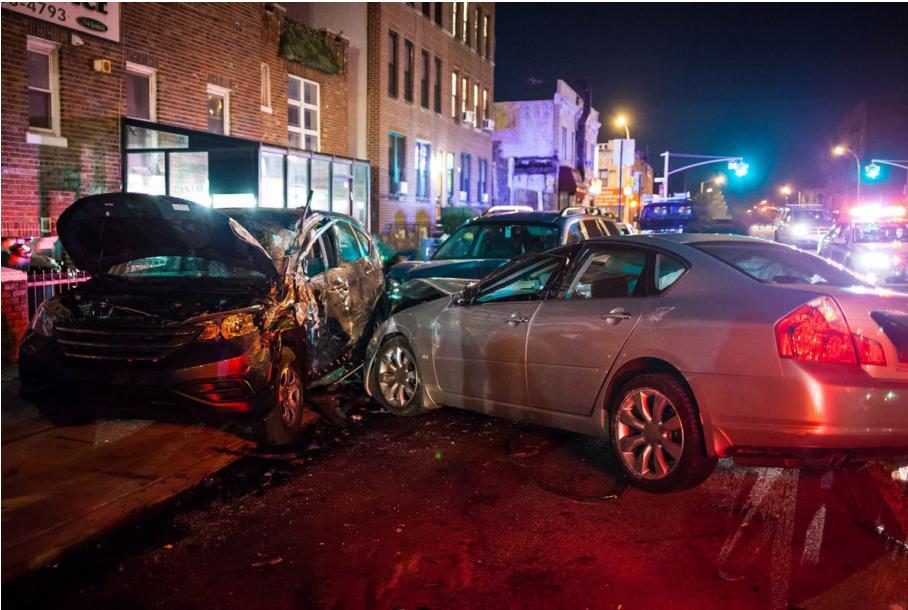


◆ 结论： ζ 增大，振荡程度减小，过渡过程时间增长。

综合过渡过程时间和振荡程度，一般希望二阶系统工作在 $\zeta = 0.4 \sim 0.8$ 的欠阻尼状态，此时过渡过程时间和振荡特性均可接受。

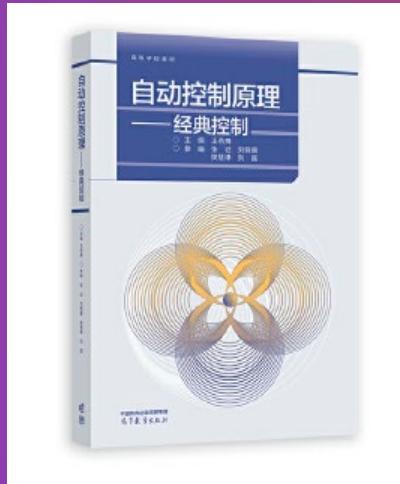
工程中常取 $\zeta = \frac{1}{\sqrt{2}} = 0.707$ 。

■ 车辆跟驰事故(Uber无人车的碰撞事故)



约90%车辆追尾都是因为与前车没有保持安全距离

如何在车辆跟驰控制中跟前车始终维持安全距离?



第三章：控制系统的时域分析

第10讲 控制系统的动态性能分析-Part 2

Transient Stability Analysis of Control Systems – Part 2

本讲内容

- 一、欠阻尼系统的性能指标
- 二、案例：车辆跟驰控制



一、欠阻尼系统的性能指标

$$c(t) = 1 - \frac{1}{\sqrt{1-\zeta^2}} e^{-\zeta\omega_n t} \sin(\omega_d t + \theta), t \geq 0,$$

1、**上升时间** t_r ：即单位阶跃响应从0上升到第一个稳态值所需要的时间。

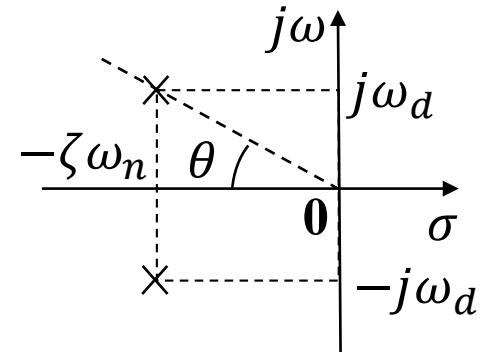
$$\text{令 } c(t_r) = 1 - \frac{1}{\sqrt{1-\zeta^2}} e^{-\zeta\omega_n t_r} \sin(\omega_d t_r + \theta) = 1,$$

$$\text{则 } \sin(\omega_d t_r + \theta) = 0,$$

单位阶跃响应第一次到达1时 $\omega_d t_r + \theta = \pi$, ? 能否为 $k\pi$?

因此 $t_r = \frac{\pi - \theta}{\omega_d}$ 。 **记**

$$\omega_d = \omega_n \sqrt{1 - \zeta^2}, \quad \theta = \arccos \zeta$$



结论：要使响应速度加快，即减小 t_r ，须使阻尼比 ζ 减小，自然振荡角频率 ω_n 增大。

一、欠阻尼系统的性能指标

$$c(t) = 1 - \frac{1}{\sqrt{1-\zeta^2}} e^{-\zeta\omega_n t} \sin(\omega_d t + \theta), t \geq 0,$$

2、**峰值时间** $\frac{dc(t)}{dt} = 0$ 的时间即为所求。

$$\omega_d = \omega_n \sqrt{1 - \zeta^2}$$

$$\text{令 } \frac{dc(t)}{dt} = \frac{e^{-\zeta\omega_n t}}{\sqrt{1-\zeta^2}} \sin(\omega_d t + \theta) \zeta \omega_n - \frac{e^{-\zeta\omega_n t}}{\sqrt{1-\zeta^2}} \cos(\omega_d t + \theta) \omega_d = 0$$

$$\text{即 } \sin(\omega_d t + \theta) \zeta - \sqrt{1 - \zeta^2} \cos(\omega_d t + \theta) = 0, \quad \theta = \arccos \zeta$$

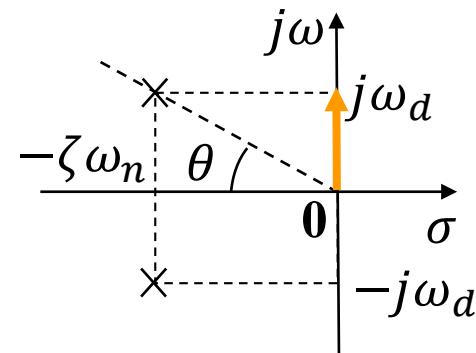
得 $\sin \omega_d t = 0$,

峰值时间是输出到达第一个最大值时对应的时间，

因此令 $\omega_d t = \pi$ 得 $t_p = \frac{\pi}{\omega_d}$ 。

记

结论：要使响应速度加快，即 t_p 减小，须使阻尼比 ζ 减小，自然振荡角频率 ω_n 增大。



一、欠阻尼系统的性能指标

$$c(t) = 1 - \frac{1}{\sqrt{1-\zeta^2}} e^{-\zeta\omega_n t} \sin(\omega_d t + \theta), t \geq 0,$$

$$t_p = \frac{\pi}{\omega_d}$$

3、超调量 $\sigma_p = \frac{c(t_p) - c(\infty)}{c(\infty)} \times 100\%$

$$c(t_p) = 1 - \frac{1}{\sqrt{1-\zeta^2}} e^{-\frac{\zeta\pi}{\sqrt{1-\zeta^2}}} \sin(\pi + \theta),$$

由 $\sin(\pi + \theta) = -\sin\theta = -\sqrt{1 - \zeta^2}$,

得 $c(t_p) = 1 + e^{\frac{-\pi\zeta}{\sqrt{1-\zeta^2}}}$, 而 $c(\infty) = 1$, 则 $\sigma_p = e^{\frac{-\pi\zeta}{\sqrt{1-\zeta^2}}} \times 100\%$ 。

记

结论：超调量只与阻尼比有关。要想超调量减小，需阻尼比 ζ 增大。

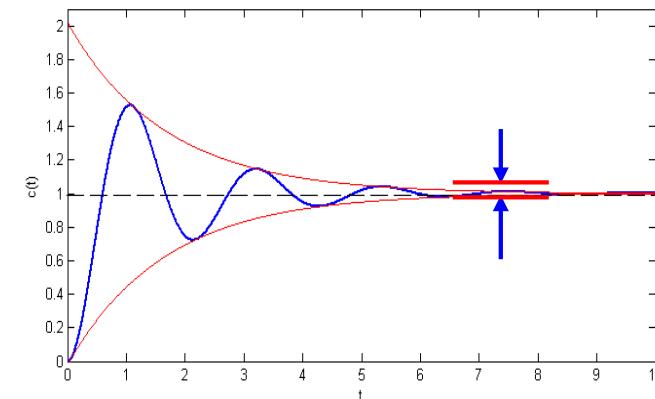
一、欠阻尼系统的性能指标

$$c(t) = 1 - \frac{1}{\sqrt{1-\zeta^2}} e^{-\zeta\omega_n t} \sin(\omega_d t + \theta), t \geq 0,$$

4. 调整时间

利用包络线来求。定义上下包络线

$$c_1(t) = 1 + \frac{e^{-\zeta\omega_n t}}{\sqrt{1-\zeta^2}}, \quad c_2(t) = 1 - \frac{e^{-\zeta\omega_n t}}{\sqrt{1-\zeta^2}},$$



➤ 可行性分析：取其中较大者与 $c(t)$ 比较大小。由于

$$c_1(t) - c(t) = \frac{e^{-\zeta\omega_n t}}{\sqrt{1-\zeta^2}} (1 + \sin(\omega_d t + \theta)) \geq 0,$$

因此用包络线来求调整时间，最大误差为一个周期的时间。

➤ 求解：令 $c_1(t) - 1 = \Delta$ ，即 $\frac{e^{-\zeta\omega_n t_s}}{\sqrt{1-\zeta^2}} = \Delta$ ，则 $t_s = \frac{1}{\zeta\omega_n} \ln \frac{1}{\Delta\sqrt{1-\zeta^2}}$ 。

一、欠阻尼系统的性能指标

$$t_s = \frac{1}{\zeta \omega_n} \ln \frac{1}{4\sqrt{1-\zeta^2}}, \text{ 如图所示。}$$

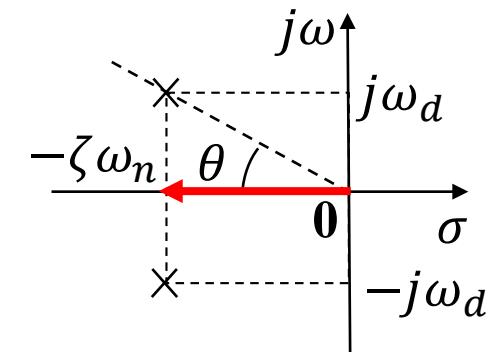
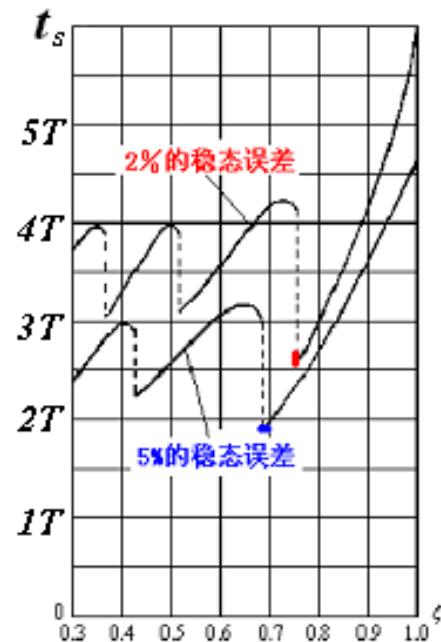
由图知，在 $\zeta=0.69 (\Delta=5\%)$ 或 $\zeta=0.77 (\Delta=2\%)$ 时， t_s 达到最小值。此后随 ζ 的增大几乎线性上升。

工程中常取

$$t_s \approx \begin{cases} \frac{4}{\zeta \omega_n}, & \Delta = 2\% \\ \frac{3}{\zeta \omega_n}, & \Delta = 5\% \end{cases}$$

记

记



结论：要 t_s 减小，须使阻尼比 ζ 增大，自然振荡角频率 ω_n 增大。

一、欠阻尼系统的性能指标

$$c(t) = 1 - \frac{1}{\sqrt{1-\zeta^2}} e^{-\zeta\omega_n t} \sin(\omega_d t + \theta), t \geq 0,$$

4. 调整时间

? 可否利用峰值时间来求?

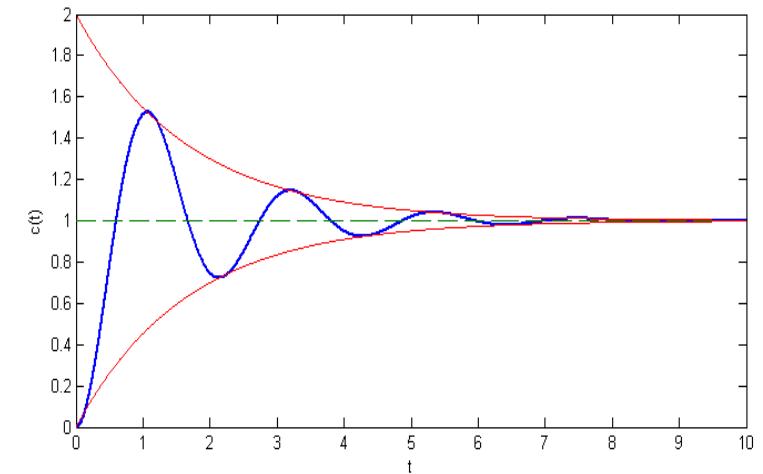
由 $t_p = \frac{\pi}{\omega_d}$ 带入得 $c(t_p) = 1 \pm e^{-\zeta\omega_n t_p}$,

定义光滑曲线 $c_1(t) = 1 + e^{-\zeta\omega_n t}$, $c_2(t) = 1 - e^{-\zeta\omega_n t}$ 。

令 $c_1 - 1 = \Delta$ 得 $t'_s = \frac{1}{\zeta\omega_n} \ln \frac{1}{\Delta}$, 则 $t_s \approx \begin{cases} \frac{4}{\zeta\omega_n}, & \Delta = 2\% \\ \frac{3}{\zeta\omega_n}, & \Delta = 5\% \end{cases}$

➤ 误差分析: 由于 $c_1 - c = e^{-\zeta\omega_n t} \left(1 + \frac{1}{\sqrt{1-\zeta^2}} \sin(\omega_d t + \theta) \right)$

曲线 $c_1(t)$ 存在小于 $c(t)$ 的值。而包络线求出的 t_s 比真实值大。



一、欠阻尼系统的性能指标

$$c(t) = 1 - \frac{1}{\sqrt{1-\zeta^2}} e^{-\zeta\omega_n t} \sin(\omega_d t + \theta), t \geq 0,$$

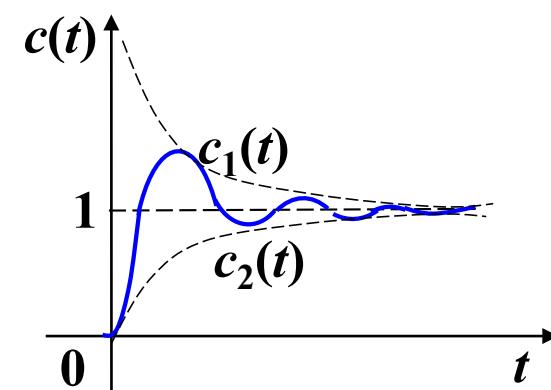
5. 振荡次数 N : 在调节时间内响应曲线穿越其稳态值次数的一半。

响应曲线振荡周期为 $T_d = \frac{2\pi}{\omega_d} = \frac{2\pi}{\omega_n \sqrt{1-\zeta^2}}$,

由 $t_s \approx \frac{4}{\zeta\omega_n}$, $\Delta=2\%$ 或 $\frac{3}{\zeta\omega_n}$, $\Delta=5\%$

则振荡次数 $N = \frac{t_s}{T_d} \approx \begin{cases} \frac{4\sqrt{1-\zeta^2}}{2\pi\zeta}, & \Delta=2\%, \\ \frac{3\sqrt{1-\zeta^2}}{2\pi\zeta}, & \Delta=5\%. \end{cases}$

结论: 振荡次数 N 只与阻尼比 ζ 有关, ζ 越大, 振荡次数越小。



一、欠阻尼系统的性能指标

$$t_r = \frac{\pi - \theta}{\omega_d} \quad t_p = \frac{\pi}{\omega_d} \quad \sigma_p = e^{\frac{-\pi\zeta}{\sqrt{1-\zeta^2}}} \times 100\% \quad \frac{3}{\zeta\omega_n}, \Delta = 5\%$$

【总结】

- 当 ω_n 一定时，要减小 t_r 和 t_p ，须减少 ζ ；而要减小 t_s ，须使 ζ 增大，即响应初期速度与总体响应速度相互矛盾。
- 当 ζ 一定时，增大 ω_n 可使 t_r 、 t_p 和 t_s 都减少，提高快速性。
- σ_p 和 N 只取决于 ζ 。 ζ 越小， σ_p 越大，振荡次数 N 越大。

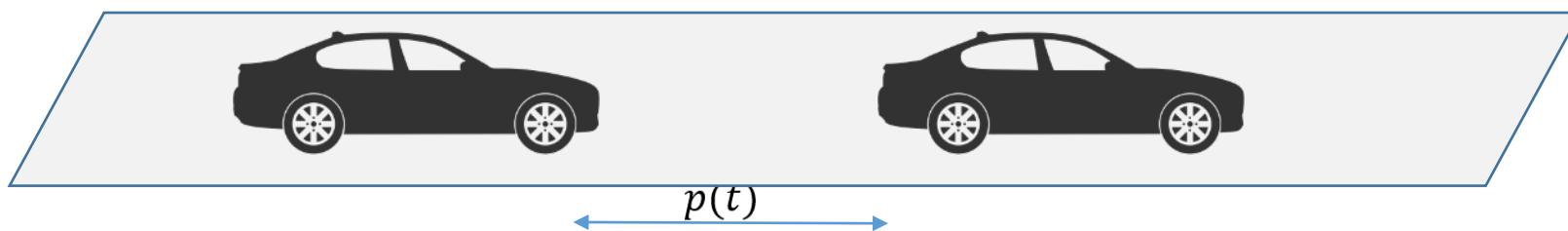
在设计控制系统时，一般根据 σ_p 的要求选择 ζ 的值（一般 $\zeta = 0.4 \sim 0.8$ ）。再调节 ω_n 来满足时间指标要求。

？其他阻尼下系统的动态性能指标能否套用欠阻尼的公式？

二、车辆跟驰控制案例

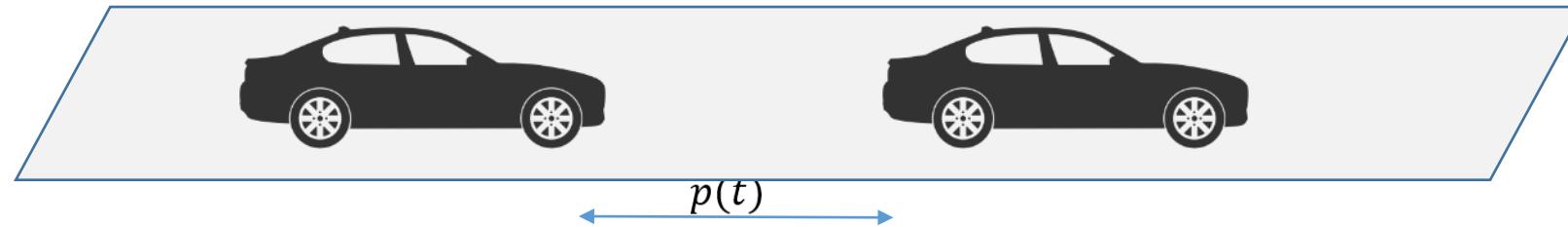
【案例：车辆跟驰】汽车初始速度为 v^* (m/s)，与前车的相对初始位移为10 米，前车按照 v^* (m/s)的速度匀速行驶。控制器设计过程中可获取汽车的速度 $v(t)$ 和相对距离 $p(t)$ 。请设计后车加速度 $u(t)$ 实现车辆跟驰，且与前车**保持相对距离致5 m**，同时

- 1) 车辆跟驰动态过程中，相对距离不能小于4.5 m；
- 2) 要求调节时间 $t_s \leq 2$ 秒($\Delta = 2\%$)。



二、车辆跟驰控制案例

【案例：车辆跟驰】汽车初始速度为 v^* (m/s)，与前车的相对初始位移为10 米，前车按照 v^* (m/s)的速度匀速行驶。控制器设计过程中可获取汽车的速度 $v(t)$ 和相对距离 $p(t)$ 。请设计后车加速度 $u(t)$ 实现车辆跟驰，且与前车保持相对距离致5 m，同时



$$\text{设 } y(t) = 10 - p(t), x(t) = v(t) - v^*, y(0) = 0, x(0) = 0$$

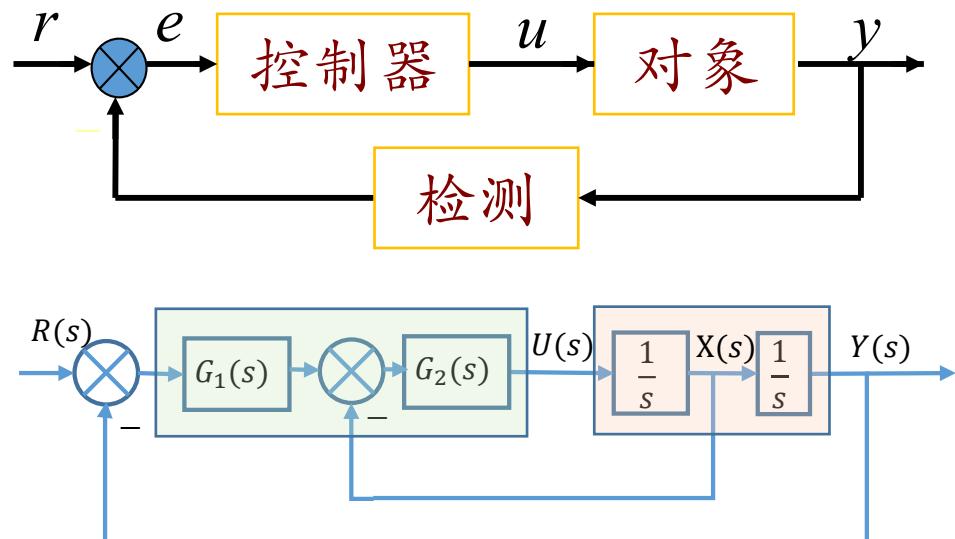
求导

$$\begin{cases} \dot{y}(t) = -\dot{p}(t) = v(t) - v^* = x(t) \\ \dot{x}(t) = \dot{v}(t) = u(t) \end{cases} \quad \xrightarrow{\text{拉式变换}} \quad \begin{cases} sY(s) = X(s) \\ sX(s) = U(s) \end{cases}$$

二、车辆跟驰控制案例

【案例：车辆跟驰】汽车初始速度为 v^* (m/s)，与前车的相对初始位移为10 米，前车按照 v^* (m/s)的速度匀速行驶。控制器设计过程中可获取汽车的速度 $v(t)$ 和相对距离 $p(t)$ 。请设计后车加速度 $u(t)$ 实现车辆跟驰，且与前车保持相对距离致5 m，同时

设 $y(t) = 10 - p(t)$, $x(t) = v(t) - v^*$



期望输出: $y_{ref} = 10 - 5 = 5$

参考输入: $r(t) = 5l(t)$

最大超调量:

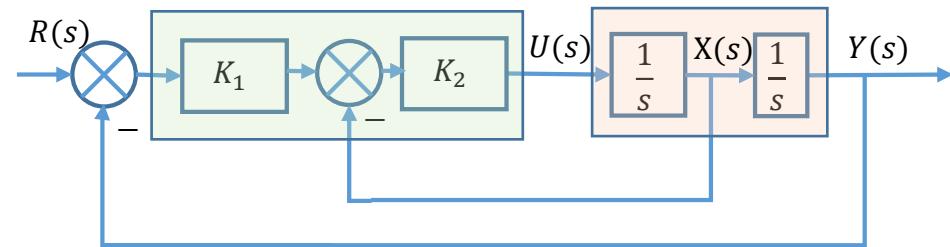
$$\sigma_p \leq \frac{(10-4.5)-5}{5} \times 100\% = 10\%$$

调节时间: $t_s \leq 2$ s

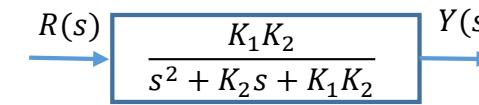
$$G_1(s) = K_1 \quad G_2(s) = K_2$$

如何结合欠阻尼二阶系统性能指标设计控制器参数 K_1 和 K_2 呢？

二、车辆跟驰控制案例



结构图
化简



最大超调量: $\sigma_p \leq 10\%$

$$\begin{cases} \sigma_p = e^{\frac{-\pi\xi}{\sqrt{1-\xi^2}}} \leq 0.1 \\ t_s = \frac{4}{\xi\omega_n} \leq 2 \end{cases}$$

调节时间: $t_s \leq 2$ s

$$\begin{cases} K_1K_2 = \omega_n^2 \\ K_2 = 2\xi\omega_n \end{cases}$$

$$\rightarrow \begin{cases} \xi \geq 0.59 \\ \xi\omega_n \geq 2 \end{cases} \rightarrow \begin{cases} \xi = 0.7 \\ \omega_n = 3 \end{cases} \rightarrow \begin{cases} K_2 = 4.2 \\ K_1 = 2.2 \end{cases}$$

控制器: S域 $U(S) = K_2(K_1(R(s) - Y(s)) - X(s))$

时域 $u(t) = 4.2 \times (2.2 \times (5l(t) - y(t)) - x(t))$

$v(t) - v^*$

$10 - p(t)$

二、车辆跟驰控制案例

【案例：车辆跟驰】 仿真算例1：控制参数 $K_1 = 4, K_2 = 3$

The screenshot shows a Python development environment with two main panes. The left pane displays the source code for `CarEnvironment.py`, while the right pane shows a plot and the Python console.

Code (CarEnvironment.py):

```
195     # 检测事件发生，先把事件全部拿过来，再看发生的是哪一件事。
196     for event in pygame.event.get():
197         # 检测关闭按钮被点击的事件
198
199         if event.type == pygame.QUIT or (
200             event.type == KEYDOWN and (
201                 event.key == K_ESCAPE or event.key == K_q
202             )):
203             # 如果按下屏幕右上方的X号就会退出
204             print(event)
205             pygame.quit()
206             exit()
207
208
209
210
211
212
213
214     Background_sprites.draw(screen)
215     Background_sprites.drawtexts()
216
217
218     Car_sprites.draw(screen)
219     Car_sprites.speedx = 0
220
221
222
223     # errorx = (Car_sprites.speedx - Car_sprites_Follow.speedx)*0.6
224     # errorx = (Car_sprites.x - Car_sprites_Follow.x-1000)/100
225     Car_sprites_Follow.draw(screen)
226     # Car_sprites_Follow.speedx = (errorx*0.0001+errorx*3
227     # Pygame.time.get_ticks()*0.0001
228     # errorx = (Car_sprites_Follow.speedx-(Car_sprites.speedx))*1.2
229     # errorx = (Car_sprites.x - Car_sprites_Follow.x)/100
230
231     y = 10 - errorx
232     # state = state*(4.2*(2*(5-y)-errorx))/120
233     state = state*(3*(2*(5-y)-errorx))/120
234     Car_sprites_Follow.speedx = state
235
236     p_store[count,0] = (pygame.time.get_ticks()-3000)/1000
237     p_store[count,1] = errorx
238     count = count + 1
239     print("当前车距(m):", "str(round(errorx,3))+')米')
240
241     # Car_sprites_Follow.speedx = 1
242     # Car_sprites_front.draw(screen)
243
244     if pygame.time.get_ticks()>0000 and flag == 1:
245
246         plt.plot(p_store[1:count-1,0], p_store[1:count-1,1], linestyle='--')
247         plt.grid()
248         plt.plot(p_store[1:count-1,0], np.ones(count-2)*4.5, linestyle='--', color='r')
249         plt.xlim([0,4])
250         plt.ylim([0,10])
251         plt.xticks(range(11))
252         plt.xlabel('时间(s)')
253         plt.ylabel('相对距离(m)')
254         plt.savefig('Curve.jpg', dpi = 600)
255         flag = 0
256
257
258     #update
259     Car_sprites.update()
260     Car_sprites_Follow.update()
261     # Car_sprites_front.update()
262     Background_sprites.update()
263
264
265
266     Pygame.display.update()
```

Plot: A line graph titled "Curve.jpg" showing the relative distance (m) over time (s). The x-axis ranges from 0 to 4 seconds, and the y-axis ranges from 0 to 10 meters. The plot shows a series of points connected by dashed lines, representing the distance between the lead vehicle and the follower vehicle over time. A red dashed line at y=4.5 serves as a reference line.

Console:

```
离前车距离: 9.0米
离前车距离: 5.0米
<Event(256-Quit )>
```

二、车辆跟驰控制案例

【案例：车辆跟驰】 仿真算例2：控制参数 $K_1 = 2.2, K_2 = 4.2$

The screenshot shows a Python development environment and a Jupyter Notebook interface.

Code Editor: Displays the `CarEnvironment.py` file. The code implements a vehicle-following model using Pygame. It handles events like quit and keydown, updates vehicle positions, and plots the trajectory. Key parameters include $K_1 = 2.2$ and $K_2 = 4.2$.

```
# 检测事件发生，先把事件全部拿过来，再看发生的是哪一件事。
for event in pygame.event.get():
    # 检测关闭按钮或点击的事件
    if event.type == pygame.QUIT or (
        event.type == KEYDOWN and (
            event.key == K_ESCAPE or event.key == K_q
        )):
        # 如果按下屏幕右上方的X号就会退出
        print(event)
        pygame.quit()
        exit()

# 背景精灵绘制
Background_sprites.draw(screen)
Background_sprites.drawtexts()

# 车辆精灵绘制
Car_sprites.draw(screen)
Car_sprites.speedx = 0

# 计算误差
error = (Car_sprites.speedx - Car_sprites.Follow.speedx)*0.6
# error = (Car_sprites.x - Car_sprites.Follow.x)*1000/100
Car_sprites.Follow.speedx = error*1*error*3
if pygame.time.get_ticks()>3000:
    error = (Car_sprites.Follow.speedx-(Car_sprites.speedx))*1.2
    errorx = (Car_sprites.x - Car_sprites.Follow.x)/100
    y = 10 - errorx
    state = state+(4.2*(2.2*(5-y)-error))/120
    state = state+(4.2*(2.2*(5-y)-error))/120
    Car_sprites.Follow.speedx = state

p_store[count,0] = (pygame.time.get_ticks()-3000)/1000
p_store[count,1] = errorx
count = count + 1
print('离前车距离: '+str(round(errorx,3))+'米')

# Car_sprites.Follow.speedx = 1
# Car_sprites_front.draw(screen)

if pygame.time.get_ticks()>7000 and flag == 1:
    plt.plot(p_store[1:count-1,0], p_store[1:count-1,1], linestyle='--')
    plt.grid()
    plt.plot(p_store[1:count-1,0], np.ones(count-2)*4.5, linestyle='--', color='r')
    plt.xlim([0,4])
    plt.ylim([0,16])
    plt.xlabel('时间(s)')
    plt.ylabel('相对距离(m)')
    plt.savefig('curve.jpg', dpi = 600)
    flag = 0

# 更新
Car_sprites.update()
Car_sprites.Follow.update()
# Car_sprites_front.update()
Background_sprites.update()
pygame.display.update()
```

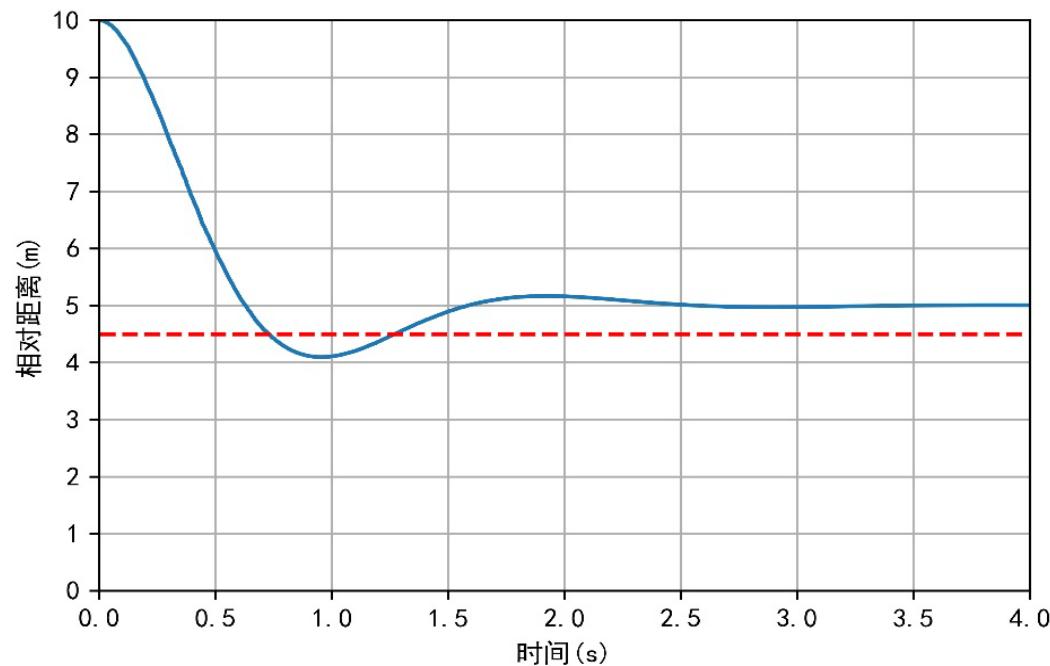
Jupyter Notebook: Shows the execution of the code in cell [11]. The output displays the relative distance between the front vehicle and the follower at each time step, with a constant value of 5.0 meters.

```
In [11]:  
Out[11]:  
离前车距离: 5.0米  
<event(256-Quit {})>
```

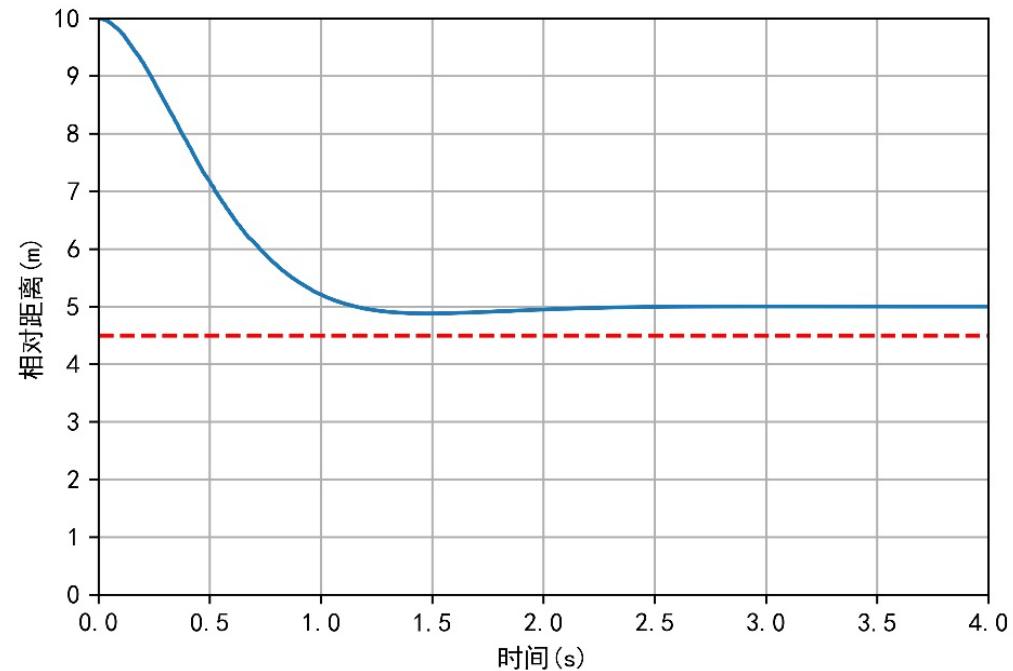
二、车辆跟驰控制案例

【案例：车辆跟驰】

算例1：参数 $K_1 = 4$ $K_2 = 3$



算例2：参数 $K_1 = 2.2$ $K_2 = 4.2$



- 算例1的相对距离超过警戒线，调节时间超过2秒
- 算例2的相对距离并未超过警戒线，调节时间在2秒之内

思考：如果后车的初始速度不为 v^* 的时候，该如何设计呢？

- 控制系统的暂态性能：
 - 暂态性能指标：7个
 - 一阶系统的性能指标
 - 二阶系统的性能指标
 - 案例：车辆跟驰控制
- 作业：
 - 作业3.3和3.5
- 大作业：

第一问：如何设计控制器？使得轮式机器人俯仰角在 0.2 rad 的偏置下，1秒内稳定到平衡点 $\pm 0.01 \text{ rad}$ 附近。