

2024-2025 学年度春季



课程名称：《自动控制原理（一）》

第12讲 控制系统的稳态性能

课程学时：共56学时

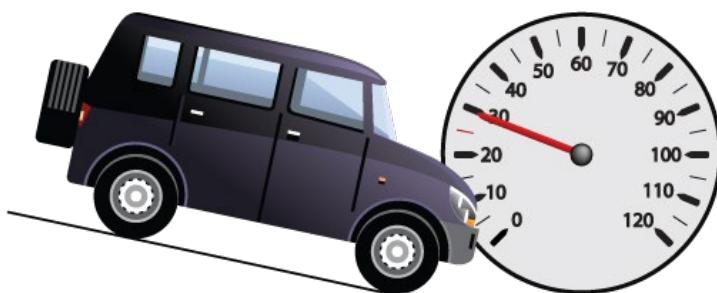
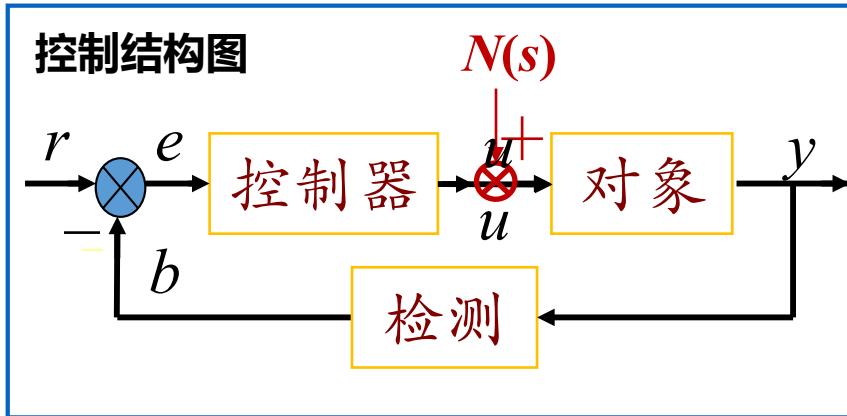
课程性质：专业基础课

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

■ 车辆定速巡航

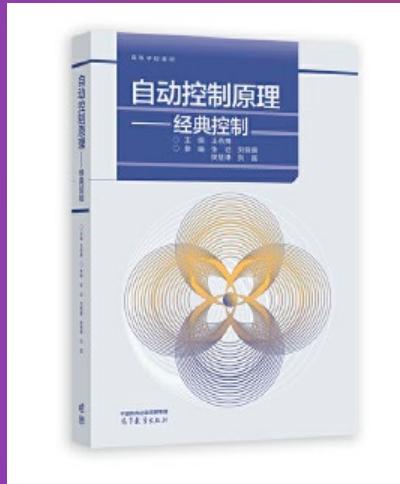


斜坡上运动

$$\dot{v}(t) = u(t) \rightarrow \dot{v}(t) = u(t) + n(t)$$

重力分量作为外部干扰作用在加速度上

为什么车辆最后
还能按照预设速度行驶?



第三章：控制系统的时域分析

第12讲 控制系统的稳态性能分析

Steady Stability Analysis of Control Systems

本讲内容

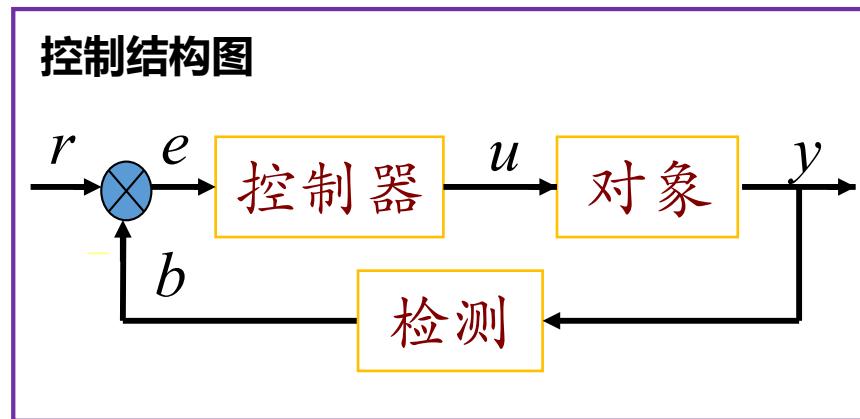
- 一、 稳态误差的定义
- 二、 扰动输入下控制系统的稳态误差
- 三、 案例：定速巡航控制综合设计



一、稳态误差的定义

□ 稳态误差：

【误差】对图示典型控制系统，其**误差** $e(t)$ 是参考输入信号 $r(t)$ 和反馈信号 $b(t)$ 的差值，即 $e(t) = r(t) - b(t)$ 。



稳态响应：控制系统最终状态的表现形式。

稳态误差：系统实际输出值与希望输出值之间的最终偏差。

一、稳态误差的定义

□ 稳态误差的计算：

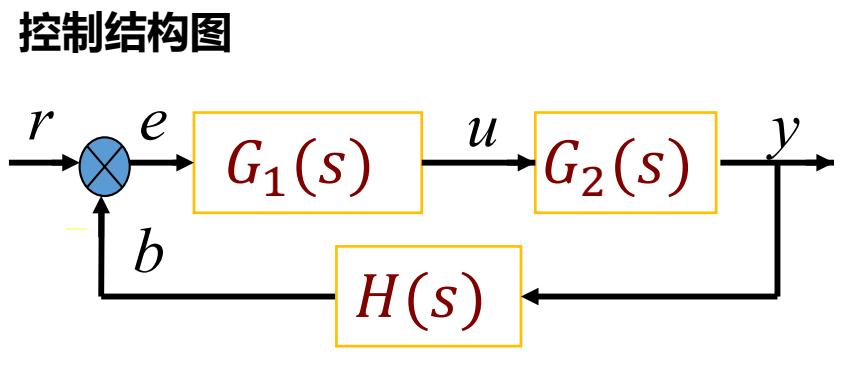
时域：误差 $e(t)$ 在 $t \rightarrow \infty$ 时的值，记为 e_{ss} ，即

$$e_{ss} = e(\infty) = \lim_{t \rightarrow \infty} e(t) = \lim_{t \rightarrow \infty} [r(t) - b(t)]。$$

S域：(终值定理法) 根据终值定理，即

$$e_{ss} = \lim_{t \rightarrow \infty} e(t) = \lim_{s \rightarrow 0} s E(s)$$

如何根据控制结构图求稳态误差呢？



求误差传递函数

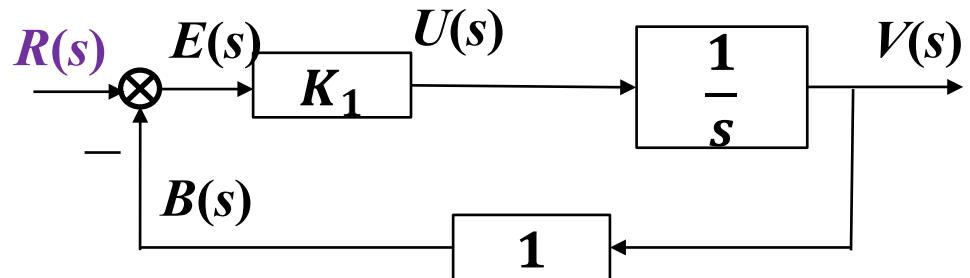
$$\Phi_{er}(s) = \frac{E(s)}{R(s)} = \frac{1}{1 + G_1(s)G_2(s)H(s)}$$

根据终值定理，

$$e_{ss} = \lim_{s \rightarrow 0} s E(s) = \lim_{s \rightarrow 0} s \Phi_{er}(s) R(s)$$

一、稳态误差的定义

【案例】设车辆的动力学为 $\dot{v}(t) = u(t)$ ，车辆初始速度为0 (m/s)，拟控制车辆以30 (m/s)的速度定速巡航。请计算稳态误差？



方法：

求误差传递函数

$$\Phi_{er}(s) = \frac{E(s)}{R(s)} = \frac{1}{1 + G_1(s)G_2(s)H(s)}$$

根据终值定理，

$$e_{ss} = \lim_{s \rightarrow 0} s E(s) = \lim_{s \rightarrow 0} s \Phi_{er}(s)R(s)$$

步骤 1：求误差传递函数

$$\Phi_{er}(s) = \frac{E(s)}{R(s)} = \frac{1}{1 + \frac{K_1}{s}} = \frac{s}{s + K_1}$$

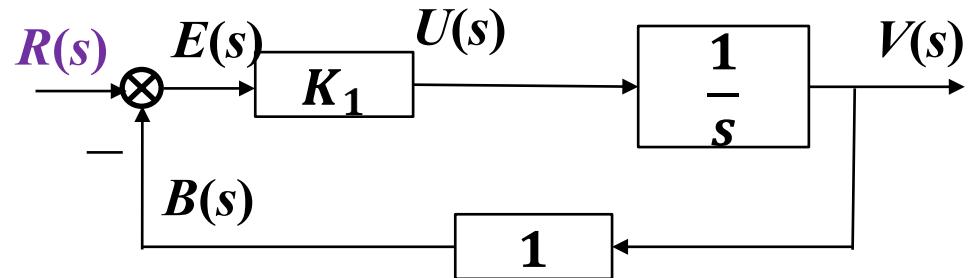
步骤 2：根据终值定理

$$\begin{aligned} e_{ss} &= \lim_{s \rightarrow 0} s E(s) = \lim_{s \rightarrow 0} s \Phi_{er}(s)R(s) \\ &= \lim_{s \rightarrow 0} s \times \frac{s}{s + K_1} \times \frac{1}{s} \\ &= 0 \end{aligned}$$

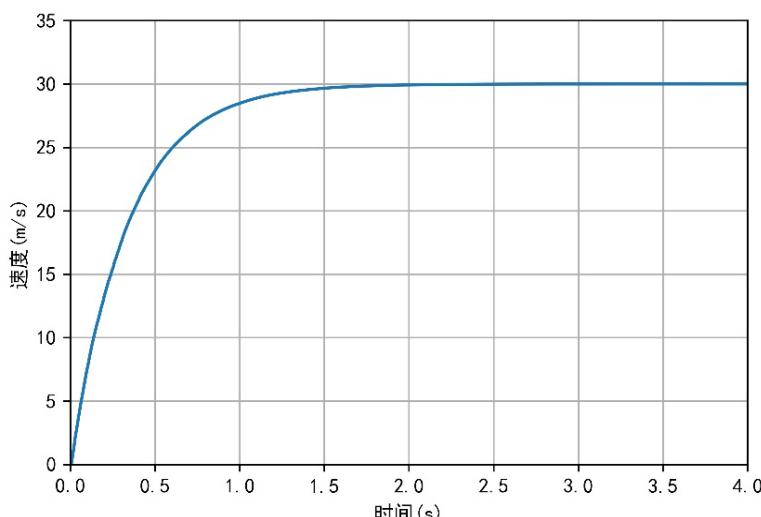
一、稳态误差的定义

重点

【案例】设车辆的动力学为 $\dot{v}(t) = u(t)$ ，车辆初始速度为0 (m/s)，拟控制车辆以30 (m/s)的速度定速巡航。请计算稳态误差？



控制器: $u(t) = K_1 e(t)$



速度曲线 (无干扰)

File Edit Search Source Run Debug Consoles Projects Tools View Help

...01 本科(04 本科教学(01 本科生课程(02 自控原理(22-23年春季人工智能2101-2102班)Code\CarDriving

CarEnvironment_CourseControl.py x

```
239
240     e = 30 - v
241     # statev = statev+e*v/120
242     # statev = statev+(statev-v*10)*1/120
243     statev = statev+(3*e*v)*1/120
244
245     Background_sprites.speedx = -statev/1.2
246
247     Car_sprite.Follow.speedx = 0
248     Car_sprites_Follow.speedbackground = Background_sprites.speedx
249
250     p_store[count,0] = (pygame.time.get_ticks()-3000)/1000
251     p_store[count,1] = v
252     count = count + 1
253     print("速度: "+str(round(v,3))+'米/秒')
254     Background_sprites.v = v
255
256     # Car_sprites_Follow.speedx = 1
257     # Car_sprites_Front.draw(screen)
258
259     if pygame.time.get_ticks()>7000 and flag == 1:
260
261         plt.plot(p_store[1:count-1,0], p_store[1:count-1,1], linestyle='.')
262         plt.plot(p_store[1:count-1,0], np.ones(count-2)*30, linestyle='--',color='r')
263         plt.ylim([0,4])
264         # plt.yticks(range(36))
265         plt.xlabel('时间(s)')
266         plt.ylabel('速度(米/秒)')
267         plt.savefig('curve.jpg',dpi = 600)
268         flag = 0
269
270
271     #update
272     # Car_sprites.update()
273     Car_sprites_Follow.update()
274     # Car_sprites_Front.update()
275     Background_sprites.update()
276
277     pygame.display.update()
278
279
280
```

Help Variable Explorer Plots Files

Console 1/k

速度: 33.333米/秒
<Event(768-KeyDown {'unicode': '\x1lb', 'key': 27, 'mod': 4896, 'scancode': 41, 'window': None}>

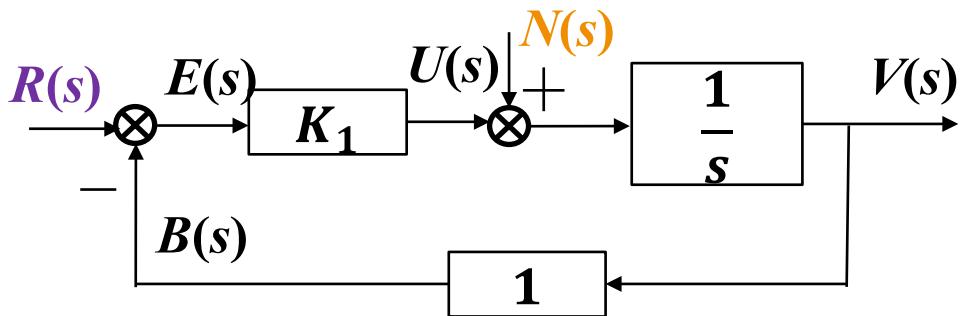
In [8]:

IPython console History

虚拟仿真平台($K_1 = 3$)

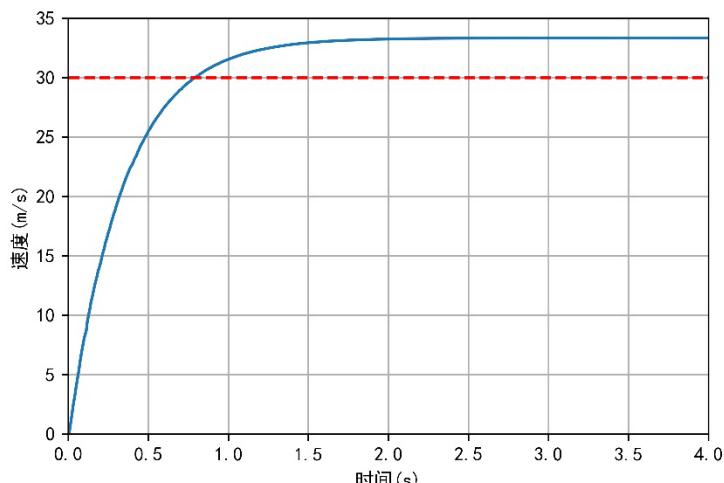
二、扰动输入下控制系统的稳态误差

【案例】设车辆的动力学为 $\dot{v}(t) = u(t)$ 车辆初始速度为0 (m/s)，拟控制车辆以30 (m/s)的速度定速巡航，若存在 $n(t) = 10l(t)$ 的扰动信号。请计算稳态误差？



控制器: $u(t) = K_1 e(t) + n(t), \quad n(t) = 10l(t)$

速度曲线
(有干扰)



```

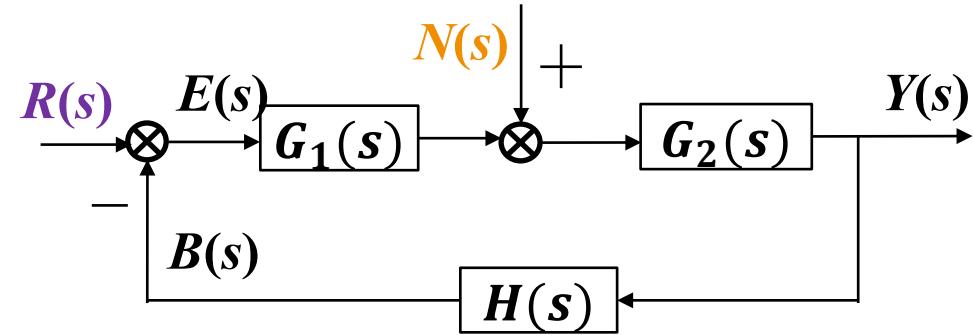
File Edit Search Source Run Debug Consoles Projects Tools View Help
...01 本科04 本科教学01 本科生课程02 自控原理(22-23年春季人工智能2101-2102班)Code\CarDriving\CarEnvironment_CourseControl.py
CarEnvironment_CourseControl.py x
238     e = 30 - v
239     # statev = statev+5*e*1/120
240     # statev = statev*(statev-v*10)*1/120
241     statev = statev*(3*e+10)*1/120
242
243     Background_sprites.speedx = -statev/1.2
244
245     Car_sprites_Follow.speedx = 0
246     Car_sprites_Follow.speedbackground = Background_sprites.speedx
247
248     p_store[count,0] = (pygame.time.get_ticks()-3000)/1000
249     p_store[count,1] = v
250     count = count + 1
251     print("速度: "+str(round(v,3))+"米/秒")
252     Background_sprites.v = v
253
254     # Car_sprites_Follow.speedx = 1
255     # Car_sprites_front.draw(screen)
256
257     if pygame.time.get_ticks()>7000 and flag == 1:
258
259         plt.plot(p_store[1:count-1,0], p_store[1:count-1,1], linestyle='--')
260         plt.plot(p_store[1:count-1,0], np.ones(count-2)*30, linestyle='--', color='r')
261
262         plt.ylim([0,45])
263         plt.xlim([0,4])
264         # plt.xticks(range(36))
265         plt.xlabel('时间(s)')
266         plt.ylabel('速度(m/s)')
267         plt.savefig('Curve.jpg',dpi = 600)
268         flag = 0
269
270
271     #update
272     # Car_sprites.update()
273     Car_sprites_Follow.update()
274     # Car_sprites_front.update()
275     Background_sprites.update()
276     pygame.display.update()
277
278
279
280
<Event(768-Keydown {'unicode': '\x1b', 'key': 27, 'mod': 4096, 'scancode': 41, 'window': None}>
In [7]:

```

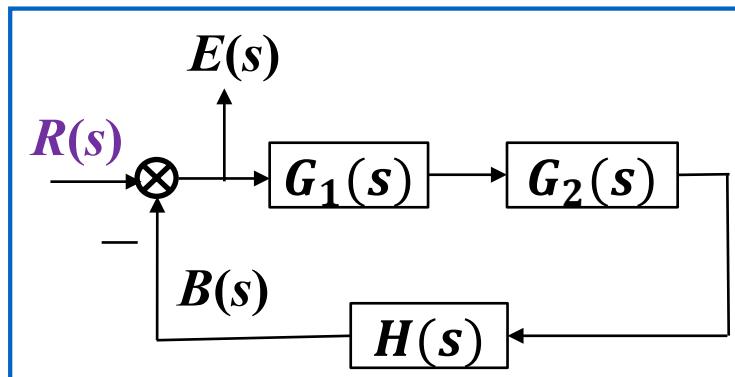
虚拟仿真平台($K_1 = 3$, 有干扰)

二、扰动输入下控制系统的稳态误差

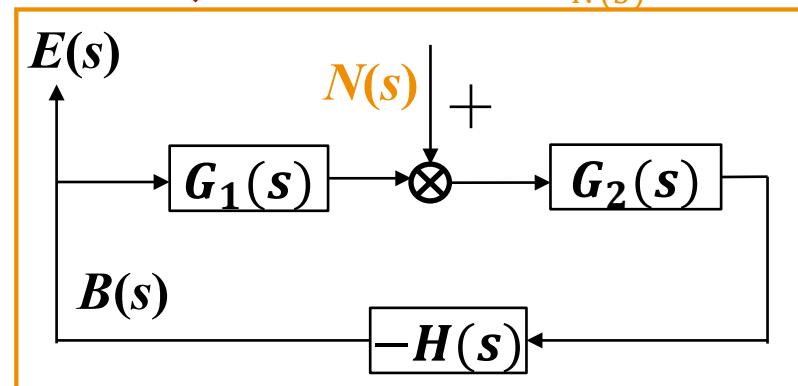
□ 扰动输入下的稳态误差计算:



计算 $\Phi_{er}(s) = \frac{E(s)}{R(s)}$



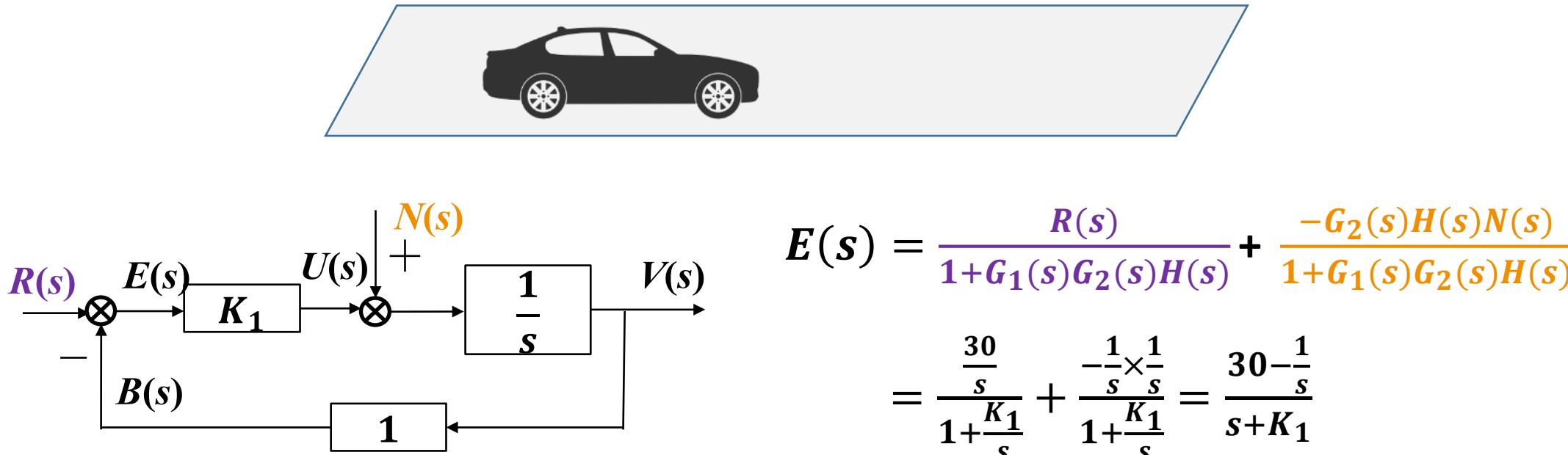
计算 $\Phi_{en}(s) = \frac{E(s)}{N(s)}$



$$E(s) = \Phi_{er}(s)R(s) + \Phi_{en}(s)N(s) = \frac{R(s)}{1+G_1(s)G_2(s)H(s)} + \frac{-G_2(s)H(s)N(s)}{1+G_1(s)G_2(s)H(s)}$$

三、定速巡航控制综合设计

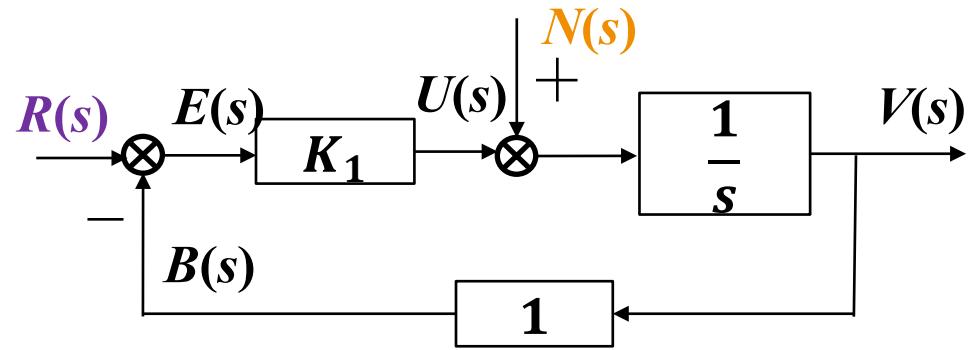
【案例】设车辆的动力学为 $\dot{v}(t) = u(t)$ ，车辆初始速度为0 (m/s)，拟控制车辆以30 (m/s)的速度定速巡航。车辆控制系统如下图所示， $N(t) = l(t)$ 为控制输入的外部干扰，请问车辆是否能够通过调节参数 K_1 消除稳态误差？



$$e_{ss} = \lim_{s \rightarrow 0} s E(s) = \lim_{s \rightarrow 0} \frac{30s-1}{s+K_1} = -\frac{1}{K_1}$$

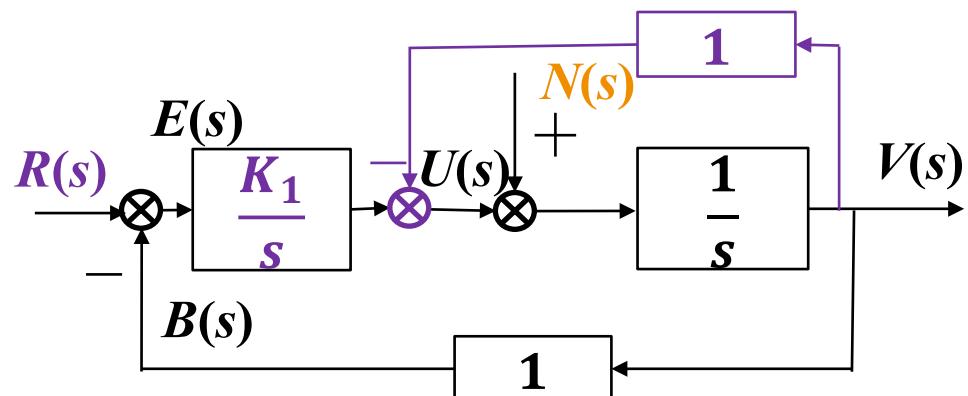
三、定速巡航控制综合设计

【案例】设车辆的动力学为 $\dot{v}(t) = u(t)$ ，车辆初始速度为0 (m/s)，拟控制车辆以30 (m/s)的速度定速巡航。如何设计控制器，消除稳态误差？



$$\text{S域: } U(s) = K_1 E(s)$$

$$\text{时域: } u(t) = K_1 e(t)$$

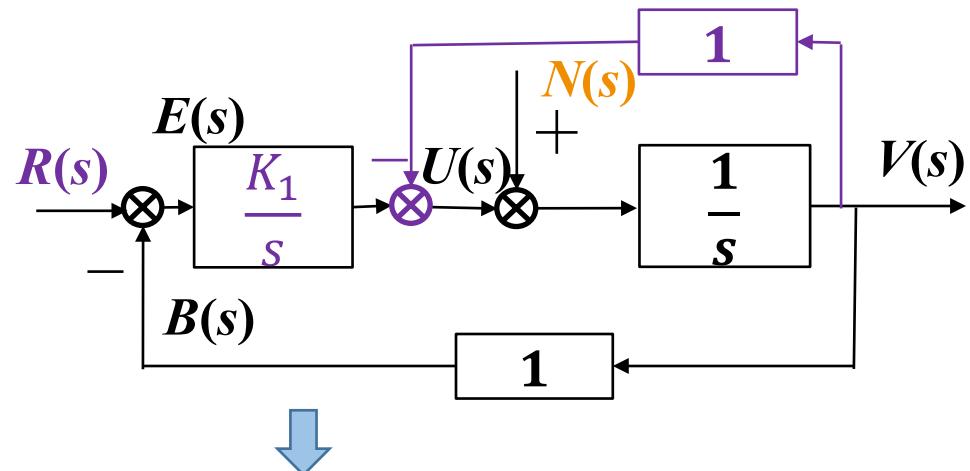


$$\text{S域: } U(s) = \frac{K_1}{s} E(s) - V(s)$$

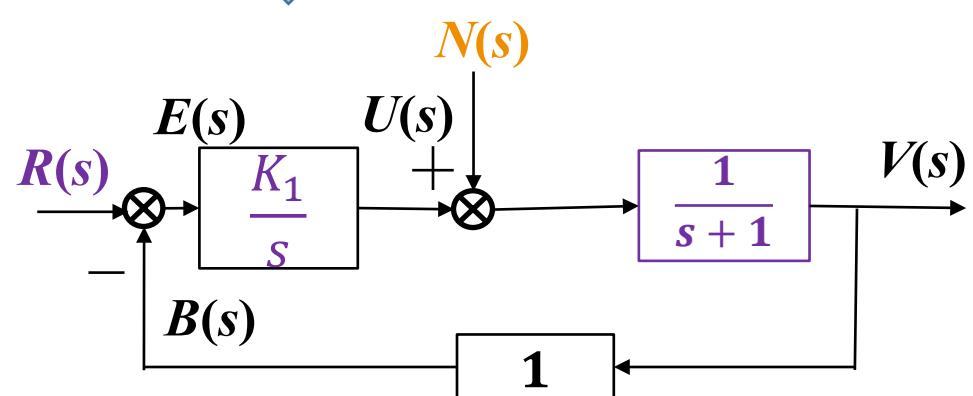
$$\text{时域: } \begin{cases} \dot{z}(t) = e(t) \\ u(t) = K_1 z(t) - v(t) \end{cases}$$

三、定速巡航控制综合设计

【案例】设车辆的动力学为 $\dot{v}(t) = u(t)$ ，车辆初始速度为0 (m/s)，拟控制车辆以30 (m/s)的速度定速巡航。如何设计控制器，消除稳态误差？



$$\begin{aligned}E(s) &= \frac{R(s)}{1+G_1(s)G_2(s)H(s)} + \frac{-G_2(s)H(s)N(s)}{1+G_1(s)G_2(s)H(s)} \\&= \frac{\frac{30}{s}}{1+\frac{K_1}{s(s+1)}} + \frac{-\frac{1}{s+1} \times \frac{1}{s}}{1+\frac{K_1}{s(s+1)}} = \frac{30s-1}{s^2+s+K_1}\end{aligned}$$

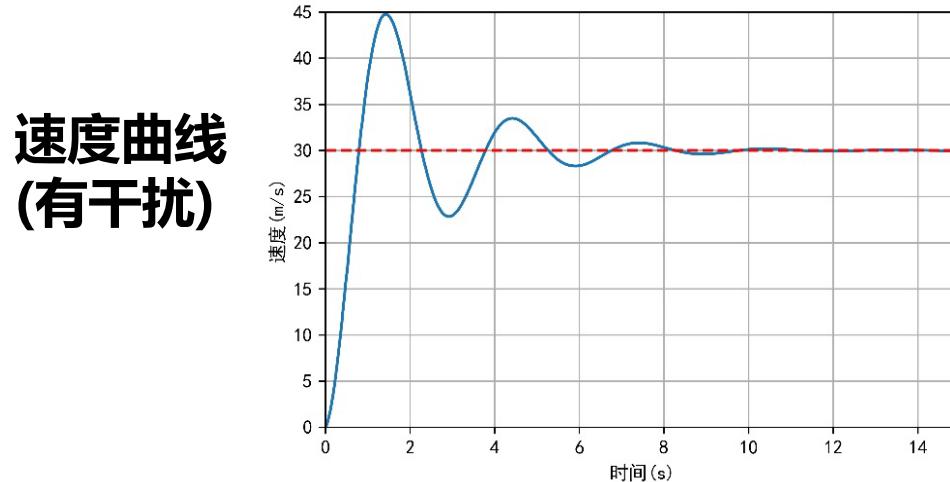
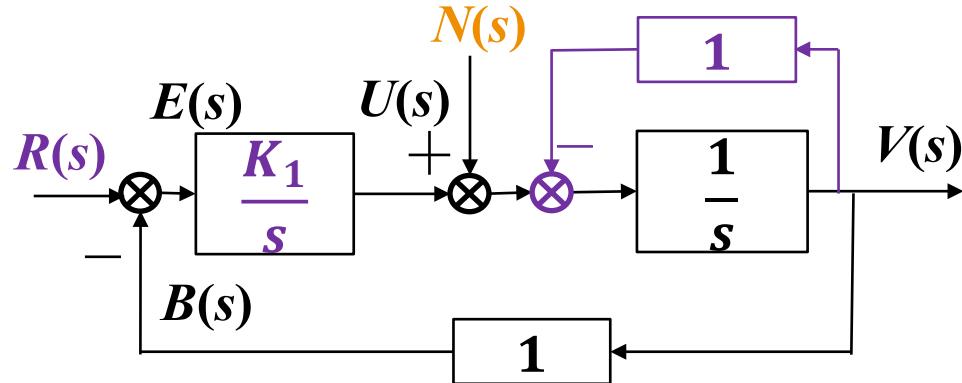


$$e_{ss} = \lim_{s \rightarrow 0} s E(s) = \lim_{s \rightarrow 0} \frac{s(30s-1)}{s^2+s+K_1} = 0$$

三、定速巡航控制综合设计

重点

【案例】设车辆的动力学为 $\dot{v}(t) = u(t)$ ，车辆初始速度为0 (m/s)，拟控制车辆以30 (m/s)的速度定速巡航。如何设计控制器，消除稳态误差？



改进的控制器：

$$\begin{cases} \dot{\mathbf{z}}(t) = \mathbf{e}(t) \\ \mathbf{u}(t) = K_1 \mathbf{z}(t) - \mathbf{v}(t) + \mathbf{n}(t) \end{cases}$$

The screenshot shows two side-by-side code editors. The left editor displays Python code for a car driving simulation, specifically 'CarEnvironment.CourseControl.py'. The right editor shows a real-time simulation of a car driving on a road, with various performance metrics displayed in the status bar.

```
File Edit Search Source Run Debug Consoles Projects Tools View Help
...01 本科(0) 本科教学(0) 本科生课程(0) 自控原理(22-23年春季人工智能2101-2102班) CodeCarDriving\CarEnvironment_CourseControl.py
CarEnvironment.CourseControl.py
148
149     # def drawText(self):
150     #     self.Band_M.my_font = pygame.font.SysFont("Times new roman", 30)
151     #     self.Band_M.text_surface = self.Band_M.my_font.render('Add %d' %self.c*20), True, (255, 255, 255)
152     #     screen.blit(self.Band_M.text_surface, (self.Band_x - self.Band_M.text_surface.get_width(), 55+ self.y+ self.B
153
154
155
156
157
158 p_store = np.zeros((50000,2))
159
160
161 pygame.init()
162 pygame.mixer.init()
163
164 # 生产窗体以及窗口标题
165 # 颜色：深蓝带黄
166 WIDTH , HEIGHT = 2000, 1200
167 screen = pygame.display.set_mode([WIDTH, HEIGHT])
168
169 pygame.display.set_caption("Car Driving")
170
171 # 添加系统时钟，用于设置帧的刷新
172 FPS = 120
173 clock = pygame.time.Clock()
174
175
176 x = 0
177
178 y = 300
179
180 z = 1
181
182 state = 0
183 statev = 0
184 count = 1
185 flag = 1
186
187 # pygame.display.flip() # 画面的刷新
188
189
190 # Car_Sprites = Car(WIDTH, HEIGHT)
191 # Car_Sprites.x = 1500
```

Console 1/1

速度
29.09米/秒
速度: 29.096米/秒
速度: 29.701米/秒
速度: 29.705米/秒
速度: 29.71米/秒
速度: 29.714米/秒
速度: 29.719米/秒
速度: 29.723米/秒
速度: 29.728米/秒
速度: 29.733米/秒
速度: 29.738米/秒
速度: 29.743米/秒
速度: 29.747米/秒

<Event(708-keydown {'unicode': 'x1b', 'key': 27, 'mod': 4096, 'scancode': 41, 'window': None}>

In [4]:

conda: xiaokang (Python 3.8.10) USP Python: down Line 147, Col 50 UTF-8 CRLF Mon 38 18°C 23:19

虚拟仿真平台($K_1 = 3$, 改进控制器, 有干扰)

- 控制系统的稳态性能分析：
 - 概念：稳态误差（系统实际输出值与希望输出值之间的最终偏差）
 - 计算：**终值定理求解稳态误差**
 - 案例：**车辆定速巡航控制综合设计**
 - 思考：如何改善车辆的动态性能呢？
- 作业：
 - **作业3.8和3.10**