

# Tema 5

## Diseño de controladores

---

Javier Valls

Dpto. Ingeniería Electrónica

1. Introducción y objetivos
2. Representación vectorial de números complejos
3. El lugar de las raíces
4. Diseño de controladores en sistemas continuos
5. Control Proporcional (P)
6. Controlador proporcional e integral (PI)
7. Controlador proporcional y derivativo (PD)
8. Controlador proporcional, integral y derivativo (PID)
9. Implementación de controladores en sistemas discretos
10. Conclusiones
11. Bibliografía

# Introducción y objetivos

---

# Introducción

Respuesta transitoria de un sistema

→ depende de sus polos

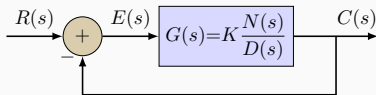
Sistema  $G(s) = K \frac{N(s)}{D(s)}$

→ polos en  $D(s) = 0$

Sistema realimentado

$$T(s) = \frac{C(s)}{R(s)} = \frac{G(s)}{1 + G(s)} = \frac{K \frac{N(s)}{D(s)}}{1 + K \frac{N(s)}{D(s)}} = \frac{KN(s)}{D(s) + KN(s)}$$

→ polos en  $D(s) + KN(s) = 0$



¿Cómo son los polos de la función de transferencia  $T(s)$ ?

¿Cómo cambian los polos de  $T(s)$  al cambiar el parámetro  $K$ ?

¿Puede llegar a ser inestable  $T(s)$  para algún valor de  $K$ ?

¿Cómo conseguir que el sistema tenga la respuesta deseada?

**Lugar de las raíces:** representación gráfica de los polos del lazo cerrado al variar un parámetro del sistema

- método de análisis y diseño para determinar la respuesta transitoria y la estabilidad
- permite describir cualitativamente cómo es la respuesta transitoria del lazo al variar un parámetro
- permite visualizar gráficamente el margen de estabilidad

**Inclusión de un controlador en el lazo:**

- variar un parámetro no proporciona la suficiente flexibilidad para conseguir cualquier tipo de respuesta
- es necesario incluir un controlador que añada polos/ceros al sistema para modificar la respuesta de la forma deseada

Al finalizar este tema, el alumno debe ser capaz de:

- entender el método del lugar de las raíces y sus reglas básicas para dibujarlo
- representar con Matlab el lugar de las raíces de sistemas continuos y discretos
- analizar y diseñar sistemas con control proporcional, integral y derivativo en el dominio continuo y discreto
- implementar un controlador digital a partir de su función de transferencia

## Representación vectorial de números complejos

---

# Repaso: operaciones básicas con complejos

**Numero complejo:**  $p = a + jb = M\angle\theta$

Conv. coordenadas cartesianas a polares:

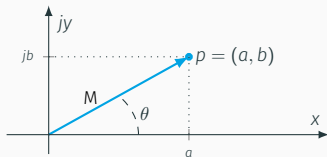
- $M = \sqrt{a^2 + b^2}$
- $\theta = \text{atan}(b/a)$

Conv. coordenadas polares a cartesianas:

- $a = \text{Re}\{p\} = M \cdot \cos(\theta)$
- $b = \text{Im}\{p\} = M \cdot \sin(\theta)$

Operaciones ( $p_1 = a_1 + jb_1 = M_1\angle\theta_1$ ,  $p_2 = a_2 + jb_2 = M_2\angle\theta_2$ ):

- $p_1 \pm p_2 = (a_1 \pm a_2) + j(b_1 \pm b_2)$
- $p_1 \cdot p_2 = (M_1 \cdot M_2)\angle(\theta_1 + \theta_2)$
- $\frac{p_1}{p_2} = \frac{M_1}{M_2}\angle(\theta_1 - \theta_2)$



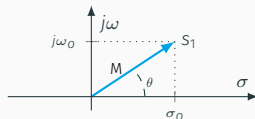


# Representación vectorial de números complejos

**Numero complejo:**  $s_1 = \sigma_0 + j\omega_0 = M\angle\theta$

Vector  $s_1$ :

- origen  $\rightarrow (0, 0)$ , final  $\rightarrow (\sigma_0, \omega_0)$

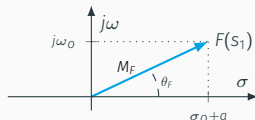


**Función de variable compleja:**  $F(s)=s+a$

N. compl.  $F(s_1)=F(s)|_{s=s_1}=(\sigma_0+a)+j\omega_0=M_F\angle\theta_F$

Vector  $F(s_1)$ :

- origen  $\rightarrow (0, 0)$ , final  $\rightarrow (\sigma_0+a, \omega_0)$



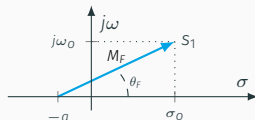
**Representación alternativa de  $F(s)=s+a$**

$F(s)$  tiene un cero en  $-a$

$\rightarrow$  desplazar  $-a$  unidades  $F(s_1)$

Vector  $F(s_1)$ :

- origen  $\rightarrow (-a, 0)$ , final  $\rightarrow (\sigma_0, \omega_0)$



# Representación vectorial de números complejos

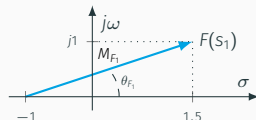
Representación alternativa de  $F(s)=s+a$  evaluada en un punto  $s$   
Vector:

- origen en el cero  $(-a)$
- final en el punto  $s$

**E** Función  $F_1(s)=s+1$  evaluada en  $s_1=1.5+j$

Vector  $F(s_1)$ ;

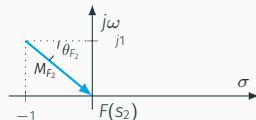
- origen  $\rightarrow (-1, 0)$ , final  $\rightarrow (1.5, 1)$



**E** Función  $F_2(s)=s+(1-j)$  evaluada en  $s_2=0$

Vector:

- origen  $\rightarrow (-1, 1)$ , final  $\rightarrow (0, 0)$



# Evaluación en un punto de funciones con múltiples ceros y polos

Función de transferencia:  $F(s) = \frac{\prod_{i=1}^m (s + z_i)}{\prod_{j=1}^n (s + p_j)}$

$$F(s_1) = F(s)|_{s_1} = \frac{\prod_{i=1}^m (s_1 + z_i)}{\prod_{j=1}^n (s_1 + p_j)} = M_{s_1} \angle \theta_{s_1}$$

$$M_{s_1} = \frac{\prod_{i=1}^m M_{z_i}}{\prod_{j=1}^n M_{p_j}}$$

$$\theta_{s_1} = \sum_{i=1}^m \theta_{z_i} - \sum_{j=1}^n \theta_{p_j}$$

**E** Evaluación de una función  $F(s) = \frac{(s+1)}{s(s+2)}$  en el punto  $s_1 = -3 + j4$

$$F(s) = \frac{(s+1)}{s(s+2)} = \frac{F_{z_1}(s)}{F_{p_1}(s)F_{p_2}(s)} \xrightarrow{s_1 = -3+j4} F(s)|_{s_1} = M_{s_1} \angle \theta_{s_1}$$

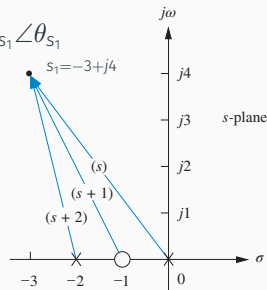
$$F_{z_1}(s_1) = (s+1)|_{s_1} \rightarrow M_{z_1} \angle \theta_{z_1} = 4.47 \angle 116.6^\circ$$

$$F_{p_1}(s_1) = (s)|_{s_1} \rightarrow M_{p_1} \angle \theta_{p_1} = 5 \angle 126.9^\circ$$

$$F_{p_2}(s_1) = (s+2)|_{s_1} \rightarrow M_{p_2} \angle \theta_{p_2} = 4.12 \angle 104^\circ$$

$$M_{s_1} = \frac{M_{z_1}}{M_{p_1} \cdot M_{p_2}} = 0.217$$

$$\theta_{s_1} = \theta_{z_1} - (\theta_{p_1} + \theta_{p_2}) = -114^\circ$$



## El lugar de las raíces

---

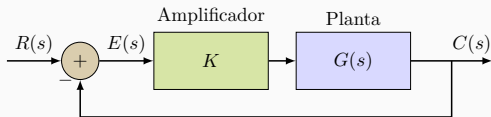
- El lugar de las raíces: definición y ejemplo
- Propiedades
- Reglas básica para dibujar el lugar de las raíces
- El lugar de las raíces en sistemas discretos

# El lugar de las raíces

**Lugar de las raíces:** representación gráfica de los polos del lazo cerrado al variar un parámetro del sistema

**Función de transferencia:**

$$T(s) = \frac{C(s)}{R(s)} = \frac{K \cdot G(s)}{1 + K \cdot G(s)}$$



Polos del lazo cerrado  $\longrightarrow 1 + K \cdot G(s) = 0$

**E** Sistema de posicionamiento de una cámara

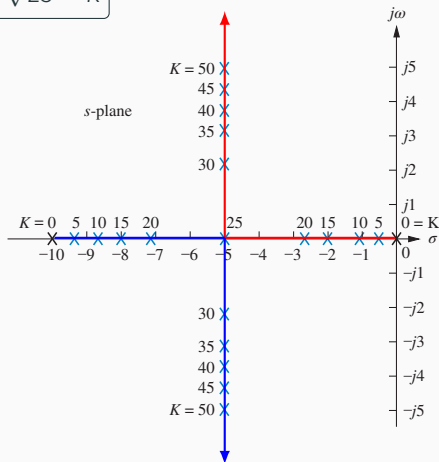
$$G(s) = \frac{1}{s \cdot (s + 10)} \longrightarrow T(s) = \frac{C(s)}{R(s)} = \frac{K}{s^2 + 10 \cdot s + K}$$

Polos de  $T(s)$ :  $s^2 + 10 \cdot s + K = 0 \longrightarrow s_{1,2} = -5 \pm \sqrt{25 - K}$

## E Sistema de posicionamiento de una cámara

Polos de  $T(s)$   $\longrightarrow$   $s_{1,2} = -5 \pm \sqrt{25 - K}$

$K$	Pole 1	Pole 2
0	-10	0
5	-9.47	-0.53
10	-8.87	-1.13
15	-8.16	-1.84
20	-7.24	-2.76
25	-5	-5
30	$-5 + j2.24$	$-5 - j2.24$
35	$-5 + j3.16$	$-5 - j3.16$
40	$-5 + j3.87$	$-5 - j3.87$
45	$-5 + j4.47$	$-5 - j4.47$
50	$-5 + j5$	$-5 - j5$



## Propiedades del lugar de las raíces

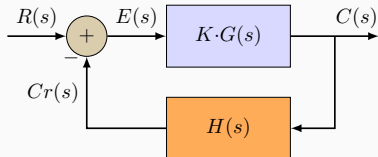
---



# Propiedades del lugar de las raíces

Función de transferencia del sistema en **lazo cerrado**:

$$T(s) = \frac{C(s)}{R(s)} = \frac{K \cdot G(s)}{1 + K \cdot H(s) \cdot G(s)}$$



Función de transferencia en **lazo abierto**:  $T_{OL}(s) = K \cdot G(s) \cdot H(s)$

Polos de  $T(s)$ :  $1 + K \cdot G(s) \cdot H(s) = 1 + T_{OL}(s) = 0$

$$K \cdot G(s) \cdot H(s) = -1 \begin{cases} |K \cdot G(s) \cdot H(s)| = 1 \\ \angle(G(s) \cdot H(s)) = (2i + 1) \cdot 180^\circ \text{ con } i \in \mathbb{Z} \end{cases}$$

**Lugar de las raíces:** Valores de  $s$  que cumplen  $\angle(T_{OL}(s)) = (2i + 1) \cdot 180^\circ$

→ Siempre se encontrará en esos puntos un  $K = \frac{1}{|G(s) \cdot H(s)|}$

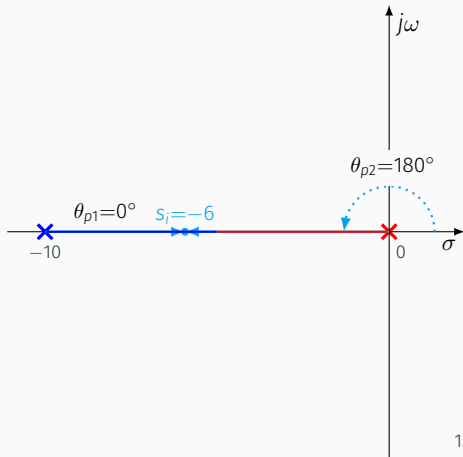
# Propiedades del lugar de las raíces

## E Sistema de posicionamiento de una cámara

$$G(s) = \frac{1}{s(s+10)} \rightarrow T(s) = \frac{C(s)}{R(s)} = \frac{K}{s^2 + 10s + K} \rightarrow s_{1,2} = -5 \pm \sqrt{25 - K}$$

Puntos  $s_i = \sigma_i$  con  $-10 \leq \sigma_i \leq 0$

$$\angle G(s_i) = -(\theta_{p1} + \theta_{p2}) = -180^\circ$$



# Propiedades del lugar de las raíces

## E Sistema de posicionamiento de una cámara

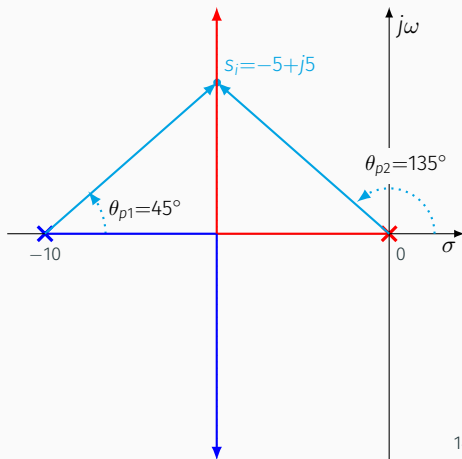
$$G(s) = \frac{1}{s \cdot (s+10)} \rightarrow T(s) = \frac{C(s)}{R(s)} = \frac{K}{s^2 + 10 \cdot s + K} \rightarrow s_{1,2} = -5 \pm \sqrt{25 - K}$$

Puntos  $s = \sigma_i$  con  $-10 \leq \sigma_i \leq 0$

$$\angle G(s_i) = -(\theta_{p1} + \theta_{p2}) = -180^\circ$$

Puntos  $s_i = -5 \pm j\omega_i$

$$\angle G(s_i) = -(\theta_{p1} + \theta_{p2}) = -180^\circ$$



# Propiedades del lugar de las raíces

## E Sistema de posicionamiento de una cámara

$$G(s) = \frac{1}{s \cdot (s+10)} \rightarrow T(s) = \frac{C(s)}{R(s)} = \frac{K}{s^2 + 10 \cdot s + K} \rightarrow s_{1,2} = -5 \pm \sqrt{25 - K}$$

Puntos  $s = \sigma_i$  con  $-10 \leq \sigma_i \leq 0$

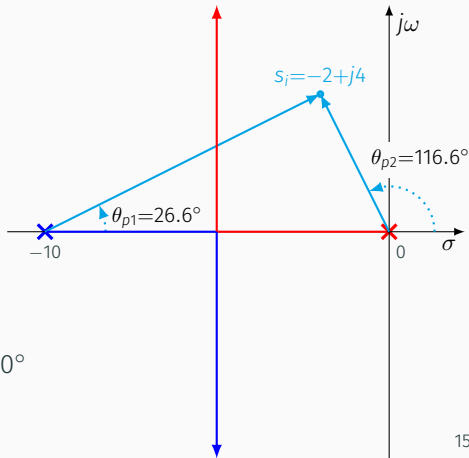
$$\angle G(s_i) = -(\theta_{p1} + \theta_{p2}) = -180^\circ$$

Puntos  $s_i = -5 \pm j\omega_i$

$$\angle G(s_i) = -(\theta_{p1} + \theta_{p2}) = -180^\circ$$

Resto de puntos

$$\angle G(s_i) = -(\theta_{p1} + \theta_{p2}) \neq (2i + 1)180^\circ$$



## Reglas básicas para dibujar el lugar de la raíces

---

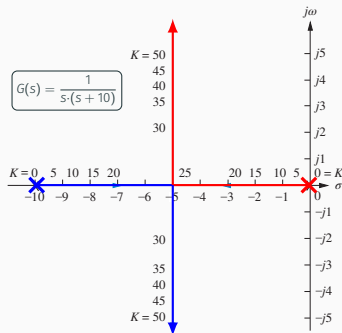
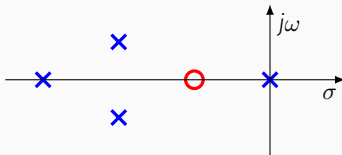
# Reglas básicas para dibujar el lugar de las raíces

Hay 5 reglas que nos permiten esbozar el lugar de las raíces:

1.- El **número de ramas** es igual al número de polos del lazo cerrado

2.- Es **simétrico respecto al eje real**

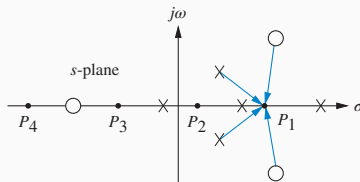
→ Los ceros y polos de la función de transferencia solo pueden ser reales o complejos conjugados



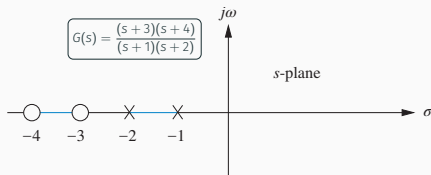
# Reglas básicas para dibujar el lugar de la raíces

## 3.- Comportamiento en el eje real

→ Los polos o ceros complejos conjugados no contribuyen (anulan sus ángulos en el eje real)



→ En el eje real el lugar de las raíces existe con  $K>0$  a la izquierda de un número impar de ceros o polos



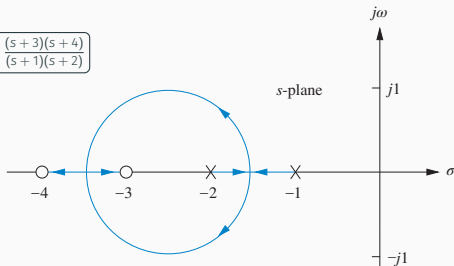
# Reglas básicas para dibujar el lugar de las raíces

- 4.- Las ramas del lugar de las raíces empiezan ( $K=0$ ) en los polos y finalizan ( $K=\infty$ ) en los ceros de la función de transferencia en lazo abierto

$$T(s) = \frac{G(s)}{1+G(s)} = \frac{K \frac{N(s)}{D(s)}}{1 + K \frac{N(s)}{D(s)}} = \frac{KN(s)}{D(s) + KN(s)}$$

Polos:  $D(s) + KN(s) = 0$   $\begin{cases} \text{Si } K \rightarrow 0 \Rightarrow \text{polos : } D(s) + KN(s) \rightarrow D(s) \\ \text{Si } K \rightarrow \infty \Rightarrow \text{polos : } D(s) + KN(s) \rightarrow KN(s) \end{cases}$

$$G(s) = \frac{(s+3)(s+4)}{(s+1)(s+2)}$$





# Reglas básicas para dibujar el lugar de la raíces

## 5.- Comportamiento asintótico:

Las funciones de transferencia tienen el mismo número de polos y ceros, si se cuentan los que tienen en el infinito

→  $T(s)$  tiene un polo en  $\infty$  si

$$\lim_{s \rightarrow \infty} T(s) = \infty$$

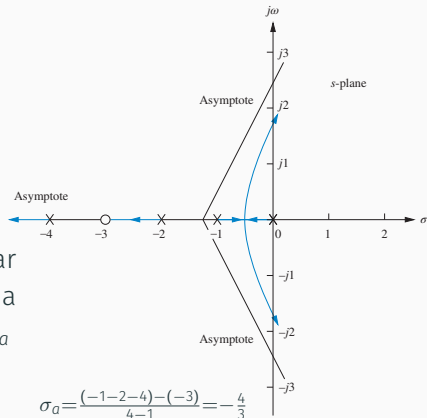
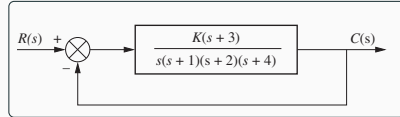
→  $T(s)$  tiene un cero en  $\infty$  si

$$\lim_{s \rightarrow \infty} T(s) = 0$$

Cuando se tiene un cero en  $\infty$  el lugar de las raíces tiende asintóticamente a una recta con origen en  $-\sigma_a$  y pendiente  $\pm\theta_a$

$$\rightarrow \sigma_a = \frac{\sum p_i - \sum z_j}{\# \text{polos finitos} - \# \text{ceros finitos}}$$

$$\rightarrow \theta_a = \frac{(2k+1)\pi}{\# \text{polos finitos} - \# \text{ceros finitos}}$$



$$\sigma_a = \frac{(-1-2-4)-(-3)}{4-1} = -\frac{4}{3}$$

$$\theta_a = \frac{(2k+1)\pi}{3} \text{ con } k = 0, 1 \text{ y } 2$$



# El lugar de las raíces con Matlab

```
% Definición de la función
% de transferencia G(s)
>> G=tf([1,5],[1,2,10])
```

G =

$$\frac{s + 5}{s^2 + 2s + 10}$$

Continuous-time tr. func.

```
% Lugar de las raíces
```

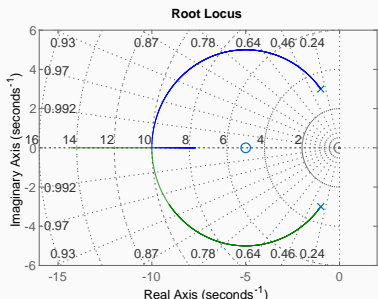
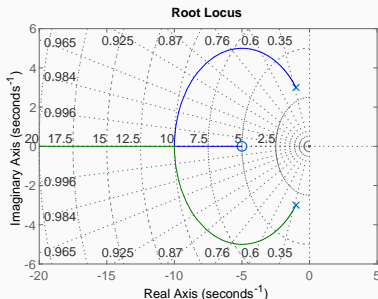
```
>> rlocus(G)
```

```
>> sgrid
```

```
% rlocus para valores de
% ganancia indicados en k
```

```
>> k=(0:0.01:20);
```

```
>> rlocus(G,k)
```



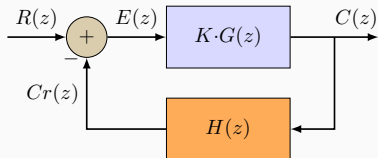
## El lugar de las raíces en sistemas discretos

---

# El lugar de las raíces en sistemas discretos

Función de transferencia del sistema en **lazo cerrado**:

$$T(z) = \frac{C(z)}{R(z)} = \frac{K \cdot G(z)}{1 + K \cdot H(z) \cdot G(z)}$$



Función de transferencia en **lazo abierto**:  $T_{OL}(z) = K \cdot G(z) \cdot H(z)$

Polos de  $T(z)$ :  $1 + K \cdot G(z) \cdot H(z) = 1 + T_{OL}(z) = 0$

$$K \cdot G(z) \cdot H(z) = -1 \begin{cases} |K \cdot G(z) \cdot H(z)| = 1 \\ \angle(G(z) \cdot H(z)) = (2i + 1) \cdot 180^\circ \text{ con } i \in \mathbb{Z} \end{cases}$$

**Lugar de las raíces:** Valores de  $s$  que cumplen  $\angle(T_{OL}(z)) = (2i + 1) \cdot 180^\circ$

- Las mismas reglas de dibujo que en el plano  $s$
- Solo cambia su uso/interpretación en el plano  $z$



# El lugar de las raíces con Matlab

% Definición de la función  
% de transferencia  $G(z)$

```
>> T=1;  
>> G=zpk([], [0.25 0.75], 1, T)  
G =
```

$$\frac{1}{(z-0.25)(z-0.75)}$$

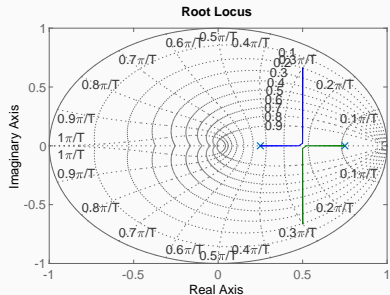
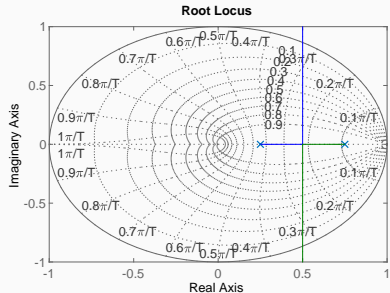
Sample time: 1 seconds  
Discrete-time z/p/gain model.

% Lugar de las raíces

```
>> rlocus(G)  
>> zgrid
```

% rlocus para valores de  
% ganancia indicados en k

```
>> k=(0:1e-3:0.5);  
>> rlocus(G,k)  
>> zgrid
```



# Diseño de controladores en sistemas continuos

---

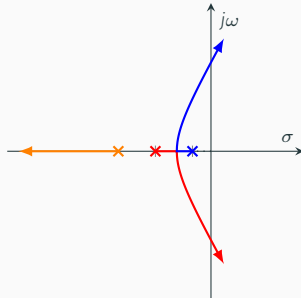
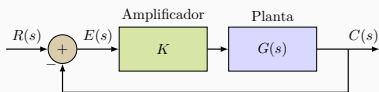
# Diseño de la respuesta transitoria mediante el lugar de las raíces

**Objetivo:** seleccionar la respuesta transitoria a través de la elección de la ubicación de los polos en lazo cerrado

Modificar el valor de la constante de proporcionalidad en el lazo permite moverse por el lugar de las raíces

→ **Control proporcional**

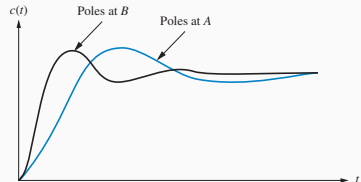
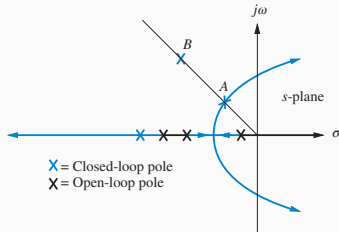
- Poca flexibilidad para forzar un tipo de respuesta
- Al variar la  $K$  mejoramos algún parámetro y empeoramos otro
  - compromiso entre la velocidad de la respuesta y error en estado estacionario



# Modificación del lugar de las raíces

Si queremos mejorar un parámetro del sistema manteniendo otro hay que modificar la ubicación de los polos del sistema

## E Mejora del tiempo de respuesta manteniendo el OS%

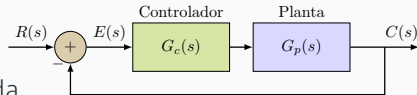


→ incluir un **controlador (compensador)** que añada polos y/o ceros para modificar la respuesta de la forma deseada



## ¿Cómo diseñar el controlador?

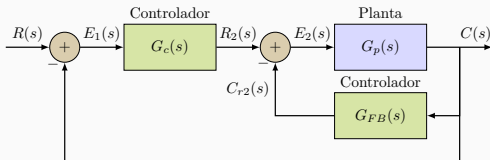
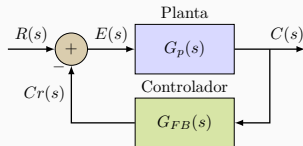
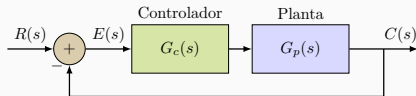
- Añadir un cero en el sistema desplaza los polos hacia la izquierda
  - mejora el tiempo de respuesta
  - **Controlador proporcional derivativo (PD)**
- Añadir un polo en el origen incrementa el tipo de sistema
  - reduce el error en estado estacionario
  - puede deteriorar el tiempo de respuesta
  - **Controlador proporcional integral (PI)**
- Añadir un polo en el origen y un cero permite combinar ambos efectos
  - reduce el error en estado estacionario
  - mejorar el tiempo de respuesta
  - **Controlador proporcional integral y derivativo (PID)**



# Controladores

## ¿Dónde se ubica el controlador?

- En el camino directo  
→ Controlador en cascada
- En el camino de realimentación  
→ Controlador "feedback"
- Otras topologías  
→ Controlador en cascada combinado con otro realimentado en el lazo interno



**A** En esta asignatura nos centramos solo en los controladores en cascada

# Controladores en sistemas discretos

Las acciones de control y estrategias de diseño son similares a las de los sistemas continuos

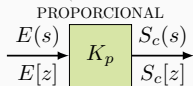
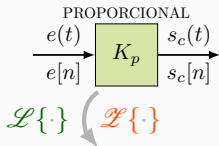
- **Control P:** mejorar la respuesta desplazándose por el lugar de las raíces
- **Control PI:** aumentar el tipo de sistema para anular el error en estado estacionario
  - Se añade un polo en  $z=1$  (un integrador) y un cero cercano a él para cancelar su efecto sobre la respuesta
- **Control PD:** desplazar los polos para mejorar el tiempo de respuesta
  - Se añade un cero para modificar el lugar de las raíces
  - Diferencia con los sistemas continuos → hay que necesario añadir un polo en  $z=0$  para que el controlador sea implementable
- **Control PID:** Combina las estrategias anteriores para beneficiarse de ambas acciones de control

## Control proporcional (P)

---

# Controlador P

Controlador P:



Salida del controlador P:  
proporcional al error

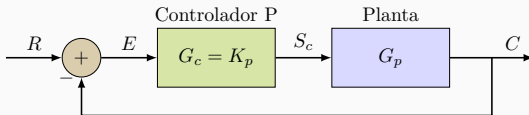
- Sist. continuos: error  $s_c(t)=s_p(t)=K_p e(t)$ ,
- Sist. discretos: error  $s_c[n]=s_p[n]=K_p e[n]$ ,

$$G_c(s) = \frac{S_c(s)}{E(s)} = K_p$$

$$G_c(z) = \frac{S_c(z)}{E(z)} = K_p$$

•  $K_p$ : constante de proporcionalidad

Sistema realimentado con controlador P:



El controlador P solo tiene ganancia (no tiene polos/ceros)

→ Cambiar  $K_p$  cambia los polos del sistema al moverse por el lugar de las raíces

# Acción de control P

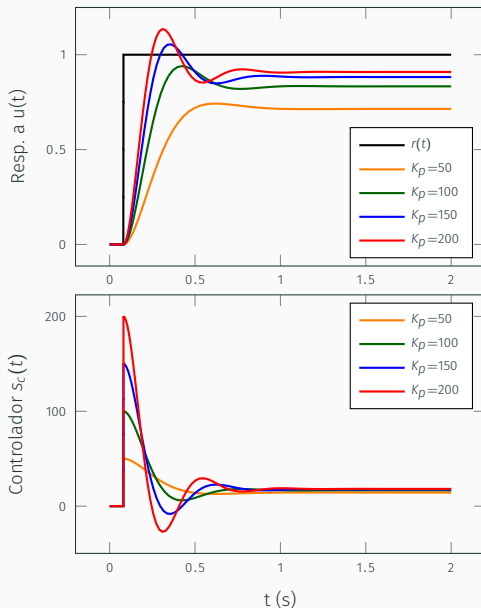
Control P:

$$s_c(t) = K_p e(t)$$

$\uparrow\uparrow K_p \rightarrow \uparrow\uparrow$  actuación con el mismo error

- $\uparrow$  tiempo de respuesta
- $\uparrow\uparrow$  oscilaciones
- $\downarrow\downarrow e_s(\infty)$ , pero no se anula

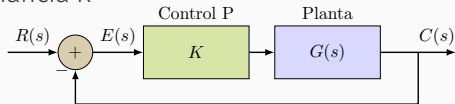
**E** Control P en el sist. Tipo 0



# Diseño del controlador P conociendo $G_p(s)$

El controlador solo aplica una ganancia  $K$

$$T(s) = \frac{C(s)}{R(s)} = \frac{K \cdot G(s)}{1 + K \cdot G(s)}$$



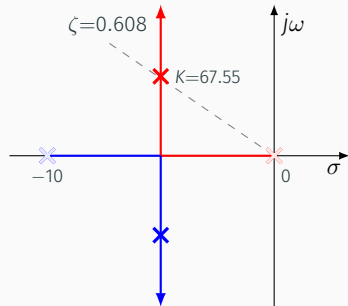
## E Control P en el sistema de posicionamiento de una cámara

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

Especificación:  $OS\% = 9\% \rightarrow \zeta = 0.608$

$$\downarrow K=67.55$$

$$T(s) = \frac{C(s)}{R(s)} = \frac{67.55 \cdot G(s)}{1 + 67.55 \cdot G(s)}$$



# Control proporcional (P)

Si el orden del lazo es mayor que 2:

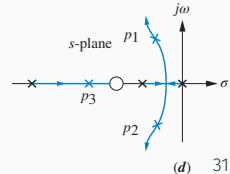
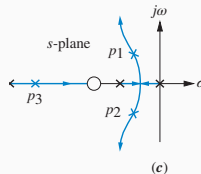
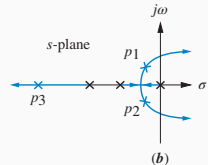
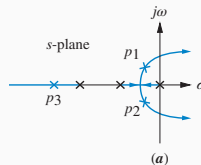
- cambiar  $K$  para aproximar la respuesta por la de 2º orden
- utilizar las fórmulas que facilitan su especificación ( $T_s, T_p, \%OS$ )

Aproximaciones a la respuesta de 2º orden:

→ Polos alejados (más a la izquierda en el plano- $s$ ) de los polos dominantes (al menos 5 veces)

→ Cancelación polo-cero

**Método limitado a las raíces existente en el lugar de las raíces**



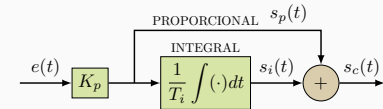


## Controlador proporcional e integral (PI)

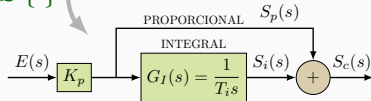
---

# Controlador PI en sistemas continuos

Controlador PI:



$\mathcal{L}\{\cdot\}$



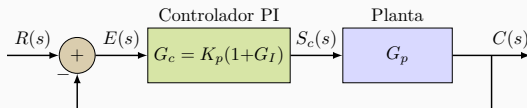
Salida del controlador PI:  $s_c(t) = s_p(t) + s_i(t)$   
proporcional al

- error  $s_p(t) = K_p e(t)$ ,
- y al error acumulado  $s_i(t) = \frac{K_p}{T_i} \int e(t) dt$

$$G_c(s) = \frac{S_c(s)}{E(s)} = K_p \left( 1 + \frac{1}{T_i s} \right)$$

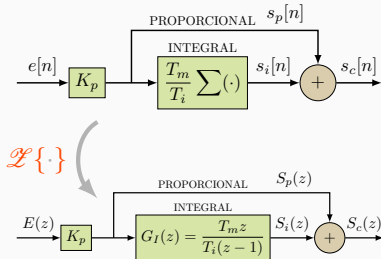
- $K_p$ : constante de proporcionalidad
- $T_i$ : constante de tiempo integral

Sistema realimentado con controlador PI:



# Controlador PI en sistemas discretos

Controlador PI:



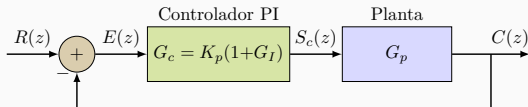
Salida del controlador PI:  $s_c[n] = s_p[n] + s_i[n]$   
proporcional al

- error  $s_p[n] = K_p e[n]$ ,
- y al error acumulado  
$$s_i[n] = \frac{K_p T_m}{T_i} \sum e[n] = s_i[n-1] + \frac{K_p T_m}{T_i} e[n]$$

$$G_c(z) = \frac{S_c(z)}{E(z)} = K_p \left( 1 + \frac{T_m z}{T_i(z-1)} \right)$$

- $K_p$ : constante de proporcionalidad
- $T_i$ : constante de tiempo integral
- $T_m$ : periodo de muestreo

Sistema realimentado con controlador PI:



# Acción de control PI

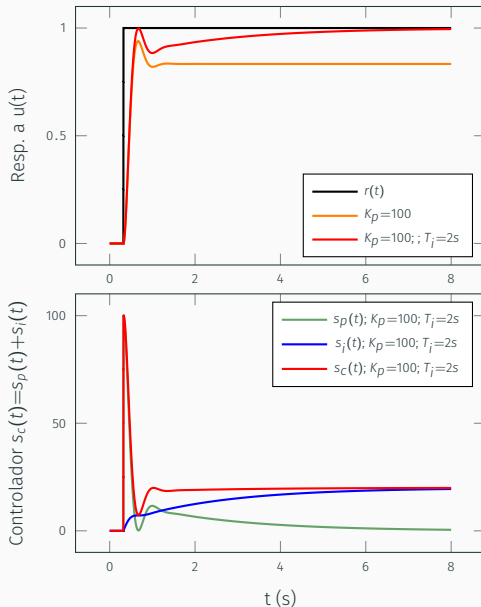
Control PI:

$$\begin{aligned} s_c(t) &= s_p(t) + s_i(t) \\ &= K_p e(t) + \frac{K_p}{T_i} \int e(t) dt \end{aligned}$$

La componente I del error crece (se acumula) hasta que elimina el error

$$e_s(\infty) \rightarrow 0$$

**E** Control PI en el sist. Tipo 0



# Acción de control PI

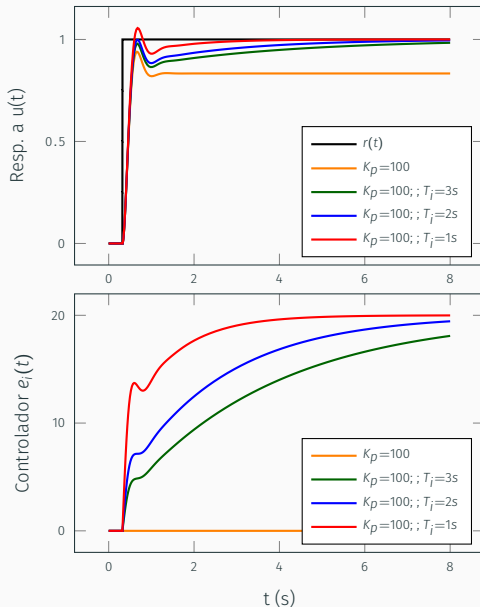
Control PI:

$$\begin{aligned} s_c(t) &= s_p(t) + s_i(t) \\ &= K_p e(t) + \frac{K_p}{T_i} \int e(t) dt \end{aligned}$$

Transitorio similar al del control P

- $\uparrow\uparrow \frac{K_p}{T_i}$ :
  - $\uparrow$  tiempo de respuesta
  - $\uparrow\uparrow$  oscilaciones

$T_i$  proporcional al tiempo que tarda en anular el error (3 o 4 veces)



# Polos y ceros del controlador PI

## Controlador PI en sist. continuo:

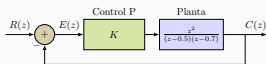
$$G_c(s) = K_p \left( 1 + \frac{1}{T_i s} \right) = K \left( \frac{s+a}{s} \right)$$

- 1 polo en  $p = 0$  (1 integrador)  
→ incrementa tipo de sistema
- 1 cero en  $-a = -1/T_i$   
→  $T_i \uparrow \Rightarrow a \rightarrow 0$   
cero muy cercano al polo  
no modifica la respuesta  
→  $T_i \downarrow \Rightarrow a$  se aleja de  $p$  hacia  $-\infty$   
integra más rápido  
modifica la respuesta
- $K = K_p$

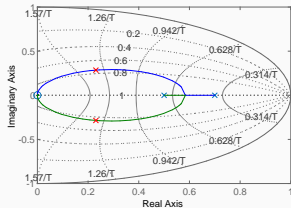
## Controlador PI en sist. discreto:

$$G_c(z) = K_p \left( 1 + \frac{T_m z}{T_i (z-1)} \right) = K \left( \frac{z-a}{z-1} \right)$$

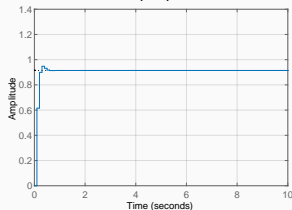
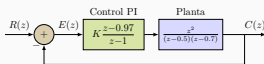
- 1 polo en  $p = 1$  (1 integrador)  
→ incrementa tipo de sistema
- 1 cero en  $a = 1/(1 + T_m/T_i)$   
→  $T_i \uparrow \Rightarrow a \rightarrow 1$   
cero muy cercano al polo  
no modifica la respuesta  
→  $T_i \downarrow \Rightarrow a$  se aleja de  $p$  hacia 0  
integra más rápido  
modifica la respuesta
- $K = K_p/a$

Control P:  $K=1$ 

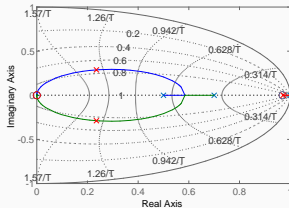
Root Locus



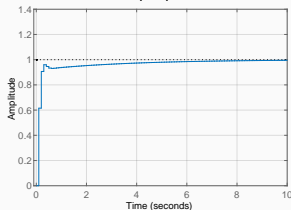
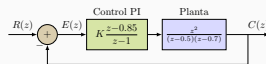
Step Response

Control PI:  $K=1, a=0.97, T_i=3.2$ 

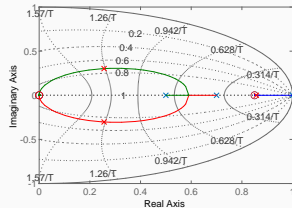
Root Locus



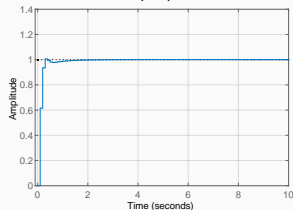
Step Response

Control PI:  $K=1, a=0.85, T_i=0.5$ 

Root Locus



Step Response



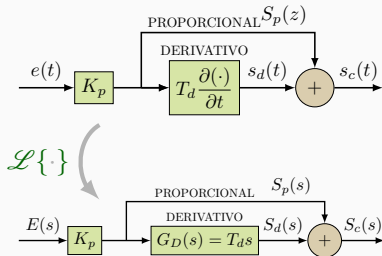
## Controlador proporcional y derivativo (PD)

---



# Controlador PD en sistemas continuos

Controlador PD:



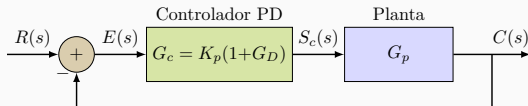
Salida del controlador PD:  $s_c(t) = s_p(t) + s_d(t)$   
proporcional al

- error  $s_p(t) = K_p e(t)$ ,
- y a la velocidad de cambio del error  
 $s_d(t) = K_p T_d \frac{\partial e(t)}{\partial t}$

$$G_c(s) = \frac{S_c(s)}{E(s)} = K_p (1 + T_d s)$$

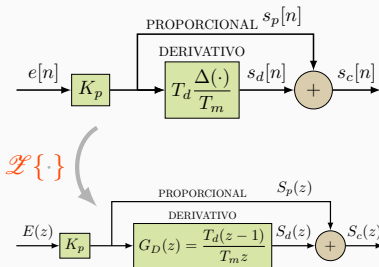
- $K_p$ : constante de proporcionalidad
- $T_d$ : constante de tiempo derivativa

Sistema realimentado con controlador PD:



# Controlador PD en sistemas discretos

Controlador PD:



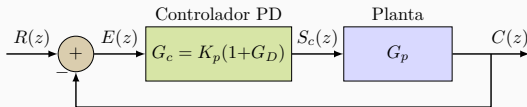
Salida del controlador PD:  $s_c[n] = s_p[n] + s_d[n]$   
proporcional al

- error  $s_p[n] = K_p e[n]$ ,
- y a la velocidad de cambio del error  
$$s_d(t) = K_p T_d \frac{\Delta e[n]}{T_m} = K_p T_d \frac{e[n] - e[n-1]}{T_m}$$

$$G_c(z) = \frac{S_c(z)}{E(z)} = K_p \left( 1 + \frac{T_d(z-1)}{T_m z} \right)$$

- $K_p$ : constante de proporcionalidad
- $T_d$ : constante de tiempo derivativa
- $T_m$ : periodo de muestreo

Sistema realimentado con controlador PD:

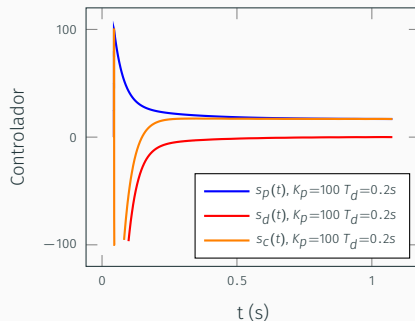
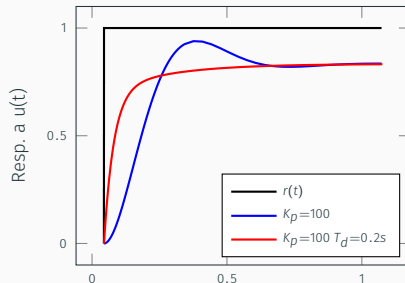
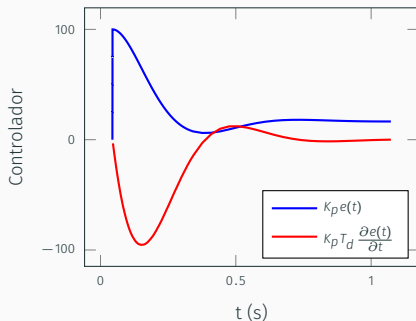


# Acción del control PD

Control PD:

$$\begin{aligned} s_c(t) &= s_p(t) + s_d(t) \\ &= K_p e(t) + K_p T_d \frac{\partial e(t)}{\partial t} \end{aligned}$$

La componente derivativa del error aumenta con la velocidad de cambio del error.



# Acción del control PD

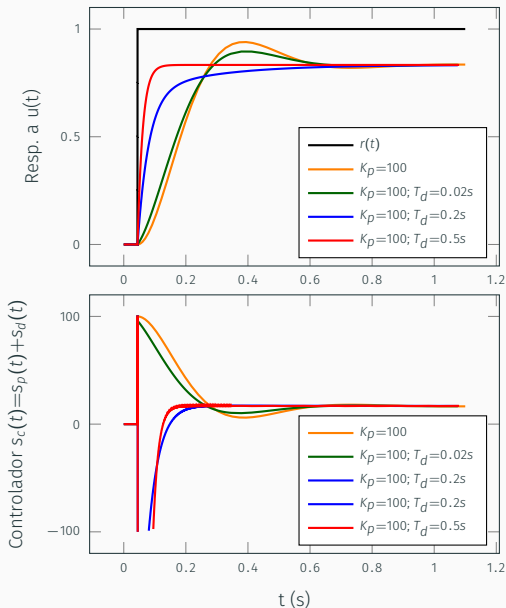
Control PD:

$$\begin{aligned}s_c(t) &= s_p(t) + s_d(t) \\ &= K_p e(t) + K_p T_d \frac{\partial e(t)}{\partial t}\end{aligned}$$

La componente derivativa del error aumenta con la velocidad de cambio del error.

Tiene efecto anticipativo, en proporción  $T_d$

Reduce la oscilación



# Polos y ceros del controlador PD

Controlador PD en sist. continuo:

$$G_c(s) = K_p(1 + T_d s) = K(s + a)$$

- 1 cero en  $-a = -1/T_d$
- $K = K_p/a$

Controlador PD en sist. discreto:

$$G_c(z) = K_p \left( 1 + \frac{T_d}{T_m} \frac{(z-1)}{z} \right) = K \left( \frac{z-a}{z} \right)$$

- 1 polo en  $p = 0$
- 1 cero en  $a = 1/(1 + T_d/T_m)$
- $K = K_p/a$

Diseño del PD si se conocen los polos/ceros de la planta ( $G_p$ )

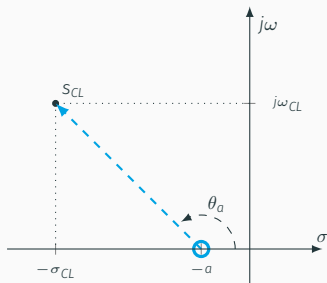
- Tiempo de establecimiento  $T_s = 4/\sigma$  ( $\sigma \rightarrow$  parte real del polo en un sist. continuo)
- En sist. continuos  $T_s \downarrow$  desplazando los polos dominantes hacia la izquierda ( $\sigma \uparrow$ )
- En sist. discreto  $T_s$  disminuye reduciendo el radio los polos dominantes ( $|p_z| = |e^{-\sigma T_m}| \downarrow$  con  $\sigma \uparrow$ )

Se puede encontrar el valor del cero "a" para modificar la ubicación de los polos del sistema realimentado y mejorar  $T_s$

# Sistemas continuos: diseño del PD conociendo $G_p(s)$

Queremos que el lugar de las raíces pase por el punto  $s_{CL} = -\sigma_{CL} + j\omega_{CL}$

→ Hemos de colocar un cero en  $-a$



$$\text{Como } \tan(\theta_a) = \frac{\omega_{CL}}{-(\sigma_{CL} - a)}$$

$s_{CL}$  cumple la condición de fase:

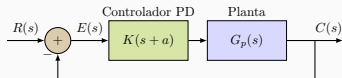
$$\theta_a + \theta_{OL} = (2i+1) \cdot \pi$$

$\theta_{OL}$ : ángulo de la función en lazo abierto en  $s_{CL}$  (sin el controlador)

$$(1^\circ) \quad \theta_a = (2i+1) \cdot \pi - \theta_{OL}$$

$$(2^\circ) \quad a = \frac{\omega_{CL}}{\tan(\theta_a)} + \sigma_{CL}$$

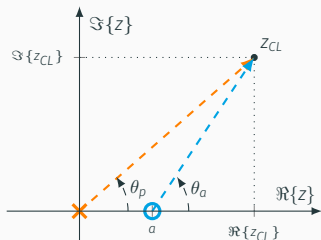
El valor de  $K$  se encuentra aplicando la condición del módulo del lugar de las raíces:  $|G_c(s_{CL}) \cdot G_p(s_{CL})| = 1$ , con  $G_c(s) = K(s+a)$



# Sistemas discretos: diseño del PD conociendo $G_p(z)$

Queremos que el lugar de las raíces pase por el punto  $z_{CL}$

- Hemos de colocar un cero en  $a$
- Añadimos un polo en 0



$$\text{Como } \tan(\theta_a) = \frac{\Im\{z_{CL}\}}{\Re\{z_{CL}\} - a}$$

$z_{CL}$  cumple la condición de fase:

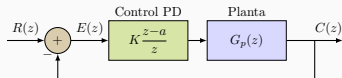
$$\theta_a - \theta_p + \theta_{OL} = (2i+1) \cdot \pi$$

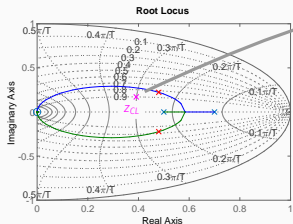
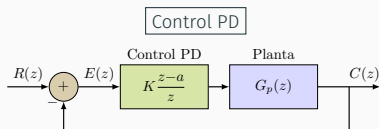
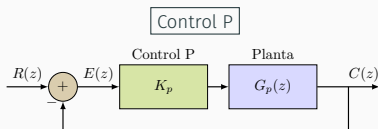
$\theta_{OL}$ : ángulo de la función  
en lazo abierto en  $z_{CL}$   
(sin el controlador)

$$(1^\circ) \quad \theta_a = (2i+1) \cdot \pi - \theta_{OL} + \theta_p$$

$$(2^\circ) \quad a = \Re\{z_{CL}\} - \frac{\Im\{z_{CL}\}}{\tan(\theta_a)}$$

$$K \text{ debe cumplir } |G_c(z_{CL}) \cdot G_p(z_{CL})| = 1, \text{ con } G_c(z) = K \frac{z-a}{z}$$





$$G_p(z) = \frac{z^2}{(z-0.5)(z-0.7)}$$

$$G_{PD}(z) = K \frac{z-a}{z}$$

$$z_{CL} = 0.39 + j0.17$$

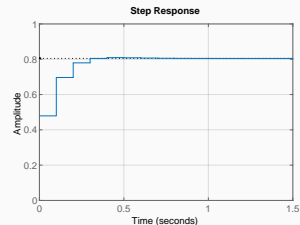
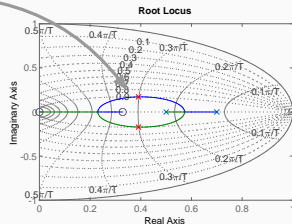
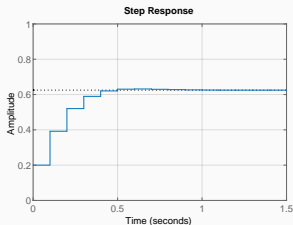
$$\theta_{OL} = \angle(G_p(z_{CL})) = 2.33 \text{ rad}$$

$$\theta_p = \angle(z_{CL}) = 0.41 \text{ rad}$$

$$\theta_a = \pi - \theta_{OL} + \theta_p = 1.21 \text{ rad}$$

$$a = \Re\{z_{CL}\} - \frac{\Im\{z_{CL}\}}{\tan(\theta_a)} = 0.32$$

$$K = \frac{1}{|G_{PD}(z_{CL}) \cdot G_p(z_{CL})|} = 0.92$$





## Controlador proporcional, integral y derivativo (PID)

---

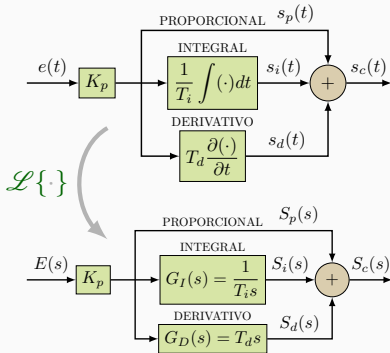
# Controladores P, PI, PD y PID

## Acciones de control:

- **Control P:** proporcional al error
  - Modifica la respuesta desplazándose por el lugar de las raíces
- **Control PI:** proporcional al error acumulado
  - Aumenta el tipo de sistema (busca  $e_s(\infty) = 0$ )
  - Se añade un integrador (polo en  $s=0$  o en  $z=1$ ) y un cero cercano a él para cancelar su efecto sobre la respuesta
- **Control PD:** proporcional a la velocidad de cambio del error
  - Desplaza los polos para mejorar el tiempo de respuesta
  - Se añade un cero (también un polo en  $z=0$  en los sist. discretos) para modificar el lugar de las raíces
- **Control PID:** Combina las estrategias anteriores para beneficiarse de las 3 acciones de control a la vez
  - Las posibilidades de modificación de la respuesta del PID son mayores que la mera aplicación en cascada de un PI y un PD

# Controlador PID en sistemas continuos

Controlador PID:



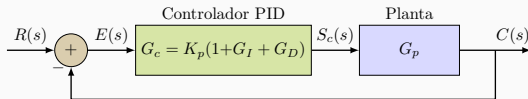
Salida del controlador PID:  $s_c(t) = s_p(t) + s_i(t) + s_d(t)$   
proporcional al

- error  $s_p(t) = K_p e(t)$ ,
- error acumulado  $s_i(t) = \frac{K_p}{T_i} \int e(t) dt$
- y a la velocidad de cambio del error  $s_d(t) = K_p T_d \frac{\partial e(t)}{\partial t}$

$$G_c(s) = \frac{S_c(s)}{E(s)} = K_p \left( 1 + \frac{1}{T_i s} + T_d s \right)$$

- $K_p$ : constante de proporcionalidad
- $T_i$ : constante de tiempo integral
- $T_d$ : constante de tiempo derivativa

Sistema realimentado con PID:



*% PID en forma paralela*

```
>> Kp=2; Ki=4; Kd=3;
```

```
>> Cp=pid(Kp,Ki,Kd)
```

```
Cp =
```

$$K_p + K_i * \frac{1}{s} + K_d * s$$

with  $K_p = 2$ ,  $K_i = 4$ ,  $K_d = 3$

Continuous-time PID controller in parallel form.

```
>>
```

*% PID en forma estandar*

```
>> Kp=4; Ti=0.5; Td=1.5;
```

```
>> Cs=pidstd(Kp,Ti,Td)
```

```
Cs =
```

$$K_p * \left(1 + \frac{1}{Ti} * \frac{1}{s} + Td * s\right)$$

with  $K_p = 4$ ,  $Ti = 0.5$ ,  $Td = 1.5$

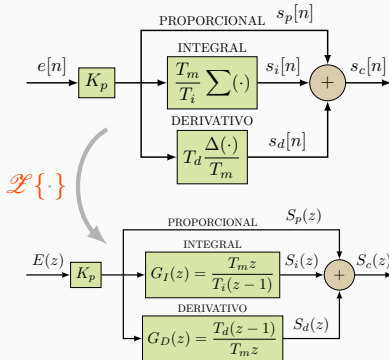
Continuous-time PID controller in standard form

$$K_i = K_p / T_i$$

$$K_d = K_p \cdot T_d$$

# Controlador PID en sistemas discretos

## Controlador PID:



Salida del controlador PID:  $s_c[n] = s_p[n] + s_i[n] + s_d[n]$   
proporcional al

- error  $s_p[n] = K_p e[n]$ ,
- error acumulado  

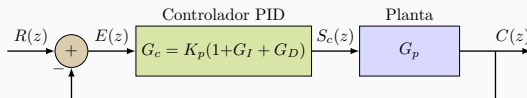
$$s_i[n] = \frac{K_p T_m}{T_i} \sum e[n] = s_i[n-1] + \frac{K_p T_m}{T_i} e[n]$$
- y a la velocidad de cambio del error  

$$s_d[n] = K_p T_d \frac{\Delta e[n]}{T_m} = K_p T_d \frac{e[n] - e[n-1]}{T_m}$$

$$G_c(z) = \frac{S_c(z)}{E(z)} = K_p \left( 1 + \frac{T_m z}{T_i(z-1)} + \frac{T_d(z-1)}{T_m z} \right)$$

- $K_p$ : constante de proporcionalidad
- $T_i$ : constante de tiempo integral
- $T_d$ : constante de tiempo derivativa
- $T_m$ : periodo de muestreo

## Sistema realimentado con PID:



## % PID en forma paralela

```
>> Kp=2; Ki=4; Kd=3; Tm=0.1;
C = pid(Kp,Ki,Kd,'Ts',Tm,'IFormula','BackwardEuler',
        'DFormula','BackwardEuler')
C =
```

$$K_p + K_i * \frac{T_s * z}{z-1} + K_d * \frac{z-1}{T_s * z}$$

with Kp = 2, Ki = 4, Kd = 3, Ts = 0.1

Sample time: 0.1 seconds

Discrete-time PID controller in parallel form.

$$K_i = K_p / T_i$$

$$K_d = K_p \cdot T_d$$

## % PID en forma estandar

```
>> Kp=2; Ti=0.5; Td=1.5; Tm=0.1;
C = pidstd(Kp,Ti,Td,'Ts',Tm,'IFormula','BackwardEuler',
        'DFormula','BackwardEuler')
C =
```

$$K_p * \left( 1 + \frac{1}{T_i} * \frac{T_s * z}{z-1} + T_d * \frac{z-1}{T_s * z} \right)$$

with Kp = 2, Ti = 0.5, Td = 1.5, Ts = 0.1

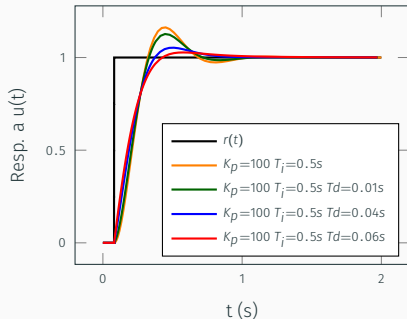
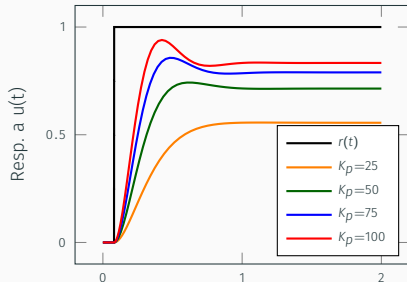
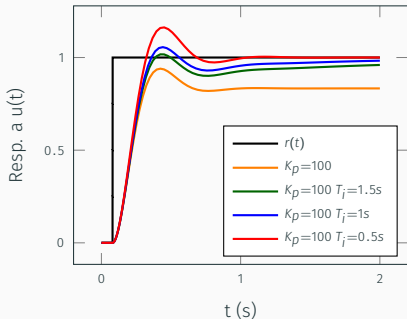
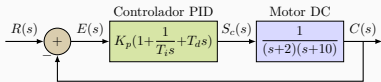
Sample time: 0.1 seconds

Discrete-time PID controller in standard form

# Diseño de controladores PID

- Diseño por prueba y error
  - Ajustar 1º  $K_p$ , 2º  $T_i$  y 3º  $T_d$
- Procedimientos de sintonización
  - Reglas de Ziegler-Nichols

## E Control de velocidad de un motor DC



# Polos y ceros del controlador PID

## Controlador PID en sist. continuos:

$$G_c(s) = K_p \left( 1 + \frac{1}{T_i s} + T_d s \right) = K \left( \frac{s^2 + 2\zeta\omega_n s + \omega_n^2}{s} \right)$$

- 1 polo en  $p = 0$  (1 integrador)  
→ incrementa tipo de sistema
- 2 ceros en  $s_{1,2} = -\zeta\omega_n \pm \omega_n\sqrt{\zeta^2 - 1}$   
→  $\Re$  o  $\mathbb{C}$  conj. dependiendo de  $\zeta \Rightarrow$  posibilidad de cancelar p/z

### • Parámetros:

- $K = K_p T_d$
- $\omega_n = \frac{1}{\sqrt{T_i T_d}}$
- $\zeta = \frac{1}{2} \sqrt{\frac{T_i}{T_d}}$

## Controlador PID en sist. discretos:

$$G_c(z) = K_p \left( 1 + \frac{T_m z}{z-1} + \frac{T_d (z-1)}{z} \right) = K \left( \frac{z^2 - 2e^{-\sigma T} \cos(\omega T) z + e^{-2\sigma T}}{z(z-1)} \right)$$

- 1 polo en  $p_1 = 1$  (1 integrador) y otro en  $p_2 = 0$  (no afecta a la resp.)  
→ incrementa tipo de sistema
- 2 ceros  
→  $\Re$  o  $\mathbb{C}$  conjugados  $\Rightarrow$  Diseño complejo



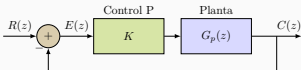
Diseño independiente del control P, PD y el PI

- 1º aplicar P, 2º mejorar velocidad de respuesta con PD y 3º mejorar el estado estacionario con PI
- 1º aplicar P, 2º mejorar el estado estacionario con PI y 3º mejorar velocidad de respuesta con PD
  - Podrían modificarse las prestaciones alcanzadas en el 2º paso al aplicar el 3º paso

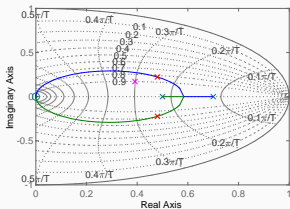
Diseño directo del control PID

- Control de la ubicación de los 2 ceros
- Se pueden buscar cancelaciones de polos reales o complejos conjugados

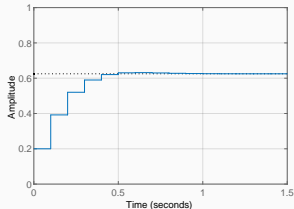
Control P



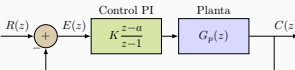
Root Locus



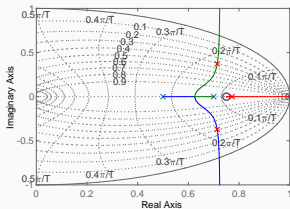
Step Response



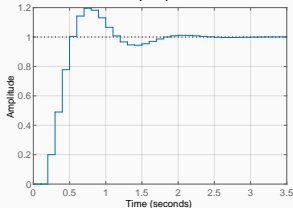
Control PI



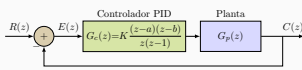
Root Locus



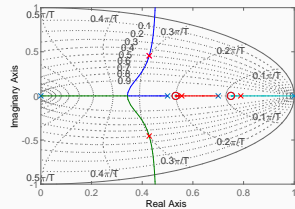
Step Response



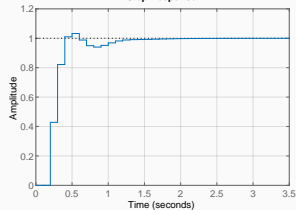
Control PID



Root Locus



Step Response



# Implementación de controladores en sistemas discretos

---

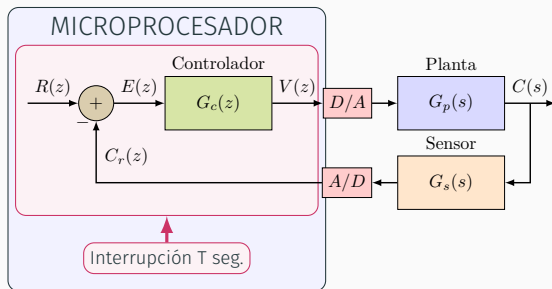
- Implementación de controladores en el microprocesador
- Efecto del “windup” del integrador y su eliminación
- Filtro derivativo
- El problema de la zona muerta en motores

# Implementación de controladores en el microprocesador

El controlador discreto se implementa con un microprocesador o microcontrolador

→ Programar en el  $\mu P$  la resta y la ecuación en diferencias del controlador

→ Se ejecuta cada  $T$  segundos (periodo de muestreo)



## E Implementación de un control PI

$$\text{Func. transf: } G_c(z) = \frac{V(z)}{E(z)} = \frac{10(z-0.9)}{z-1}$$

$$\text{Ec. dif: } v[n] = v[n-1] + 10e[n] - 9e[n-1]$$

**% pseudocode**

```
c_r = read_ADC(ch0);  
e = r - c_r;  
v = v1 + 10*e - 9*e1;  
v1 = v;  
e1 = e;  
write_DAC(ch0,v);
```

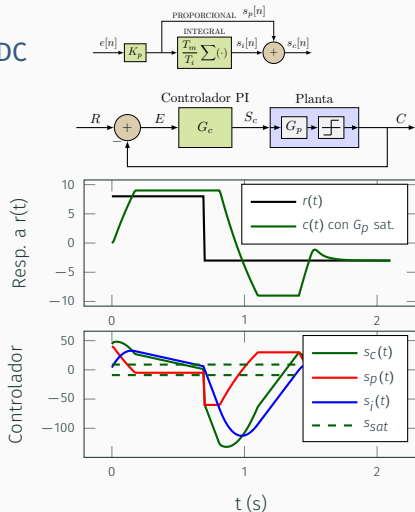
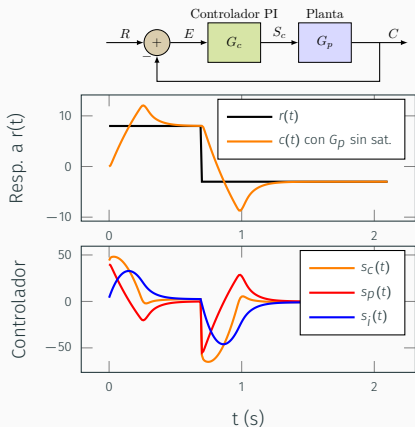
## Windup del integrador y su eliminación

---

# Efecto del “windup” del integrador

- Problema que surge si la planta/actuador presenta saturación a la salida (ej. motor)
- El integrador sigue incrementando de valor  $\rightarrow$  no cambia la respuesta si está saturada
- Resultado: la duración del sobreamortiguamiento y tiempo de establecimiento aumenta

## E Control PI de velocidad de un motor DC

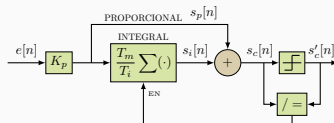


# Anti-Windup basado en integración condicional

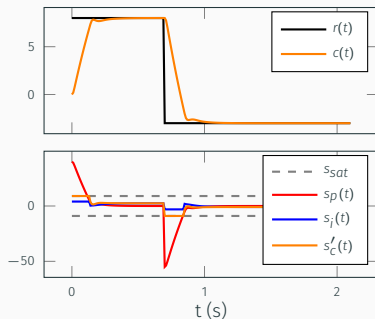
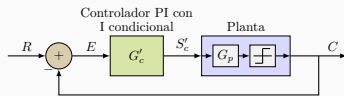
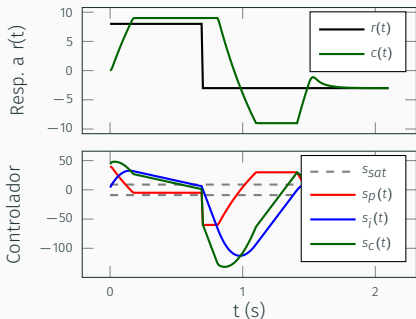
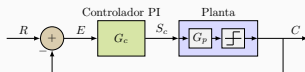
El integrador solo está activo si la salida del controlador no supera los umbrales de saturación

- $\uparrow\uparrow$  activar también el integrador cuando hay saturación y  $\text{sign}\{e[n]\} \neq \text{sign}\{s_c[n]\}$

## Controlador PI con I condicional



## E Control de velocidad de un motor DC



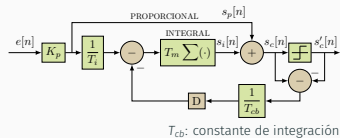


# Anti-Windup basado en “back-calculation”

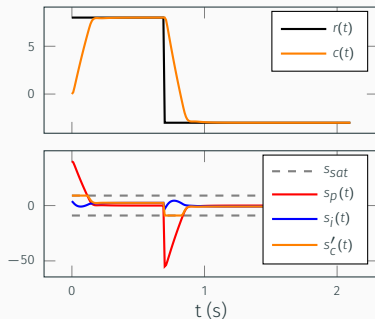
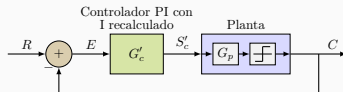
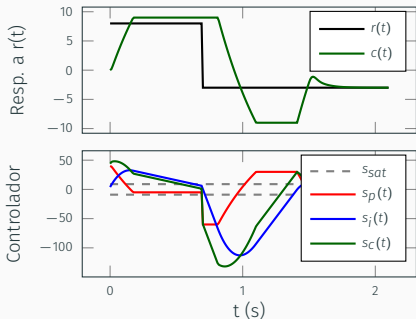
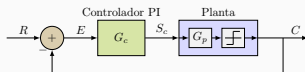
El integrador se incrementa (decrementa) cuando se alcanza el límite de saturación mínimo (máximo) de forma proporcional a la diferencia entre la salida del controlador y su valor saturado

$$s_i[n] = T_m \sum \left[ \frac{K_p}{T_i} e[n] - \frac{1}{T_{cb}} (s_c[n-1] - s'_c[n-1]) \right]$$

## Controlador PI con I recalculado



## E Control de velocidad de un motor DC



## Filtro derivativo en el PID

---

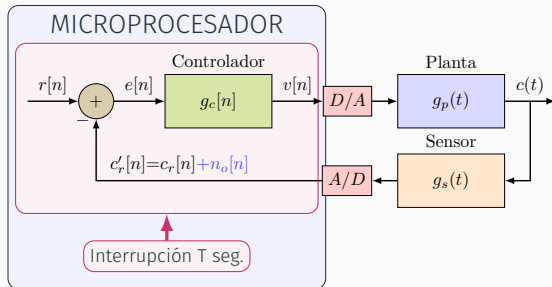
# El ruido en los sistemas de control

El ruido está presente en todos los sistemas de control

→ Lo introduce el entorno y los dispositivos electrónicos

→ El sensor mide la variable a controlar

→ También captura el ruido del sensor y del resto del sistema



**Ruido:** perturbación aleatoria, sus características dependen de la fuente que lo origina

## **E** Ruido térmico (lo introducen los dispositivos electrónicos)

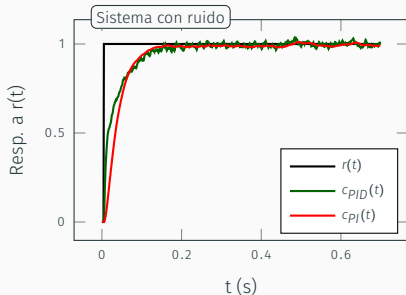
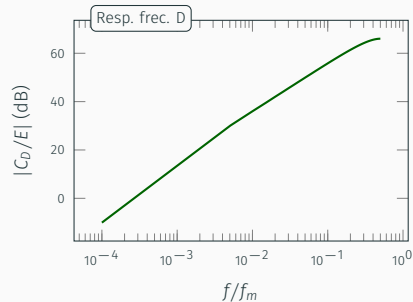
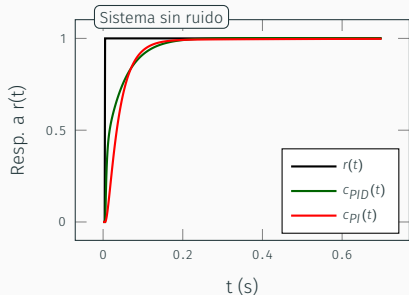
→ Ruido blanco, componentes frecuenciales en todo el ancho de banda

→ Pequeña amplitud con cambios rápidos de su magnitud

# Problema de la inclusión de la acción derivativa (D) de control

- La acción derivativa es proporcional a la velocidad de cambio del error
- Amplifica el ruido (cambios rápidos de la señal)
- $c_D[n] = \frac{e[n] - e[n-1]}{T_m} \rightarrow \frac{C_D(z)}{E(z)} = \frac{z-1}{T_m z}$

## E Control PI vs. PID con/sin ruido



# Acción de control derivativa con filtro (DF)

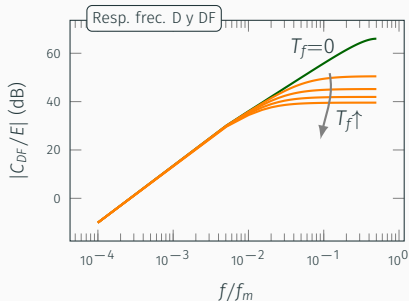
- Acción DF = filt. paso bajo + acción D

$$\frac{C_{DF}(z)}{E(z)} = \frac{1}{T_f + \frac{T_m z}{z-1}} = \frac{\frac{1}{\alpha} z}{z^{\frac{1+\alpha}{\alpha}} - 1} \cdot \frac{z-1}{T_m z}$$

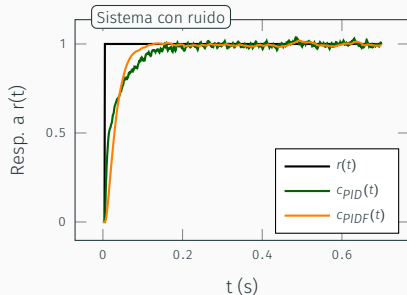
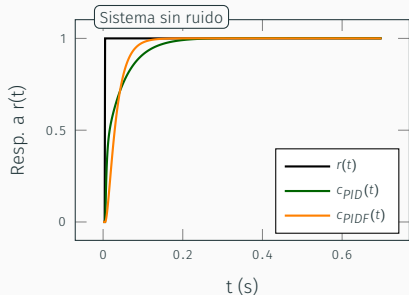
$$T_f = \alpha T_m$$

$$\text{Filtro } (z_p = \frac{\alpha}{1+\alpha})$$

Controlador D



## E Control PID vs. PID con filtro



## % PID en forma paralela

```
>> Kp=2; Ki=4; Kd=3; Tm=0.1;
C = pid(Kp,Ki,Kd,Tf,'Ts',Tm,'IFormula','BackwardEuler',
        'DFormula','BackwardEuler')
C =
```

$$K_p + K_i * \frac{T_s * z}{z-1} + K_d * \frac{1}{T_f + T_s * z / (z-1)}$$

with Kp = 2, Ki = 4, Kd = 3, Tf = 5, Ts = 0.1

Sample time: 0.1 seconds

Discrete-time PIDF controller in parallel form.

## % PID en forma estandar

```
>> Kp=2; Ti=0.5; Td=1.5; Nd=0.3; Tm=0.1;
C = pidstd(Kp,Ti,Td,Nd,'Ts',Tm,'IFormula','BackwardEuler',
        'DFormula','BackwardEuler')
C =
```

$$K_p * \left( 1 + \frac{1}{T_i} * \frac{T_s * z}{z-1} + T_d * \frac{1}{(T_d/N) + T_s * z / (z-1)} \right)$$

with Kp = 2, Ti = 0.5, Td = 1.5, N = 0.3, Ts = 0.1

Sample time: 0.1 seconds

Discrete-time PIDF controller in standard form

$$\begin{aligned} K_i &= K_p / T_i \\ K_d &= K_p \cdot T_d \\ T_f &= T_d / N_d \end{aligned}$$

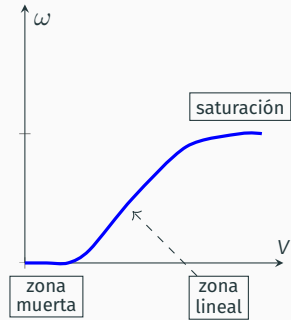
## Problema de la zona muerta en motores

---

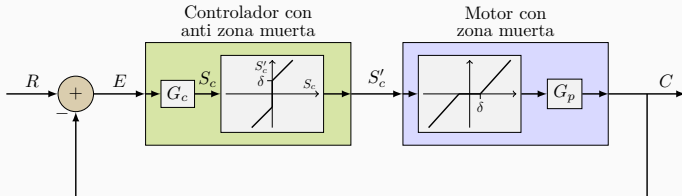
# Controlador con compensación de la zona muerta en motores

## No linealidades en motores

- **Zona muerta:** el motor no responde a pequeños voltajes debido a la fricción
- **Saturación;** la velocidad no aumenta a partir de un cierto voltaje



## Controlador con compensación de la zona muerta





# Conclusiones

---

# Conclusiones

En este tema

- se ha introducido el método del lugar de las raíces
  - representación gráfica de la evolución de los polos de la función de transferencia en lazo cerrado cuando cambia un parámetro del sistema
  - válido para sistemas continuos y discretos
- se han presentado los métodos de diseño de controladores continuos y discretos basados en las acciones de control proporcional, integral y derivativa
- se ha introducido el procedimiento de implementación de los controladores discretos en un microprocesador
- y se han introducido algunos algoritmos para resolver problemas concretos de la implementación de sistemas de control (wind-up del integrador, filtrado de ruido y compensación de la zona muerta de motores).

## Bibliografía

---

- Norman S. Nise, **Control systems engineering**, Wiley 2017
  - Capítulo 8: secciones 8.1 a 8.4 y 8.7
  - Capítulo 9: secciones 9.1 a 9.4
  - Capítulo 13: secciones 13.9 y 13.11
- M.S. Fadali A. Visioli, **Digital Control Engineering Analysis and Design**, Elsevier 2019
  - Capítulo 5: secciones 5.1 a 5.4
  - Capítulo 6: secciones 6.1, 6.2 y 6.4