



Mapeamento . . .

Diferenciação . . .

Integração Numérica

Aproximação por . . .

Controle PID

Discretização de . . .

Homepage

Página de Rosto



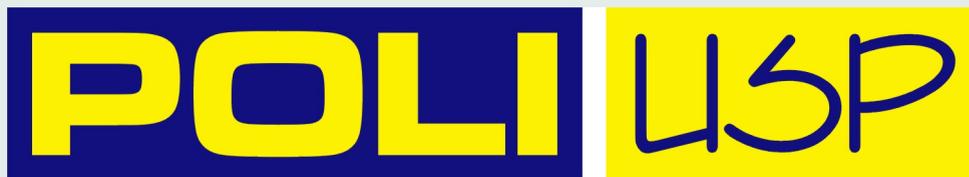
Página 1 de 42

Voltar

Full Screen

Fechar

Desistir



PTC5611 - Controle Digital de Sistemas Dinâmicos

Cap. 5: Discretização de Sistemas Contínuos

Prof. Bruno Augusto Angélico

2021



Capítulo 5 - Discretização de Sistemas Contínuos

[Mapeamento . . .](#)[Diferenciação . . .](#)[Integração Numérica](#)[Aproximação por . . .](#)[Controle PID](#)[Discretização de . . .](#)[Homepage](#)[Página de Rosto](#)[Página 2 de 42](#)[Voltar](#)[Full Screen](#)[Fechar](#)[Desistir](#)

O objetivo deste capítulo é definir uma função de transferência discreta, $C_D(z)$, que seja aproximadamente igual a uma função de transferência de sistemas contínuos $C(s)$, na faixa de frequências de interesse. Pode-se com isto, por exemplo, projetar um controlador no domínio s e converter para o domínio z para ser implementado digitalmente em hardware. Alguns métodos serão discutidos, tendo como base a referência [Franklin].



1. Mapeamento Casado de Polos e Zeros

Pode-se, inicialmente, considerar $z = e^{sT_s}$ e sua inversa $s = (1/T_s) \ln(z)$ como a transformação exata entre z e s . Tal transformação é conhecida como *zero-pole matching*, traduzido aqui para mapeamento casado de polos e zeros.

- Todos os polos e zeros no plano- s são mapeados no plano- z como $z = e^{sT_s}$.
- Zeros de $C(s)$ no infinito são mapeados em $z = -1$. O ponto $z = -1$ representa a maior frequência possível na função de transferência discreta.
 - ★ Para obter um sistema bi-próprio, todos os zeros em $s = \infty$ são mapeados em $z = -1$ (termo $(z + 1)$);
 - ★ Para obter sistema estritamente próprio, um dos zeros no infinito no plano- s também é mapeado em um zero no

Mapeamento...

Diferenciação...

Integração Numérica

Aproximação por...

Controle PID

Discretização de...

Homepage

Página de Rosto



Página 3 de 42

Voltar

Full Screen

Fechar

Desistir



infinito no plano- z , enquanto que os outros são mapeados em $z = -1$. Com isto, há um zero finito a menos do que polos finitos em $C_D(z)$.

- O ganho de um sistema discreto deve ser ajustado tal que $C(s)|_{s=0} = C_D(z)|_{z=1}$.

O comando em MATLAB para transformar uma função de transferência contínua $C(s)$ em uma discreta $C_D(s)$ com período de amostragem T_s , através de mapeamento casado de polos e zeros, é: `C_D = c2d(C,T_s,'matched')`.

🔗 **Exemplo 1: Zero-Pole Matching:** Encontre o equivalente discreto de

$$C(s) = \frac{a}{s + a}$$

utilizando mapeamento casado de polos e zeros.

Solução: note que $C(s)$ possui um polo finito em $s = -a$, que é

Mapeamento...

Diferenciação...

Integração Numérica

Aproximação por...

Controle PID

Discretização de...

Homepage

Página de Rosto



Página 4 de 42

Voltar

Full Screen

Fechar

Desistir



mapeado em $z = e^{-aT_s}$, e um zero infinito, que, primeiramente, é mapeado em um zero em $z = -1$. Logo,

$$C_D(z) = K \frac{z + 1}{z - e^{-aT_s}}$$

Como,

$$C(s)|_{s=0} = 1 = C_D(z)|_{z=1} = K \frac{2}{1 - e^{-aT_s}},$$

tem-se que $K = \frac{1 - e^{-aT_s}}{2}$. Portanto,

$$C_D(z) = \frac{(z + 1)(1 - e^{-aT_s})}{2(z - e^{-aT_s})}$$

Mapeando, por outro lado, $s = \infty$ em $z = \infty$, pois assim há um polo finito e nenhum (um a menos) zero finito, tem-se

$$C_D(z) = K \frac{1}{z - e^{-aT_s}}$$

[Mapeamento...](#)[Diferenciação...](#)[Integração Numérica](#)[Aproximação por...](#)[Controle PID](#)[Discretização de...](#)[Homepage](#)[Página de Rosto](#)[Página 5 de 42](#)[Voltar](#)[Full Screen](#)[Fechar](#)[Desistir](#)



Como,

$$C(s)|_{s=0} = 1 = C_D(z)|_{z=1} = K \frac{1}{1 - e^{-aT_s}},$$

tem-se que $K = 1 - e^{-aT_s}$. Portanto,

$$C_D(z) = \frac{1 - e^{-aT_s}}{z - e^{-aT_s}}$$

Apesar de o mapeamento casado de polos e zeros mapear perfeitamente um ponto no plano- s em um ponto no plano- z , e vice-versa, as funções $e^{(\cdot)}$ e $\ln(\cdot)$ são transcendentais. A seguir serão apresentadas algumas transformações aproximadas, porém mais simples, do plano- s para o plano- z .

Mapeamento...

Diferenciação...

Integração Numérica

Aproximação por...

Controle PID

Discretização de...

Homepage

Página de Rosto



Página 6 de 42

Voltar

Full Screen

Fechar

Desistir



2. Diferenciação Numérica

O método da diferença em avanço (*forward rectangular*) é uma técnica simples que substitui a derivada de uma função por

$$\left. \frac{de(t)}{dt} \right|_{t=nT_s} = \frac{e(t + T_s) - e(t)}{T_s} \quad (1)$$

A derivada no plano- s corresponde à multiplicação por s . Por outro lado, um deslocamento em avanço de uma amostra corresponde na multiplicação por z no plano- z . Assim,

$$s = \frac{z - 1}{T_s} \quad (2)$$

Para transformar uma função do plano- s para o plano- z pelo método da diferença em avanço deve-se fazer

$$C_D(z) = C(s) \Big|_{s=\frac{z-1}{T_s}} \quad (3)$$

Mapeamento...

Diferenciação...

Integração Numérica

Aproximação por...

Controle PID

Discretização de...

Homepage

Página de Rosto



Página 7 de 42

Voltar

Full Screen

Fechar

Desistir



Utilizando por outro lado a diferença em atraso (*backward rectangular*)

$$\left. \frac{de(t)}{dt} \right|_{t=nT_s} = \frac{e(t) - e(t - T_s)}{T_s} \quad (4)$$

Um deslocamento em atraso de uma amostra corresponde na multiplicação por z^{-1} no plano- z . Com isso,

$$s = \frac{1 - z^{-1}}{T_s} = \frac{z - 1}{T_s z} \quad (5)$$

Portanto, para transformar uma função do plano- s para o plano- z pelo método da diferença em atraso basta fazer

$$C_D(z) = C(s) \Big|_{s=\frac{z-1}{T_s z}} \quad (6)$$

A Figura 1 apresenta graficamente as duas aproximações para derivada consideradas.



Mapeamento...

Diferenciação...

Integração Numérica

Aproximação por...

Controle PID

Discretização de...

Homepage

Página de Rosto



Página 9 de 42

Voltar

Full Screen

Fechar

Desistir

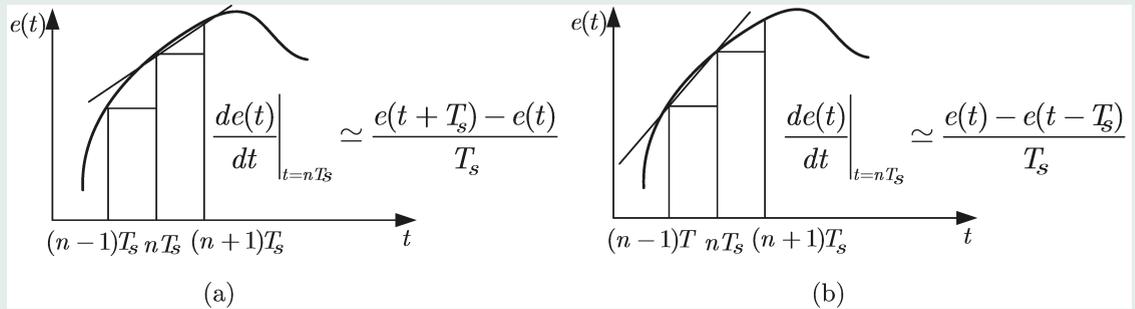


Figure 1: Aproximações para derivada. (a) *forward rectangular*; (b) *backward rectangular*.



3. Integração Numérica

Considere um sistema dado por

$$C(s) = \frac{U(s)}{E(s)} = \frac{a}{s + a}. \quad (7)$$

A equação diferencial equivalente é dada por

$$\dot{u}(t) + au(t) = ae(t). \quad (8)$$

Logo,

$$u(t) = \int_0^t [-au(\tau) + ae(\tau)] d\tau \quad (9)$$

Para $t = nT_s$, tem-se

$$\begin{aligned} u(nT_s) &= \int_0^{nT_s} [-au(\tau) + ae(\tau)] d\tau \\ &= \int_0^{nT_s - T_s} [-au(\tau) + ae(\tau)] d\tau + \int_{nT_s - T_s}^{nT_s} [-au(\tau) + ae(\tau)] d\tau \\ &= u(nT_s - T_s) + \begin{cases} \text{área de } -au(\tau) + ae(\tau) \\ \text{para } nT_s - T_s \leq \tau < nT_s \end{cases} \end{aligned} \quad (10)$$

[Homepage](#)[Página de Rosto](#)[Página 10 de 42](#)[Voltar](#)[Full Screen](#)[Fechar](#)[Desistir](#)



A área de $-au(\tau) + ae(\tau)$ no intervalo $nT_s - T_s \leq \tau < nT_s$ pode ser aproximada conforme ilustrado na Figura 2.

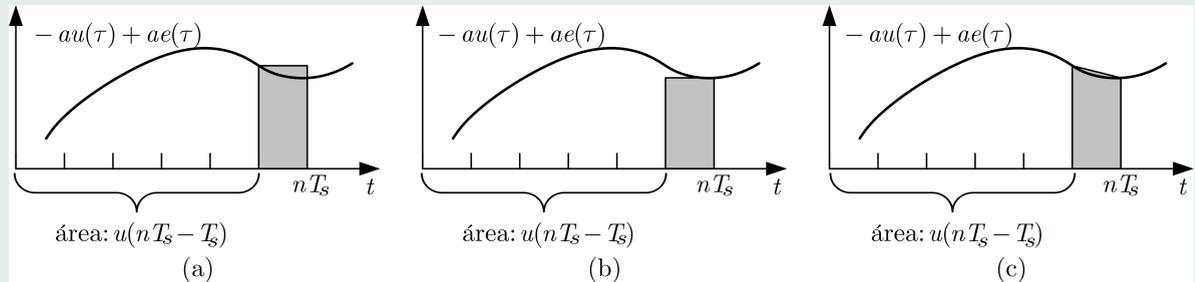


Figure 2: Aproximações para integral. (a) *forward rectangular*; (b) *backward rectangular*; (c) *trapezoidal*.

Utilizando a aproximação *forward rectangular* (Figura 2 (a)), pode-se verificar que

$$\begin{aligned} u(nT_s) &= u(nT_s - T_s) + T_s [-au(nT_s - T_s) + ae(nT_s - T_s)] \\ &= u([n - 1]T_s) + T_s [-au([n - 1]T_s) + ae([n - 1]T_s)] \end{aligned} \quad (11)$$



A sequência obtida é dada por

$$u[n] = u[n - 1] + T_s (-au[n - 1] + ae[n - 1]). \quad (12)$$

Aplicando a transformada- z em (12), obtém-se

$$U(z) = z^{-1}U(z) - aT_s z^{-1}U(z) + aT_s z^{-1}E(z), \quad (13)$$

ou seja,

$$\frac{U(z)}{E(z)} = \frac{aT_s z^{-1}}{1 - (1 - aT_s)z^{-1}} = \frac{aT_s}{z - 1 + aT_s} = \frac{a}{(z - 1)/T_s + a} \quad (14)$$

Utilizando agora a aproximação *backward rectangular* (Figura 2 (b)), pode-se verificar que

$$\begin{aligned} u(nT_s) &= u(nT_s - T_s) + T_s [-au(nT_s) + ae(nT_s)] \\ &= u([n - 1]T_s) + T_s [-au(nT_s) + ae(nT_s)] \end{aligned} \quad (15)$$

A sequência obtida é dada por

$$u[n] = u[n - 1] + T_s (-au[n] + ae[n]) \quad (16)$$



Aplicando a transformada- z em (16), obtém-se

$$U(z) = z^{-1}U(z) - aT_s U(z) + aT_s E(z) \quad (17)$$

ou seja,

$$\frac{U(z)}{E(z)} = \frac{aT_s}{1 + aT_s - z^{-1}} = \frac{aT_s z}{z + aT_s z - 1} = \frac{a}{(z - 1) / (T_s z) + a} \quad (18)$$

Uma outra regra que pode ser utilizada é a trapezoidal, como mostrado na Figura 2 (c)). Com isso,

$$\begin{aligned} u(nT_s) &= u(nT_s - T_s) + T_s [-au(nT_s) + ae(nT_s)] + \\ &\quad \frac{T_s}{2} \{[-au(nT_s - T_s) + ae(nT_s - T_s)] - [-au(nT_s) + ae(nT_s)]\} \\ &= u(nT_s - T_s) + \frac{T_s}{2} [-au(nT_s - T_s) + ae(nT_s - T_s) - au(nT_s) + ae(nT_s)] \end{aligned} \quad (19)$$

Mapeamento...

Diferenciação...

Integração Numérica

Aproximação por...

Controle PID

Discretização de...

Homepage

Página de Rosto



Página 13 de 42

Voltar

Full Screen

Fechar

Desistir



A sequência obtida é dada por

$$u[n] = u[n - 1] + \frac{T_s}{2} (-au[n - 1] + ae[n - 1] - au[n] + ae[n]) \quad (20)$$

Aplicando a transformada- z e separando os termos em $U(z)$ e $E(z)$, obtém-se

$$U(z) \left[1 + a\frac{T_s}{2} - z^{-1} + a\frac{T_s}{2}z^{-1} \right] = E(z) \left[a\frac{T_s}{2}z^{-1} + a\frac{T_s}{2} \right], \quad (21)$$

ou seja,

$$\begin{aligned} \frac{U(z)}{E(z)} &= \frac{a\frac{T_s}{2}z^{-1} + a\frac{T_s}{2}}{1 + a\frac{T_s}{2} - z^{-1} + a\frac{T_s}{2}z^{-1}} = \frac{aT_s(z + 1)}{2(z - 1) + aT_s(z + 1)} \\ &= \frac{a}{(2/T_s) [(z - 1)/(z + 1)] + a}. \end{aligned} \quad (22)$$

A tabela seguinte resume as três abordagens vistas.



$C(s)$	Método	$C_D(z)$
$\frac{a}{s + a}$	Forward	$\frac{a}{(z - 1)/T_s + a}$
$\frac{a}{s + a}$	Backward	$\frac{a}{(z - 1)/(T_s z) + a}$
$\frac{a}{s + a}$	Trapezoidal	$\frac{a}{(2/T_s) [(z - 1)/(z + 1)] + a}$

Note que $C_D(z)$ é obtido de $C(s)$ fazendo as substituições apresentadas na tabela seguinte:

[Homepage](#)[Página de Rosto](#)[Página 15 de 42](#)[Voltar](#)[Full Screen](#)[Fechar](#)[Desistir](#)



Método	Aproximação
Forward	$s \leftarrow \frac{z - 1}{T_s}$
Backward	$s \leftarrow \frac{z - 1}{T_s z}$
Trapezoidal	$s \leftarrow \frac{2}{T_s} \frac{z - 1}{z + 1}$

As transformações dos métodos *Forward* e *Backward* de integração são equivalentes aos respectivos métodos de derivação.

O método trapezoidal é também conhecido como método de Tustin, ou transformação bilinear. Uma outra forma de se obter a transformação bilinear é observando que $z = e^{sT_s} \Rightarrow s =$

[Homepage](#)[Página de Rosto](#)[Página 16 de 42](#)[Voltar](#)[Full Screen](#)[Fechar](#)[Desistir](#)



$(1/T_s) \ln(z)$. A função $\ln(z)$ pode ser expandida como

$$\ln(z) = 2 \left(\frac{z-1}{z+1} + \frac{1}{3} \left(\frac{z-1}{z+1} \right)^3 + \frac{1}{5} \left(\frac{z-1}{z+1} \right)^5 + \dots \right). \quad (23)$$

Logo,

$$\begin{aligned} s = \frac{1}{T_s} \ln(z) &= \frac{2}{T_s} \left(\frac{z-1}{z+1} + \frac{1}{3} \left(\frac{z-1}{z+1} \right)^3 + \frac{1}{5} \left(\frac{z-1}{z+1} \right)^5 + \dots \right) \\ &\approx \frac{2}{T_s} \frac{z-1}{z+1}, \end{aligned} \quad (24)$$

ou seja,

$$C_D(z) \approx C(s) \Big|_{s=\frac{2}{T_s} \frac{z-1}{z+1}} \quad (25)$$

Para a regra *forward rectangular*, $z = 1 + T_s s$. Substituindo $s = j\omega$, tem-se que $z = 1 + jT_s \omega$. Vide Figura 3 (a). Um sistema estável em s pode então ser mapeado em um sistema instável em z .

[Homepage](#)[Página de Rosto](#)[Página 17 de 42](#)[Voltar](#)[Full Screen](#)[Fechar](#)[Desistir](#)



Para um polo real estável no plano s ser mapeado em um polo estável no plano z pela aproximação retangular para frente, deve-se garantir que $|1 + T_s s| < 1$. Suponha, por exemplo, que s seja real. Assim,

$$-1 < 1 + T_s s < 1 \Rightarrow -2 < T_s s < 0$$

Com isso, conclui-se que: (i) $s < 0$, o que é trivial; (ii) supondo $s = -s_0$, $s_0 > 0$, tem-se que $T_s s_0 < 2 \Rightarrow T_s < \frac{2}{s_0}$.

Para a regra *backward rectangular*, $z = \frac{1}{1 - T_s s}$. Adicionando e subtraindo $1/2$, obtém-se

$$z = \frac{1}{2} + \left(\frac{1}{1 - T_s s} - \frac{1}{2} \right) = \frac{1}{2} + \frac{1 + T_s s}{2(1 - T_s s)} \quad (26)$$

Substituindo $s = j\omega$,

$$z = \frac{1}{2} + \frac{1 + j\omega T_s}{2(1 - j\omega T_s)} \quad (27)$$

Mapeamento...

Diferenciação...

Integração Numérica

Aproximação por...

Controle PID

Discretização de...

Homepage

Página de Rosto



Página 18 de 42

Voltar

Full Screen

Fechar

Desistir



Como $\left| \frac{11 + j\omega T_s}{21 - j\omega T_s} \right| = \frac{1}{2}$, pode-se observar que o mapeamento de $j\omega$ no plano- z será um circunferência, como ilustrado na Figura 3 (b). O semiplano da esquerda do plano- s é mapeado para dentro da circunferência da Figura 2 b.

Por fim, para a transformação bilinear, com $s = j\omega$, tem-se

$$z = \frac{2 + j\omega T_s}{2 - j\omega T_s} \quad (28)$$

Como $|z| = 1$, a reta $j\omega$ do plano- s é mapeado na circunferência unitária no plano- z . O semiplano da esquerda do plano- s é mapeado no círculo unitário no plano- z , como ilustrado na Figura 3 (c).

O comando em MATLAB para transformar uma função de transferência contínua $C(s)$ em uma discreta $C_D(z)$ com período de amostragem T_s , através da transformada bilinear, é:
`C_D = c2d(C,T_s,'tustin')`.

Mapeamento...

Diferenciação...

Integração Numérica

Aproximação por...

Controle PID

Discretização de...

Homepage

Página de Rosto



Página 19 de 42

Voltar

Full Screen

Fechar

Desistir

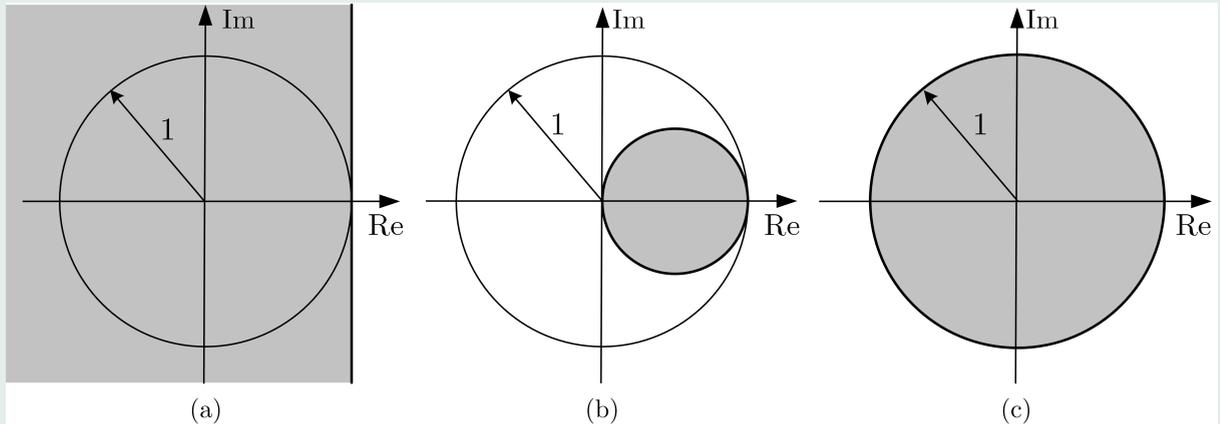


Figure 3: Mapeamento do semiplano da esquerda plano- s no plano- z para (a) *forward rectangular*; (b) *backward rectangular*; (c) *bilinear*.

✎ **Exemplo 2: Transformação Bilinear:** Projete um filtro digital utilizando transformação bilinear no seguinte filtro analógico

$$C(s) = \frac{1}{0,1s + 1}$$

utilizando $T_s = 0,1$ s.

Solução:



$$C_D(z) = \frac{1}{0,1 \frac{2}{0,1} \frac{z-1}{z+1} + 1} = \frac{z+1}{3z-1}$$

Utilizando o MATLAB, o exemplo pode ser resolvido como

```
clear all; clc;
num_C = 1; den_C = [0.1 1];
T_s = 0.1;
C = tf(num_C, den_C);
disp('C_D(z) = ');
C_D = c2d(C, T_s, 'tustin')
```

Apesar de toda a região estável do plano- s (semiplano da esquerda) ser mapeada dentro de toda a região estável do plano- z (círculo unitário), há distorções (*warping*). Seja ω_A um ponto no eixo imaginário do plano- s . Como visto, pontos sobre o eixo imaginário do plano- s são mapeados em pontos sobre a circunferência unitária no plano- z e vice-versa. Utilizando a transformação

[Homepage](#)[Página de Rosto](#)[Página 21 de 42](#)[Voltar](#)[Full Screen](#)[Fechar](#)[Desistir](#)



bilinear de $s = 0 + j\omega_A$ para z , tem-se

$$j\omega_A = \frac{2}{T_s} \frac{z - 1}{z + 1} \quad (29)$$

Considere ω_D , tal que,

$$z = e^{sT_s} = e^{j\omega_D T_s} \quad (30)$$

Ao substituir (30) em (29) pode-se medir a distorção de frequência que ocorre quanto o método de Tustin é utilizado para discretizar uma função de transferência, pois

$$j\omega_A = \frac{2}{T_s} \frac{e^{j\omega_D T_s} - 1}{e^{j\omega_D T_s} + 1} = \frac{2}{T_s} \frac{e^{j\omega_D T_s/2} (e^{j\omega_D T_s/2} - e^{-j\omega_D T_s/2})}{e^{j\omega_D T_s/2} (e^{j\omega_D T_s/2} + e^{-j\omega_D T_s/2})} = \frac{2}{T_s} \frac{2j \sin(\omega_D T_s/2)}{2 \cos(\omega_D T_s/2)}, \quad (31)$$

ou seja,

$$\omega_A = \frac{2}{T_s} \tan(\omega_D T_s/2) \quad (32)$$

Se $\tan(\omega_D T_s/2) \approx \omega_D T_s/2$, ou seja, quando T_s é pequeno (altas frequências de amostragem), $\omega_A \approx \omega_D$ e, portanto, a distorção é

[Homepage](#)[Página de Rosto](#)[Página 22 de 42](#)[Voltar](#)[Full Screen](#)[Fechar](#)[Desistir](#)



minimizada. Uma forma de compensar as distorções em uma dada frequência ω_c , geralmente a frequência de canto do filtro, é utilizar o procedimento de pré-distorção (*pre-warping*) na frequência ω_c . A ideia é garantir que em ω_c , a função de transferência discreta tenha as mesmas características da função em tempo contínuo correspondente. Note que isso pode ser feito se

$$s = \frac{\omega_c}{\tan(\omega_c T_s / 2)} \frac{z - 1}{z + 1} \quad (33)$$

Portanto, com *pre-warping*, tem-se a seguinte equivalência

$$C_D(z) = C(s) \Big|_{s = \frac{\omega_c}{\tan(\omega_c T_s / 2)} \frac{z-1}{z+1}} \quad (34)$$

O comando em MATLAB para transformar uma função de transferência contínua $C(s)$ em uma discreta $C_D(s)$ com período de amostragem T_s , através da transformada bilinear e *pre-warping* em ω_c é: `C_D = c2d(C, T_s, 'prewarp', wc)`.

Mapeamento...

Diferenciação...

Integração Numérica

Aproximação por...

Controle PID

Discretização de...

Homepage

Página de Rosto



Página 23 de 42

Voltar

Full Screen

Fechar

Desistir



4. Aproximação por Segurador de Ordem Zero

Esta aproximação é utilizada quando se deseja obter uma representação discreta de um sistema incluindo o efeito do segurador de ordem zero, conforme apresentado na Figura 4.

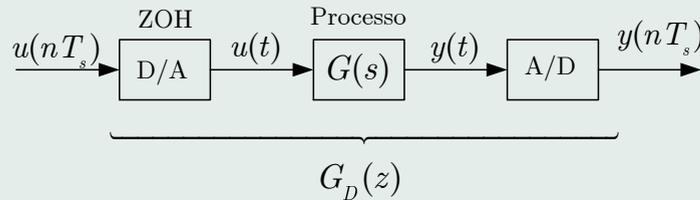


Figure 4: Equivalência do subsistema ZOH + processo + A/D.

O equivalente discreto de $G(s)$ com ZOH pode ser obtido como:

$$G_D(z) = \mathcal{Z} \left\{ \frac{1 - e^{-T_s s}}{s} G(s) \right\} = \mathcal{Z} \left\{ \frac{G(s)}{s} \right\} - \mathcal{Z} \left\{ e^{-T_s s} \frac{G(s)}{s} \right\} \quad (35)$$



Como $e^{-T_s s}$ é o atraso de um período de amostragem, verifica-se que

$$\mathcal{Z} \left\{ e^{-T_s s} \frac{G(s)}{s} \right\} = z^{-1} \mathcal{Z} \left\{ \frac{G(s)}{s} \right\}. \quad (36)$$

Portanto,

$$G_D(z) = (1 - z^{-1}) \mathcal{Z} \left\{ \frac{G(s)}{s} \right\} \quad (37)$$

MATLAB: `G_D = c2d(G, T_s, 'zoh')`.

Considere que a resposta ao degrau do sistema em tempo discreto seja coincidente com a do sistema em tempo contínuo nos instantes de amostragem, ou seja,

$$\mathcal{Z}^{-1} \left[G_D(z) \frac{1}{1 - z^{-1}} \right] = \mathcal{L}^{-1} \left[G(s) \frac{1}{s} \right] \Big|_{t=nT_s}. \quad (38)$$



Assim,

$$G_D(z) \frac{1}{1 - z^{-1}} = \mathcal{Z} \left\{ \mathcal{L}^{-1} \left[G(s) \frac{1}{s} \right] \right\} = \mathcal{Z} \left[\frac{G(s)}{s} \right], \quad (39)$$

ou ainda,

$$G_D(z) = (1 - z^{-1}) \mathcal{Z} \left[\frac{G(s)}{s} \right]. \quad (40)$$

Por essa razão, a aproximação por ZOH também é conhecida com **invariância ao degrau**.

Homepage

Página de Rosto



Página 26 de 42

Voltar

Full Screen

Fechar

Desistir



5. Controle PID

É de longe o tipo de controle mais utilizado na indústria. O controle PID na forma paralela é apresentado na Figura 5.

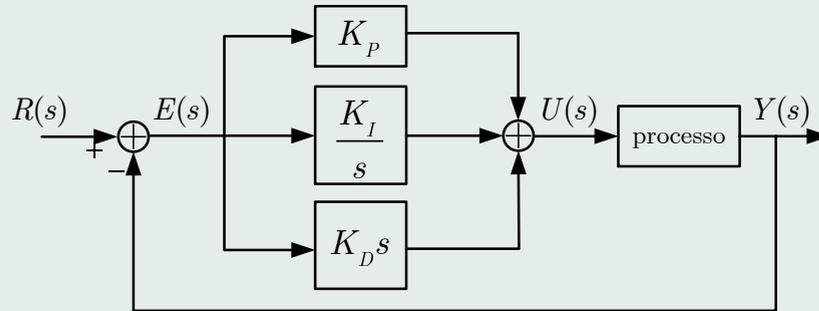


Figure 5: Controlador PID na forma paralela.

Homepage

Página de Rosto



Página 27 de 42

Voltar

Full Screen

Fechar

Desistir



Ação de controle é dada por:

$$u(t) = K_P e(t) + K_I \int_0^t e(\tau) d\tau + K_D \frac{de(t)}{dt}$$

$$\Rightarrow \frac{U(s)}{E(s)} = K_P + \frac{K_I}{s} + K_D s \quad (41)$$

Outra estrutura de controle PID é apresentada na Figura 6.

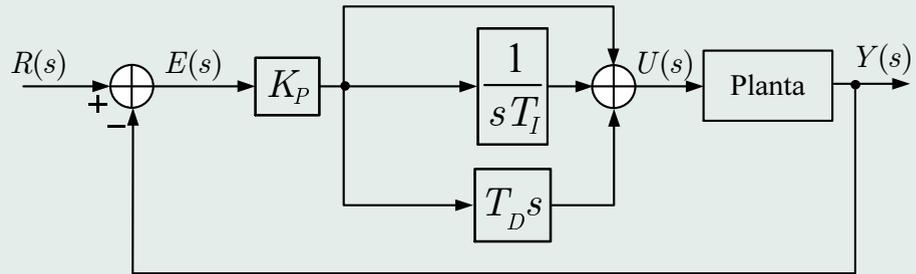


Figure 6: outra forma de controlador PID.

Tal esquema é um caso particular da forma ISA, sendo mais usual. Note que:



$$u(t) = K_P \left(e(t) + \frac{1}{T_I} \int_0^t e(\tau) d\tau + T_D \frac{de(t)}{dt} \right)$$
$$\Rightarrow \frac{U(s)}{E(s)} = K_P \left(1 + \frac{1}{T_I s} + T_D s \right) \quad (42)$$

Problema da ação integrativa \rightarrow *windup*.

O problema maior é a volta do sistema a uma condição normal após algum tempo em saturação.

Solução: *anti-windup* \rightarrow limitar ou “congelar” a ação integrativa quando o sistema entre em saturação, evitando o acúmulo de erro indesejável no integrador.

Uma forma muito utilizada de PID com *anti-windup* é representada na Figura 7:

Mapeamento...

Diferenciação...

Integração Numérica

Aproximação por...

Controle PID

Discretização de...

Homepage

Página de Rosto



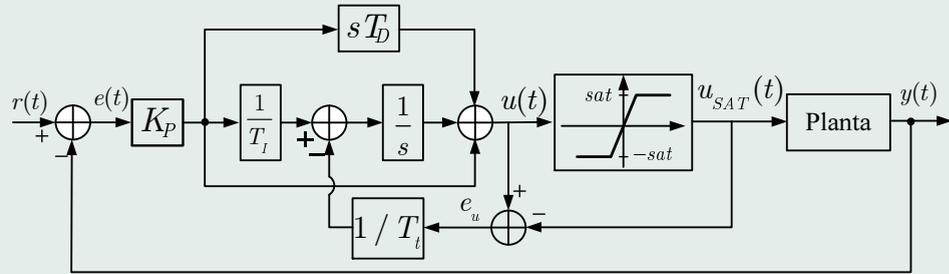
Página 29 de 42

Voltar

Full Screen

Fechar

Desistir

Figure 7: Controlador PID com *anti-windup*.

T_t é a constante de tempo de atracamento do *anti-windup*. Uma regra prática consiste em assumir $T_D \leq T_t \leq T_I = \sqrt{T_I T_D}$ (Astron).

Para utilizar o anti-windup em um sistema prático, deve-se inserir na malha um modelo de saturação do atuador.

Sobre a ação derivativa... Considere isoladamente agora a



ação derivativa, tal que

$$U(s) = K_P T_D s E(s) \quad (43)$$

- F.T não própria.
- Ganho aumenta com a frequência, e o sistema fica extremamente sensível a ruídos de alta frequência.

Solução:

$$U(s) = K_P \frac{T_D s}{(T_D/N)s + 1} E(s) \quad (44)$$

com $3 \leq N \leq 20$. Essa forma acrescenta um polo em $-N/T_D$, e o ganho da parcela derivativa é saturado a partir de $\omega = N/T_D$.

Mapeamento...

Diferenciação...

Integração Numérica

Aproximação por...

Controle PID

Discretização de...

Homepage

Página de Rosto



Página 31 de 42

Voltar

Full Screen

Fechar

Desistir

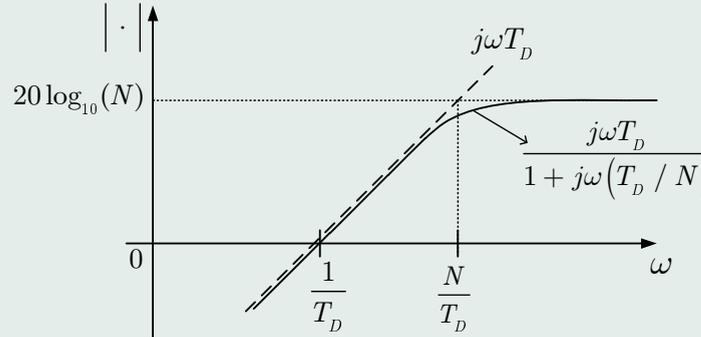


Figure 8: Comparação da resposta em frequência do termo derivativo.

Variações abruptas na referência → *kick derivativo*.

Solução: a parcela derivativa pode atuar diretamente na realimentação de saída, com sinal trocado:

$$U(s) = -K_P \frac{T_D s}{(T_D/N)s + 1} Y(s), \quad (45)$$

[Homepage](#)[Página de Rosto](#)

◀

▶

◀

▶

[Página 32 de 42](#)[Voltar](#)[Full Screen](#)[Fechar](#)[Desistir](#)

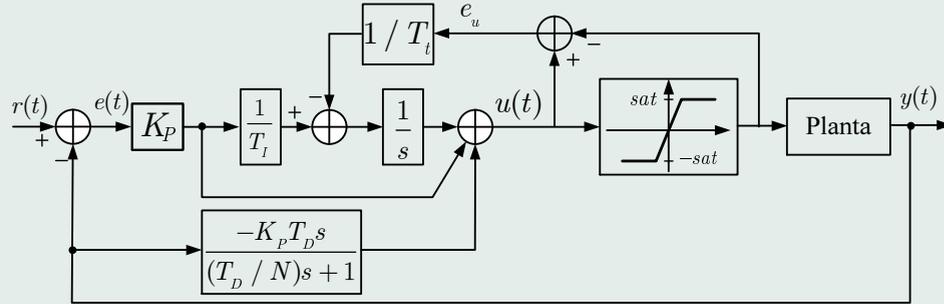


Figure 9: Controlador PID realista com *anti-windup*.

Os efeitos independentes dos ganhos K_P , K_I e K_D na resposta de malha fechada do sistema são resumidos na Tabela 1:

Table 1: Efeitos dos parâmetros K_P , K_I e K_D em um controle PID.

	t_r	$U.P.$	t_s	e_{ss}	Estabilidade
$\uparrow K_P$	Decresce	Aumenta	Aumenta pouco	Decresce	Degrada
$\uparrow K_I$	Decr. Pouco	Aumenta	Aumenta	Decr. Muito	Degrada
$\uparrow K_D$	Decr. Pouco	Decresce	Decresce	Influi Pouco	Melhora*

* Desde que o sistema seja livre de ruídos de altas frequências.

- Mapeamento ...
- Diferenciação ...
- Integração Numérica
- Aproximação por ...
- Controle PID
- Discretização de ...

- Homepage
- Página de Rosto

- ◀
- ▶
- ◀
- ▶

Página 33 de 42

- Voltar
- Full Screen
- Fechar
- Desistir



6. Discretização de Controladores PID

O PID básico é dado por

$$u(t) = K_P \left(e(t) + \frac{1}{T_I} \int_0^t e(\tau) d\tau + T_D \frac{de(t)}{dt} \right) \quad (46)$$

Pode ser dividido como:

$$u_P(t) = K_P e(t), \quad (47)$$

$$u_I(t) = \frac{K_P}{T_I} \int_0^t e(\tau) d\tau, \quad (48)$$

e,

$$u_D(t) = K_P T_D \frac{de(t)}{dt} \quad (49)$$

Discretização da parte proporcional:

$$u_P[n] = K_P e[n] \quad (50)$$

Mapeamento...

Diferenciação...

Integração Numérica

Aproximação por...

Controle PID

Discretização de...

Homepage

Página de Rosto



Página 34 de 42

Voltar

Full Screen

Fechar

Desistir



Discretização da parte integrativa:

- Retangular para trás:

$$u_I[n] = \frac{K_P T_s}{T_I} \sum_{k=0}^n e[k] \Rightarrow u_I[n] = u_I[n-1] + \frac{K_P T_s}{T_I} e[n], \quad (51)$$

- Tustin:

$$u_I[n] = u_I[n-1] + \frac{K_P T_s}{2T_I} (e[n] + e[n-1]), \quad (52)$$

Discretização da parte derivativa:

- Retangular para trás:

$$u_D[n] = \frac{K_P T_D}{T_s} (e[n] - e[n-1]) \quad (53)$$

[Homepage](#)[Página de Rosto](#)[Página 35 de 42](#)[Voltar](#)[Full Screen](#)[Fechar](#)[Desistir](#)



- Retangular para trás com filtro:

$$\frac{U(s)}{E(s)} = K_P \left(1 + \frac{1}{T_I s} + \frac{T_D s}{(T_D/N)s + 1} \right), \quad (54)$$

$$\frac{U_D(z)}{E(z)} = \frac{K_P N T_D}{(T_D + N T_s)} \frac{z - 1}{z - \frac{T_D}{(T_D + N T_s)}} \quad (55)$$

$$u_D[n] = \frac{T_D}{(T_D + N T_s)} u_D[n-1] + \frac{K_P N T_D}{(T_D + N T_s)} (e[n] - e[n-1]) \quad (56)$$

- Retangular para trás com filtro e derivada da saída:

$$u_D[n] = \frac{T_D}{(T_D + N T_s)} u_D[n-1] - \frac{K_P N T_D}{(T_D + N T_s)} (y[n] - y[n-1]) \quad (57)$$

Mapeamento...

Diferenciação...

Integração Numérica

Aproximação por...

Controle PID

Discretização de...

Homepage

Página de Rosto



Página 36 de 42

Voltar

Full Screen

Fechar

Desistir



Assim, a ação de controle corresponde à soma das três parcelas, ou seja:

$$u[n] = u_P[n] + u_I[n] + u_D[n] \quad (58)$$

Esta formulação é conhecida na literatura como PID posicional, tendo como vantagem o fato das parcelas proporcional, integrativa e derivativa serem obtidas separadamente.

Com controlador é digital, um *anti windup* consiste em “congelar” a ação integrativa quando há saturação:

```
% PID posicional com anti-windup (v.1)
if (u[n-1] >= sat || u[n-1] <= -sat)
    e_i[n]=0;
else
    e_i[n]=r[n]-y[n];
end
e[n] = r[n]-y[n];
u_P[n] = K_P*e[n];
u_I[n] = u_I[n-1] + (K_P*T_s/T_I)*e_i[n]; % Ex: backward
u_D[n] = T_D / (T_D+N*T_s) * u_D[n-1] - (K_P*N*T_D) / (T_D+N*T_s) * (y[n]-y[n-1]);
u[n]=u_P[n]+u_I[n]+u_D[n];
```

[Homepage](#)[Página de Rosto](#)[Página 37 de 42](#)[Voltar](#)[Full Screen](#)[Fechar](#)[Desistir](#)



Assume-se que a saturação do atuador é conhecida e igual a $\pm\text{sat}$.

Outra forma de anti-*windup* consiste no equivalente discreto do sistema da Figura 9, que pode ser realizada pela seguinte lógica (e_u representa a diferença do sinal de controle antes e depois da saturação):

```
% PID posicional com anti-windup (v.2)
e_u = u[n-1] - sat;
if (e_u > 0 || e_u < 0)
a = (T_s/T_t)*e_u; % Correção do anti-windup
else
a = 0;
e[n] = r[n]-y[n];
u_P[n] = K_P*e[n];
u_I[n] = u_I[n-1] + (K_P*T_s/T_I)*e[n] - a*e_u; % Ex: backward
u_D[n] = T_D/(T_D+N*T_s)*u_D[n-1]-(K_P*N*T_D)/(T_D+N*T_s)*(y[n]-y[n-1]);
u[n]=u_P[n]+u_I[n]+u_D[n];
```

Outra forma de PID digital é denominada PID incremental ou PID de velocidade. Tal forma é obtida tomando-se a diferença do



PID de posição no instante n e no instante $(n - 1)$, ou seja;

$$\Delta u[n] = u[n] - u[n - 1] = \Delta u_P[n] + \Delta u_I[n] + \Delta u_D[n] \quad (59)$$

Assim,

$$u[n] = u[n - 1] + \Delta u_P[n] + \Delta u_I[n] + \Delta u_D[n] \quad (60)$$

Assumindo a aproximação retangular para trás nas parcelas integrativa e derivativa, os termos incrementais são dados por:

$$\Delta u_P[n] = u_P[n] - u_P[n - 1] = K_P(e[n] - e[n - 1]), \quad (61)$$

$$\Delta u_I[n] = u_I[n] - u_I[n - 1] = \frac{K_P T_s}{T_I} \sum_{k=0}^n e[k] - \frac{K_P T_s}{T_I} \sum_{k=0}^{n-1} e[k] = \frac{K_P T_s}{T_I} e[n] \quad (62)$$

[Homepage](#)[Página de Rosto](#)[Página 39 de 42](#)[Voltar](#)[Full Screen](#)[Fechar](#)[Desistir](#)



e, para a derivada ideal,

$$\begin{aligned}\Delta u_D[n] &= u_D[n] - u_D[n-1] = \\ &= \frac{K_P T_D}{T_s} [(e[n] - e[n-1]) - (e[n-1] - e[n-2])] = \\ &= \frac{K_P T_D}{T_s} (e[n] - 2e[n-1] + e[n-2]).\end{aligned}\quad (63)$$

Se o método de Tustin for utilizado para a parcela integrativa, chega-se em:

$$\Delta u_I[n] = \frac{K_P T_s}{2T_I} (e[n] + e[n-1]),\quad (64)$$

Se a derivada da saída (com sinal trocado) for considerada ao invés da derivada do erro, tem-se:

$$\Delta u_D[n] = -\frac{K_P T_D}{T_s} (y[n] - 2y[n-1] + y[n-2])\quad (65)$$

[Homepage](#)[Página de Rosto](#)[Página 40 de 42](#)[Voltar](#)[Full Screen](#)[Fechar](#)[Desistir](#)



Ainda, se for considerada a forma realista com filtro na parcela derivativa e aproximação retangular para trás, tem-se que:

$$\Delta u_D[n] = \frac{T_D}{(T_D + NT_s)} \Delta u_D[n-1] - \frac{K_P NT_D}{(T_D + NT_s)} (y[n] - 2y[n-1] + y[n-2]). \quad (66)$$

Esta forma não é diretamente válida para P e PD.

Vantagens: pode reduzir problemas com precisão numérica no cálculo dos termos incrementais. Se houver algum erro em um passo do algoritmo, o sinal de controle ficará com o último valor válido.

O *anti-windup* na forma incremental pode ser incrementado como:

Mapeamento...

Diferenciação...

Integração Numérica

Aproximação por...

Controle PID

Discretização de...

Homepage

Página de Rosto



Página 41 de 42

Voltar

Full Screen

Fechar

Desistir



Mapeamento...

Diferenciação...

Integração Numérica

Aproximação por...

Controle PID

Discretização de...

Homepage

Página de Rosto



Página 42 de 42

Voltar

Full Screen

Fechar

Desistir

```
% PID incremental com anti-windup
e[n] = r[n]-y[n];
Du_P = K_P*(e[n]-e[n-1]);
Du_I=(K_P*T_s/(2*T_I))*(e[n] + e[n-1]);
u[n] = u[n-1] + Du_P + Du_I;
if (u[n] ≥ sat)
u[n] = sat;
elseif (u[n] ≤ - sat)
u[n] = -sat;
end
```