



## **Controle Automático III**

**Prof. Dr.Eng.  
Fernando Passold**

Última atualização: 6 de maio de 2020

“... acredito que a vida é uma coisa inteligente, que as coisas não acontecem por acaso.”  
(Steve Jobs)

GOOD THINGS  
COME TO THOSE  
WHO WAIT

GOOD THINGS  
COME TO THOSE  
WHO WORK THEIR  
ASSES OFF AND  
NEVER GIVE UP

Este material foi desenvolvido para a disciplina de “Controle Automático III” do curso de Engenharia Elétrica da UPF, que poderia ser chamada também de “Controle Discreto no Tempo” ou simplesmente de “Controle Digital Clássico”.

Este material não pretende ser auto-suficiente em si mesmo, apesar da quantidade de Apêndices, mas visa ser usado como continuidade para disciplinas anteriores na área de Controle Automático, uma vez que pressupõe que o leitor já tenha adquirido fundamentos anteriores na área de Controle Automático “clássico” no “mundo analógico”. Isto é, que esteja ao menos familiarizado com transformadas de Laplace e projetos de controladores (análogicos) usando Diagrama do Lugar das Raízes (*Root-Locus*) e preferencialmente que também, domine o projeto de controladores de Avanço-Atraso (*Lead-Lag*) no domínio frequêncial.

Este material foi criado usando software livre, multiplataforma e de domínio público: LaTeX (pacote MiKTeX / MacTeX) e editor TeXmaker. Foi adotada a fonte Palatino.

$G > \wedge \vee$

©Prof. Dr. Eng. Fernando Passold, 6 de maio de 2020.

# Sumário

<b>4 Teoria de Erros</b>	<b>7</b>
4.1 Introdução . . . . .	7
4.2 Realimentação unitária . . . . .	7
4.3 Tipos de Sistemas (ou FTMA( $z$ )'s) . . . . .	8
4.3.1 Detectando Componente(s) de ação Integral . . . . .	9
4.4 Entradas (Referências) típicas . . . . .	11
4.5 Determinando os erros . . . . .	11
4.5.1 Erros para Entrada Degrau . . . . .	12
4.5.2 Erros para Entrada Rampa . . . . .	13
4.6 Tabela Resumo . . . . .	15
4.7 Problema . . . . .	15
<b>5 Estabilidade</b>	<b>19</b>
5.1 Introdução . . . . .	19
5.2 Resposta Entrada Limitadas . . . . .	19
5.3 Pólos e Zeros no Plano- $z$ . . . . .	20
5.3.1 Caso 1) $F(z)$ com Pólos Reais . . . . .	20
5.3.2 Caso 2) $F(z)$ com Pólos Complexos . . . . .	21
5.4 Método de Jury . . . . .	24
5.4.1 Exemplos . . . . .	25
5.5 Determinando $K_{\max}$ . . . . .	27
5.6 Efeitos do Período de Amostragem . . . . .	29
5.7 Diagrama do Lugar das Raízes (RL) . . . . .	32
5.7.1 Construindo o Diagrama do Lugar das Raízes . . . . .	33
5.7.2 Exemplos de Traçados de RL . . . . .	34
<b>6 Controladores Digitais Clássicos</b>	<b>43</b>
6.1 Introdução . . . . .	43
6.2 Formato Geral . . . . .	44
6.3 Controlador Proporcional . . . . .	46
6.3.1 Determinando $K_{\max}$ . . . . .	48
<b>7 Estudo de Caso</b>	<b>55</b>

7.1	Planta Adotada . . . . .	55
7.2	Controlador Proporcional . . . . .	56
7.3	Controlador Dead-Beat . . . . .	58
7.3.1	Exemplo <sub>1</sub> : O controlador Deadbeat . . . . .	58
7.3.2	Exemplo <sub>2</sub> : Projeto de Controlador Deadbeat . . . . .	59
<b>8</b>	<b>Projeto por Emulação</b>	<b>67</b>
8.1	Introdução . . . . .	67
8.2	Método MPZ . . . . .	68
8.2.1	Plano-s × Plano- $\hat{s}$ . . . . .	68
8.3	Método de Euler . . . . .	70
8.3.1	Exemplo <sub>1</sub> : Caso de um Controlador de 1a-ordem . . . . .	70
8.3.2	Exemplo <sub>2</sub> : Caso de um Controlador por Avanço de Fase . . . . .	70
8.4	Método de Tustin . . . . .	74
8.4.1	Exemplo <sub>1</sub> : Equação de diferenças usando o Método de Tustin . . . . .	75
8.4.2	Exemplo <sub>2</sub> : Projeto de Controlador Digital usando o Método de Tustin . . . . .	76
8.5	Transformações Bilineares . . . . .	78
<b>9</b>	<b>PID</b>	<b>79</b>
9.1	Formas de Implementação . . . . .	79
9.1.1	Formas de Integração Numérica . . . . .	80
9.1.2	Formas de Diferenciação Numérica . . . . .	81
9.1.3	Formas de Implementação das Ações Proporcional-Integral-Derivativa . . . . .	81
9.2	Ajuste de Controladores PID . . . . .	81
9.3	Acrescentando o Filtro Derivativo . . . . .	82
9.4	Acrescentando ação anti-windup . . . . .	82
<b>10</b>	<b>Abordagem por Espaço de Estados</b>	<b>83</b>
10.1	Introdução . . . . .	83
10.2	Definições . . . . .	84
10.3	Ferramentas de Controle Clássico . . . . .	86
10.4	Teoria de Controle no Espaço de Estados . . . . .	86
10.5	Ferramentas para projeto em espaço de estados . . . . .	88
<b>A</b>	<b>Equações Auxiliares</b>	<b>91</b>
A.1	Tabela de Derivadas . . . . .	91
A.2	Equação do Segundo Grau . . . . .	91
A.3	Relações de Euler . . . . .	91
A.4	Progressão Geométrica . . . . .	91
A.5	Definição da Transformada $Z$ . . . . .	93
A.6	Propriedades da Transformada $Z$ . . . . .	93
A.7	Tabela de Transformadas $Z$ . . . . .	95

---

<b>Anexos</b>	<b>90</b>
<b>B Respostas temporais no Plano-<math>\mathcal{Z}</math></b>	<b>97</b>
B.1 Casos de Pólos Reais . . . . .	97
B.2 Pólos Complexos . . . . .	98
<b>C Transformada <math>\mathcal{Z}</math> - Uso do MATLAB</b>	<b>99</b>
C.1 Transformada $\mathcal{Z}$ Inversa . . . . .	99
C.2 ZOH ou BoG(z) . . . . .	102
<b>D Resposta ao Degrau</b>	<b>105</b>
D.1 Sistemas com $0 < \zeta < 1$ . . . . .	106
D.2 Relações entre plano-s e plano-z . . . . .	108
<b>E Revisão Controladores Clássicos</b>	<b>111</b>
E.1 Introdução . . . . .	111
E.2 Características principais dos controladores “clássicos” . . . . .	112
<b>F Comandos do MATLAB</b>	<b>115</b>
<b>G Gráficos de <math>u[kT]</math> e <math>e[kT]</math></b>	<b>121</b>
G.1 Gráfico de $u[kT]$ . . . . .	121
G.2 Gráfico de $e[kT]$ . . . . .	123



# Capítulo 4

## Teoria de Erros

### 4.1 Introdução

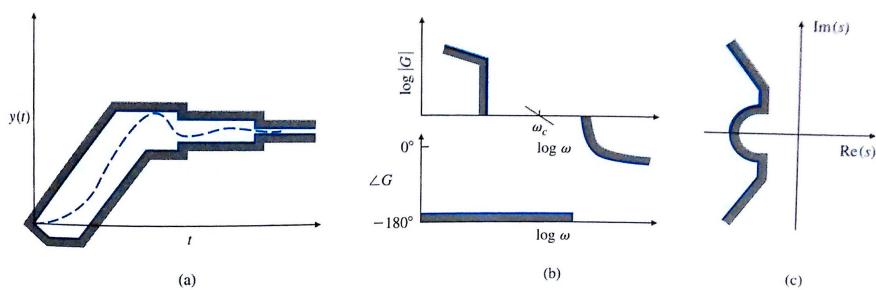
Relação com as características desejáveis num sistema de controle.

4\_Teoria\_Erros/  
teoria\_erro.tex

As especificações de um sistema de controle incluem:

- desempenho em regime permanente (precisão);
- desempenho em regime transitório (formas de onda);
- robustez (degradação do controle frente à mudanças no sistema).

A figura 4.1 mostra especificações típicas para um sistema de controle, mostrando as restrições vinculadas à uma entrada degrau para o sistema em malha-fechada.



Adaptado de Franklin et al. (1994).

Figura 4.1: Exemplos de especificações de controle quanto à: (a) resposta temporal, (b) resposta em frequência e (c) especificações de pólos-zeros.

### 4.2 Sistema “padrão” de realimentação unitária

Seja um típico sistema de controle em malha fechada do tipo SISO (single-input, single-output) como o mostrado na figura 4.2.

Note que:

- este sistema controla o valor medido do erro, não a variável de saída.
- a unidade para medição de  $Y$  é a mesma que a adotada para a referência  $Y_r$ , para que o bloco somador atue de forma conveniente.

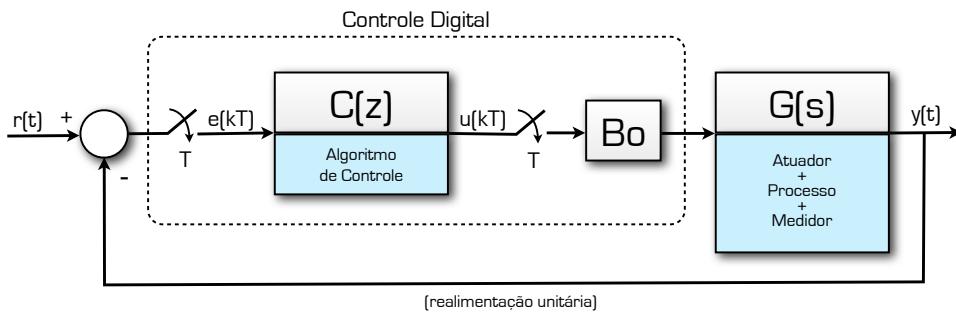


Figura 4.2: Sistema “padrão” de realimentação unitária.

- qualquer sistema em malha fechada com realimentação unitária ( $H(s) = 1$ ), tende a igualar a entrada e a saída (Ogata, 1995, p. 99 e 210).
- $H(s)$  atua como um “conversor de unidades” entre  $Y$  e  $Y_r$ .

Podemos redesenhar o diagrama anterior (figura 4.2) como o mostrado na figura 4.3 e então:

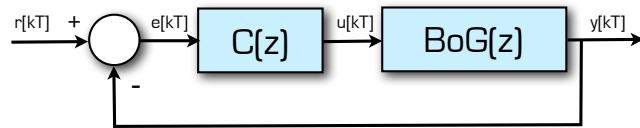


Figura 4.3: Sistema “padrão” no formato digital.

$$\frac{Y(z)}{R(z)} = \frac{C(z) \cdot BoG(z)}{1 + C(z)BoG(z)} = \frac{FTMA(z)}{1 + FTMA(z)}$$

isolando o erro:

$$E(z) = R(z) - Y(z) = R(z) \cdot \left[ 1 - \frac{FTMA(z)}{1 + FTMA(z)} \right] = R(z) \left[ \frac{1 + FTMA(z) - FTMA(z)}{1 + FTMA(z)} \right]$$

notamos então que o comportamento do erro é dado por:

$$E(z) = \left[ \frac{1}{1 + FTMA(z)} \right] \cdot R(z) \quad (4.1)$$

ou seja, que o erro é função de  $R$  (referência) e da  $FTMA(z)$ , ou:

$$\text{Erro} = f[R, FTMA]$$

O que acontece então com o erro em regime permanente (precisão)?

Procedendo com o cálculo do erro em regime permanente (quando  $t \rightarrow \infty$ ):

$$\lim_{k \rightarrow \infty} e(kT) = \lim_{z \rightarrow 1} (z-1) \cdot E(z)$$

Para o nosso caso:

$$\lim_{k \rightarrow \infty} e(kT) = \lim_{z \rightarrow 1} (z-1) \left[ \frac{1}{1 + FTMA(z)} \right] R(z) \quad (4.2)$$

Para continuar com a análise, necessitamos levantar hipóteses sobre entradas (referências) do sistema e ordem de sua equação transferência ( $FTMA(z)$ ). Normalmente se trabalha com entradas: degrau, rampa e parábola.

## 4.3 Tipos de Sistemas (ou $FTMA(z)$ 's)

Os processos podem ser classificados em relação ao número de integradores presentes nos mesmos.

De forma geral podemos representar a FTMA(z) como sendo:

$$\text{FTMA}(z) = \frac{N(s)}{D(s)} = \frac{K(z-z_1)(z-z_2)\cdots(z-z_i)}{(z-1)^n(z-p_1)(z-p_2)\cdots(z-p_j)}$$

onde: K é uma constante qualquer (eventualmente um ganho de algum controlador);  $j \geq i$ , e n se refere ao número de integradores presentes dentro da FTMA(z).

Mas como distinguir um integrador (ou mais) dentro da FTMA(z) (no plano-z)?

### 4.3.1 Detectando Componente(s) de ação Integral

- Como seria uma integração numérica?

#### Exemplo 1: Integração numérica Retangular

A figura 4.4 mostra um caso de integração numérica retangular.

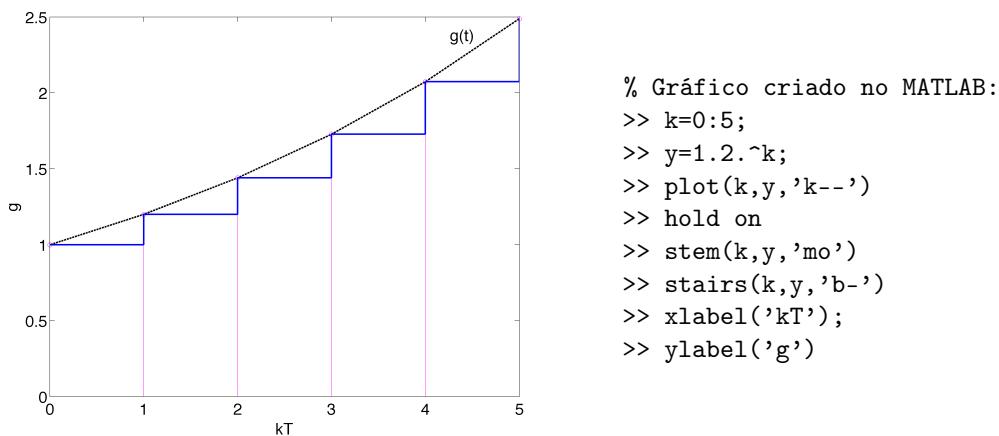


Figura 4.4: Método de integração numérica retangular.

onde:  $i(k)$  = integral de 0 à k. Então:

$$\begin{aligned} i[0] &= 0 \\ i[1] &= i[0] + T \cdot g[0] \\ i[2] &= i[1] + T \cdot g[1] \\ &\vdots \\ i[k] &= i[k-1] + T \cdot g[k-1] \end{aligned} \tag{4.3}$$

Note que a expressão (4.3) está no formato de uma equação de diferenças, então:

$$\begin{aligned} I(z) &= z^{-1} \cdot I(z) + T \cdot z^{-1} \cdot G(z) \\ I(z)[1 - z^{-1}] &= T z^{-1} G(z) \\ I(z) &= \left( \frac{T z^{-1}}{1 - z^{-1}} \right) G(z) \end{aligned}$$

ou:

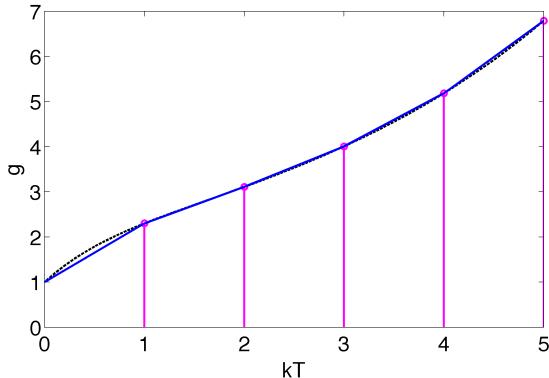
$$I(z) = \frac{T}{(z-1)} \cdot G(z) \tag{4.4}$$

Repare no pólo em  $z = 1$ .

Se fosse adotado outro método de integração numérica, ainda assim teríamos o termo  $(z-1)$  aparecendo no denominador da função.

**Exemplo 2: Integração numérica Trapezoidal**

Acompanhando pela figura 4.5:



```
>> % Gráfico no MATLAB:  
>> % fplot(@(t) 1.4.^t-0.9.^t-0.2.^t+2, [0 5])  
>> t=0:0.1:5; % tempo variando de 0,1 segundos  
>> k=0:5; % amostras a cada T=1 seg  
>> y=1.4.^t-0.9.^t-0.2.^t+2;  
>> yd=1.4.^k-0.9.^k-0.2.^k+2; % versão discreta  
>> figure; plot(t,y,'k--');  
>> hold on;  
>> stem(k,yd,'mo');  
>> plot(k,yd,'b-')
```

Figura 4.5: Método de integração numérica trapezoidal.

Lembrando que a área de um trapézio é dada por:

$$A = \frac{(B + b) \cdot h}{2}$$

agora obteremos algo como:

$$\begin{aligned} i[0] &= 0 \\ i[1] &= i[0] + \frac{T}{2} (g[0] + g[1]) \\ &\vdots \\ i[k] &= i[k-1] + \frac{T}{2} (g[k-1] + g[k]) \end{aligned} \quad (4.5)$$

Transformando (4.5) numa expressão em  $z$ , obtemos:

$$\begin{aligned} I(z) &= z^{-1} \cdot I(z) + \frac{T}{2} (z^{-1} + 1) \cdot G(z) \\ &= \frac{T/2}{(1-z^{-1})} (1+z^{-1}) \cdot G(z) \\ I(z) &= \frac{T/2}{(z-1)} (z+1) G(z) \end{aligned} \quad (4.6)$$

Note novamente em (4.6) o termo  $(z-1)$  no denominador, ou seja, que volta a aparecer o pólo em  $z=1$ .

Assim, de maneira geral, se pode afirmar que uma função transferência contendo integradores é do tipo:

$$FTMA(z) = \frac{N(z)}{D(z)} = \frac{N(z)}{(z-1)^n D'(z)}$$

ou

$$FTMA(z) = \frac{N(z)}{D(z)} = \frac{K(z-z_1)(z-z_2)\cdots(z-z_i)}{(z-1)^n (z-p_1)(z-p_2)\cdots(z-p_j)}$$

onde  $n$  se refere ao número de integradores presentes na função. Note que caso exista ação integral, está presente ao menos um pólo em  $z=1$  ( $n > 1$ ).

E assim, se pode dizer que a  $FTMA(z)$  é do tipo 0, 1, 2 ou mais, dependendo do valor de  $n$  (número de integradores).

Voltando aos tipos de  $FTMA(z)$ , podemos então classificar a  $FTMA(z)$  em:

- Tipo 0:  $n = 0$ , corresponde a um processo sem integrador:

$$G(z) = \frac{N(z)}{D(z)} = \frac{K(z+z_1)(z+z_2)\dots}{1(z+p_1)(z+p_2)\dots}$$

- Tipo 1:  $n = 1$ , sistema com 1 integrador:

$$G(z) = \frac{N(z)}{D(z)} = \frac{K(z+z_1)(z+z_2)\dots}{(z-1)(z+p_1)(z+p_2)\dots}$$

- Tipo 2:  $n = 2$ , sistema com 2 integradores:

$$G(z) = \frac{N(z)}{D(z)} = \frac{K(z+z_1)(z+z_2)\dots}{(z-1)^2(z+p_1)(z+p_2)\dots}$$

## 4.4 Entradas (Referências) típicas

Na área de controle as entradas mais comuns adotadas para excitar ou comandar um sistema são:

- Entrada Degrau:

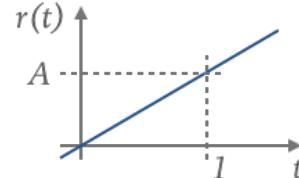
$$u(t) = \begin{cases} 0, & t < 0 \\ A \cdot 1, & t \geq 0 \end{cases} \xrightarrow{\mathcal{Z}} U(z) = \frac{A \cdot z}{z-1}$$

onde  $A$  corresponde à amplitude da entrada degrau;

- Entrada Rampa:

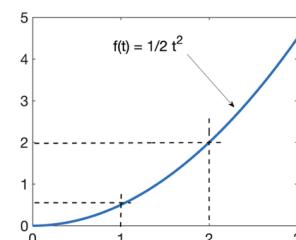
$$r(t) = \begin{cases} 0, & t < 0 \\ A \cdot t, & t \geq 0 \end{cases} \xrightarrow{\mathcal{Z}} R(z) = \frac{A \cdot T z}{(z-1)^2}$$

onde  $A$  corresponde à inclinação (*slope*) da rampa.



- Entrada Parabólica:

$$r(t) = \begin{cases} 0, & t < 0 \\ A \cdot \frac{1}{2} t^2, & t \geq 0 \end{cases} \xrightarrow{\mathcal{Z}} R(z) = \frac{A \cdot T^2 z(z+1)}{2(z-1)^3}$$



- Senoíde(!?):

$$r(t) = A \cdot \sin(\alpha kT) \xrightarrow{\mathcal{Z}} R(z) = \frac{A \cdot z \sin(\alpha T)}{z^2 - [2 \cos(\alpha T)]z + 1}$$

## 4.5 Determinando os erros

Uma vez revisados os tipos de sinais de entrada (referências) tipicamente adotados em sistemas de controle e considerando diferentes tipos de sistema (sistema tipo 0 = sem integrador; sistema tipo 1 = 1 integrador. etc), podemos agora passar a analisar como fica o erro para diferentes tipos de entradas e sistemas.

### 4.5.1 Erros para Entrada Degrau

A entrada degrau (unitária) é definida como:

$$R(z) = U(z) = \frac{z}{z-1}$$

neste caso a equação do erro fica como:

$$\begin{aligned} e_{\text{step}}(\infty) &= \lim_{k \rightarrow \infty} e(kT) = \lim_{z \rightarrow 1} (z-1) \cdot \frac{1}{1 + \text{FTMA}(z)} \cdot R(z) \\ &= \lim_{z \rightarrow 1} (z-1) \cdot \frac{1}{1 + \text{FTMA}(z)} \cdot \frac{z}{(z-1)} \\ e_{\text{step}}(\infty) &= \frac{1}{1 + \lim_{z \rightarrow 1} \text{FTMA}(z)} = \frac{1}{1 + K_p} \end{aligned} \quad (4.7)$$

onde o termo  $\lim_{z \rightarrow 1} \text{FTMA}(z)$  é conhecido como  $K_p$ =**constante (de erro estático) de posição** (similar à mesma constante de erro dos sistemas contínuos no tempo), ou seja:

$$K_p = \lim_{z \rightarrow 1} \text{FTMA}(z) \quad (4.8)$$

Falta agora analisar  $K_p$  e  $e_{\text{step}}(\infty)$  para diferentes tipos de sistemas.

1. Sistema tipo 0: **sem integrador**:

Se a  $\text{FTMA}(z)$  **não possui** integradores, teremos uma equação como:

$$\text{FTMA}(z) = \frac{K(z-z_1)(z-z_2) \cdots (z-z_i)}{(z-p_1)(z-p_2) \cdots (z-p_j)}$$

$$K_p = \lim_{z \rightarrow 1} \text{FTMA}(z)$$

$$K_p = \lim_{z \rightarrow 1} \frac{K(z-z_1)(z-z_2) \cdots (z-z_i)}{(z-p_1)(z-p_2) \cdots (z-p_j)} = \frac{\text{cte}_1}{\text{cte}_2} = \text{cte}_3$$

$$e_{\text{step}} = \frac{1}{1 + K_p} = \frac{1}{1 + \text{cte}_3} = \text{cte}_4$$

ou seja,  $K_p \neq 0$  e consequentemente  $e_{\text{step}} \neq 0$  (não nulo, **mas** finito).

Significa que, para sistemas do tipo 0 (sem integrador), **independente do ganho que for adotado no sistema em MF**, o erro nunca será eliminado! Sempre será um valor diferente de 0 – mesmo para entrada degrau (menor ordem).

2. Sistema tipo 1 (**1 integrador**):

Se a  $\text{FTMA}(z)$  contém pelo menos 1 integrador, então a  $\text{FTMA}(z)$  é do tipo:

$$\text{FTMA}(z) = \frac{K(z-z_1)(z-z_2) \cdots (z-z_i)}{(z-1)(z-p_1)(z-p_2) \cdots (z-p_j)}$$

$$K_p = \lim_{z \rightarrow 1} \text{FTMA}(z) = \lim_{z \rightarrow 1} \frac{\text{cte}_1}{(z-1)^0 \cdot (z-p_1)(z-p_2) \cdots (z-p_j)} = \infty$$

e o erro fica:

$$e_{\text{step}} = \frac{1}{1 + K_p} = \frac{1}{1 + \infty} = 0$$

neste caso,  $K_p \rightarrow \infty$  e consequentemente  $e(\infty) = 0$ .

Isto significa que se o sistema possuir **pelo menos um integrador**, obteremos **erro nulo para entrada degrau**, (variando pouca coisa em regime permanente, tanto menos quanto menor for a quantização adotada no sistema).

**Exemplo:** Simulando sistema simples em malha-fechada com 1 integrador.

Repare no resultado obtido na simulação mostrada na figura 4.6.

Simule o mesmo sistema sem o bloco “Integrador” e repare nas respostas obtidas.

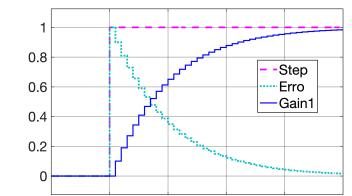
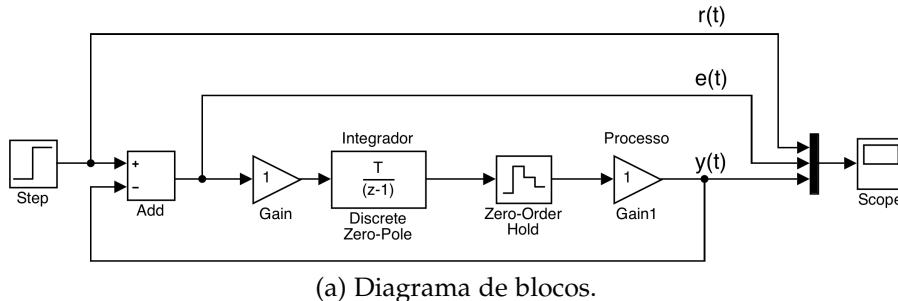


Figura 4.6: Processo em MF contendo apenas um integrador discreto (numérico).

## 4.5.2 Erros para Entrada Rampa

Neste caso:

$$R(z) = \frac{Tz}{(z-1)^2}$$

e neste caso  $e_{ramp}(\infty)$  fica:

$$\begin{aligned} e_{ramp}(\infty) &= \lim_{k \rightarrow \infty} e(kT) = \lim_{z \rightarrow 1} \frac{(z-1)}{\left[1 + FTMA(z)\right]} \frac{Tz}{(z-1)^2} \\ &= \lim_{z \rightarrow 1} \frac{Tz}{(z-1) + (z-1)FTMA(z)} \\ e_{ramp}(\infty) &= \frac{T}{\lim_{z \rightarrow 1} (z-1)FTMA(z)} = \frac{T}{K_v} \end{aligned} \quad (4.9)$$

onde  $K_v = \lim_{z \rightarrow 1} (z-1)FTMA(z)$ , corresponde ao **ganho (de erro estático) de velocidade**.

Da mesma forma que procedemos para o caso anterior, vamos analisar o comportamento do erro, dependendo do tipo do sistema.

1. Sistema tipo 0 (sem integrador):

A  $FTMA(z)$  seria uma equação do tipo:

$$FTMA(z) = \frac{K(z-z_1)(z-z_2) \cdots (z-z_i)}{(z-p_1)(z-p_2) \cdots (z-p_j)}$$

$$K_v = \lim_{z \rightarrow 1} (z-1)FTMA(z)$$

$$K_v = \lim_{z \rightarrow 1} \frac{(z-1)^0 K(z-z_1)(z-z_2) \cdots (z-z_i)}{(z-p_1)(z-p_2) \cdots (z-p_j)} = 0$$

$$e_{ramp}(\infty) = \frac{T}{K_v} = \frac{T}{0} = \infty$$

Isto é, o limite não converge, o **erro só aumenta** à medida que a rampa continua (e o tempo passa).

2. Sistema tipo 1 (**1 integrador**):

Se a  $FTMA(z)$  possui 1 integrador então sua equação é do tipo:

$$FTMA(z) = \frac{K(z-z_1)(z-z_2) \cdots (z-z_i)}{(z-1)(z-p_1)(z-p_2) \cdots (z-p_j)}$$

$$K_v = \lim_{z \rightarrow 1} (z-1) \cdot FTMA(z)$$

$$K_v = \lim_{z \rightarrow 1} \frac{(z-1) \cdot (z-z_1) \cdots (z-z_n)}{(z-1)^{\cancel{1}}(z-p_1) \cdots (z-p_n)}$$

$$K_v = \frac{cte_1}{cte_2} = cte_3$$

$$e_{Ramp}(\infty) = \frac{T}{K_v} = \frac{T}{cte_3} = cte_4$$

Isto é, para entrada rampa e sistemas tipo 1,  $e_{Ramp}(\infty) \neq 0$ : um valor finito, **mas limitado**; isto é, existe um erro mas este fica restrito à um certo valor.

3. Sistema tipo 2 (com **2 integradores**):

Se  $FTMA(z)$  possui 2 integradores então sua equação é do tipo:

$$FTMA(z) = \frac{K(z-z_1)(z-z_2) \cdots (z-z_i)}{(z-1)^{\cancel{2}}(z-p_1)(z-p_2) \cdots (z-p_j)}$$

$$K_v = \lim_{z \rightarrow 1} (z-1) \cdot FTMA(z)$$

$$K_v = \lim_{z \rightarrow 1} \frac{(z-1) \cdot K(z-z_1) \cdots (z-z_n)}{(z-1)^{\cancel{2}}(z-p_1) \cdots (z-p_m)}$$

$$K_v = \frac{cte_1}{\underbrace{\lim_{z \rightarrow 1} (z-1) \cdot cte_2}_{\rightarrow 0}} = \frac{cte_1}{\rightarrow 0} = \infty$$

$$e_{Ramp}(\infty) = \frac{T}{K_v} = \frac{T}{\infty} = 0$$

Isto significa que, neste caso, conseguimos anular o erro para uma entrada rampa se estiverem presentes **2 integradores** no sistema à ser controlado (na  $FTMA(z)$ ).

## Conclusão Final

Poderíamos continuar o raciocínio com o cálculo envolvendo erro para entradas de ordem superior (parábola, polinômios de 3a-ordem, etc), mas a análise já realizada permite alcançar uma conclusão final:

Num sistema à ser controlado, acrescentamos tantos integradores no processo ( $FTMA(z)$  contendo **polos em  $z=1$** ) quanto os necessários para eliminar (zerar) o erro para o tipo de entrada desejada.

## 4.6 Tabela Resumo

Segue uma **tabela resumo** sobre Teoria de Erros (ou Precisão):

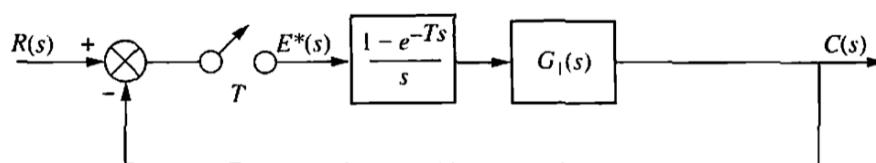
Entradas:		Degrau	Rampa	Parábola
Sistema:		$\frac{\alpha z}{z-1}$	$\frac{\alpha T z}{(z-1)^2}$	$\frac{\alpha T^2 z(z+1)}{2(z-1)^3}$
Tipo 0	$F(z) = \frac{N(z)}{D(z)}$	$\frac{\alpha}{1+K_p}$	$\infty$	$\infty$
Tipo 1	$F(z) = \frac{N(z)}{(z-1)D'(z)}$	0	$\frac{\alpha T}{K_v}$	$\infty$
Tipo 2	$F(z) = \frac{N(z)}{(z-1)^2 D'(z)}$	0	0	$\frac{\alpha T^2}{K_a}$
onde: $K_p = \lim_{z \rightarrow 1} FTMA(z)$ = ganho de posição. $K_v = \lim_{z \rightarrow 1} (z-1) FTMA(z)$ = ganho de velocidade. $K_a = \lim_{z \rightarrow 1} (z-1)^2 FTMA(z)$ = ganho de aceleração.				

Tabela 4.1: Tabela resumo sobre erros.(Ogata, 1995, pág. 355)

## 4.7 Problema

**Problema:** Encontre o erro em regime permanente para entrada degrau, rampa e parabólica, para o sistema mostrado na figura 4.7 quando:

$$G_1 = \frac{10}{s(s+1)}$$



(Extraído de Nise (2011), Example 13-9.)

Figura 4.7: Avaliação de erros de regime permanente para sistema realimentado de controle digital.

**Obs.:** considere  $T = 0,1$  (segundos) para o período de amostragem.

**Solução:** Primeiramente determinamos a FTMA(s), como sendo o produto de ZOH com a planta:

$$FTMA(s) = B_0(s) \cdot G(s) = \frac{10(1 - e^{-1 \cdot T \cdot s})}{s^2(s+1)}$$

A transformada  $\mathcal{Z}$  resulta em:

$$\begin{aligned} \text{FTMA}(z) &= \text{BoG}(z) = (1 - z^{-1}) \mathcal{Z} \left\{ \frac{G(s)}{s} \right\} \\ &= (1 - z^{-1}) \mathcal{Z} \left\{ \frac{10}{s^2(s+1)} \right\} \\ &= 10(1 - z^{-1}) \mathcal{Z} \left\{ \left[ \frac{1}{s^2} - \frac{1}{s} + \frac{1}{(s+1)} \right] \right\} \\ &= 10(1 - z^{-1}) \left[ \frac{Tz}{(z-1)^2} - \frac{z}{(z-1)} + \frac{z}{(z-e^{-T})} \right] \\ \text{FTMA}(z) &= 10 \left[ \frac{T}{(z-1)} - 1 + \frac{(z-1)}{(z-e^{-T})} \right] \end{aligned}$$

Para a **entrada degrau**, ficamos com:

$$K_p = \lim_{z \rightarrow 1} \text{FTMA}(z) = \infty; \quad \therefore \quad e_{\text{Degrau}}(\infty) = \frac{1}{1 + K_p} = 0 \quad \Rightarrow \quad (\text{erro nulo})$$

Para uma **entrada rampa**, teremos:

$$K_v = \frac{1}{T} \lim_{z \rightarrow 1} (z-1) \text{FTMA}(z) = 10; \quad \therefore \quad e_{\text{Rampa}}(\infty) = \frac{1}{K_v} = 0,1 \quad (\text{erro } \neq 0, \text{não nulo, mas limitado!})$$

Para a **entrada parabólica**:

$$K_a = \frac{1}{T^2} \lim_{z \rightarrow 1} (z-1)^2 \text{FTMA}(z) = 0; \quad \therefore \quad e(\infty) = \frac{1}{K_a} = \infty \quad \Rightarrow \quad (\text{erro infinito; não converge})$$

Comprovando mediante simulações... e com a ajuda do Matlab:

```
>> % (chl3p6) NISE, pag. 771, example 13.9
>> T=0.1; % período de amostragem adotado
>> numg=10;
>> deng=poly([0 -1]); % informando os pólos de G(s)
>> G=tf(numg,deng) % criando uma transfer function no MATLAB, no caso para G(s)
G =
  10
  _____
  s^2 + s
>> % Falta incluir (e calcular) sustentador de ordem zero (ZOH) ou BoG(z)
>> BoG=c2d(G, T, 'zoh'); % Calcula BoG(z) usando sustentador de ordem-zero
>> zpk(BoG) % apresenta BoG(z) no formato de ganho/zeros/pólos
0.048374 (z+0.9672)
  _____
  (z-1) (z-0.9048)

Sample time: 0.1 seconds
Discrete-time zero/pole/gain model.

>> Kp=dcgain(BoG) % Calculando Kp (BoG(z)|z=1)

Kp =
  Inf

>> BoGKv=BoG* tf([1 -1],1,T); %
>> zpk(BoGKv) % apenas para entender o que o MATLAB está fazendo
0.048374 (z-1) (z+0.9672)
  _____
  (z-1) (z-0.9048)

Sample time: 0.1 seconds
```

Discrete-time zero/pole/gain model.

```

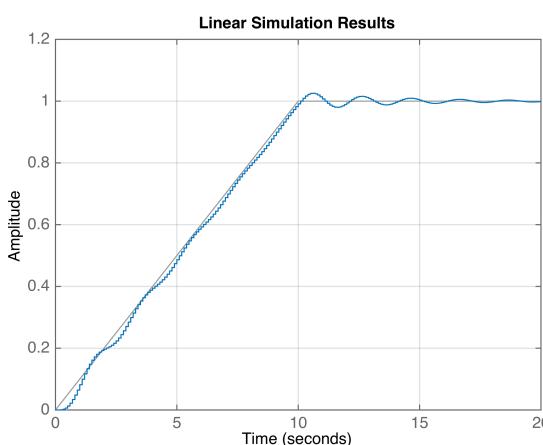
>> Kv=dcgain(BoGKv)

Kv = 1.0000
>> % Simulando a resposta deste sistema para entrada rampa+degrau
>> ftmf = feedback( 1*BoG, 1);
>> step(ftmf); % gera um gráfico onde percebemos que com K=1
>> % (ganho de malha fechada unitário), é gerado um overshoot de 77,9%
>> % e tempo de resposta, ts = 15,3 segundos
>> % fabricando vetor de entrada u[kT]
>> t=o:T:20; % simulando os 20 primeiros segundos
>> u=0.1*t; % criando a rampa com inclinação suave (0.1)
>> % a rampa leva 10 segundos para atingir amplitude 1,0
>> % mas não queremos só uma rampa; truncando rampa apos t=10 segundos
>> 10/T
ans = 100
>> % 10 segundos -> k=100
>> u(101:201)=1; % truncando a rampa
>> figure; stairs(t,u) % Verificando rampa criada
>> % Aplicando a rampa sobre a FTMF (não a FTMA)
>> ftmf=feedback(BoG,1);

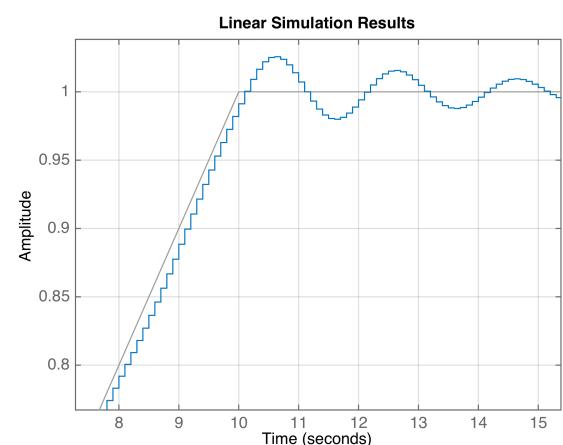
>> zpk(ftmf)
0.048374 (z+0.9672)

(z^2 - 1.856z + 0.9516)
Sample time: 0.1 seconds
Discrete-time zero/pole/gain model.
>> % Usaremos a função 'dlsim' que infelizmente não trabalha com tranfer-function:
>> [numd,dend]=tfdata(ftmf,'v');
>> figure; dlsim(numd,dend,u2)
>> % note que ' xlabel', Time vai até 200 (amostras) equivalente à 20 segundos
>> % Já a função 'lsim' trabalha com tranfer functions:
>> figure; lsim(ftmf,u) % gera o mesmo resultado que 'dlsim'
```

A última parte do código anterior gera o gráfico mostrado na figura: 4.8.



(a) Resposta de k=0 até 200



(b) 'Zoom' destacando parte da resposta.

Figura 4.8: Resposta a entrada rampa+degrau.



# Capítulo 5

## Estabilidade

### 5.1 Introdução

5\_Estabilidade/  
5\_estabilidade.tex

Trata-se do **objetivo** mais importante para um sistema de controle.

Para sistemas lineares e invariantes no tempo são BIBO estáveis (entrada limitada e saída limitada) se a aplicação de uma entrada limitada produz uma saída limitada. Mas atenção, este conceito é suficiente para sistemas contínuos no tempo. Para sistemas discretos (amostrados) ainda depende do período de amostragem adotado!

A estabilidade de um sistema em malha fechada depende de:

- Ganhos;
- Pólos;
- Zeros;
- Atrazos;
- $T$ , período de amostragem.

### 5.2 Regime transitório de um sistema submetido a uma entrada limitada (degrau)

Suponha que temos o sistema mostrado na figura 5.1.

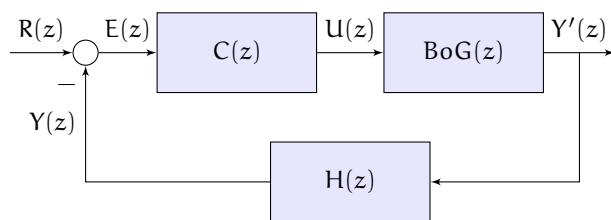


Figura 5.1: Sistema em malha fechada.

onde:

$$\frac{Y(z)}{R(z)} = \frac{C(z)BoG(z)}{1 + C(z)BoG(z)H(z)} = FTMF(z) = \frac{N(z)}{D(z)}$$

conforme mostra a figura 5.2.

Da figura 5.2 percebemos que, se for aplicada uma entrada degrau,  $Y(z)$  fica:

$$Y(z) = R(z) \cdot FTMF(z) = \frac{z}{(z-1)} \cdot \frac{N(z)}{D(z)} \quad (5.1)$$



Figura 5.2: Função de malha-fechada.

onde:  $\frac{z}{(z-1)}$  é a transformada  $Z$  do degrau.

Os zeros finitos de  $FTMF(z)$  são as raízes de  $N(z) = 0$ . E os pólos de  $FTMF(z)$  são as raízes de  $D(z) = 0$ .

Se o método da decomposição em termos simples (frações parciais) for aplicado sobre a equação 5.1 teremos:

$$\frac{Y(z)}{z} = \underbrace{\frac{c_0}{z-1}}_{\text{Pólos Reais}} + \underbrace{\sum_i \frac{A_i}{z-p_i}}_{\text{Pólos Complexos}} + \underbrace{\sum_j \left( \frac{A_j}{(z-p_j)} + \frac{\bar{A}_j}{(z-\bar{p}_j)} \right)}_{\text{Pólos Complexos}} + \underbrace{\sum_r \sum_{t=1}^{m_r} \frac{B_{rt}}{(z-p_r)^t}}_{\text{Pólos Repetidos}} \quad (5.2)$$

Lembrando que:

$$A_i = \frac{FTMF(z)(z-p_i)}{z} \Big|_{z=p_i}$$

cujos resíduos, no caso de pólos complexos, resulta em:

$$\begin{aligned} A_j &= a \pm jb = Ae^{j\phi} && \text{onde:} && p_j &= \rho e^{j\theta} \\ p_j &= \alpha \pm \beta = \rho e^{j\theta} && && c_j &= A e^{j\varphi} \end{aligned}$$

e, para o caso de pólos repetidos, o cálculo dos resíduos fica como:

$$B_{rt} = \frac{1}{(m_r-t)!} \left\{ \frac{d^{m_r-t}}{dz^{m_r-t}} \frac{(z-p_r)^{m_r}}{z} FTMF(z) \right\} \Big|_{z=p_r}$$

Lembrando ainda que a transformada inversa  $Z$  de um polo real é dada por:

$$z^{-1} \left\{ \frac{A_i z}{z-p_i} \right\} = A_i \cdot p_i^k$$

obtemos:

$$y(kT) = c_0 + \underbrace{\sum_i A_i p_i^k}_{\text{Pólos Reais}} + \underbrace{\sum_j (A_j p_j^k + \bar{A}_j \bar{p}_j^k)}_{\text{Pólos Complexos}} + \underbrace{\sum_r \sum_{t=1}^{m_r} B_{rt} \binom{k}{j-1} p_r^{k-t-1}}_{\text{Pólos Repetidos}} \quad (5.3)$$

onde:  $c_0$  = termo em função da entrada (define o regime permanente da saída); os outros termos ( $\sum$ 's) = função das características de  $F(z)$ , definem o regime transitório que depende dos pólos de  $F(z)$  ponderados pelos coeficientes  $A_i$  ou  $A_j$ , cujas magnitudes são determinadas pelo numerador de  $F(z)$ .

## 5.3 Representação de Pólos no Plano Complexo (Z)

A seguir estudaremos como os pólos contribuem na resposta  $y(kT)$

### 5.3.1 Caso 1) $F(z)$ com Pólos Reais

Neste caso,  $y(kT)$  depende dos termos  $c_i p_i^k$ . A figura 5.3 mostra os mapeamentos possíveis para pólos no plano  $Z$  e suas respectivas repercuções no domínio amostrado.

Analizando a figura 5.3 nota-se:

**Caso 1 :**  $p_i > 1$ , polo instável, saída divergente, tanto mais quanto mais distante o polo estiver de  $z = 1$ .

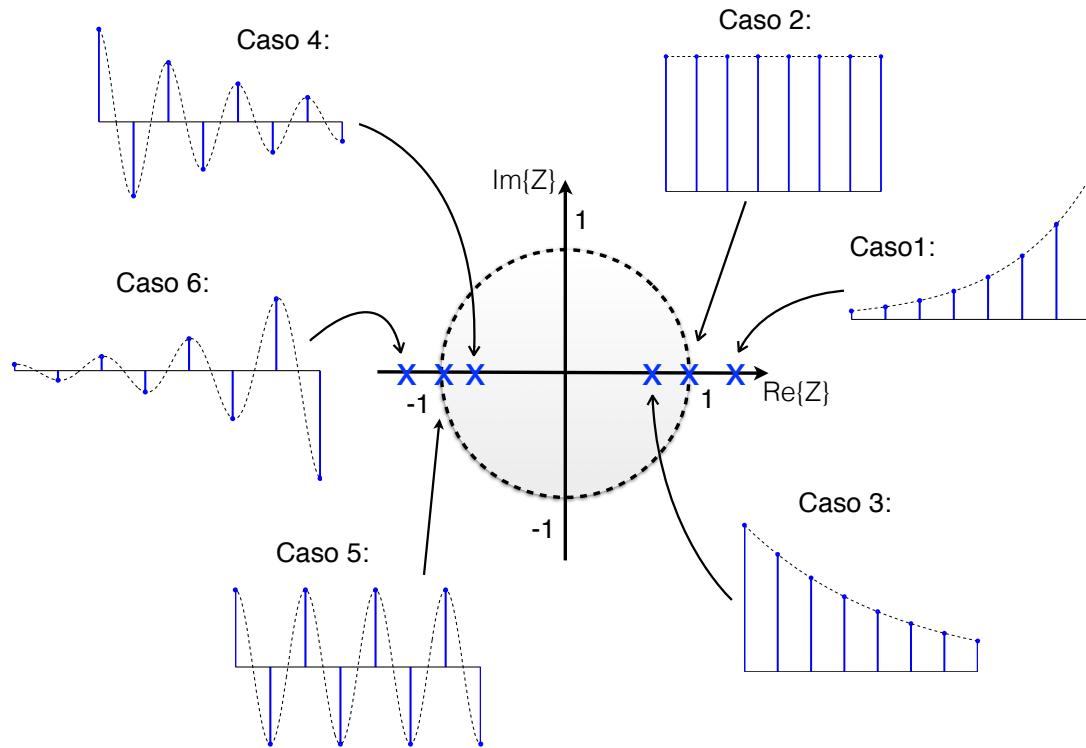


Figura 5.3: Respostas transientes para casos de simples pólos reais.

**Caso 2 :**  $p_i = 1$ , sistema em ressonância, o pôlo se encontra sobre  $z = 1$ . Este tipo de sistema corre o risco de não ser estável para certas entradas.

**Caso 3 :**  $0 < p_i < 1$ , saída convergente (decrescente), apresentando valores decrescentes tão rapidamente quanto mais próximo de  $z = 0$  estiver este pôlo.

**Caso 4 :**  $-1 < p_i < 0$ , saída converge (decrece com o tempo), mas como  $p_i$  é negativo, forma-se uma sequencia convergente alternada.

**Caso 5 :**  $p = -1$ , sistema oscilando com amplitude constante (enquanto  $|p_i| = 1$ ) na frequencia de  $f = 1/(2T)$ .

**Caso 6 :**  $p_i < -1$ , outro pôlo instável, sistema apresenta saída divergente (crescente) e alternada (sinal negativo para  $p_i$ ).

### 5.3.2 Caso 2) F(z) com Pólos Complexos

Neste caso,  $F(z)$  apresenta pôlos conjugados complexos:  $c_j p_j^k + \bar{c}_j \bar{p}_j^k$ , onde:

$$c_j = a + jb$$

$$p_j = \alpha + j\beta$$

Representando o pôlo na forma polar, resulta em:

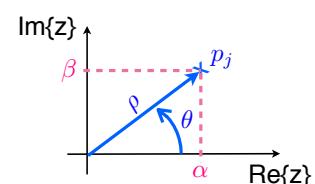
$$p_j = \rho e^{j\theta}$$

onde:

$$c_j = A e^{j\varphi}$$

$$\rho = \sqrt{\alpha^2 + \beta^2}$$

$$\theta = \tan^{-1} \left( \frac{\beta}{\alpha} \right)$$



então:

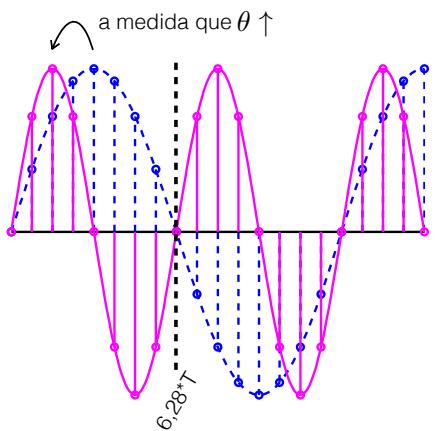
$$\begin{aligned} c_j p_j^k + \bar{c}_j \bar{p}_j^k &= A \rho^k e^{j(k\theta+\varphi)} + A \rho^k e^{-j(k\theta+\varphi)} \\ &= A \rho^k [e^{j(k\theta+\varphi)} + e^{-j(k\theta+\varphi)}] \\ &= 2A \rho^k \cos(k\theta + \varphi) \end{aligned} \quad (5.4)$$

ou seja, a saída se transforma numa exponencial sobreposta com uma senoide amostrada nos instantes k.

Supondo que  $k\theta = 2\pi$  (múltiplos cossenos), temos:  $t = kT \quad \therefore \quad k = \frac{t_{pp}}{T}$ .

Isolando em  $t_{pp}$  e substituindo em  $k\theta = 2\pi$ , obtemos  $t_{pp} = \frac{2\pi T}{\theta}$ , onde  $t_{pp}$  significa um pseudo-período.

A figura 5.4 mostra o efeito amostrado da eq. (5.5).

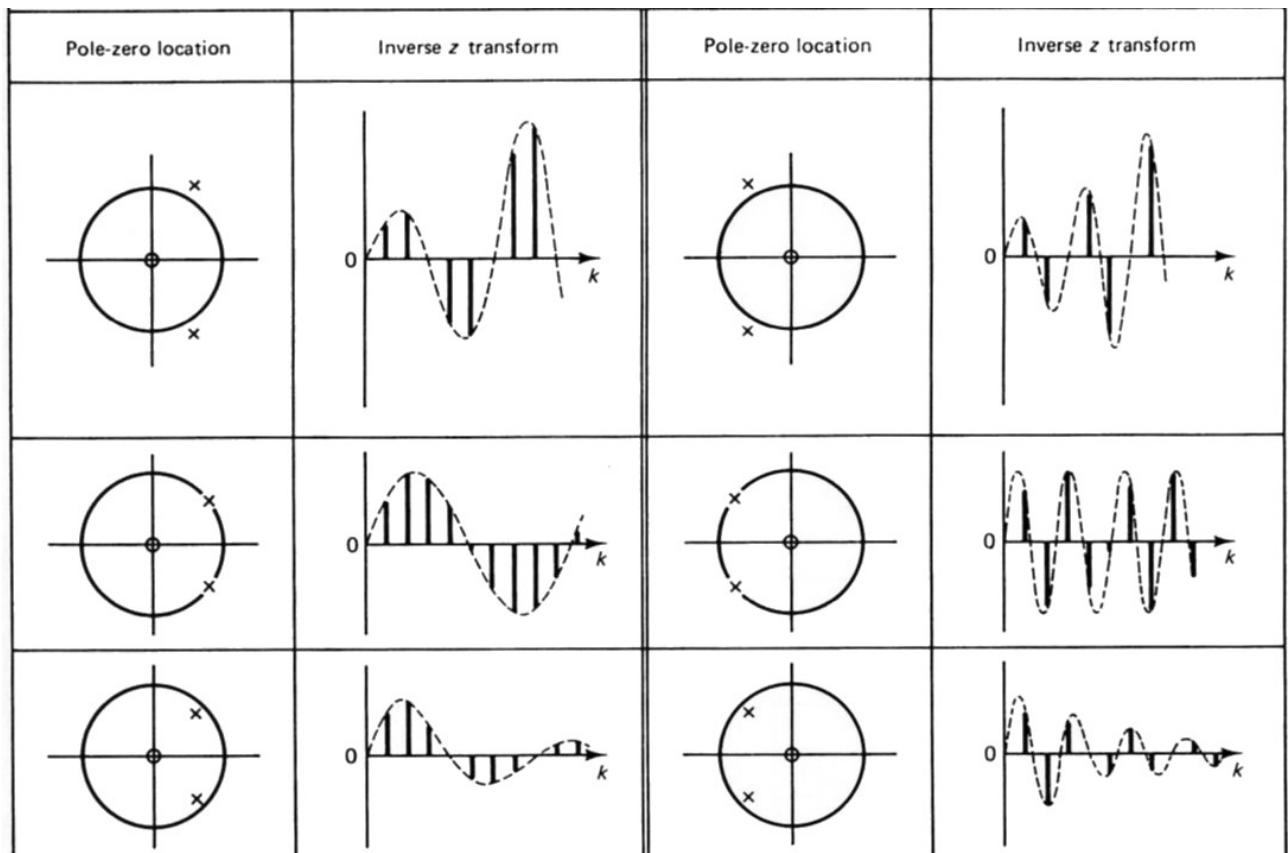


Script adotado no MATLAB para criar figura ao lado:

```
>> fplot(@(k) 2*4*5*cos(k*(pi/8)+3*pi/2), [0 20])
>> hold on
>> fplot(@(k) 2*4*5*cos(k*(pi/4)+3*pi/2), [0 20])
>> k=0:20;
>> y1=2*4*5*cos(k*(pi/8)+3*pi/2);
>> y2=2*4*5*cos(k*(pi/4)+3*pi/2);
>> stem(k,y1,'bo')
>> stem(k,y2,'mo')
```

Figura 5.4: Impacto de  $\theta$  sobre a eq. (5.5).

A figura 5.5 mostra diferentes casos de pólos complexos no plano-z e respectivas respostas transientes no tempo discreto.



$$\frac{z[z - e^{\alpha T} \cos(\omega T)]}{z^2 - 2e^{-\alpha T}z \cos(\omega T) + e^{-2\alpha T}} = z \left\{ e^{-\alpha kT} \cos(\omega kT) \right\}$$

Figura 5.5: Representações gráficas da Transformada Inversa Z.[Cadzow (1973),Kuo (1992)]

## 5.4 Método de Jury

O método de Jury<sup>1</sup> permite determinar a estabilidade de um sistema discreto linear através da análise dos coeficientes da sua equação característica. Este método seria uma analogia ao critério de estabilidade de Routh-Hurwitz usado com sistemas contínuos no tempo.

Seja o polinômio  $EC(z)$  extraído da **equação característica** de um sistema discreto:

$$EC(z) = a_n z^n + a_{n-1} z^{n-1} + \dots + a_1 z + a_0 = 0$$

o sistema é estável se e somente se suas raízes estiverem contidas dentro do círculo unitário no plano  $\mathbb{Z}$ .

Isto implica em **3 condições preliminares**<sup>2</sup>

- (i)  $EC(z)|_{z=1} = EC(1) > 0$
- (ii)  $(-1)^n EC(-1) > 0$ , ou:  $EC(z)|_{z=-1} = EC(-1) \begin{cases} > 0 & \text{para } n \text{ par} \\ < 0 & \text{para } n \text{ ímpar} \end{cases}$
- (iii)  $|a_0| < a_n$

Uma vez atendidas as condições anteriores parte-se para o arranjo (ou Tabela) de Jury (análogo ao do método de Routh-Hurwitz):

Linha	$z^0$	$z^1$	$z^2$	...	$z^{n-1}$	$z^n$
1	$a_0$	$a_1$	$a_2$	...	$a_{n-1}$	$a_n$
2	$a_n$	$a_{n-1}$	$a_{n-2}$	...	$a_1$	$a_0$
3	$b_0$	$b_1$	$b_2$	...	$b_{n-1}$	
4	$b_{n-1}$	$b_{n-2}$	$b_{n-3}$	...	$b_0$	
:	:	:			:	
:	:	:			:	
$2n - 5$	$p_0$	$p_1$	$p_2$	$p_3$		
$2n - 4$	$p_3$	$p_2$	$p_1$	$p_0$		
$2n - 3$	$q_0$	$q_1$	$q_2$			

A tabela de Jury é construída da seguinte forma:

- A tabela consiste em  $(2n - 3)$  linhas. Para um sistema de segunda-ordem, a tabela possui apenas 1 linha.
- Primeira linha: seus elementos consistem dos coeficientes de  $D(z)$  arranjados em potências ascendentes em  $z$ .
- Segunda linha: seus elementos consistem nos coeficientes de  $D(z)$  arranjados em potências descendentes de  $z$ .

Os outros elementos dependem dos determinantes:

$$\begin{aligned} b_k &= \begin{vmatrix} a_0 & a_{n-k} \\ a_n & a_k \end{vmatrix}, \quad k = 0, 1, 2, \dots, n-1 \\ c_k &= \begin{vmatrix} b_0 & b_{n-1-k} \\ b_{n-1} & b_k \end{vmatrix}, \quad k = 0, 1, 2, \dots, n-2 \\ d_k &= \begin{vmatrix} c_0 & c_{n-2-k} \\ c_{n-2} & c_k \end{vmatrix}, \quad k = 0, 1, 2, \dots, n-3 \\ q_0 &= \begin{vmatrix} p_0 & p_3 \\ p_3 & p_0 \end{vmatrix}, \quad q_2 = \begin{vmatrix} p_0 & p_1 \\ p_3 & p_2 \end{vmatrix} \end{aligned}$$

Note que:

<sup>1</sup>Eliah Ibraham Jury (nascido em 23/05/1923) é um engenheiro americano nascido em Bagdad (Iraque) que concluiu seu Doutorado em Engenharia na Universidade Colúmbia de Nova York em 1953. Ele foi professor na Engenharia Elétrica da Universidade de Califórnia em Berkeley e na Universidade de Miami. Ele desenvolveu avanços na transformada Z, usados em controle digital e processamento digital de sinais.

<sup>2</sup>Atenção: Ogata (1995) referencia a equação característica como:  $EC(z) = a_0 z^n + a_1 z^{n-1} + \dots + a_{n-1} z + a_n - 0$  que faz inverter a primeira pré-condição (no caso dele) para:  $|a_n| < a_0$ .

- Os elementos de qualquer linha par (linhas 2, 4, ...) são simplesmente o reverso da linha imediatamente precedente ímpar (linhas 1, 3, ...).
- A última linha ( $2n - 3$ ) possui apenas 3 elementos.

Uma vez finalizado o arranjo, pode-se concluir que o sistema será estável se:

$$\left. \begin{array}{l} |a_0| < a_n \\ |b_0| > |b_{n-1}| \\ |c_0| > |c_{n-2}| \\ |d_0| > |d_{n-3}| \\ \vdots > \vdots \\ |p_0| > |p_3| \\ |q_0| > |q_2| \end{array} \right\} (n-1) \text{ restrições}$$

### 5.4.1 Exemplos

**Exemplo 1:** Sistema de 2a-ordem:

Num sistema de 2a-ordem,  $n = 2$  e neste caso, a tabela de Jury contém apenas 1 linha. Assim, para estabilidade:

$$\begin{aligned} EC(1) &> 0 \\ EC(-1) &> 0 \\ |a_0| &< a_2 \end{aligned}$$

Por exemplo, para o sistema: se  $EC(z) = z^2 - 0,25 = 0$ , as pré-condições necessárias para estabilidade são:

$$\begin{aligned} (1) \quad EC(1) &> 0 \quad \therefore EC(1) = (1)^2 - 0,25 = 1 - 0,25 = 0,75 \quad \therefore 0,75 > 0 \quad \checkmark \\ (2) \quad EC(-1) &> 0 \quad \therefore EC(-1) = (-1)^2 - 0,25 = 1 - 0,25 = 0,75 \quad \therefore 0,75 > 0 \quad \checkmark \\ (3) \quad |a_0| &< a_2 \quad \therefore |-0,25| < 1 \quad \checkmark \end{aligned}$$

Desta forma, percebe-se que todas as condições foram satisfeitas, ou seja, o sistema é estável.

De fato, as raízes para o polinômio  $EC(z)$  estão localizados nos pontos:  $p_1 = -0,5$  e  $p_2 = 0,5$ , resultado de:  $z = \pm\sqrt{(0,25)}$  ou seja:  $EC(z) = (z - 0,5)(z + 0,5)$ .

**Exemplo 2:** Exemplo de sistema de 3ª ordem:

Determinar a estabilidade de um sistema discreto cuja equação característica é:

$$EC(z) = z^3 - 1,2z^2 - 1,375z - 0,25 = 0$$

As pré-condições para estabilidade são:

$$\begin{aligned} (1) \quad EC(1) &> 0 \quad \therefore EC(1) = (1)^3 - 1,2(1)^2 - 1,375(1) - 0,25 \\ &= 1 - 1,2 - 1,375 - 0,25 \\ &= -1,825 \quad \therefore -1,825 \not> 0 \quad \times \\ (2) \quad EC(-1) &< 0 \quad \therefore EC(-1) = (-1)^3 - 1,2(-1)^2 - 1,375(-1) - 0,25 \\ &= -1 - 1,2 + 1,375 - 0,25 \\ &= -1,0750 \quad \therefore -1,0750 < 0 \quad \checkmark \\ (3) \quad |a_0| &< a_2 \quad \therefore |-0,25| < 1 \quad \checkmark \end{aligned}$$

Note que a condição  $EC(1) > 0$  não foi cumprida, o que significa que este sistema possui 1 pólo fora do círculo unitário, e portanto é instável.

As raízes de  $EC(z)$  podem ser calculadas usando-se alguma ferramenta numérica como o Matlab:

```
>> EC = [1 -1.2 -1.375 -0.25];
>> roots(EC)
ans =
1.9646
```

```
-0.5199
-0.2448
>>
```

E assim confirma-se que uma das raízes da  $EC(z)$  está fora do círculo unitário (a raiz em  $z = 1.9646$ ).

Note que podemos usar o próprio Matlab para determinar os valores numéricos para as condições de teste, usando a função '» `polyval( EC, <valor> )`', por exemplo:

```
>> EC=[1 -1.2 -1.375 -0.25]; %ingressando polinomio
>> polyval(EC, 1)
ans =
-1.8250
>> polyval(EC, -1)
ans =
-1.0750
>>
```

**Exemplo 3:** Seja outra equação característica:

$$EC(z) = z^3 + 3,3z^2 + 4z + 0,8 = 0$$

As pré-condições para estabilidade resultam em:

(1) $EC(1) > 0$	$EC(1) = (1)^3 + 3,3(1)^2 + 4(1) + 0,8$	
	$EC(1) = 3 + 3,3 + 4 + 0,8$	
	$EC(1) = 9,1$	$9,1 > 0 \quad \checkmark$
(2) $EC(-1) < 0$	$EC(-1) = (-1)^3 + 3,3(-1)^2 + 4(-1) + 0,8$	
	$EC(-1) = -1 + 3,3 - 4 + 0,8$	
	$EC(-1) = -0,9$	$-0,9 < 0 \quad \checkmark$
(3) $ a_0  < a_3$	$ 0,8  < 1$	$\checkmark$

Como as 3 pré-condições foram atendidas, se faz necessário montar o Arranjo de Jury:

Linha	$z^0$	$z^1$	$z^2$	$z^3$
1	0,8	4,0	3,3	1,0
2	1,0	3,3	4,0	0,8
3	$b_0$	$b_1$	$b_2$	

onde:

$$\begin{aligned} b_0 &= \begin{vmatrix} a_0 & a_3 \\ a_3 & a_0 \end{vmatrix} = a_0^2 - a_3^2 = -0,36 \\ b_1 &= \begin{vmatrix} a_0 & a_2 \\ a_3 & a_1 \end{vmatrix} = a_0a_1 - a_2a_3 = -0,1 \\ b_2 &= \begin{vmatrix} a_0 & a_1 \\ a_3 & a_2 \end{vmatrix} = a_0a_2 - a_1a_3 = -1,36 \end{aligned}$$

e assim:

Linha	$z^0$	$z^1$	$z^2$	$z^3$
1	0,8	4,0	3,3	1,0
2	1,0	3,3	4,0	0,8
3	-0,36	-0,1	-1,36	

a condição para estabilidade é:  $|b_0| > |b_2|$ , neste caso, a condição  $|-0,36| > |-1,36|$  não é cumprida, e isto significa que neste caso, este sistema em malha-fechada apresenta pólos de MF fora do círculo unitário (instáveis).

De fato, as raízes de  $EC(z)$  estão em:

```
>> EC=[1 3.3 4 0.8]
EC =
    1.0000    3.3000    4.0000    0.8000
>> roots(EC)
ans =
-1.5268 + 0.9574i
-1.5268 - 0.9574i
-0.2463
>>
```

percebemos que as 2 raízes complexas estão fora do círculo unitário – ver figura 5.6.

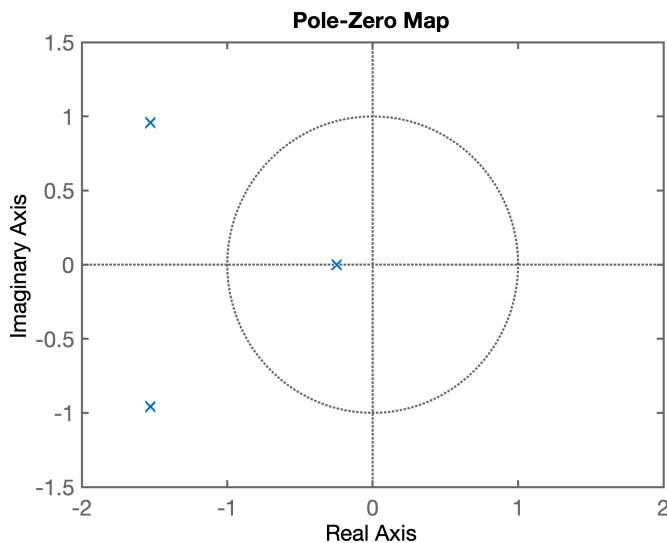


Figura 5.6: Pólos em MF da  $EC(z)$  do exemplo 3, mostrando 2 pólos complexos instáveis (fora do círculo unitário).

## 5.5 Aplicação do Método de Jury (Determinação de $K_{\max}$ )

### Exemplo de Planta de Segunda ordem:

Analice a estabilidade do seguinte sistema em malha-fechada (fig 5.7) [Phillips and Nagle (1994); pág. 191, 231]:

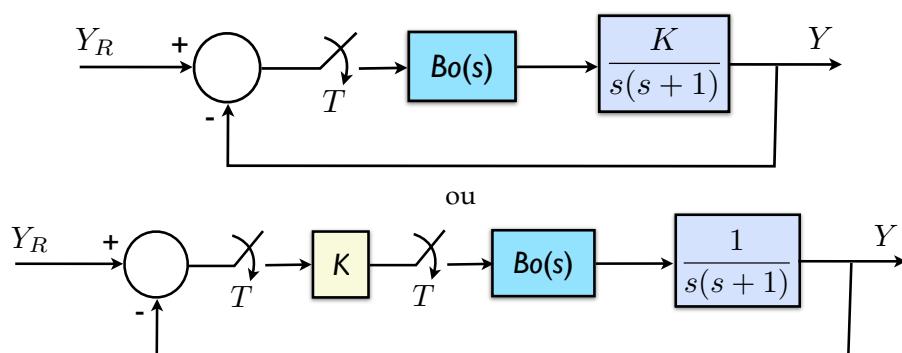


Figura 5.7: Sistema exemplo.

Sua função transferência é dada por:

$$\frac{Y(z)}{Y_R(z)} = \frac{BoG(z)}{1 + BoG(z)}$$

Calculando BoG(z):

$$\begin{aligned} \mathcal{Z}\{Bo(s)G(s)\} &= (1-z^{-1})\mathcal{Z}\left\{\frac{1}{s^2(s+1)}\right\} \\ &= (1-z^{-1})\mathcal{Z}\left\{\frac{1}{1s^2} - \frac{1}{1s} + \frac{1}{1(s+1)}\right\} \\ &= \frac{(z-1)}{z} \left[ \frac{Tz}{(z-1)^2} - \frac{z}{(z-1)} + \frac{z}{(z-e^{-T})} \right] \\ BoG(z) &= \frac{K[z(T+e^{-T}-1)+(1-Te^{-T}-e^{-T})]}{(z-1)(z-e^{-T})} \end{aligned}$$

que pode ser organizado neste outro formato:

$$BoG(z) = \frac{K(b_1 z + b_0)}{(z-1)(z-a)} \quad \text{ou:} \quad BoG(z) = \frac{K b_1 \left( \frac{b_1}{b_1} z + \frac{b_0}{b_1} \right)}{(z-1)(z-a)}$$

onde:  $a = e^{-T}$  (corresponde ao polo do sistema de 1a-ordem em  $s = -1$  mapeado no plano-z);

$$b_1 = T + e^{-T} - 1;$$

$$b_0 = 1 - Te^{-T} - e^{-T}; e,$$

T corresponde ao período de amostragem adotado.

A equação característica deste sistema fica:

$$\begin{aligned} EC(z) &= 1 + BoG(z) \\ &= (z-1)(z-a) + K(b_0 + b_1 z) \\ &= z^2 - az - z + a + Kb_0 + Kb_1 z \\ EC(z) &= z^2 + (Kb_1 - a - 1)z + (Kb_0 + a) \end{aligned}$$

Usando-se o método de Jury para identificar os requisitos para estabilidade deste sistema temos:

$$\begin{aligned} EC(z) &= z^2 + (Kb_1 - a - 1)z + (Kb_0 + a) \\ &= a_2 \quad a_1 \quad a_0 \end{aligned}$$

Aplicando os critérios (iniciais) de Jury:

$$\begin{aligned} (1) \quad EC(1) &> 0 \quad \rightarrow \quad \begin{aligned} EC(1) &= (1)^2 + (Kb_1 - a - 1)(1) + (Kb_0 + a) \\ &= 1 + Kb_1 - a - 1 + Kb_0 + a \\ &= Kb_1 + Kb_0 \\ &= K(b_1 + b_0) \end{aligned} \\ \text{Então:} \quad &K > 0 \end{aligned}$$

**Válido apenas se:**  $(b_1 + b_0) > 0$

$$\begin{aligned} (2) \quad EC(-1) &> 0 \quad \rightarrow \quad \begin{aligned} EC(-1) &= (-1)^2 + (Kb_1 - a - 1)(-1) + (Kb_0 + a) \\ &= 1 - Kb_1 + a + 1 + Kb_0 + a \\ &= 2 + 2a \\ &\text{o que leva à:} \end{aligned} \\ &K(b_0 - b_1) > -2a - 2 \end{aligned}$$

$$K > \frac{-2(1+a)}{b_0 - b_1}$$

$$K < \frac{2(1+a)}{b_0 - b_1}$$

**Válido apenas se:**  $(b_0 - b_1) > 0$

e finalmente:

$$|Kb_0 + a| < 1$$

de onde sai que:

$$(3) \quad |a_0| < a_2 \quad \rightarrow \quad K < \frac{1-a}{b_0}$$

**Válido apenas de:**  $b_0 > 0$

## 5.6 Efeitos do Período de Amostragem

A idéia agora é verificar se as faixas de ganho variam quando varia o período de amostragem num sistema.

Verificando-se para o caso do sistema anterior, os efeitos causados para 2 diferentes valores para o período de amostragem:

$$T = \begin{cases} 0,1 \text{ segundos, caso: (a)} \\ 1,0 \text{ segundo, caso: (b)} \end{cases}$$

a) Quando  $T = 0,1$  (segundos) teremos:

$$\begin{aligned} a &= e^{-T} &= 0,904837418035960 \\ b_1 &= e^{-T} + T - 1 &= 0,004837418035960 \\ b_0 &= 1 - e^{-T} - Te^{-T} &= 0,004678840160445 \end{aligned}$$

Aplicando-se os critérios de estabilidade, ocorrerá:

$$(1) \quad K > 0 \quad \boxed{\text{Se: } b_1 + b_0 > 0} \quad \text{Neste caso:} \quad 0,0048374 + 0,0046788 = 0,0095 > 0 \quad \text{Ok ✓}$$

$$(2) \quad K < \frac{2(1+a)}{b_0 - b_1}, \quad \boxed{\text{Se: } b_0 - b_1 > 0} \quad 0,004678840160445 - 0,004837418035960 > 0 \quad -1,585778755150841 \times 10^{-4} \not\geq 0 \quad \text{Ops... ✗} \\ \text{Nem chegamos a calcular o valor do ganho} \\ (\text{condição inicial não atendida})$$

$$(3) \quad K < \frac{1-a}{b_0} \quad \boxed{\text{Se: } b_0 > 0} \quad 0,0046788 > 0 \quad \text{Ok ✓} \\ \text{Então:} \quad K < \frac{1 - 0,9048}{0,004678840160445} \\ \boxed{K < 20,338925609931366} \quad \text{Ok ✓}$$

então, a condição para estabilidade quando  $T = 0,1$  se resume à:

$$0 < K < 20,3389$$

b) Quando que  $T = 1,0$  (segundos) teremos:

$$\begin{aligned} a &= e^{-T} &= 0,367879441171442 \\ b_1 &= e^{-T} + T - 1 &= 0,367879441171442 \\ b_0 &= 1 - e^{-T} - Te^{-T} &= 0,264241117657115 \end{aligned}$$

Aplicando-se os critérios de estabilidade, ocorrerá:

$$(1) \quad K > 0$$

**Se:**  $b_1 + b_0 > 0$

Neste caso:

$$0,3679 + 0,2642 = 0,6321 > 0$$

Ok ✓

$$(2) \quad K < \frac{2(1+a)}{b_0 - b_1}, \quad \text{Se: } b_0 - b_1 > 0$$

$$0,2642 - 0,3679 > 0$$

$$-0,1036 > 0$$

Ops... ✗

Nem chegamos a calcular o valor do ganho  
(condição inicial também não atendida)

$$(3) \quad K < \frac{1-a}{b_0}$$

**Se:**  $b_0 > 0$

$$0,2642 > 0$$

Ok ✓

**Então:**

$$K < \frac{1 - 0,3679}{0,2642}$$

$$K < 2,392211191177333$$

Ok ✓

então, a condição para estabilidade se resume à:

$$0 < K < 2,3922$$

Perceba então que quando o período de amostragem aumenta, o valor para o ganho máximo diminui, uma relação inversamente proporcional.

## Conclusão

Nota-se que a medida que aumenta o período de amostragem  $T$ , se é obrigado a abaixar o ganho  $K$  do controlador, para manter a estabilidade do sistema.

$$\uparrow T \Rightarrow \downarrow K$$

## Comportamento de um sistema na situação do ganho máximo

Vamos verificar o que acontece quando  $T = 1,0$  (segundos) e  $K = 2,3922$ :

Para este valor de  $K$  a equação do sistema fica:

$$BoG(z) = \frac{K(b_0 + b_1 z)}{(z-1)(z-a)} = \frac{2,3922(0,3679z + 0,2642)}{z^2 - 1,3679z + 0,3679}$$

assim, a função transferência em malha-fechada fica:

$$FTMF(z) = \frac{BoG(z)}{1 + BoG(z)}$$

e sua equação característica fica:

$$\begin{aligned} EC(z) &= 1 + BoG(z) = 0 \\ &= 1 + \frac{2,3922(0,3679z + 0,2642)}{z^2 - 1,3679z + 0,3679} = 0 \\ &= z^2 - 0,4878z + 1 = 0 \end{aligned}$$

## 5.6. Efeitos do Período de Amostragem

## Samplingático III

ou seja, as raízes para  $EC(z)$  (pólos do sistema em MF) ficam em:

$$\begin{aligned} z_1 &= 0,2439 + j0,9698 \\ z_2 &= 0,2439 - j0,9698 \end{aligned}$$

que equivale à  $z = 1,0 \angle 1,3244(\text{rad}) = 1,0 \angle 75,88^\circ$  ou como:  $z = 1,0 \angle \pm \omega T$ , significa que este sistema oscilará na frequência de  $\omega \cdot T = 1,3244 \therefore \omega = 1,3244/T = 1,3244/1 = 1,3244$  (rad/s) (ou  $f = 2\pi/\omega = 2 \cdot 3,1416/1,3244 = 0,21078$  (Hz), com período de oscilação de:  $T_{\text{Senoidal}} = 1/f = 1/0,21078 = 4,7443$  (segundos)).

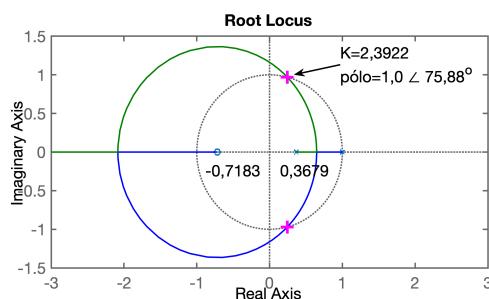
De fato, se:

$$BoG(z) = \frac{(b_1 z + b_0)}{(z - 1)(z - a)} = \frac{0,36788(z + 0,7183)}{(z - 1)(z - 0,3679)}$$

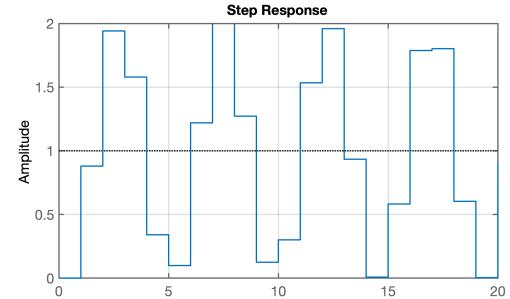
fechando a malha de realimentação com  $K = 2,3922$ , obteremos:

$$\begin{aligned} FTMF(z) &= \frac{BoG(z)}{1 + BoG(z)} \\ &= \frac{0,88005(z + 0,7183)}{(z^2 - 0,4878z + 1)} \end{aligned}$$

onde confirmamos pólos de MF complexos com módulo = 1 (no limiar da instabilidade). Um diagrama no lugar das raízes para este sistema resulta na figura 5.8.



(a) Root Locus para sistema  $BoG(z)$ .



(b) Resposta ao degrau para  $K = 2,3922$ .

Figura 5.8: RL e resposta ao degrau para  $BoG(z)$  com  $K = 2,3922$ .

Usando o MATLAB para realizar os cálculos anteriores:

```
>> T=1.0;
>> a=exp(-T)
a = 0.3679
>> b1=exp(-T)+T-1
b1 = 0.3679
>> bo=1-exp(-T)-T*exp(-T)
bo = 0.2642
>> K=(1-a)/bo
K = 2.3922
>> %Montando BoG(z) SEM incluir o ganho K
>> BoG=tf([b1 bo], poly([1 a]),T);
>> zpk(BoG)

Zero/pole/gain:
0.36788 (z+0.7183)

(z-1) (z-0.3679)

Sampling time: 1
>> %Fechando a malha para obter EC(z) (agora sim, acrescentando o ganho):
>> ftmf=feedback(K*BoG,1)

Transfer function:
```

```

0.88 z + 0.6321
-----
z^2 - 0.4878 z + 1
Sampling time: 1>> zpk(ftmf)

0.88005 (z+0.7183)
-----
(z^2 - 0.4878z + 1)

Sample time: 1 seconds
>> %Recuperando EC(z):
>> [num_mf,den_mf,T]=tfdata(ftmf,'v')
num_mf =           0    0.8800    0.6321
den_mf =      1.0000   -0.4878    1.0000
T =             1
>> %Encontrando as raízes de EC(z):
>> p_mf=roots(den_mf)
p_mf =
  0.2439 + 0.9698 i
  0.2439 - 0.9698 i
>> %ou poderíamos ter realizado:
>> polos_MF=pole(ftmf)
polos_MF =
  0.2439 + 0.9698 i
  0.2439 - 0.9698 i
>> %Determinando polo em coordenadas polares:
>> abs(pолос_MF(1))
ans =
  1.0000
>> w=angle(pолос_MF(1))
w =
  1.3244
>> %calculando angulo em graus:
>> angulo=w*180/pi
angulo =
  75.8822
>> %prevendo frequencia de oscilação:
>> w_osc=w/T
w_osc =
  1.3244
>> freq=w_osc/(2*pi)
freq =
  0.2108
>> %correspondendo ao período de oscilação de:
>> T_osc=1/freq
T_osc =
  4.7442
>> %Plotando o RL + pólos de MF:
>> rlocus(BoG)
>> hold on; %para manter figura anterior e permitir sobreposição de gráficos
>> plot(pолос_MF,'m+', 'MarkerSize',12, 'LineWidth',3)
>> %e verificando a resposta ao degrau:
>> figure; step(ftmf)
>> axis([0 20 0 2])

```

## 5.7 Diagrama do Lugar das Raízes (RL)

- O efeito do Ganho ( $K$ ) num sistema e/ou do período de amostragem na estabilidade absoluta ou relativa do sistema em malha fechada deve ser investigada adicionalmente às características da resposta transitória. O método do Lugar das Raízes (RL) é muito útil neste sentido.
- O método do Lugar das Raízes (Root Locus (RL)) para sistemas contínuos no tempo pode ser estendido

para os sistemas de tempo discreto sem muitas modificações uma vez que a equação característica do sistema de controle discreto segue o mesmo formato que no sistema de controle contínuo no tempo.

Em muitos sistemas LTI (Lineares e Invariantes no Tempo), a equação característica pode seguir um dos seguintes formatos:

$$\begin{aligned} 1 + G(z)H(z) &= 0 \\ 1 + GH(z) &= 0 \end{aligned}$$

combinando ambas as equações anteriores, podemos definir a equação característica como sendo:

$$1 + L(z) = 0 \quad (5.5)$$

onde  $L(z) = G(z)H(z)$  ou  $L(z) = GH(z)$ .  $L(z)$  é popularmente conhecida como a função transferência do impulso para o laço aberto. A equação 5.5 pode ser escrita como:

$$L(z) = -1$$

uma vez que  $L(z)$  expressa uma quantidade complexa ela pode ser dividida em 2 equações, uma equação relacionada com os ângulos e outra com a magnitude. Isto resulta em 2 critérios à serem seguidos:

- Critério Angular:  $\angle L(z) = \pm 180^\circ(2k + 1)$ ,  $k = 0, 1, 2, \dots$
- Critério da Magnitude:  $|L(z)| = 1$

Os valores de  $z$  que satisfazem ambos os critérios correspondem às raízes da equação característica ou aos **polos de malha-fechada**. Antes de construir o diagrama do Lugar das Raízes, a equação  $1 + L(z) = 0$  deve ser reorganizada na seguinte forma:

$$1 + K \frac{(z+z_1)(z+z_2)\cdots(z+z_m)}{(z+p_1)(z+p_2)\cdots(z+p_n)} = 0 \quad (5.6)$$

onde  $z_i$ 's e  $p_i$ 's correspondem respectivamente aos zeros e polos da função transferência em malha aberta ( $FTMA(z)$ ),  $m$  é o número de zeros e  $n$  é o número de pólos.

### 5.7.1 Construindo o Diagrama do Lugar das Raízes

As regras para construção do RL para sistemas digitais são as mesmas que as usadas para sistemas no tempo contínuo:

1. O diagrama de RL é simétrico com relação ao eixo real. O número de traçados do RL corresponde ao número de polos de malha aberta.
2. Os traçados do RL partem dos polos de malha aberta com  $K = 0$  e terminam nos zeros de malha aberta, com  $K = \infty$ . Na ausência de zeros de malha aberta, o RL tende para  $-\infty$  quando  $K \rightarrow \infty$ . O número de traçados que tendem para  $-\infty$  corresponde a diferença entre o número de polos e de zeros.
3. Uma região do eixo real fará parte do RL se o número de polos mais o número de zeros à direita desta parte for ímpar.
4. Se existem  $n$  polos de malha aberta e  $m$  zeros de malha aberta, então  $n - m$  traçados do RL vão tender para  $-\infty$  ao longo de linhas **retas assintóticas** desenhadas à partir de um único ponto  $s = \sigma$  que é chamado de centro do loci:

$$\sigma = \frac{\sum (\text{Partes reais dos pólos de malha aberta}) - \sum (\text{Partes reais dos zeros de malha aberta})}{n - m} \quad (5.7)$$

O ângulo de partida das assintotas é dado por:

$$\phi = \frac{180^\circ(2q + 1)}{n - m}, \quad q = 0, 1, \dots, n - m - 1 \quad (5.8)$$

5. As raízes da equação abaixo definem os pontos de partida (*breakaway*) e chegada (*break in*) dos traçados de RL:

$$\frac{dK}{dz} = 0 \quad (5.9)$$

onde  $K$  é expresso em função de  $z$  da equação característica do sistema. Esta é uma condição necessária porém, não é suficiente. Devemos verificar se as soluções encontradas recaem sobre o RL sendo traçado.

6. A interseção (se existe) do RL com o **círculo unitário** pode ser determinada à partir do arranjo de Routh-Hurwitz.

7. O ângulo de partida para **pólos complexos** de malha aberta é dado por:

$$\phi_p = 180^\circ + \phi \quad (5.10)$$

onde  $\phi$  corresponde à contribuição final de todos os ângulos dos pólos e zeros de malha aberta para este polo:

$$\phi = \sum_i \Psi_i - \sum_{j \neq p} \gamma_j \quad (5.11)$$

onde  $\Psi_i$ 's correspondem às contribuições dos ângulos gerados pelos zeros e  $\gamma_j$ 's correspondem às contribuições dos ângulos gerados pelos polos.

8. O ângulo de chegada relacionado com o **zero complexo** é dado por:

$$\phi_z = 180^\circ - \phi \quad (5.12)$$

onde  $\phi$  é o mesmo valor determinado pela regra anterior.

9. O ganho no ponto  $z_o$  sobre o RL é determinando através de:

$$K = \frac{\prod_{j=1}^n |z_o + p_j|}{\prod_{i=1}^m |z_o + z_i|} \quad (5.13)$$



## 5.7.2 Exemplos de Traçados de RL

### Exemplo\_1

Para o caso do exemplo anterior, reproduzido novamente na figura 5.9 (ver seção 5.5, pág. 27).

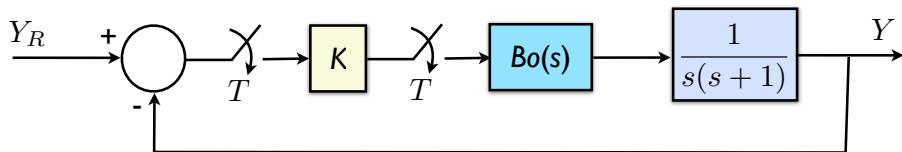


Figura 5.9: Sistema exemplo de 2a-ordem.

Com  $T = 1,0$  (segundos), teremos:

$$BoG(z) = \frac{K(b_0 + b_1 z)}{(z-1)(z-a)} = \frac{0,36788K(z+0,7183)}{(z-1)(z-0,3679)} = \frac{K(0,3679z+0,2642)}{z^2 - 1,3679z + 0,3679}$$

O lugar das raízes se origina em  $z = 1$  e  $z = 0,3679$  e termina em  $z = -0,7183$  e  $z = -\infty$ .

O número de assíntotas equivale à:  $n_p - n_z$ , então  $(2 - 1) = 1$ , temos 1 assíntota. O ângulo de partida da assíntota é dado por:

$$\text{Ângulo da assíntota} = \alpha = \frac{(2N + 1) \widehat{\pi}}{(n_p - n_z)}, \quad N = 0, 1, 2, \dots$$

onde:  $N = 0$  corresponde a assíntota com o menor ângulo com respeito ao eixo real. Embora  $N$  assuma um número infinito de valores, o ângulo repete a si mesmo a medida que  $N$  aumenta. O número de diferentes assíntotas é dado por:  $n_p - n_z$ .

Neste caso:

$$\begin{aligned} \alpha &= \frac{(2N + 1) \cdot 180}{n_p - n_z} = \frac{180^\circ}{1} = 180^\circ, \quad p/N=0 \\ &= \frac{3 \cdot 180^\circ}{1} = 540^\circ, \quad p/N=1 \end{aligned}$$

Os **pontos de partida** (*break-way*) e **pontos de chegada** (*break-in*) podem ser determinados à partir das raízes de:

$$\frac{dK}{dz} = \frac{D'(z)N(z) - D(z)N'(z)}{N^2(z)} = 0 \quad (5.14)$$

onde: o apóstrofe indica diferenciação em relação à  $z$ ,  $D(z)$  corresponde ao denominador da FTMA( $z$ ) e  $N(z)$  é o numerador da FTMA( $z$ ). Lembrando que:

$$\frac{du}{dv} = \frac{vdu - udv}{v^2}$$

e que a equação característica  $EC(z) = 0$  pode ser escrita como:

$$\begin{aligned} EC(z) &= 1 + FTMA(z) = 0 \\ &= 1 + \frac{K \cdot N(z)}{D(z)} \end{aligned}$$

então:

$$K = -\frac{D(z)}{N(z)}$$

Para o caso do nosso sistema, os pontos de partida/chegada são dados por:

$$\frac{d}{dz} [BoG(z)] = 0$$

calculando a derivada:

$$\begin{aligned} \frac{d}{dz} [BoG(z)] &= \frac{d}{dz} \left[ \frac{0,36788K(z+0,7183)}{(z-1)(z-0,3679)} \right] \\ &= \frac{(z^2 - 1,3368z + 0,368)(0,368K) - (0,368Kz + 0,264K)(2z - 1,368)}{(z^2 - 1,368z + 0,3679)^2} = 0 \end{aligned}$$

trabalhando esta equação, temos:

$$\begin{aligned} 0,368Kz^2 - 0,503424Kz + 0,135424K - 0,736Kz^2 + 0,503424Kz - 0,528Kz + 0,361152K &= 0 \\ 0,368Kz^2 - 0,503424Kz + 0,135424K - 0,736Kz^2 - 0,024576Kz + 0,461152K &= 0 \\ -0,368Kz^2 - 0,538Kz + 0,496576K &= 0 \end{aligned}$$

descobrindo as raízes para a equação anterior obtemos:

$$\begin{aligned} \Delta &= b^2 - 4ac & \therefore & \Delta = 0,278784 - 4(-0,368)(0,496576) = 1,0097439 \\ z &= \frac{-b \pm \sqrt{\Delta}}{2a} & \therefore & z = \frac{0,528 \pm 1,0048601}{-(2 \cdot 0,368)} \\ & & & z_1 = -2,0826904 \\ & & & z_1 = +0,6479077 \end{aligned}$$

e nestes pontos  $K$  vale:

- $p/z = z_1 = -2,0826904$ ,  $BoG(z) = 1$ , então:

$$\begin{aligned} & \left| \frac{0,368K(z+0,717)}{(z^2 - 1,368z + 0,368)} \right|_{z=-2,0826904} = 1 \\ & \left| \frac{0,368K(-2,08 + 0,717)}{(-2,08)^2 - 1,368(-2,08) + 0,368} \right| = 1 \\ & \left| \frac{-0,501584K}{7,53984} \right| = 1 \\ K &= \frac{7,53984}{0,501584} \\ K &= 15,03 \end{aligned}$$

- $p/z = z_1 = 0,6479077$ ,  $BoG(z) = 1$ , então:

$$\begin{aligned} & \left| \frac{0,368K(z+0,717)}{(z^2 - 1,368z + 0,368)} \right|_{z=0,6479077} = 1 \\ & \left| \frac{0,368K(0,65 + 0,717)}{(0,65)^2 - 1,368(0,65) + 0,368} \right| = 1 \\ & \left| \frac{-0,503056K}{-0,0987} \right| = 1 \\ K &= \frac{0,0987}{0,503056} \\ K &= 0,196 \end{aligned}$$

Usando o MATLAB:

```
>> T=1.0;
>> % Entrando com dados do sistema
>> a=exp(-T);
>> b1=exp(-T)+T-1;;
>> bo=1-exp(-T)-T*exp(-T)
>> % Montando BoG(z):
>> num=[b1 bo];
>> den=conv([1 -1],[1 -a]);
>> BoG=tf(num,den,T)

Transfer function:
 0.3679  z +  0.2642
-----
z^2 - 1.368  z + 0.3679

Sampling time: 1
>> zpk(BoG)

Zero/pole/gain:
0.36788  (z+0.7183)
-----
(z-1)  (z-0.3679)

Sampling time: 1

>> rlocus(BoG)
>> axis equal
>> % Verificando a resposta do sistema para entrada degrau
>> % usando o ganho máximo K=2.3922:
>> K=(1-a)/bo
K = 2.3922
>> FTMF=feedback(K*BoG,1)

Transfer function:
 0.88  z +  0.6321
-----
z^2 - 0.4878  z + 1

Sampling time: 1
```

```
>> zpk(FIMF)
Zero/pole/gain: 0.88005 (z+0.7183)
_____
(z^2 - 0.4878z + 1)

Sampling time: 1
>> figure;
>> step(FIMF)
```

O código anterior permite obter as figuras 5.10(a) e (b).

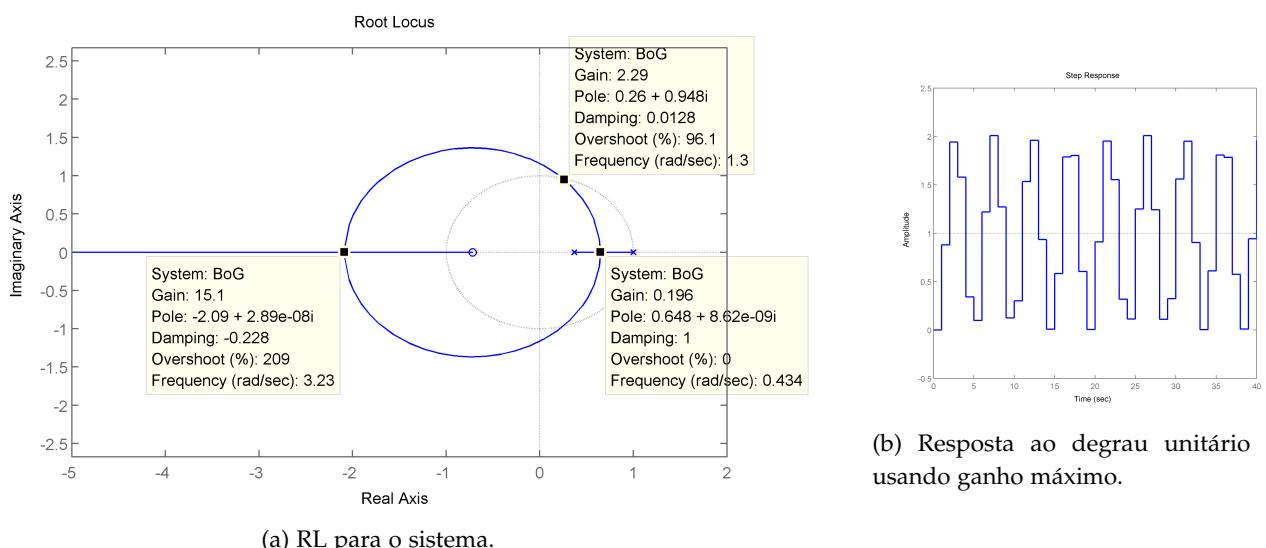


Figura 5.10: RL e resposta ao degrau unitário usando ganho máximo para sistema usado como exemplo.

### Ajustando o sistema anterior para outros valores de ganho

Se fosse desejada uma resposta com sobrepasso máximo de 20% poderíamos ter ajustado o ganho no correspondente valor para  $\zeta$ :

Lembrando que:

$$\zeta = \frac{-\ln \%OS/100}{\sqrt{\pi^2 + \ln^2 (\%OS/100)}} \\ = 0,4559$$

este valor de  $\zeta$  pode ser “impresso” no diagrama do Lugar das Raízes deste sistema para servir como guia para definição da posição desejada para os pólos de malha-fechada do sistema e consequente valor K do ganho necessário – no Matlab:

```
>> zeta=(-log(20/100))/(sqrt(pi^2+log(20/100)^2))
zeta =
0.4559
>> figure; % abre nova janela grafica
>> rlocus(BoG);
>> hold on
>> zgrid(zeta,o)
>> axis equal
>> rlocfind(BoG)
Select a point in the graphics window

selected_point =
0.5787 + 0.4183i
```

```

ans =
0.5547
>> % Encontramos o ganho como sendo:>> k=0.5547;
>> % Fechamos a malha com este ganho e verificamos resposta do sistema
>> novo_mf=feedback(k*BoG,1)

Transfer function:
0.2041 z + 0.1466
-----
z^2 - 1.164 z + 0.5145

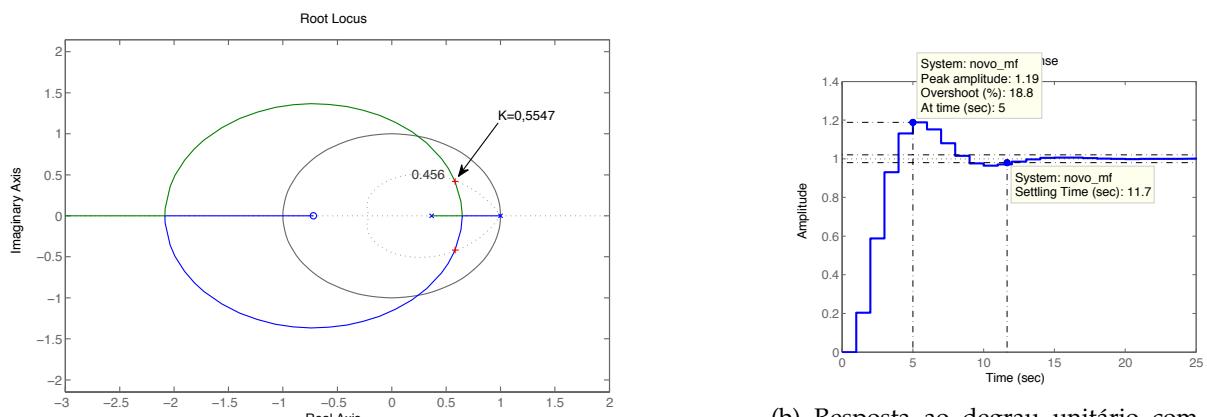
Sampling time: 1
>> zpk(novo_mf)

Zero/pole/gain:
0.20406 (z+0.7183)
-----
(z^2 - 1.164z + 0.5145)

Sampling time: 1
>> % Verificando resposta para entrada degrau:
>> step(novo_mf)

```

A figura 5.11(a) mostra o diagrama do Lugar das Raízes com a linha guia para o  $\zeta = 0,4559$  selecionado e a figura 5.11(b) mostra a correspondente resposta do sistema para um degrau unitário aplicado à sua entrada com  $K = 0,5547$  que satisfaz o valor desejado para  $\zeta$ .



(a) Determinando o valor do ganho para %OS ( $\zeta$ ) desejado através do RL.

(b) Resposta ao degrau unitário com  $K = 0,5547$ .

Figura 5.11: RL e resposta ao degrau para  $BoG(z)$  com outro valor de ganho.

### Exemplo\_2

Seja o sistema mostrado na figura 5.12.

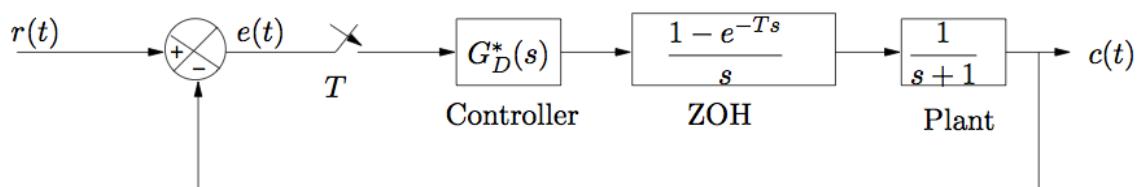


Figura 5.12: Sistema de controle discreto no tempo – exemplo 2.

Para  $T = 0,5$  segundos teremos:

$$\begin{aligned} Z\{Bo(s) G(s)\} &= Z\left\{\frac{1-e^{-Ts}}{s} \cdot \frac{1}{s+1}\right\} \\ &= (1-z^{-1})Z\left\{\frac{1}{s(s+1)}\right\} \\ &= (1-z^{-1})Z\left\{\frac{1}{s} - \frac{1}{s+1}\right\} \\ &= \frac{z-1}{z} \left[ \frac{z}{z-1} - \frac{z}{z-e^{-T}} \right] \\ &= \frac{1-e^{-T}}{z-e^{-T}} \end{aligned}$$

Vamos supor que será usado um **controlador integral – PI**, isto é,  $C(z) = \frac{Kz}{z-1}$ , assim teremos:

$$\begin{aligned} FTMA(z) &= C(z) \cdot BoG(z) \\ &= \frac{Kz}{(z-1)} \cdot \frac{(1-e^{-T})}{(z-e^{-T})} \end{aligned}$$

A equação característica fica então:

$$\begin{aligned} 1 + FTMA(z) &= 0 \\ 1 + \frac{Kz(1-e^{-T})}{(z-1)(z-e^{-T})} &= 0 \end{aligned}$$

que para  $T = 0,5$  resulta em:

$$L(z) = FTMA(z) = \frac{0,3935Kz}{(z-1)(z-0,6065)} = \frac{0,3935Kz}{z^2 - 1.607z + 0.6065} = \frac{N(z)}{D(z)}$$

$L(z)$  possui polos em  $z = 1$  e  $z = 0,6065$  e um zero em  $z = 0$ .

Os pontos de partida e chegada podem ser calculados a partir da equação (5.14):

$$\frac{dK}{dz} = \frac{D'(z)N(z) - D(z)N'(z)}{N^2(z)} = 0$$

que neste caso resulta em:

$$K = -\frac{D(z)}{N(z)} = -\frac{(z-1)(z-0,6065)}{0,3935z} = -\frac{z^2 - 1.607z + 0.6065}{0,3935z}$$

calculando  $\frac{dK}{dz}$  no MATLAB symbolic:

```
>> syms z
>> d=z^2-1.607*z+0.6065;
>> n=0.3935*z;
>> ganho= -d/n;
>> vpa(ganho)
ans =
-(2.5412960609911054637865311308767*(z^2 - 1.607*z + 0.6065))/z
>> %Note:
>> 1/0.3935
ans = 2.5413

>> dganhao=diff(ganho,z);
>> vpa(dganhao)
ans =
(2.5412960609911054637865311308767*(z^2 - 1.607*z + 0.6065))/z^2 -
(2.5412960609911054637865311308767*(2.0*z - 1.607))/z
```

```

>> simplifyFraction(dganho)
ans =
-(2000*z^2 - 1213)/(787*z^2)
>> 2000/787
ans = 2.5413
>> 1213/787
ans = 1.5413>> p=[2.5413 1.5413];
>> roots(p)
ans = -0.6065
>> %z^2 = -0.6065
>> sqrt(0.6065)
ans = 0.7788
>>

```

ou seja:

$$\begin{aligned}
\frac{dK}{dz} &= \frac{(2z - 1,607)(0,3935z) - (0,3935)(z^2 - 1,607z + 0,6065)}{(0,3935z)(0,3935z)} \\
&= \frac{(2z - 1,607)(0,3935z) - (0,3935)(z^2 - 1,607z + 0,6065)}{(0,3935z)(0,3935z)} \\
&= \frac{2,5413z^2 - 1,5413}{z^2}
\end{aligned}$$

$$\frac{dK}{dz} = \frac{z^2 - 0,6065}{0,3935z^2} = 0$$

cujas raízes resultam em:

$$\begin{aligned}
z^2 &= 0,6065 \\
z_1 &= 0,7788 \\
z_2 &= -0,7788
\end{aligned}$$

O **ganho crítico  $K_u$**  pode ser encontrado usando o **critério da magnitude**:

$$|FTMA(z)| = 1$$

como neste caso,  $FTMA(z)$  inclui o parâmetro variável (ganho)  $K$ , podemos reescrever a equação anterior como:

$$|K \cdot FTMA(z)| = 1$$

e isolando os termos  $K$  e  $FTMA(z)$ , obtemos:

$$|FTMA(z)| = \frac{1}{K}$$

O que aplicado neste caso resulta em:

$$\left| \frac{0,3935z}{(z-1)(z-0,6065)} \right| = \frac{1}{K}$$

Como o **ganho crítico  $K_u$**  corresponde ao ponto em que  $z = -1$ , então neste caso:

$$\begin{aligned}
\left| \frac{0,3935}{(-2)(-1,6065)} \right| &= \frac{1}{K_u} \\
K_u &= 8,1652
\end{aligned}$$

Assim, este sistema permanecerá estável para a faixa de ganho igual à:  $0 < K < 8,165$ .

A figura 5.13 representa o diagrama de RL. Dois traços partem dos dois polos em  $K = 0$ . Se vamos aumentando o valor de  $k$ , um dos traços segue até o zero e o outro tende a infinito (negativo).

A figura 5.13 pode ser obtida usando o MATLAB como mostra a listagem à seguir:

5.7. Diagrama do Lugar das Raízes (RL)

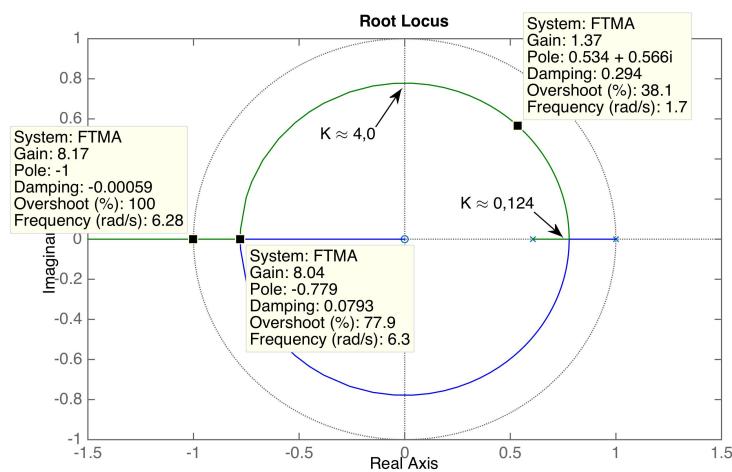


Figura 5.13: RL do sistema do exemplo 2 para  $T = 0,5$ .

```
>> clear all
>> num=1;
>> den=[1 1];
>> G=tf(num,den)
Transfer function:
1
_____
s + 1

>> T=0.5;
>> BoG=c2d(G,T)

Transfer function:
0.3935
_____
z - 0.6065

Sampling time: 0.5
>> num_c=[1 0];
>> den_c=[1 -1];
>> c=tf(num_c,den_c,T)

Transfer function:
z
_____
z - 1

Sampling time: 0.5
>> FIMA=series(c,BoG)

Transfer function:
0.3935 z
_____
z^2 - 1.607 z + 0.6065

Sampling time: 0.5
>> zpk(FIMA)

Zero/pole/gain:
0.39347 z
_____
(z-1) (z-0.6065)
```

```
Sampling time: 0.5
>> rlocus(FIMA)>>
```

Se para o sistema anterior fosse usado  $T = 1$  segundo, teríamos:

$$G(z) = \frac{0,6321Kz}{(z-1)(z-0,3679)}$$

com os pontos de partida e chegada dados por:

$$z^2 = 0,3679 \quad \rightarrow \quad z_1 = 0,6065 \text{ e } z_2 = -0,6065$$

e o ganho crítico, seria:  $K_u = 4,328$ .

Se para o sistema anterior fosse usado agora  $T = 2$  segundo, teríamos:

$$G(z) = \frac{0,8647Kz}{(z-1)(z-0,1353)}$$

e o ganho crítico, seria agora:  $K_u = 2,626$ .



Note como a medida que aumenta  $T$ , diminui a faixa para o ganho  $K$ .

# Capítulo 6

## Controladores Digitais Clássicos

### 6.1 Introdução

Este capítulo trata do projeto de controladores digitais “clássicos”, isto é, para sistemas SISO (*Single Input, Single Output*), no plano-z, usando como ferramenta de projeto, diagramas de Lugar da Raízes (em que pese suas eventuais limitações).

A figura 6.1 mostra a sequencia de estudo prevista para este capítulo.

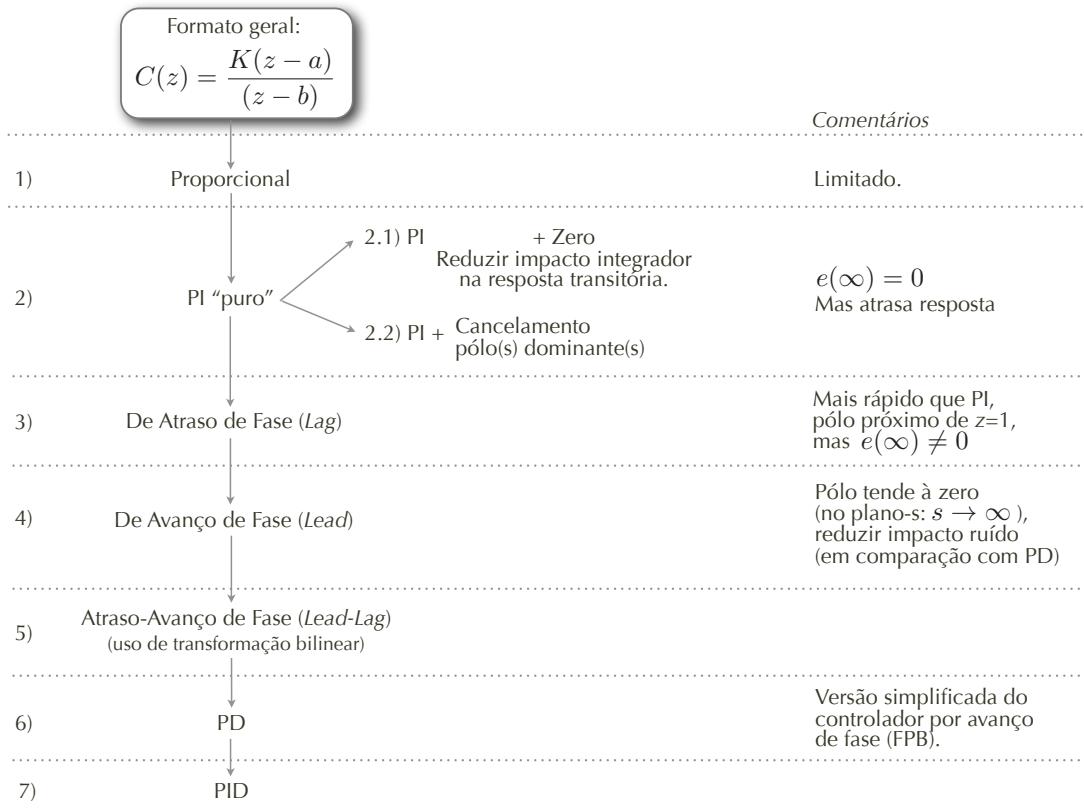


Figura 6.1: Sequencia de estudo prevista para este capítulo.

## 6.2 Formato Geral

De modo geral, os controladores digitais a serem estudados nesta seção obedecem ao formato genérico:

$$C(z) = K \cdot \frac{(z - a)}{(z - b)}$$

e todos serão aplicados sobre o seguinte tipo de planta (ver figura 6.2):

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

com  $T = 1$  (segundo).

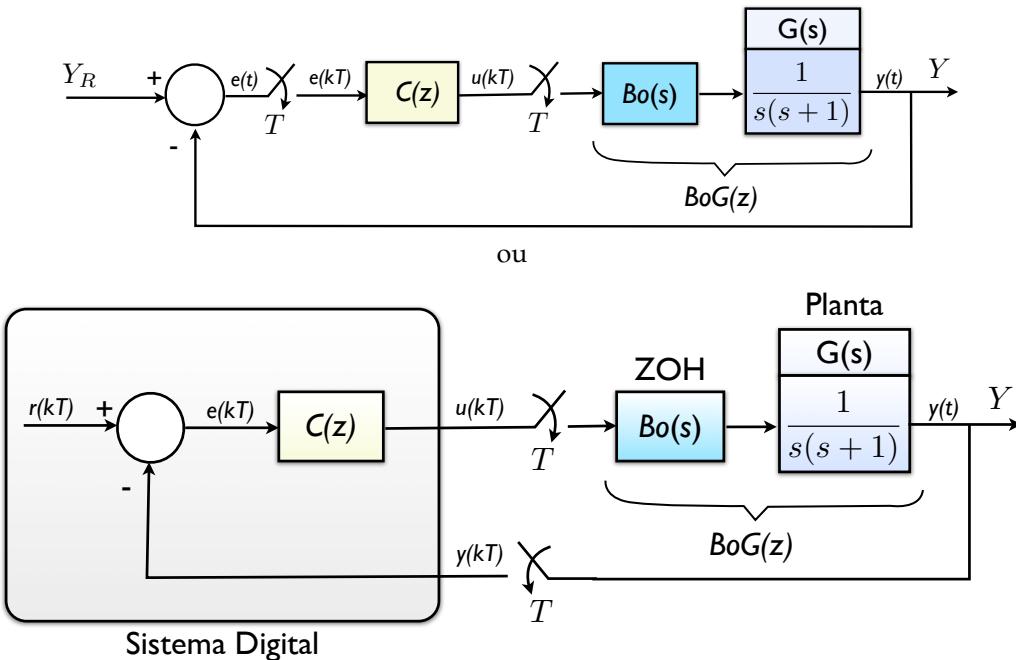


Figura 6.2: Sistema usando como exemplo.

Este sistema acompanhado de um sustentador de ordem zero,  $B_0G(z)$ , se transforma em:

$$B_0G(s) = (1 - e^{-sT}) \cdot \left( \frac{1}{s^2} - \frac{1}{s} + \frac{1}{s+1} \right)$$

transformando para plano  $\mathcal{Z}$  temos<sup>1</sup>:

$$\begin{aligned} B_0G(z) &= (1 - z^{-1}) \cdot Z \left\{ \frac{1}{s^2} - \frac{1}{s} + \frac{1}{s+1} \right\} \\ &= (1 - z^{-1}) \cdot \left[ \frac{Tz}{(z-1)^2} - \frac{z}{z-1} + \frac{z}{z-e^{-T}} \right] \\ &= \left[ \frac{(ze^{-T} - z + Tz) + (1 - e^{-T} - Te^{-T})}{(z-1)(z-e^{-T})} \right] \end{aligned}$$

para  $T = 1$ , temos então:

$$B_0G(z) = \frac{0,3678z + 0,2644}{(z-1)(z-0,3678)} = \frac{0,3678 \cdot (z + 0,7183)}{z^2 - 1,3678z + 0,3678} = \frac{a(z+b)}{(z-1)(z-a)}$$

onde  $a$ =pólo da planta resultado da digitalização da mesma ( $a = e^{-T}$ ); e  $b$ =zero da planta.

Esta expressão também pode ser obtida no MATLAB fazendo-se:

<sup>1</sup>A dedução apresentada de forma mais completa aparece na seção 5.5, pág. 27

```

>> num=1;
>> den=[1 1 0];
>> T=1;
>> [numd,dend]=c2dm(num,den,T)
numd =
      0      0.3679      0.2642
      1.0000    -1.3679      0.3679
>> printsys(numd,dend,'z')
num/den =
      0.36788 z + 0.26424
      _____
      z^2 - 1.3679 z + 0.36788
>> roots(numd)
ans = -0.7183
>> roots(dend)
ans =
      1.0000
      0.3679
>>

```

Ou também podemos usar outros comandos no MATLAB para obter os mesmos resultados:

```

>> num=1;
>> den=[1 1 0];
>> T=1;
>> G=tf(num,den)
Transfer function:
  1
  -----
  s^2 + s
>> zpk(G)
Zero/pole/gain:
  1
  -----
  s (s+1)
>> BoG=c2d(G,T)
Transfer function:
  0.3679 z + 0.2642
  -----
  z^2 - 1.368 z + 0.3679
Sampling time: 1
>> zpk(BoG)
Zero/pole/gain:
  0.36788 (z+0.7183)
  -----
  (z-1) (z-0.3679)
Sampling time: 1
>>

```

A resposta deste sistema para um impulso unitário,  $R(z)=1$  seria:  $Y(z) = G(z) \cdot 1$ , cuja resposta inicial no tempo poderia ser obtida pelo método da divisão longa em  $Z$ , ou seja:

$$\begin{array}{r}
0,3678z + 0,2644 \\
-0,3678z + 0,5031 + 0,1353z^{-1} \\
\hline
+0,7675 -0,1353z^{-1} \\
-0,7675 +1,0497z^{-1} + 0,2823z^{-2} \\
\hline
0,9145z^{-1} -0,2823z^{-2}
\end{array}
\qquad
\begin{array}{l}
z^2 - 1,3678z + 0,3678 \\
0,3678z^{-1} + 0,7675z^{-2} + 0,9145z^{-3} + \dots
\end{array}$$

ou seja,  $y(kT) = 0z^0 + 0,3678z^{-1} + 0,7675z^{-2} + 0,9145z^{-3} + \dots$ , ou ainda,  $y(0)=0$ ;  $y(1)=0,3678$ ;  $y(2)=0,7675$ ;  $y(3)=0,9145$ ; tendendo para 1. Note que pelo teorema do valor final (ver ??, pág. ??):

$$f(\infty) = \lim_{k \rightarrow \infty} f(k) = \lim_{z \rightarrow 1} (1 - z^{-1})F(z)$$

a resposta da planta em regime permanente para um impulso unitário aplicado à sua entrada resulta em:

$$y(\infty) = \lim_{k \rightarrow \infty} y(kT) = \lim_{z \rightarrow 1} (1 - z^{-1}) \cdot Y(z)$$

que neste caso nos dá:

$$\begin{aligned} y(\infty) &= \lim_{z \rightarrow 1} (1 - z^{-1}) \cdot Y(z) = \lim_{z \rightarrow \infty} \left( \frac{z-1}{z} \right) \cdot \left[ \frac{0,36788z + 0,26424}{(z-1)(z-0,36788)} \right] \\ y(\infty) &= \lim_{z \rightarrow 1} \frac{0,36788z + 0,26424}{z(z-0,36788)} = \frac{0,6321}{0,6321} = 1 \end{aligned}$$

## 6.3 Simples Controlador Proporcional, K

Fechando uma malha de realimentação unitária, este sistema resulta em:

$$FTMF(z) = \frac{Y(z)}{R(z)} = \frac{FTMA(z)}{1 + FTMA(z)} = \frac{F(z)}{1 + F(z)}$$

onde  $FTMA(z) = F(z)$  tem o seguinte formato:

$$F(z) = \frac{K(az + b)}{z^2 - (1 + a)z + a}$$

com  $a = 0,3678$  ( $a = e^T$ ) e  $b = 0,2644$ .

Os pólos em malha fechada serão as raízes da equação:  $[1 + F(z)] = 0$ . Esta também é a equação característica deste sistema:  $EC(z)$ :

$$FTMF(z) = \frac{0,36788(z + 0,7183)}{z^2 - z + 0,6321} = \frac{0,36788(z + 0,7183)}{(z - 0,5 - j0,6182)(z - 0,5 + j0,6182)}$$

Portanto, se obtém um equação característica no formato:

$$EC(z) = 1 + F(z) = z^2 - (1 + a)z + a + Kaz + Kb = 0$$

Quando  $K = 1$ , obtém-se:

$$EC(z) = z^2 - z + 0,6321$$

cujos pólos estarão em:  $z = 0,5 \pm j0,6182$  ou  $z = 0,8907 \angle \pm 45,55^\circ$ , portanto, ainda dentro do círculo unitário – este sistema responde então de forma subamortecida para uma entrada degrau.

Resolvendo no MATLAB temos:

```
>> ftmf=feedback(BoG,1) % fechando malha com H(s)=1 e realimentacao unitaria
Transfer function:
0.3679 z + 0.2642
_____
z^2 - z + 0.6321
Sampling time: 1
>> zpk(ftmf)
Zero/pole/gain:
0.36788 (z+0.7183)

(z^2 - z + 0.6321)
Sampling time: 1
>> [num_mf,den_mf]=tfdata(ftmf,'v') %Separando numerador e denominador da TF
num_mf =
      0    0.3679    0.2642
den_mf =
      1.0000   -1.0000    0.6321
>> polos_mf=roots(den_mf) %Descobrindo onde foram parar os polos de MF:
polos_mf =
    0.5000 + 0.6182i
    0.5000 - 0.6182i
```

```

>> [theta ,rho]=cart2pol(real(polos_mf(1)),imag(polos_mf(1)))
>> % Verificando se estao fora do circulo unitario em Z
theta = 0.8907
rho = 0.7951
>> deg=(rho*180)/pi % transformando de (rad) -> (deg)
deg = 45.5536
>>

```

Note que como este é um sistema do tipo 1 (planta com 1 integrador), ele responde com erro nulo para a entrada degrau, depois que realimentamos este sistema com um simples controlador proporcional:

Lembrando da definição do erro em regime permanente (eq. (4.2) – seção 4, pág. 7), temos:

$$\lim_{k \rightarrow \infty} e(kT) = \lim_{z \rightarrow 1} (z-1) \cdot \frac{1}{1+F(z)} \cdot Y_r(z)$$

O  $e(\infty)$  para entrada degrau unitário fica:

$$e(\infty)|_{u(t)} = \lim_{z \rightarrow 1} (z-1) \cdot \frac{1}{1+F(z)} \cdot \frac{z}{(z-1)} = \frac{1}{1 + \lim_{z \rightarrow 1} F(z)} = \frac{1}{1 + K_p}$$

onde:  $K_p$  = ganho de posição; e  $F(z) = FTMA(z)$ .

Neste caso em particular,  $BoG(z)$  ou  $FTMA(z)$  é um sistema do tipo 1 (com 1 integrador – pólo em  $z = 1$ ):

$$B_0 G(z) = FTMA(z) = \frac{0,36788(z+0,7183)}{(z-1)(z-0,3679)}$$

então:

$$\lim_{k \rightarrow \infty} e(kT) = \frac{1}{1 + \lim_{z \rightarrow 1} \frac{N(z)}{(z-1)D'(z)}} = \frac{1}{1 + K_p}$$

onde,  $K_p \rightarrow \infty$  e consequentemente,  $e(\infty) = 0$ .

Simulando no MATLAB, se obtém a resposta mostrada na figura 6.3:

```
>> step(ftmf)
```

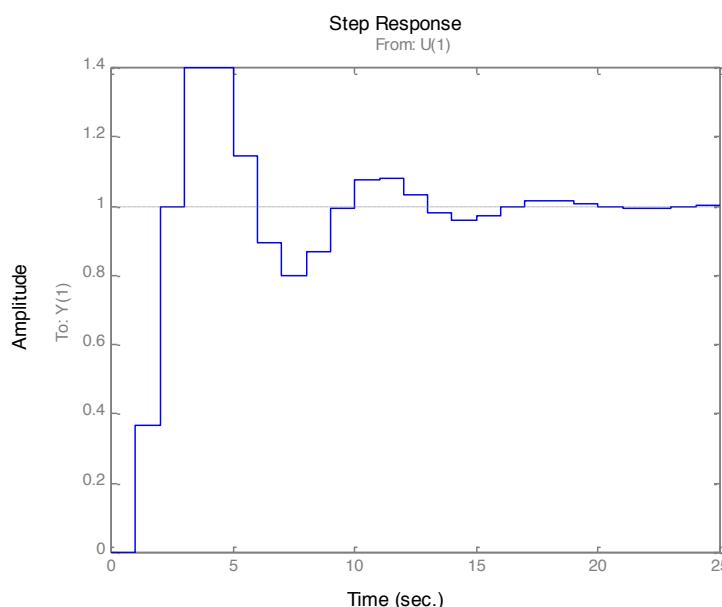


Figura 6.3: Resposta ao degrau unitário de  $BoG(z)$  em malha-fechada quando  $K = 1$ .

### 6.3.1 Determinando $K_{\max}$

Note porém que este ganho ( $K = 1$ ), gera um *overshoot* muito elevado para a resposta esperada para o sistema.

Podemos calcular qual o valor de  $K_{\max}$  que podemos incorporar neste sistema antes de que ele fique instável, usando o **método de Jury**.

Analisando:  $EC(z) = a_n z^n + a_{n-1} z^{n-1} + \dots + a_1 z + a_0 = 0$ , temos

Neste caso:  $EC(z) = z^2 - (1+a)z + a + Kaz + Kb = 0$ , onde  $a = 0,3679$  e  $b = 0,2642$ , então:

$$EC(z) = z^2 + (Ka - 1 - a)z + (a + Kb)$$

$$EC(z) = z^2 + (0,3678K - 1,3679)z + (0,3678 + 0,2644K) = 0$$

Observando as pré-condições de estabilidade:

- a)  $|a_0| < a_2$ , que neste caso, se transforma em:  
 $|a + Kb| < 1$ , que resulta em:  
 $K < \frac{1-a}{b}$ , ou seja,  $K < 2,3922$ , e:  
 $K > \frac{1-a}{b}$ , ou seja,  $K > 2,3922$  (condição desprezada).
- b)  $EC(1) > 0$ , para sistemas de ordem par, então:  
 $EC(1) = 1 + Ka - 1 - a + a + Kb > 0$ ,  
daqui concluímos que:  $K > 0$ ;
- c)  $(-1)^n EC(-1) > 0$ , então:  $EC(-1) > 0$ , verificando:  
 $EC(-1) = 1 - Ka + 1 + a + a + Kb > 0$ ,  
 $K(b - a) > -2 - 2a$ ,  
 $K < \frac{2+2a}{b-a}$ , que neste caso, nos fornece:  
 $K < -12.1986$  (condição refutada: ganho negativo).

Analisando as condições, chegamos à conclusão de que o ganho neste caso (note que  $T = 1$ ), deve ficar entre:  $0 < K < 2,3922$ .

Usando-se o MATLAB para comprovar estes resultados, temos:

```
>> rlocus(numd,dend)
>> x=[-1:0.1:1]; y=sqrt(ones(1,length(x))-x.^2);
>> hold on
>> plot(x,y,':',x,-y,':')
>>axis('equal')
```

que gera o gráfico mostrado na figura 6.4.

Podemos confirmar o valor máximo que poderia ser adotado para  $K$ , fazendo no MATLAB:

```
>> rlocfind(numd,dend)
Select a point in the graphics window
selected point =
    0.2615 + 0.9455 i
ans =
    2.2789}
```

Conferindo no gráfico gerado na figura 6.5.

Se para o  $K = 1$  pode ser percebido um *overshoot* de 40%, diminuindo-se este ganho, este valor pode ser baixado se levarmos em consideração ainda que este sistema em malha fechada responde como um sistema de segunda ordem, então basta definir valores convenientes para o fator de amortecimento (principalmente),  $\zeta$ , e para a freqüência natural,  $\omega_n$ . Um valor melhor tolerado, seria um *overshoot* de 5% que se obtém com  $\zeta \approx 0,7$ .

Podemos graficar no MATLAB uma linha guia para este valor a fim de descobrir que valor de ganho  $K$  deveria ser adotado:

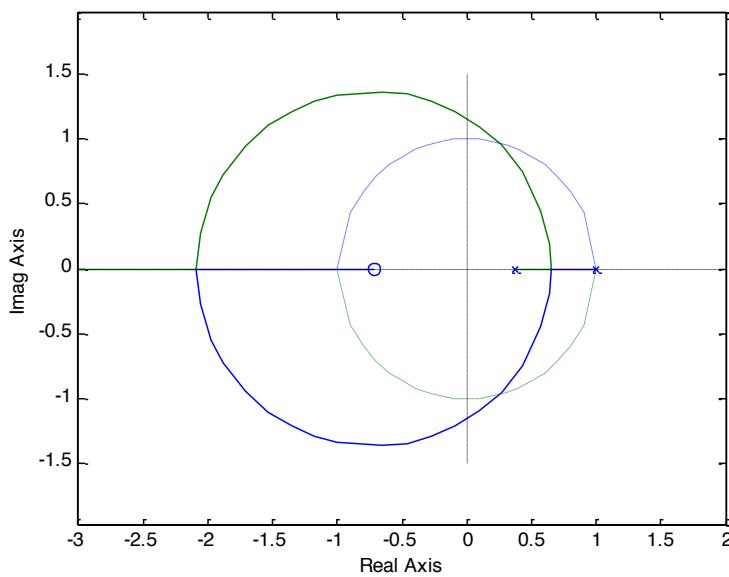


Figura 6.4: RL para  $BoG(z) = \frac{0.3678(z+0.7183)}{(z-1)(z-0.3678)}$ .

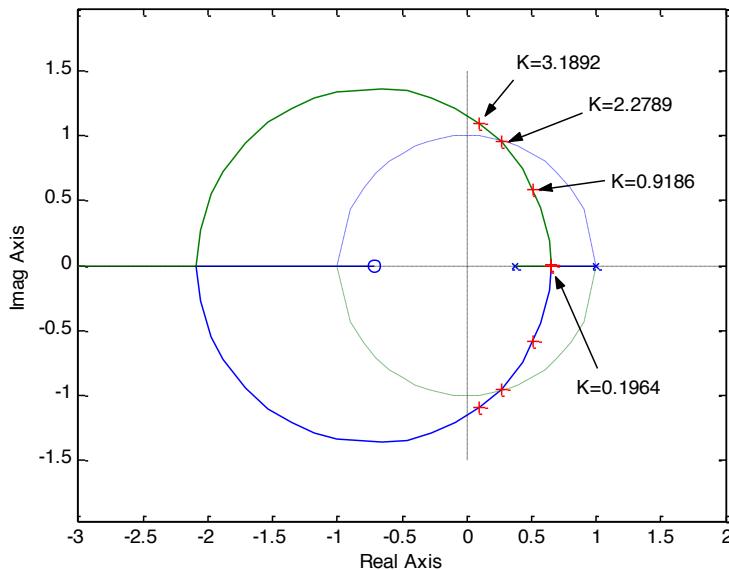


Figura 6.5: Sintonizando o controlador proporcional no RL.

```

>> zgrid(0.7,0.5,'new')
>> rlocus(numd,dend)>> axis('equal')
>> hold on
>> rlocfind(numd,dend)
Select a point in the graphics window
selected point =
    0.6077 + 0.2537i
ans =
    0.3284
>>

```

O gráfico gerado pelo MATLAB é mostrado na figura 6.6.

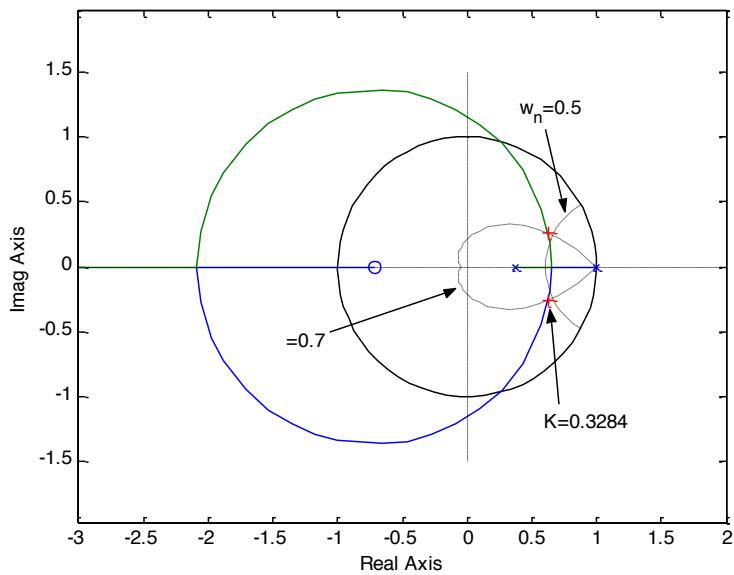


Figura 6.6: Ajustando ganho do controlador proporcional para limitar %OS.

Comprovando a saída no domínio tempo, podemos fazer no MATLAB:

```
>> K=0.32;
>> [num_mf,den_mf]=feedback(K*numd,dend,1,1,-1);
>> printsys(num_mf,den_mf,'z')
num/den =
 0.11772 z + 0.084557
  -----
  z^2 - 1.2502 z + 0.45244
>>dstep(num_mf,den_mf)
```

o que gera o gráfico mostrado na figura 6.7.

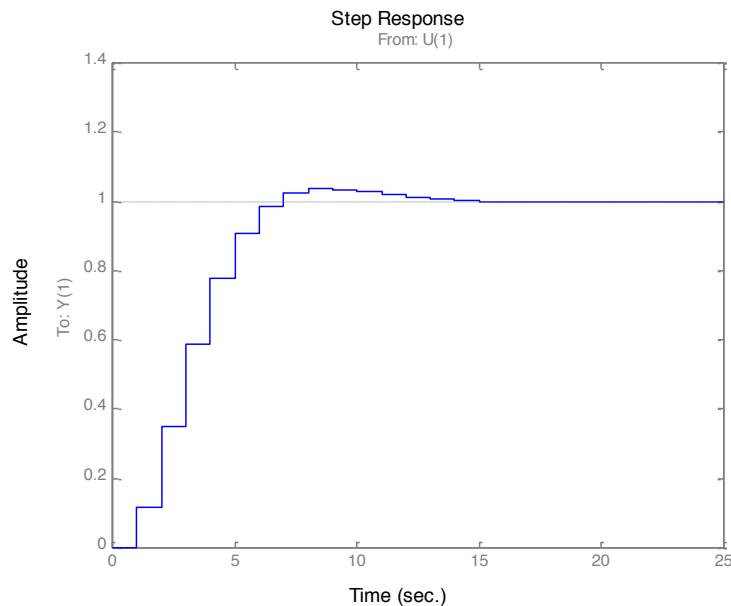


Figura 6.7: Resposta ao degrau depois de nova sintonia do controlador proporcional – note o menor overshoot.

O valor em regime permanente para  $y(kT)$  quando  $R(z) =$  entrada degrau unitário, resulta em:

$$y(\infty) = \lim_{z \rightarrow 1} (z - 1) \cdot \text{FTMF}(z) \cdot \frac{z}{z - 1} = \lim_{z \rightarrow 1} \text{FTMF}(z)$$

usando o MATLAB para calcular  $y(\infty)_{\text{Degrau}}$ :

```
>> polyval(num_mf, 1) / polyval(den_mf, 1)
ans =
1.0000}
>>
```

Se fosse desejado simular um outro tipo de entrada ao sistema, poderíamos fazê-lo facilmente no MATLAB. Supondo o caso de se querer simular uma entrada parabólica:

```
>> for i=0:10, u(i+1)=i^2; end
>> plot(u, 'b: ')
>> hold on
>> dlsim(num_mf, den_mf, u)
```

Que gera o gráfico mostrado na figura 6.8.

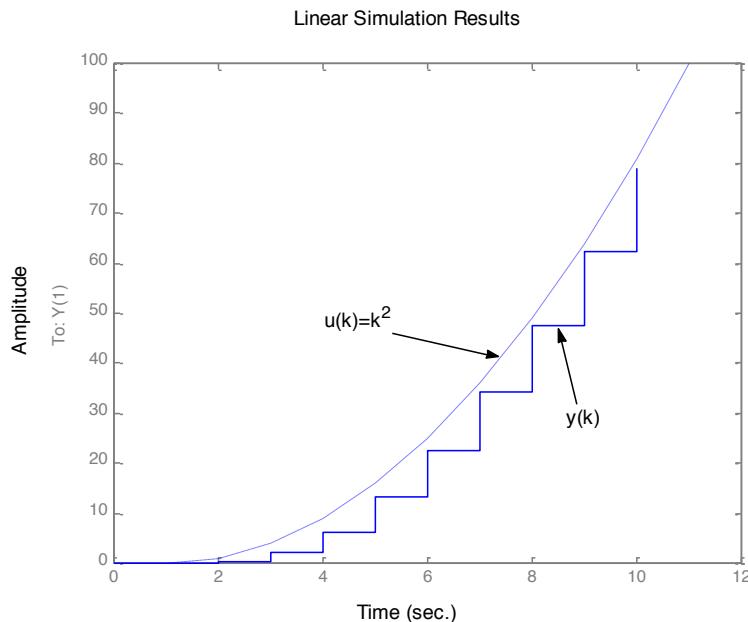


Figura 6.8: Resposta do sistema a uma entrada parabolica.

Simulando uma entrada rampa para o caso anterior:  $R(z) = \frac{Tz}{(z - 1)^2}$ , temos:

```
>> num_r=[1 0];
>> den_r=conv([1 -1],[1 -1])
>> [num_mf2,den_mf2]=series(num_mf,den_mf,num_r,den_r);
>> printsys(num_mf2,den_mf2,'z');
num/den =
0.11772 z^2 + 0.084557 z
_____
z^4 - 3.2502 z^3 + 3.9528 z^2 - 2.155 z + 0.45244
>>
```

Determinando o erro em regime permanente temos:

$$y(\infty) = \text{cte}$$

que no MATLAB pode ser calculado fazendo-se:

<< Falta revisar esta parte (7 nov 2013) >>

ou seja, observamos que este controlador (proporcional) não é suficiente para garantir erro nulo para entrada rampa, falta um integrador.

Simulando a entrada rampa no sistema para comprovar o erro em regime permanente; neste caso:

$$r(kT) = z^{-1} \left\{ \frac{Tz}{(z-1)^2} \right\} = k \cdot T:$$

```
>> for i=0:25; k(i+1)=i; r(k+1)=k*T; end;
>> plot(k,r,'b:');
>> hold on
>> dlsim(num_mf,den_mf,r)
>>
```

se gera o gráfico mostrado na figura 6.9.

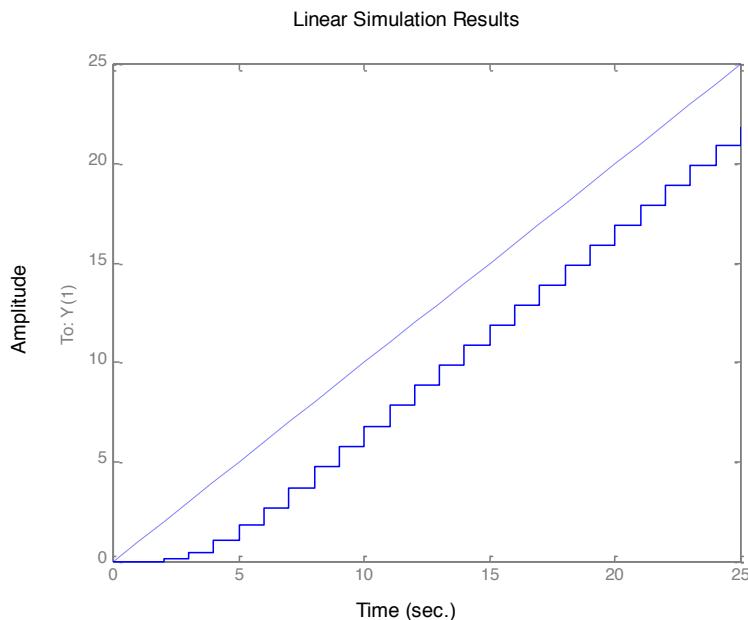


Figura 6.9: Resposta do sistema à uma entrada rampa.

## Pólos Dominantes no Plano-Z

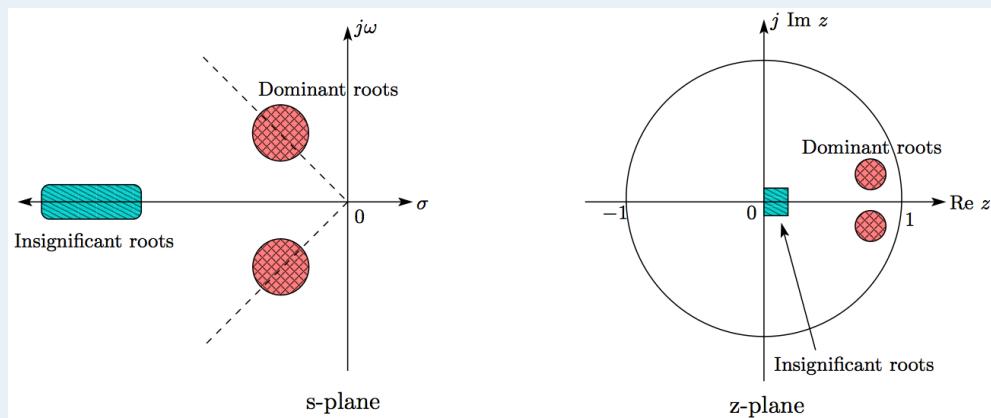
Assim como ocorre no plano-s, algumas raízes no plano-z geram efeito mais preponderante que outros. É importante no projeto de um controlador separar estas raízes consideradas dominantes.

No plano-s, as raízes localizadas próximas do eixo  $j\omega$  no semi-plano esquerdo eram consideradas dominantes porque sua correspondente resposta temporal mostra um decaimento mais lento. As raízes que estão localizadas mais longe do eixo  $j\omega$  correspondem à respostas que decaim mais rápido.

No plano-z, as raízes dominantes são aquelas que estão dentro e próximas do círculo unitário enquanto a região (no plano-z) menos significativa é que está próxima da origem do plano-z.

O eixo real negativo é geralmente evitado uma vez que sua resposta temporal é naturalmente oscilatória (sinal alternando entre instantes de amostragem).

A figura AA mostra as regiões para raízes dominantes e menos, tanto no plano-s quanto no plano-z.



Obs.: Extraído de: Kar and Majhi[Module 4, Lecture 2].

Figura AA: Mapeamento de pólos e zeros de um sistema de segunda ordem.

No plano-z, as raízes próximas da origem são menos significativas do ponto de vista de *overshoot* e fator amortecimento.. Entretanto, estas raízes não podem ser completamente desconsideradas uma vez que um número excessivo de pólos em relação à zeros gera um efeito de *delay* na região inicial da resposta no tempo. Isto é, adicionar pólos em  $z = 0$  não vai afetar nem o *overshoot* e nem o fator amortecimento, mas o tempo de resposta vai apresentar um *delay* adicional de um período de amostragem.

A melhor maneira de simplificar (tonar menos complexo, com menor ordem) um sistema no plano-z é trocar os pólos próximos da origem por pólos em  $z = 0$  o que vai simplificar a análise uma vez que pólos em  $z = 0$  correspondem a puros atrasos no tempo.



# Capítulo 7

## Estudo de Caso Diferentes Controladores Digitais

### 7.1 Planta Adotada

A seguinte planta será usada para realizar o projeto de diferentes tipos de controladores digitais usando como ferramenta o diagrama do lugar das raízes:

$$G(s) = \frac{1}{(s+10)(s+2)(s+1)}$$

que amostrada à  $T = 0,1$  segundos, resulta em:

$$BoG(z) = \frac{0.00012224(z+2.747)(z+0.1903)}{(z-0.9048)(z-0.8187)(z-0.3679)}$$

O diagrama do lugar das raízes correspondente à esta planta é mostrado na figura 7.1.

Como requisito de controle para esta planta, foi definido *overshoot* máximo de: %OS < 5% e tempo de assentamento,  $T_s$  como sendo menor ou igual à metade do valor encontrado para o controlador proporcional.

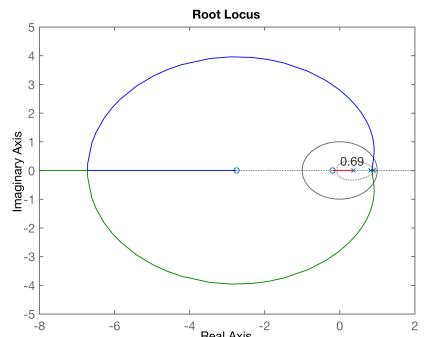


Figura 7.1: RL encontrado para  $BoG(z)$ .

Comandos usados no MATLAB:

```
>> G=tf([1], poly([-10 -1 -2]));
>> zpk(G)
>> T=0.1;
>> BoG=c2d(G,T);
>> zpk(BoG)
```

## 7.2 Controlador Proporcional

A ideia é fechar a malha com um simples controlador proporcional mas sintonizando-se o ganho de forma que o  $\%OS < 5\%$ . Como  $\zeta$  está diretamente associado com  $\%OS$ , determina-se então o valor do fator de amortecimento,  $\zeta$ , com base no  $\%OS$  requerido:

```
%\begin{lstlisting}
>> OS=5; % OS = 5%
>> zeta=(-log(OS/100))/(sqrt(pi^2+(log(OS/100)^2)))
zeta = 0.6901
%\end{lstlisting}
```

Através do diagrama do lugar das raízes (RL), foi determinando o ganho de  $K = 18$ , conforme pode ser visto na figura 7.2.

O ganho encontrado anteriormente permitiu alcançar um tempo de assentamento de  $t_s = 3,23$  segundos com overshoot de 4,88% conforme mostra a figura 7.3. Então, o valor desejado para o tempo de assentamento será de  $t_s(\text{Desejado}) = 1,6250$  segundos.

A figura 7.4 mostra as amplitudes desenvolvidas para a ação de controle proporcional. O gráfico de  $u[kT]$  foi obtido percebendo que:

$$U(z) = C(z) \cdot E(z)$$

mas como:  $E(z) = R(z) - Y(z)$ , então:

$$U(z) = C(z) [R(z) - Y(z)]$$

sabemos também que:  $Y(z) = U(z) \cdot BoG(z)$ , então:

$$U(z) = C(z) [R(z) - U(z) \cdot BoG(z)]$$

$$U(z) [1 + C(z) \cdot BoG(z)] = C(z) \cdot R(z)$$

$$\begin{aligned} U(z) &= \frac{C(z) \cdot R(z)}{1 + C(z) \cdot BoG(z)} \\ &= \frac{C(z)}{1 + FTMA(z)} \cdot R(z) \end{aligned}$$

portanto, pode-se usar a função `step(.)` do MATLAB para obter o gráfico de  $u[kT]$ , bastando realizar:

$$>> step\left(\frac{C(z)}{1 + FTMA(z)}\right)$$

ou neste caso:

```
%\begin{lstlisting}
>> aux=K/(1+K*BoG); % aux = var. auxiliar
>> zpk(aux)
18 (z-0.9048) (z-0.8187) (z-0.3679)
-----
(z-0.3534) (z^2 - 1.736z + 0.7679)

>> figure; step(aux)
>> title('control action')
%\end{lstlisting}
```

Os comandos anteriores geraram a figura 7.4, donde percebe-se que foi desenvolvida uma ação de controle que atingiu a amplitude máxima de  $u_{\max} = 18$ .

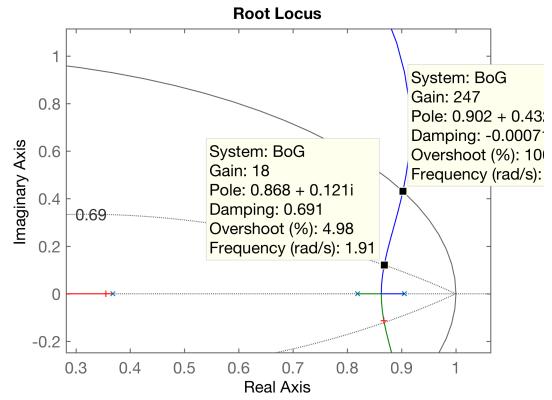


Figura 7.2: Ganco encontrado para controlador Proporcional.

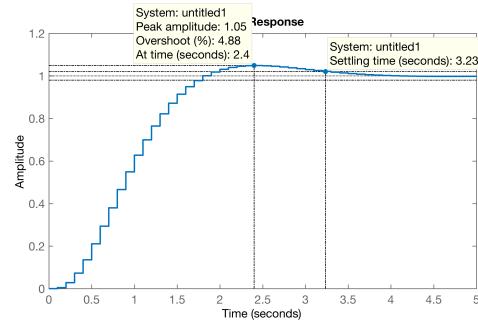


Figura 7.3: Resposta ao degrau, controlador proporcional.

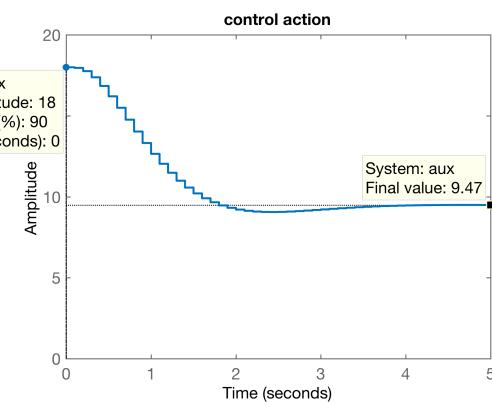


Figura 7.4: Amplitudes desenvolvidas pelo controlador proporcional.

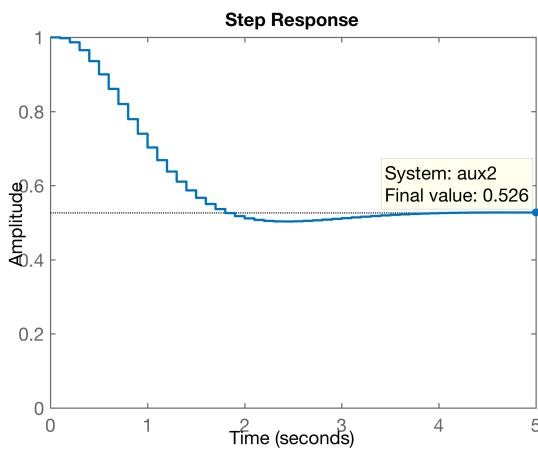


Figura 7.5: Sinal do erro alcançado para o controlador proporcional.

A figura 7.5 mostra o sinal de erro resultante para o controlador proporcional. Note que o erro em regime permanente não foi anulado:  $e_{\text{step}}(\infty) = 0,527$ .

O código usado para obter o gráfico de  $e[kT]$  segue:  
Notar que:

$$\begin{aligned} E(z) &= R(z) - Y(z) \\ E(z) &= R(z) - E(z) \cdot FTMA(z) \\ E(z)[1 + FTMA(z)] &= R(z) \\ E(z) &= \frac{R(z)}{1 + DTM(z)} \\ E(z) &= \left[ \frac{1}{1 + FTMA(z)} \right] \cdot R(z) \end{aligned}$$

então basta realizar no MATLAB:

```
>> step(1/FTMA)
```

```
>> aux2=1-FTMF;
>> zpk(aux2)

(z-0.9048) (z-0.8187) (z-0.3679)
_____
(z-0.3534) (z^2 - 1.736z + 0.7679)

>> figure; step(aux2)
>> title ('Error signal')
>> axis([0 5 0 1])
>> dcgain(aux2) % valor do erro em regime permanente
ans = 0.5263
>> % Note que ans*18 = 9,4734 (valor em regime permanente do sinal de controle)
```

## 7.3 Controlador Dead-beat (ou Controle por tempo mínimo, finito)

**Idéia:** fazer o sistema em malha fechada atingir o regime permanente no menor tempo possível (no tempo mínimo absoluto,  $k = 1$ ) ou dentro de um número finito de períodos amostrados.

6\_Controle\_Digital  
controle\_digital  
\_deadbeat\_1.tex

Versão anterior:  
Controlador por Zda  
controlador\_zdau.tex

Para tanto,  $C(z)$  deve conter todos os pólos e zeros de  $BoG(z)$ , isto é:

- Os pólos de  $C(z)$  contêm os zeros de  $BoG(z)$ ;
- Os zeros de  $C(z)$  contêm os pólos de  $BoG(z)$ ;

e além disto, são acrescentados tantos integradores quanto os necessários para anular (zerar) o erro em regime permanente para o tipo de entrada especificado, ou:

$$C(z) = K_c \cdot \frac{\prod_{n=1}^{PE} (z - p_{BoG(z)_n})}{\prod_{m=1}^{ZE} (z - z_{BoG(z)_m}) \cdot (z - 1)^w} \quad (7.1)$$

onde:

$K_c$	= ganho do controlador;
$ZE$	= quantidade de zeros estáveis da planta ( $BoG(z)$ );
$PE$	= quantidade de pólos estáveis da planta;
$z_{BoG(z)_n}$	= $n$ -ésimo zero da planta;
$p_{BoG(z)_m}$	= $m$ -ésimo pólo da planta;
$(z - 1)$	= são integradores impostos pelo integrador no sistema em malha fechada;
$w$	= número de integradores necessários para tornar nulo o erro em regime permanente – depende do tipo de entrada com a qual se está esperando operar a planta (degrau, rampa, parábola, etc).

Note que o controlador **não cancela pólos ou zeros instáveis da planta**. Se estes existirem, estes são desconsiderados na equação de  $C(z)$ .

### 7.3.1 Exemplo 1: O controlador Deadbeat

Seja a função transferência de uma planta digital dada por:

$$BoG(z) = \frac{z + 0,6}{3z^2 - z - 1}$$

Suponha que acrescentaremos o controlador digital em cascata com a planta:

$$C(z) = \frac{3z^2 - z - 1}{(z - 1)(z + 0,6)}$$

↑  
 Integrador  
 (propositoal)

Desta forma, a função transferência de malha aberta torna-se:

$$FTMA(z) = C(z)BoG(z) = \frac{1}{z - 1}$$

E assim a função transferência de malha-fechada fica:

$$FTMF(z) = \frac{FTMA(z)}{1 + FTMA(z)} = \frac{\frac{1}{z-1}}{1 + \frac{1}{z-1}} = \frac{\frac{1}{z-1}}{\frac{z-1+1}{z-1}} = \frac{1}{z} \quad (7.2)$$

Note que a eq. (7.2) quando excitada por uma entrada degrau unitário resulta em:

$$Y(z) = \frac{z}{(z-1)} \cdot \frac{1}{z} = \frac{1}{z-1} = z^{-1} + z^{-2} + \dots$$

↓  
 Entrada  
 Degrau

ou seja, a saída  $y(k)$  representa a resposta do sistema em malha-fechada para uma entrada degrau unitário e que inicia em  $k = 1$ , ou seja, um período de amostragem seguinte ao acionamento do sistema.

Note que  $y(k)$  alcança a resposta desejada igual a 1 com apenas 1 período de amostragem de atraso e que a saída permanece neste estado indefinidamente. Ou seja, que logo após 1 único período de amostragem, o sistema já se encontra em regime permanente.

Este tipo de resposta é conhecida como resposta “*dead beat*”.

Entretanto deve ser percebido que o controlador  $C(z)$  não garante que oscilações (*ripples*) não vão ocorrer entre dois instantes de amostragem de  $y(k)$ , ou seja, entre os instantes amostrados da resposta do sistema.

### 7.3.2 Exemplo: Projeto de Controlador Deadbeat

Seja a planta indicada à seguir, projetar um controlador pelo critério do tempo mínimo, garantindo ainda erro nulo em regime permanente para entrada degrau unitário. Dados da planta:

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

**Solução:**

- Primeiramente determinamos a função transferência amostrada da planta:  $B_0 G(z)$ . Usando o MATLAB, podemos fazer:

```

>> num=10;
>> help poly
POLY Convert roots to polynomial.
  POLY(A), when A is an N by N matrix, is a row vector with
  N+1 elements which are the coefficients of the
  characteristic polynomial, DET(lambda*EYE(SIZE(A))-A) .

>> den=poly([-1 -10]);
>> G=tf(num,den);
>> zpk(G) %confirmado que entramos corretamente com os dados para G(s)
Zero/pole/gain:
  10
  -----
  (s+10) (s+1)

>> %Resta discretizar G(s):
>> T=0.1; %guardando o período de amostragem adotado.
>> BoG=c2d(G,T) %aplicando o ZOH sobre G(s)
Transfer function:
  0.0355 z + 0.02465
  -----
  z^2 - 1.273 z + 0.3329
Sampling time: 0.1
>> zpk(BoG) %mostrando dados expandidos
Zero/pole/gain:
  0.035501 (z+0.6945)
  -----
  (z-0.9048) (z-0.3679)
Sampling time: 0.1

```

Então  $BoG(z)$  assume o formato:  $BoG(z) = \frac{0.035501(z + 0.6945)}{(z - 0.9048)(z - 0.3679)}$ .

- Baseado na eq. (7.1) notamos que neste caso, necessitamos que nosso controlador tenha o seguinte formato:

$$C(z) = \frac{K_c \cdot (z - 0.9048)(z - 0.3679)}{(z + 0.6945)(z - 1)} \quad (7.3)$$

note que com o objetivo de anular o erro em regime permanente para uma entrada degrau foi introduzido um integrador (termo  $(z - 1)$ ) no controlador. De qualquer forma, seria necessário acrescentar algum pólo extra ao controlador a fim de garantir sua realizabilidade (grau do denominador  $\geq$  grau do numerador). Usando o MATLAB para especificar  $C(z)$  podemos fazer:

```
>> % para compor C(z). Ou separar os polos e zeros
help tfdata
— help for lti/tfdata —

TFDATA Quick access to transfer function data.

[ NUM,DEN ] = TFDATA( SYS ) returns the numerator(s) and denominator(s)
of the transfer function SYS. For a transfer function with NY
outputs and NU inputs, NUM and DEN are NY-by-NU cell arrays where
the (I,J) entry specifies the transfer function from input J to
output I. SYS is first converted to transfer function if necessary.

For a single SISO model SYS, the syntax
[ NUM,DEN ] = TFDATA( SYS, 'v' )
returns the numerator and denominator as row vectors rather than
cell arrays.

>> [num_BoG,den_BoG]=tfdata(BoG,'v') %extraindo dados de BoG(z)
num_BoG =
      0      0.0355      0.0247
den_BoG =
      1.0000    -1.2727      0.3329
>> roots(num_BoG)
ans =    -0.6945
>> polos_BoG=roots(den_BoG)
polos_BoG =
      0.9048
      0.3679
>> zeros_BoG=roots(num_BoG)
zeros_BoG =
     -0.6945
>> num_c=poly(pолос_BoG); %construindo numerador de C(z), polinomialmente
>> den_c=poly([1 zeros_BoG]); %montando o denominador de C(z), incluindo o integrador
>> C=tf(num_c,den_c,T)
Transfer function:
z^2 - 1.273 z + 0.3329
_____
z^2 - 0.3055 z - 0.6945
Sampling time: 0.1
>> zpk(C) %verificando formato do controlador...
Zero/pole/gain:
(z-0.9048) (z-0.3679)
_____
(z-1) (z+0.6945)
Sampling time: 0.1
```

- O passo seguinte é fechar a malha de realimentação, incluindo a determinação de  $K_c$ . Usando o MATLAB:

```
>> FIMA=series(C,BoG) %determinando primeiro a FTMA(z)
Transfer function:
0.0355 z^3 - 0.02053 z^2 - 0.01956 z + 0.008206
```

```

z^4 - 1.578 z^3 + 0.02728 z^2 + 0.7821 z - 0.2312
Sampling time: 0.1
>> zpk(FTMA) % confirmando localizacao final dos polos e zeros de C(z)BoG(z):Zero/pole/gain:
0.035501 (z+0.6945) (z-0.9048) (z-0.3679)

(z+0.6945) (z-0.9048) (z-1) (z-0.3679)
Sampling time: 0.1
% note que usando o comando minreal(..) se poderia cancelar polos-zeros numericamente, por exemplo:
>> FTMAR=minreal(FTMA,1e-4);
>> zpk(FTMAR) % versao numerica com polos-zeros cancelados
Zero/pole/gain:
0.035501

(z-1)
Sampling time: 0.1
% mas continuar com FTMA(z) permite vizualizar os cancelamentos polos-zeros
% no diagrama do lugar das raizes

```

Notamos então que a função transferência em malha-aberta,  $FTMA(z)$  fica:

$$FTMA(z) = C(z) \cdot BoG(z) = \frac{0.035501}{(z-1)} \quad (7.4)$$

ou seja, resultou exatamente no que se pretendia.

Inspecionando a eq. (7.4) percebe-se que nosso sistema em malha-aberta contém apenas 1 pólo em  $z = 1$ . Quando o laço de realimentação for fechado, com a introdução do ganho do controlador,  $K_c$ , se percebe que este pólo de malha-aberta partirá do ponto  $z = 1$  e “caminhará” até  $z = -\infty$ , a medida que  $K_c$  aumenta. Ou seja, teremos como resultado apenas 1 pólo, real, em malha-fechada.

Lembrando a forma como pólos simples reagem no tempo para uma entrada degrau, fica fácil perceber que quanto mais próximo da origem do plano-z estiver o pólo de malha-fechada, tão mais rápido será a resposta do sistema (ver Apêndice: B.1). Como queremos que a planta entre em regime permanente no menor tempo possível, tentaremos fixar o pólo de malha-fechada em  $z = 0$ , ajustando de forma adequada o ganho do controlador.

Usando novamente o MATLAB para determinar o  $K_c$  teremos:

```

% continuando com o RL da FTMA completa para confirmar os cancelamentos
>> rlocus(FTMA)
>> axis equal

```

Os últimos comandos no MATLAB geram o gráfico mostrado na figura 7.6.

Procedemos com mais comandos no MTLA para “sintonia” do controlador:

```

>> hold on % mantem RL anterior na janela grafica
>> [K, polos_MF]=rlocfind(FTMA)
Select a point in the graphics window

selected_point =
    0.0036 - 0.0031i

K =
    28.0686

polos_MF =
    -0.6945
    0.9048
    0.3679
    0.0035

```

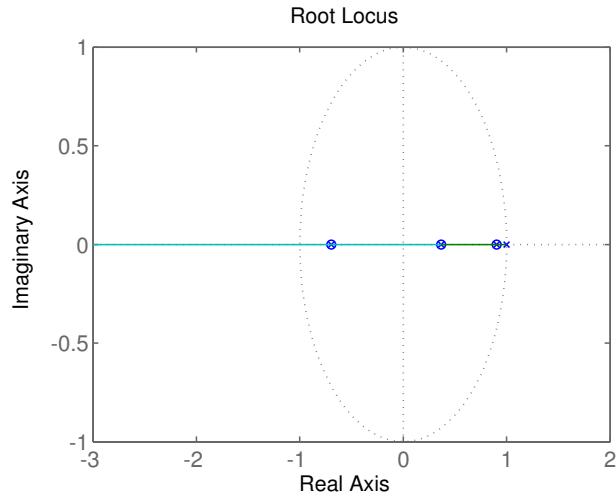


Figura 7.6: RL para sistema com controlador Deat-beat.

```

>> % Se tentou posicionar os polos de MF sobre a origem do plano-z
>> % verificando o resultado obtido...
>> figure ; % abre nova janela grafica >> FTMF=feedback (K*FTMA, 1) ; % fechando a malha com o ganho
encontrado
>> zpk(FTMF) % confirmando polos e zeros de MF
Zero/pole/gain :
0.99645 (z+0.6945) (z-0.9048) (z-0.3679)
(z+0.6945) (z-0.9048) (z-0.3679) (z-0.00355)

Sampling time: 0.1
>> % Hum... notamos um polo em z=-0.6945 (gera oscilação saída controlador)
>> step (FTMF)

```

Os últimos comandos geram as figuras 7.7(a) e (b).

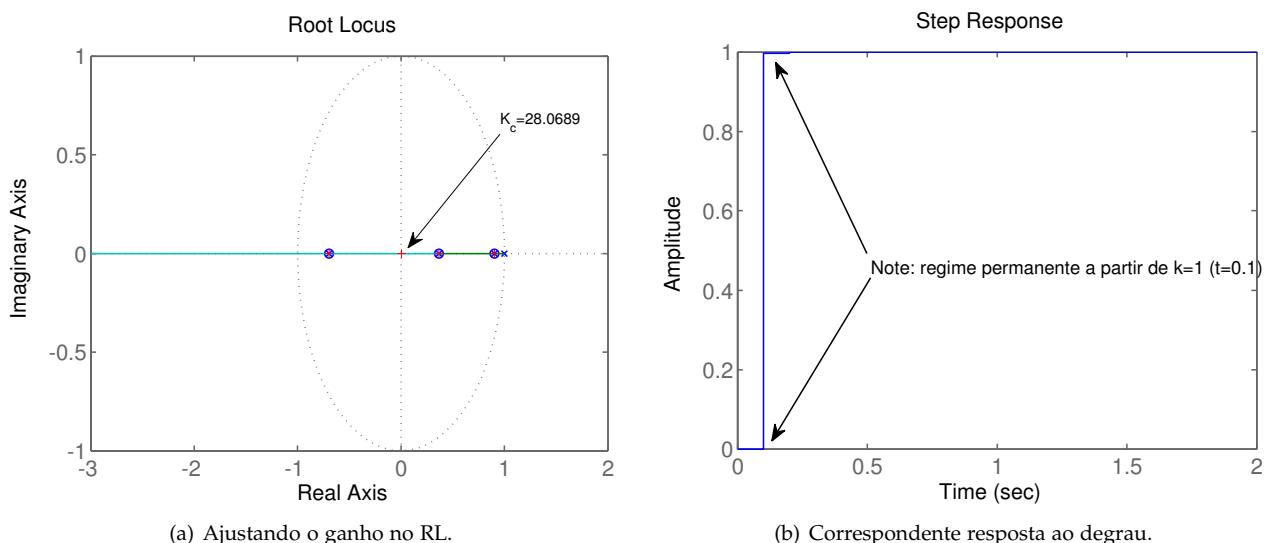


Figura 7.7: RL ajustado e resultado da saída do sistema com controlador deat-beat sintonizado.

Se percebe pela figura 7.7(b) que a planta entrou em regime permanente praticamente depois da primeira amostra ( $k = 1$ ). Podemos ainda determinar algebraicamente o valor de regime permanente da saída do

sistema em malha fechada:

$$y(\infty) = \lim_{k \rightarrow \infty} y(k) = \lim_{z \rightarrow 1} (1 - z^{-1}) \text{FTMF}(z) \frac{1}{(1 - z^{-1})}$$

Usando o MATLAB para determinar o valor de  $y(\infty)$ :

```
>> [num_mf,den_mf]=tfdata(FTMF, 'v')
num_mf =
      0     0.9965    -0.5762    -0.5490     0.2303
den_mf =
      1.0000    -0.5818    -0.5489     0.2331    -0.0008
>> y_infny=polyval(num_mf,1)/polyval(den_mf,1)
y_infny =
      1.0000
```

Mas este tipo de resposta têm sua contraparte: um elevado esforço de controle. Para comprovar as amplitudes envolvidas com o sinal de controle,  $u(k)$  podemos simular este sistema no MATLAB/Simulink – ver figura 7.8.

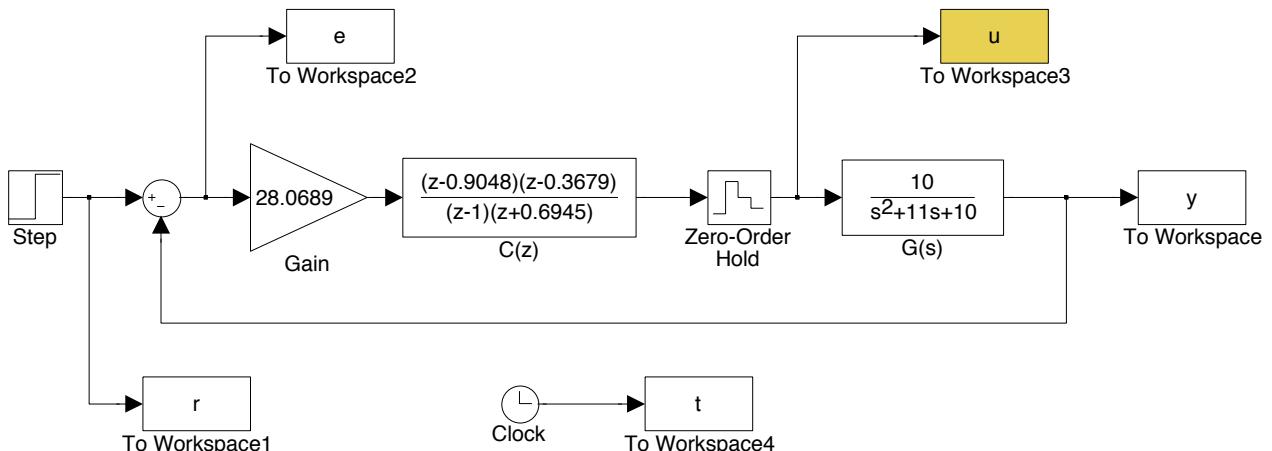


Figura 7.8: Simulação do controlador Dead-beat no MATLAB/Simulink.

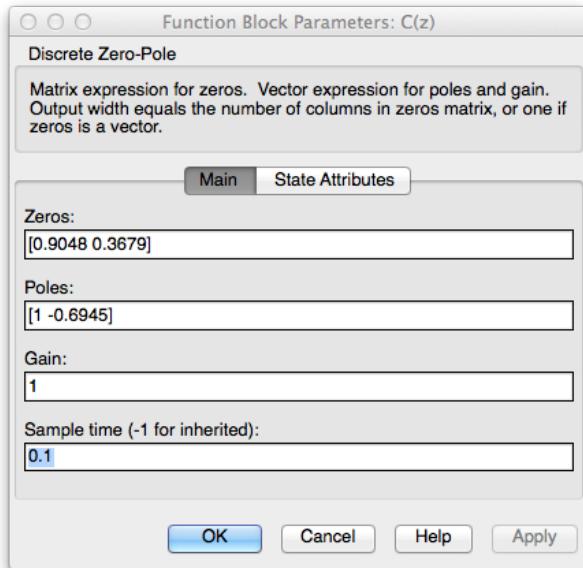
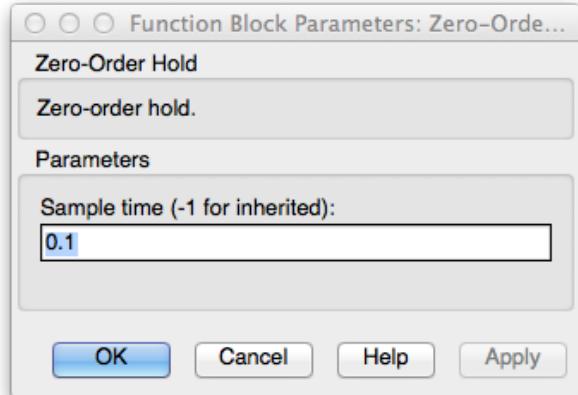
Note que os blocos relacionados com  $C(z)$  (*Discrete Zero Pole Transfer Fcn* no Simulink) e o sustentador de ordem zero devem explicitamente terem seus parâmetros editados para confirmar que são discretos, isto é, com período de amostragem definido em  $T = 0.1$  – ver figuras 7.9(a) e (b).

E também devemos ajustar os parâmetros de simulação no Simulink. Selecione **Simulation > Configuration Parameters** e ajuste o campo **Solver options > Type: Fixed-step, Solver: ode4 (Runge-Kutta)** (conforme sugestão de (Nise, 2012)) e **Fixed step size:** para 0.1 – conforme mostra a figura 7.10.

Uma vez feito os ajustes no Simulink é possível visualizar os resultados obtidos através dos comandos a seguir, que geram os gráficos da figura 7.11(a) e (b):

```
>> figure; plot(t,y)
>> title('Ação de Controle [u(k)]')
>> xlabel('tempo (s)');
>> ylabel('u(k)')
>> figure; stairs(t,u)
>> title('Ação de Controle [u(k)]')
>> xlabel('tempo (s)');
>> ylabel('u(k)')
```

Observando a figura 7.11(b) se percebe que o sinal de controle oscilou entre  $27.0489 \leq u(k) \leq 28.0689$  (foram usados os comandos `> max(u)` e `> min(u)` para descobrir os maiores valores de oscilação de  $u(k)$ ) – uma faixa de excursão provavelmente muito maior que a suportada pelo conversor D/A ou driver de potência conectado fisicamente entre a saída do controlador digital e a entrada da planta.

(a) Parâmetros de  $C(z)$ .

(b) Parâmetros do ZOH.

Figura 7.9: Ajuste de parâmetros de blocos discretos no Simulink.

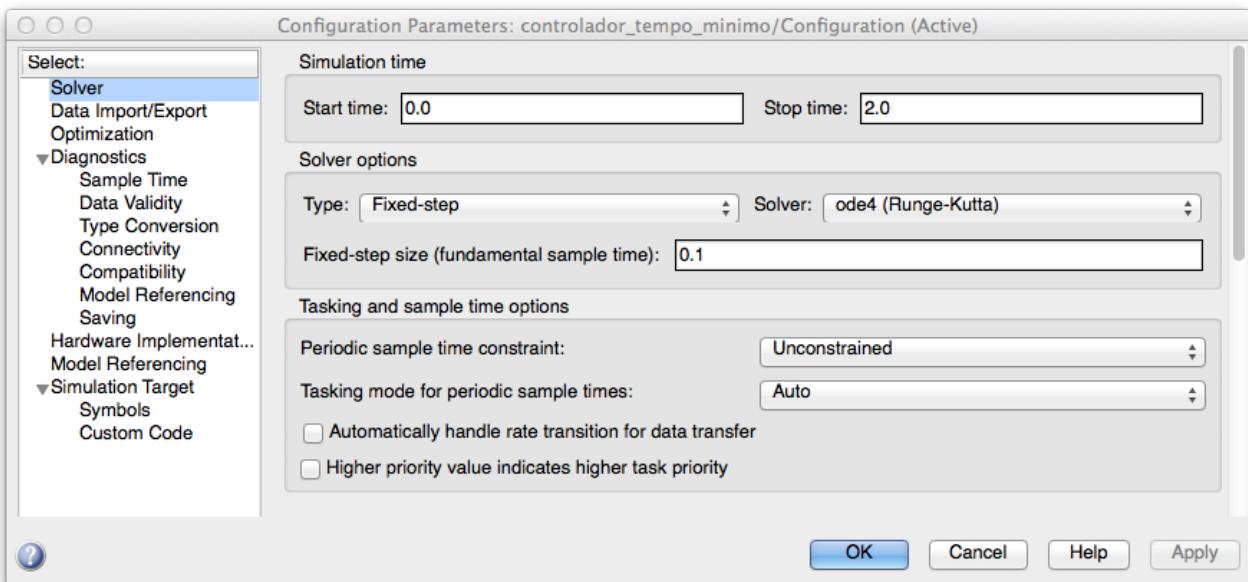


Figura 7.10: Parâmetros de Simulação para o controlador dead-beat.

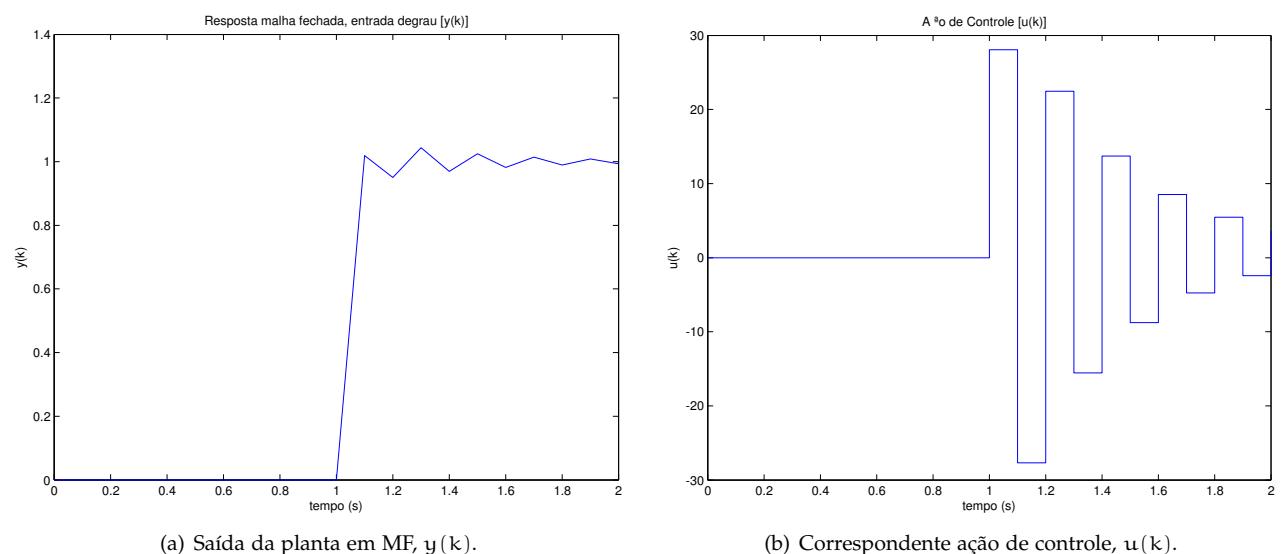


Figura 7.11: Saída do sistema e correspondente ação de controle para o controlador dead-beat.



# Capítulo 8

## Projeto por Emulação

### 8.1 Introdução

9\_bilinear/  
trans\_bilinear\_  
projetos.tex

Um sistema contendo tanto um sinal contínuo no tempo quanto um sinal discreto é dito um sistema de dados amostrado [Franklin et al. (1994), Cap. 8: Digital Control].

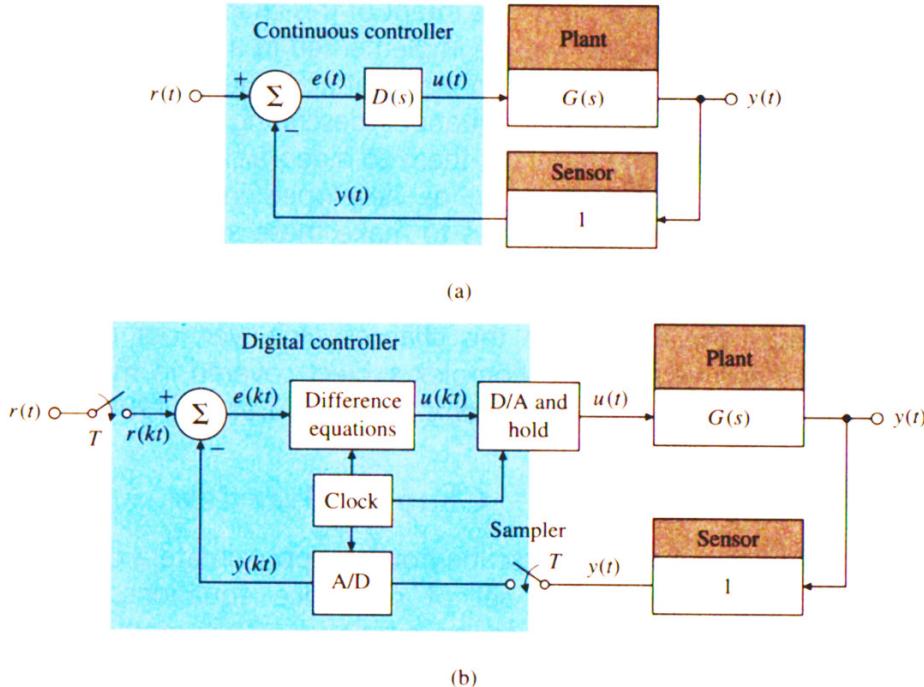
Assumimos que o período de amostragem é fixo. Na prática, sistemas de controle digitais algumas vezes possuem períodos de amostragem variáveis e/ou diferentes períodos de amostragem entre seus ramos de realimentação. Normalmente, a lógica do computador inclui um sinal de **clock** que fornece um pulso, ou **interrupção**, a cada  $T$  segundos, um conversor A/D envia uma sequencia numérica binária para o computador a cada instante de tempo que a interrupção ocorre. Outra alternativa de implementação, as vezes referenciada como **free-running** consiste em acessar o conversor A/D depois que cada ciclo de execução do código foi completada. No primeiro caso, o período de amostragem é preciso e fixo. No último caso, o período de amostragem depende do tamanho do código à ser executado, o qual eventuais laços de repetição e desvios condicionais (IFs) devem ser evitados, o que caso contrário, poderia fazer variar a quantidade de código à ser executada, fazendo variar o período de amostragem.

Também num sistema de controle digital deve estar presente um *sampler* (amostrador) e um conversor A/D para a entrada de comando (ou referência),  $r(t)$ , o que gera o sinal discretizado  $r(kT)$  que juntamente com o saída amostrada do sistema,  $y(kT)$  gera o sinal de erro discretizado:  $e(kT) = r(kT) - y(kT)$  – ver figura 8.1.

Um sistema contínuo de compensação pode ser aproximado por uma equação de diferenças que seria uma versão discreta de uma equação diferencial e pode ser adotado para emular o comportamento dinâmico de um compensador contínuo,  $C(s)$ , se o período de amostragem for curto o suficiente. O resultado desta equação de diferenças seria um sinal discreto  $u(kT)$  a cada instante de amostragem. Este sinal seria convertido para o mundo contínuo,  $u(t)$  através de um conversor D/A e um sustentador (*Holder*). O conversor D/A converte o número binário numa tensão analógica e o sustentador manteria a última tensão analógica gerada durante os instantes de amostragem.

Existem duas técnicas básicas para determinar a equação de diferenças que define o controlador digital. Uma delas, chamada **emulação**, consistem em projetar o compensador contínuo no tempo,  $C(s)$  usando métodos clássicos de controle automático e então realizar uma aproximação de  $C(s)$  para uma representação discreta,  $C(z)$  usando um dos seguintes métodos:

- Método de Euler;
- Método de Tustin;
- Método da Correspondência Pólo-Zero (MPZ = *Matched Pole-Zero*).



Obs.: Extraído de [Franklin et al. (1994)].

Figura 8.1: Diagrama de blocos para um sistema básico de controle: (a) sistema contínuo e (b) incluindo um computador digital.

## 8.2 Método da Correspondência Pólo-Zero (ou MPZ)

Este método, o da correspondência pólo-zero leva em conta a definição da transformada  $\mathcal{Z}$ , extrapolando então a relação entre os planos  $s$  e  $z$ , obedecendo a relação:

$$z = e^{sT} \quad (\text{Pela definição})$$

como:  $s = \alpha + j\omega$ , então teremos:

$$z = e^{T(\alpha+j\omega)} = e^{\alpha T} e^{j\omega T}$$

$$z = e^{\alpha T} (\cos \omega T + j \sin \omega T)$$

$$z = e^{\alpha T} \angle \omega T$$

uma vez que  $(\cos \omega T + j \sin \omega T) = 1 \angle \omega T$ .

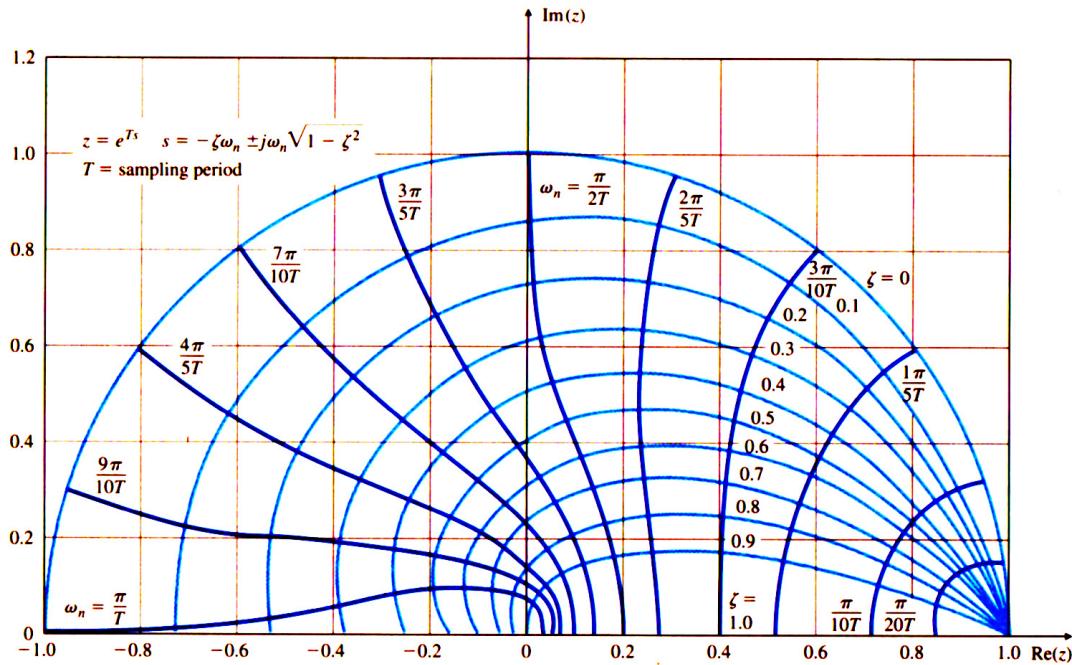
Desta forma, cada região do plano-s pode ser mapeada na correspondente região do plano-z.

Se tomamos a transformada  $\mathcal{Z}$  da função amostrada  $x(k)$ , então os pólos de  $X(z)$  estão relacionados com os pólos de  $X(s)$  obedecendo a relação  $z = e^{sT}$ . O problema é que temos que considerar também que  $s = (1/T) \ln z$  é a inversa de  $z = e^{sT}$ . Sendo assim, se temos  $P(z)$  e substituímos  $z = e^{Ts}$ , obtemos  $P(e^{Ts})$  como resultado da conversão para o plano-s. De modo similar, se temos  $P(s)$  e substituímos  $s = (1/T) \ln z$ , obtemos  $P[(1/T) \ln z]$  como resultado da conversão para o plano-z. Infelizmente, ambas transformações podem ser levadas à caso, respeitando detalhes da complicada transformada mas conduzem a funções trancendentais [Nise (2012), Cap. 13: Digital Control Systems].

### 8.2.1 Relações entre o plano-s e o plano-z

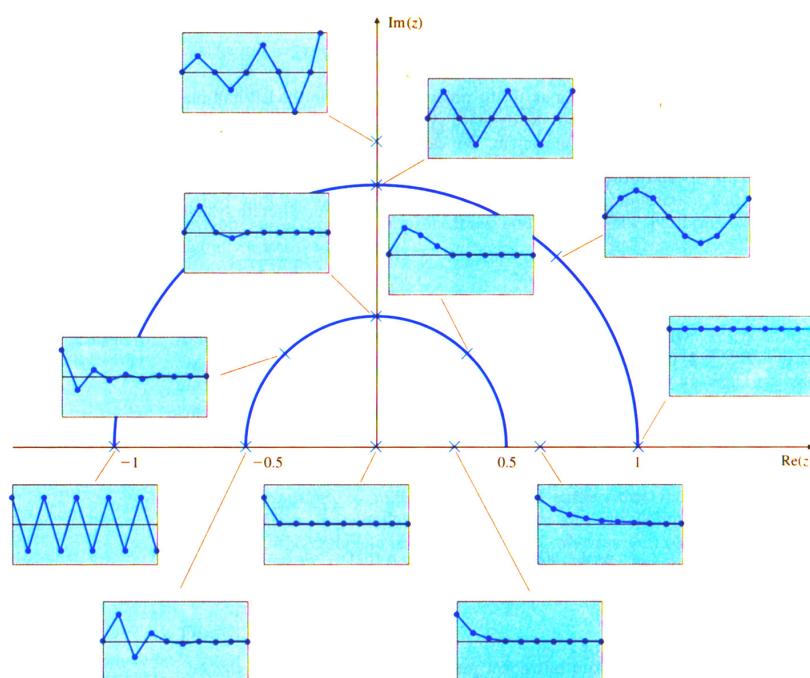
A figura 8.2 mostra as linhas de mapeamento resultantes para valores constantes de  $\zeta$  e para valores constantes de  $w_n$  baseado na relação:  $z = e^{sT}$ .

E a figura 8.3 mostra diferentes respostas no tempo resultantes para diferentes posições para um pólo em malha fechada.



Obs.: Extraído de [Franklin et al. (1994), Cap. 8: Digital Control].

Figura 8.2: Frequencia natural ( $\omega_n$ ) e fator de amortecimento ( $\zeta$ ) no plano- $\mathcal{Z}$ .



Obs.: Extraído de [Franklin et al. (1994), Cap. 8: Digital Control].

Figura 8.3: Sequencias de tempo associadas com polos no plano- $\mathcal{Z}$ .

## 8.3 Método de Euler

Uma maneira particularmente simples de fazer com que um computador digital se aproxime da solução em tempo-real de uma equação diferencial é usar o método de Euler (também conhecido como regra retangular direta – *forward rectangular rule*). Ela é resultado da seguinte aproximação:

$$\frac{dx}{dt} = \lim_{\Delta t \rightarrow 0} \frac{\Delta x}{\Delta t}$$

que aplicada neste caso em particular resulta em:

$$\dot{x} \approx \frac{x(k) - x(k-1)}{T}$$

onde:

- $T$  = corresponde ao período de amostragem (em segundos),  $T = t_k - t_{k-1}$ ,
- $t_k$  =  $kT$ ,
- $x(k)$  = valor de  $x$  no instante  $t_k$ , ou  $x(kT)$
- $x(k-1)$  = valor de  $x$  no instante  $t_{k-1}$ , ou  $x[(k-1)T]$

Esta aproximação pode ser usada no lugar de várias derivadas que apareçam na equação diferencial do controlador e resulta num conjunto de equações que pode ser calculada por um computador digital. Estas *equações de diferenças* são resolvidas repetidamente a cada  $T$  período de tempo. Para sistemas com margem de banda da ordem de uns poucos Hertz, taxas de amostragem resultam normalmente na ordem de 1 KHz, e assim os períodos de amostragem ficarão em torno de 1 ms e os erros envolvidos na aproximação resultarão bastante pequenos [Franklin et al. (1994)].

### 8.3.1 Exemplo<sub>1</sub>: Caso de um Controlador de 1a-ordem

Usando o método de Euler, encontre a equação de diferenças que necessita ser programada num computador se  $C(s)$  é dado como:

$$C(s) = \frac{U(s)}{E(s)} = K_o \frac{s + a}{s + b}$$

**Solução:** Primeiro determinamos a equação diferencial que corresponde à  $C(s)$ :

$$(s + b)U(s) = K_o(s + a)E(s)$$

por inspeção percebemos que a correspondente equação diferencial é:

$$\dot{u} + bu = K_o(\dot{e} + ae) \quad (8.1)$$

Usando o método de Euler para aproximar a eq. (8.1) obtermos a seguinte equação de diferenças:

$$\frac{u(k) - u(k-1)}{T} + bu(k) = K_o \left[ \frac{e(k) - e(k-1)}{T} + ae(k) \right] \quad (8.2)$$

Rearrumando a eq. (8.2) obtemos:

$$u(k) = u(k-1) + T \left\{ -bu(k) + K_o \left[ \frac{e(k) - e(k-1)}{T} + ae(k) \right] \right\} \quad (8.3)$$

A equação (8.3) determina o novo valor de controle,  $u(k)$ , uma vez conhecida a amostra passada do sinal de controle,  $u(k-1)$  e os novos e valores passados do sinal de erro,  $e(k)$  e  $e(k-1)$ .

### 8.3.2 Exemplo<sub>2</sub>: Caso de um Controlador por Avanço de Fase

Encontre uma implementação digital para compensador de avanço de fase definido para a seguinte planta(Exemplo 5.13 de [Franklin et al. (1994)]):

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

usando taxas de amostragem de: a) 20 e b) 40 Hz. Simule as equações de controle para uma entrada degrau e compare o resultado obtido entre o controlador analógico e o digital. O sistema deve responder em malha-fechada com fator de amortecimento  $\zeta > 0,5$  e com frequência natural  $w_n > 7$  (rad/s).

Na ocasião deste exercício, diferentes versões para o compensador de avanço de fase (testes de pares polo-zero) foram realizadas, como mostra a figura 8.4.

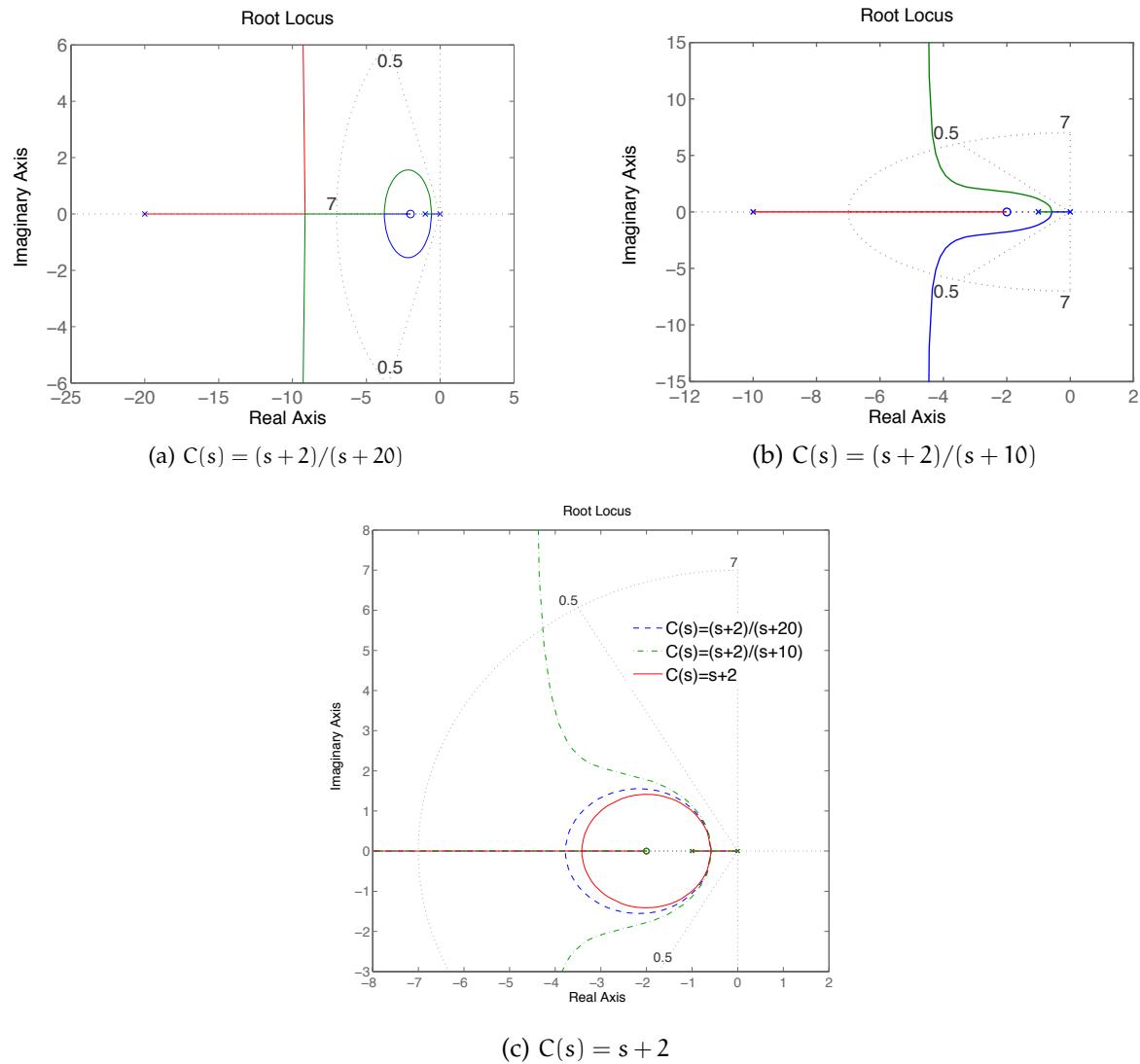


Figura 8.4: Lugar das Raízes para  $G(s) = 1/[s(s+1)]$  e 3 casos de  $C(s)$ .

A função transferência em malha aberta resulta em:

$$\text{FTMA}(s) = C(s)G(s) = \frac{70(s+2)}{s(s+10)(s+1)}$$

onde  $K_o = 70$ . Este ganho resulta na resposta ao degrau mostrada na figura 8.5.

**Solução:** a função transferência do compensador de avanço de fase para esta planta fica (baseado no exemplo 5.13 de [Franklin et al. (1994)]):

$$C(s) = 70 \frac{s+2}{s+10}$$

Desta forma, os parâmetros para a eq. (8.3) são  $a = 2$ ,  $b = 10$  e  $K_o = 70$ .

a) Para a taxa de amostragem de 20 Hz,  $T = 0,05$  (segundos) e a eq. (8.3) pode ser simplificada para:

$$u(k) = 0,5u(k-1) + 70[e(k) - 0,9e(k-1)]$$

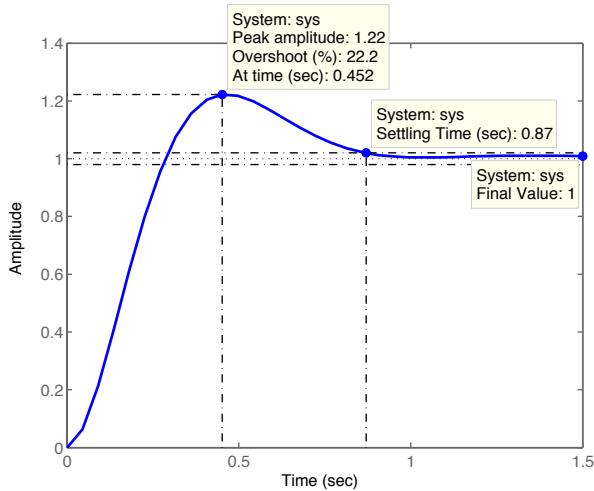


Figura 8.5: Resposta ao degrau para  $G(s) = 1/[s(s + 1)]$  e  $C(s) = 70(s + 2)/(s + 10)$ .

Colocando no formato de  $C(z)$  temos:

$$U(z) = 0,5z^{-1}U(z) + 70[E(z) - 0,9z^{-1}E(z)]$$

trabalhando a expressão anterior obtemos:

$$U(z)[1 - 0,5z^{-1}] = 70E(z)[1 - 0,9z^{-1}]$$

$$C(z) = \frac{U(z)}{E(z)} = \frac{70(1 - 0,9z^{-1})}{(1 - 0,5z^{-1})} = \frac{70(z - 0,9)}{(z - 0,5)}$$

b) Para a taxa de amostragem de 40 Hz,  $T = 0,025$  (segundos), a eq. (8.3) simplificada resulta em:

$$u(k) = 0,75u(k-1) + 70[e(k) - 0,95e(k-1)]$$

Colocando no formato de  $C(z)$  obtemos:

$$C(z) = \frac{U(z)}{E(z)} = \frac{70(1 - 0,95z^{-1})}{[1 - 0,75z^{-1}]} = \frac{70(z - 0,95)}{(z - 0,75)}$$

O código no MATLAB para calcular a resposta contínua ao degrau fica:

```
>> num=70*[1 2]
>> den=conv([1 1 0],[1 10])
>> [numcl,dencl]=feedback(num,den,1,1)
>> step (numcl,dencl)
```

Que resulta no gráfico mostrado na figura 8.5.

A figura 8.6 mostra o diagrama em blocos do sistema simulado usando tanto o compensador contínuo no tempo (a) quanto o simulador discreto (b).

As respostas ao degrau para (a)  $T=0.05$  e (b)  $T=0.025$  aparecem na figura 8.7.

**Conclusão:** Este método (de Euler) é mais adequado quanto menor for o período de amostragem,  $T$ . Para grandes valores de  $T$ , este método pode resultar em instabilidade, mesmo quando o sistema sendo discretizado é estável no tempo contínuo.

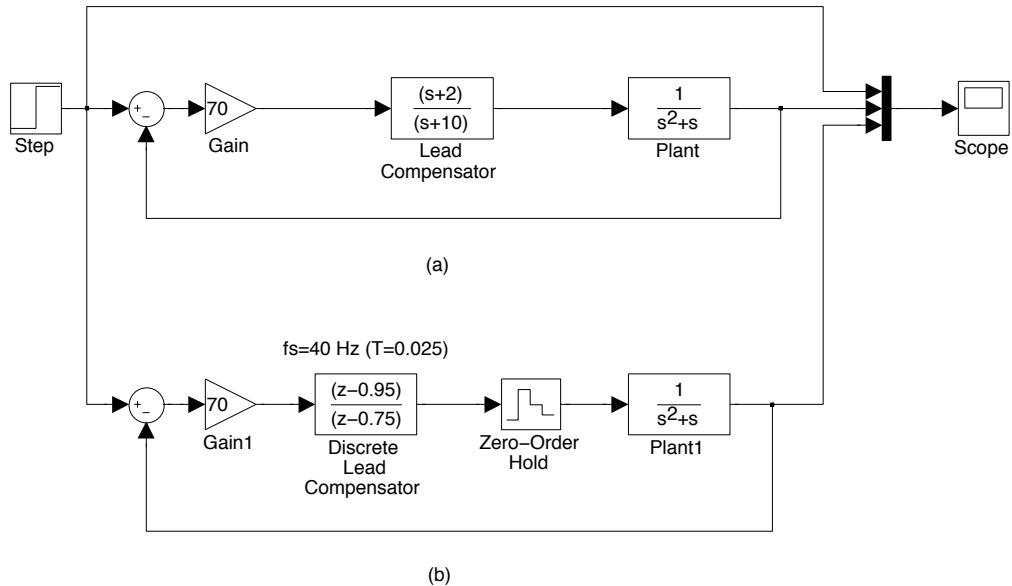


Figura 8.6: Simulação do compensador do exemplo 8.3.2.

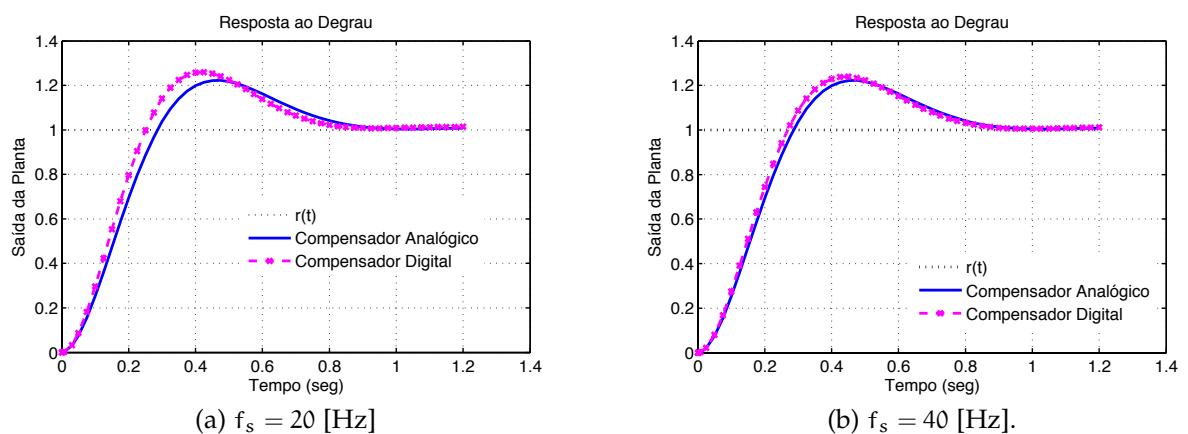


Figura 8.7: Respostas obtidos para o método de discretização de Euler.

## 8.4 Método de Tustin

Outro método de discretização é tratar o problema como uma aproximação (integração) numérica. Suponha que:

$$\frac{U(s)}{E(s)} = C(s) = \frac{1}{s}$$

o qual é uma integração. Assim sendo:

$$u(kT) = \int_0^{kT-T} e(t)dt + \int_{kT-T}^{kT} e(t)dt$$

pode ser reescrito como:

$$u(kT) = u(kT-T) + \text{área entre } e(t) \text{ sobre a última } T \quad (8.4)$$

O método de Tustin realiza **integração numérica trapezoidal** a cada passo do período de amostragem, tentando aproximar  $e(t)$  através de uma linha direta entre dois instantes de amostragem. Reescrevendo  $u(kT)$  como  $u(k)$  e  $u(kT-T)$  como  $u(k-1)$  para simplificar a escrita, transformamos a equação (8.4) em:

$$u(k) = u(k-1) + \frac{T}{2} [e(k-1) + e(k)]$$

tomando a transformada- $\mathcal{Z}$  da expressão anterior, obtemos:

$$\frac{U(z)}{E(z)} = \frac{T}{2} \left( \frac{1+z^{-1}}{1-z^{-1}} \right) = \frac{1}{\frac{T}{2} \left( \frac{1-z^{-1}}{1+z^{-1}} \right)}$$

Para  $C(s) = a/(s+a)$ , aplicando esta aproximação para a integração resulta em:

$$D(z) = \frac{\frac{a}{2}}{\frac{T}{2} \left( \frac{1-z^{-1}}{1+z^{-1}} \right) + a}$$

Note que se substituirmos:

$$s = \frac{2}{T} \left( \frac{1-z^{-1}}{1+z^{-1}} \right) \quad (8.5)$$

para cada ocorrência de  $s$  em qualquer  $C(s)$ , obtemos  $C(z)$  baseado na equação de integração trapezoidal.

Este é o chamado método de **transformação bilinear de Tustin** [Franklin et al. (1994)]. A idéia é a de que nos instantes amostrados, a transformação bilinear seja capaz de preservar o mesmo tipo (valor) da resposta de um controlador contínuo no tempo. Em resumo este tipo de transformação é usado para **transformar um compensador contínuo no tempo**,  $C(s)$  na sua versão digital,  $C(z)$ . Explicitamente a transformação bilinear de Tustin é dada por:

$$s = \frac{2}{T} \frac{(z-1)}{(z+1)} \quad (8.6)$$

e sua inversa é dada por:

$$\begin{aligned} z &= \frac{-\left(s + \frac{2}{T}\right)}{\left(s - \frac{2}{T}\right)} \\ &= \frac{1 + \frac{T}{2}s}{1 - \frac{T}{2}s} \end{aligned} \quad (8.7)$$

Quanto menor for o intervalo de amostragem,  $T$  (ou maior a frequência de amostragem), mais a versão discretizada se aproxima do comportamento no domínio-frequência do compensador no domínio contínuo de tempo. A eq. (8.6) é a que entre os métodos de integração numérica, permite levar a um resultado mais próximo da resposta (em tempo e frequência) que realmente seria obtido usando o compensador no domínio

contínuo de tempo.

Åström and Wittenmark (1990), propuseram uma referência para **selecionar o período de amostragem à ser adotado**. Eles concluíram que o valor de  $T$  em segundos deve estar na faixa de  $0,15/\omega_{\Phi M}$  até  $0,5/\omega_{\Phi M}$ , onde  $\omega_{\Phi M}$  corresponde a frequência (em rad/s) na qual o diagrama de magnitude da função transferência analógica do compensador mais planta passa por 0 dB.

#### 8.4.1 Exemplo 1: Equação de diferenças usando o Método de Tustin

Voltando ao exemplo da seção 8.3.2 e usando taxa de amostragem de  $f_s = 15$  [Hz], podemos usar o MATLAB para computar a aproximação de Tustin.

**Solução** A forma contínua do compensador estudado na seção 8.3.2 era:

$$C(s) = 70 \frac{(s+2)}{(s+10)}$$

Usando declarações na linha de comandos do MATLAB, podemos fazer:

```
>> numc=70*[1 2];
>> denc=[1 10];
>> T=1/15;
>> C=tf(numc,denc);
>> zpk(C)
Zero/pole/gain:
70 (s+2)
_____
(s+10)
>> Cd=c2d(C,T,'tustin')
Transfer function:
56 z - 49
_____
z - 0.5
Sampling time: 0.066667
>> zpk(Cd)
Zero/pole/gain:
56 (z-0.875)
_____
(z-0.5)
Sampling time: 0.066667
>>
```

que produz:

$$C(z) = \frac{56 - 49z^{-1}}{1 - 0,5z^{-1}}$$

que pode ser transscrito para uma equação de diferenças resultando em:

$$u(k) = 0,223u(k-1) + 70e(k) - 59,12e(k-1)$$

que é similar a obtida usando o método de Euler.

**Problema:** Comparar a resposta deste controlador usando o método de Tustin com os resultados obtidos quando o mesmo compensador contínuo no tempo foi discretizado usando o método de Euler (seção 8.3.2, pág. 70). Para os mesmos 2 valores de período de amostragem: a)  $f_s = 20$  (Hz) e b)  $f_s = 40$  (Hz).

### 8.4.2 Exemplo 2: Projeto de Controlador Digital usando o Método de Tustin

Seja um sistema (planta) descrito pela equação:

$$G(s) = \frac{1}{s(s+6)(s+10)} \quad (8.8)$$

onde o projeto de um compensador por avanço resultou num sistema compensado que opera com sobre sinal máximo de 20% e um tempo de acomodação de 1,1 segundos. Defina a equação de diferenças que corresponde ao compensador contínuo transformado para digital usando o método de Tustin. A equação do compensador é dada por:

$$C(s) = \frac{1977(s+6)}{(s+29,1)} \quad (8.9)$$

#### Solução:

Com base na eq. (8.8) e (8.9) se percebe que a frequência que corresponde à 0 dB,  $\omega_{\Phi M}$ , para  $C(s)G(s)$ , é de 5,8 rad/s. Com base na linha guia proposta por Åström and Wittenmark (1990), o menor valor recomendado para T deveria estar na faixa de  $0,15/\omega_{\Phi M} = 0,026$  até  $0,5/\omega_{\Phi M} = 0,086$  segundos. Iremos adotar o valor de  $T = 0,01$  segundos.

Substituindo a eq. (8.6) na eq. (8.9) com  $T = 0,01$  segundos, leva à:

$$C(z) = \frac{1778z - 1674}{z - 0,746} \quad (8.10)$$

e a função transferência da planta usando sustentador de ordem zero (ZOH), com  $T = 0,01$  segundos resulta em:

$$\frac{(1,602 \times 10^{-7}z^2) + (6,156 \times 10^{-7}z)(1,478 \times 10^{-7})}{z^3 - 2,847z^2 + 2,699z - 0,8521} \quad (8.11)$$

As equações anteriores podem ser obtidas usando o seguinte código no MATLAB:

```
>> num=1;
>> den=poly([0 -6 -10]);
>> G=tf(num,den);
>> zpk(G)
1

s (s+10) (s+6)
Continuous-time zero/pole/gain model.
>> num_c=1977*[1 6];
>> den_c=[1 29.1];
>> C=tf(num_c,den_c);
>> zpk(C)
1977 (s+6)

(s+29.1)
Continuous-time zero/pole/gain model.
>> %convertendo para controle digital
>> T=0.01;
>> help c2d
c2d Converts continuous-time dynamic system to discrete time.

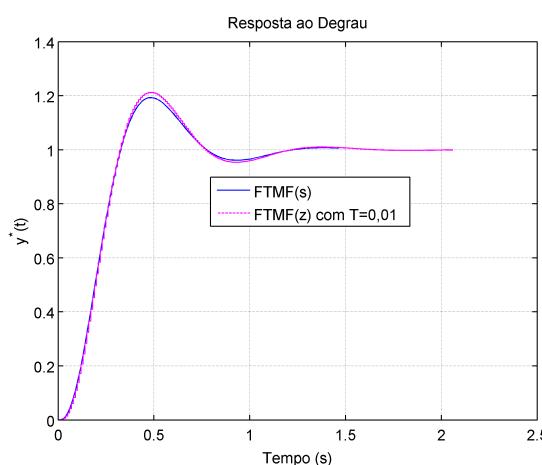
SYSD = c2d(SYSC,TS,METHOD) computes a discrete-time model SYSD with
sampling time TS that approximates the continuous-time model SYSC.
The string METHOD selects the discretization method among the following:
'zoh'      Zero-order hold on the inputs
'foh'      Linear interpolation of inputs
'impulse'  Impulse-invariant discretization
'tustin'   Bilinear (Tustin) approximation.
'matched'  Matched pole-zero method (for SISO systems only).
The default is 'zoh' when METHOD is omitted. The sampling time TS should
be specified in the time units of SYSC (see "TimeUnit" property).
```

```

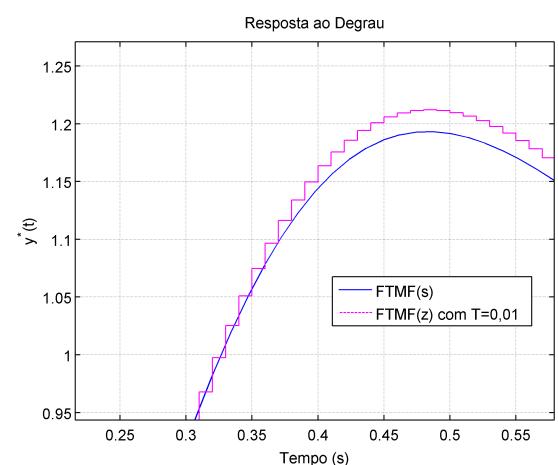
>> Cd=c2d(C,T,'tustin');
>> zpk(Cd)
  1777.7 (z-0.9417)
  _____
  (z-0.746)
Sample time: 0.01 seconds
Discrete-time zero/pole/gain model.
>> %convertendo a planta para o mundo digital usando ZOH:
>> Gd=c2d(G,T)
Gd =
  1.602e-07 z^2 + 6.156e-07 z + 1.478e-07
  _____
  z^3 - 2.847 z^2 + 2.699 z - 0.8521
Sample time: 0.01 seconds
Discrete-time transfer function.
>> zpk(Gd)
  1.6016e-07 (z+3.586) (z+0.2574)
  _____
  (z-1) (z-0.9418) (z-0.9048)
Sample time: 0.01 seconds
Discrete-time zero/pole/gain model.
>> ftma=series(C,G);      %FTMA(s)
>> ftmad=series(Cd,Gd);   %FTMA(z) com T=0,01 (s)
>> ftmf=feedback(ftma,1); %FTMF(s)
>> ftmfd=feedback(ftmad,1);%FTMF(z)
>> [y, t]=step(ftmf);    %resposta degrau, sistema contínuo
>> plot(t,y)
>> [yd1, td1]=step(ftmfd);
>> hold on;
>> stairs(td1,yd1, 'm-')
>> legend('FTMF(s)', 'FTMF(z) com T=0,01');
>> title('Resposta ao Degrau');
>> xlabel('Tempo (s)');
>> ylabel('y*(t)');
>> grid

```

A resposta para entrada degrau para o sistema compensando é mostrada na figura 8.8, que compara as respostas do compensador contínuo no tempo versus compensadores digitais usando  $T = 0,086$  (s),  $T = 0,026$  (s) e  $T = 0,01$  (s).



(a) Resposta ao degrau.



(b) "Zoom" sobre o gráfico (a).

Figura 8.8: Respostas em malha-fechada para o exemplo 2.

## 8.5 Outras Transformações Bilineares

Note que o método de Tustin trata-se de um caso específico da aplicação de um método de transformação bilinear do tipo:

$$z = \frac{as + b}{cs + d}$$

e sua inversa:

$$s = \frac{-dz + b}{cz - a}$$

A idéia é sempre buscar uma simples transformação capaz de atuar em ambos os lados (plano-s e plano-z) através de algum método de substituição direta [Nise (2012), Cap. 13: Digital Control Systems].

Por exemplo, a transformação bilinear:

$$z = \frac{s+1}{s-1} \quad \text{e sua inversa:} \quad s = \frac{z+1}{z-1}$$

é proposta por Kuo (1992) para aproximar raízes presentes no interior do círculo unitário  $|z| = 1$  para o semi-plano esquerdo do plano-s. No caso, esta transformação recebeu o nome de *transformação-r*.

Outro tipo de transformação bilinear é a *transformada-w* onde:

$$\omega = \frac{1}{\ln(z)}$$

que tenta transformar o círculo unitário do plano-z para o eixo imaginário de outro plano complexo  $\omega$  [Kar and Majhi]. Neste caso:

$$\omega = \frac{2}{T} \frac{(z-1)}{(z+1)} \quad (8.12)$$

e su inversa:

$$\begin{aligned} z &= \frac{\frac{2}{T} + \omega}{\frac{2}{T} - \omega} \\ &= \frac{1 + \frac{\omega T}{2}}{1 - \frac{\omega T}{2}} \end{aligned} \quad (8.13)$$

que é uma variação da proposta originária de [Ogata (1995), Cap. 4: Design of Discrete-Time Control Systems by Conventional Methods]:

$$z = \frac{w+1}{w-1} \quad \text{e sua inversa:} \quad w = \frac{z+1}{z-1}$$

Em comum as últimas 2 transformadas permitem realizar todo um projeto de controlador discreto usando o mapeamento em frequencia, tal qual se fazia usando Diagramas de Bode quando estávamos no plano-s (ou no mundo contínuo). Neste último caso, estas transformadas permitem converter BoG(z) em BoG(w) e lidar com o projeto do controlador tal qual se fazia antes usando Diagrama de Bode – o que muda agora é que ao invés do Diagrama de Bode ser feito no plano-s é realizado sobre o plano-w.

# Capítulo 9

## PID Digital

Este capítulo está organizado da seguinte forma:

1. Formas de Implementação para o PID “digital”
  - a) Formato de posição
  - b) Formato de velocidade
2. Métodos de Sintonia para o controlador PID
  - a) Método de Ziegler-Nichols (entrada degrau);
  - b) Método do Relé;
  - c) Método de Cohen-Coon;
  - d) Outros Métodos.
3. Acrescentando o Filtro Derivativo
4. Acrescentando característica anti-windup

### 9.1 Formas de Implementação

O PID no domínio de sinais contínuos no tempo pode ser descrito como (Seborg et al. (1989)):

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt} \quad (9.1)$$

A versão digital (adotando integração retangular) que pode ser adotada para implementar este algoritmo aparece na equação abaixo:

$$u[k] = K_p e[k] + K_i \sum_{k=0}^n e[k]T + K_d \frac{(e[k] - e[k-1])}{T} \quad (9.2)$$

que também pode ser implementado na forma:

$$u[k] = K_c \left[ e[k] + \frac{T}{\tau_i} \sum_{k=0}^n e[k] + \frac{\tau_d}{T} (e[k] - e[k-1]) \right] \quad (9.3)$$

Esta implementação se baseia no modo direto de derivação da lei de controle do PID da equação (9.1). A integral foi aproximada via um somatório e a componente derivativa foi obtida através de uma equação de diferenças de primeira ordem. Estas formas de implementações também são conhecidas como **PID no formato de posição** (Seborg et al. (1989)).

Existem outras formas ainda de implementar algoritmos de PID digitais, entre elas o **formato de velocidade** (Seborg et al. (1989)) e o PID com 2 graus de liberdade (ou PID 2 DOF).

O PID no formato de velocidade passa a ser (Seborg et al. (1989)):

$$u[k] = u[k-1] + K_c \left[ (e[k] - e[k-1]) + \frac{T}{\tau_i} e[k] + \frac{\tau_d}{T} (e[k] - 2e[k-1] + e[k-2]) \right] \quad (9.4)$$

ou (Nascimento Jr. and Yoneyama (2000)):

$$u[k] = u[k-1] + a_0 e[k] + a_1 e[k-1] + a_2 e[k-2] \quad (9.5)$$

onde:

$$\begin{aligned} a_0 &= K_p + K_i T + \frac{K_d}{T} \\ a_1 &= -\left( K_p + 2 \frac{K_d}{T} \right) \\ a_2 &= \frac{K_d}{T} \end{aligned}$$

O formato de velocidade para implementação do PID traz com vantagens:

1. Este formato inerentemente leva em conta uma provisão para reset da ação integral uma vez que o somatório de erros não é explicitamente calculado;
2. O sinal de controle  $\Delta u[k] = u[k] - u[k-1]$  já está numa forma que pode ser diretamente utilizado para controlador elementos que exigem mudanças no sinal de controle, como válvulas acionadas por motor de passo;
3. Este formato não requer inicialização de parâmetros de entrada (como  $e[k] = 0$ , para  $k \leq 0$  e  $\sum_{i=1}^k = 0$ ), como é necessário na eq. (9.2); principalmente em situações em chaveamento de operação de um planta saindo do modo manual para automático.

### 9.1.1 Formas de Integração Numérica

A ação de controle no formato digital pode ser aproximado numericamente de 2 formas:

1. Integração Retangular;
2. Integração Trapezoidal.

#### Integração Retangular

Neste caso o integrador pode ser sintetizado numericamente como:

$$\begin{aligned} i[0] &= 0 \\ i[1] &= i[0] + e[0] \cdot T \\ i[2] &= i[1] + e[1] \cdot T \\ &\vdots \\ i[k] &= i[k-1] + e[k-1] \cdot T \\ &\downarrow \\ I(z) &= z^{-1} I(z) + T \cdot z^{-1} \cdot E(z) \\ I(z) &= \frac{Tz^{-1}}{(1-z^{-1})} \cdot E(z) \\ \text{ou} \\ I(z) &= \frac{T}{(z-1)} \cdot E(z) \end{aligned}$$

### Integração Trapezoidal

No caso de se optar por integração trapezoidal, temos que nos recordar primeiramente da área de um trapézio:

$$A = \frac{(a + b)}{2} \cdot h \quad (9.6)$$

Levando em conta a eq. (9.6), a ação integral pode ser sintetizada como:

$$\begin{aligned} i[0] &= 0 \\ i[1] &= i[0] + T \cdot \left( \frac{e[0] + e[1]}{2} \right) \\ i[2] &= i[1] + T \cdot \left( \frac{e[1] + e[2]}{2} \right) \\ &\vdots \\ i[k] &= i[k-1] + T \cdot \left( \frac{e[k-1] + e[k]}{2} \right) \\ \downarrow \\ I(z) &= z^{-1} I(z) + T \cdot \left( \frac{z^{-1} E(z) + E(z)}{2} \right) \\ I(z) &= \frac{T}{2} \cdot \frac{(1+z^{-1})}{(1-z^{-1})} \cdot E(z) \\ \text{ou} \\ I(z) &= \frac{T}{2} \cdot \frac{(z+1)}{(z-1)} \cdot E(z) \end{aligned}$$

### 9.1.2 Formas de Diferenciação Numérica

Uma diferenciação numérica pode ser facilmente obtida através de:

$$\begin{aligned} \frac{\partial e(t)}{\partial t} &\cong \frac{e[k] - e[k-1]}{T} \\ \downarrow \\ D(z) &= \frac{(1-z^{-1})}{T} \cdot E(z) \\ \text{ou} \\ D(z) &= \frac{(z-1)}{Tz} \cdot E(z) \end{aligned}$$

O problema (inerente à qualquer diferenciação) é que a equação anterior ressaltará sobremaneira a presença de ruídos presentes na malha de realimentação. Motivo pelo qual, acrescenta-se à solução anterior um filtro derivativo que não é mais que a aproximação numérica de um filtro passa-baixa de 1a-ordem. Este filtro será estudado mais adiante.

### 9.1.3 Formas de Implementação das Ações Proporcional-Integral-Derivativa

Incluir figura mostrando formato paralelo para implementação preferencial do controlador PID

## 9.2 Ajuste de Controladores PID

O ajuste dos controladores foi realizado baseado no método de Ziegler-Nichols (Seborg et al. (1989); ?; ?), conforme ilustra a tabela 9.1 à seguir. Onde  $K_u$  refere-se ao ganho limite, máximo ou "ultimate gain" que pode ser aplicado num sistema e que torna a saída da planta oscilatória (com amplitude constante ainda). O período no qual o sistema começa a oscilar é caracterizado como  $T_u$  (ou "ultimate period").

Se for utilizado um controlador Proporcional puro com ganho  $K_p$  maior que  $K_u$  o sistema se torna instável. Se este ganho,  $K_p < K_u$  a saída do sistema tende a ser sub-amortecida.

Controlador	$K_c$	$\tau_i$	$\tau_d$
P	$0.5 K_u$	—	—
PI	$0.45 K_u$	$T_u/1.2$	—
PID	$0.6 K_u$	$T_u/2$	$T_u/8$

Tabela 9.1: Relações de Ziegler-Nichols para ajuste de controladores PID.

**Exemplo:** Seja a planta:

$$G(s) = \frac{1}{(s+1)(s+2)(s+10)}$$

sintonizar o controlador PID, formato velocidade para a mesma, usando  $T = 0, 1$ .

### Solução

## 9.3 Acresentando o Filtro Derivativo

Pode ser obtido de maneira simples através da equação:

$$\frac{T_d}{N} \cdot \frac{dy}{dt} + y = y$$

## 9.4 Acresentando ação anti-windup

Segundo Åström and Wittenmark (1990), a sequencia de codificação de um PID digital com ação anti-windup pode ser definida como:

1.  $p[k] = K(r[k] - y[k])$
2.  $d[k] = \frac{T_d}{(T_d + N \cdot T)} \{d[k-1] - K \cdot (y[k] - y[k-1])\}$
3.  $v = p[k] + i[k] + d[k]$
4.  $u = \text{Sat}(v)$
5.  $e[k] = r[k] - y[k]$
6.  $i[k] = i[k-1] - \left(\frac{K \cdot T}{K_i}\right) e[k] - \left(\frac{K \cdot T}{T_t}\right) (u[k] - v)$

onde  $N$  é o parâmetro que está relacionado com o filtro derivativo (normalmente se adota o valor 0,2  
 $\text{Sat}(\cdot)$  se refere ao operador de Saturação do aturador;

$T_t$  se refere à constante de tempo de “reset” para a ação integral;  
 normalmente definida como:  $T_t = \frac{1}{2} T_i$  para ação PI (somente);  
 ou  $T_t = \sqrt{T_i \cdot T_d}$  para o caso de controlador PID.

## Capítulo 10

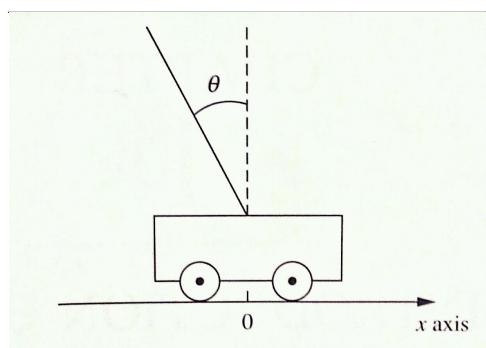
# Teoria de Controle Moderna – Abordagem por Espaço de Estados

### 10.1 Introdução

Considere o problema de tentar equilibrar (verticalmente) uma régua (ou bastão) na palma de uma mão. É óbvio que algum tipo de *controle* deve ser adotado de forma a evitar que o bastão caia da mão. Se um tipo de controle adequado for utilizado, o bastão poderá continuar (eventualmente oscilando) na posição vertical. Este é um exemplo de **problema de regulação**, onde o objetivo de controle é manter o sistema funcionando próximo do seu ponto de equilíbrio.

Agora considere um acréscimo ao problema anterior. Além de balancear o bastão na mão de forma a que o mesmo não caia, a ideia é ainda caminhar dentro de uma trilha pré-determinada. Manter o posicionamento ao longo da trilha constitui um **problema de seguimento** que deve ser cumprido enquanto se regula (se balanceia) o bastão.

Um sistema que incorpora as principais características de um problema como “manter um bastão equilibrado na vertical enquanto se caminha” é mostrado na figura 10.1 que é conhecido como o problema do pêndulo invertido sobre um carrinho. O carrinho é construído de tal forma a que só possa se movimentar numa única dimensão, no caso, ao longo do eixo  $x$  (note que sua movimentação é restringida a esta única dimensão), e o pêndulo pode se mover apenas dentro de um único plano, como o indicado pelo ângulo  $\theta$ .



[Extraído de Vaccaro (1995).]

Figura 10.1: Pêndulo invertido montado sobre um carrinho.

O problema típico de controle neste caso é regular  $\theta$  e  $x$  próximo de zero, mesmo na presença de perturbações no pêndulo ou no carrinho. Este é um problema similar ao de estabilizar um foguete durante seu lançamento. O foguete seria representado pelo pêndulo, que cai à menos que um sistema de controle seja

adotado.

No caso do sistema “carrinho-pêndulo”, um torque estabilizador é providenciado através do movimento do carrinho, enquanto que no caso do foguete, o torque é providenciado por um mecanismo que possibilita o giro dos motores de propulsão (ou *gimballed thruster engines* – ver figura 10.2). Outra diferença é que o problema do foguete possui 2 graus de liberdade – o foguete deve ser equilibrado em relação à 2 eixos ortogonais – enquanto que o pêndulo possui apenas 1 único grau de liberdade. Independentemente disto, um sistema de controle capaz de regular o sistema “carrinho-pêndulo” contém os ingredientes essenciais do sistema de controle de um foguete para um de seus eixos.

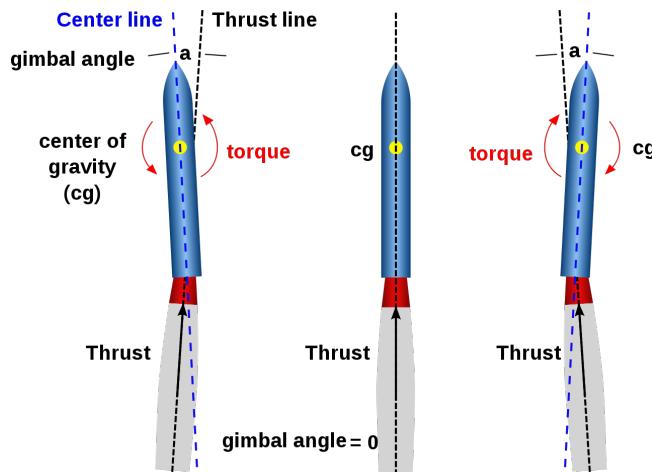


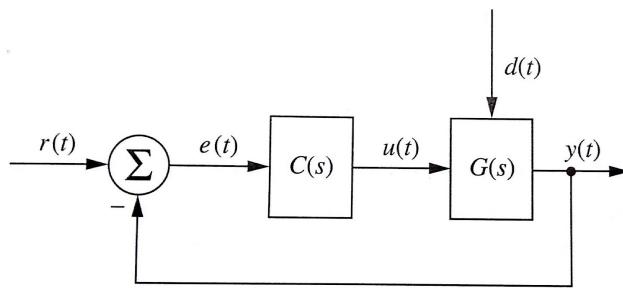
Figura 10.2: Sistema de controle propulsor de um foguete (*gimballed thruster control*).

O uso de modelos no espaço de estados se tornou sinônimo de “teoria de controle moderna” nos anos 1960. Embora outras abordagens úteis para o projeto de sistemas de controle (controle robusto, adaptativo, etc) foram desenvolvidas nos anos 1970 e 1980, estas abordagens não foram bem integradas nos currículos de graduação. A teoria de controle no espaço de estados é uma ferramenta útil para resolver uma série de problemas de controle e é passível de ser ministrado em cursos de graduação Vaccaro (1995).

Fundamentos de álgebra linear e teoria de espaço de estados são usados para projetar sistemas digitais de controle. Adicionalmente ferramentas clássicas no domínio frequência como diagramas de Bode e Nyquist são usados para analisar sistemas digitais de controle projetados usando técnicas no espaço de estados. Se assume que o leitor já tenha fundamentos prévios de transformadas de Laplace e  $\mathcal{Z}$ . E adicionalmente, conhecimentos de álgebra linear e matrizes serão úteis.

## 10.2 Definições

A figura 10.3, apresenta o protótipo de um típico sistema de controle usando a teoria clássica de controle, onde  $G(s)$  representa a função transferência do sistema (também chamado de *planta* ou *processo*) que será controlada. A saída (realimentada) da planta é denotada por  $y(t)$ ;  $r(t)$  se refere à *entrada de referência*. O problema de controle aqui é especificar a função de transferência  $C(s)$  de forma a fazer o sistema em malha-fechada alcançar determinadas especificações ou objetivos de controle. Estas podem incluir estabilidade e restrições para uma entrada (referência) degrau de forma a que sejam respeitados certos valores de percentual de sobre-sinal máximo, máximo tempo de acomodação e máximo valor de erro em regime permanente. Outro objetivo para o sistema de controle é sua capacidade de *rejeição à perturbações*. O sinal  $d(t)$  da figura 10.3 representa a perturbação. Na teoria clássica de controle,  $d(t)$  deve ser adicionado ao sinal  $u(t)$  ou  $y(T)$ . Outra característica que pode ser desejável num sistema de controle seria sua “*insensibilidade à erros no modelo da planta*”. Notar que o modelo  $G(s)$  da planta pode conter parâmetros cujos valores não são (ou não podem ser) conhecidos com precisão. Mas o sistema de controle deve continuar funcionando bem mesmo na presença de incertezas quanto à estes valores.



Extraído de Vaccaro (1995).

Figura 10.3: Típico sistema de controle analógico.

Alguns detalhes:

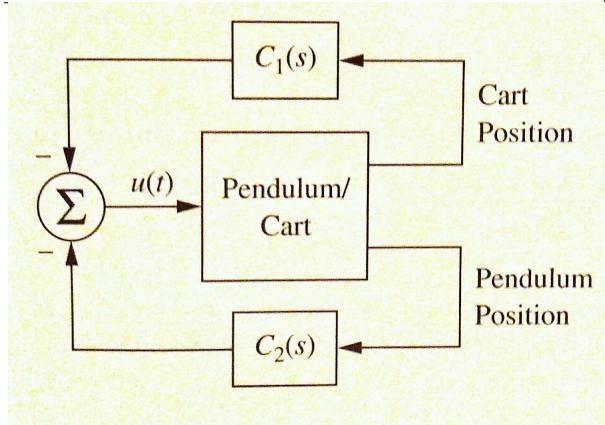
- Se  $r(t)$  é nulo durante todo o tempo, então  $C(s)$  é chamado de **regulador** e o projeto de  $C(s)$  se constitui num **problema de regulação**. Neste caso, o propósito de  $C(s)$  é manter a saída  $y(t)$  nula mesmo na presença de perturbações externas no sistema.
- Uma generalização para o problema de regulação é o caso do **problema de ajuste** (ou *setpoint problem*). Neste caso,  $r(t)$  é uma constante (entrada degrau) e o propósito de  $C(s)$  é manter a saída no valor constante da referência, chamado neste caso de **setpoint**, mesmo na presença de perturbações externas no sistema.
- Se  $r(t)$  não é constante todo o tempo (gerador de trajetório para um robô ou CNC por exemplo), então  $C(s)$  é chamado de **Compensador** ou **Controlador**, e o projeto de  $C(s)$  se transforma num **problema de projeto de servo-controlador** (ou *Servo design problem*). O sinal  $e(t)$  corresponde ao sinal de erro entre a entrada de referência e a saída atual da planta, e  $u(t)$  corresponde ao sinal de atuação sobre a planta. Note que  $u(t)$  é gerado pelo compensador para forçar a planta a se comportar da forma desejada.

Note que a teoria clássica de controle, uma pessoa aprende a projetar um compensador para controlar o comportamento de uma única variável, ou seja, a saída  $y(t)$ , como a mostrada na figura 10.3. Entretanto, num sistema “pêndulo-carrinho”, existem no mínimo 2 variáveis de interesse: a posição do pêndulo ( $\theta$ ) e a posição do carrinho ( $x$ ). Suponha ainda que alguém queira que sejam controladas estas 2 variáveis. Suponha ainda que um dos objetivos seja manter o pêndulo equilibrado mas também manter o carrinho na posição  $x = 0$  independente de perturbações à este sistema. Ou suponha ainda que  $x$  segue uma trajetória sinusoidal.

Uma abordagem clássica simples seria projetar laços de controle separados para cada variável, onde cada uma é considerada como uma saída e então somar os 2 sinais de controle (cada um resultante de cada malha separada de controle), como mostra a figura 10.4. Na figura 10.4,  $C_1(S)$  seria projetado usando uma função transferência associando o sinal de controle  $u(t)$  com a posição do carrinho ( $x(t)$ ), enquanto  $C_2(S)$  seria projetada usando uma função transferência relacionando  $u(t)$  com o ângulo do pêndulo ( $\theta$ ).

Infelizmente, os dois laços de controle (um para o ângulo do pêndulo e outro para a posição do carrinho) envolve objetivos conflitantes, assim, eles ficam “oscilando” entre si. Suponha o seguinte cenário: suponha que o pêndulo inicie equilibrado com o carrinho na posição  $x = 0$  e que uma perturbação como uma batida no pêndulo que faça inclinar-se para a direita (sentido horário). De forma a equilibrar o pêndulo, o laço de controle relacionado com o ângulo do pêndulo, gera um sinal de controle para o carrinho acelerar para a direita. Entretanto, tão logo o carrinho comece a se mover, o sinal de erro relacionado com a malha de controle de posição do carrinho (que foi especificado para ser zero) aumenta e assim, este controlador ( $C_1(s)$ ) enviará um sinal de controle especificando para o carrinho se mover para a esquerda de forma a reduzir este erro. O que resulta numa “competição” entre os 2 controladores desta malha: com um dos controladores especificando um sinal de controle exigindo que o carrinho se move para a direita enquanto o outro laço de controle exige que o carrinho se move para a esquerda.

Depois de um processo (cansativo!?, não-determinístico) de tentativa e erro, um compensador pode ser projetado para acomodar estes objetivos conflitantes. Entretanto a teoria clássica de controle não é capaz de fornecer ferramentas matemáticas convenientes para coordenar o controle de mais de uma variável ao mesmo



[Extraído de Vaccaro (1995).]

Figura 10.4: Sistema de controle analógico em laço duplo para o problema do pêndulo invertido.

tempo. Já a técnica baseado em espaço de estados, permite lidar com este tipo de situação.

### 10.3 Ferramentas de Controle Clássico

O projeto de um compensador analógico normalmente é realizado com o auxílio de diagramas no domínio frequência como diagramas de Bode ou de Nyquist ou diagramas no plano-s com o de lugar das raízes. As técnicas de projeto são normalmente procedimentos gráficos para projetar compensadores de primeira ou segunda ordem (como os de avanço ou avanço-atraso). Adicionalmente muitas técnicas clássicas de controle envolvem aproximações porque são baseadas na análise do sistema “padrão” de segunda-ordem, um sistema com 2 pólos e sem zeros. Apesar destas características, procedimentos clássicos de projeto continuam sendo úteis nas seguintes situações:

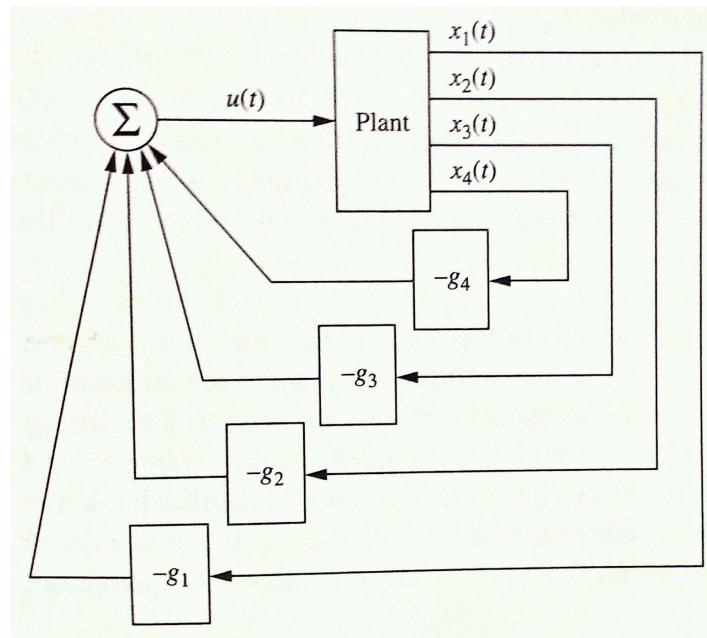
1. Sempre que a planta é um sistema padrão de segunda-ordem.
2. Sempre que a planta possui um par de “pólos dominantes” de tal forma que possa ser aproximada para um sistema padrão de segunda-ordem.
3. Sempre que um modelo matemático preciso do sistema à ser controlado não está disponível. Neste caso, dados experimentais (como o Diagrama de Bode) podem ainda ser adotados para projetar um compensador.

Ferramentas gráficas são úteis para analisar sistemas de controle para levantar informações relativas à estabilidade – por exemplo, margens de ganho e de fase.

### 10.4 Teoria de Controle no Espaço de Estados

Normalmente o protótipo típico de um sistema de controle baseado na teoria de espaço de estados segue uma estrutura de regulador com a mostrada na figura 10.5. O sistema sob controle (a planta) é caracterizada com um certo número de “variáveis de estado”; na figura 10.5 aparecem 4 variáveis de estado, chamadas de  $x_1(t)$ ,  $x_2(t)$ ,  $x_3(t)$  e  $x_4(t)$ . Os sinais (medidos) de realimentação estão relacionados com as n variáveis de estado. O sinal de controle  $u(t)$  é formado pela soma resultante da multiplicação de cada uma das  $x_i(t)$  variáveis de estado por uma constante de ganho  $g_i$ . Por se tratar de um problema de regulação, não existe nenhuma entrada externa ao sistema; seria o equivalente a uma entrada externa (referência) nula (ou zero). O projeto deste controlador envolve então especificar os ganhos  $g_1, \dots, g_4$  que ponderem as variáveis de estado de tal forma que o sinal de atuação resultante mantenha as variáveis de estado próximas do zero mesmo em face à perturbações. Note que na figura 10.5 existe um sistema de controle analógico porque as variáveis de estado não foram amostradas (ou discretizadas). Obviamente existe uma forma de projetar este tipo de

controlador usando variáveis de estados amostradas – o que caracteriza sistemas modernos de controle digital.



[Extraído de Vaccaro (1995).]

Figura 10.5: Sistema analógico de controle no espaço de estados (regulador) para uma planta de quarta-ordem.

Voltando ao problema de regulação do pêndulo-carrinho discutido anteriormente, vamos considerar agora como este problema pode ser formulado usando a técnica de espaço de estados. Primeiramente, uma **descrição no espaço de estados** do sistema teria que ser obtida. Para o sistema da figura 10.1, esta descrição resultaria num conjunto de equações que descrevem como a entrada  $u(t)$  produz as variáveis de estado  $x_i(t)$ . Tal modelo aparece descrito no Cap. 3 de Vaccaro, 1995. Será notado que as variáveis de estado podem ser determinadas da seguinte forma:  $x_1(t)$  representa a posição angular do pêndulo,  $x_2(t)$  representa a velocidade angular do pêndulo,  $x_3(t)$  e  $x_4(t)$  representam a velocidade do carrinho. Através do cálculo dos ganhos de realimentação,  $g_1, \dots, g_4$ , é possível se projetar um regulador que faça com que *todas* as variáveis de estado converjam à zero.

A abordagem por variável de estados trata *automaticamente* da coordenação entre a regulação do ângulo do pêndulo e o movimento do carrinho através da descrição matemática da planta, que detalha como as variáveis de estado estão relacionadas entre si e em relação à entrada. Adicionalmente, a teoria de espaço de estados fornece um teste para determinar quando todas as variáveis podem ser reguladas à zero através de uma simples entrada (ou através de várias entradas no caso de sistemas multivariáveis). Especificamente, se é possível determinar que a planta satisfaça a uma propriedade matemática chamada **controlabilidade**, então todas as variáveis de estado podem ser reguladas à zero. Por exemplo, o teste para controlabilidade é capaz de responder à seguinte questão: “É possível balancear 2 pêndulos simultaneamente com um simples carrinho?” O teste de controlabilidade estabelece as condições sob as quais isto é possível, e equações associadas com o projeto de realimentação de estados esclarecem com isto seja possível. (A resposta aqui é: é possível equilibrar 2 pêndulos simultaneamente se eles possuem diferentes comprimentos).

A teoria de espaço de estados pode ser estendida para problemas de seguimento na qual uma trajetória de referência é especificada para a(s) entrada(s) do sistema. É possível lidar com problemas combinados de regulação e seguimento no qual algumas das variáveis de estado devem ser reguladas até zero enquanto outras devem seguir uma trajetória de referência. Novamente neste caso, o modelo no espaço de estados permite estabelecer uma coordenação automática entre todas as variáveis envolvidas. Um exemplo disto ocorre com o próprio sistema pêndulo-carrinho. Ao invés de um puro problema de regulação com discutido anteriormente, suponha que o problema inclua uma mudança na posição do carrinho. Isto é, é desejado que o carrinho de move da posição A até a posição B enquanto mantém o pêndulo equilibrado. Se um sistema de controle for projetado usando a técnica de espaço de estados e se um comando for dado para requerer que o carrinho de move para a direita de A para B, um fenômeno surpreendente ocorrerá (o ponto B está à direita do ponto

A). Neste caso, o carrinho primeiramente se moverá para a esquerda! A razão se torna clara. Se o carrinho inicialmente se mover para a direita como foi comandado, o pêndulo começará a cair para a esquerda, e a única maneira de evitar que o pêndulo caia será fará o carrinho se mover para a esquerda. Em outras palavras, o carrinho não poderá realizar um progresso na direção do ponto B sem implicar em que o pêndulo caia.

De alguma forma, este sistema de controle seria ‘inteligente’ o bastante para se dar conta disto e assim, mover inicialmente o carrinho na direção errada, o suficiente para fazer o pêndulo inclinar para a direita. Então, quando o pêndulo estiver inclinado “apenas o suficiente”, o carrinho se moverá para a posição B de tal forma que o pêndulo voltará ao equilíbrio assim que o carrinho alcançar a posição B. Como o sistema de controle fica “inteligente o suficiente” para saber que o carrinho deve primeiramente se mover na direção errada e depois saber quanto o pêndulo deve se inclinar “apenas o suficiente” para que o carrinho possa ser movido para a direita? A resposta reside no fato de que o modelo no espaço de estados da planta fornece uma descrição completa do inter-relacionamento de todas as variáveis, e assim o conhecimento mencionando anteriormente fica codificado dentro do modelo.

Este exemplo ilustra 2 pontos importantes sobre as técnicas de espaço de estados. A primeira é a de que estas técnicas são bastante poderosas. Elas naturalmente conseguem lidar com a coordenação entre múltiplas variáveis. Adicionalmente, o projeto para sistemas de ordem elevada usa as mesmas equações algébricas lineares usadas para sistemas de menor ordem. A única diferença é que as equações para sistemas de ordem superior requerem um computar com melhor desempenho computacional para realizar os cálculos.

O segundo ponto importante sobre as técnicas de espaço de estados é que poder que as mesmas exibem provêm devido em grande parte do modelo em espaço de estados da planta. Se o modelo for acurado, então sua teoria permite alcançar excelentes resultados. Porém se o modelo da planta não resulta num reflexo exato do estado atual do sistema, os resultados podem ser quase piores. Isto leva ao próximo objetivo que seja “**controle robusto**”, que propicia técnicas para projetar sistemas de controle para plantas cuja descrição matemática não é muito exata. A robustez de um sistema de controle pode ser analisada usando critérios clássicos de margens de estabilidade (margens de ganho e margens de fase).

## 10.5 Ferramentas para projeto em espaço de estados

**POSICIONAMENTO DE PÓLOS USANDO REALIMENTAÇÃO DE ESTADOS.** Neste caso, a técnica mais básica para realizar o projeto de reguladores usando modelos no espaço de estados é através do posicionamento de pólos através da realimentação de estados. É possível demonstrar que se uma planta satisfaz uma certa propriedade matemática (controlabilidade), então os pólos do sistema em malha-fechada podem ser arbitrariamente posicionados numa determinada (desejada) posição realimentando-se uma combinação linear de variáveis de estado. O posicionamento dos pólos pode ser obtido para um sistema de ordem qualquer, e o cálculo dos ganhos de realimentação se reduzem a encontrar a solução para um sistema de equações lineares.

A técnica de posicionamento dos pólos pode ser comparada à ferramenta de controle clássico associada com o lugar das raízes. O lugar das raízes pode ser usado para calcular o valor de um simples ganho de forma a posicionar os pólos de malha-fechada. Entretanto, os pólos de malha-fechada não podem ser arbitrariamente posicionados; eles devem pertencer ao diagrama do lugar das raízes. Isto é diferente de posicionar pólos através da realimentação de estados, que permite calcular o  $n$  ganhos (para cada uma  $n$ -ésimas ordens do sistema) de forma a posicionar os pólos em qualquer posição desejada.

Ocorre que as margens de estabilidade para um regulador no espaço de estados dependem da localização dos pólos de malha-fechada. Uma desvantagem do projeto de regulador usando posicionamento de pólos é que o projetista deve escolher a localização dos pólos sem ser capaz de prever as margens de estabilidade do regulador resultante. Uma vez que o regulador é projetado, as margens de estabilidade devem ser verificadas (usando um diagrama de Nyquist por exemplo). Se as margens de estabilidade não forem adequadas (atendidas), o projetista deve selecionar um novo conjunto de localizações para os pólos de malha-fechada e calcular outro regulador.

Note então que a especificação de pólos de malha-fechada não resulta num único conjunto de ganhos de realimentação (Cap. 9 de Vaccaro (1995)). Esta falta de singularidade (resultado único) pode ser explorada usando-se algoritmos que buscam por conjuntos de ganhos que correspondam às desejadas localizações para os pólos de malha-fechada de forma que **otimizem** um critério de robustez – ou seja, exige o uso de técnicas de otimização.

**OBSERVADORES.** A implementação de regulador baseado no posicionamento de pólos requer que todas as variáveis de estado da planta possam ser monitorados (medidos) através de sensores de forma a serem usados em malhas de realimentação. Esta exigência pode ser excessivamente restritiva: algumas variáveis de estado podem ser difíceis de medir (e desta forma, muito caras para medir), e algumas podem não ter um significado físico direto. Para os casos nos quais apenas um sub-conjunto das variáveis de estado podem ser medidas, ainda assim é possível se projetar um sistema chamado de “**observador**” que estima os valores e a(s) entrada(s) para a planta. A combinação de um observador de estados junto com ganhos de realimentação de estados, resulta num **regulador dinâmico** – isto é, num regulador com pólos e zeros – ao contrário do regulador por realimentação de estados que se constitui num conjunto de ganhos.

Um problema pode advir quando se usam observadores num sistema de controle. Se o observador não for projetado de forma apropriada, é possível que as margens de estabilidade do regulador baseado em observador possam resultar ser muito piores que as margens de estabilidade de um regulador baseado em realimentação de estados. Mas é possível se projetar observadores que permitem atingir quase todas se não todas as margens de estabilidade fixadas para um regulador baseado em realimentação de estados (Cap. 7 de Vaccaro (1995)).

**SISTEMAS DE SEGUIMENTO.** É possível introduzir uma entrada de referência na estrutura de um regulador para obter-se um sistema de seguimento de trajetória. O Cap. 8 de Vaccaro (1995) trata de problemas de seguimento e regulação que são unificados usando-se o conceito de **trajetórias de estado de entrada-zero** (ou *zero-input state trajectories*). Se demonstra que seguir uma trajetória com erro zero em regime permanente frequentemente exige o uso de dinâmicas adicionais no compensador ao invés de simplesmente usar realimentações constantes de estados. Um procedimento completo para realizar o projeto de um sistema de seguimento que é robusto em relação à perturbações persistentes e à inexactidões em modelos é mostrada no Cap. 8 de Vaccaro (1995). Este procedimento fornece uma forma direta de projetar sistemas de seguimento e é baseado em conceitos clássicos relacionados com tipos de sistema e erros de regime permanente. Os sistemas de seguimento são projetados usando-se realimentação completa de estados assim como observadores.



# Apêndice A

## Equações Auxiliares

### A.1 Tabela de Derivadas

$$\frac{d}{dx} [u(x)^n] = n \cdot u(x)^{n-1} \cdot \frac{d}{dx} u(x)$$

$$\frac{d}{dx} [u(x) \cdot v(x)] = u(x) \frac{d}{dx} v(x) + v(x) \frac{d}{dx} u(x)$$

$$\frac{d}{dx} \left[ \frac{u(x)}{v(x)} \right] = \frac{v(x) \frac{d}{dx} u(x) - u(x) \frac{d}{dx} v(x)}{v(x)^2}$$

### A.2 Equação do Segundo Grau

$$x^2 - Sx + P = 0$$
$$(x - x_1)(x - x_2) = 0$$

$$(a + b)^2 = a^2 + 2ab + b^2$$

$$(a - b)^2 = a^2 - 2ab + b^2$$

### A.3 Relações de Euler

$$e^{jx} + e^{-jx} = 2 \cos(x)$$
$$e^{jx} - e^{-jx} = 2j \sin(x)$$
$$\cos(x) = \frac{e^{jx} + e^{-jx}}{2}$$
$$\sin(jx) = \frac{e^{jx} - e^{-jx}}{2j}$$

### A.4 Progressão Geométrica

$$S_n = a + aq + aq^2 + \dots + aq^{n-1}$$

$$S_n = \sum_{i=0}^{n-1} aq^n = \frac{a - aq^n}{1 - q}$$

$$S_{\infty} = \sum_{i=0}^{\infty} aq^n = \frac{a}{1-q}, \quad \text{se } |q| < 1$$

## A.5 Definição da Transformada Z

Seja  $f^*(t)$  o resultado de do sinal contínuo  $f(t)$  discretizado no tempo por um trem de impulsos  $\delta_T(t)$  ocorrendo à cada instante  $T$  de amostragem:

$$f^*(t) = \sum_{k=0}^{\infty} f(kT) \cdot \delta(t - kT) = f(t) \cdot \delta_T(t)$$

Então a Transformada de Laplace do sinal discretizado fica:

$$\mathcal{L}\{f^*(t)\} = \sum_{k=0}^{\infty} f(kT) e^{-Ts}$$

A transformada Z é o resultado da transformação:

$$z = e^{Ts}$$

onde:

$$s = \frac{1}{T} \ln z$$

ou, se temos a disposição as amostras do sinal discretizado temos:

$$F(z) = \sum_{k=0}^{\infty} f(kT) z^{-k}$$

## A.6 Propriedades da Transformada Z

Teorema/Propriedade	Nome (Comentários)
1. $\mathcal{Z}\{a \cdot f(k) + b \cdot g(k)\} = a \cdot F(z) + b \cdot G(z)$	Linearidade: Adição, subtração e multiplicação por constante. Onde $a$ e $b$ são constantes.
2. $\mathcal{Z}\{f(k+1)\} = z F(z) - z f(0)$	Avanço no Tempo
3. $\mathcal{Z}\{x(t - nT)\} = x^{-n} X(z)$	Atraso no Tempo
4. $\begin{aligned} \mathcal{Z}\{e^{-at} x(t)\} &= \sum_{k=0}^{\infty} x(kT) \cdot e^{-akT} z^{-k} \\ &= \sum_{k=0}^{\infty} x(kT) (z e^{aT})^{-k} = X(z e^{aT}) \end{aligned}$	Translação Complexa
5. $\mathcal{Z}\{f(t) * g(t)\} = F(z) \cdot G(z)$	Convolução: Multiplicação no domínio frequência
6. $f(0) = \lim_{z \rightarrow \infty} F(z)$	Teorema do Valor Inicial
7. $f(\infty) = \lim_{k \rightarrow \infty} f(k) = \lim_{z \rightarrow 1} (1 - z^{-1}) F(z)$	Teorema do Valor Final



## A.7 Tabela de Transformadas Z

	$F(s)$	$f(t)$	$f(kT)$ ou $f(k)$	$F(z)$	Obs.
1	1	$\delta(t)$	$\delta(kT)$	1	Função Impulso
2	$e^{-kTs}$	$\delta(t - kT)$	$\delta(t - kT)$	$z^{-k}$	Função Impulso deslocada no tempo
3	$\frac{1}{s}$	$u(t)$	$u(k)$ ou 1	$\frac{z}{z-1}$	Degrau Unitário
3a			$a^k$	$\frac{z}{z-a}$	
4	$\frac{1}{s^2}$	$t$	$kT$	$\frac{Tz}{(z-1)^2}$	Rampa
5	$\frac{2}{s^3}$	$t^2$	$(kT)^2$	$\frac{T^2 z(z+1)}{(z-1)^3}$	Parábola
5a	$\frac{1}{s^3}$		$\frac{1}{2!}(kT)^2$	$\frac{T^2}{2} \frac{z(z+1)}{(z-1)^3}$	
5b	$\frac{1}{s^4}$		$\frac{1}{3!}(kT)^3$	$\frac{T^3}{6} \frac{z(z^2+4z+1)}{(z-1)^4}$	
5c	$\frac{1}{s^m}$		$\lim_{a \rightarrow \infty} \frac{(-1)^{m-1}}{(m-1)!} \frac{\partial^{m-1}}{\partial a^{m-1}} e^{-akT}$	$\lim_{a \rightarrow \infty} \frac{(-1)^{m-1}}{(m-1)!} \frac{\partial^{m-1}}{\partial a^{m-1}} \frac{z}{(z - e^{-aT})}$	
6	$\frac{1}{s+a}$	$e^{-at}$	$e^{-akT}$	$\frac{z}{z - e^{-aT}}$	Exponencial
6a	$\frac{1}{(s+a)^2}$	$t e^{at}$	$kT e^{-akT}$	$\frac{Tze^{-aT}}{(z - e^{-aT})^2}$	Pólos Múltiplos
6b	$\frac{1}{(s+a)^3}$		$\frac{1}{2}(kT)^2 e^{-akT}$	$\frac{T^2}{2} e^{-aT} \frac{z(z + e^{-aT})}{(z - e^{-aT})^3}$	
6c	$\frac{1}{(s+a)^m}$		$\frac{(-1)^{m-1}}{(m-1)!} \frac{\partial^{m-1}}{\partial a^{m-1}} (e^{-akT})$	$\frac{(-1)^{m-1}}{(m-1)!} \frac{\partial^{m-1}}{\partial a^{m-1}} \frac{z}{z - e^{-aT}}$	
7	$\frac{a}{s(s+a)}$	$1 - e^{-at}$	$1 - e^{-akT}$	$\frac{z(1 - e^{-aT})}{(z-1)(z - e^{-aT})}$	Exponencial Decrescente
7a	$\frac{a}{s^2(s+a)}$		$\frac{1}{a} (akT - 1 + e^{-akT})$	$\frac{z[z(aT - 1 + e^{-aT}) + (1 - e^{-aT} - aTe^{-aT})]}{a(z-1)^2 (z - e^{-aT})}$	
8	$\frac{\omega}{s^2 + \omega^2}$	$\sin \omega t$	$\sin \omega kT$	$\frac{z \sin \omega T}{z^2 - 2z \cos \omega T + 1}$	Senóide
9	$\frac{s}{s^2 + \omega^2}$	$\cos \omega t$	$\cos \omega kT$	$\frac{z(z - \cos \omega T)}{z^2 - 2z \cos \omega T + 1}$	Cossenóide
10	$\frac{\omega}{(s+a)^2 + \omega^2}$	$e^{-at} \sin \omega t$	$e^{-akT} \sin \omega kT$	$\frac{ze^{-aT} \sin \omega T}{z^2 - 2ze^{-aT} \cos \omega T + e^{-2aT}}$	Senóide amortecida
11	$\frac{s+a}{s^2 + \omega^2}$	$e^{-at} \cos \omega t$	$e^{-akT} \cos \omega kT$	$\frac{z^2 - ze^{-aT} \cos \omega T}{z^2 - 2ze^{-aT} \cos \omega T + e^{-2aT}}$	Cossenóide amortecida



## Apêndice B

# Respostas temporais no Plano- $\mathcal{Z}$

### B.1 Casos de Pólos Reais

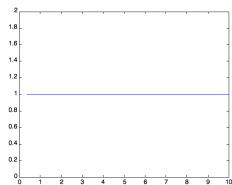
Lembrando que:

$$\mathcal{Z}^{-1} \left\{ \frac{c \cdot z}{z - \alpha} \right\} = c \cdot \alpha^k$$

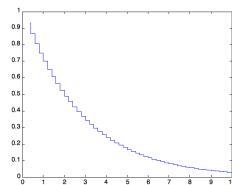
Percebe-se que:

$$\mathcal{Z}^{-1} \left\{ \frac{c \cdot z}{z - \alpha} \right\} \rightarrow 0 \quad \text{tanto mais rápido quanto: } |\alpha| \rightarrow 0$$

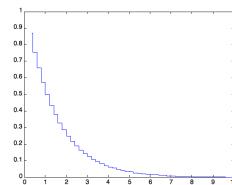
Podemos simular algumas respostas para diferentes posições de pólos reais dentro do círculo unitário no plano- $\mathcal{Z}$ :



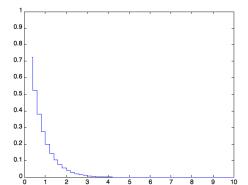
$$\mathcal{Z}^{-1} \left\{ \frac{1 \cdot z}{z - 1} \right\} = 1$$



$$\mathcal{Z}^{-1} \left\{ \frac{1 \cdot z}{z - 0,7} \right\} = 0,7^k$$



$$\mathcal{Z}^{-1} \left\{ \frac{1 \cdot z}{z - 0,5} \right\} = 0,5^k$$



$$\mathcal{Z}^{-1} \left\{ \frac{1 \cdot z}{z - 0,2} \right\} = 0,2^k$$



$$\mathcal{Z}^{-1} \left\{ \frac{1 \cdot z}{z + 0,7} \right\} = (-0,7)^k$$



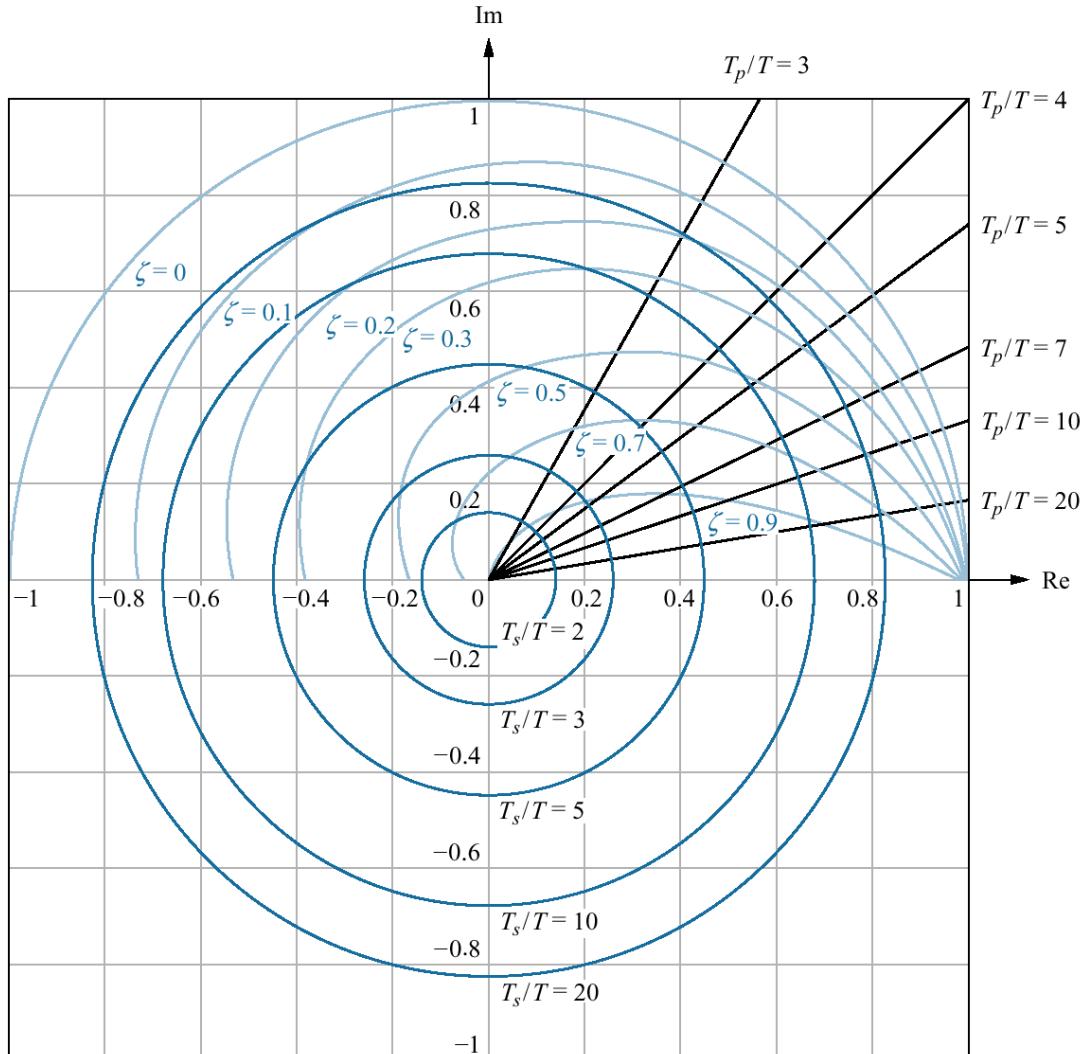
$$\mathcal{Z}^{-1} \left\{ \frac{1 \cdot z}{z + 0,5} \right\} = (-0,5)^k$$



$$\mathcal{Z}^{-1} \left\{ \frac{1 \cdot z}{z + 0,2} \right\} = (-0,2)^k$$

## B.2 Caso de Pólos Complexos

A figura B.1 ilustra o plano-z ressaltando diferentes fatores de amortecimento ( $\zeta$ ), tempos de resposta ( $T_s$ ) normalizados e tempos de pico ( $T_p$ ) normalizados.



Obs.: Extraído de Nise (2012).

Figura B.1: Plano-z com referências para diferentes valores de  $\zeta$ ,  $T_s$  e  $T_p$

## Apêndice C

# Transformada Z - Uso do MATLAB

### C.1 Transformada Z Inversa (Usando o MATLAB)

Como usar o MATLAB para realizar Transformadas Inversas de Z<sup>1</sup>

Para “ativar” o symbolic toolbox é necessário antes de mais nada definir que “variáveis” o MATLAB deve tratar com simbólicas. Por exemplo:

```
>> z=sym('z')
z =
z
>>
```

e podemos também fazer:

```
>> k=sym('k');
```

teremos algo como:

```
>> whos
  Name      Size            Bytes  Class
  ans       1x1              8  double array
  k         1x1             126  sym object
  z         1x1             126  sym object
Grand total is 5 elements using 260 bytes
>>
```

Para realizar transformadas inversas de Z, simbólicas, podemos usar a função `iztrans` do Symbolic Toolbox do MATLAB.

**Exemplo 1:** Seja:  $F(z) = \frac{z}{z-2}$ , descobrir  $f(kT)$ .

No MATLAB faríamos:

```
>>iztrans(z/(z-2))
ans =
2^n
>>
```

ou poderíamos fazer:

```
>> iztrans(z/(z-2),k)
ans =
```

<sup>1</sup>Necessário o “Symbolic Math Toolbox”

```

z^k
>>

```

O que nós dá:  $f(kT) = 2^k$ .

Lembrando da tabela, temos que:  $\mathcal{Z}\{a^k\} = \frac{z}{z-a}$ .

**Exemplo 2:** Seja:

$$F(z) = \frac{0,5z}{(z-1)(z-0,5)}$$

determine  $f(kT)$ .

Usando o MATLAB:

```

>> iztrans(0.5*z/((z-1)*(z-0.5)), k)
ans =
1-(1/2)^k
>>

```

ou seja,  $f(kT) = 1 - 1/2^k = 1 - 0,5^k$ , que é o resultado que se pode obter resolvendo este problema pelo método das frações parciais:

$$\frac{F(z)}{z} = \frac{0,5}{(z-1)(z-0,5)} = \frac{A}{z-1} + \frac{B}{z-0,5}$$

resolvendo:

$$\begin{aligned} A &= (z-1) \cdot F(z)|_{z=1} \quad \therefore A = \left. \frac{(z-1) \cdot 0,5}{(z-1)(z-0,5)} \right|_{z=1} \quad \therefore A = \frac{0,5}{1-0,5} = 1 \\ B &= (z-0,5) \cdot F(z)|_{z=0,5} \quad \therefore B = \left. \frac{(z-0,5) \cdot 0,5}{(z-1)(z-0,5)} \right|_{z=0,5} \quad \therefore B = \frac{0,5}{0,5-1} = -1 \end{aligned}$$

finalmente:

$$F(z) = \frac{F(z)}{z} = \frac{1}{z-1} - \frac{1}{z-0,5}$$

ou:

$$F(z) = \frac{z}{z-1} - \frac{z}{z-0,5}$$

Consultando a tabela de transformadas temos que:

$$f(kT) = 1 \cdot 1^k - 1 \cdot (0,5)^k = u(k) - 0,5^k$$

**Exemplo 3:** Seja:

$$Y(z) = \frac{z+4}{(z-1)(z-2)}$$

obtenha  $y(kT)$ :

Pelo método da expansão em frações parciais:

$$\frac{Y(z)}{z} = \frac{z+4}{z(z-1)(z-2)} = \frac{a}{z} + \frac{b}{z-1} + \frac{c}{z-2}$$

$$\begin{aligned} a &= z \cdot F(z)|_{z=0} \quad \therefore a = \left. \frac{z \cdot (z+4)}{z(z-1)(z-2)} \right|_{z=0} \quad \therefore a = \frac{4}{(-1)(-2)} = 2 \\ b &= (z-1) \cdot F(z)|_{z=1} \quad \therefore b = \left. \frac{(z-1) \cdot (z+4)}{z(z-1)(z-2)} \right|_{z=1} \quad \therefore b = \frac{5}{(1)(-1)} = -5 \\ c &= (z-2) \cdot F(z)|_{z=2} \quad \therefore c = \left. \frac{(z-2) \cdot (z+4)}{z(z-1)(z-2)} \right|_{z=2} \quad \therefore c = \frac{6}{(2)(1)} = 3 \end{aligned}$$

$$\frac{Y(z)}{z} = \frac{2}{z} + \frac{(-5)}{z-1} + \frac{3}{z-2}$$

$$Y(z) = \frac{2z}{z} + \frac{(-5)z}{z-1} + \frac{3z}{z-2}$$

Lembrando da Tabela:

$$Z^{-1} \left\{ \frac{z}{z-a} \right\} = a^k$$

$$Z^{-1} \left\{ \frac{z}{z-1} \right\} = u(kT)$$

temos então que:

$$y(kT) = 2\delta(k) - 5 \cdot u(k) + 3 \cdot (2)^k$$

Usando o MATLAB ficaria:

```
>> iztrans((z+4)/((z-1)*(z-2)),k)
ans =
2*charfcn[0](k)-5+3*2^k
>>
```

ou simplesmente:  $y(kT) = 2\delta(k) - 5 \cdot u(k) + 3 \cdot 2^k$ .

---

**Exemplo 4:** Determine  $y(kT)$  para:

$$Y(z) = \frac{z^3 - 2z^2 + 2z}{(z-1)^3}$$

Usando o MATLAB teríamos:

```
>> iztrans((z^3-2*z^2+2*z)/((z-1)^3),k)
ans =
1-1/2*k+1/2*k^2
>>
```

ou simplesmente:  $y(kT) = 1 - \frac{1}{2}k + \frac{1}{2}k^2$ .

---

**Exemplo 5:** Obter  $f(kT)$  para:

$$F(z) = \frac{z^2 + z + 0,25}{(z - 0,5)^2}$$

Resolvendo pelo método dos resíduos:

$$F(z) = \frac{z^2 + z + 0,25}{(z - 0,5)^2} = \frac{(z + 0,5)(z + 0,5)}{(z - 0,5)(z - 0,5)}$$

Como são pólos múltiplos:

$$\text{Resíduo}_{[p_m]} = \frac{1}{(m-1)!} \cdot \frac{d^{m-1}}{dz^{m-1}} \left[ (z - p_m)^m \cdot F(z) \cdot z^{k-1} \right] \Big|_{z=p_m}$$

Então:

$$\begin{aligned}
 \text{Resíduo}_{[0,5]_{(2)}} &= \frac{d}{dz} \left[ (z-0,5)^2 \cdot \frac{(z+0,5)^2 \cdot z^{k-1}}{(z-0,5)^2} \right] \Big|_{z=0,5} = \frac{d}{dz} \left[ (z^2 + z + 0,25) \cdot z^{k-1} \right] \Big|_{z=0,5} \\
 &= \frac{d}{dz} \left[ z^{k+1} + z^k + 0,25z^{k-1} \right] \Big|_{z=0,5} \\
 &= (k+1)z^{k+1-1} + kz^{k-1} + 0,25(k-1)z^{k-2} \Big|_{z=0,5} \\
 &= (k+1)z^k + kz^{k-1} + 0,25(k-1)z^{k-2} \Big|_{z=0,5} \\
 &= z^k \left[ (k+1) + kz^{-1} + \frac{1}{4}(k-1)z^{-2} \right] \Big|_{z=0,5} \\
 &= (0,5)^k \left[ (k+1) + k \cdot \left(\frac{1}{2}\right)^{-1} + \frac{1}{4}(k-1) \cdot \left(\frac{1}{2}\right)^{-2} \right] \\
 &= (0,5)^k [4k] \\
 &= 4k \cdot (0,5)^k \\
 f(kT) &= 4k(0,5)^k
 \end{aligned} \tag{C.1}$$

Mas...

$$f(0) = \lim_{z \rightarrow \infty} \left( \frac{z^2 + z + 0,25}{z^2 - z + 0,25} \right) \tag{C.2}$$

$$\begin{aligned}
 &= \lim_{z \rightarrow \infty} \left( \frac{1 - \frac{1}{z} + \frac{0,25}{z^2}}{1 - \frac{1}{z} + \frac{0,25}{z^2}} \right) \\
 &= 1
 \end{aligned} \tag{C.3}$$

mas para  $k = 0$  de  $f(kT)$  encontrado anteriormente (eq. (C.1)), obterímos:  $f(0) = 0$  que “não bate” com o encontrado na (eq. C.2). Então falta compensar (ajustar) para este termo inicial e assim  $f(kT)$  fica igual à:

$$f(kT) = \delta(k) + 4k(0,5)^k$$

Usando o MATLAB:

```
>> f=iztrans((z^2+z+0.25)/((z-0.5)^2),k)
f =
charfcn [0](k)+4*(1/2)^k*k
>>
```

ou seja:  $y(kT) = \delta(k) + 4 \cdot \left(\frac{1}{2}\right)^k \cdot k$ .

## C.2 Sistemas com Sustentador de Ordem Zero [ $B_0 G(z)$ ]

Note que um sustentador de ordem zero pode ser modelado como:

$$\begin{aligned}
 B_0(s) &= \frac{S(s)}{E(s)} \\
 &= \mathcal{L}\{u(t) - u(t-T)\} \\
 &= \frac{1}{s} - \frac{e^{-Ts}}{s} \\
 B_0(s) &= \frac{1 - e^{-Ts}}{s}
 \end{aligned}$$

Mas agora quando associamos o sustentador a um processo,  $G(s)$ , o sistema fica:

Antes (no plano-s):

$$\frac{S(s)}{E(s)} = B_0(s) \cdot G(s)$$

agora que estamos no plano-z, isto é, digitalizamos o sistema, ficamos com:

$$\begin{aligned} \frac{S(z)}{E(z)} &= z\left\{B_0(s) \cdot G(s)\right\} \\ &= z\left\{B_0(s) \cdot G(s)\right\} \\ &= z\left\{\frac{1-e^{-Ts}}{s} \cdot G(s)\right\} \\ &= z\left\{\frac{G(s)}{s}\right\} - z\left\{\frac{e^{-Ts}G(s)}{s}\right\} \end{aligned}$$

Se fizermos  $F(s) = \frac{G(s)}{s}$ , temos:

$$z\left\{B_0(s) \cdot G(s)\right\} = z\left\{F(s)\right\} - z\left\{e^{-Ts} \cdot F(s)\right\}$$

como  $e^{-Ts} \cdot F(s)$  equivale a um deslocamento no tempo de  $f(t)$ :

$$e^{-Ts} \cdot F(s) \Rightarrow f(t-T)$$

então:

$$z\left\{f[(k-1)T]\right\} = z^{-1} \cdot F(z)$$

e assim:

$$z\left\{B_0(s) \cdot G(s)\right\} = z\left\{F(s)\right\} - z^{-1} \cdot z\left\{F(s)\right\}$$

ou simplesmente:

$$B_0 G(z) = (1 - z^{-1}) \cdot z\left\{\frac{G(s)}{s}\right\}$$

que é o método que adotaremos daqui por diante.



## Apêndice D

# Resposta ao Degrau e localização dos pólos

Seja um sistema de 2<sup>a</sup>-ordem cuja equação diferencial seja:

$$\frac{d^2y}{dx^2} + 2\zeta\omega_n \frac{dy}{dx} + 2\omega_n^2 y = 2\omega_n^2 x$$

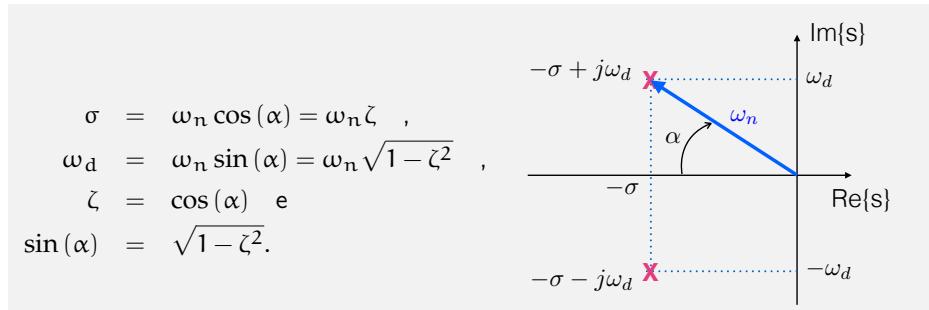
sua transformada de Laplace resulta em:

$$G(s) = \frac{Y(s)}{R(s)} = \left[ \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \right] \text{ ou } G(s) = \frac{b}{s^2 + \alpha s + b}$$

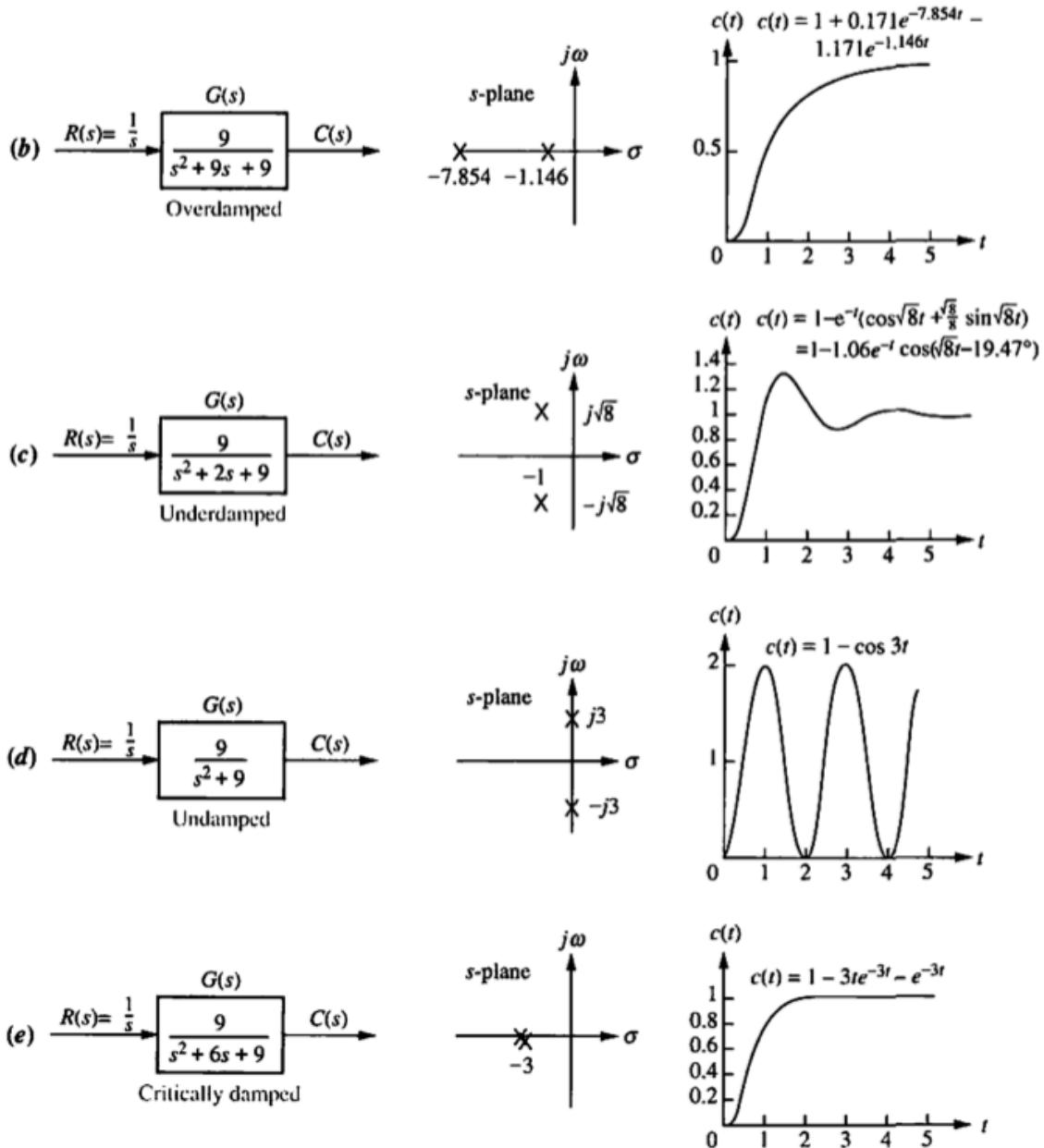
supondo o caso de raízes complexas para o denominador de  $G(s)$ , temos pólos que se localizam em:

$$s = \sigma \pm j\omega_d \quad , \text{ ou } s = -\zeta\omega_n \pm j\omega_n\sqrt{1-\zeta^2},$$

onde:



Note que quando este sistema é submetido à uma entrada degrau unitário  $\left( R(s) = \frac{1}{s} \right)$ , resulta em diferentes tipos de resposta – ver figura D.1.



Obs.: Extraído de Nise (2012).

Figura D.1: Respostas típicas para sistemas de 2a-ordem submetidos à uma entrada degrau unitário.

- quando  $\zeta = 0$  temos um dos pólos sobre o eixo  $s = 0$  resultando num sistema oscilatório;
- quando  $0 < \zeta < 1$ , temos 2 pólos complexos resultando num sistema sub-amortecido;
- quando  $\zeta = 1$  temos 2 pólos reais resultando num sistema criticamente amortecido;

## D.1 Sistemas com $0 < \zeta < 1$

A resposta de um sistema de 2a-ordem,  $C(s)$  do tipo sub-amortecido, submetido à uma entrada degrau, resulta em:

$$C(s) = \frac{\omega_n^2}{s(s^2 + 2\zeta\omega_n s + \omega_n^2)} = \frac{K_1}{s} + \frac{K_2 s + K_3}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

cuja expansão em frações parciais leva à:

$$C(s) = \frac{1}{s} - \frac{(s + \zeta\omega_n) + \frac{\zeta}{\sqrt{1-\zeta^2}}\omega_n\sqrt{1-\zeta^2}}{(s + \zeta\omega_n)^2 + \omega_n^2(1-\zeta^2)} \quad (\text{D.1})$$

A Transforma inversa de Laplace da eq. (D.1) produz:

$$\begin{aligned} c(t) &= 1 - e^{-\zeta\omega_n t} \left[ \cos(\omega_n \sqrt{1-\zeta^2} t) + \frac{\zeta}{\sqrt{1-\zeta^2}} \sin(\omega_n \sqrt{1-\zeta^2} t) \right] \\ &= 1 - \frac{1}{\sqrt{1-\zeta^2}} e^{-\zeta\omega_n t} \cos(\omega_n \sqrt{1-\zeta^2} t - \phi) \end{aligned}$$

onde:  $\phi = \tan^{-1} \left( \frac{\zeta}{\sqrt{1-\zeta^2}} \right)$ .

## Parâmetros típicos de sistema de segunda ordem

Baseado em  $G(s) \rightarrow G(s) = \frac{b}{s^2 + ax + b} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$

podemos extrair certos parâmetros típicos de um sistema de segunda ordem:

- **Frequência natural de oscilação:**  $\omega_n$ , corresponde a frequência de oscilação do sistema sem amortecimento (quando a parte imaginária dos pólos,  $a = 0$ , com pólos sobre o eixo  $j\omega$  em:  $\pm j\sqrt{b}$ ,  $\omega_n = \sqrt{b}$ ,  $b = \omega_n^2$ ).
- **Fator de Amortecimento:**  $\zeta$ , pólos complexos contêm parte real:  $\sigma = -a/2$ . A magnitude deste coeficiente modula o decaimento exponencial da senóide amortecida:

$$\zeta = \frac{\text{Exponential decay frequency}}{\text{Natural frequency}} = \frac{|\sigma|}{\omega_n} = \frac{a/2}{\omega_n}$$

onde:  $a = 2\zeta\omega_n$ .

- **Tempo de Pico (Peak time):**  $T_p$ , instante de tempo no qual ocorre o valor máximo de saída do sistema. O **Tempo do pico**,  $T_p$ , é dado por:

$$T_p = \frac{\pi}{\omega_n \sqrt{1-\zeta^2}}$$

- **Percentual de Sobrepasso (Percent Overshoot):** %OS, quantidade de sobrepasso do sinal de saída que excede o valor de regime estacionário. O percentual de sobrepasso, %OS, é previsto através de:

$$\%OS = e^{-\left(\zeta\pi/\sqrt{1-\zeta^2}\right)} \times 100$$

ou, especificado o valor de %OS, pode-se determinar o fator de amortecimento,  $\zeta$ , através de:

$$\zeta = \frac{-\ln(\%OS/100)}{\sqrt{\pi^2 + \ln^2(\%OS/100)}}$$

E o seguinte código no MATLAB, permite calcular o valor de  $\zeta$  a partir do valor atribuído para %OS:

```
>> OS=5; %percentual de sobrepasso fornecido em %
>> zeta=(-log(OS/100))/sqrt(pi^2+(log(OS/100)^2))
zeta =
    0.6901
>>
```

- **Tempo de Assentamento (Settling time):**  $T_s$ , tempo requerido para que as oscilações fiquem abaixo de  $\pm 2\%$  da banda (região) de assentamento do valor de regime estacionário. O tempo de acomodação ou assentamento (ou settling time),  $T_s$ , pode ser determinado através de:

$$T_s = \frac{-\ln(0,02\sqrt{1-\zeta^2})}{\zeta\omega_n} \quad \text{ou} \quad T_s = \frac{4}{\zeta\omega_n}$$

(Válido para:  $0 < \zeta < 0,9$ )

- **Tempo de Subida (Rise time):**  $T_r$ , tempo exigido para a saída variar entre 0,1 e 0,9 de seu valor final. E finalmente o tempo de subida (*rise time*),  $T_r$ , pode ser determinado de:

$$T_r = \frac{0,6 + 2,16\zeta}{\omega_n}$$

## D.2 Relações entre plano-s e plano-z

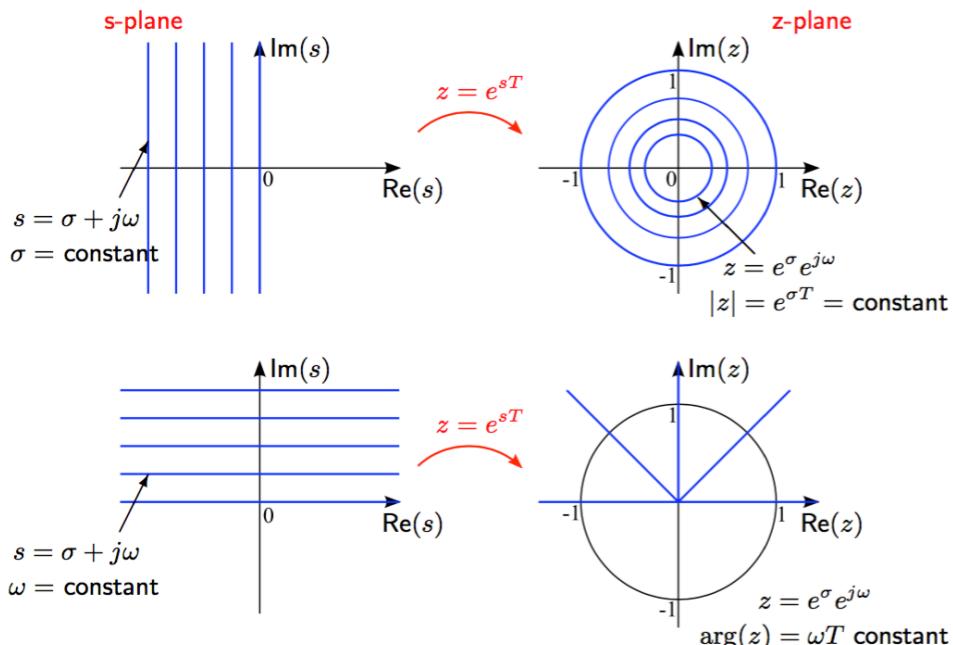


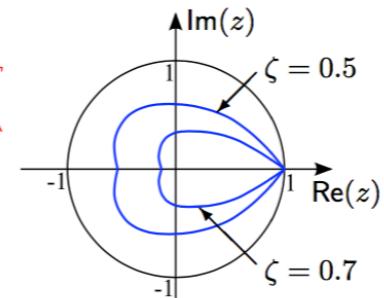
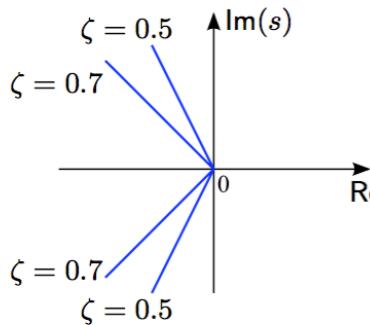
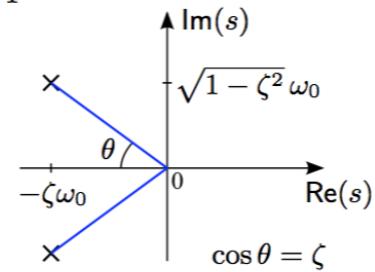
Figura D.2: Relações de entre plano-s e plano-z quando a real ( $\omega$ ) ou imaginária ( $\omega$ ) são constantes.

Pole locations for constant damping ratio  $\zeta < 1$

$$s^2 + \zeta\omega_0 s + \omega_0^2 = 0$$

$\Downarrow$

$$s = -\zeta\omega_0 \pm j\sqrt{1 - \zeta^2}\omega_0$$



$$s = -\zeta\omega_0 + j\sqrt{1 - \zeta^2}\omega_0: \zeta = \text{constant}$$

$$z = e^{-\zeta\omega_0 T} e^{-j\sqrt{1 - \zeta^2}\omega_0 T}$$

Figura D.3: Relações de entre plano-s e plano-z quando a real ( $\omega$ ) ou imaginária ( $\omega$ ) são constantes.

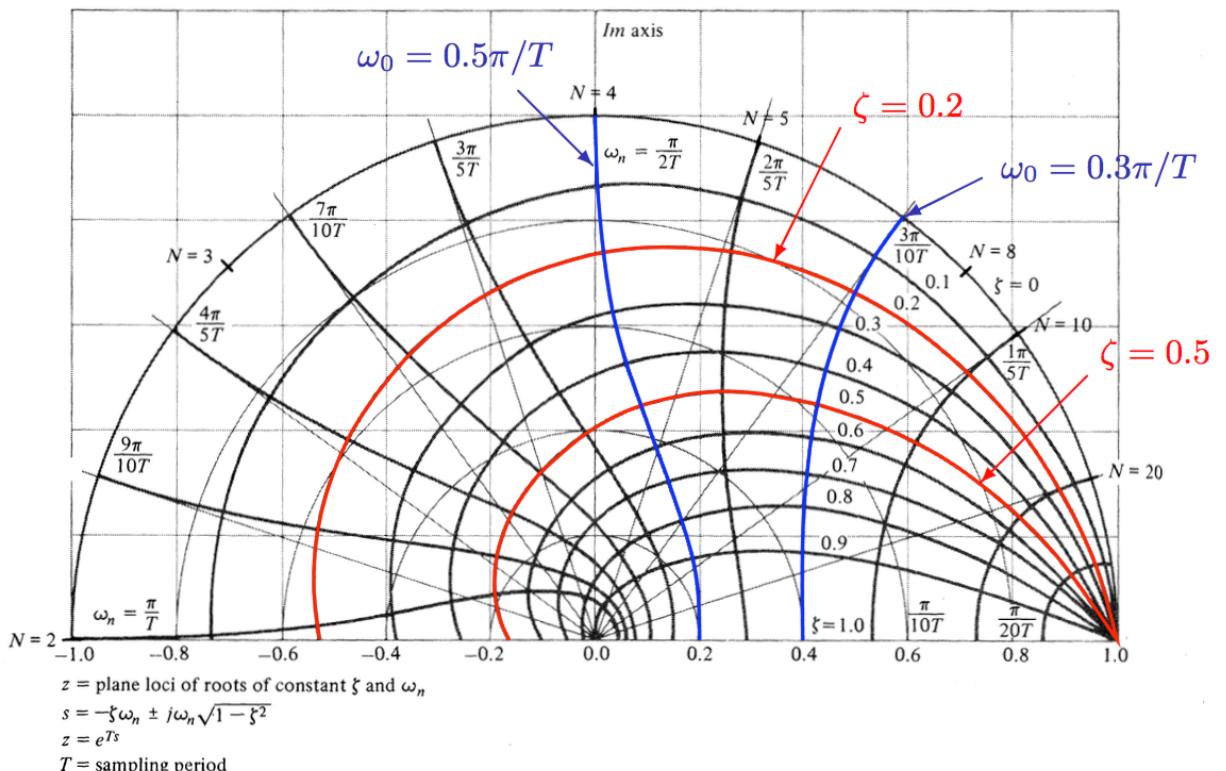


Figura D.4: Relações de entre plano-s e plano-z quando a real ( $\omega$ ) ou imaginária ( $\omega$ ) são constantes.



## Apêndice E

# Revisão Controladores Clássicos

### E.1 Introdução

Quando trabalhamos com o projeto de um controlador normalmente estamos preocupados com:

- Limitações no ajuste de ganhos:
  - Ajustar somente o ganho proporcional conduz a uma resposta transitória determinada unicamente pela posição original dos pólos da planta no diagrama do Lugar das Raízes (RL=Root Locus);
  - Desta forma se fica limitado a respostas que somente existem ao longo do RL original da planta.
  - Uma forma de resolver esta limitação é aumentar a ordem do sistema ou compensar o sistema com pólos e zeros adicionais de modo que o sistema resultante (compensado) resulte num RL que passe por pólos de malha-fechada desejados para o sistema compensado dentro de uma faixa de ganho desejável.
- Melhorar a resposta transitória do sistema em malha-fechada;
- Reduzir o erro estacionário (ou de regime permanente):
  - Podem ser usados compensadores de forma independente para melhorar as características do erro em regime permanente.
  - Duas configurações são possíveis:
    - \* Compensadores em Cascata – ver figura E.1(a).
    - \* Compensador na Malha de Realimentação (*Feedback compensator*) – ver figura E.1(b).

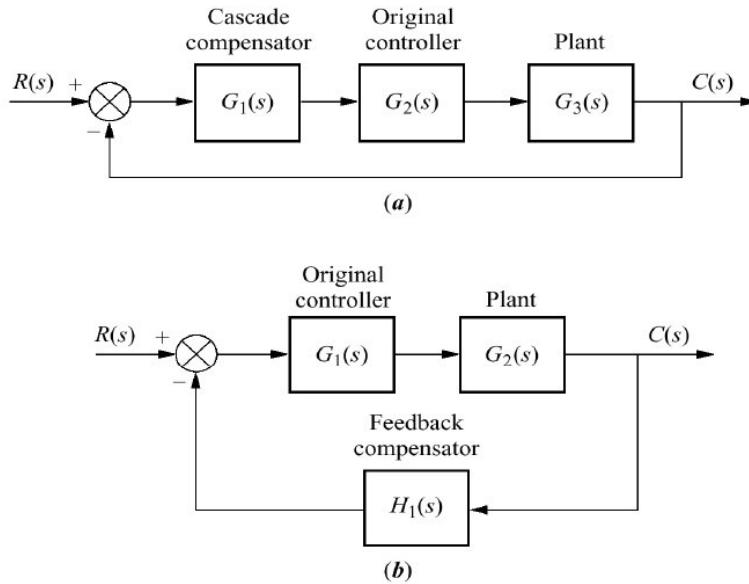


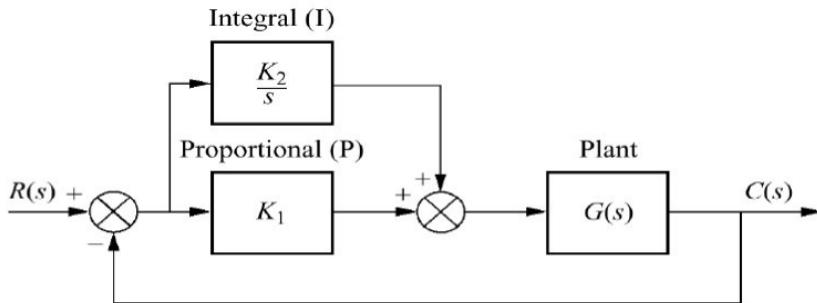
Figura E.1: Configurações possíveis para controladores.

O projeto de controladores em cascata (ver figura E.2) permite:

- reduzir o erro em regime permanente;
- melhorar a resposta transitória (acelerar o sistema);

$$G_c(s) = K_1 + \frac{K_2}{s} = \frac{K_1 \left( s + \frac{K_2}{K_1} \right)}{s}$$

← zero próximo do pólo  
 ← pólo na origem

Figura E.2: Exemplo de controlador PI (com zero extra em  $s = -\frac{K_2}{K_1}$ , além do pólo na origem).

## E.2 Características principais dos controladores “clássicos”

Os seguintes controladores “clássicos” normalmente adotados são:

**PI:** zera o erro de regime permanente; consiste num zero próximo do pólo que fica localizado sobre o círculo unitário,  $z = 1$  (ou na origem do plano-s,  $s = 0$ ). Exige rede ativa no mundo contínuo (isto é, implica necessariamente o emprego de amplificador operacional). Modifica intencionalmente o RL original do sistema (planta) – ver figura E.3.

**Atraso de Fase:** introduz um atraso no sistema (*Lag*), reduz significativamente o erro de regime permanente, pode ser implementado por rede passiva (no plano-s, controlador contínuo no tempo); consiste num

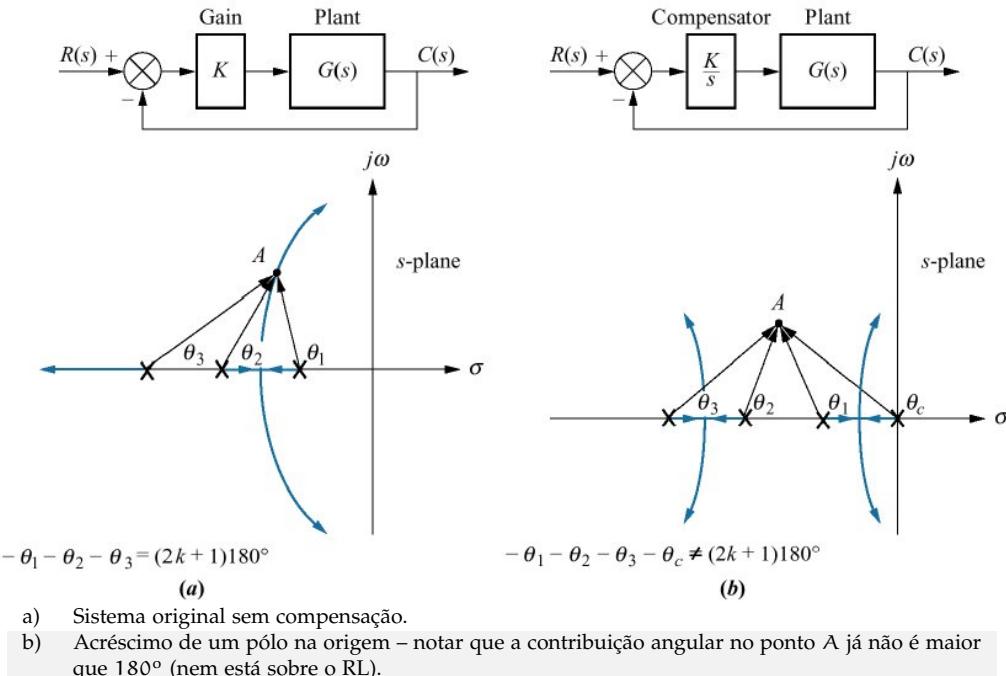


Figura E.3: Controlador PI: Anula o erro de regime permanente mas modifica o RL.

par polo-zero localizados próximos do círculo unitário ( $z = 1$  no plano-z; no plano-s este par estaria localizado próximo da origem), o polo mais próximo do círculo unitário (ou da origem do plano-s) que em comparação com seu zero – ver figura E.4. Adquire comportamento próximo de um PI, quanto mais próximo da origem no plano-s estiver seu polo.

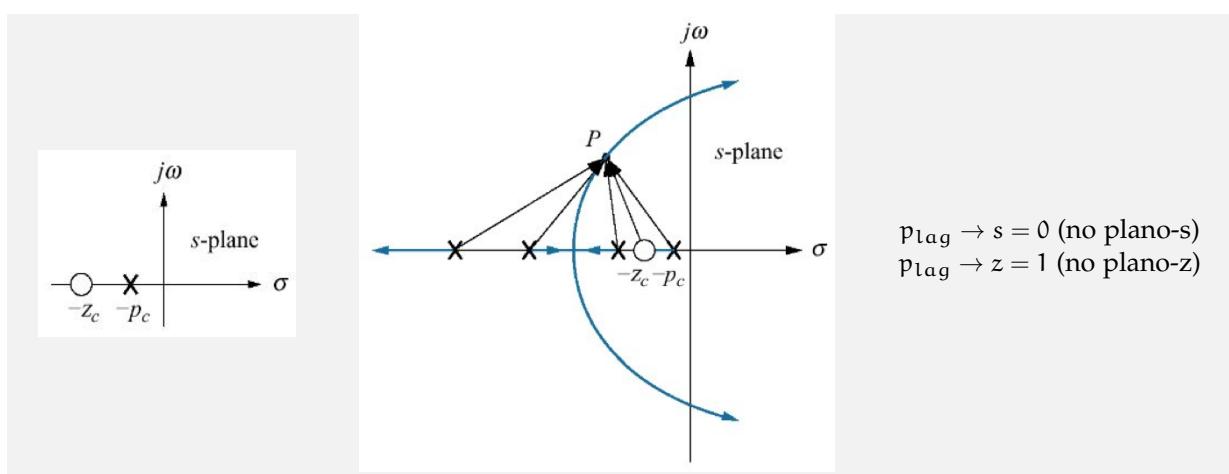


Figura E.4: Característica típica de Controlador por Atraso de Fase.

**PD:** melhora resposta transitória (acelera resposta do sistema), só pode ser implementado por rede ativa no caso de controladores no plano-s (contínuos no tempo).

**Avanço de Fase:** melhora resposta transitória, isto é, acelera, “lead” a resposta do sistema, ao mesmo tempo em que reduz o erro de regime permanente, mas realiza estes objetivos de forma independente. Implica deliberadamente em modificar o RL original do sistema sem controlador. Consiste num zero + um polo (que está o mais próximo possível da origem, no plano-z, que em comparação com o zero, que atua praticamente sozinho (o zero atua como a parte “derivativa” do controlador). No plano-s, o polo está o mais distante possível do eixo imaginário que o zero. Notar que um compensador Detivativo pode ser aproximado por um compensador de avanço de fase. Vantagens com relação ao controlador PD: menor sensibilidade ao ruído (sensores ruidosos). Ver figura E.5. Mas o erro em regime permanente e a resposta transitória não podem ser manipulados de forma independente.

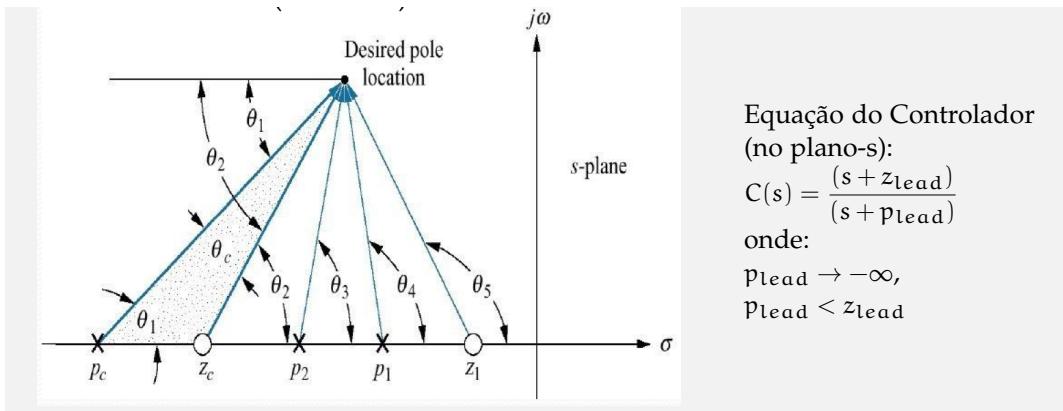


Figura E.5: Efeito característico causado por controlador por Avanço no RL de um sistema.

**PID:** zera o erro de regime permanente e melhora a resposta transitória (acelera o sistema) de forma independente.

**Avanço-Atraso de Fase:** busca mesclar as características dos controladores de Atraso + de Avanço, sem entretanto garantir erro nulo em regime permanente. Neste tipo de controlador, primeiramente se está preocupado em definir a resposta transitória (realizada pela parte que avança a fase) e depois se reduz o erro (através da parte que atrasa a fase), Projeto: ①  $C_{Lead} \rightarrow$  ②  $C_{Lag}$ .

$$C(s) = K \frac{(s + z_{lag})(s + z_{lead})}{(s + p_{lag})(s + p_{lead})}$$



## Apêndice F

# Comandos do MATLAB

<code>abs(x)</code>	Obtém o valor absoluto de $x$ .
<code>angle(x)</code>	Calcula o ângulo em radianos.
<code>atan(x)</code>	Calcula $\arctan(x)$ (sem levar em conta o quadrante resultante).
<code>atan2(y,x)</code>	Determina o ângulo ( $\tan^{-1}(y/x)$ ) correto levando em conta o quadrante onde está localizado o par $(x,y)$ .
<code>axis([xmin,xmax, ymin,ymax])</code>	Fixa os limites de um gráfico no eixo X e Y.
<code>bode(G)</code>	PLota o diagrama de Bode para a função $G(s)$ .
<code>bode(G,w)</code>	Plota o diagrama de Bode para a função $G(s)$ para a faixa de frequências contidas no vetor $w$ .
<code>[theta,rho]=cart2pol(x,y)</code>	Realiza a transformação polar do ponto $(x,y)$ do plano cartesiano para o plano polar resultando num vetor de amplitude $\rho$ com ângulo de inclinação de $\theta$ (radianos): $(x,y) \rightarrow \rho\angle\theta$ .
<code>[numd,dend]=c2dm(num,den,T)</code>	Converte $\frac{\text{num}(s)}{\text{den}(s)}$ para $\frac{\text{num}(z)}{\text{den}(z)}$ usando Sustentador de Ordem Zero (ZOH) e período de amostragem $T$ .
<code>BoG=c2d(G,T)</code>	Converte a <i>transfer function</i> $G(s)$ para $BoG(z)$ usando o Sustentador de Ordem Zero (ZOH) com período de amostragem fixo em $T$ (segundos).
<code>BoG=c2d(G,T,'zoh')</code>	Converte a <i>transfer function</i> $G(s)$ para $BoG(z)$ usando o Sustentador de Ordem Zero (ZOH) com período de amostragem fixo em $T$ (segundos).
<code>Cd=c2d(C,T,'tustin')</code>	Converte a <i>transfer function</i> $C(s)$ para $Cd(z)$ usando o método de Tustin (transformação bilinear do plano-s para o plano-z), usando período de amostragem $T$ (segundos). Útil para “digitalizar” um compensador projetado no domínio frequência, por exemplo, de avanço-atraso (Lead-Lag) para o “mundo digital”.
<code>clc</code>	“Limpa” a janela de comandos do MATLAB e reposiciona cursor para o topo desta janela.
<code>clear</code>	Limpas as variáveis da área de trabalho (“zera” a memória de trabalho do MATLAB).
<code>dcgain(G)</code>	Encontra o ganho DC (estático) para $G(s)$ (isto é, $s = 0$ ) ou para $G(z)$ (isto é, $z = 1$ ).
<code>end</code>	Dentro de uma rotina (arquivo .m) do MATLAB determina o fim do laço de repetição ou do bloco de comparação.
<code>exp(a)</code>	Obtém $e^a$ .
<code>FTMF=feedback(G,H)</code>	Encontra $FTMF = \frac{G}{1+GH}$ considerando realimentação negativa.
<code>FTMF=feedback(G,H,sign)</code>	Encontra $FTMF = \frac{G}{1 \pm GH}$ . $sign$ determina o tipo de realimentação adotado, por padrão o MATLAB considera $-1$ (realimentação negativa), mas o usuário pode especificar uma realimentação positiva se for o caso.
<code>get(BoG)</code>	Apresenta na tela os atributos (ou propriedades) de um objeto no MATLAB. Se $BoG$ for do tipo <i>transfer function</i> retorna os atributos deste tipo de objeto.
<code>grid</code>	Acrescenta quadriculado nas janelas gráficas atuais.
<code>hold on</code>	Ativa sobreposição dos próximos comandos gráficos na janela atualmente ativa.
<code>hold off</code>	“Desliga” sobreposições de gráficos numa mesma janela gráfica.
<code>imag(polo)</code>	Recupera a parte imaginária do polo.
<code>a=input('str')</code>	Permite a entrada (digitação pelo teclado) de valores para a variável $a$ , acompanhado da mensagem $str$ ( <i>string</i> ).
<code>inv(P)</code>	Encontra a inversa da matriz $P$ (quando há).
<code>length(P)</code>	Retorna a dimensão do vetor $P$ .
<code>log(x)</code>	Calcula o logaritmo natural de $x$ (ou $\log_e x$ ).
<code>log10(x)</code>	Calcula o logartimo na base de $x$ (ou $\log_{10} x$ ).
<code>margin(G)</code>	Encontra as margens de ganho e de fase para a <i>transfer function</i> $G(s)$ .
<code>max(P)</code>	Encontra o valor máximo contido dentro do vetor $P$ .



<code>nichols(G,w)</code>	Desenha o diagrama de Nichols para a <i>transfer function</i> $G(s)$ sobre a faixa de frequências determinada pelo vetor $w$ . O parâmetro $w$ é opcional.
<code>nyquist(G,w)</code>	Desenha o diagrama de Nyquist para a <i>transfer function</i> $G(s)$ sobre a faixa de frequências determinada pelo vetor $w$ . O parâmetro $w$ é opcional.
<code>[num,den]=ord2(wn,z)</code>	Cria um sistema de segunda-ordem, $G(s) = \frac{1}{[s^2 + 2\zeta\omega_n s + \omega^2]}$ .
<code>plot(t1,y1, t2,y2, t3,y3)</code>	Plota $x_1$ versus $t_1$ , $y_2$ versus $t_2$ , $y_3$ versus $t_3$ no mesmo gráfico.
<code>pole(G)</code>	Encontra os pólos para a <i>transfer function</i> $G(s)$ .
<code>den=poly([-p1 -p2 -p3])</code>	Fomra o polinômio $den = (s + p_1)(s + p_2)(s + p_3)$ ( $s$ para o plano-s, ou $z$ para o plano-z, depende o uso posterior de <code>den</code> ).
<code>polyval(P, a)</code>	Substitui o valor $a$ sobre o polinômio $P$ .
<code>real(polo)</code>	Encontra as partes reais do vetor <code>polo</code> .
<code>residue(numf,denf)</code>	Encontra os resíduos (expansão em frações parciais) para $F(s) = \frac{numf(s)}{denf(s)}$ .
<code>[K,polos_mf]=rlocfind(FTMA)</code>	Permite uma seleção interativa de um ponto do diagrama do lugar das raízes (não desenhado por esta função) para a <i>transfer function</i> $FTMA$ , retornando o ganho de malha fechada $K$ e a correspondente localização dos pólos de malha fechada <code>polos_mf</code> .
<code>rlocus(FTMA, K)</code>	Gera o gráfico corresponde ao diagrama do Lugar das Raízes ( <i>root locus</i> ) para a <i>transfer function</i> $FTMA$ dentro da faiza de valores de ganho (para malha fechada) especificados pelo vetor $K$ . O parâmetro $K$ é opcional.
<code>roots(den)</code>	Encontra as raízes do polinônimo (vetor) <code>den</code> .
<code>FTMA=series(C,BoG)</code>	Determina a <i>transfer function</i> resultante da álgebra de blocos em série para: $FTMA(z) = C(z) \cdot BoG(z)$ (neste caso no plano-z, mas serve igualmente para o plano-s).
<code>sgrid(z,wn)</code>	Sobrepõe linhas guia de $z(\zeta)$ e $wn(\omega_n)$ sobre um diagrama de lugar das raízes – neste caso no plano-s (especificamente).
<code>zgrid(z,wn)</code>	Sobrepõe linhas guia de $z(\zeta)$ e $wn(\omega_n)$ sobre um diagrama de lugar das raízes – neste caso no plano-z (especificamente).
<code>sin(x)</code>	Encontra o $\sin(x)$ . $x$ em radianos.
<code>sqrt(a)</code>	Calcula $\sqrt{a}$ .
<code>step(G1,G2,... Gn)</code>	Plota as respostas das <i>transfer function</i> $G_1$ até $G_n$ num único gráfico. Os parâmetros $G_2$ até $G_n$ são opcionais.
<code>subplot(xyz)</code>	Organiza uma janela gráfica no formato da matriz especificado pelos valores de $x$ (linhas) $\times$ $y$ (colunas), deixando a mesma pronta para aceitar os próximos comandos gráficos à partir da célula $z$ .
<code>text(x,y,'str')</code>	Acrescenta o texto ( <i>string</i> ) <code>str</code> na janela gráfica atualmente ativa, na posição $(x, y)$ .
<code>[K,p,z]=tf2zp(numg,deng)</code>	Encontra o ganho $dc$ (estático), ( $K$ ), pólos ( $p$ ) e zeros ( $z$ ) de $G(s) = \frac{numg}{deng}$ . Esta função não funciona sobre objetos <i>transfer functions</i> do MATLAB.
<code>C=tf(num,den,T)</code>	Cria um objeto <i>transfer function</i> $C(z) = \frac{num(z)}{den(z)}$ , neste caso, no plano-z já que foi fornecido o terceiro parâmetro $T$ , período de amostragem, adotado para esta função transferênciia. O terceito parâmetro ( $T$ ) é opcional e quando não fornecido cria uma função transferênciia no plano-s. O MATLAB agrega aos atributos do objeto <i>transfer function</i> a informação sobre se a mesma trabalha no plano-s ou no plano-z. Este “conhecimento” é o que permite o MATLAB executar de forma diferenciada (aplicar diferentes métodos de resolução em) comandos como <code>step</code> ou <code>rlocus</code> conforme a <i>transfer function</i> se encontre sobre o plano-s ou plano-z.
<code>[num,den,T]=tfdata(FTMA,'v')</code>	Extraí o numerador ( <code>num</code> ) e denominador ( <code>den</code> ) da <i>transfer function</i> $FTMA(z) = \frac{num(z)}{den(z)}$ , neste caso, no plano-z, já que também retornará o período de amostragem previsto para esta função de transferênciia, $T$ (segundos). O terceito parâmetro de retorno para esta função ( $T$ ) não será necessário se a <i>transfer function</i> em questão trabalhar sobre o plano-s.
<code>title('str')</code>	Coloca o título ( <i>string</i> ) <code>str</code> na janela gráfica atualmente ativa.
<code>xlabel('str')</code>	Coloca a ( <i>string</i> ) <code>str</code> no eixo $x$ da janela gráfica atualmente ativa.
<code>xlabel('str')</code>	Coloca a ( <i>string</i> ) <code>str</code> no eixo $y$ da janela gráfica atualmente ativa.
<code>zpk(numg,deng)</code>	Mostra na janela de comandos, o ganho, pólos e zeros da função $G(s) = \frac{numg(s)}{deng(s)}$ – neste caso, esta função não está trabalhando com <i>transfer function</i> .
<code>zpk(BoG)</code>	Mostra na janela de comandos, de forma fatorada, o ganho, pólos e zeros da função $BoG(z)$ – neste caso, o parâmetro único de entrada <code>BoG</code> deve ser uma <i>transfer function</i> .
<code>C=zpk(z,p,k,Ts)</code>	Cria um objeto do tipo <i>transfer function</i> $C(z)$ cujo numerador é formado pelos zeros fornecidos pelo vetor <code>z</code> , o denominador é formado pelos pólos especificados no vetor <code>p</code> , obedecendo ao ganho $dc$ (estático) <code>k</code> e trabalhando com período de amostragem de <code>Ts</code> (segundos). Se o quarto parâmetro (de entrada) não for fornecido (o <code>Ts</code> neste caso), o MATLAB entende que deve criar uma <i>transfer function</i> no plano-s.

`lsim(sys,u,t)` **Simula** no tempo contínuo a resposta de um sistema (contínuo ou discreto), `sys` (*transfer function*) linear para a entrada arbitrária `u` (tantas colunas quantas entradas possuir o sistema `sys`) no tempo contínuo `t` (em segundos).

Exemplo: Excitando um sistema (`sys`) com uma entrada sinusoidal:

```
>> t = 0:0.01:5;
>> u = sin(t);
>> lsim(sys,u,t)
```

`[y,t]=lsim(sys,u,t)` **Simula** no tempo contínuo a resposta de um sistema (contínuo ou discreto), `sys` linear para a entrada arbitrária `u` no tempo contínuo `t` (em segundos), retornando a resposta do sistema na variável `y` nos instantes de tempo `t`.

Exemplo:

```
>> t=0:0.1:10;
>> alpha=2;
>> ramp=alpha*t; % Gerando sinal de entrada
>> model=tf(1,[1 20 3]); % Função transferência do sistema
>> [y,t]=lsim(model,ramp,t);
>> plot(t,y)
```



# Apêndice G

## Gráficos de $u[kT]$ e $e[kT]$

### G.1 Gráfico de $u[kT]$

O gráfico de  $u[kT]$  pode ser obtido através da linha de comandos do MATLAB adotando-se um expediente especial em relação ao comando `step( )`.

Notar que:

$$U(z) = C(z) \cdot E(z)$$

onde  $C(z)$  corresponde à função transferência do controlador e  $E(z)$  à função transferência do erro.

Porém, sabemos que:  $E(z) = R(z) - Y(z)$  e sabemos ainda que:  $Y(z) = U(z) \cdot BoG(z)$ , então:

$$\begin{aligned} U(z) &= C(z) \cdot [R(z) - Y(z)] \\ &= C(z) \cdot [R(z) - U(z) \cdot BoG(z)] \\ &= C(z) \cdot R(z) - U(z) \cdot \underbrace{C(z) \cdot BoG(z)}_{FTMF(z)} \\ &= \frac{C(z) \cdot R(z)}{1 + C(z) \cdot BoG(z)} \\ U(z) &= \left[ \frac{C(z)}{1 + C(z) \cdot BoG(z)} \right] \cdot R(z) \end{aligned} \quad (G.1)$$

Notar que quando realizamos no MATLAB um comando como: `step(FTMF)` estamos na realidade realizando:

$$y[kT] = Z^{-1} \left\{ \underbrace{U(z)}_{Z\{\text{Degrau}\}} \cdot FTMF(z) \right\}$$

onde  $U(z)$  neste caso, é a transformada  $Z$  de um degrau unitário. Então realizar no MATLAB a declaração:

$$\text{step}(FTMF) = Z^{-1} \{ Z\{\text{Degrau}\} \cdot FTMF(z) \}$$

Isto significa que se agora queremos o gráfico de  $u[kT]$ , levando em conta a eq.(G.1), basta realizar:

```
>> aux = C/(1 + K*C*BoG); % faz com que o MATLAB avalie a eq. para U(z)
>> figure % abre nova janela gráfica
>> step(aux)
```

Podemos testar esta abordagem sobre a plata adotada para o estudo de caso. Seja:

$$G(s) = \frac{1}{(s+10)(s+2)(s+1)}$$

que foi amostrada à  $T = 0,1$  segundos, rendendo a versão digitalizada:

$$BoG(z) = \frac{0.00012224(z+2.747)(z+0.1903)}{(z-0.9048)(z-0.8187)(z-0.3679)}$$

O projeto de um controlador por avanço de fase para esta planta resultou em:

$$C(z) = 183.6605 \cdot \frac{(z - 0.84)}{(z - 0.2)}$$

e a  $FTMA(z)$  resultou em:

$$FTMA(z) = \frac{183,6605 \cdot 12,224 \times 10^{-5} (z + 2.747)(z - 0.84)(z + 0.1903)}{(z - 0.9048)(z - 0.8187)(z - 0.3679)(z - 0.2)}$$

Estes dados todos podem ser recuperados com facilidade se ao final do projeto foi utilizado o comando `save <filename>` para gravar um arquivo de dados contendo as variáveis de trabalho usadas no MATLAB (Workspace). Para recuperar estes dados basta usar o comando `load <filename.mat>`. Neste caso, por exemplo:

```
>> load planta_05set2017.mat % recupera o workspace com dados do Cont. de Avanço (Lead1)
>> who % este comando permite verificar os dados recuperados, entre eles:
```

Your variables are:

```
BoG          FTMF_C_PI_Zero
C_Lag1       FTMF_C_PI_cancela
C_Lag2       G
C_Lag3       K
C_Lead1      K_C_Lag1
C_PI         K_C_Lead1
... % etc...
>>
```

Uma vez recuperados os dados associados com o projeto de um controlador ou terminado seu projeto, o gráfico de  $u[kT]$  para uma entrada degrau pode ser facilmente obtido realizando-se:

```
>> aux = (K_C_Lead1*C_Lead1)/(1+K_C_Lead1*C_Lead1*BoG); % determinando a eq. para U(z)
>> figure; % abre nova janela gráfica
>> step (K_degrau_lead1*aux) % obtendo o gráfico de u[kT]
>> title('Actuator signal (Lead1)')
```

O script anterior gera o gráfico mostrado na figura G.1.

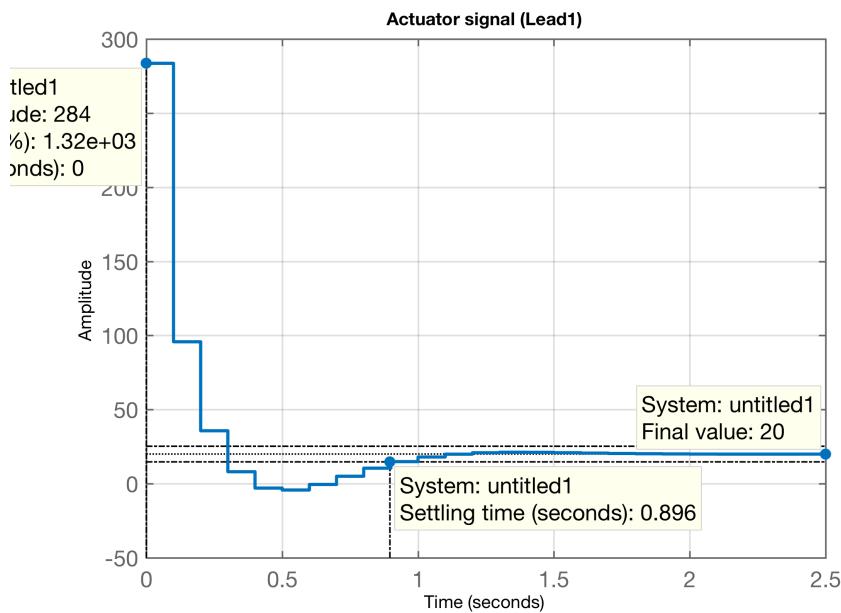


Figura G.1: Gráfico de  $u[kT]$  para um controlador por Avanço de Fase.

Notar que o valor em regime permanente de  $u[kT]$  também pode ser previsto na janela de comandos do MATLAB, lembrando do **teorema do valor final**:

$$y(\infty) = \lim_{k \rightarrow \infty} y[kT] = \lim_{z \rightarrow 1} (1 - z^{-1})F(z)$$

Neste caso, queremos descobrir  $u(\infty)$  para uma entrada degrau, então, necessitamos determinar:

$$u(\infty) = \lim_{z \rightarrow 1} \underbrace{\left( \frac{z-1}{z} \right)}_{(1-z^{-1})} \cdot \underbrace{\left[ \frac{z}{z-1} \right]}_{z\{\text{Degrau}\}} \cdot \underbrace{\left[ \frac{C(z)}{1 + FTMA(z)} \right]}_{U(z)}$$

o que no MATLAB pode ser realizado através do comando `dcgain(.)`. Note que `dcgain(.)` realiza justamente:

$$\text{dcgain}(F) = \lim_{z \rightarrow 1} \underbrace{\left( \frac{z-1}{z} \right)}_{(1-z^{-1})} \cdot \underbrace{\left[ \frac{z}{z-1} \right]}_{z\{\text{Degrau}\}} \cdot F(z)$$

no caso desta planta, note que `dchgain(.)` resulta em:

```
>> dcgain(K_degrau_lead1*aux)
ans = 20.0000
>>
```

## G.2 Gráfico de $e[kT]$

De maneira semelhante ao que foi realizado no item anterior, é possível ser obter mais facilmente um gráfico de  $e[kT]$  sem necessidade de montar um modelo no Simulink.

A função transferência do erro é dada por:

$$E(z) = R(z) - Y(z)$$

mas sabemos que:  $Y(z) = E(z) \cdot C(z) \cdot BoG(z)$ , então:

$$\begin{aligned} E(z) &= R(z) - E(z) \cdot C(z) \cdot BoG(z) \\ E(z) [1 + C(z)BoG(z)] &= R(z) \\ E(z) &= \frac{R(z)}{1 + C(z) \cdot BoG(z)} \\ &= \left[ \frac{1}{1 + C(z) \cdot BoG(z)} \right] \cdot R(z) \end{aligned} \tag{G.2}$$

assim, no MATLAB, realizamos:

$$e[kT] = \text{step} \left( \frac{1}{1 + C(z) \cdot BoG(z)} \right)$$

Testando sobre a planta usada como estudo de caso, ocorre:

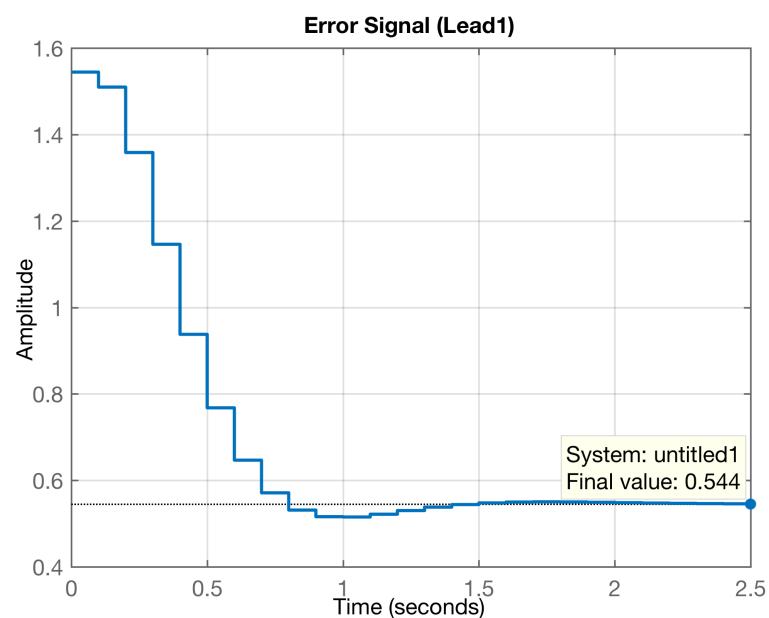
```
>> load planta_05set2017.mat
>> aux2=1/(1+K_C_Lead1*C_Lead1*BoG);
>> step (K_degrau_lead1*aux2)
>> grid
>> title('Error Signal (Lead1)')
```

que resulta no gráfico mostrado na figura G.2. Notar que da mesma forma que realizado no item anterior, é possível na janela de comandos do MATLAB prever o valor do erro em regime permanente:

$$e(\infty) = e[kT]|_{k \rightarrow \infty} = \text{dcgain} \left( \frac{1}{1 + C(z) \cdot BoG(z)} \right)$$

ou neste caso em particular:

```
>> dcgain(K_degrau_lead1*aux2)
ans = 0.5445
>>
```

Figura G.2: Gráfico de  $e[kT]$ .

# Referências Bibliográficas

- Gene F. Franklin, J. David Powell, and Abbas Emami-Naeini. *Feedback Control of Dynamic Systems*. Addison-Wesley Publishing Company, 1994. Biblioteca da UPF: Número de Chamada: 681.51 F831f 3.ed. 1994 (2 exemplares).
- Katsuhiko Ogata. *Discrete-Time Control Systems*, 2nd. ed. Upper Saddle River: Prentice Hall, 1995. URL [http://een.iust.ac.ir/profs/Jahed/digital%20control/e%20book/discrete-time-control\\_systems.pdf](http://een.iust.ac.ir/profs/Jahed/digital%20control/e%20book/discrete-time-control_systems.pdf). (Acessado em 14 set 2013). Biblioteca da UPF: Número de Chamada: 681.5 O34di 2.ed.-1995 (2 exemplares).
- Norman S. Nise. *Control Systems Engineering* – 6th. ed. John Wiley & Sons, Inc, 2011.
- J. A. Cadzow. *Discrete Time Systems: An Introduction with Interdisciplinary Applications*. Prentice Hall International, 1973.
- Benjamin C. Kuo. *Digital Control Systems*, 2nd. ed. Saunders College Publishing / Harcourt Brace Jovanovich College Publisher, 1992. Biblioteca da UPF: Número de Chamada: 681.51 K96d 2.ed.-1992 (1 exemplar).
- Charles L. Phillips and H. Troy Nagle. *Digital Control Systems Analysis and Design*, 2nd. ed. Prentice-Hall International Inc., 1994.
- Indrani Kar and S. Majhi. Course: Digital control systems. [http://nptel.iitm.ac.in/courses/108103008/PDF/module4/m4\\_lec2.pdf](http://nptel.iitm.ac.in/courses/108103008/PDF/module4/m4_lec2.pdf). Dept. of Electronics and Electrical Eng., Indian Institute of Technology Guwahati, Guwahati, Assam, India.(Acessado em 7 nov 2013).
- Norman S. Nise. *Control Systems Engineering* – 6th. ed. LTC, 2012. Biblioteca da UPF: Número de Chamada: 681.5 N724e 6.ed.-2012 (4 exemplares).
- Karl Johan Åström and Björn Wittenmark. *Computer Controlled Systems: theory and design*, 2nd. ed. Prentice Hall International, 1990. Biblioteca da UPF: Número de Chamada 681.51 A859c 2.ed.-1990 (1 exemplar).
- Dale E. Seborg, Thomas F. Edgar, and Duncan A. Mellichamp. *Process Dynamics and Control*. John Wiley & Sons, 1989.
- Cairo Lúcio Nascimento Jr. and Takashi Yoneyama. *Inteligência Artificial em Controle e Automação*. Editora Edgard Blücher, 2000.
- Richard J. Vaccaro. *Digital Control: A State-Space Approach*. McGraw-Hill, Inc., 1995. ISBN 0-07-066781-0. Biblioteca da UPF: Número de Chamada: 681.51 V114d 1995 1a.ed. 1995 (2 exemplares).

# Índice Remissivo

- $E(z)$ , 8
- $\text{FTMF}(z)$ , 8
- $\text{FTMF}(z), H(s)$ , 19
- $H(s)$ , conversor de unidades, 8
- $K_a$ , ganho de aceleração, 15
- $K_p$ , ganho de posição, 12
- $K_v$ , ganho de velocidade, 13
- Setpoint*, 87
- ação integral, 9
- BIBO estável, 19
- Compensador, 87
- Contralabilidade, 89
- Controlador, 87
- Controle robusto, 90
- Critério da magnitude
  - Exemplo de uso, 41
  - Ganho máximo, 41
- Critério de Estabilidade
  - Método de Jury, 24
  - Routh-Hurwitz, 24
- Definições
  - Compensador, 87
  - Controlador, 87
  - Regulador, 87
  - Servo-Controlador, 87
  - Setpoint problem, 87
- Diagrama do Lugar das Raízes, 32
- Efeitos do Período de Amostragem, 29
- Equação característica,  $EC(z)$ , 24
- Erro
  - Ganho de aceleração, 15
  - Ganho de posição, 12
  - Ganho de velocidade, 13
- Estabilidade, 19
- Ganho crítico, 41
  - Determinação, 41
- Ganho Máximo, 30
- Ganho máximo
  - Determinação, 41
  - Método de Jury, 50
- Integração
  - retangular, 9
  - trapezoidal, 10
- MATLAB
- `dlsim`, 16
- `iztrans`, 101
- `lsim`, 16
- `sym`, 101
- Cálculo de  $\zeta$ , 109
- Transformada Inversa de  $Z$ , 101
- Variáveis Simbólicas, 101
- Observador de estados, 91
- Otimização, 91
- Problema de regulação, 87
- Projeto de Servo, 87
- Regras para RL, 33
- Regulação, 85
- Regulador, 87
- Regulador dinâmico, 91
- Respostas Pólos Reais em  $Z$ , 99
- RL
  - Critério da Magnitude, 33
- RL ou *Root Locus*, 32
- Seguimento, 85
- Servo, 87
- Setpoint problem, 87
- Tabela Resumo Teoria de Erros, 15
- Teorema Valor Final, 95
- Tracking, 85
- Transformação bilinear
  - Método de Tustin, 76