

INSTITUTO DE TECNOLOGIA DA UFPA
FACULDADE DE ENGENHARIA ELÉTRICA E BIOMÉDICA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA
ELÉTRICA

Antonio da Silva Silveira

INTRODUÇÃO AO CONTROLE DIGITAL, MODERNO E
PREDITIVO

Belém – PA

ASILVEIRA.UFPA.BR | © 2024 ITEC-UFPA

SUMÁRIO

Apresentação do conteúdo deste “apostilivro”	5
1 CAPÍTULO 1	7
1.1 INTRODUÇÃO AOS SISTEMAS DE TEMPO DISCRETO .	7
1.2 TRANSFORMADA Z	11
1.3 APROXIMAÇÃO PELO SEGURADOR DE ORDEM ZERO	16
1.4 EQUAÇÕES A DIFERENÇAS	19
1.4.1 Solução de equações a diferenças homogêneas	20
1.4.2 Estabilidade e as equações a diferenças	24
1.4.3 Relação entre os domínios z e s	25
1.5 ESTABILIDADE DE SISTEMAS DISCRETOS	26
1.5.1 Critério de Jury	27
1.6 TEOREMA DO VALOR FINAL E TEOREMA DO VALOR INICIAL	29
1.7 EQUIVALENTES DISCRETOS A PARTIR DE APROXIMAÇÕES	30
1.7.1 Aproximações tipo Forward Difference, Backward Difference e Tustin (Bilinear)	31
1.7.2 Efeitos da amostragem em sistemas obtidos por aproximações	35
1.8 MODELAGEM EXPERIMENTAL DE SISTEMAS DINÂMICOS	37
1.8.1 Sistemas de primeira ordem com atraso – FOPDT	38
1.8.2 Sistemas de segunda ordem com atraso – SOPDT	39
1.8.3 Modelagem via Estimador dos Mínimos Quadrados	42
1.8.4 Otimalidade do estimador MQ	46
1.9 PROBLEMAS	48
2 CAPÍTULO 2	51
2.1 INTRODUÇÃO	51
2.2 CONTROLADORES DIGITAIS BASEADOS EM CONTROLADORES ANALÓGICOS	52
2.2.1 PID digital via aproximação <i>Backward</i>	56
2.2.2 PID digital via aproximação <i>Forward</i>	57
2.2.3 PID digital via aproximação de Tustin	58
2.2.4 PID com filtro na parcela derivativa: síntese via Tustin ...	62
2.2.5 PID com filtro na parcela derivativa: síntese <i>Backward</i> ..	66
2.3 SINTONIA DE CONTROLADORES DIGITAIS BASEADA EM CONTROLADORES ANALÓGICOS	67
2.3.1 Sintonia PID via Controle por Modelo Interno	68

2.3.2	Controladores digitais baseados no domínio w	69
2.3.3	Sintonia PID baseada em modelo discreto	77
2.4	COMPENSADORES AVANÇO E ATRASO	79
2.4.1	Compensador avanço-atraso de primeira ordem	81
2.4.2	Conceitos de estabilidade e desempenho robustos	81
2.5	PERTURBAÇÕES DETERMINÍSTICAS E ESTOCÁSTICAS	85
2.5.1	Perturbação de carga	85
2.5.2	Perturbação tipo ruído de medição	87
2.5.3	Perturbação estocástica	88
2.6	PROBLEMAS	88
3	CAPÍTULO 3	91
3.1	INTRODUÇÃO	91
3.2	SISTEMAS MIMO EM ESPAÇO DE ESTADOS	95
3.3	RELAÇÃO ENTRE ESPAÇO DE ESTADOS E FUNÇÃO DE TRANSFERÊNCIA	96
3.4	ESCOLHA DAS VARIÁVEIS DE ESTADO	97
3.5	MATRIZ DE TRANSIÇÃO DE ESTADOS	101
3.6	CONTROLE POR REALIMENTAÇÃO DE ESTADO	101
3.6.1	Controlabilidade	102
3.6.2	Observabilidade	103
3.6.3	Regulação via realimentação de estado	104
3.6.4	Determinação do ganho de realimentação pelo método de Ackermann	107
3.7	OBSERVADOR DE ESTADO	108
3.7.1	Teste de observador de estado	112
4	CAPÍTULO 4	115
4.1	INTRODUÇÃO	115
4.2	PROJETO DE CONTROLADORES NO ESPAÇO DE ESTADOS	116
4.3	GENERALIZAÇÃO DO PROCEDIMENTO DE AUMENTAÇÃO PELA INCLUSÃO DE INTEGRADOR	125
4.3.1	Augmentação pela variação do sinal de controle, $u_a(t)$: ...	125
4.4	COMPENSADOR DINÂMICO E O PRINCÍPIO DA SEPARAÇÃO	136
4.4.1	Compensador dinâmico de regulação	137
4.4.2	Compensador dinâmico no caso servo	138
4.5	COMPENSADORES DINÂMICOS DIGITAIS	139
4.5.1	Regulador digital via realimentação de estado estimado ...	139
4.5.2	Augmentação de sistemas discretos no espaço de estado ...	140
4.5.3	Compensador dinâmico digital no caso servo	141

5 CAPÍTULO 5	143
5.1 INTRODUÇÃO	143
5.2 PRELIMINARES SOBRE SISTEMAS DINÂMICOS LINE- ARES	144
5.2.1 Linearidade	144
5.2.2 Linearização local	145
5.2.3 Linearização local no espaço de estados	148
6 CAPÍTULO 6	161
6.1 INTRODUÇÃO AO CONTROLE PREDITIVO	161
6.2 MODELOS COM ATRASO OU TEMPO MORTO	163
6.3 MODELOS COM PERTURBAÇÕES DETERMINÍSTICA E ESTOCÁSTICA	168
6.4 ESTIMADOR DOS MÍNIMOS QUADRADOS RECURSIVO	178
6.5 CONTROLE PREDITIVO BASEADO NO PREDITOR DE SMITH	187
6.6 CONTROLE DE VARIÂNCIA MÍNIMA GENERALIZADO	189
6.6.1 GMVC de ordem mínima: caso posicional	197
6.6.2 GMVC de ordem mínima: caso incremental	205
6.7 GMVC NO ESPAÇO DE ESTADO	212
6.7.1 GMVSS: caso incremental	216
6.7.2 Obtenção de $E(z^{-1})$ e $F(z^{-1})$ do GMVC via GMVSS ...	217
6.8 CONTROLE PREDITIVO DE VARIÂNCIA MÍNIMA	219
6.8.1 Sintonia de N_y no PMVC	220
6.9 CONTROLE POR MATRIZ DINÂMICA	222
6.9.1 Preditor na estratégia DMC	224
6.9.2 Lei de Controle DMC	229
6.10 CONTROLE PREDITIVO GENERALIZADO	231
6.10.1 Preditor de variância mínima usado no GPC	232
6.10.2 Projeto do preditor do GPC com base no GMVSS	238
6.10.3 Lei de controle do GPC	246
REFERÊNCIAS	255

APRESENTAÇÃO DO CONTEÚDO DESTE “APOSTILIVRO”

- Capítulo 1: Sistemas de tempo discreto, a dados amostrados e controlados por computador. Transformada z . “Transformada ZOH”. Equações a diferenças. Estabilidade de sistemas discretos. Características de resposta temporal de sistemas FOPDT e SOPDT. Modelagem e identificação de sistemas.
- Capítulo 2: Controladores PID digitais. Controladores digitais baseados em controladores analógicos. Projeto baseado no domínio w . Compensadores avanço-atraso (lead-lag). Conceitos de estabilidade e desempenho robustos.
- Capítulo 3: Sistemas dinâmicos em espaço de estados: representações, controlabilidade, observabilidade, realimentação de estado.
- Capítulo 4: Projeto de controladores no espaço de estados via alocação de autovalores. Observadores de Estado. Compensadores dinâmicos via realimentação de estado estimado. Controladores digitais no espaço de estados.
- Capítulo 5: (em desenvolvimento) Linearização de sistemas dinâmicos. Controle Linear Quadrático (pendente). Estimação ótima de estado.
- Capítulo 6: Controle preditivo baseado em modelo.

1 CAPÍTULO 1

Sistemas de tempo discreto, a dados amostrados e controlados por computador. Transformada z . “Transformada ZOH”. Características de resposta temporal. Estabilidade de sistemas discretos (Critério de Jury). Modelagem e identificação de sistemas.

1.1 INTRODUÇÃO AOS SISTEMAS DE TEMPO DISCRETO

A teoria de Controle Digital estabelece toda uma nova família de métodos de projeto e de análise de sistemas no domínio de tempo discreto que pode ser estudada independentemente das relações existentes com sistemas contínuos, mas considerando uma parte introdutória dessa teoria é interessante relacioná-la com sistemas de tempo contínuo e revisar os mecanismos que mapeiam os domínios discreto e contínuo.

Um sistema de controle digital enxerga o mundo real pela perspectiva de um computador que trabalha em instantes discretos e com dados amostrados com base na periodicidade do *clock* de processamento do algoritmo (cf. Fig. 1). Dessa maneira, sinais de tempo contínuo $y(t)$ tornam-se sequências numéricas amostradas, $y(k)$, por um conversor A-D (Analogico-Digital) e é desta forma que uma variável medida de um processo pode ser processada pelo algoritmo de controle com a finalidade de gerar um sinal de comando $u(k)$ que deve ser enviado ao processo para mantê-lo sob as condições de operação desejadas.

O envio de um sinal de comando $u(k)$ de volta à planta, por um controlador digital, requer que um dado numérico, na memória do com-

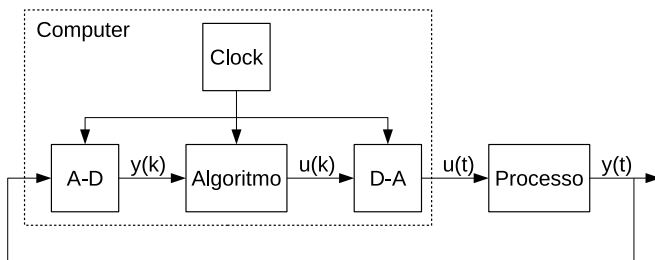


Figura 1 – Diagrama de blocos de um sistema de controle por computador: sistema híbrido, digital e contínuo interligados por conversores digital-analógico (D-A) e analógico-digital (A-D).

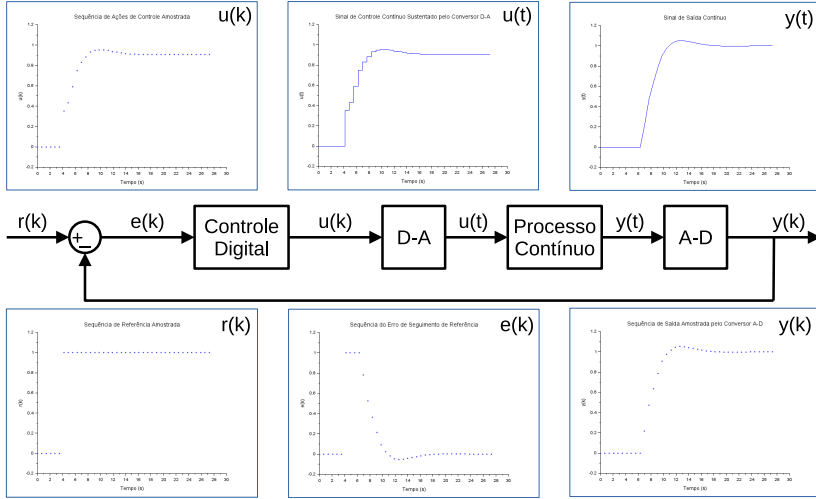


Figura 2 – Diagrama de blocos de um sistema de controle por computador: detalhes relativos aos sinais contínuos e sinais amostrados que trafegam na malha de controle por computador.

putador, torne-se um sinal elétrico (ou outro fisicamente realizável), ou seja, requer a geração do sinal contínuo $u(t)$ pelo uso de um dispositivo de conversão D-A (Digital-Analógico). Como os dados para o conversor D-A chegam em instantes discretos, esses dispositivos de conversão são normalmente construídos com *buffers* que seguram (*hold*) o sinal analógico gerado até que uma nova informação de geração seja requisitada. Tal descrição, do ponto de vista dos sinais contínuos e dos sinais amostrados que trafegam em uma malha de controle por computador, é detalhada na Fig. 2.

A velocidade em que os computadores atuais operam permite considerar que o intervalo entre a conversão A-D, o processamento do algoritmo de controle, até a conversão D-A, pode ser desprezado, mas o intervalo em que o conversor D-A permanece em *hold* até que toda essa sequência se repita para a chegada de uma nova conversão D-A, caracteriza as restrições de projeto e análise de controladores digitais e os diferencia da análise contínua. Nesse período em que o controlador permanece em *hold*, denominado período de amostragem, o processo controlado está em malha aberta (MA), portanto é fundamental que esse intervalo seja apropriadamente selecionado e levado em conta para fins de análise de desempenho e estabilidade da malha de controle.

“O período de amostragem (T_s) é normalmente periódico”. Parece pleonasmo, mas na verdade não é, pois em algumas estratégias de controle digital o período de amostragem pode não ser constante, mas, de todo o modo, tais casos estão fora do escopo deste texto.

O período de amostragem também pode ser representado em função da frequência ou taxa de amostragem ($f_s = 1/T_s$ Hz ou $\omega_s = 2\pi/T_s$ rad/s). Essas informações são os elos entre os domínios de análise contínuo e discreto e permitem estabelecer os mapas de equivalência entre eles através de aproximações e a Transformada z.

A escolha de T_s ou f_s está limitada às condições mínimas exigidas conforme o **Teorema de Amostragem** de *Nyquist (EUA) - Shannon (EUA) - Kotelnikov (RUS) - Whittaker (UK)*, doravante designado neste material simplesmente por Teorema de Amostragem de Nyquist. De maneira simplificada, esse Teorema estabelece que a frequência de amostragem (f_s) deve ser ao menos duas vezes mais veloz que a frequência mais rápida do fenômeno/sinal que se quer digitalizar, para que não se perca informação com a digitalização do sinal. Por exemplo, músicas são comumente digitalizadas em 44,1 kHz, pois a máxima frequência audível por seres humanos seria de 22,05 kHz.

Assumindo que a máxima frequência de interesse de um sinal ou sistema é ω_i rad/s, o Teorema de Amostragem de Nyquist exige que a frequência de amostragem seja:

$$\omega_s \geq 2\omega_i \quad (1.1)$$

e que define também a **frequência de Nyquist**, dada por:

$$\omega_{Nyq} = \frac{\omega_s}{2}. \quad (1.2)$$

A frequência de Nyquist pode ser compreendida ao interpretarmos de forma reversa o próprio Teorema de Nyquist: qualquer que seja a frequência $\omega_s \mid \omega_s \geq 2\omega_i$, apenas as frequências $\omega_d \leq \omega_{Nyq}$ poderão ser armazenadas e reconstruídas. Também, do ponto de vista de um sistema de controle, $\omega_d \leq \omega_{Nyq}$ estabelece o limite espectral para o projeto do controlador. Ou seja, qualquer que seja a solução que se queira estabelecer com o sistema de controle projetado, tal solução deverá ficar necessariamente restrita a $\omega_d \leq \omega_{Nyq}$, pois as frequências acima da frequência de Nyquist não são amostradas.

O Teorema de Nyquist estabelece o critério mínimo de amostragem para que se evite a subamostragem, mas mais adiante veremos que a superamostragem é um problema quando existir pouca memória e/ou

pouco poder de processamento no computador que realiza o procedimento de amostragem. Além disso, não há um teorema que estabeleça um limite superior para a taxa de amostragem, sendo esse um limite tecnológico e definido pelo hardware de aquisição.

No projeto e implementação de sistemas de controle em malha fechada o objetivo não é apenas registrar um sinal digitalmente de forma a permitir a sua posterior reconstrução, mas registrar, avaliar, planejar a ação de controle e aplicar essa ação, repetindo todo esse procedimento a cada T_s período de tempo. Nesse sentido, a definição ou a escolha de T_s não se trata simplesmente de uma questão de amostragem, mas também de como esse período interfere na ação de controle em termos da potência do sinal de controle e consequentemente na eficiência energética, no desempenho e na estabilidade da malha de controle digital. Portanto, no caso de projeto de sistemas de controle digitais, T_s deve ser considerado como um parâmetro adicional de sintonia de controlador.

Devido à importância da adequada seleção de T_s para a eficiência energética, para o desempenho e para a estabilidade de uma malha de controle digital, uma regra que não pode ser violada é aquela estabelecida pelo Teorema de Amostragem de Nyquist, de tal forma que, sempre que possível, e, visando segurança e confiabilidade da malha de controle digital, utilizaremos o seguinte critério:

- **Para um Sistema de Controle em malha fechada:** o critério prático proposto por Astrom e Wittenmark (2011) é de que a frequência de amostragem ω_s seja ao menos o dobro do mínimo exigido pelo Teorema de Nyquist:

$$\omega_s \geq 4\omega_i. \quad (1.3)$$

É importante frisar que **no caso de amostragem**, com a exclusiva finalidade de amostrar um sinal ou sistema cuja máxima frequência de interesse é ω_i , como é o caso da aplicação da Transformada z que estudaremos em seguida, o Teorema de Nyquist é que define a escolha, tal como se mostrou em (1.1).

Exemplo 1.1 *Para o sistema mostrado a seguir, qual é a menor frequência de amostragem, segundo o Teorema de Amostragem de Nyquist, para que seja possível digitalizar o modelo do sistema sem perdas?*

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

Cada pólo determina uma frequência crítica do sistema, sendo

1 rad/s, 2 rad/s, 3 rad/s, essas frequências, respectivamente. Sendo assim, para que não haja perdas na digitalização, a frequência de amostragem ω_s rad/s mínima seria o dobro da maior frequência de interesse, que neste caso é de 3 rad/s. Portanto,

$$\omega_s = 6\text{rad/s} = 2\pi f_s,$$

$$f_s = \frac{3}{\pi}\text{Hz} = \frac{1}{T_s},$$

$$T_s = \frac{\pi}{3}\text{seg}$$

sendo $\frac{\pi}{3}s$ o maior período de amostragem que se poderia utilizar. Analogamente, se pode dizer também que a menor frequência de amostragem que respeitaria o Teorema seria a de $\frac{3}{\pi}\text{Hz}$.

É importante destacar que somente a metade da frequência de amostragem, isto é, $\frac{f_s}{2}\text{Hz}$, pertence à banda amostrada, sendo chamada também de **frequência de Nyquist**, $f_{Nyq} = \frac{f_s}{2}$. Frequências acima de $\frac{f_s}{2}\text{Hz}$ sofrerão distorções, impossibilitando a reconstrução do sinal analógico a partir do sinal digital.

1.2 TRANSFORMADA Z

A transformada z é a análoga discreta da transformada de Laplace. Ou seja, permite a introdução do conceito de função de transferência (FT) e a análise de sistemas discretos no domínio da frequência. Possibilita gerar um mapa de uma sequência temporal em uma função de uma variável complexa z . A transformada z de uma função no tempo, $f(t)$, para $t \geq 0$, ou para uma sequência de valores amostrados, $f(kT_s)$, para valores não-negativos inteiros de k , é definida como

$$F(z) = Z[f(t)] = Z[f(kT_s)] = \sum_{k=0}^{\infty} f(kT_s)z^{-k} \quad (1.4)$$

Expandindo-se o somatório em (1.4) verifica-se que

$$F(z) = f(0) + f(T_s)z^{-1} + f(2T_s)z^{-2} + \dots + f(kT_s)z^{-k} + \dots \quad (1.5)$$

e que os termos z^{-k} nessa série indicam o instante no tempo em que a função $f(kT_s)$ assumiu determinado valor em amplitude.

Usaremos como exemplo uma transformada elementar de um

típico sinal de testes utilizado em controle de processos: o degrau unitário. Considerando que este seja aplicado no instante $t = 0$, sendo este sinal amostrado com $T_s = 1\text{s}$,

$$U(z) = Z[u(k)] = \sum_{k=0}^{\infty} 1z^{-k} \quad (1.6)$$

$$\begin{aligned} U(z) &= 1 + z^{-1} + z^{-2} + \dots \\ U(z) - 1 &= z^{-1} + z^{-2} + \dots \\ \frac{U(z)-1}{z^{-1}} &= \frac{z^{-1}+z^{-2}+\dots}{z^{-1}} = \underbrace{1 + z^{-1} + z^{-2} + \dots}_{U(z)} \\ U(z) - 1 &= z^{-1}U(z) \rightarrow U(z) [1 - z^{-1}] = 1 \end{aligned}$$

$$U(z) = \frac{1}{1 - z^{-1}} = \frac{z}{z - 1}. \quad (1.7)$$

A transformada z da maioria dos sinais e sistemas fisicamente realizáveis já foi tabelada e pode ser encontrada com grande facilidade em livros sobre controle digital e através de uma rápida busca na Internet. Para sistemas de maior ordem e complexidade é recomendável que se utilize softwares como MATLAB, ScicosLab ou SciLab para realizar as transformadas.

A seguir, são fornecidos alguns exemplos da aplicação da transformada z em algumas classes típicas de sistemas dinâmicos lineares:

Exemplo 1.2 *Para o sistema mostrado a seguir, com polos reais e distintos, obtenha a transformada z do sistema usando um período de amostragem que respeite o teorema de amostragem de Nyquist:*

$$G(s) = \frac{20}{s^2 + 9s + 20}$$

Determinando-se os polos do sistema para a realização da expansão em frações parciais:

$$\Delta = 9^2 - 4 \times 20 = 81 - 80 = 1$$

$$s_{1,2} = \frac{-9 \pm \sqrt{\Delta}}{2}$$

$$s_1 = -5$$

$$s_2 = -4$$

Fazendo a expansão em frações parciais, obtém-se:

$$\begin{aligned}
 G(s) &= \frac{5 \times 4}{(s+5)(s+4)} = \frac{A}{(s+5)} + \frac{B}{(s+4)} \\
 A &= \frac{5 \times 4}{(s+5)(s+4)}(s+5) \Big|_{s=-5} = -20 \\
 B &= \frac{5 \times 4}{(s+5)(s+4)}(s+4) \Big|_{s=-4} = 20 \\
 G(s) &= \frac{-20}{(s+5)} + \frac{20}{(s+4)}
 \end{aligned}$$

Verifica-se na tabela de transformada z as respectivas classes de sistemas que são necessárias para se obter $G(z)$ da seguinte forma:

$$G(z) = Z \left\{ \frac{-20}{(s+5)} \right\} + Z \left\{ \frac{20}{(s+4)} \right\}$$

onde $Z\{.\}$ designa a transformada z e $Z^{-1}\{.\}$ a transformada z inversa.
Usando-se

$$\frac{1}{s+a} \xleftrightarrow[T_s]{Z\{.\}} \frac{z}{z - e^{-aT_s}}$$

$$G(z) = \left\{ \frac{-20z}{z - e^{-5T_s}} \right\} + \left\{ \frac{20z}{z - e^{-4T_s}} \right\}$$

Para a escolha do período de amostragem T_s , observa-se que a frequência mais veloz envolvida no sistema é a que está relacionada ao polo em -5 , isto é, de frequência 5rad/s . Sendo assim, a frequência de amostragem que respeita o Teorema de Nyquist seria $\omega_s \geq 2 \times 5\text{rad/s}$. Usando a condição mínima, resulta em

$$\omega_s = 10 = 2\pi f_s$$

$$f_s = 1,5915\text{Hz}$$

$$T_s = \frac{1}{f_s} = 0,6283\text{s}$$

$$G(z) = \left\{ \frac{-20z}{z - 0,0432} \right\} + \left\{ \frac{20z}{z - 0,081} \right\}$$

$$G(z) = \frac{-20z^2 + 1,62z + 20z^2 - 0,864z}{(z - 0,0432)(z - 0,081)}$$

$$G(z) = \frac{0,7560z}{z^2 - 0,1242z + 0,0035}$$

Exemplo 1.3 Para o sistema subamortecido mostrado a seguir, obtenha a transformada z deixando o seu resultado em função do período de amostragem T_s .

$$G(s) = \frac{8}{s^2 + 2s + 4} \quad (1.8)$$

Ao contrário do que foi realizado no exemplo anterior, uma outra possibilidade de se obter a transformada z é a consideração do uso direto da tabela de transformadas para classes de sistemas conhecidos. Para o caso sob análise pode-se aplicar a seguinte relação de transformação descrita em Astrom e Wittenmark (2011):

$$\frac{\omega}{(s+a)^2 + \omega^2} \xrightarrow[T_s]{Z\{\cdot\}} \frac{e^{-aT_s} z^{-1} \sin(\omega T_s)}{1 - 2e^{-aT_s} z^{-1} \cos(\omega T_s) + e^{-2aT_s} z^{-2}}. \quad (1.9)$$

Para a aplicação direta da transformada tabelada, faz-se então a decomposição do sistema na forma exigida, como

$$G(s) = \frac{8}{s^2 + 2s + 4} = K_s \frac{\omega}{(s+a)^2 + \omega^2}. \quad (1.10)$$

O termo K_s incluído se refere ao ganho estático do sistema.

Para se determinar os parâmetros a e ω faz-se a comparação entre os polinômios característicos do sistema e o que consta na transformação tabelada, tal que

$$\begin{aligned} s^2 + 2s + 4 &= s^2 + 2as + (a^2 + \omega^2) \\ 2 &= 2a \rightarrow a = 1 \\ 4 &= (a^2 + \omega^2) \rightarrow \omega = \sqrt{3} \end{aligned} \quad (1.11)$$

De posse do valor de ω determina-se o valor de $K_s = 8\sqrt{3}/3$ e está

pronta a relação de transformação, como segue:

$$G(s) = \frac{8\sqrt{3}}{3} \left[\frac{\sqrt{3}}{(s+1)^2 + (\sqrt{3})^2} \right]. \quad (1.12)$$

Resultando, assim, no seguinte equivalente discreto dado em função do período de amostragem T_s :

$$G(z) = \frac{8\sqrt{3}}{3} \left[\frac{e^{-T_s} z^{-1} \sin(\sqrt{3}T_s)}{1 - 2e^{-T_s} z^{-1} \cos(\sqrt{3}T_s) + e^{-2T_s} z^{-2}} \right]. \quad (1.13)$$

Exemplo 1.4 Aplique a transformada z sobre o sistema com polo múltiplos repetidos mostrado a seguir:

$$G(s) = \frac{s^2 + 2s + 3}{(s+1)^3} \quad (1.14)$$

No caso de polos repetidos a expansão em frações parciais é realizada de forma diferenciada, como segue:

$$G(s) = \frac{s^2 + 2s + 3}{(s+1)^3} = \frac{b_1}{(s+1)^3} + \frac{b_2}{(s+1)^2} + \frac{b_3}{(s+1)}$$

$$b_1 = \left[\left(\frac{s^2 + 2s + 3}{(s+1)^3} \right) (s+1)^3 \right] \Big|_{s=-1} = 1 - 2 + 3 = 2$$

$$b_2 = \frac{1}{1!} \frac{d}{ds} \left[\left(\frac{s^2 + 2s + 3}{(s+1)^3} \right) (s+1)^3 \right] \Big|_{s=-1} = 2s + 2 = 0$$

$$b_3 = \frac{1}{2!} \frac{d^2}{ds^2} \left[\left(\frac{s^2 + 2s + 3}{(s+1)^3} \right) (s+1)^3 \right] \Big|_{s=-1} = \frac{2}{2!} = 1$$

$$G(s) = \frac{s^2 + 2s + 3}{(s+1)^3} = \frac{2}{(s+1)^3} + \frac{0}{(s+1)^2} + \frac{1}{(s+1)}$$

Deste ponto em diante, basta consultar a tabela de transformadas z descritas nas principais obras da literatura de Controle Digital, como em Astrom e Wittenmark (2011), para converter $G(s)$ ao domínio z ,

como segue:

$$\frac{2}{(s+a)^3} \xrightarrow[Z\{\cdot\}]{T_s} \frac{T_s^2 e^{-aT_s} (1 - e^{-aT_s} z^{-1}) z^{-1}}{(1 - e^{-aT_s} z^{-1})^3} \quad (1.15)$$

$$\frac{1}{s+a} \xrightarrow[Z\{\cdot\}]{T_s} \frac{1}{1 - e^{-aT_s} z^{-1}} \quad (1.16)$$

Aplicando-se essas duas transformadas tabeladas sobre o sistema expandido $G(s)$, obtém-se:

$$G(z) = \frac{T_s^2 e^{-T_s} (1 - e^{-T_s} z^{-1}) z^{-1}}{(1 - e^{-T_s} z^{-1})^3} + \frac{1}{1 - e^{-T_s} z^{-1}} \quad (1.17)$$

Como os três polos de $G(s)$ estão localizados em -1 , a frequência de interesse desse sistema opera em 1 rad/s. Portanto, para fins de amostragem a digitalização deste sistema exigiria que a frequência de amostragem fosse $\omega_s \geq 2$ rad/s, atendendo ao Teorema de Amostragem, tal que

$$f_s = \frac{\omega_s}{2\pi} \text{ Hz}$$

$$T_s = \frac{1}{f_s} \leq 3.1415927 \text{ segundo}$$

1.3 APROXIMAÇÃO PELO SEGURADOR DE ORDEM ZERO

O Segurador de Ordem Zero ou Zero-Order Hold (ZOH) é o método de discretização que fornece um equivalente discreto com base no princípio de um sistema de aquisição de dados e controle digital, ou seja, considera o intervalo em que o sistema sob análise encontra-se em **hold**, pois o sinal aplicado na entrada da planta é uma sequência em degrau proveniente do conversor D-A, tal como destacado anteriormente no diagrama de blocos da Fig. 1 e que é simplificado e redesenhado à forma mostrada na Fig. 3 para evidenciar o processo digitalizado, tal como o algoritmo de controle digital o acessa.

Desde o ano de 2013 passei a adotar a seguinte nomenclatura para o método de aproximação via ZOH: **Transformada ZOH**, tal como fizeram Astrom e Wittenmark (2011). Isto se deve ao fato deste modelo de discretização usufruir da própria Transformada z, porém, com a inclusão de um *buffer* 1/s na entrada da processo contínuo antes da aplicação da Transformada z, garantindo assim a inclusão da parte

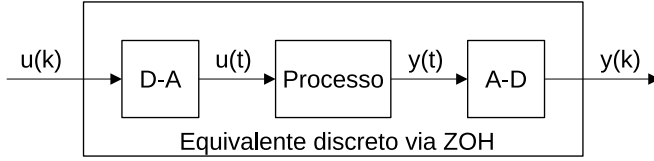


Figura 3 – Equivalente discreto via ZOH de um processo contínuo sob a perspectiva de um computador.

responsável pelo conversor D-A no modelo. Logo, **a Transformada ZOH é específica para a discretização do modelo de processo. Ou seja, tudo que não for algoritmo de controle/estimador/preditor é processo**, do-ravante designarei esta separação de *separação processo-computador*.

O método de transformação via ZOH é realizado da seguinte forma:

$$G(z) = (1 - z^{-1}) Z \left[\frac{G(s)}{s} \right], \quad (1.18)$$

onde $G(s)$ é o processo que se quer discretizar via método ZOH.

A característica *ordem zero* é obtida já que após a Transformada z ser aplicada sobre o modelo com integrador, $G(s)/s$, o inverso do integrador no domínio discreto, $1 - z^{-1}$, é multiplicado sobre a parte $Z \left[\frac{G(s)}{s} \right]$, eliminando a *ordem* adicional, mas mantendo a característica de *causalidade* das etapas A-D e D-A, i.e., a inclusão de 1 (um) atraso de tempo discreto por causa da forma como as etapas de conversão A-D e D-A operam no algoritmo de controle digital, sendo este atraso discreto explicitamente representado pelo z^{-1} no sistema mostrado a seguir (assumido *estritamente próprio*, ou seja, o grau do numerador é menor que o do denominador do sistema):

$$G(z) = \frac{B(z)z^{-1}}{A(z)}, \quad (1.19)$$

em que

$$B(z) := b_0 + b_1 z^{-1} + \dots + b_{n_b} z^{-n_b}, \quad (1.20)$$

$$A(z) := 1 + a_1 z^{-1} + \dots + a_{n_a} z^{-n_a}. \quad (1.21)$$

É importante destacar que alguns autores preferem incluir o z^{-1} explicitado no numerador de (1.19) como parte integrante do polinômio

$B(z)$. No entanto, veremos mais adiante um caso mais geral em que este atraso é simbolizado por $d \geq 1$, representando o atraso (ou tempo morto) discreto, tal que um sistema estritamente próprio com atraso seja dado por:

$$G(z) = \frac{B(z)}{A(z)} z^{-d}. \quad (1.22)$$

Sistemas com atraso e soluções de sistemas de controle preditivo para a compensação do atraso são discutidos com maior profundidade no Capítulo 6.

Exemplo 1.5 *Transformada ZOH sobre um sistema genérico de primeira ordem:*

$$G(s) = \frac{a}{s+a} = \frac{1}{\left(\frac{1}{a}\right)s+1} \rightarrow \frac{G(s)}{s} = \frac{a}{s(s+a)}$$

Fazendo-se a expansão em frações parciais, aplica-se então a Transformada z às partes, tal que

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

que após ser multiplicado por $(1-z^{-1})$ resulta em

$$G(z) = \frac{(1-e^{-aT_s})}{z-e^{-aT_s}} = \frac{(1-e^{-\frac{T_s}{\tau}})}{z-e^{-\frac{T_s}{\tau}}}, \quad \tau = \frac{1}{a}. \quad (1.23)$$

Reescrevendo-se a função de transferência em (1.23) na forma de filtro,

$$G(z^{-1}) = \frac{\left(1-e^{-\frac{T_s}{\tau}}\right)z^{-1}}{1-\left(e^{-\frac{T_s}{\tau}}\right)z^{-1}}, \quad (1.24)$$

fica evidente o atraso z^{-1} no numerador de $G(z^{-1})$, decorrente da inclusão dos conversores D-A e A-D ao modelo do sistema.

Astrom e Wittenmark (2011) apresentam em seu livro uma tabela de *Transformadas ZOH*. Portanto, deve-se ter cuidado, já que o método ZOH é válido para a discretização de sistemas existentes no

mundo físico real e que são acessados por computador. Ou seja, respeitam a *separação processo-computador*. Logo, não se deve confundir a Transformada z com a ZOH, já que esta última acrescenta uma aproximação do sistema A-D, D-A respectivamente conectado à saída e à entrada do processo contínuo.

Em softwares como MATLAB e SCILAB, a Transformada ZOH a partir de um sistema descrito no domínio s pode ser obtida usando-se as respectivas funções: *c2d*, *dscr*. No entanto, o caminho contrário, do discreto para o contínuo, apenas o MATLAB fornece via função *d2c*. Pelo SCILAB tal procedimento não é tão simples, mas é possível de se realizar usando-se a função *horner*.

1.4 EQUAÇÕES A DIFERENÇAS

Sistemas dinâmicos discretos podem ser descritos por equações denominadas *equações a diferenças*. Tomemos como exemplo o seguinte modelo de sistema de segunda ordem, dado pela seguinte função de transferência (FT) no domínio z :

$$G(z) = \frac{Y(z)}{U(z)} = \frac{b_0 z + b_1}{z^2 + a_1 z + a_2},$$

$$(z^2 + a_1 z + a_2) Y(z) = (b_0 z + b_1) U(z). \quad (1.25)$$

Pela transformada z , assumindo a hipótese de que $F(z)$ seja um terminal de entrada ou de saída de um modelo descrito por função de transferência (portanto, condições iniciais nulas), a variável z ora se aplica como um operador de avanço ou atraso de tempo discreto,

$$z^{n_s} F(z) := f(k + n_s) \text{ ou } z^{-n_s} F(z) := f(k - n_s),$$

$$k \in \mathbb{Z}^{0+} : \text{domínio de tempo discreto,} \quad (1.26)$$

$$n_s \in \mathbb{Z} : \text{deslocamento de tempo discreto,}$$

ora se aplica como a variável complexa definida no domínio discreto da frequência $z = e^{sT_s} = e^{j\omega T_s}$. Tal relação será válida desde que se trate de um sistema cujas condições iniciais sejam nulas ou que seja um sistema estável (ASTROM; WITTENMARK, 2011), caso análogo ao da transformada de Laplace em que $s = j\omega$ é a variável complexa definida no domínio contínuo da frequência e s é também simultaneamente um operador de diferenciação no tempo $s := d/dt$.

A equação a diferenças do sistema em (1.25) pode ser obtida pela

aplicação da propriedade de deslocamento no tempo da transformada z , convertendo o sistema para o domínio do tempo discreto, como segue:

$$y(k+2) + a_1y(k+1) + a_2y(k) = b_0u(k+1) + b_1u(k). \quad (1.27)$$

Esta é a equação a diferenças descrita na forma em *avanço*. A forma *filtrada*, em atraso, denominada também de Auto-Regressiva, é obtida recuando-se o tempo discreto n_a -passos para trás, onde n_a é a ordem do modelo e que neste caso $n_a = 2$:

$$y(k) + a_1y(k-1) + a_2y(k-2) = b_0u(k-1) + b_1u(k-2). \quad (1.28)$$

Em (1.28), considere que $y(k)$ é a saída do sistema no instante de tempo discreto k , onde este $k = 0, 1, 2, \dots$ assume somente valores inteiros (tempo discreto), enquanto que $u(k)$ é a entrada desse sistema no instante de tempo k .

Observe que a forma matemática da equação a diferenças em (1.28) é convidativa para implementações em linguagens de programação que lidam com iterações em intervalos discretos, ou seja, que recorrem a dados de memória atuais e passados, como o exemplo da Tabela 1, que corresponde a uma implementação em MATLAB para simular o modelo mostrado em (1.28). Outro exemplo, em linguagem c para Arduino, com o código de um PID digital apresentando a lei de controle como uma equação a diferenças, é fornecido na Tabela 2.

Tabela 1 – Exemplo de código MATLAB para simular um sistema discreto de segunda ordem descrito na forma de equação a diferenças.

```
for k=1:101,
    y(k) = -a1*y(k-1) -a2*y(k-2) +b0*u(k-1) +b1*u(k-2);
    u(k) = 1;
end
```

1.4.1 Solução de equações a diferenças homogêneas

A solução de uma equação a diferenças homogênea, isto é, com entrada nula para todo k ($u(k) = 0 \forall k$), permite a análise das características da resposta $y(k)$ de um sistema dinâmico discreto no tempo.

Consideremos o exemplo mostrado a seguir como forma de in-

Tabela 2 – Código em C para Arduino como exemplo de PID digital embarcado e sua relação com as equações a diferenças.

```
float y = 0.0;
float u[3] = 0.0;
float e[4] = 0.0;
float yr = 0.0;

// Controller parameters
const float ts = 0.1; // Sampling period
const float Kc = 0.8; // Proportional gain
const float Ti = 0.2; // Integral period
const float Td = 1.2; // Derivative period
const float q0 = Kc + (Kc*Td/ts);
const float q1 = -Kc + (Kc*ts/Ti) -(2*Kc*Td/ts);
const float q2 = Kc*Td/ts;

void setup()
{
    Serial.begin(9600);
    Serial.setTimeout(5); // How long to wait till parseFloat is over
}

void loop()
{
    if (Serial.available())
    {
        y = Serial.parseFloat();
        yr = Serial.parseFloat();
        e[0] = yr-y;
        u[0] = u[1] +q0*e[0] +q1*e[1] +q2*e[2]; // <<<<<<< LEI DE CONTROLE
        Serial.print(u[0],4);

        e[2] = e[1]; // Shifting data for the next loop
        e[1] = e[0];
        u[1] = u[0];
    }
}
```

troduzir a solução de equações a diferenças homogêneas.

Exemplo 1.6 *Consideremos o seguinte exemplo de segunda ordem, $n_a = 2$, para uma equação homogênea:*

$$y(k) + a_1 y(k-1) + a_2 y(k-2) = 0. \quad (1.29)$$

Aplicando-se a propriedade da transformada z mostrada em (1.26),

$$(1 + a_1 z^{-1} + a_2 z^{-2}) Y(z) = A(z) Y(z) = 0 \quad (1.30)$$

ou mesmo

$$(z^2 + a_1 z + a_2) Y(z) = A(z) Y(z) = 0. \quad (1.31)$$

Os polos do sistema são as raízes $\lambda_i \in \mathbb{C}$, $i = [1; 2]$, do polinômio característico $A(z)$, calculadas a partir de:

$$\lambda^2 + a_1 \lambda + a_2 = 0. \quad (1.32)$$

A solução para o exemplo de segunda ordem em (1.29) é obtida pelo cálculo das duas raízes da expressão mostrada em (1.32). Já para um sistema de ordem n , a solução pode ser generalizada para

$$y(k) = \sum_i c_i \lambda_i^k, \quad (1.33)$$

onde c_i , $i = 1, 2, \dots, n$, corresponde às condições iniciais em $k = 0, 1, 2, \dots, n_i$, e.g. $y(0)$, $y(1)$, $\dots, y(n_i)$, tal que, no caso de um sistema de segunda ordem, por exemplo assumindo as seguintes condições iniciais,

$$\begin{aligned} y(0) &= 1 \\ y(1) &= 0 \end{aligned}$$

os parâmetros c_1 e c_2 seriam determinados com base nas raízes conhecidas λ_1 e λ_2 , pela solução do seguinte sistema:

$$\begin{aligned} y(0) = 1 &= c_1 (\lambda_1)^0 + c_2 (\lambda_2)^0 \\ y(1) = 0 &= c_1 (\lambda_1)^1 + c_2 (\lambda_2)^1 \end{aligned}$$

Sistemas de ordem maior que dois acarretarão equações de maior grau (equivalente à ordem do modelo) e a complexidade dos cálculos

também aumentará, possivelmente requerendo o auxílio de softwares computacionais específicos em certos casos.

Como as raízes do polinômio no domínio z são as mesmas do polinômio em λ , pode-se relacionar os polos do sistema no domínio z com as características de resposta temporal da saída $y(k)$, permitindo-se caracterizar a estabilidade de um sistema dinâmico com relação à solução da sua equação a diferenças, isto é:

- Sistemas **assintoticamente estáveis** serão aqueles em que as raízes λ_i estão associadas à saída $y(k)$ tendendo a zero quando $k \rightarrow \infty$.
- Sistemas **marginalmente estáveis** serão aqueles em que as raízes λ_i estão associadas à saída $y(k)$ constante ou com oscilação harmônica quando $k \rightarrow \infty$.
- Sistemas **instáveis** serão aqueles em que as raízes λ_i estão associadas à saída $y(k)$ tendendo ao infinito quando $k \rightarrow \infty$.

Exemplo 1.7 Determine a solução da equação a diferenças, homogênea, mostrada a seguir:

$$y(k) + 2y(k-1) - 8y(k-2) = 0, \quad (1.34)$$

$$A(z) = z^2 + 2z - 8 = 0.$$

As raízes de $A(z)$ são $\lambda_1 = 2$, $\lambda_2 = -4$.

Para calcular as raízes no MATLAB ou SCILAB pode-se usar a seguinte função:

$$\text{roots}([1 \ 2 \ -8])$$

Portanto, a solução para o modelo auto-regressivo em (1.34) é dada por

$$y(k) = c_1(2)^k + c_2(-4)^k, \quad (1.35)$$

faltando determinar os parâmetros c_1 e c_2 com base nas condições iniciais da saída $y(k)$. Digamos que $y(0) = y(1) = 1$,

$$\begin{aligned} y(0) = 1 &= c_1(2)^0 + c_2(-4)^0 \\ y(1) = 1 &= c_1(2)^1 + c_2(-4)^1 \end{aligned}$$

logo,

$$\begin{aligned} 1 &= c_1 + c_2 \\ 1 &= 2c_1 - 4c_2 \end{aligned}$$

$$c_1 = 0,833333, \quad c_2 = 0,166666$$

Exemplo 1.8 *Determine a solução da equação a diferenças, **não homogênea** (meramente para ilustrar a diferença para o caso homogêneo), mostrada a seguir:*

$$G(z) = \frac{Y(z)}{U(z)} = \frac{b_0}{z+a_1}$$

$$y(k+1) + a_1 y(k) = b_0 u(k)$$

$$y(k) + a_1 y(k-1) = b_0 u(k-1)$$

$$y(k) = -a_1 y(k-1) + b_0 u(k-1)$$

$$y(1) = -a_1 y(0) + b_0 u(0)$$

$$y(2) = -a_1 [-a_1 y(0) + b_0 u(0)] + b_0 u(1) = (-a_1)^2 y(0) - a_1 b_0 u(0) + b_0 u(1)$$

$$y(3) = -a_1 y(2) + b_0 u(2) = (-a_1)^3 y(0) + (-a_1)^2 b_0 u(0) + (-a_1) b_0 u(1) + b_0 u(2)$$

$$y(k) = (-a_1)^k y(0) + b_0 \sum_{i=0}^{k-1} [(-a_1)^i u(k-i-1)]$$

1.4.2 Estabilidade e as equações a diferenças

Dependendo das raízes λ_i pode-se resumir o comportamento qualitativo da sequência de saídas $y(k)$ conforme apresentado na Tabela 3 para raízes reais e na Tabela 4 para raízes complexas:

Tabela 3 – Comportamento qualitativo de $y(k)$ com base em $\lambda_i \in \mathbb{R}$

λ	Sequência $\{y(k), k \geq 0\}$
$\lambda > 1$	Crescente
$\lambda = 1$	Constante
$0 < \lambda < 1$	Decrescente
$-1 < \lambda < 0$	Decrescente com sinal alternado
$\lambda = -1$	Constante com sinal alternado
$\lambda < -1$	Crescente com sinal alternado

Os resultados dessa análise sobre as soluções de equações a dife-

Tabela 4 – Comportamento qualitativo de $y(k)$ com base em $\lambda_i \in \mathbb{C}$.

λ	Sequência $\{y(k), k \geq 0\}$
$ \lambda < 1$	Decrescente e oscilatória
$ \lambda = 1$	Oscilatória
$ \lambda > 1$	Crescente e oscilatória

renças estão diretamente ligados a análise de estabilidade de sistemas discretos a partir do estudo do **círculo unitário no Plano- z** (cf. Fig. 4). Sendo assim, pode-se resumir que:

- Sistemas com todos os polos contidos no círculo unitário são assintoticamente estáveis;
- Sistemas com polos sobre o círculo unitário são marginalmente estáveis;
- Sistemas com um ou mais polos fora do círculo unitário são instáveis.

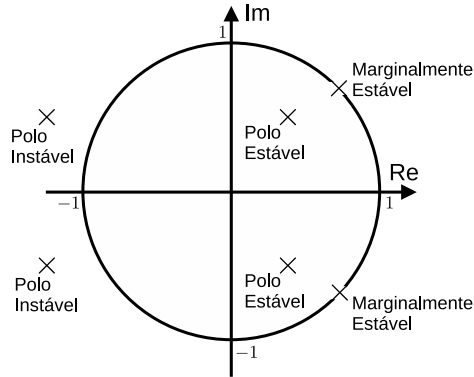


Figura 4 – Círculo unitário no Plano- z : classificação de estabilidade em relação à posição das raízes dos polinômios característicos de sistemas dinâmicos descritos no domínio z .

1.4.3 Relação entre os domínios z e s

A relação entre o domínio da frequência discreto z e o contínuo s pode ser feita a partir da análise da propriedade de deslocamento no

tempo da Transformada s e da Transformada z .

Se considerarmos um sistema linear contínuo qualquer $G(s)$, uma representação desse sistema com um atraso de θ segundos é descrita por

$$G_d(s) = G(s)e^{-\theta s}. \quad (1.36)$$

A Transformada z de $G(s)$ é dada por $Z[G(s)] = G(z)$. A representação desse sistema discreto com atraso de θ_d instantes de tempo discretos, i.e.,

$$\theta_d = \frac{\theta}{T_s}, \quad (1.37)$$

onde T_s é o período de amostragem, é descrita por

$$G_d(z) = Z[G(s)]z^{-\theta_d} = Z[G_d(s)]. \quad (1.38)$$

Desse modo,

$$z^{-\left(\frac{\theta}{T_s}\right)} := e^{-\theta s}, \quad (1.39)$$

tal que a relação entre o domínio z e o domínio s é estabelecida por (1.39) quando $\theta = T_s$ e $s = j\omega$, pois

$$\begin{aligned} z^{-1} &= e^{-j\omega T_s}, \\ z &= e^{j\omega T_s}. \end{aligned} \quad (1.40)$$

Com base na relação mostrada em (1.39), verifica-se que para $z^{-2} = e^{-2j\omega T_s}$, $z^{-3} = e^{-3j\omega T_s}$, ..., $z^{-n} = e^{-nj\omega T_s}$. Sendo assim, o domínio z possui uma relação exponencial com o domínio $s = j\omega$, sendo que ω é ponderado pelo período de amostragem T_s .

1.5 ESTABILIDADE DE SISTEMAS DISCRETOS

Para definirmos o conceito de estabilidade de uma forma geral, consideremos um sistema – possivelmente não-linear e variante no tempo – descrito por

$$x(k+1) = f(x(k), k) \quad (1.41)$$

e que $x^0(k)$ e $x(k)$ são soluções desse sistema (1.41) quando as condições iniciais são, respectivamente, $x^0(k_0)$ e $x(k_0)$. Pode-se então dizer que:

- *ESTABILIDADE: a solução $x^0(k)$ do sistema em (1.41) é estável se para um dado $\varepsilon > 0$ exista um $\delta(\varepsilon, k_0) > 0$ tal que todas as soluções com $\|x(k_0) - x^0(k_0)\| < \delta$ sejam tais que $\|x(k) - x_0(k)\| < \varepsilon$ para $k \geq k_0$. Isto quer dizer que, dado um estímulo impulsivo do tipo delta de Kronecker aplicado no instante k_0 sobre o sistema em (1.41), este será dito estável se houver decaimento de energia entre uma condição futura e uma condição inicial $x_0(k)$, energia esta representada pela norma $\|x(k) - x_0(k)\|$.*

Um conceito de estabilidade mais simples de se analisar é o de **Estabilidade Assintótica** de sistemas lineares, definido a seguir:

-
- *ESTABILIDADE ASSINTÓTICA DE SISTEMAS LINEARES: um Sistema Linear Discreto Invariante no Tempo é assintoticamente estável se e somente se todos os seus pólos/autovalores estiverem localizados dentro do círculo unitário (cf. Tab. 4).*
-

Obs: note também que a estabilidade assintótica implica em estabilidade no senso geral e em BIBO estabilidade (*bounded-input bounded-output* ou entrada-limitada saída-limitada) – caso em que a estabilidade é garantida devido às limitações naturais das variáveis de entrada e saída da planta (e.g., saturações).

Com os mencionados conceitos de estabilidade apresentados, fica evidente que a forma mais simples e eficaz de se avaliar a estabilidade de sistemas lineares discretos é pela determinação de seus pólos. No entanto, existem outras formas de avaliar a estabilidade de sistemas lineares discretos, como o Critério de Jury, que dispensa o cálculo das raízes de sistemas polinomiais; análise do Lugar Geométrico das Raízes; Critério de Nyquist; Diagrama de Bode; via Funções de Lyapunov. Neste texto, enfatizaremos apenas o método por Critério de Jury, enquanto que os métodos gráficos serão utilizados e comentados durante o resto do curso em Controle Digital pelo uso em trabalhos de projeto de controladores.

1.5.1 Critério de Jury

O critério de Jury é a versão discreta do de Routh-Hurwitz, sendo um algoritmo interessante para a análise de estabilidade de sistemas lineares discretos, pois dispensa a verificação dos pólos do sistema sob

análise. Considere o seguinte exemplo (ASTROM; WITTENMARK, 2011):

$$G(z) = \frac{B(z)}{\alpha_0 z^n + \alpha_1 z^{n-1} + \dots + \alpha_n}. \quad (1.42)$$

A tabela de Jury do sistema (1.42), quando $\alpha_0 > 0$, é formada por

$$\begin{array}{cccccc} \alpha_0 & \alpha_1 & \alpha_2 & \cdots & \alpha_{n-1} & \alpha_n \\ \alpha_n & \alpha_{n-1} & \cdots & \alpha_2 & \alpha_1 & \alpha_0 \\ \beta_0 & \beta_1 & \cdots & \beta_{n-2} & \beta_{n-1} & \\ \beta_{n-1} & \beta_{n-2} & \cdots & \beta_1 & \beta_0 & \\ c_0 & \cdots & c_{n-3} & c_{n-2} & & \\ c_{n-2} & c_{n-3} & \cdots & c_0 & & \\ & & & & & \\ & & & & & \\ & & & & & \\ z_0 & z_1 & & & & \\ z_1 & z_0 & & & & \\ & & & & & \\ v_0 & & & & & \end{array} \quad (1.43)$$

onde,

$$\begin{aligned} \beta_0 &= \alpha_0 - \frac{\alpha_n}{\alpha_0} \alpha_n; \quad \beta_1 = \alpha_1 - \frac{\alpha_n}{\alpha_0} \alpha_{n-1}; \quad \cdots; \quad \beta_k = \alpha_k - \frac{\alpha_n}{\alpha_0} \alpha_{n-k} \\ c_k &= \beta_k - \frac{\beta_{n-1}}{\beta_0} \beta_{n-1-k}; \quad \cdots \end{aligned} \quad (1.44)$$

- *O critério de estabilidade de Jury estabelece que, se $\alpha_0 > 0$, então todos os pólos do sistema estarão dentro do círculo unitário se e somente se todos os termos da primeira coluna das linhas ímpares forem positivos.*
- *Se nenhum elemento da primeira coluna das linhas ímpares for nulo, o número de raízes fora do círculo unitário é igual ao número de elementos negativos. Se tiver um ou mais elementos nulos nessa coluna, não há como afirmar o número de pólos instáveis.*
- Uma condição necessária para a garantia de estabilidade é a ve-

rificação de $A(z \rightarrow 1)$ e $A(z \rightarrow -1)$ da seguinte forma:

$$A(1) > 0 \quad (1.45)$$

$$(-1)^n A(-1) > 0 \quad (1.46)$$

onde $A(z)$ é o polinômio característico da função de transferência sob análise, dada por

$$G(z) = \frac{B(z)}{A(z)} \text{ ou } G(z^{-1}) = \frac{B(z^{-1})z^{-1}}{A(z^{-1})}. \quad (1.47)$$

Se o teste em (1.45) **ou** (1.46) falhar, o sistema é instável. No entanto, se ambos os testes derem certo, ainda é necessário construir a tabela de Jury para completar a verificação da estabilidade.

Exemplo 1.9 *Aplicação do critério de Jury sobre um sistema de primeira ordem:*

$$\frac{Y(z)}{Y_r(z)} = \frac{(1 - z_d)}{z - z_d}. \quad (1.48)$$

A tabela de Jury para a função em (1.48) é construída com base em (1.42) e (1.43), sendo esta

$$\begin{array}{rcc} & z^1 & z^0 \\ \text{linha 1} & 1 & -z_d \\ \text{linha 2} & -z_d & 1 \\ \text{linha 3} & \beta_0 & \end{array} \quad (1.49)$$

$$\beta_0 = 1 - \frac{(-z_d)}{1} \times (-z_d) \rightarrow \beta_0 = 1 - z_d^2. \quad (1.50)$$

De acordo com o critério de estabilidade de Jury, o sistema em (1.48) será estável se $\beta_0 > 0$, que fornece a relação elementar de que pólos estáveis no plano- z estão contidos dentro do círculo unitário.

1.6 TEOREMA DO VALOR FINAL E TEOREMA DO VALOR INICIAL

- Teorema do valor inicial (TVI): aplicável somente a sistemas estritamente próprios, ou seja, cuja ordem do numerador é menor

que a do denominador,

$$f(0) = \lim_{z \rightarrow \infty} (1 - z^{-1}) F(z). \quad (1.51)$$

- Teorema do valor final (TVF): aplicável somente a sistemas assintoticamente estáveis,

$$f(\infty) = \lim_{k \rightarrow \infty} f(k) = \lim_{z \rightarrow 1} \left(\frac{z-1}{z} \right) F(z) = \lim_{z \rightarrow 1} (1 - z^{-1}) F(z^{-1}). \quad (1.52)$$

1.7 EQUIVALENTES DISCRETOS A PARTIR DE APROXIMAÇÕES

Considere um sistema representado por um modelo de primeira ordem, contínuo e descrito pela função de transferência

$$\frac{Y(s)}{U(s)} = \frac{1}{s+1}. \quad (1.53)$$

Como o sistema em (1.53) tem condições iniciais nulas (ou também porque é assintoticamente estável), a variável s do “domínio de Laplace” pode ser considerada simultaneamente uma variável complexa e um operador matemático, ou seja, $s := d/dt$. Desse modo pode-se reescrever (1.53) como sendo

$$\frac{dy(t)}{dt} + y(t) = u(t). \quad (1.54)$$

Em uma implementação digital de simulação desse sistema em (1.54), dt é uma aproximação de um intervalo entre duas medidas de $y(t)$, ou seja,

$$\frac{dy(t)}{dt} \approx \frac{y(kT_s + 1) - y(kT_s)}{T_s} \quad (1.55)$$

ou considerando simplesmente os instantes de tempo discretos $k \in \mathbb{N}$,

$$\frac{dy(t)}{dt} \approx \frac{y(k+1) - y(k)}{T_s}, \quad (1.56)$$

tal que, quanto menor o período de amostragem, mais precisa torna-se a aproximação em relação ao sistema contínuo em (1.54). Observe a

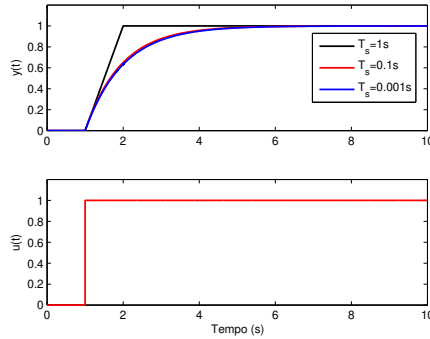


Figura 5 – Efeito do período de amostragem em uma aproximação discreta de um sistema contínuo.

influência de T_s nos gráficos mostrados na Fig. 5 para a simulação do equivalente discreto de (1.54), sendo este descrito por

$$\begin{aligned} \frac{y(k+1) - y(k)}{T_s} + y(k) &= u(k) \\ y(k+1) &= [1 - T_s]y(k) + T_s u(k) \end{aligned} \quad (1.57)$$

A aproximação utilizada neste exemplo [cf. Eq. (1.56)] é conhecida como *Forward Difference* e se baseia no método retangular de Euler (FRANKLIN et al., 1997). O código em MATLAB para reprodução deste exemplo é apresentado na Tabela 5.

1.7.1 Aproximações tipo Forward Difference, Backward Difference e Tustin (Bilinear)

As “aproximações”, como o próprio nome sugere, apenas aproximam sinais e sistemas contínuos a equivalentes discretos aproximados, diferentemente da aplicação da Transformada z sobre uma sequência amostrada, que de fato, fornece uma função polinomial matematicamente exata dessa sequência, desde que a taxa de amostragem selecionada seja adequada para a representação do sinal analógico original.

Mesmo não sendo exatos fisicamente, os equivalentes discretos calculados a partir de aproximações são simples e rápidos de serem obtidos, exigindo tratamento algébrico simples para sistemas de ordem

Tabela 5 – Código de simulação da aproximação discreta com variação do intervalo de amostragem.

```
% Exemplo aproximação backward difference
clear all; close all; clc;

for i=1:3,
switch i,
    case 1
        ts=1; % período de amostragem
        nit=11; % número de iterações
        cor='k'; % cor da linha para o gráfico
    case 2
        ts=0.1; % período de amostragem
        nit=101; % número de iterações
        cor='r'; % cor da linha para o gráfico
    case 3
        ts=0.001; % período de amostragem
        nit=10001; % número de iterações
        cor='b'; % cor da linha para o gráfico
end

u(1:1/ts)=0; u(1/ts+1:nit)=1; % Degrau de entrada

y(1)=0; % Condição inicial da saída do sistema
for k=1:nit,
    y(k+1)=(1-ts)*y(k)+ts*u(k);
end
y=y(1:end-1); % exclui a ultima amostra de y

t=0:ts:nit*ts-ts; % Gera vetor de tempo com base em nit e ts

figure(1);
subplot(211),
    plot(t,y,cor,'linewidth',1.5);
    if i==1, hold; end
subplot(212),
    plot(t,u,'r','linewidth',1.5);
end

subplot(211),
    ylabel('y(t)');
    legend('T_s=1s','T_s=0.1s','T_s=0.001s',0);
    axis([t(1) t(end) 0 1.2]);
subplot(212),
    ylabel('u(t)'); xlabel('Tempo (s)');
    axis([t(1) t(end) 0 1.2]);
```

reduzida.

A aproximação discreta utilizada na seção anterior, mostrada em (1.56), é conhecida como **forward difference**. Reescrevendo-se a Eq. (1.56) em termos de s e, z , como um operador de deslocamento no tempo discreto, é possível verificar que

$$\begin{aligned} \frac{d}{dt}y(kT_s) &\approx \frac{(z-1)}{T_s}y(kT_s), \quad T_s \rightarrow 0, \\ s &:= \frac{1}{T_s}(z-1), \end{aligned} \quad (1.58)$$

que estabelece um mapa aproximado entre o plano- s e o plano- z .

No caso tipo **backward difference**, a Eq. (1.56) modifica-se pela forma em que a aproximação retangular é posicionada sobre o “ponto” do sinal que será usado (cf. Fig. 6), como se o método olhasse para trás, tal que

$$\frac{d}{dt}y(kT_s) \approx \frac{(z-1)}{T_s}y(kT_s - 1), \quad T_s \rightarrow 0,$$

ou mesmo

$$\begin{aligned} \frac{d}{dt}y(kT_s + 1) &\approx \frac{(z-1)}{T_s}y(kT_s), \quad T_s \rightarrow 0, \\ s &:= \frac{1}{T_s} \left(\frac{z-1}{z} \right) \end{aligned} \quad (1.59)$$

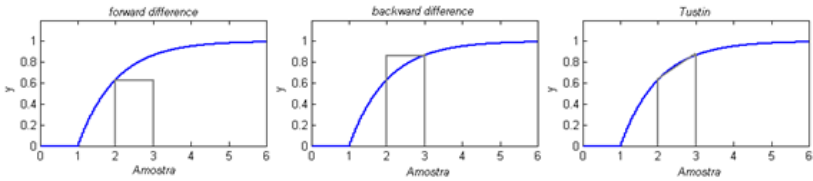


Figura 6 – Análise gráfica de como a aproximação do sinal é feita pelos métodos forward, backward e Tustin.

As aproximações forward e backward são denominadas aproximações retangulares, tal como justificado na apresentação gráfica desses métodos na Fig. 6. Uma alternativa mais precisa a estas aproximações retangulares é a forma trapezoidal, conhecida como aproximação de Tustin ou Transformação Bilinear (FRANKLIN et al., 1997), também

mostrada na Fig. 6 e cujo mapa com o plano- s é obtido pela soma dos métodos forward e backward, tal que

$$\frac{d}{dt}y(kT_s) + \frac{d}{dt}y(kT_s + 1) \approx \frac{2}{T_s} (z - 1) y(kT_s), \quad T_s \rightarrow 0,$$

$$s := \frac{2}{T_s} \frac{(z - 1)}{(z + 1)} \quad (1.60)$$

Na Tabela 6 são mostrados os três métodos discutidos que mapeiam s em z com base em T_s . Note que o **equivalente discreto** de um sistema contínuo é então definido por duas partes: **método de discretização** e o **período de amostragem**. Consequentemente, pode-se dizer que o plano- z pode conter múltiplas representações de um mesmo e único sistema contínuo (ASTROM; WITTENMARK, 2011).

Tabela 6 – Mapeamento aproximado entre o plano- s e plano- z .

Método	Aproximação
Forward	$s := \frac{1}{T_s} (z - 1)$
Backward	$s := \frac{1}{T_s} \left(\frac{z-1}{z} \right)$
Tustin	$s := \frac{2}{T_s} \frac{(z-1)}{(z+1)}$

Esse mapeamento múltiplo do domínio s para o domínio z somente é possível porque o plano- z é “flexível” em termos de sua “densidade numérica”. Ou seja, o período de amostragem é capaz de modificar a capacidade do domínio z de representar mais ou menos modos dinâmicos, tal que **quanto menor se torna o valor de T_s , mais denso se torna o plano- z** . Isso explica como o plano- s , definido em um universo de discurso infinito, pode ser representado dentro de um círculo unitário. Consequentemente, quando $T_s \rightarrow 0$, o universo de discurso do plano- z é dividido por zero e o círculo unitário passa a ter densidade infinita, fazendo com que s e z coincidam.

Na prática, a coincidência entre s e z é hipotética até que se possa considerar precisamente que $T_s \rightarrow dt \rightarrow 0$. No entanto, seria esta a melhor solução? Neste curso, defenderei a alternativa contrária, buscando a utilização da menor quantidade de recursos de processamento possível.

Confira na Tabela 6 que cada método de aproximação tem uma complexidade específica e que determina a precisão da aproximação. Além disto, a escolha do período de amostragem pode influenciar na localização dos pólos mapeados para o plano- z .

Na Fig. 7 é possível verificar as consequências dos mapeamentos para os três métodos de aproximação sob análise. A região hachurada representa as possibilidades de mapeamento de pólos estáveis em s para o domínio z , para os métodos Forward, Backward e Tustin.

Pelo método Forward, raízes no semi-plano esquerdo do plano- s podem ser pólos instáveis (ou zeros de fase não-mínima) em z (FRANKLIN et al., 1997).

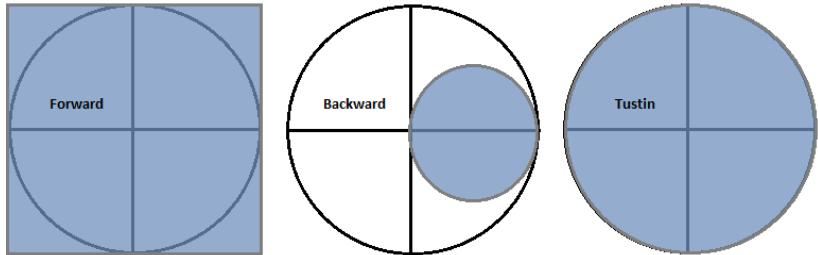


Figura 7 – Consequências da aproximação para os casos Forward, Backward e Tustin.

1.7.2 Efeitos da amostragem em sistemas obtidos por aproximações

De uma maneira geral, os maiores problemas em controle digital, quando métodos de aproximação são utilizados, estão relacionados à sub-amostragem.

O Teorema de Amostragem de Nyquist pode ser considerado como limitante de modo a evitar os problemas de sub-amostragem. No entanto, quando são utilizados métodos de discretização por aproximações, o erro inserido pela aproximação de Euler para a derivada inviabiliza a consideração de Nyquist. Via de regra, para aplicações em controle digital, se espera que o algoritmo de controle possa ter ciclos de 4 a 10 vezes mais rápidos que a dinâmica fundamental que se quer controlar. No caso de sistemas mal modelados, não-lineares, incertos, de dinâmicas muito velozes (e.g., fenômenos eletromagnéticos), pode ser necessário que o algoritmo de controle opere em taxas 20 ou mais vezes velozes que a dinâmica a ser controlada.

A elevação indiscriminada da frequência de amostragem pode também trazer problemas de super-amostragem. Estes, apesar de não possuírem um teorema limite, como no caso mínimo do Teorema da

Amostragem, exigem bom senso do projetista na hora de definir a taxa de amostragem (i.e., $f_s = 1/T_s$). Se f_s for demasiadamente grande, dinâmicas lentas podem passar a se comportar, na prática em sistemas embarcados, como fenômenos estáticos. Os pólos lentos são *espremidos* no lado direito do círculo unitário do plano- z para *abrir* espaço às dinâmicas mais velozes na região esquerda do plano- z . Ou seja, um polo 0,999...9 vai exigir memória suficiente do *hardware* para diferenciar este número de um polo discreto em 1,0.

Um simples teste para mostrar o efeito da super-amostragem pode ser feito rapidamente com o MATLAB. Para um sistema lento descrito por

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

obtenha dois equivalentes discretos pelo método de Tustin, sendo um a 10Hz e outro a 10MHz. Verifique a localização do pólo discreto obtido nos dois casos e você verá que o caso de 10MHz se tornou um “integrador” puro (polo discreto em 1).

O efeito da super-amostragem torna-se preocupante em sistemas multivariáveis ou os que incorporam um grande número de variáveis e dinâmicas distintas (e.g., modelos no espaço de estados). Sistemas mecatrônicos compreendem variáveis mecânicas e elétricas, sendo as elétricas muitas vezes mais velozes que as mecânicas. Na discretização, as variáveis mecânicas podem ser *espremidas* no círculo unitário de forma a abarcar as dinâmicas elétricas no mesmo modelo discreto. Sendo assim, o engenheiro deve ter estes conceitos em mente na hora de decidir qual modelo deve ser discretizado para o projeto de sistemas de controle digitais.

Exercício 1.1 Para os sistemas mostrados a seguir, obtenha as aproximações forward, backward e de Tustin, avaliando a localização dos pólos e zeros em cada caso. Depois, utilizando o Simulink, compare os equivalentes discretos obtidos com os seus respectivos modelos contínuos, a partir da resposta ao degrau.

a) $G(s) = \frac{0,9 \times 2^2}{s^2 + 2 \times 0,3 \times 2s + 2^2}$. Use $T_s = 0.1s$.

b) $G(s) = 0,9 \times \frac{1}{2s+1}$. Use $T_s = 0.1s$.

1.8 MODELAGEM EXPERIMENTAL DE SISTEMAS DINÂMICOS

A modelagem de sistemas dinâmicos é uma parte fundamental para se projetar sistemas de controle, sistemas de estimação e de predição. No entanto, em diversos momentos ao longo da trajetória de aprendizagem sobre a Teoria de Controle, nos acostumamos a resolver exercícios numéricos onde o modelo $G(\cdot) = Y(\cdot)/U(\cdot) = B(\cdot)/A(\cdot)$ é apresentado sem qualquer consideração acerca de como este foi obtido. Simplesmente, o modelo é fornecido para que se realize atividades didáticas, como, por exemplo, de análise do lugar das raízes, resposta em frequência, resposta no tempo, avaliar a sensibilidade às perturbações e aos sistemas de controle projetados.

O contexto didático relativo ao modelo $G(\cdot)$ discutido anteriormente é importante, mas para que a Teoria de Controle possa ser devidamente aplicada em problemas reais, se faz necessário o estudo de métodos de modelagem de sistemas dinâmicos que permitam que processos reais possam ser descritos na forma de modelos matemáticos. Historicamente, os primeiros métodos de modelagem para aplicações em sistemas de controle eram constituídos de modelos fenomenológicos, obtidos a partir de conhecimentos da Física. No entanto, dependendo da área de aplicação, tais modelos podem demandar grande conhecimento da Física e também de grande quantidade de tempo para finalizar a modelagem. Já em aplicações de automação, há a consideração de ambientes industriais com algumas dezenas a milhares de malhas de controle de processos; e se considerada a automação implementada nos eletrodomésticos, em veículos, entre outros produtos automatizados, os números de modelos e o tempo de modelagem necessários extrapolam a capacidade de qualquer especulação. Desse modo, métodos velozes e automatizados de modelagem passam a ser imprescindíveis à Engenharia. Alguns destes métodos de modelagem, em caráter introdutório, são discutidos nesta seção, tais como:

- Modelagem baseada na resposta ao degrau de sistemas de primeira e segunda ordem;
- Modelagem baseada em identificação de sistemas e estimação paramétrica via mínimos quadrados.

1.8.1 Sistemas de primeira ordem com atraso – FOPDT

Uma planta estável com características de resposta de primeira ordem e com atraso (FOPDT, *first-order plus dead time*) (COELHO; COELHO, 2004) é usualmente descrita, na forma contínua e via função de transferência, por

$$G(s) = \frac{Y(s)}{U(s)} = \frac{K_s e^{-\theta s}}{\tau s + 1} \quad (1.61)$$

onde K_s é o ganho estático da planta, isto é,

$$K_s = \frac{y(\infty) - y(0)}{u(\infty) - u(0)}, \quad (1.62)$$

θ é o atraso de transporte (ou tempo morto) contínuo e τ é a constante de tempo que corresponde ao polo $s = -1/\tau$ associado à frequência $1/\tau$ rad/s.

A característica elementar da resposta ao degrau de um processo de primeira ordem é a ausência de modos oscilatórios. Como exemplo, considere os dois modelos mostrados a seguir:

$$G_{p1}(s) = \frac{e^{-s}}{s + 1}, \quad (1.63)$$

$$G_{p2}(s) = \frac{e^{-s}}{(s + 1)(0.1s + 1)}. \quad (1.64)$$

Observe, na Fig. 8, que a resposta ao degrau de ambos os sistemas em (1.63) e (1.64), fornece praticamente a mesma característica de resposta no tempo, apesar de que $G_{p2}(s)$ está sendo representado com um pólo a mais e cuja constante de tempo é 0,1s, mas que pouco influencia na dinâmica do sistema, já que o pólo dominante é o de dinâmica mais lenta. De um ponto de vista prático, $G_{p2}(s)$ poderia ser aproximado por uma representação equivalente a $G_{p1}(s)$.

Na resposta temporal de processos que podem ser modelados como sistemas de primeira ordem, o que se tem é a ausência do sobressinal, uma constante de tempo τ que corresponde ao início da subida da curva de resposta até que esta atinja 63,2% do valor de regime. Além disso, no exemplo da Fig. 8, o degrau unitário foi aplicado no instante $t = 0$ s, sendo que o sistema levou 1,0 segundo para reagir devido ao atraso de transporte.

O desempenho de sistemas tipo FOPDT é comumente relacio-

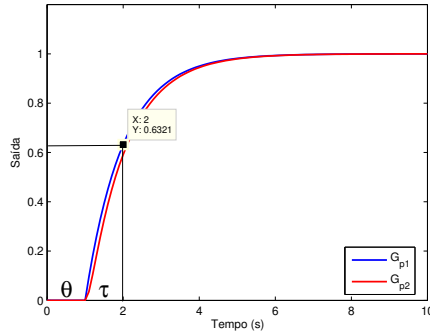


Figura 8 – Resposta ao degrau unitário dos modelos em (1.63) e (1.64).

nado às seguintes características (COELHO; COELHO, 2004):

- Tempo de subida (t_r , *rise time*): intervalo de tempo que leva para a saída ir de 10% a 90% do valor de estabilização. É diferente para sistemas sub-amortecidos, onde considera-se o intervalo de 0% a 100%;
- Tempo de estabilização (t_s , *settling time*): tempo para que a saída entre e permaneça na faixa de $\pm 2\%$ do seu valor em regime permanente.

Pela mera observação e análise da resposta ao degrau torna-se possível obter os parâmetros do modelo FOPDT, experimentalmente.

1.8.2 Sistemas de segunda ordem com atraso – SOPDT

Um modelo contínuo de segunda ordem com atraso (SOPDT, *second-order plus dead time*) pode ser representado pela seguinte forma canônica:

$$G(s) = \frac{K_s \omega_n^2 e^{-\theta s}}{s^2 + 2\zeta \omega_n s + \omega_n^2}, \quad (1.65)$$

onde ω_n é a frequência natural do sistema (em rad/s) e ζ é o fator de amortecimento, tal que

- $\zeta < 1$, sistema sub-amortecido;
- $\zeta = 1$, sistema criticamente amortecido;
- $\zeta > 1$, sistema sobre-amortecido.

Em termos da modelagem experimental de sistemas de segunda ordem via análise da resposta ao degrau, se considera que a característica predominante da saída do sistema é a transição sub-amortecida. Na Fig. 9 é mostrada a resposta ao degrau unitário para a classe de sistemas sub-amortecidos baseados em (1.65), onde são destacadas algumas características importantes, como (COELHO; COELHO, 2004):

- Tempo de subida, t_r : para sistemas sub-amortecidos é o intervalo de tempo para a resposta ir de 0% a 100% do valor da saída em regime, $y(\infty)$.
- Tempo de pico, t_p : é o tempo até o primeiro pico de sobre-sinal,

$$t_p = \pi / \omega_n \sqrt{1 - \zeta^2}. \quad (1.66)$$

- Tempo de estabilização, t_s : tempo até a resposta entrar na faixa de $\pm 2\%$ do valor da saída em regime $y(\infty)$, onde uma aproximação razoável é obtida por

$$t_s = 4 / \zeta \omega_n. \quad (1.67)$$

- Resposta ao degrau em regime permanente (*steady-state*), $y(\infty)$:

$$y(\infty) = \lim_{s \rightarrow 0} sY(s) = \lim_{z \rightarrow 1} (1 - z^{-1}) Y(z^{-1}). \quad (1.68)$$

- Percentual de *overshoot*, PO :

$$PO = \frac{y(t_p) - y(\infty)}{y(\infty)} = e^{-\left(\pi\zeta / \sqrt{1-\zeta^2}\right)}. \quad (1.69)$$

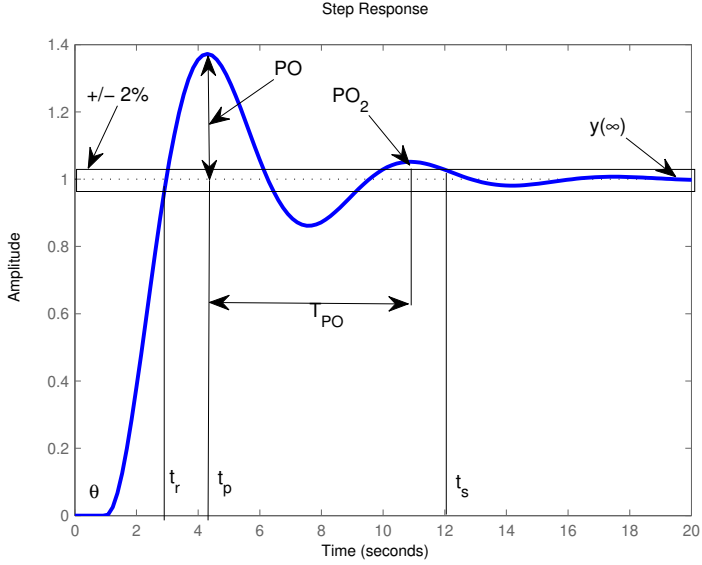


Figura 9 – Resposta ao degrau de um sistema sub-amortecido.

Utilizando a curva de reação ao degrau mostrada na Fig. 9 pode-se obter o ganho estático K_s tal como em (1.62), o fator de amortecimento ζ e a frequência natural ω_n , de sistemas sub-amortecidos ($\zeta < 1$), a partir de

$$\zeta = \left\{ \sqrt{1 + \left[\frac{2\pi}{\ln\left(\frac{PO_2}{PO}\right)} \right]^2} \right\}^{-1}, \quad (1.70)$$

$$\omega_n = \frac{2\pi}{T_{PO}\sqrt{1 - \zeta^2}}. \quad (1.71)$$

Para a modelagem experimental de sistemas FOPDT e SOPDT pode-se implementar algoritmos de processamento de sinais que executem a varredura dos dados de saída em busca das características que permitam os cálculos e a obtenção dos parâmetros do modelo, substituindo um analista humano para tal tarefa, tal como se fez no trabalho de Lemes et al. (2010).

1.8.3 Modelagem via Estimador dos Mínimos Quadrados

A modelagem experimental de sistemas dinâmicos via Estimador dos Mínimos Quadrados (*Least-Squares Estimator*), apresentada neste capítulo, engloba apenas noções básicas e introdutórias de uma área bem mais ampla, denominada Identificação de Sistemas (COELHO; COELHO, 2004).

O Estimador dos Mínimos Quadrados, doravante chamado de estimador MQ, para aplicação em modelagem experimental, opera como um estimador paramétrico que, com base nos sinais de entrada e saída amostrados de um processo, permite obter os parâmetros de uma função de transferência que tenha sido proposta para explicar tal relação de entrada e saída. Isto é, a função $G(z) = B(z)z^{-1}/A(z)$ do sistema não é conhecida em relação a sua ordem, nem mesmo em relação aos seus parâmetros.

O processo a ser modelado é assumido como uma caixa-preta, em que se desconhece a sua constituição interna e que apenas os dados de entrada, $u(k)$, e de saída, $y(k)$, estão disponíveis, tal que o modelo estimado, assumindo uma estrutura linear e SISO, seja descrito por:

$$\hat{G}(z^{-1}) = \frac{\hat{B}(z^{-1})z^{-1}}{\hat{A}(z^{-1})} = \frac{(\hat{b}_0z^0 + \hat{b}_1z^{-1} \dots + \hat{b}_{n_b}z^{-n_b})z^{-1}}{1 + \hat{a}_1z^{-1} + \dots + \hat{a}_{n_a}z^{-n_a}} \quad (1.72)$$

O acento circunflexo sobre os parâmetros nesta função de transferência destacam que o modelo e seus parâmetros são estimados. Desse modo, este modelo estimado produz uma saída estimada,

$$\hat{Y}(z^{-1}) = \hat{G}(z^{-1})U(z^{-1}), \quad (1.73)$$

quando um sinal de entrada conhecido é aplicado em $U(z^{-1})$. Sendo assim, a tarefa de modelagem experimental se baseia em selecionar a adequada ordem $n_a > n_b$, para $\hat{G}(z^{-1})$, tal que a saída estimada convirja para a saída real medida do processo: $\hat{Y}(z^{-1}) \rightarrow Y(z^{-1})$, ao longo de todo o experimento.

É improvável que um modelo matemático estimado de um processo, seja ele linear ou não linear, consiga reproduzir com total exatidão a saída de um experimento real. Dessa maneira pode-se dizer que a saída medida do processo é constituída de uma parte estimada mais uma parcela que corresponde a um processo estocástico $1/\hat{A}(z^{-1})$ cuja

entrada corresponde ao erro de modelagem, $W(z^{-1})$:

$$Y(z^{-1}) = \hat{Y}(z^{-1}) + \frac{1}{\hat{A}(z^{-1})}W(z^{-1}). \quad (1.74)$$

Substituindo a saída estimada $\hat{Y}(z^{-1}) = \frac{\hat{B}(z^{-1})z^{-1}}{\hat{A}(z^{-1})}U(z^{-1})$ na equação da saída medida obtém-se a seguinte expressão:

$$Y(z^{-1}) = \frac{\hat{B}(z^{-1})z^{-1}}{\hat{A}(z^{-1})}U(z^{-1}) + \frac{1}{\hat{A}(z^{-1})}W(z^{-1}). \quad (1.75)$$

Aplicando-se a Transformada-z inversa no modelo mostrado em (1.75), obtém-se a equação a diferenças que estabelece uma relação entre a saída medida $y(k)$, a entrada $u(k)$ usada no experimento e o erro de modelagem, $w(k)$, também conhecido como ruído de processo:

$$\begin{aligned} y(k) = & -\hat{a}_1 y(k-1) - \dots - \hat{a}_{n_a} y(k-n_a) \\ & + \hat{b}_0 u(k-1) + \dots + \hat{b}_{n_b} u(k-n_b-1) + w(k) \end{aligned} \quad (1.76)$$

O ruído de processo, $w(k)$, é assumido como sendo um ruído Gaussiano, média nula e variância σ_w^2 , definida como sendo:

$$\sigma_w^2 = \mathbb{E} \left[(\mathbf{w} - \mu_{\mathbf{w}})^2 \right] = \frac{1}{N} \sum_{k=1}^N [w(k) - \mu_{\mathbf{w}}]^2, \quad (1.77)$$

$$\mathbf{w}^T = \begin{bmatrix} w(1) & \dots & w(N) \end{bmatrix}, \quad (1.78)$$

onde $\mathbb{E}[\cdot]$ denota o operador esperança matemática, \mathbf{w} é o vetor de erros, $\mu_{\mathbf{w}}$ é o valor médio de \mathbf{w} e N é o número total de registros realizados no experimento. Em aplicações reais o ruído de processo não é perfeitamente Gaussiano, mas tal consideração costuma garantir uma boa aproximação do ruído $w(k) = y(k) - \hat{y}(k)$ medido experimentalmente (VICARIO, 2014).

O modelo descrito por equação a diferenças, mostrado em (1.76), é conhecido pelo nome AutoRegressivo com entradas eXógenas (ARX), ou também por modelo Controlado AutoRegressivo (CAR). Existem outras estruturas de modelos que são usualmente empregadas em métodos experimentais de modelagem, como ARIX, ARMAX, ARIMAX, mas que ficam fora do escopo desta abordagem introdutória.

O problema de modelagem experimental usando o estimador

MQ, proposto nesta seção, se baseia em selecionar a estrutura ARX via escolha de $n_a > n_b$, tal que o erro de estimação, $w(k) = y(k) - \hat{y}(k)$, seja mínimo. Este problema de otimização foi proposto por Carl Friedrich Gauss em 1795. No entanto, a seguir, para melhor ilustrar como o estimador opera, sem considerar tal problema de otimização neste primeiro momento, é demonstrado o resultado do método de Gauss do ponto de vista de um problema de regressão linear.

Considere a seguinte estrutura ARX de segunda ordem, com $n_a = 2$, $n_b = 1$:

$$y(k) = -\hat{a}_1 y(k-1) - \hat{a}_2 y(k-2) + \hat{b}_0 u(k-1) + \hat{b}_1 u(k-2) + w(k). \quad (1.79)$$

Assumindo que os parâmetros são invariantes no tempo, este modelo pode ser reescrito na seguinte forma vetorial:

$$y(k) = \phi^T(k) \hat{\theta} + w(k), \quad (1.80)$$

$$\phi^T(k) = \begin{bmatrix} -y(k-1) & -y(k-2) & u(k-1) & u(k-2) \end{bmatrix}, \quad (1.81)$$

$$\hat{\theta}^T = \begin{bmatrix} \hat{a}_1 & \hat{a}_2 & \hat{b}_0 & \hat{b}_1 \end{bmatrix}, \quad (1.82)$$

onde $\phi(k)$ é conhecido como o vetor de regressores (pois detém as amostras passadas da saída e da entrada) e $\hat{\theta}$ é definido como o vetor de parâmetros estimados.

Considere, também, que uma sequência $u(k) = \{u(1), \dots, u(N)\}$ foi aplicada ao processo real, gerando uma sequência de registros da saída medida, $y(k) = \{y(1), \dots, y(N)\}$. Esta sequência registrada de entrada e de saída pode ser representada em uma forma matriz-vetor,

como segue:

$$\underbrace{\begin{bmatrix} y(1) \\ y(2) \\ \vdots \\ y(N) \end{bmatrix}}_{\mathbf{Y}} = \underbrace{\begin{bmatrix} -y(1-1) & -y(1-2) & u(1-1) & u(1-2) \\ -y(2-1) & -y(2-2) & u(2-1) & u(2-2) \\ \vdots & \vdots & \vdots & \vdots \\ -y(N-1) & -y(N-2) & u(N-1) & u(N-2) \end{bmatrix}}_{\mathbf{\Phi}} \underbrace{\begin{bmatrix} \hat{a}_1 \\ \hat{a}_2 \\ \hat{b}_0 \\ \hat{b}_1 \end{bmatrix}}_{\hat{\boldsymbol{\theta}}} + \underbrace{\begin{bmatrix} w(1) \\ w(2) \\ \vdots \\ w(N) \end{bmatrix}}_{\mathbf{w}} \quad (1.83)$$

$$\mathbf{Y} = \mathbf{\Phi}\hat{\boldsymbol{\theta}} + \mathbf{w}. \quad (1.84)$$

O vetor \mathbf{Y} é chamado de vetor de medidas e a matriz $\mathbf{\Phi}$, de dimensão $N \times (n_a + n_b + 1)$, é conhecida como matriz de regressores ou matriz de informação.

Para se determinar o vetor de parâmetros estimados aplicam-se as seguintes manipulações algébricas à equação (1.84): a primeira delas é assumir que $\mathbf{w} = 0$ para que

$$\mathbf{Y} = \mathbf{\Phi}\hat{\boldsymbol{\theta}} = \hat{\mathbf{Y}}. \quad (1.85)$$

Após, multiplica-se esta expressão, pela esquerda, por $\mathbf{\Phi}^T$, tal que

$$\mathbf{\Phi}^T \mathbf{Y} = \mathbf{\Phi}^T \mathbf{\Phi} \hat{\boldsymbol{\theta}}. \quad (1.86)$$

Caso $\mathbf{\Phi}^T \mathbf{\Phi}$ seja uma matriz inversível, a solução para o problema de estimação dos Mínimos Quadrados é dada por:

$$\hat{\boldsymbol{\theta}} = (\mathbf{\Phi}^T \mathbf{\Phi})^{-1} \mathbf{\Phi}^T \mathbf{Y}. \quad (1.87)$$

A matriz $\mathbf{P} = (\mathbf{\Phi}^T \mathbf{\Phi})^{-1}$ é conhecida como a matriz de covariância do erro de estimação dos Mínimos Quadrados. Trata-se de uma matriz simétrica, de dimensão $(n_a + n_b + 1) \times (n_a + n_b + 1)$, semidefinida positiva ($\mathbf{P} \geq 0$). Quanto menor for o seu traço, $\text{tr}(\mathbf{P})$, isto é, quanto menor for a magnitude da soma dos elementos da diagonal principal de \mathbf{P} , melhor. Destaca-se, no entanto, que a consideração de uma \mathbf{P} melhor ou pior depende de uma análise relativa entre ao menos dois

modelos estimados e comparados, onde cada um tem uma matriz \mathbf{P} associada à respectiva solução de estimação.

Por definição, uma outra maneira de se representar a matriz de covariância do erro de estimação paramétrica é dada por:

$$\mathbf{P} = \mathbb{E} \left[\left(\theta - \hat{\theta} \right) \left(\theta - \hat{\theta} \right)^{\mathbf{T}} \right]. \quad (1.88)$$

1.8.4 Otimalidade do estimador MQ

O resultado mostrado na equação (1.87),

$$\hat{\theta} = \mathbf{P} \mathbf{\Phi}^{\mathbf{T}} \mathbf{Y},$$

é ótimo, pois equivale ao mesmo resultado obtido pelo problema de minimização da potência do erro de estimação dos Mínimos Quadrados, cuja função custo é definida por:

$$J_{MQ} = \sum_{i=1}^N [w(i)]^2 = \mathbf{w}^{\mathbf{T}} \mathbf{w}. \quad (1.89)$$

Este vetor $\mathbf{w} = \mathbf{Y} - \mathbf{\Phi} \hat{\theta}$ é o vetor coluna contendo os N erros de estimação entre a saída medida $y(k)$ e a saída estimada $\hat{y}(k) = \phi^T(k) \hat{\theta}$. A função custo a ser minimizada pode então ser expandida, conforme

$$J_{MQ} = \left(\mathbf{Y} - \mathbf{\Phi} \hat{\theta} \right)^{\mathbf{T}} \left(\mathbf{Y} - \mathbf{\Phi} \hat{\theta} \right) = \left[\mathbf{Y}^{\mathbf{T}} - \left(\mathbf{\Phi} \hat{\theta} \right)^{\mathbf{T}} \right] \left(\mathbf{Y} - \mathbf{\Phi} \hat{\theta} \right), \quad (1.90)$$

$$J_{MQ} = \left(\mathbf{Y}^{\mathbf{T}} - \hat{\theta}^{\mathbf{T}} \mathbf{\Phi}^{\mathbf{T}} \right) \left(\mathbf{Y} - \mathbf{\Phi} \hat{\theta} \right), \quad (1.91)$$

e reescrita da seguinte forma:

$$J_{MQ} = \mathbf{Y}^{\mathbf{T}} \mathbf{Y} - \mathbf{Y}^{\mathbf{T}} \mathbf{\Phi} \hat{\theta} - \hat{\theta}^{\mathbf{T}} \mathbf{\Phi}^{\mathbf{T}} \mathbf{Y} + \hat{\theta}^{\mathbf{T}} \mathbf{\Phi}^{\mathbf{T}} \mathbf{\Phi} \hat{\theta}. \quad (1.92)$$

A minimização de J_{MQ} em função dos parâmetros estimados $\hat{\theta}$ é obtida por:

$$\frac{\partial J_{MQ}}{\partial \hat{\theta}} = -\mathbf{\Phi}^{\mathbf{T}} \mathbf{Y} - \mathbf{\Phi}^{\mathbf{T}} \mathbf{Y} + 2 \mathbf{\Phi}^{\mathbf{T}} \mathbf{\Phi} \hat{\theta} = 0 \quad (1.93)$$

e cujo resultado ótimo resulta na mesma expressão mostrada em (1.87):

$$\hat{\theta} = (\Phi^T \Phi)^{-1} \Phi^T \mathbf{Y} = \mathbf{P} \Phi^T \mathbf{Y}. \quad (1.94)$$

1.9 PROBLEMAS

Problema 1.1 Para os sistemas mostrados a seguir, verifique se estes são BIBO estáveis e aplique o teorema do valor final (TVF) (quando possível) para determinar o valor da saída em regime permanente, i.e. $y(k \rightarrow \infty)$, para quando uma entrada do tipo degrau unitário é aplicada:

a)

$$G(z) = \frac{0.09056z + 0.08194}{z^2 - 1.724z + 0.7408}$$

b)

$$G(z) = \frac{0.01765z^2 + 0.07475z + 0.0195}{z^3 - 3.372z^2 + 3.583z - 1.221}$$

Problema 1.2 Para os sistemas mostrados a seguir, determine os equivalentes discretos pelo método ZOH usando uma frequência de amostragem $f_s = 10\text{Hz}$:

a)

$$G(s) = \frac{0.9e^{-1s}}{s + 1}$$

b)

$$G(s) = \frac{8e^{-2s}}{s^2 + 2s + 4}$$

c)

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

d)

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

Problema 1.3 Observe o gráfico mostrado na Fig. 10 e obtenha um modelo contínuo por função de transferência que descreva o comporta-

mento de entrada e saída. Após, analise e discuta as condições mínimas em termos da frequência de amostragem considerando o Teorema de Amostragem de Nyquist-Shannon (i.e., $f_s = 2f_c$ para FOPDT ou $f_s = 2f_n$ para SOPDT subamortecido) e as recomendações para aplicações de análise e projeto de sistemas de controle digitais (i.e., f_s 4–10 vezes mais veloz que f_c para FOPDT, ou f_n para SOPDT subamortecido).

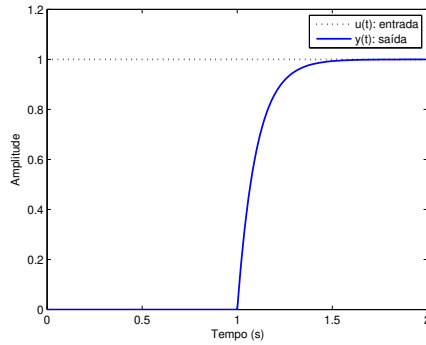


Figura 10 – Resposta de um sistema de primeira ordem com atraso de transporte.

Problema 1.4 Os sistemas $G_1(s)$ e $G_2(s)$ mostrados a seguir apresentam os diagramas de Bode mostrados na Figura 11. De acordo com o Teorema de Amostragem de Nyquist, qual é a frequência de Nyquist (f_s para garantir a mínima condição de amostragem) para $G_1(s)$ e para $G_2(s)$? E quais deveriam ser as mínimas frequências de amostragem para $G_1(s)$ e para $G_2(s)$ se a aplicação fosse a obtenção dos equivalentes discretos via ZOH para análise e projeto de malhas de controle? Obtenha estes equivalentes $G_1(z)$ e $G_2(z)$.

$$G_1(s) = \frac{10^2}{s^2 + 2 \times 0,1 \times 10s + 10^2}$$

$$G_2(s) = \frac{1}{0,1s + 1}$$

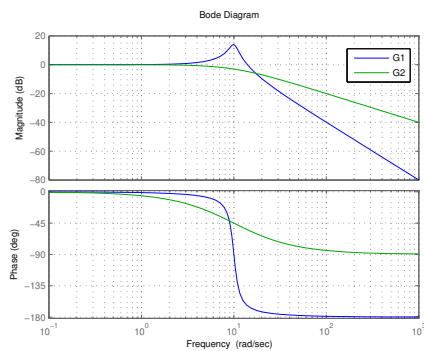


Figura 11 – Diagramas de Bode de um sistema FOPDT e um SOPDT subamortecido, ambos contínuos.

2 CAPÍTULO 2

Controladores digitais baseados em controladores analógicos. Controladores PID digitais. Compensadores “lead-lag”. Conceitos de estabilidade e desempenho robustos.

2.1 INTRODUÇÃO

Nesta etapa do curso se espera que o leitor tenha revisado satisfatoriamente os métodos de discretização apresentados e tenha um conhecimento razoável sobre como realizá-los usando tabelas de transformadas e auxiliado por computador. Também, que seja capaz de modelar experimentalmente processos tipo FOPDT e SOPDT a partir de curvas de reação pela resposta ao degrau e que entenda as características de causalidade inerentes de como um computador acessa o mundo físico, tal como uma malha de controle digital o faz. Isto estabelece a “Transformada ZOH” como o método padrão para a representarmos sistemas físicos a partir da perspectiva de um computador.

Neste capítulo trataremos então de *controladores digitais baseados em controladores analógicos* da família PID e sua relação com compensadores do tipo avanço (*lead*) e atraso (*lag*), introduzindo, neste contexto, conceitos de estabilidade relativa via margens de estabilidade obtidas através da análise de resposta em frequência e por meio de índices de estabilidade chamados de margens de ganho e de fase.

Veremos, também, como os métodos de análise e projeto de sistemas de controle contínuos podem ser utilizados para projetar sistemas de controle discretos com base na *transformada w*, onde considera-se que o equivalente discreto via ZOH de um processo $G(z)$, que incorpora os conversores D-A e A-D, é convertido de volta ao domínio da frequência contínuo (domínio w), por meio de uma aproximação inversa de Tustin.

Na parte de sintonia dos controladores PID, também de sistemas *lead-lag*, consideraremos o estudo por meio da análise de resposta em frequência e também com base no método chamado de Controle por Modelo Interno (do inglês, *Internal Model Control*), IMC. Iniciaremos com uma sintonia IMC voltada ao PID contínuo e finalizaremos com uma versão totalmente digital, baseada em modelos definidos no domínio z .

2.2 CONTROLADORES DIGITAIS BASEADOS EM CONTROLADORES ANALÓGICOS

Vamos restringir nosso estudo nesta seção aos controladores da família PID. No entanto, isto nos dará entendimento suficiente para discretizar por métodos de aproximação outros controladores lineares contínuos existentes.

Começemos com as três topologias mais comuns de controladores PID: PID Ideal, PID Série e PID Paralelo, respectivamente representados pelos diagramas das figuras 12, 13 e 14. Apesar destas se apresentarem, graficamente, de maneira distinta, elas são matematicamente equivalentes. Por exemplo, a relação entre o PID Ideal e o Paralelo nos permite verificar que $K_i = K_p/T_i$ e que $K_d = K_p T_d$.

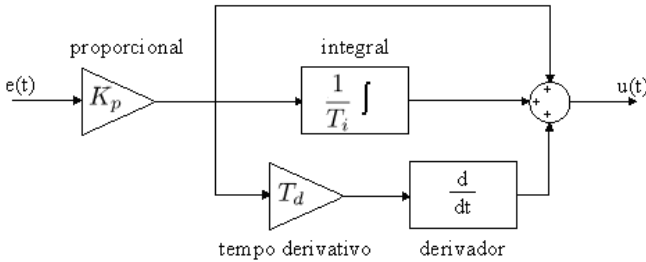


Figura 12 – PID Ideal.

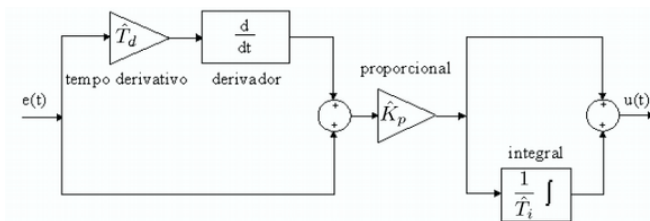


Figura 13 – PID Série.

Conforme apresentado no Capítulo 1, o domínio discreto permite múltiplas representações de um mesmo sistema contínuo, cujo elo entre contínuo e discreto é definido pelo *período de amostragem* e o *método de discretização*. Desse modo é possível afirmar que não há um único

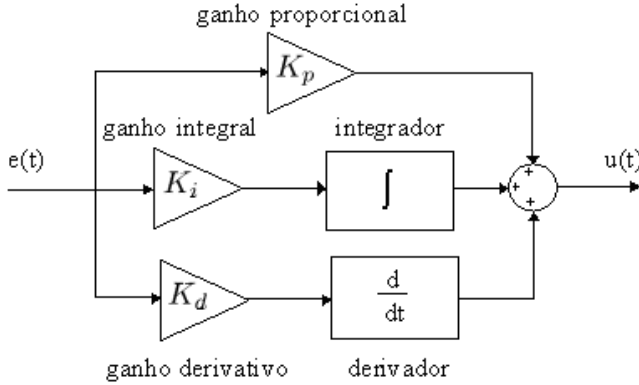


Figura 14 – PID Paralelo.

PID digital baseado no PID contínuo. É possível, inclusive, discretizar a parte derivativa e a parte integral utilizando aproximações distintas (ASTROM; WITTENMARK, 2011).

Consideremos então que $y(t)$, $u(t)$, $r(t)$, $e(t) = r(t) - y(t)$, são, respectivamente, os sinais de saída da planta, sinal de controle, sinal de referência e sinal de erro de seguimento de referência. A lei de controle do PID Paralelo é descrita por

$$u(t) = k_p e(t) + k_i \int e(t) dt + k_d \frac{d}{dt} e(t), \quad (2.1)$$

onde k_p , k_i , k_d , são os ganhos do PID conforme apresentado no diagrama da Figura 14.

De uma maneira intuitiva, podemos considerar que no domínio discreto a ação integral seria simplesmente o acúmulo das amostras do erro ponderadas por um $dt := T_s$ e que a ação derivativa poderia ser a variação do erro dentro de um intervalo $dt := T_s$, tal que

$$u(k) = k_p e(k) + k_i T_s \sum_{n=0}^{\infty} e(k-n) + \frac{k_d}{T_s} [e(k) - e(k-1)]. \quad (2.2)$$

Observe em (2.2) que a aproximação adotada para a parte integral pode gerar sérios problemas em relação ao somatório de infinitos termos $e(k-n)$ em um *hardware* com pouca memória ou não habilitado

para o uso de variáveis do tipo ponto flutuante. Por exemplo, em aplicações embarcadas em microcontroladores com pequena quantidade de memória RAM a lei de controle em (2.2) poderia ser inviável, por exigir um somatório de infinitos termos.

Uma solução mais econômica pode ser considerada: se recordarmos sobre o ZOH, que parte do princípio de que existe um *buffer* que sustenta o comando $u(k)$ até que um novo comando seja calculado no próximo período de amostragem, podemos estabelecer que todo controlador digital pode usufruir do **incremento de controle**, ou seja,

$$u(k) = u(k-1) + \Delta u(k), \quad \Delta = 1 - z^{-1}. \quad (2.3)$$

$u(k)$ é então composto do valor já sustentado no *buffer* do segurador mais o *incremento*, $\Delta u(k)$.

Algoritmos incrementais permitem então a aproximação da ação integral sem que para isto seja necessário o uso de somatório de infinitos termos ou alguma aproximação nessas proporções. O cálculo do incremento de controle pode então ser obtido a partir de

$$\Delta u(k) = u(k) - u(k-1). \quad (2.4)$$

Retirando o primeiro termo do somatório (i.e., $n = 0$) em (2.2), esta equação pode ser reescrita como

$$u(k) = k_p e(k) + k_i T_s e(k) + k_i T_s \sum_{n=1}^{\infty} e(k-n) + \frac{k_d}{T_s} [e(k) - e(k-1)]. \quad (2.5)$$

Partindo da equação original de $u(k)$, em (2.2), pode-se obter $u(k-1)$ deslocando toda a expressão 1-passo para trás no tempo discreto, tal que

$$u(k-1) = k_c e(k-1) + k_i T_s \sum_{n=0}^{\infty} e(k-n-1) + \frac{k_d}{T_s} [e(k-1) - e(k-2)]. \quad (2.6)$$

A diferença entre (2.5) e (2.6), i.e., $\Delta u(k) = u(k) - u(k-1)$, elimina o somatório e compacta essa aproximação do PID Paralelo à

seguinte lei de controle digital:

$$\begin{aligned}\Delta u(k) = & \left[k_p + k_i T_s + \frac{k_d}{T_s} \right] e(k) \\ & + \left[-k_p - 2\frac{k_d}{T_s} \right] e(k-1) + \left[\frac{k_d}{T_s} \right] e(k-2).\end{aligned}\quad (2.7)$$

Observe que se reescrevêssemos a lei de controle do PID Digital mostrado em (2.7) como

$$\Delta u(k) = s_0 e(k) + s_1 e(k-1) + s_2 e(k-2), \quad (2.8)$$

e se aplicássemos a Transformada z em (2.8), o seguinte sistema no domínio da frequência discreta seria obtido:

$$\begin{aligned}\frac{U(z^{-1})}{E(z^{-1})} &= \frac{s_0 + s_1 z^{-1} + s_2 z^{-2}}{1 - z^{-1}}, \\ \frac{U(z)}{E(z)} &= \frac{s_0 z^2 + s_1 z^1 + s_2}{z^2 - z},\end{aligned}\quad (2.9)$$

em que a relação entre a sintonia do PID contínuo paralelo e o PID digital sintetizado seria:

$$\begin{aligned}s_0 &= k_p + k_i T_s + \frac{k_d}{T_s}, \\ s_1 &= -k_p - 2\frac{k_d}{T_s}, \\ s_2 &= \frac{k_d}{T_s}.\end{aligned}\quad (2.10)$$

É importante destacar que essa forma apresentada para se obter uma síntese de controle PID digital, talvez, não seja a mais simples e direta, apesar de considerar aspectos intuitivos de como aproximar as operações de integração e de derivação no tempo, para empregar soluções computacionais em controle PID. No entanto, veremos a seguir que a utilização dos métodos de aproximação *Backward*, *Forward* e Tustin, aplicados diretamente sobre o controlador PID descrito no domínio s ,

$$U(s) = k_p E(s) + k_i \frac{1}{s} E(s) + k_d s E(s), \quad (2.11)$$

pode ser mais simples e direta.

2.2.1 PID digital via aproximação *Backward*

Considere o PID no domínio s mostrado a seguir:

$$\frac{U(s)}{E(s)} = k_p + k_i \frac{1}{s} + k_d s. \quad (2.12)$$

Com base na aproximação *Backward*,

$$s := \frac{1 - z^{-1}}{T_s}, \quad (2.13)$$

sintetiza-se o seguinte PID digital:

$$\frac{U(z)}{E(z)} = k_p + k_i \frac{T_s}{1 - z^{-1}} + k_d \frac{1 - z^{-1}}{T_s}, \quad (2.14)$$

$$\frac{U(z)}{E(z)} = \frac{k_p - k_p z^{-1} + k_i T_s + \left(\frac{k_d}{T_s}\right) (1 - 2z^{-1} + z^{-2})}{1 - z^{-1}}. \quad (2.15)$$

O PID digital via aproximação *Backward*, na forma filtrada, é descrito por:

$$\frac{U(z)}{E(z)} = \frac{\left(k_p + k_i T_s + \frac{k_d}{T_s}\right) + \left(-k_p - 2\frac{k_d}{T_s}\right) z^{-1} + \left(\frac{k_d}{T_s}\right) z^{-2}}{1 - z^{-1}}. \quad (2.16)$$

Multiplicando-se a forma filtrada por z^2 no numerador e no denominador, obtém-se a forma em função de transferência, como segue:

$$\frac{U(z)}{E(z)} = \frac{\left(k_p + k_i T_s + \frac{k_d}{T_s}\right) z^2 + \left(-k_p - 2\frac{k_d}{T_s}\right) z + \left(\frac{k_d}{T_s}\right)}{z^2 - z}. \quad (2.17)$$

Aplicando-se a Transformada z inversa sobre a forma filtrada do PID via aproximação *Backward*, obtém-se a equação a diferenças, no tempo discreto k , que implementa esse controlador:

$$u(k) = u(k-1) + s_0 e(k) + s_1 e(k-1) + s_2 e(k-2), \quad (2.18)$$

$$\begin{aligned}
s_0 &= k_p + k_i T_s + \frac{k_d}{T_s} \\
s_1 &= -k_p - 2\frac{k_d}{T_s}, \\
s_2 &= \frac{k_d}{T_s}.
\end{aligned} \tag{2.19}$$

É importante destacar que esta síntese PID é causal e pode ser aplicada, pois a solução $u(k)$ depende apenas das informações passadas e do estado atual do erro de seguimento de referência $e(k)$.

2.2.2 PID digital via aproximação *Forward*

Considere o PID no domínio s mostrado a seguir:

$$\frac{U(s)}{E(s)} = k_p + k_i \frac{1}{s} + k_d s. \tag{2.20}$$

Com base na aproximação *Forward*,

$$s := \frac{z - 1}{T_s}, \tag{2.21}$$

sintetiza-se o seguinte PID digital:

$$\frac{U(z)}{E(z)} = k_p + k_i \frac{T_s}{z - 1} + k_d \frac{z - 1}{T_s}, \tag{2.22}$$

$$\frac{U(z)}{E(z)} = \frac{k_p z - k_p + k_i T_s + \left(\frac{k_d}{T_s}\right) (z^2 - 2z + 1)}{z - 1}, \tag{2.23}$$

que resulta na seguinte representação por função de transferência:

$$\frac{U(z)}{E(z)} = \frac{\left(\frac{k_d}{T_s}\right) z^2 + \left(k_p - 2\frac{k_d}{T_s}\right) z + \left(-k_p + k_i T_s + \frac{k_d}{T_s}\right)}{z - 1}. \tag{2.24}$$

É possível observar que a síntese PID via aproximação *Forward* é acausal, pois sua função de transferência descreve uma realização sistêmica imprópria, a qual o grau do polinômio do numerador é maior que o do denominador.

Para enfatizar a relação acausal desta síntese PID, reescreve-se a função de transferência após multiplicar numerador e denominador por z^{-1} , que fornece:

$$\frac{U(z)}{E(z)} = \frac{\left(\frac{k_d}{T_s}\right)z + \left(k_p - 2\frac{k_d}{T_s}\right) + \left(-k_p + k_i T_s + \frac{k_d}{T_s}\right)z^{-1}}{1 - z^{-1}}. \quad (2.25)$$

Após, aplica-se a Transformada z inversa para se obter a equação a diferenças dessa síntese PID:

$$u(k) = u(k-1) + s_0 e(k+1) + s_1 e(k) + s_2 e(k-1), \quad (2.26)$$

$$\begin{aligned} s_0 &= \frac{k_d}{T_s}, \\ s_1 &= k_p - 2\frac{k_d}{T_s}, \\ s_2 &= -k_p + k_i T_s + \frac{k_d}{T_s}. \end{aligned} \quad (2.27)$$

Na equação a diferenças do PID via aproximação *Backward* é possível verificar que para se determinar a ação de controle $u(k)$ é necessário conhecer um termo futuro do erro de seguimento de referência, $e(k+1)$.

2.2.3 PID digital via aproximação de Tustin

Considere o PID no domínio s mostrado a seguir:

$$\frac{U(s)}{E(s)} = k_p + k_i \frac{1}{s} + k_d s. \quad (2.28)$$

Com base na aproximação de Tustin,

$$s := \frac{2}{T_s} \left(\frac{z-1}{z+1} \right), \quad (2.29)$$

sintetiza-se o seguinte PID digital:

$$\frac{U(z)}{E(z)} = k_p + k_i \frac{T_s}{2} \left(\frac{z+1}{z-1} \right) + k_d \left[\frac{2}{T_s} \left(\frac{z-1}{z+1} \right) \right], \quad (2.30)$$

$$\frac{U(z)}{E(z)} = \frac{k_p z^2 - k_p + \left(\frac{k_i T_s}{2}\right) (z^2 + 2z + 1) + \left(\frac{2k_d}{T_s}\right) (z^2 - 2z + 1)}{(z^2 - 1)}. \quad (2.31)$$

Organizando este sistema chega-se a seguinte função de transferência do PID digital via aproximação de Tustin:

$$\frac{U(z)}{E(z)} = \frac{\left(k_p + \frac{k_i T_s}{2} + \frac{2k_d}{T_s}\right) z^2 + \left(k_i T_s - \frac{4k_d}{T_s}\right) z + \left(-k_p + \frac{k_i T_s}{2} + \frac{2k_d}{T_s}\right)}{(z^2 - 1)}. \quad (2.32)$$

Multiplicando por z^{-2} o numerador e o denominador deste PID digital, obtém-se a sua forma filtrada, dada por:

$$\frac{U(z)}{E(z)} = \frac{\left(k_p + \frac{k_i T_s}{2} + \frac{2k_d}{T_s}\right) + \left(k_i T_s - \frac{4k_d}{T_s}\right) z^{-1} + \left(-k_p + \frac{k_i T_s}{2} + \frac{2k_d}{T_s}\right) z^{-2}}{(1 - z^{-2})}, \quad (2.33)$$

onde aplicando-se a Transformada z inversa obtém-se a equação a diferenças que implementa essa síntese PID:

$$u(k) = u(k-2) + s_0 e(k) + s_1 e(k-1) + s_2 e(k-2), \quad (2.34)$$

$$\begin{aligned} s_0 &= k_p + \frac{k_i T_s}{2} + \frac{2k_d}{T_s}, \\ s_1 &= k_i T_s - \frac{4k_d}{T_s}, \\ s_2 &= -k_p + \frac{k_i T_s}{2} + \frac{2k_d}{T_s}. \end{aligned} \quad (2.35)$$

Apesar da síntese PID via aproximação de Tustin ser capaz de fornecer uma estrutura de controle causal, a sua parcela derivativa acarreta ruídos na ação de controle, chamados, no jargão da área, de *chattering*. Devido a isto, se torna inviável, do ponto de vista de implementações reais em controle PID, o uso direto dessa síntese, sendo possível destacar este problema no exemplo mostrado a seguir.

Exemplo 2.1 Para o processo

$$G(z) = \frac{Y(z)}{U(z)} = \frac{0,2607407z + 0,1988134}{z^2 - 1,0315524z + 0,449329}, \quad (2.36)$$

verifique, por simulação, o funcionamento do PID digital nas sínteses Backward e de Tustin usando os seguintes parâmetros de sintonia: $T_s = 0,4$ s, $k_p = 0,1$, $k_i = 0,5$, $k_d = 0,05$.

O resultado da implementação das duas sínteses PID digitais é apresentado na Figura 15. É importante observar que a síntese via Backward é diretamente aplicável, enquanto que a via Tustin apresenta o sinal de controle com ruído ou *chattering*. Uma das soluções para este problema é apresentada mais adiante neste capítulo, com base na inclusão de um filtro passa-baixas na parcela derivativa do PID.

O código de simulação em Scilab que permite a reprodução deste exemplo é mostrado na Tabela 7.

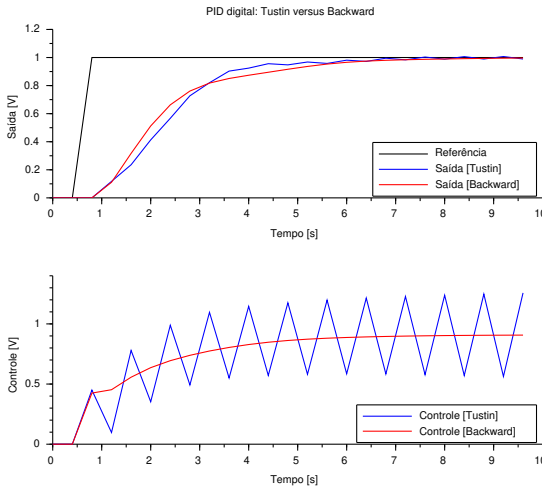


Figura 15 – Controle PID digital: comparando as aproximações via Tustin e Backward, com destaque ao ruído ou *chattering* na implementação via Tustin.

Tabela 7 – Código em Scilab para reproduzir o Exemplo 2.1.

```

// PID digital: aproximação Backward versus Tustin
clear; xdel(winsid()); clc;

// Modelo exemplo
Ks = 1.1; // ganho [V/V]
wn = 2; // freq. natural [rad/s]
zeta = 0.5; // fator de amortecimento [adim.]
s=%s; // s de Laplace
Gs = syslin('c', (Ks*wn^2)/(s^2+2*zeta*wn*s+wn^2) );

// Discretização do sistema via ZOH
Ts = 0.4;
z=%z; // z da Transf. z
Gz = ss2tf( dscr( Gs, Ts ) );
Az = coeff(Gz.den); Az = flipdim(Az,2);
a1 = Az(2); a2 = Az(3);
Bz = coeff(Gz.num); Bz = flipdim(Bz,2);
b0 = Bz(1); b1 = Bz(2);

// Sintonia PID
kp=0.1; ki = 0.5; kd = 0.05;
// Tustin
s0 = kp +ki*Ts/2 +2*kd/Ts;
s1 = ki*Ts -4*kd/Ts;
s2 = -kp +ki*Ts/2 +2*kd/Ts;

// Conferindo o PID via Tustin
Cz1 = syslin(Ts, (s0*z^2+s1*z+s2)/(z^2-1) );
disp('Cz1='); disp(Cz1);

Cs2 = syslin('c', kp+ki/s+kd*s);
Cz2 = horner(Cs2, (2/Ts)*( (z-1)/(z+1) ) );
disp('Cz2='); disp(Cz2);

// Setup da simulação
tfinal = 10; // tempo total em segundos
N = round(tfinal/Ts); // Total de amostras
t=0:Ts:N*Ts-Ts; // vetor de tempo para os plots

// Sinal de referência da malha de controle
r(1:2)=0; r(3:N)=1; // degrau unitário

// Condições iniciais
y(1:2)=0; u(1:2)=0; e(1:2)=0;

// Simulação usando o PID-Tustin
for k=3:N
    // Simula processo controlado
    y(k)=-a1*y(k-1)-a2*y(k-2)+b0*u(k-1)+b1*u(k-2);

    // Controlador PID (obtido via Tustin)
    e(k)=r(k)-y(k);
    u(k)=u(k-2) +s0*e(k)+s1*e(k-1)+s2*e(k-2);
end

// Resultados da simulação
subplot(211); plot2d(t,r,1); plot2d(t,y,2);
legend('Referência','Saída');
ylabel('Saída [V]'); xlabel('Tempo [s]');
subplot(212); plot2d(t,u,2);
ylabel('Controle [V]'); xlabel('Tempo [s]');

// Simulação usando o PID-Backward
// PID Backward
s0b = kp+ki*Ts+kd/Ts;
s1b = -kp-2*kd/Ts;
s2b = -kd/Ts;
for k=3:N
    // Simula processo controlado
    y(k)=-a1*y(k-1)-a2*y(k-2)+b0*u(k-1)+b1*u(k-2);

    // Controlador PID (obtido via Backward)
    e(k)=r(k)-y(k);

```

```

u(k)=u(k-1) +s0b*e(k)+s1b*e(k-1)+s2b*e(k-2);
end

// Resultados da simulação
subplot(211); plot2d(t,y,5);
legend('Referência','Saída [Tustin]','Saída [Backward]',4);
ylabel('Saída [V]'); xlabel('Tempo [s]');
title('PID digital: Tustin versus Backward');
subplot(212); plot2d(t,u,5);
ylabel('Controle [V]'); xlabel('Tempo [s]');
legend('Controle [Tustin]','Controle [Backward]',4);

```

2.2.4 PID com filtro na parcela derivativa: síntese via Tustin

As estruturas ideais de controle PID (Série, Ideal e Paralela) são não causais, já que o grau do polinômio do numerador é maior que o do denominador. Isto é,

$$C(s) = \frac{U(s)}{E(s)} = k_p + k_i \frac{1}{s} + k_d s = \frac{k_d s^2 + k_p s + k_i}{s}. \quad (2.37)$$

Portanto, é natural que a implementação deste sistema de controle, nesta estrutura, gere problemas, tal como na síntese digital via aproximação de Tustin. Também, Astrom e Wittenmark (2011) destacam que uma ação de controle puramente derivativa não deve ser implementada, pois gera grande amplificação do ruído de medição realimentado ao controlador.

Um alternativa ao problema de causalidade do PID é uma versão com a parcela derivativa filtrada usando-se um filtro passa-baixas, acrescentando mais um parâmetro de sintonia ao PID, que é a frequência de corte ω_c [rad/s] (ou $f_c = \frac{\omega_c}{2\pi}$ [Hz]). Esta estrutura alternativa é descrita pelo seguinte modelo de controlador:

$$C(s) = \frac{U(s)}{E(s)} = k_p + k_i \frac{1}{s} + k_d s \left(\frac{\omega_c}{s + \omega_c} \right). \quad (2.38)$$

Com base na aproximação de Tustin,

$$s := \frac{2}{T_s} \left(\frac{z - 1}{z + 1} \right), \quad (2.39)$$

sintetiza-se o seguinte PID digital com filtro passa-baixas na parcela

derivativa:

$$\frac{U(z)}{E(z)} = k_p + \frac{k_i T_s}{2} \left(\frac{z+1}{z-1} \right) + \frac{2k_d}{T_s} \left(\frac{z-1}{z+1} \right) \left\{ \frac{\omega_c}{\left[\frac{2}{T_s} \left(\frac{z-1}{z+1} \right) \right] + \omega_c} \right\}, \quad (2.40)$$

$$\frac{U(z)}{E(z)} = k_p + \frac{k_i T_s}{2} \left(\frac{z+1}{z-1} \right) + k_d \omega_c \left[\frac{(z-1)}{(z-1) + \frac{\omega_c T_s}{2} (z+1)} \right], \quad (2.41)$$

$$\frac{U(z)}{E(z)} = \frac{S(z)}{Q(z)} = \frac{s_0 z^2 + s_1 z + s_2}{q_0 z^2 + q_1 z + q_2}, \quad (2.42)$$

$$\begin{aligned} s_0 &= k_p + \frac{k_p \omega_c T_s}{2} + \frac{k_i T_s}{2} + \frac{k_i \omega_c T_s^2}{4} + k_d \omega_c \\ s_1 &= -2k_p + k_i T_s - 2k_d \omega_c \\ s_2 &= k_p - \frac{k_p \omega_c T_s}{2} + \frac{k_i \omega_c T_s^2}{4} - \frac{k_i T_s}{2} + k_d \omega_c \end{aligned} \quad (2.43)$$

$$\begin{aligned} q_0 &= 1 + \frac{\omega_c T_s}{2} \\ q_1 &= -2 \\ q_2 &= 1 - \frac{\omega_c T_s}{2} \end{aligned} \quad (2.44)$$

Aplicando-se a Transformada z inversa a esta síntese PID obtém-se a seguinte equação a diferenças de controlador:

$$\begin{aligned} u(k) &= \\ &= \frac{1}{q_0} [-q_1 u(k-1) - q_2 u(k-2) + s_0 e(k) + s_1 e(k-1) + s_2 e(k-2)]. \end{aligned} \quad (2.45)$$

Com a introdução de mais um parâmetro ao controlador PID, o ω_c , a complexidade de sintonia aumenta, deixando o operador ou o projetista com a demanda de selecionar T_s , k_p , k_i , k_d e ω_c . Do ponto de vista prático da escolha de ω_c deve-se ter em mente a filtragem de alta frequência, onde o ruído de medição se manifesta, evitando, sempre,

selecionar uma frequência de corte dentro da banda passante da malha de controle.

Astrom e Wittenmark (2011) apresentaram a parcela derivativa, nesta versão filtrada do PID, de uma maneira diferente, dada por

$$sk_d \approx sk_d \frac{1}{\frac{k_d}{N}s + 1} = sk_d \frac{\frac{N}{k_d}}{s + \frac{N}{k_d}}. \quad (2.46)$$

Desse modo, eles propuseram o parâmetro $N \geq 0$, $N \in \mathbb{Z}$, que se popularizou nos equipamentos de automação como um parâmetro adimensional, mas que podemos verificar que se relaciona com a frequência ω_c , como segue:

$$N = \text{int}[\omega_c k_d], \quad (2.47)$$

onde $\text{int}[\cdot]$ designa o operador que converte $\omega_c k_d$ para inteiro.

Em termos práticos, quanto menor for o valor de N menor será a frequência de corte ω_c do filtro da parcela derivativa. E, se $N \rightarrow \infty$, retorna-se ao caso puramente derivativo e que gera o *chattering* e amplifica ruídos de medição.

No Exemplo 2.2 mostra-se uma aplicação desta síntese digital com filtro na parcela derivativa.

Exemplo 2.2 *Neste exemplo repetiremos a implementação do PID digital via aproximação de Tustin realizada no Exemplo 2.1, mas desta vez utilizando o PID com filtro passa-baixas na parcela derivativa mostrado em (2.38) e a sua síntese digital, via aproximação de Tustin, mostrada em (2.42).*

Para fins de comparação com o resultado do Exemplo 2.1, mostrado na Figura 15, os mesmos parâmetros de sintonia foram utilizados: $T_s = 0,4$ s, $k_p = 0,1$, $k_i = 0,5$, $k_d = 0,05$. Neste caso filtrado utilizou-se a frequência de corte do filtro da parcela derivativa como sendo $\omega_c = 5$ rad/s. O resultado da simulação é apresentado na Figura 16 e pode ser reproduzido usando-se o código para Scilab mostrado na Tabela 8.

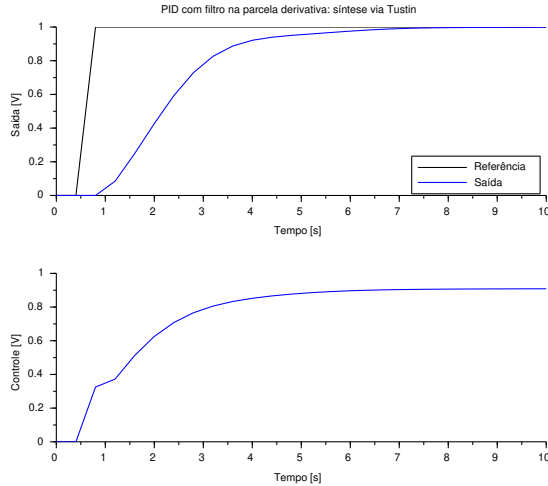


Figura 16 – Controle PID com filtro passa-baixas na parcela derivativa com $w_c = 5$ rad/s: síntese digital via aproximação de Tustin.

Tabela 8 – Código em Scilab para reproduzir o Exemplo 2.2.

```
// PID digital: aproximação de Tustin do PID com filtro na
// parcela derivativa.
clear; xdel(winsid()); clc;

// Modelo exemplo
Ks = 1.1; // ganho [V/V]
wn = 2; // freq. natural [rad/s]
zeta = 0.5; // fator de amortecimento [adim.]
s=%s; // s de Laplace
Gs = syslin('c', (Ks*wn^2)/(s^2+2*zeta*wn*s+wn^2));

// Discretização do sistema via ZOH
Ts = 0.4;
z=%z; // z da Transf. z
Gz = ss2tf(dscr(Gs, Ts));
Az = coeff(Gz.den); Az = flipdim(Az,2);
a1 = Az(2); a2 = Az(3);
Bz = coeff(Gz.num); Bz = flipdim(Bz,2);
b0 = Bz(1); b1 = Bz(2);

// Sintonia PID com filtro PB em D: wc em rad/s
kp = 0.1; ki = 0.5; kd = 0.05; wc = 5;

// Tustin
s0 = kp + kp*wc*Ts/2 + ki*Ts/2 + ki*wc*(Ts^2)/4 + kd*wc;
s1 = -2*kp + ki*Ts - 2*kd*wc;
s2 = kp - kp*wc*Ts/2 + ki*wc*(Ts^2)/4 - ki*Ts/2 + kd*wc;
q0 = 1+wc*Ts/2;
q1 = -2;
q2 = 1 -wc*Ts/2;

// Conferindo o PID via Tustin
Cz1 = syslin(Ts, (s0*z^2+s1*z+s2)/(q0*z^2 +q1*z +q2));
```

```

disp('Cz1='); disp(Cz1);

Cs2 = syslin('c', kp+ki/s* (kd*s)*(wc/(s+wc)) );
Cz2 = horner(Gs2, (2/Ts)*( (z-1)/(z+1) ) );
disp('Cz2='); disp(Cz2);

// Setup da simulação
tfinal = 10*Ts; // tempo total em segundos
N = round(tfinal/Ts); // Total de amostras
t=0:Ts:N*Ts-Ts; // vetor de tempo para os plots

// Sinal de referência da malha de controle
r(1:2)=0; r(3:N)=1; // degrau unitário

// Condições iniciais
y(1:2)=0; u(1:2)=0; e(1:2)=0;

for k=3:N
// Simula processo controlado
y(k)=-a1*y(k-1)-a2*y(k-2)+b0*u(k-1)+b1*u(k-2);

// Controlador PID (obtido via Tustin)
e(k)=r(k)-y(k);
u(k)=(1/q0)*(-q1*u(k-1)-q2*u(k-2) ...
+s0*e(k)+s1*e(k-1)+s2*e(k-2) );
end

// Resultados da simulação
fig1=scf();
subplot(211); plot2d(t,r,1); plot2d(t,y,2);
legend('Referência','Saída',4);
ylabel('Saída [V]'); xlabel('Tempo [s]');
title('PID com filtro na parcela derivativa: síntese via Tustin');
subplot(212); plot2d(t,u,2);
ylabel('Controle [V]'); xlabel('Tempo [s]');

```

2.2.5 PID com filtro na parcela derivativa: síntese *Backward*

Seja o controlador PID Paralelo com filtro passa-baixas na parcela derivativa, de frequência de corte ω_c rad/s, dado por

$$C(s) = k_p + k_i \frac{1}{s} + k_d s \left(\frac{\omega_c}{s + \omega_c} \right). \quad (2.48)$$

Aplicando-se a aproximação *Backward*,

$$s := \frac{1 - z^{-1}}{T_s}, \quad (2.49)$$

obtém-se a realização discreta desse PID via síntese *Backward*, da seguinte maneira:

$$C(z) = k_p + k_i \left(\frac{T_s}{1 - z^{-1}} \right) + k_d \left[\frac{1 - z^{-1}}{T_s} \right] \left[\frac{\omega_c}{\left(\frac{1 - z^{-1}}{T_s} \right) + \omega_c} \right], \quad (2.50)$$

$$C(z) = k_p + k_i \left(\frac{T_s}{1 - z^{-1}} \right) + k_d \left[\frac{1 - z^{-1}}{T_s} \right] \left[\frac{\omega_c T_s}{1 - z^{-1} + \omega_c T_s} \right], \quad (2.51)$$

$$C(z) = \frac{U(z)}{E(z)} = \frac{S(z)}{Q(z)} = \frac{s_0 + s_1 z^{-1} + s_2 z^{-2}}{q_0 + q_1 z^{-1} + q_2 z^{-2}}, \quad (2.52)$$

$$\begin{aligned} s_0 &= k_p + k_p \omega_c T_s + k_i T_s + k_i T_s^2 \omega_c + k_d \omega_c, \\ s_1 &= -2k_p - \omega_c T_s k_p - k_i T_s - 2k_d \omega_c, \\ s_2 &= k_p + k_d \omega_c, \end{aligned} \quad (2.53)$$

$$\begin{aligned} q_0 &= 1 + \omega_c T_s, \\ q_1 &= -2 - \omega_c T_s, \\ q_2 &= 1. \end{aligned} \quad (2.54)$$

Aplicando-se a Transformada z inversa a esta síntese PID obtém-se a seguinte equação a diferenças de controlador:

$$\begin{aligned} u(k) &= \\ &= \frac{1}{q_0} [-q_1 u(k-1) - q_2 u(k-2) + s_0 e(k) + s_1 e(k-1) + s_2 e(k-2)]. \end{aligned} \quad (2.55)$$

2.3 SINTONIA DE CONTROLADORES DIGITAIS BASEADA EM CONTROLADORES ANALÓGICOS

Certamente, existem mais métodos de sintonia de controladores PID registrados na literatura de Controle e Sistemas do que seremos capazes de estudar em nossas vidas.

Métodos de sintonia PID heurísticos e não paramétricos (não baseados em modelo), como os de John G. Ziegler e Nathaniel B. Nichols, foram os primeiros a serem propostos, ainda em 1942, e são estudados e utilizados pela academia e indústria desde então. Outros métodos clássicos, como, por exemplo, a sintonia PID baseada no Lugar Geométrico das Raízes (LGR), também tem sido, há décadas, relatados na literatura (OGATA, 1987).

Desse modo, a sintonia PID, por ser um tema muito amplo, seu estudo requer que decidamos por um subconjunto desta área da Teoria

de Controle, tal que seja possível tratá-la nesta compacta seção. Nesse sentido, nos concentraremos em métodos de sintonia de controladores PID baseados em modelo.

2.3.1 Sintonia PID via Controle por Modelo Interno

Consideremos a sintonia PID via Controle por Modelo Interno (IMC), também chamada de sintonia lambda ou λ -*tuning*, apresentada por Morari e Zafriou (1989). Nela, o PID contínuo $C(s)$, em malha fechada com o processo controlado $G(s)$, comportam-se na transferência da referência de $R(s)$ para a saída $Y(s)$, dinamicamente, segundo um modelo desejado $G_{MF}(z)$, conforme:

$$\frac{Y(s)}{R(s)} = \frac{C(s)G(s)}{1 + C(s)G(s)} = G_{MF}(s), \quad (2.56)$$

onde,

$$G_{MF}(s) = \frac{1}{\tau_{MF}s + 1}. \quad (2.57)$$

Este modelo desejado em malha fechada, ou modelo de referência, é definido com base na escolha da constante de tempo de malha fechada τ_{MF} , as vezes representado pelo símbolo λ e por isso o nome λ -*tuning*, cujo valor pode ser associado à saída

$$y(\tau_{MF}) = 0,632y(\infty). \quad (2.58)$$

Isto é, o tempo τ_{MF} segundos é o tempo necessário para que a saída controlada $y(t)$ atinja 63,2% do seu valor de estabilização $y(t) = r(t)$ quando $t \rightarrow \infty$ e $r(t)$ é uma referência do tipo degrau.

A sintonia IMC apresentada por Morari e Zafriou (1989) considera três classes de processos controlados: primeira ordem, segunda ordem e sistema integrador (tipo-1, com 1 só integrador). Tal sintonia é fornecida na Tabela 9, onde os parâmetros do processo controlado e o valor τ_{MF} são utilizados para calcular os respectivos ganhos k_p , k_i , k_d , do PID.

É importante destacar que a sintonia mostrada na Tabela 9 considera processos com atraso de transporte, latência ou tempo morto, de θ segundos (ou minutos, horas, dias). Caso o processo não possua atraso pode-se considerar $\theta = 0$ e calcular os ganhos normalmente.

Uma outra observação acerca deste método de sintonia é que

para impor a dinâmica de malha fechada desejada, $G_{MF}(s)$, ocorrem cancelamentos dinâmicos (cancelamentos matemáticos entre termos do numerador e do denominador) entre o PID $C(s)$ e o processo controlado $G(s)$. Portanto, como fisicamente os cancelamentos dinâmicos não ocorrem com a exatidão que se permite realizar matematicamente, caso o processo seja instável em malha aberta, não será possível aplicar este método de sintonia diretamente.

Tabela 9 – Sintonia PID IMC (MORARI; ZAFIRIOU, 1989).

Modelo do Processo	Sintonia PID
$G(s) = k_s \frac{1}{\tau s + 1} e^{-\theta s}$	$k_p = \frac{2\tau + \theta}{k_s(2\tau_{MF} + \theta)}$ $k_i = \frac{2k_p}{(2\tau + \theta)}$ $k_d = \frac{k_p \tau \theta}{2(\theta + \tau)}$
$G(s) = k_s \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} e^{-\theta s}$	$k_p = \frac{2\zeta}{k_s \omega_n (\theta + \tau_{MF})}$ $k_i = \frac{k_p \omega_n}{2\zeta}$ $k_d = \frac{k_p}{2\zeta \omega_n}$
$G(s) = k_s \frac{1}{s} e^{-\theta s}$	$k_p = \frac{\theta + 2\tau_{MF}}{k_s (\theta + \tau_{MF})^2}$ $k_i = \frac{k_p}{(\theta + 2\tau_{MF})}$ $k_d = \frac{k_p}{4k_i}$

De uma maneira geral, este e outros métodos de sintonia para PIDs contínuos são extremamente úteis, mas no contexto de controle digital esbarrarão no fato de que os métodos de inteligência computacional empregados na modelagem experimental de sistemas dinâmicos estimam diretamente o modelo $G(z)$ no domínio da frequência discreta $z = e^{j\omega T_s}$, inviabilizando o uso direto de $G(s)$ e exigindo uma etapa para converter $G(z)$ em $G(s)$, para permitir sintonizar o controlador $C(s)$ que após é convertido em $C(z)$ para então, por fim, ser aplicado no controle de $C(z)$.

Para auxiliar o emprego de métodos de sintonia PID contínuos nesse cenário que acabamos de descrever, de modelos obtidos no domínio z , iremos considerar o emprego de um método chamado de Transformada w .

2.3.2 Controladores digitais baseados no domínio w

Nesta seção se fará uma breve discussão sobre um domínio particular de projeto denominado como domínio w , apresentado por Ogata

(1987) em projetos de controladores baseados em análise de resposta em frequência.

A análise e projeto de sistemas no domínio w nos permite utilizar as técnicas do domínio da frequência contínuo, s , para projetar controladores contínuos e posteriormente discretizá-los para o domínio z através da aproximação de Tustin.

A Transformação w ou a transformada ao domínio w se dá pela seguinte relação:

$$G(s) \xrightarrow[\text{ZOH}]{T_s} G(z) \xrightarrow[\text{Tustin}]{T_s} G(w), \quad (2.59)$$

onde a transformação de Tustin indicada em (2.59) se dá na direção discreta para contínua, logo, utilizando a mesma relação da Tabela 6 mostrada no Capítulo 1, mas reescrevendo-se em função do domínio contínuo w , onde de z chega-se em w aplicando-se a seguinte relação:

$$z := \frac{1 + (T_s/2)w}{1 - (T_s/2)w}. \quad (2.60)$$

O sistema $G(w)$ é então uma versão aproximada contínua de um sistema $G(s)$ como se tivesse dispositivos de conversão D-A e A-D conectados aos seus terminais de entrada e saída, respectivamente. Desse modo, quando um controlador $C(w)$, sintonizado no domínio w , é testado sobre $G(w)$, se tem menores desvios quando $C_w(z)$, um equivalente discreto de $C(w)$, é aplicado sobre $G(z)$, já que o modelo $G(w)$ incorpora os conversores D-A e A-D. O mesmo não acontece quando se projeta um controlador $C(s)$ para uma planta $G(s)$.

Consideremos alguns exemplos, mostrados a seguir, da aplicação da Transformada w nas classes de processos de primeira e segunda ordem, por serem usualmente empregadas no projeto de controladores PID.

Exemplo 2.3 *Aplique a Transformada w ao modelo de primeira ordem mostrado a seguir:*

$$G(z) = \frac{b_0 z}{z + a_1}. \quad (2.61)$$

$$z := \frac{1 + (T_s/2)w}{1 - (T_s/2)w}, \quad (2.62)$$

$$G(w) = \frac{b_0 \left[\frac{1+(T_s/2)w}{1-(T_s/2)w} \right]}{\left[\frac{1+(T_s/2)w}{1-(T_s/2)w} \right] + a_1}, \quad (2.63)$$

$$G(w) = \frac{b_0 + b_0 (T_s/2) w}{(1 + a_1) + [(T_s/2) - a_1 (T_s/2)] w}. \quad (2.64)$$

O uso desta equação e também uma forma generalizada de realizar transformações polinomiais via o algoritmo de Horner é apresentado no código para Scilab da Tabela 10.

Tabela 10 – Código em Scilab para auxiliar o entendimento do Exemplo 2.3.

```
// Transformada w de sistema de 1a ordem
clear; xdel(winsid()); clc;

// Exemplo com base em um G(s) <--Ts/ZOH--> G(z)
s = %s;
Gs = syslin('c', 1/(s+1));

Ts = 0.1;
Gz = ss2tf(dscr(Gs,Ts));
Bz=coeff(Gz.num); Bz=flipdim(Bz,2);
b0 = Bz(1);
Az=coeff(Gz.den); Az=flipdim(Az,2);
a1 = Az(2);

// Transformada w
w=poly(0,'w');
Gw = (b0+b0*(Ts/2)*w)/((1+a1) + (1-a1)*(Ts/2)*w );
disp('G(w)='); disp(Gw);

// Transf. w assistida por Scilab via função horner
Gw2 = horner(Gz, (1+(Ts/2)*w)/(1-(Ts/2)*w) );
disp('Solução assistida por Scilab = ');
disp(Gw2); // Comparar com Gw
```

Exemplo 2.4 *Aplique a Transformada w ao modelo de segunda ordem mostrado a seguir:*

$$G(z) = \frac{b_0 z + b_1}{z^2 + a_1 z + a_2}. \quad (2.65)$$

$$z := \frac{1 + (T_s/2) w}{1 - (T_s/2) w}, \quad (2.66)$$

$$G(w) = \frac{b_0 \left[\frac{1+(T_s/2)w}{1-(T_s/2)w} \right] + b_1}{\left[\frac{1+(T_s/2)w}{1-(T_s/2)w} \right]^2 + a_1 \left[\frac{1+(T_s/2)w}{1-(T_s/2)w} \right] + a_2}, \quad (2.67)$$

$$G(w) = \frac{\frac{(b_0+b_1)+[(b_0-b_1)(T_s/2)]w}{1-(T_s/2)w}}{\frac{(1+a_1+a_2)+[T_s(1-a_2)]w+[(T_s/2)^2(1+a_1+a_2)]w^2}{1-2(T_s/2)w+(T_s/2)^2w^2}}, \quad (2.68)$$

$$G(w) = \frac{(b_0 + b_1) - [2b_1 (T_s/2)] w - \left[(b_0 - b_1) (T_s/2)^2 \right] w^2}{(1 + a_1 + a_2) + [T_s (1 - a_2)] w + \left[(T_s/2)^2 (1 - a_1 + a_2) \right] w^2}. \quad (2.69)$$

*O uso desta equação e também uma forma generalizada de re-
alizar transformações polinomiais via o algoritmo de Horner é apresen-
tado no código para Scilab da Tabela 11.*

Tabela 11 – Código em Scilab para auxiliar o entendimento do Exemplo 2.4.

```
// Transformada w de sistema de 2a ordem
clear; xdel(winsid()); clc;

// Exemplo com base em um G(s) <--Ts/ZOH--> G(z)
s = %s; wn = 2; zeta = 0.5;
Gs = syslin('c', wn^2/(s^2+2*zeta*wn*s+wn^2));

Ts = 0.1;
Gz = ss2tf(dscr(Gs,Ts));
Bz=coeff(Gz.num); Bz=flipdim(Bz,2);
b0 = Bz(1); b1 = Bz(2);
Az=coeff(Gz.den); Az=flipdim(Az,2);
a1 = Az(2); a2 = Az(3);

// Transformada w
w=poly(0,'w');
Gw = ( (b0+b1) -2*b1*(Ts/2)*w -(b0-b1)*(Ts/2)^2*w^2 ) / ...
      ( (1+a1+a2) +Ts*(1-a2)*w +(Ts/2)^2*(1-a1+a2)*w^2 );
disp('G(w)='); disp(Gw);

// Transf. w assistida por Scilab via função horner
Gw2 = horner(Gz, (1+(Ts/2)*w)/(1-(Ts/2)*w) );
disp('Solução assistida por Scilab = ');
disp(Gw2); // Comparar com Gw
```

Exemplo 2.5 *Projeto de um controlador PID digital com base em um PID sintonizado no domínio w.*

Considere o seguinte sistema subamortecido:

$$G(s) = \frac{k_s \omega_n^2}{s^2 + 2\zeta \omega_n s + \omega_n^2}, \quad (2.70)$$

onde $k_s = 1$, $w_n = 2 \text{ rad/s}$, $\zeta = 0,1$.

Considerando a frequência de amostragem $\omega_s = 10\omega_n = 2\pi f_s$, aproxima-se o período de amostragem para $T_s = 1/f_s \approx 0,3s$.

Realizando-se a transformação mostrada em (2.59),

$$G(s) \xleftrightarrow[ZOH]{T_s} G(z) \xleftrightarrow[Tustin]{T_s} G(w),$$

$$G(z) = \frac{0,1679z + 0,1613}{z^2 - 1,558z + 0,8869}, \quad (2.71)$$

$$G(w) = \frac{-0,001934w^2 - 0,6242w + 4,247}{w^2 + 0,4377w + 4,247}. \quad (2.72)$$

O processo representado em (2.72) incorpora, de uma forma aproximada, o comportamento de $G(s)$ com ZOH conectado à entrada. Desse modo, um controlador PID contínuo no domínio w é descrito como

$$C(w) = \frac{U(w)}{E(w)} = k_p + \frac{k_i}{w} + k_d w = \frac{k_d w^2 + k_p w + k_i}{w}. \quad (2.73)$$

Observe que na malha direta, ou seja, na transferência do erro entre referência e saída, $E(w)$ para $Y(w)$, a realização $C(w)G(w)$ é **não causal**, pois o polinômio do numerador será de quarta ordem enquanto o do denominador será de terceira. Desse modo, para que seja possível realizar este projeto, será considerada somente a parcela PI, mantendo-se $k_d = 0$. Logo, para que seja possível a realização deste projeto com base em um PID, ao menos um compensador **lag** adicional é requerido para aumentar a ordem do denominador da função de transferência de malha direta.

Nos gráficos das figuras 17 e 18 é possível observar que a representação no domínio w permite uma boa aproximação de $G(z)$ de um ponto de vista macroscópico e pela análise do primeiro estágio do transitório. No entanto, os gráficos da Figura 18 destacam alguns problemas relativos à resposta impulsiva (negativa) incluída pela aproximação de Tustin, além de que $G(w)$ perde sincronismo (ou defasa-se) em relação a $G(s)$ e $G(z)$ a longo prazo.

A sintonia do PI no domínio w foi realizada por tentativa e erro onde se definiu $k_p = 0,2$ e $k_i = 0,08$. A versão discreta do PI foi obtida fazendo-se $C(w) \xrightarrow[Tustin]{T_s} C(z)$, tal que

$$C(z) = \frac{0,212z - 0,188}{z - 1}. \quad (2.74)$$

O resultado prático do método de projeto no domínio w pode ser observado na Figura 19. Nesta, observa-se que as curvas relativas à malha de controle contínua (em w) são muito próximas das curvas obtidas com a simulação discreta. Note, também, que a longo prazo os resultados ficam defasados, tal como observado no caso em malha aberta na Figura 18, o que torna evidente que o método possui limitações devido aos problemas ligados às diferenças em fase. Plotando-se o diagrama de Bode de $G(s), G(w), G(z)$, pode-se verificar, na Figura 20, as diferenças de fase.

Para reproduzir este exercício use o código de simulação em MATLAB disponível na Tabela 12.

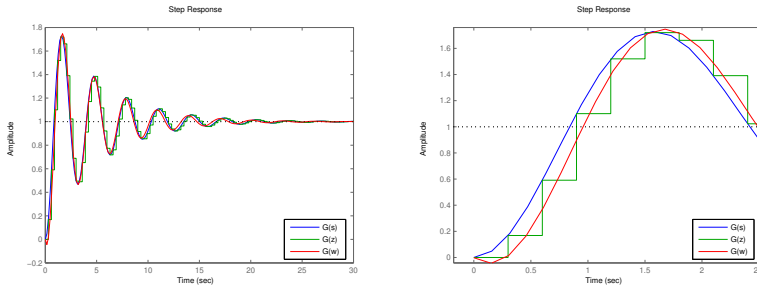


Figura 17 – Comparação entre $G(s), G(z), G(w)$. Boa aproximação no primeiro estágio do transitório.

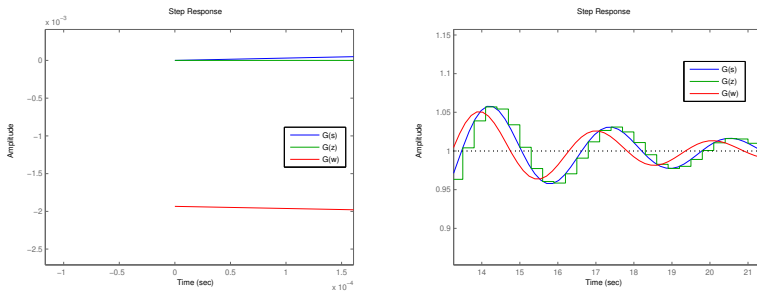


Figura 18 – Comparação entre $G(s), G(z), G(w)$. Destaque a condição não nula decorrente à utilização do método de Tustin e falhas na aproximação a longo prazo em sistemas subamortecidos.

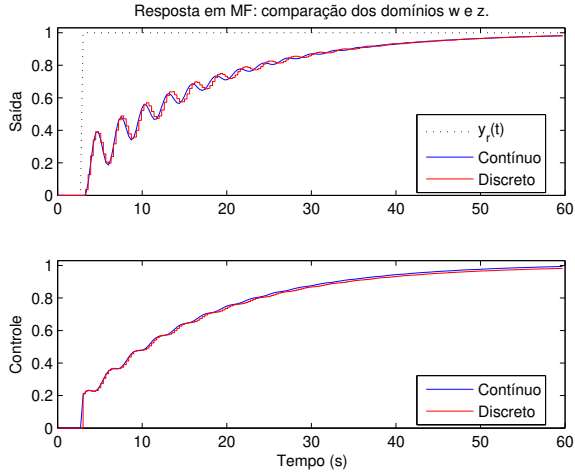


Figura 19 – Resposta ao degrau das malhas de controle contínua e digital, comprovando a boa aproximação do projeto no domínio w para aplicações em controle digital.

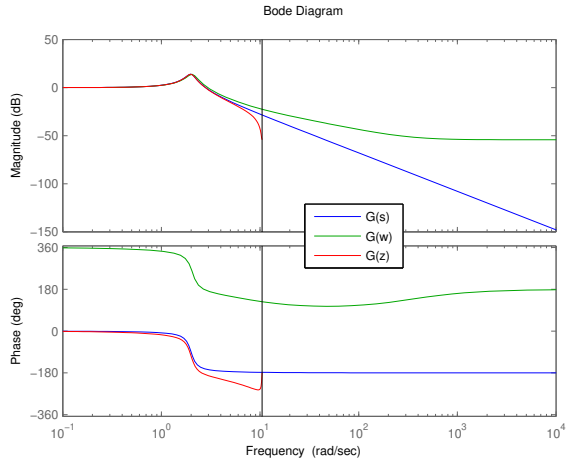


Figura 20 – Diagramas de Bode de $G(s)$, $G(w)$, $G(z)$. Destaque para as diferenças em fase.

Tabela 12 – Código em MATLAB referente ao Exemplo 2.5.

```

% Projeto de um controlador PID digital com base em um PID sintonizado
% no domínio w
clear all; close all; clc;
% Descrição da planta a ser controlada
kp = 1; wn = 2; zeta = 0.1;
Gs = tf(kp*wn^2,[1 2*zeta*wn wn^2])
As = Gs.den1; Bs = Gs.num1; % polinômios do denominador e numerador
% Modelo discreto via ZOH
% Escolha do período de amostragem
ws = 10*wn; % Freq. de amostragem 10 vezes mais veloz que wn
fs = ws/(2*pi);
Ts = 1/fs; % valor aproximadamente 0.3 segundo
Ts = 0.3; % Período de amostragem
Gz = c2d(Gs,Ts,'zoh') % Equivalente discreto via ZOH
Az = Gz.den1; Bz = Gz.num1; % polys denominado e numerador
% Modelo contínuo no domínio w por Tustin
Gw = d2c(Gz,'tustin')
Aw = Gw.den1; Bw = Gw.num1;
% Comparação dos três modelos via resposta ao degrau
figure(1);
step(Gs,Gz,Gw);
legend('G(s)', 'G(z)', 'G(w)', 0);
% Controlador PI no domínio w: C(w)
% Sintonia manual por tentativa e erro
kc = 0.2; % Ganho proporcional
ki = 0.08; % Ganho integral
kd = 0; % Ganho derivativo nulo!!!
%
%          kd*w^2 +kc*w +ki
Cw = tf([kd kc ki],[1 0]); % C(w) = -----
%          w
% Dados para as simulações no tempo contínuo e discreto
yr(1:10,1)=0; yr(11:200,1)=1; % Vetor de Referência yr
nit = length(yr); % número de iterações
t = 0:Ts:nit*Ts-Ts; % vetor de tempo
% Análise de MF no domínio do tempo contínuo
Gmf_w = feedback(Cw*Gw,1,-1); % FT de malha fechada
y=lsim(Gmf_w,yr,t);
e=yr-y; % Erro com base na simulação "fechada" do Matlab
u=lsim(Cw,e,t);
figure(2);
subplot(211),
plot(t,yr,'k',t,y,'b');
ylabel('Saída');
title('Resposta em MF: comparação dos domínios w e z.')
subplot(212),
plot(t,u,'b');
ylabel('Controle'); xlabel('Tempo (s)');
% Análise de MF no domínio de tempo discreto
Cz = c2d(Cw,Ts,'tustin'); % Controlador PID aprox. por Tustin
Gmf_z = feedback(Cz*Gz,1,-1); % FT de malha fechada digital
yz=dlsim(Gmf_z.num1,Gmf_z.den1,yr); % Simulação linear discreta
ez=yr-yz; % Erro com base na simulação discreta
uz=dlsim(Cz.num1,Cz.den1,ez);
figure(2);
subplot(211), hold;
stairs(t,yz,'r'); legend('y_r(t)', 'Contínuo', 'Discreto', 0);
subplot(212), hold;
stairs(t,uz,'r'); legend('Contínuo', 'Discreto', 0);

```

2.3.3 Sintonia PID baseada em modelo discreto

Apesar dos métodos de sintonia do PID contínuo poderem ser aplicados em implementações de PIDs digitais, vimos que há necessidade da aplicação da Transformada w ou de algum outro procedimento de transformação inversa para quando o modelo do processo já estiver definido no domínio discreto z .

Com o aumento sempre constante do uso de métodos de inteligência computacional para a modelagem experimental baseada em dados de sistemas dinâmicos, o mais provável é que os sistemas modelados já se encontrem no domínio do tempo discreto k e passíveis de uma representação direta no domínio da frequência discreta z , no caso de sistemas modelados como funções de transferência.

Neste cenário a sintonia PID descrita por SILVEIRA et al. (2021) pode ser utilizada para impor o comportamento de malha fechada descrito pelo seguinte modelo:

$$G_{MF}(z) = \frac{1 - e^{-\frac{T_s}{\tau_{MF}}}}{z - e^{-\frac{T_s}{\tau_{MF}}}}. \quad (2.75)$$

Este é exatamente igual ao equivalente discreto via Transformada ZOH do modelo desejado de malha fechada utilizado no método IMC ou λ -tuning de Morari e Zafiriou (1989),

$$G_{MF}(s) = \frac{1}{\tau_{MF}s + 1}. \quad (2.76)$$

Desse modo, o único parâmetro que precisa ser especificado para calcular os ganhos do controlador PID é a constante de tempo de malha fechada de τ_{MF} segundos (ou minutos, horas, dias) e o período de amostragem T_s .

Esta sintonia, de k_p , k_i e k_d , para os casos de sistemas discretos de primeira e de segunda ordem, é apresentada na Tabela 13. Ela decorre de um procedimento de cancelamento dinâmico entre o controlador digital $C(z)$ e o modelo do processo $G(z)$ para quando $C(z)$ é um PID Paralelo obtido via aproximação *Backward*, isto é:

$$C(z) = \frac{U(z)}{E(z)} = \frac{s_0 + s_1 z^{-1} + s_2 z^{-2}}{1 - z^{-1}}, \quad (2.77)$$

$$\begin{aligned}
s_0 &= k_p + k_i T_s + \frac{k_d}{T_s}, \\
s_1 &= -k_p - 2\frac{k_d}{T_s}, \\
s_2 &= \frac{k_d}{T_s}.
\end{aligned} \tag{2.78}$$

O modelo do processo controlado é baseado em um sistema de segunda ordem, dado por

$$G(z) = \frac{B(z)}{A(z)} z^{-1} = \frac{(b_0 + b_1 z^{-1})}{(1 + a_1 z^{-1} + a_2 z^{-2})} z^{-1}, \tag{2.79}$$

em que se considera uma aproximação pelo ganho estático ou de frequência nula, isto é, a contribuição de $B(z)$ onde $z = e^{j\omega T_s} = 1$ quando $\omega = 0$, tal que

$$G(z) \approx \frac{B(1)}{A(z)} z^{-1} = \frac{(b_0 + b_1)}{(1 + a_1 z^{-1} + a_2 z^{-2})} z^{-1}. \tag{2.80}$$

Igualando-se o modelo desejado de malha fechada à relação entre controlador e modelo do processo em malha fechada,

$$\frac{C(z)G(z)}{1 + C(z)G(z)} = G_{MF}(z), \tag{2.81}$$

verifica-se que este controlador PID digital que garante essa igualdade deve ser parametrizado como se mostra a seguir:

$$\begin{aligned}
s_0 &= \frac{1 - e^{-\frac{T_s}{\tau_{MF}}}}{b_0 + b_1} \\
s_1 &= s_0 a_1 \\
s_2 &= s_0 a_2
\end{aligned} \tag{2.82}$$

Relacionando as equações (2.78) e (2.82) obtém-se a sintonia PID mostrada na Tabela 13 que impõe o comportamento definido por $G_{MF}(z)$ à malha de controle.

Em estratégias de controle adaptativo, como a utilizada por (SILVEIRA et al., 2021), as relações entre modelo e controlador mostradas em (2.82) podem ser usadas diretamente enquanto um algoritmo de estimação paramétrica estima o modelo do processo.

Como esta técnica de sintonia considera um procedimento de cancelamento dinâmico, então, assim como na sintonia de (MORARI; ZAFIRIOU, 1989), não é possível utilizá-la diretamente em sistemas ins-táveis em malha aberta.

Tabela 13 – Sintonia PID baseada no modelo do processo discreto (SILVEIRA et al., 2021).

Modelo do Processo	Sintonia PID
$G(z) = \frac{b_0 z^{-1}}{1+a_1 z^{-1}}$	$k_p = -\frac{a_1}{b_0} \left(1 - e^{-\frac{T_s}{\tau_{MF}}}\right)$ $k_i = \frac{(1+a_1)}{T_s b_0} \left(1 - e^{-\frac{T_s}{\tau_{MF}}}\right)$ $k_d = 0$
$G(z) = \frac{b_0 z^{-1} + b_1 z^{-2}}{1+a_1 z^{-1} + a_2 z^{-2}}$	$k_p = -\frac{(a_1+2a_2)}{b_0+b_1} \left(1 - e^{-\frac{T_s}{\tau_{MF}}}\right)$ $k_i = \frac{(1+a_1+a_2)}{T_s(b_0+b_1)} \left(1 - e^{-\frac{T_s}{\tau_{MF}}}\right)$ $k_d = \frac{a_2 T_s}{b_0+b_1} \left(1 - e^{-\frac{T_s}{\tau_{MF}}}\right)$

2.4 COMPENSADORES AVANÇO E ATRASO

Compensadores $C(\cdot)$ dos tipos avanço (*lead*) e atraso (*lag*) podem definir estruturas de controle generalizadas para modificar o comportamento dinâmico do processo controlado $G(\cdot)$. O resultado desta modificação dinâmica é avaliado do ponto de vista das propriedades da função de malha direta,

$$G_{MD}(\cdot) = C(\cdot)G(\cdot), \quad (2.83)$$

que promovam desempenho servo ou regulatório e robustez às perturbações de carga, ruído e incertezas de modelagem, para o sistema em malha fechada,

$$G_{MF}(\cdot) = \frac{G_{MD}(\cdot)}{1 + G_{MD}(\cdot)}. \quad (2.84)$$

Observe que $C(\cdot)$ e $G(\cdot)$ caracterizam a possibilidade de se avaliar os compensadores avanço e atraso em ambos os domínios, s e z . No entanto, iniciaremos nosso estudo sobre estes compensadores no domínio s , já que os zeros e os polos são mais simples de serem diretamente interpretados em relação ao domínio da frequência contínua, $s = j\omega$, do que em relação ao domínio da frequência discreto, $z = e^{j\omega T_s}$.

Compensadores avanço e atraso podem ser descritos de uma

forma generalizada pela seguinte expressão (DORF; BISHOP, 1998):

$$C(s) = \frac{K \prod_{i=1}^M (s + z_i)}{\prod_{j=1}^N (s + p_j)}. \quad (2.85)$$

Note que cada z_i e p_j pode ser associado a uma frequência específica, iguais a z_i e p_j radianos por segundo, gerando, assim, um efeito derivativo (avanço) ou integral (atraso). O avanço refere-se ao acréscimo de fase em frequência, de 90° , que um zero fornece. Enquanto o atraso refere-se ao decréscimo de fase em frequência, também de 90° , que um polo fornece.

A seleção dos parâmetros z_i e p_j é, normalmente, realizada com base na análise de resposta em frequência de malha direta, $G_{MD}(s) = C(s)G(s)$. Devido a isto torna-se muito intuitivo de se realizar o projeto de compensadores avanço e atraso com o auxílio do Diagrama de Bode.

Pela análise de resposta em frequência da malha direta pode-se realizar a formatação dos diagramas de magnitude e de fase em frequência (dar forma à malha ou, no jargão da área, *loop shaping*). Isto permite que se projete sistemas de controle sem que se conheça previamente a estrutura de controlador $C(s)$ que resolve o problema. Ao invés disto, o método consiste em:

- avaliar as características de $G(s)$ em malha aberta, definir a estratégia corretiva em termos da
 - banda passante (frequência de corte),
 - dos ganhos nas frequências baixas e médias,
 - da eliminação de picos de ressonância,
 - dos avanços e atrasos de fase em frequências específicas,
- adicionar a $C(s)$ os polos e zeros que agregam à malha direta, $C(s)G(s)$, as propriedades desejadas que atendam os índices de desempenho e/ou robustez desejados (de margem de ganho (MG) e de margem de fase (MF), estudados mais adiante),
- avaliar as características de resposta ao degrau no domínio do tempo (e.g., t_s , t_r , percentual de overshoot) de $G_{MF}(s)$,
- e, avaliar o sinal de controle, que deve ser fisicamente realizável pelo atuador do processo.

Resumidamente, o projeto de compensadores avanço e atraso se baseia em análise do Diagrama de Bode de malha direta, $C(s)G(s)$, e da resposta temporal do sistema completo em malha fechada, $G_{MF}(s)$.

Apesar do método de projeto de malhas de controle avanço e atraso permitir estruturas generalizadas por z_i e p_j , para simplificar, consideraremos apenas o caso de compensadores de primeira ordem capazes de resolver problemas de ordem e complexidade similar. No entanto, é importante destacar que o projeto de compensadores avanço e atraso é didaticamente interessante para preparar o engenheiro para estudar técnicas mais sofisticadas de projeto, como os controladores ótimos baseados em métodos Lineares Quadráticos (LQ), como o LQR, o Filtro de Kalman e o LQG, estudados no Capítulo 5.

2.4.1 Compensador avanço-atraso de primeira ordem

Considere o compensador de primeira ordem mostrado a seguir (DORF; BISHOP, 1998):

$$C(s) = \frac{K(s + z_1)}{(s + p_1)} = \frac{K \frac{z_1}{p_1} (\frac{1}{z_1}s + 1)}{(\frac{1}{p_1}s + 1)}. \quad (2.86)$$

Dependendo dos valores associados a z_1 e p_1 pode-se definir que:

- $|z_1| < |p_1|$: compensador lead (de avanço de fase);
- $|z_1| \ll |p_1|$: compensador lead com ação derivativa (avanço de 90°);

$$C(s) \approx \frac{Ks}{p_1}, C(j\omega) = \frac{K\omega}{p_1} e^{+j90^\circ}; \quad (2.87)$$

- $|z_1| > |p_1|$: compensador lag (de atraso de fase);
- $|z_1| \gg |p_1|$: compensador lag com ação integral (atraso de 90°);

$$C(s) \approx \frac{Kz_1}{s}, C(j\omega) = \frac{Kz_1}{\omega} e^{-j90^\circ}. \quad (2.88)$$

2.4.2 Conceitos de estabilidade e desempenho robustos

As especificações de desempenho no domínio da frequência são essenciais aos projetos de filtros passivos, reguladores e controladores

(filtros ativos) e compensadores dinâmicos (sistemas híbridos compostos por controlador com realimentação por estimador, preditor, fusor, filtro). Tais especificações ditam o **loop-shaping**, ou seja, a forma que normalmente se busca para se ter uma malha de controle com estabilidade e desempenho robustos.

Considerando o Diagrama de Bode e seus critérios de estabilidade como ferramenta de análise, o ponto de partida é normalmente a análise em MA do modelo do processo $G(s)$. A partir deste pode-se definir a estratégia “corretiva” que normalmente reside na vizinhança da curva de magnitude do processo em MA. Destaca-se que modificações extremas na banda passante (i.e., curva de magnitude) de um sistema físico acarretam custos demasiados de energia e possivelmente de componentes. *Metaforicamente, transformar um Fusca em uma Ferrari é uma possibilidade matemática, porém tal realização física pode custar o preço da Ferrari ou mais.*

Supõe-se que o modelo do processo $G(s)$ contém os modos dinâmicos (zeros e pólos) de interesse para o projeto de sistemas de controle, tal que seu Diagrama de Bode apresente em destaque a banda passante em MA e a partir deste ponto pode-se então delimitar as condições de desempenho para a malha direta, $C(s)G(s)$, nas **frequências baixas, nas médias e nas altas**.

O diagrama de magnitude apresentado na Figura 21, extraído de Dorf e Bishop (1998), mas que pode ser encontrado de maneira similar em praticamente todos os livros de controle e sistemas, destaca graficamente as especificações de desempenho. Nessa Figura 21 pode-se verificar que:

- **Nas frequências baixas:** deve-se ter elevados ganhos para garantir que perturbações DC (i.e., perturbações de carga) possam ser rejeitadas e também para garantir o rastreamento de comandos que normalmente são sinais de baixa frequência;
- **Nas frequências médias:** na vizinhança da frequência de corte, i.e., frequências que antecedem a de corte, é de se esperar que se tenha alguma elevação de ganho, preferencialmente que não seja uma elevação do tipo *pico de ressonância*;
- **Nas frequências altas:** a partir da frequência de corte espera-se que se tenha um elevado poder de atenuação, já que nessa região encontram-se os modos dinâmicos que podem ter sido negligenciados na modelagem e também reside o ruído de medição.

De uma maneira geral, ganhos em torno de 20dB e de 10dB para

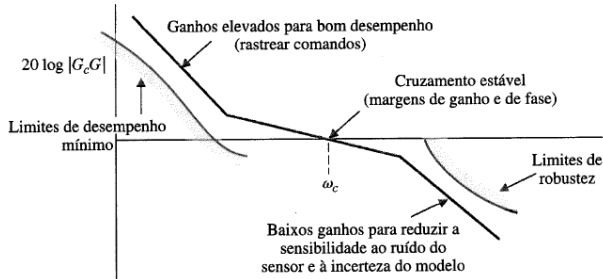


Figura 21 – Especificações de desempenho no domínio da frequência (DORF; BISHOP, 1998).

a região de baixa e média frequência, respectivamente, satisfazem os requisitos de desempenho robusto (STEVENS et al., 2016). No entanto, para algumas estruturas de compensadores e plantas, nem sempre se consegue condicionar adequadamente o ganho de malha $C(s)G(s)$ para atender a essas especificações. Nesses casos pode-se considerar o uso dos índices de robustez conhecidos como **Margem de Ganho** e **Margem de Fase**.

Na Figura 22, extraída de Seborg et al. (2003), pode-se conferir graficamente como as margens de ganho e de fase são obtidas a partir dos diagramas de magnitude e de fase no domínio da frequência. Observe, no entanto, que na Figura 22 a curva de ganhos é apresentada em taxa de amplificação (AR, *amplitude ratio*). Isto é, $MG_{dB} = 20 \log_{10} \left(\frac{1}{AR} \right)$.

A **margem de ganho** está relacionada à frequência crítica, ω_c , tal que, $MG_{AR} = \frac{1}{AR(\omega_c)}$. Esta, por sua vez, é a frequência na qual a curva de fase em frequência intercepta -180° . Quando não ocorre a interceptação diz-se que a MG é infinita. Em teoria, o valor $MG = \frac{1}{AR}$ define até que valor pode-se levar o *ganho de malha* (i.e., da malha direta), tal que o sistema em MF ainda será estável, mesmo que oscilatório (SEBORG et al., 2003).

A **margem de fase** está relacionada à frequência de corte da curva de magnitude. O ângulo de fase na frequência de corte, ϕ_g , na Figura 22, permite calcular a margem de fase como sendo $MF = 180 + \phi_g$. A MF é um índice que quantifica quanto atraso (lag) pode ser incluído na malha de controle até que esta se torne instável (SEBORG et al., 2003).

Quanto maiores forem as margens MG e MF, mais conservador será o sistema de controle projetado e maiores as chances de observar

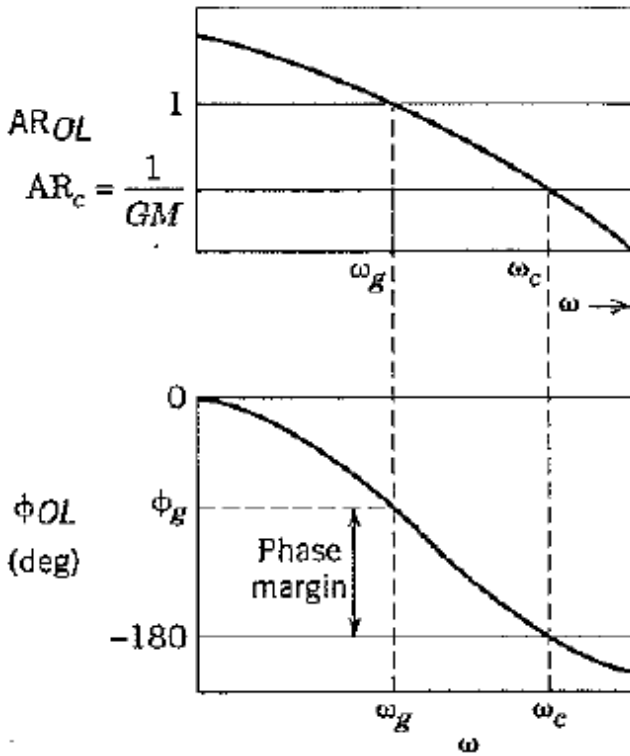


Figura 22 – Margem de ganho e margem de fase (SEBORG et al., 2003).

similaridades entre teoria e prática. Ou seja, **grandes valores de MG e MF estão relacionados a malhas de controle robustas, mas com baixo desempenho, enquanto que MG e MF pequenas estão relacionadas a malhas de controle de alto desempenho, mas não robustas.** De acordo com Seborg et al. (2003), MG em torno de 1,7-4,0 (i.e., 4,6-12,0dB) e MF de 30°-45°, garantem uma boa sintonia da malha de controle com compromisso adequado entre **desempenho e robustez**.

Alguns autores ligados à área de controle robusto, como Stevens et al. (2016), consideram a necessidade de margens ainda maiores, tais como: MG de 6dB a 15dB e MF de 30° a 60°. Margens de fase neste limite superior ainda caracterizam **desempenho robusto e estabilidade robusta** (STEVENS et al., 2016).

Exemplo 2.6 *Projeto de compensador lead-lag via loop-shaping com aná-*

lise de robustez para aplicação em malha de controle digital.

Para uma planta simples descrita por um modelo de primeira ordem, $G(s) = \frac{1}{s+1}$, projete um compensador de primeira ordem e explore os resultados na condição lag, lead e lead-lag. Avalie os diagramas de Bode de $G(s)$, $C(s)G(s)$, e também a reposta no tempo do sistema em malha fechada para cada caso. Observe as margens de ganho e de fase usando a função margin do MATLAB e confira, caso a caso, esses índices de robustez conforme apresentado nesta seção.

O código, em linguagem MATLAB, para a realização deste exemplo é apresentado na Tabela 14.

2.5 PERTURBAÇÕES DETERMINÍSTICAS E ESTOCÁSTICAS

As margens de estabilidade estudadas neste capítulo sugerem um modelo qualitativo de robustez, no que se refere às perturbações em baixa, média e alta frequência. Neste sentido, se faz necessário estabelecer também quais tipos de perturbações podem ser utilizadas para testar a malha de controle projetada.

Além da natureza frequencial da perturbação, é possível classificá-la também pela forma como esta interfere sobre a planta, que pode ser a partir da entrada, da saída, ou mesmo no caso de sistemas realimentados em que perturbações na saída medida são realimentadas à entrada do sistema de controle, que pela consideração deste estar em série com a planta (conforme análise de malha direta, $C(s)G(s)$), afetam a planta a partir da entrada após serem filtradas por um compensador $C(s)$.

Em relação à natureza frequencial da perturbação tem-se os seguintes casos: de carga ou baixa frequência (tipo degrau, por exemplo); de média frequência (sinais com conteúdo harmônico fundamental dentro da banda passante do *ganho de malha direta*); de alta frequência (ruído de medição ou incertezas sobre dinâmicas não modeladas da planta).

2.5.1 Perturbação de carga

Considerando $D_i(z^{-1})$, $D_o(z^{-1})$, como sendo perturbações de carga de entrada e de saída, respectivamente, o seguinte modelo de planta sob influência dessas perturbações pode ser apresentado:

Tabela 14 – Código MATLAB para reproduzir o Exemplo 2.6 sobre compensadores lead e lag.

```
% Exemplo lead-lag
clear all; close all; clc;

% Planta contínua
Gs = tf(1,[1 1]);

% Compensador lead-lag
K = 2; z1 = 2; p1 = 0; % z << p, lead, +90 deg
                        % z >> p, lag, -90 deg

Cs = tf(K*[1 z1],[1 p1]);

% Análise via Diag. de Bode
figure(1);
bode(Gs,Cs*Gs); legend('G(s)', 'C(s)G(s)');

% Margem de Ganho e Margem de Fase
figure(2);
margin(Cs*Gs);

% Análise via resposta ao degrau
yr(1:10,1)=0; yr(11:1000,1)=1; % Sequencia de referência
nit = length(yr); % Número de iterações para simulação linear (lsim)
Ts = 0.01; % Intervalo de simulação para a função lsim
t = 0:Ts:nit*Ts-Ts;

yma = lsim(Gs,yr,t); % Saída em MA
Gmf = feedback(Cs*Gs,1,-1); % Sistema em MF
y = lsim(Gmf,yr,t); % Saída simulada
e = yr-y; % erro de seguimento de referência
u = lsim(Cs,e,t); % Sinal de controle
figure(3);
subplot(211),
    plot(t,yr,':k',t,yma,'r',t,y,'b');
    legend('y_r(t)', 'y_MA(t)', 'y(t)',0); ylabel('Saída');
    title('Resposta em Malha Aberta versus Malha Fechada')
subplot(212),
    plot(t,u,'b'); ylabel('Controle'); xlabel('Tempo (s)');
```

$$Y(z^{-1}) = \frac{B(z^{-1})z^{-1}}{A(z^{-1})}U(z^{-1}) + \frac{1}{A(z^{-1})}D_i(z^{-1}) + D_o(z^{-1}). \quad (2.89)$$

Observe que o modelo apresentado se encontra definido no domínio discreto, mas o mesmo é também válido para sistemas descritos no domínio de Laplace.

2.5.2 Perturbação tipo ruído de medição

Perturbações tipo ruído de medição, $\nu(z^{-1})$, são equacionadas de maneira similar à perturbação $D_o(z^{-1})$, isto é, não são correlacionadas com a entrada, mas se diferenciam em relação ao conteúdo frequencial que é normalmente assumido como sendo *Gaussiano*. Exemplo:

$$Y(z^{-1}) = \frac{B(z^{-1})z^{-1}}{A(z^{-1})}U(z^{-1}) + \nu(z^{-1}). \quad (2.90)$$

Formas de simular um sinal estocástico como o $\nu(z^{-1})$ no Matlab ou Scilab estão prontas, mas você pode obter informações de como tais sinais são produzidos em Coelho e Coelho (2004). No Matlab, a função *wgn* (que se refere a *white Gaussian noise*) pode ser usada conforme o exemplo a seguir:

```
n=1000;m=1;variancia=2;
nu = wgn(n,m,variancia,'linear');
disp(var(nu));
plot(nu)
```

2.5.3 Perturbação estocástica

Uma forma de caracterizar incertezas de modelagem pode ser por meio de perturbações estocásticas.

Considerando esta seção apenas como uma contribuição de conhecimento que está, de certo modo, além do escopo desta disciplina, apenas o modelo ARX (Auto-Regressive with eXogenous inputs) será apresentado, como segue:

$$Y(z^{-1}) = \frac{B(z^{-1})z^{-1}}{A(z^{-1})}U(z^{-1}) + \frac{1}{A(z^{-1})}\xi(z^{-1}). \quad (2.91)$$

No modelo ARX a perturbação $\xi(z^{-1})$ é do tipo Gaussiana e opera como uma perturbação de entrada.

É importante destacar que quando uma malha de controle é fechada ao redor da planta, haverá então algum tipo de perturbação Gaussiana na entrada da planta. Portanto, este modelo pode ser útil para explorar este tipo de cenário, valiando por simulações ou testes analíticos, se um determinado sistema de controle é robusto ao ruído na entrada da planta.

2.6 PROBLEMAS

Problema 2.1 *Para o Diagramas de Bode mostrado na Fig. 23, analise os índices MG e MF apresentados e comente sobre a robustez e desempenho, fazendo também a solução para os casos de margem de ganho, em taxa de amplificação (amplitude ratio, AR).*

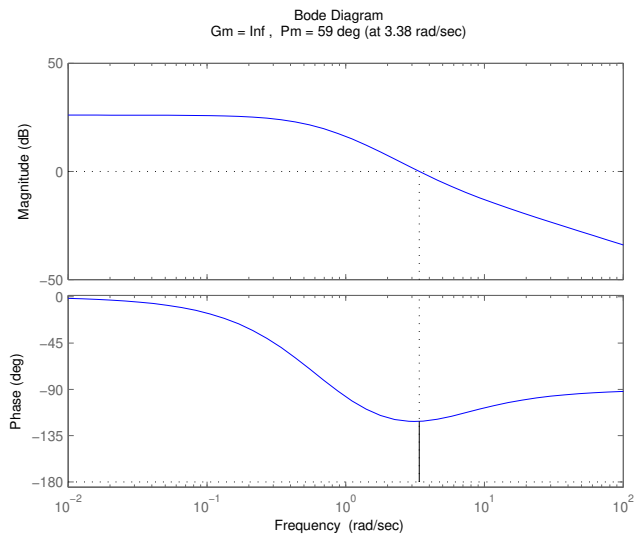


Figura 23 – Diagrama de Bode de malha aberta de um sistema $C(s)G(s)$.

3 CAPÍTULO 3

Sistemas dinâmicos em espaço de estados: representações, controlabilidade, observabilidade, realimentação de estado.

3.1 INTRODUÇÃO

No primeiro contato do estudante com a Teoria de Controle e Sistemas faz-se a consideração da descrição de sistema tipo *caixa-preta*. Isto é, considera-se somente a relação entrada(s) e saída(s), caracterizando a descrição por função (ou matriz de funções) de transferência ou forma polinomial. Neste capítulo iremos então aumentar nosso domínio sobre a representação de sistemas através da representação em *espaço de estados*.

A descrição de sistemas dinâmicos no espaço de estados traz um novo conceito de representação de sistemas, ligando a Teoria de Controle e Sistemas à Álgebra Linear e permitindo que se utilize uma descrição matricial que conecta as informações de entrada e saída de um sistema a um *vetor de variáveis de estado*, $x(t)$, conforme as seguintes equações:

- Caso contínuo:

$$\dot{x}(t) = Ax(t) + Bu(t), \quad (3.1)$$

$$y(t) = Cx(t) + Du(t). \quad (3.2)$$

- Caso discreto:

$$x(kT_s + 1) = \Phi x(kT_s) + \Gamma u(kT_s), \quad (3.3)$$

$$y(kT_s) = Cx(kT_s) + Du(kT_s), \quad (3.4)$$

onde

$$\Phi = e^{AT_s} = \sum_{i=0}^{\infty} \frac{1}{i!} (AT_s)^i, \quad (3.5)$$

$$\Gamma = \left(\int_0^{T_s} e^{Ai} di \right) B, \quad (3.6)$$

ou, quando a matriz A for não singular, i.e., inversível,

$$\Gamma = (\Phi - I) A^{-1} B. \quad (3.7)$$

Uma característica importante dessas descrições, independentemente se no domínio contínuo ou discreto, é que retornamos ao domínio do tempo, tendo simplesmente uma derivação de primeira ordem, indicada por $\dot{x}(t) = sX(s)$, ou um avanço no tempo discreto correspondente a $x(kT_s + 1) = zX(z)$. Isto significa que, não importando a ordem do sistema, tudo precisa ser conectado a partir de relações de primeira ordem. Ou seja, cada variável de estado, contida no vetor $x(\cdot)$, descreve alguma característica dinâmica do sistema, tal que um sistema de ordem n terá n variáveis de estado (ou mais).

Outra característica importante da descrição em espaço de estados é que, seja no domínio contínuo ou no discreto, **a base do sistema**, $x(\cdot)$, se mantém no caso de transformação entre os domínios contínuo e discreto, também as variáveis de entrada e saída, $u(\cdot)$ e $y(\cdot)$, respectivamente. Sendo assim, para simplificar a notação neste capítulo, iremos considerar simplesmente uma forma compacta da descrição do modelo de estados, alternando entre os casos contínuo e discreto de acordo com a necessidade.

Fixemos então nosso estudo sobre SLIT SISO na seguinte forma:

$$\dot{x} = Ax + Bu, \quad (3.8)$$

$$y = Cx + Du. \quad (3.9)$$

As equações (3.8) e (3.9) são denominadas, respectivamente, de **equação de estado** e **equação de saída**, enquanto que as matrizes A, B, C, D , são: a **matriz do sistema** que correlaciona as variáveis de estado (e que define a matriz de transição de estados e^{At}); a **matriz de entrada**, a **matriz de saída** e a **matriz de alimentação direta**, respectivamente. Para se ter uma noção *espacial* do modelo por variáveis de estados, confira o diagrama de blocos mostrado na Fig. 24.

Para melhor compreendermos como um modelo em espaço de estados pode ser definido em termos de sua base, para que esta tenha alguma utilidade matemática, algorítmica ou física, consideremos o exemplo mostrado a seguir.

Exemplo 3.1 *Realização em espaço de estados de um sistema canônico*

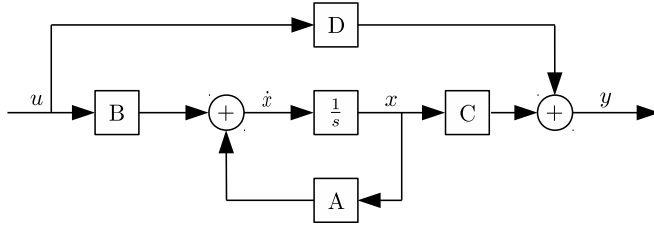


Figura 24 – Diagrama de blocos de modelo por variáveis de estados.

de segunda ordem:

$$G(s) = \frac{Y(s)}{U(s)} = \frac{K_p \omega_n^2}{s^2 + 2\zeta \omega_n s + \omega_n^2} \quad (3.10)$$

Como este sistema é de segunda ordem, i.e., $n = 2$, então o seu modelo de estados necessitará de uma base com pelo menos duas variáveis de estado, doravante designadas por x_1 e x_2 , tal que estas compõem o vetor de estados,

$$x = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^T. \quad (3.11)$$

Os estados x_1 e x_2 podem, por exemplo, descrever a seguinte relação com a variável de saída:

$$x = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^T = \begin{bmatrix} y & \dot{y} \end{bmatrix}^T \quad (3.12)$$

Esta relação pode ser interpretada da seguinte maneira: a saída de um sistema estabelece uma relação de posição de alguma variável física em relação à alguma escala, enquanto que quando esta posição varia ao longo do tempo, sua derivada dá uma noção de velocidade desta variação. Sendo assim, pode-se dizer que,

- $x_1 = y$ representa a posição de $y(t)$,
- $x_2 = \dot{y}$ representa a velocidade de $y(t)$, i.e., $\frac{d}{dt}y(t)$.

Se expandirmos a função de transferência do sistema, reescrevendo-a se utilizando da relação do operador de Laplace, $s := \frac{d}{dt}$, obtém-se

$$\ddot{y} + (2\zeta \omega_n) \dot{y} + (\omega_n^2) y = (K_p \omega_n^2) u. \quad (3.13)$$

Substituindo-se as variáveis de estado nesta expressão, esta se torna

igual a

$$\dot{x}_2 = -(2\zeta\omega_n)x_2 - (\omega_n^2)x_1 + (K_p\omega_n^2)u. \quad (3.14)$$

Esta última equação já nos permite escrever a equação de estados do sistema, bastando para isto, completar a descrição de \dot{x}_1 , tal que o seguinte sistema seja descrito por

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -(2\zeta\omega_n)x_2 - (\omega_n^2)x_1 + (K_p\omega_n^2)u \end{cases} \quad (3.15)$$

o que nos permite reescrever este sistema na forma matricial como sendo

$$\underbrace{\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix}}_{\dot{x}} = \underbrace{\begin{bmatrix} 0 & 1 \\ -(\omega_n^2) & -(2\zeta\omega_n) \end{bmatrix}}_A \underbrace{\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}}_x + \underbrace{\begin{bmatrix} 0 \\ (K_p\omega_n^2) \end{bmatrix}}_B u \quad (3.16)$$

A equação de estados deve ser capaz de reconstruir a saída. Isto significa que a matriz de saída C deve conectar os estados à saída y do sistema, sendo que isto foi definido na base do vetor de estados, tal que

$$y = x_1, \quad (3.17)$$

então a saída do sistema em espaço de estados é descrita por

$$y = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_C \underbrace{\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}}_x + \underbrace{\begin{bmatrix} 0 \end{bmatrix}}_D u \quad (3.18)$$

É importante notar que a matriz D deste sistema é uma matriz nula. Isto ocorre com a maioria das **realizações em espaço de estados**, cujas bases $x(\cdot)$ são fisicamente realizáveis. Isto quer dizer que não há **alimentação direta**, a partir da entrada u para a saída y , sem que tal alimentação não passe por dentro do sistema, ou seja, pela equação de estados (cf. Figura 24).

O exemplo apresentado nesta seção introdutória nos permite destacar uma contribuição da descrição por variáveis de estado: a possibilidade de trabalhar com variáveis adicionais que não estão disponíveis na descrição por funções de transferência. Ou seja, ao invés de trabalhar somente na compensação de $y(t)$, pode-se abrir espaço para compensa-

ção de x_1, \dots, x_n , variáveis de estado.

É importante destacar que a mudança da representação para uma forma de x_1, \dots, x_n , variáveis de estado, não altera a natureza do sistema em termos do seu número de entradas e saídas. Ou seja, o caso do Exemplo 3.1, SISO, de segunda ordem, gerou duas variáveis de estado, mas ainda é um caso SISO, com uma variável *virtual* adicional, que é a variável x_2 .

Obs: a variável x_2 , não é, então, uma variável *sensorialmente medida*. Essa variável é uma virtualização criada pelo projetista ao utilizar conhecimento sobre as equações da física do processo.

As realizações em espaço de estados, possibilitam, através de observadores de estados, a estimação de variáveis adicionais de um problema, quando estas não estão sendo medidas por sensores. Por exemplo, pode-se utilizar relações entre tensão e corrente elétrica, para, a partir de medidas de tensão, estimar/observar informações sobre a corrente elétrica, potência, etc.

Além do caso de observação de estados, onde o número de variáveis de estado é maior que o número de saídas medidas, há também o caso de **fusão de sensores**. Neste, o número de variáveis medidas pode ser maior ou igual ao número de variáveis de estado a se estimar. Como exemplo, cita-se o fusor de acelerômetros e giroscópios para a estimação de posição angular.

Neste curso iremos discutir somente o caso de **observação de estados** em sistemas SISO, discutido mais adiante neste capítulo como uma necessidade para que as variáveis de estado se tornem disponíveis para a elaboração de sistemas de controle com *realimentação total de estados* (STEVENSON et al., 2016).

Apesar de não se dar grande ênfase a sistemas MIMO neste material, ao menos uma noção estrutural e de modelagem é apresentada adiante no texto.

3.2 SISTEMAS MIMO EM ESPAÇO DE ESTADOS

Considerando um sistema MIMO generalizado, de p entradas e q saídas, seu modelo por matriz de função de transferência é descrito por

$$\begin{bmatrix} Y_1(s) \\ \vdots \\ Y_q(s) \end{bmatrix} = \begin{bmatrix} G_{11}(s) & \cdots & G_{1p}(s) \\ \vdots & \ddots & \vdots \\ G_{q1}(s) & \cdots & G_{qp}(s) \end{bmatrix} \begin{bmatrix} U_1(s) \\ \vdots \\ U_p(s) \end{bmatrix} \quad (3.19)$$

Uma realização generalizada em espaço de estados para este sistema assume a seguinte forma:

$$\begin{bmatrix} \dot{x}_1 \\ \vdots \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} + \begin{bmatrix} b_{11} & \cdots & b_{1p} \\ \vdots & \ddots & \vdots \\ b_{n1} & \cdots & b_{np} \end{bmatrix} \begin{bmatrix} u_1 \\ \vdots \\ u_p \end{bmatrix} \quad (3.20)$$

$$\begin{bmatrix} y_1 \\ \vdots \\ y_q \end{bmatrix} = \begin{bmatrix} c_{11} & \cdots & c_{1n} \\ \vdots & \ddots & \vdots \\ c_{q1} & \cdots & c_{qn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} + \begin{bmatrix} d_{11} & \cdots & d_{1p} \\ \vdots & \ddots & \vdots \\ d_{q1} & \cdots & d_{qp} \end{bmatrix} \begin{bmatrix} u_1 \\ \vdots \\ u_p \end{bmatrix} \quad (3.21)$$

Exemplo 3.2 Realização de estados de um sistema MIMO de ordem $n = 4$, com duas entradas e duas saídas:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \\ b_{41} & b_{42} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (3.22)$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} \\ c_{21} & c_{22} & c_{23} & c_{24} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (3.23)$$

3.3 RELAÇÃO ENTRE ESPAÇO DE ESTADOS E FUNÇÃO DE TRANSFERÊNCIA

$$\dot{x} = Ax + Bu$$

$$sX(s) = AX(s) + BU(s)$$

$$(sI - A)X(s) = BU(s)$$

$$X(s) = (sI - A)^{-1}BU(s) \quad (3.24)$$

$$y = Cx + Du$$

$$Y(s) = CX(s) + DU(s)$$

$$Y(s) = C(sI - A)^{-1}BU(s) + DU(s)$$

$$Y(s) = \left[C(sI - A)^{-1}B + D \right] U(s) \quad (3.25)$$

A Eq. (3.25) estabelece uma forma de conversão de uma realização em espaço de estado para função de transferência. Esta relação, estabelece também que, os **autovalores** da matriz A são equivalentes aos pólos do sistema no domínio s de Laplace:

$$\det(\lambda I - A) = \det(sI - A) = 0 \quad (3.26)$$

Note que I é uma matriz identidade de ordem igual a A .

Obs: devido à equivalência entre autovalores e pólos, um sistema será assintoticamente estável desde que os autovalores da matriz A estejam localizados no semiplano esquerdo do plano- s , para o caso contínuo, ou dentro do círculo unitário no plano- z , para o caso discreto.

3.4 ESCOLHA DAS VARIÁVEIS DE ESTADO

A escolha da base do modelo de estados, isto é, das variáveis de estado contidas no vetor de estados, é arbitrária e afirma-se que *existem infinitas realizações de estados para um mesmo sistema*. No entanto, algumas formas tradicionais, conhecidas como formas canônicas em espaço de estados, podem ser discutidas.

Neste documento iremos considerar alguns exemplos que darão subsídios ao estudante de engenharia para conseguir selecionar um conjunto de variáveis de estado que lhe serão válidas para o projeto de sistemas de controle por meio de estados estimados. Nesse sentido, vamos concentrar nossos esforços praticamente em dois casos: *sistemas sem zeros, com o método das variáveis de fase; sistemas com zeros e o método da variável auxiliar* (MAYA; LEONARDI, 2014).

Exemplo 3.3 *Represente por variáveis de estado de fase o sistema descrito pela seguinte equação (MAYA; LEONARDI, 2014, ex. 14.5, pág.*

242):

$$\frac{d^3}{dt^3}y(t) + 3\frac{d^2}{dt^2}y(t) + 4\frac{d}{dt}y(t) + 2y(t) = 5u(t)$$

$$x_1 = y, \quad x_2 = \dot{y}, \quad x_3 = \ddot{y}$$

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = x_3 \\ \dot{x}_3 = -3x_3 - 4x_2 - 2x_1 + 5u \end{cases}$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -2 & -4 & -3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 5 \end{bmatrix} u$$

$$y = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

Exemplo 3.4 *Represente por variáveis de estado o sistema MIMO descrito pelas seguintes equações (MAYA; LEONARDI, 2014, ex. 14.6, pág. 243):*

$$\ddot{y}_1 + 2\dot{y}_1 - 2y_2 = u_1$$

$$\ddot{y}_2 + y_2 - y_1 = u_2$$

$$x_1 = y_1, \quad x_2 = \dot{y}_1, \quad x_3 = y_2, \quad x_4 = \dot{y}_2$$

$$\dot{x}_2 + 2x_2 - 2x_3 = u_1$$

$$\dot{x}_4 + x_3 - x_1 = u_2$$

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -2x_2 + 2x_3 + u_1 \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = -x_3 + x_1 + u_2 \end{cases}$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -2 & 2 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

Exemplo 3.5 Sistema com zeros e o método da variável auxiliar: *Determine o modelo de estados do sistema cuja função de transferência é (MAYA; LEONARDI, 2014, ex. 14.8, pág. 244):*

$$G(s) = \frac{Y(s)}{U(s)} = \frac{20(s^2 + 2s + 17)}{s^3 + 13s^2 + 53s + 40}$$

Considerando a variável auxiliar $X(s)$, multiplica-se o sistema da seguinte forma:

$$G(s) = \frac{Y(s)}{U(s)} = \frac{20(s^2 + 2s + 17)}{s^3 + 13s^2 + 53s + 40} \left(\frac{X(s)}{X(s)} \right)$$

Após, defini-se que

$$U(s) = (s^3 + 13s^2 + 53s + 40)X(s) \rightarrow u = \ddot{x} + 13\dot{x} + 53x + 40x$$

$$Y(s) = 20(s^2 + 2s + 17)X(s) \rightarrow y = 20\ddot{x} + 40\dot{x} + 340x$$

Com base na representação pela variável auxiliar, pode-se aplicar o método das variáveis de estado de fase, isto é,

$$x_1 = x, \quad x_2 = \dot{x}, \quad x_3 = \ddot{x}$$

logo,

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = x_3 \\ \dot{x}_3 = -13x_3 - 53x_2 - 40x_1 + u \end{cases}$$

$$y = 20x_3 + 40x_2 + 340x_1$$

Uma realização do sistema $G(s)$ é então

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -40 & -53 & -13 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u$$

$$y = \begin{bmatrix} 340 & 40 & 20 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + [0] u$$

Exemplo 3.6 Sistema com zeros e o método da variável auxiliar: *Determine o modelo de estados do sistema cuja função de transferência é (MAYA; LEONARDI, 2014, prob. (f), pág. 246):*

$$\ddot{y} + 2\dot{y} + y + \int_0^t y dt = \dot{u} + 5u$$

$$Y(s) \left(s^2 + 2s + 1 + \frac{1}{s} \right) = U(s) (s + 5)$$

$$\frac{Y(s)}{U(s)} = \frac{(s + 5)}{\left(s^2 + 2s + 1 + \frac{1}{s} \right)} \left(\frac{X(s)}{X(s)} \right)$$

$$U(s) = \left(s^2 + 2s + 1 + \frac{1}{s} \right) X(s) \rightarrow u = \ddot{x} + 2\dot{x} + x + \int x dt$$

$$Y(s) = (s + 5)X(s) \rightarrow y = \dot{x} + 5x$$

$$x_1 = \int x dt, \quad x_2 = x, \quad x_3 = \dot{x},$$

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = x_3 \\ \dot{x}_3 = -2x_3 - x_2 - x_1 + u \end{cases}$$

$$y = x_3 + 5x_2$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & -1 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u$$

$$y = \begin{bmatrix} 0 & 5 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + [0] u$$

3.5 MATRIZ DE TRANSIÇÃO DE ESTADOS

Escrevendo-se a equação de estados, mas considerando uma condição inicial x_0 , é possível desenvolver a seguinte equação:

$$\dot{x} - x_0 = Ax + Bu, \quad (3.27)$$

$$sX(s) - x_0 = AX(s) + BU(s), \quad (3.28)$$

$$X(s) = (sI - A)^{-1}x_0 + (sI - A)^{-1}BU(s). \quad (3.29)$$

A matriz $(sI - A)^{-1}$ é chamada de *matriz de transição de estados* e verifica-se que possui relação direta com a matriz que relaciona os pólos de um sistema aos autovalores da matriz A , isto é:

- Matriz de transição de estados: $\phi(s) = (sI - A)^{-1}$;
- Polinômio característico do sistema: $\Delta(s) = \det(sI - A) = 0$.

A $\phi(s)$, por meio da Transformada Inversa de Laplace, nos permite obter as equações no domínio do tempo de resposta de *estado zero*, x_0 , e resposta forçada a partir de $u(t)$.

Por uma questão de limitação de tempo deste curso, vamos considerar apenas a definição de *matriz de transição de estados*, entendendo que quando modificamos os autovalores de um determinado sistema, consequentemente modificamos a dinâmica de transição de estados e isto é fundamental para que realizemos controle por realimentação de estados, que é um dos principais motivos de estudo deste curso.

3.6 CONTROLE POR REALIMENTAÇÃO DE ESTADO

O estudo de sistemas descritos no espaço de estados, suas realizações, classificações e métodos, se devem, neste curso, principalmente,

para preparar o leitor para uma introdução às *Técnicas de Controle Moderno*, cujo princípio fundamental se dá pelo controle por *realimentação de variáveis de estado*.

Na realimentação de estados o projetista pode optar pela realimentação **total** ou **parcial** de estados, sendo que o caso total será o foco para este curso introdutório, haja vista que as propriedades desse caso são inclusive bem mais atrativas do ponto de vista dos melhores resultados que podem ser obtidos, em termos de desempenho e robustez.

No caso de **realimentação total de estados**, doravante designado simplesmente de realimentação de estados, espera-se que todas as variáveis de estado estejam sendo *medidas* e estejam à disposição do sistema de controle. Quando isto não é verdade, uma realização de estados mais simples (de menor ordem) deve ser obtida, fazendo com que esse número menor de variáveis de estado esteja disponível para a realimentação.

Quando não há como medir todas as variáveis de estado e em hipótese alguma pode-se desprezar tais variáveis, pois o objetivo do sistema de controle exige que todos os estados estejam sob controle, se faz necessária a **estimação** ou **observação de estado**, podendo esta ser feita em M.A. ou M.F.. O uso do **Observador de Estado** vai possibilitar que se faça a **realimentação de estado observado**, quando não for possível a medição de todas as variáveis de estado.

O problema de *Controle por Realimentação de Estado* se apresenta como um problema **dual** ao de *Observação de Estado*, possuindo, respectivamente, os seguintes pré-requisitos de análise para o projeto de ambos: **Controlabilidade** e **Observabilidade**.

3.6.1 Controlabilidade

A *controlabilidade* é uma propriedade da representação em espaço de estados, que de um ponto de vista simplificado, condiciona se existe a possibilidade de controlar os estados $x(t)$, de uma determinada realização em espaço de estados, a partir do vetor de entradas $u(t)$. Entenda então que, *quando um sistema no espaço de estados é dito **controlável**, é porque há como modificar a dinâmica das variáveis de estado do vetor $x(t)$.*

O teste de *controlabilidade* é feito a partir da **Matriz de Controlabilidade**:

$$C_o = \begin{bmatrix} B & AB & \cdots & A^{(n-1)}B \end{bmatrix}. \quad (3.30)$$

Quando o posto da matriz C_o é dito cheio ou completo, isto é, conside-

rando n variáveis de estado, o *rank* ou posto da matriz C_o for igual a n , a realização de estados é dita controlável.

Para sistemas bem condicionados, cuja matriz A é não-singular (i.e., é inversível),

- Se $\det(C_o) \neq 0$, então o sistema é dito **controlável**.

O teste de controlabilidade independe de domínio, em termos de tempo contínuo ou discreto. Suas únicas dependências são as matrizes A e B do sistema. No Matlab a matriz C_o pode ser construída e seu posto (rank) avaliado da seguinte forma:

```
Co = ctrb(A,B)
rank(Co)
```

3.6.2 Observabilidade

A *observabilidade* é uma propriedade da representação em espaço de estados que, também de um ponto de vista simplificado, condiciona se existe a possibilidade de determinar os estados $x(t)$, a partir das observações do vetor de saídas medidas $y(t)$. Entenda então que, *quando um sistema no espaço de estados é dito **observável**, este é como um problema **dual** ao de controlabilidade, mas que evidencia se a saída medida $y(t)$ tem correlação com as mudanças dinâmicas das variáveis de estado do vetor $x(t)$.*

O teste de *observabilidade* é feito a partir da **Matriz de Observabilidade**:

$$O_b = \begin{bmatrix} C & CA & \dots & CA^{(n-1)} \end{bmatrix}^T. \quad (3.31)$$

Quando o posto da matriz O_b é dito cheio ou completo, isto é, considerando n variáveis de estado, o *rank* ou posto da matriz O_b for igual a n , a realização de estados é dita observável.

Para sistemas bem condicionados, cuja matriz A é não-singular (i.e., é inversível),

- Se $\det(O_b) \neq 0$, então o sistema é dito **observável**.

O teste de observabilidade independe de domínio, em termos de tempo contínuo ou discreto. Sua única dependência são as matrizes A e C do sistema. No Matlab a matriz O_b pode ser construída e seu posto (rank) avaliado da seguinte forma:

```
Ob = obsv(A,C)
rank(Ob)
```

3.6.3 Regulação via realimentação de estado

A regulação por realimentação de estados somente será possível se o modelo em espaço de estados for *controlável*. Também, devemos lembrar que o caso regulatório ou de regulação se refere a quando o sinal de referência do sistema de controle é nulo, i.e. $y_r(t) = 0$. Isto apenas torna a compreensão do problema de realimentação de estados mais simples, deixando o caso **servo** ou de seguimento de referência para o capítulo seguinte.

A regulação de um sistema tipo SLIT visa mantê-lo sempre no zero do modelo linear, $\begin{bmatrix} x_1 & \cdots & x_n \end{bmatrix}^T = \begin{bmatrix} 0 & \cdots & 0 \end{bmatrix}^T$, que corresponde ao ponto de equilíbrio no qual o sistema não-linear foi linearizado. O problema adicional é poder especificar também como esses estados serão levados a zero, ou seja, o condicionamento do transitório do sistema que consequentemente implica em modificar a sua matriz de transição de estados em M.F..

Uma atraente característica da regulação via realimentação de estados é que nada, além do modelo em espaço de estados, precisa ser conhecido para que se defina uma estrutura de controlador. Isto significa que a solução para o caso de realimentação total de estados fornece inerentemente a estrutura adequada que resolve o problema de *alocação dos autovalores de malha fechada* para garantir a dinâmica desejada de transição de estados. Isto foi fundamental para o avanço dos sistemas aeroespaciais que surgiram a partir da década de 1960, justamente porque não se sabia quais eram as melhores topologias de controle que resolviam os problemas, que eram do tipo MIMO e muitas vezes, instáveis (STEVENS et al., 2016).

Apesar de resolver problemas de grande complexidade, a estrutura de realimentação de estados é bastante simples e pode ser representada conforme o diagrama de blocos da Fig. 25. Observe que a complexidade reside no modelo em espaço de estados, exigindo que o vetor de estados x esteja disponível, tal que a **lei de controle** seja, simplesmente,

$$u = -Kx \quad (3.32)$$

Ao substituir a lei de controle na equação de estados, obtém-se

$$\dot{x} = Ax + B(-Kx) \quad (3.33)$$

que por sua vez define o sistema de malha fechada no espaço de estados:

$$\dot{x} = (A - BK)x = A_{cl}x \quad (3.34)$$

Note que em (3.34) há uma nova matriz de estados, A_{cl} . Consequentemente, há nova forma de transição de estados, dada agora por $(sI - A + BK)^{-1}$, já que os autovalores de malha fechada agora são outros, dados por

$$\det(sI - A + BK) = 0 \quad (3.35)$$

O problema então se resume em determinar a matriz de ganhos K que realoque os autovalores da matriz $(A - BK)$ na região desejada.

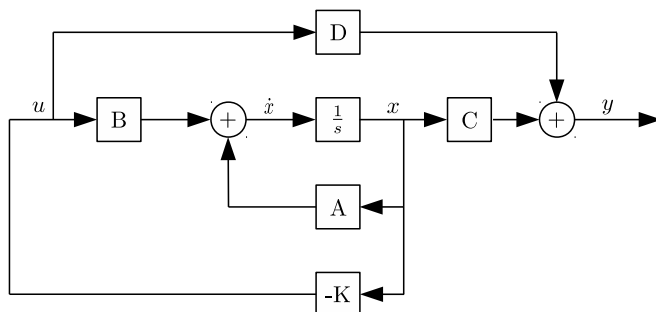


Figura 25 – Diagrama de blocos de regulação por realimentação de estados.

Obs: deve-se destacar que alguns autores não apresentam a realimentação negativa explicitamente, tal que $u = Kx$, fazendo com que a matriz de malha fechada modifique-se para $A_{cl} = (A + BK)$.

Exemplo 3.7 *Determine a matriz K , de realimentação de estados, capaz de alocar os pólos de malha fechada nas posições $-2 \pm j2\sqrt{2}$ (MAYA; LEONARDI, 2014, ex. 15.14, pág. 281):*

$$\begin{aligned} \dot{x} &= \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \\ y &= \begin{bmatrix} 1 & 0 \end{bmatrix} x \end{aligned}$$

O objetivo deste exercício é determinar o vetor de ganhos $K =$

$\begin{bmatrix} k_1 & k_2 \end{bmatrix}$, tal que a lei de controle

$$u = \begin{bmatrix} k_1 & k_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

aloque os autovalores da matriz $A_{cl} = (A - BK)$ em $-2 \pm j2\sqrt{2}$.

Devemos então transformar nossos objetivos em uma expressão matemática. Nesse sentido, o polinômio característico desejado de malha fechada é

$$\Delta_{cl}(s) = (s + 2 - j2\sqrt{2})(s + 2 + j2\sqrt{2}) = s^2 + 4s + 12$$

Para se determinar o ganho K pode-se então desenvolver a equação que relaciona a matriz $A_{cl} = (A - BK)$ com o polinômio característico de malha fechada, isto é,

$$\Delta_{cl}(s) = \det [sI - (A - BK)].$$

$$\Delta_{cl}(s) = \det \left[\begin{pmatrix} s & -1 \\ 2 & s+3 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} \begin{pmatrix} k_1 & k_2 \end{pmatrix} \right] = s^2 + (3 + k_2)s + 2 + k_1$$

Igualando-se então o polinômio característico desejado de malha fechada com o polinômio característico obtido em função de k_1, k_2 , chega-se então na expressão que resolve o problema:

$$s^2 + (3 + k_2)s + 2 + k_1 = s^2 + 4s + 12$$

$$2 + k_1 = 12 \rightarrow \boxed{k_1 = 10}$$

$$(3 + k_2) = 4 \rightarrow \boxed{k_2 = 1}$$

Portanto, o vetor de ganhos que garante a alocação dos pólos de malha fechada desejados é

$$K = \begin{bmatrix} 10 & 1 \end{bmatrix}$$

Obs: a forma de solução apresentada neste exemplo é interessante para compreendermos o problema, mas não é favorável para resolver casos mais complexos e de ordem elevada, por exemplo, para sistemas MIMO. Por isto, veremos a seguir como determinar o ganho de realimentação por uma

método mais eficiente, pelo uso da Fórmula de Ackermann.

3.6.4 Determinação do ganho de realimentação pelo método de Ackermann

Como sempre antes de executar um projeto de realimentação de estados devemos verificar se o sistema é controlável, pressupõe-se que a matriz de controlabilidade C_o esteja disponível para o projetista. Nesse sentido, é possível utilizar esse conhecimento para calcular o ganho de realimentação de estados K através do método de Ackermann, dado por:

$$K = [0 \quad 0 \quad \cdots \quad 0 \quad 1] (C_o)^{-1} \Delta_{cl}(A) \quad (3.36)$$

onde $\Delta_{cl}(A)$ é uma função polinomial matricial, construída com base no polinômio característico desejado de malha fechada e na matriz de estados A , do sistema que se quer regular.

Para se calcular $\Delta_{cl}(A)$ basta verificar a relação que este assume com a variável s , isto é,

$$\Delta_{cl}(A) \leftrightarrow \Delta_{cl}(s). \quad (3.37)$$

Tomemos o seguinte polinômio para ilustrar um exemplo:

$$\begin{aligned} \Delta_{cl}(s) &= s^2 + 3s + 2 \\ \Delta_{cl}(A) &= A^2 + 2A + 2I \end{aligned} \quad (3.38)$$

Exemplo 3.8 *Repita o Exemplo 3.7 pelo método de Ackermann:*

$$K = [0 \quad 0 \quad \cdots \quad 0 \quad 1] (C_o)^{-1} \Delta_{cl}(A)$$

A matriz de controlabilidade é

$$C_o = [B \quad AB] = \begin{bmatrix} 0 & 1 \\ 1 & -3 \end{bmatrix}$$

e a inversa de C_o é

$$(C_o)^{-1} = \frac{\text{Adj}(C_o)}{\det(C_o)} = \frac{[\text{cof}(C_o)]^T}{\det(C_o)} = \begin{bmatrix} 3 & 1 \\ 1 & 0 \end{bmatrix}$$

O polinômio característico desejado de malha fechada, referente

Tabela 15 – Código em MATLAB referente ao Exemplo 3.8.

```

A=[0 1;-2 -3]; B=[0;1]; C=[1 0]; D=0;
Co=ctrb(A,B); % matriz de controlabilidade
P=[-2+j*2*sqrt(2) -2-j*2*sqrt(2)]; % pólos/autovalores desejados de M.F.
K=acker(A,B,P) % função da fórmula de Ackermann
eig(A-B*K) % Eigenvalues (autovalores) da matriz de M.F.

sys_cl=ss( (A-B*K), [0;0], C, D ); % sistema em M.F.
t=0:0.1:10; nit=length(t); % vetor tempo de simulação e número de iterações
yr(1:nit)=0; % referência nula para regulação
x0=[1;0]; % condição inicial dos estados x1 e x2, x=[x1;x2];
lsim(sys_cl,yr,t,x0); % simulação linear do sistema em M.F. sendo regulado

```

à alocação de pólos em $-2 \pm j2\sqrt{2}$, é dado por $\Delta_{cl}(s) = s^2 + 4s + 12$, permitindo assim calcular

$$\Delta_{cl}(A) = A^2 + 4A + 12I$$

$$\Delta_{cl}(A) = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix}^2 + 4 \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix} + 12 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 10 & 1 \\ -2 & 7 \end{bmatrix}$$

Pela Fórmula de Ackermann calcula-se então o seguinte ganho de realimentação de estados:

$$K = \begin{bmatrix} 0 & 1 \end{bmatrix} (C_o)^{-1} \Delta_{cl}(A) = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} 3 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 10 & 1 \\ -2 & 7 \end{bmatrix}$$

$$K = \begin{bmatrix} 10 & 1 \end{bmatrix}$$

Os exemplos 3.7 e 3.8 são facilmente realizáveis com o auxílio do Matlab, Scilab ou outro software que possua a contribuição do matemático alemão, Ackermann, já na forma de uma função pronta. Na Tabela 15 mostra-se como seria possível resolver e testar o Exemplo 3.8 no Matlab.

3.7 OBSERVADOR DE ESTADO

O problema de observação de estado é dual ao problema de controle por realimentação de estado. Isto se deve, pois o problema de estimar estados em uma estratégia com realimentação (i.e., estimação em malha fechada), pode ser visto como um sistema de controle que

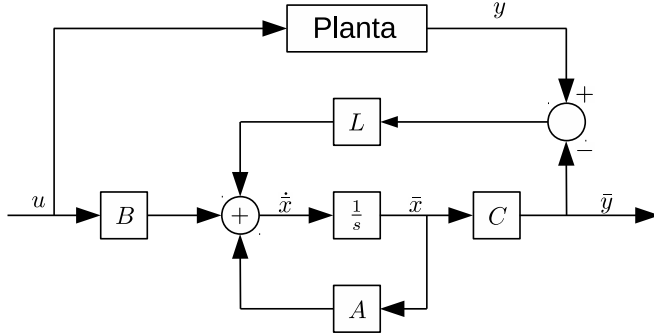


Figura 26 – Diagrama de blocos de um Observador de Estados.

busca levar a saída do estimador, $\bar{y}(t)$, à saída medida $y(t)$.

Pode-se fazer uma analogia com o problema de levar a saída medida a seguir um sinal de referência $y_r(t)$. Logo, da mesma forma que um controlador deve minimizar o erro de rastreamento, $e(t) = y_r(t) - y(t)$, o observador deve minimizar o erro de estimação da saída medida, isto é,

$$e_{est}(t) = y(t) - \bar{y}(t) = Cx(t) - C\bar{x}(t), \quad (3.39)$$

onde $\bar{x}(t)$ é o **vetor de estado estimado**.

Observe, na Eq. (3.39), que se a saída estimada convergir à saída medida, o erro de estimação tenderá a zero e, consequentemente,

$$\bar{x}(t) \rightarrow x(t) \quad (3.40)$$

A estrutura do Observador de Estado, mostrada na Fig. 26, está relatada em praticamente todos os livros de Controle e Sistemas e apesar da simplicidade, a mesma estrutura pode ser projetada de maneira heurística, como no caso do Observador de Luenberger, ou de maneira ótima e estocástica, como no caso do Filtro de Kalman (STEVENSON et al., 2016). De todo o modo, esta clássica estrutura pode permitir que a *virtualização* das realizações em espaço de estados se torne realidade, do ponto de vista da engenharia, fornecendo os estados estimados $\bar{x}(t)$ para promover a realimentação de estados.

É importante destacar, na Fig. 26, que a *caixa-preta* representada pelo bloco *Planta*, simplesmente não possui qualquer saída relativa ao seu vetor de estados $x(t)$. Esse é exatamente o cenário real para a aplicação dos observadores. A ideia é que o modelo de estados do obser-

vador recebe o mesmo estímulo de entrada e , havendo diferença entre $y(t)$ e $\bar{y}(t)$, esta é processada pelo ganho de realimentação do estimador, ou **ganho do estimador**, L . As equações que regem a dinâmica do estimador são

$$\begin{aligned}\frac{d}{dt}\bar{x} &= A\bar{x} + Bu + Le_{est} \\ \bar{y} &= C\bar{x}\end{aligned}\tag{3.41}$$

O erro de estimação, quando reescrito em função do vetor de estados estimados, traz as equações do observador para um cenário ainda mais amigável para quem já estudou controle por realimentação de estados, tal que

$$\begin{aligned}\frac{d}{dt}\bar{x} &= A\bar{x} + Bu + L(y - C\bar{x}) \\ \frac{d}{dt}\bar{x} &= (A - LC)\bar{x} + Bu + Ly\end{aligned}\tag{3.42}$$

Note que o observador é então um sistema que pode ter múltiplas entradas, a partir dos vetores $u(t)$ e $y(t)$, e a sua dinâmica é governada pelos autovalores da matriz $(A - LC)$. Sendo assim, os métodos de projeto aplicáveis ao caso de controle por realimentação de estado servem para resolver os problemas de observação de estado, bastando para isto que a realização seja observável. Como no caso do observador o problema se configura na direção da saída medida para os estados, a seguinte **dualidade** de projeto se estabelece:

Controlador		Observador
A	\leftrightarrow	A^T
B	\leftrightarrow	C^T
K	\leftrightarrow	L^T
C_o	\leftrightarrow	O_b^T

Com o uso da dualidade, o projeto de alocação de pólos via Fórmula de Ackermann pode ser diretamente utilizado da seguinte maneira:

- Ganho de controlador:

$$K = \begin{bmatrix} 0 & \cdots & 0 & 1 \end{bmatrix} (C_o)^{-1} \Delta_K(A)$$

- Ganho de observador:

$$L^T = [0 \quad \cdots \quad 0 \quad 1] (O_b^T)^{-1} \Delta_L(A^T) \quad (3.43)$$

ou

$$L = \Delta(A)(O_b)^{-1} [0 \quad \cdots \quad 0 \quad 1]^T \quad (3.44)$$

onde $\Delta_K(\cdot)$ e $\Delta_L(\cdot)$ são os polinômios característicos desejados em malha fechada para o controlador e observador, respectivamente, expressos no domínio contínuo s ou no discreto z .

A solução do problema de alocação dos autovalores de $(A - LC)$ sem a utilização da Fórmula de Ackermann também se dá da mesma maneira que no caso controlador, igualando-se o polinômio característico de MF com o polinômio característico desejado:

$$\det[sI - (A - LC)] = \Delta_L(s). \quad (3.45)$$

O ganho do estimador, no caso SISO, assume a forma de um vetor coluna, como

$$L = \begin{bmatrix} L_1 \\ L_2 \\ \vdots \\ L_n \end{bmatrix} \quad (3.46)$$

Obs 1: como o caso de observação ou estimação de estado é dual ao problema de controle, está sujeito aos similares problemas, como, por exemplo: ter offset não nulo em regime permanente; ter dinâmica oscilatória; ser sensível ao ruído de medição; entre outros. Nesse sentido, toda a análise de convergência, de estabilidade, de robustez, é fundamental de ser realizada e respeitará a dualidade entre projetos.

Obs 2: para sistemas do tipo-0 (sistemas não integradores ou sem pólos na origem do plano s), no projeto do observador pode ser necessário aumentar o modelo com a inclusão do integrador para garantir que haja offset nulo em relação ao erro de estimação. Caso contrário, sempre haverá algum offset na prática, pois sempre, na prática, o modelo será diferente da planta real. Também, as perturbações de entrada e saída geram um cenário de descasamento entre planta e modelo.

3.7.1 Teste de observador de estado

Assim como no caso controlador, há como verificar a convergência de resposta ao degrau e as propriedades de robustez a partir da análise da malha direta do observador. No entanto, como se trata de um sistema que pode ter múltiplas entradas, a partir de $u(t)$ e $y(t)$, o que pode dificultar a análise do sistema observador, recomenda-se simplificar essa análise a partir da entrada de maior interesse no problema de estimação de estado, que é a saída medida, já que esta pode conter perturbações de baixa e alta frequência, enquanto que a entrada $u(t)$ é, geralmente, um sinal conhecido. Sendo assim, os testes podem ser realizados com base nas seguintes equações:

- Teste de convergência da resposta ao degrau em M.F.:

$$\frac{d}{dt}\bar{x} = (A - LC)\bar{x} + Ly \quad (3.47)$$

$$\bar{y} = C\bar{x} \quad (3.48)$$

Nessa estrutura, a matriz de entrada vira a própria matriz L , tendo como variável de entrada a saída medida $y(t)$. Entenda que este teste permite que se verifique como a saída estimada $\bar{y}(t)$ segue a saída medida $y(t)$. Espera-se que essa convergência seja o mais precisa possível, para que se garanta que os estados também estejam sendo adequadamente estimados.

- Teste de resposta em frequência com base na malha direta de observador:

$$\begin{aligned} \frac{d}{dt}\bar{x} &= A\bar{x} + Ly \\ \bar{y} &= C\bar{x} \end{aligned} \quad (3.49)$$

sendo este equivalente à seguinte relação de ganhos no domínio s :

$$\frac{\bar{Y}(s)}{Y(s)} = C(sI - A)^{-1}L \quad (3.50)$$

Note que as equações (3.49) e (3.50) refletem a dinâmica de malha direta do sistema observador de estado, permitindo que o engenheiro possa avaliar as características de ganho e fase para a adequada seleção da frequência de cruzamento para a atenuação de ruídos e incertezas de alta frequência, além de poder avaliar as margens de ganho e de fase

pelo método de Bode, verificando a robustez do observador de estado na solução do problema de fazer a saída estimada $\bar{y}(t)$ seguir a saída medida $y(t)$, tal como um controlador busca o seguimento de um sinal de referência.

4 CAPÍTULO 4

Compensadores dinâmicos (controlador + observador). Augmentação do modelo pela inclusão de integrador. Controladores digitais no espaço de estado.

4.1 INTRODUÇÃO

Neste capítulo iremos estender nosso conhecimento sobre controle via realimentação total de estado, acrescentando o caso servo ou de seguimento de referência, estudando o projeto de observador de estado com a estrutura do Observador de Luenberger e como chegar à forma conjunta *controlador + observador*, designada pelo nome de *compensador dinâmico*.

Finalizaremos o conteúdo da disciplina retornando ao Controle Digital, no qual estudaremos o projeto de compensadores dinâmicos digitais e como realizar a simulação destes sistemas em laço iterativo tipo *while* ou *for*, além de implementar esses compensadores dinâmicos digitais em protótipos didáticos reais usando uma interface de aquisição de dados para que seja possível embarcar o sistema de controle em um computador usando Matlab, Scilab ou outros ambientes de programação.

Nesta etapa vamos considerar projetos com base em realimentação total de estado, mas que possam permitir o seguimento de referência e, preferencialmente, com *offset* nulo em regime permanente.

Antes de prosseguirmos, o leitor deve estar ciente que o caso servo ou de seguimento de referência, no projeto de controladores no espaço de estado, possui uma vasta gama de métodos relatados na literatura de Controle Moderno. Cada método acrescenta ao estudante alguma habilidade adicional no que se refere à compreensão da Álgebra Linear, da Teoria de Sistemas Lineares e da Teoria de Controle.

Como não há tempo para relatar, neste documento, um grande número de métodos de projeto, espera-se que o leitor possa buscar na bibliografia do curso, outros métodos que complementem sua formação como projetista de sistemas de controle, estimadores e filtros no espaço de estados. Principalmente, que avancem para os métodos **ótimos**, que englobam o método do Regulador Linear Quadrático (LQR) e o Filtro de Kalman (FK), culminando no compensador dinâmico do tipo Linear Quadrático Gaussiano (LQG) (STEVENSON et al., 2016).

4.2 PROJETO DE CONTROLADORES NO ESPAÇO DE ESTADOS

No projeto de controladores normalmente há a necessidade de seguimento de referência e com a garantia de *offset* nulo. Nesse sentido, apenas lembremos quão difundido é o uso de controladores do tipo Proporcional-Integral (PI) na indústria e pensemos nos porquês. Alguns destes porquês são a simplicidade do controlador e a consequente confiabilidade, mas isto se deve, principalmente, à ação integral que o PI traz, que quando aplicada a sistemas do tipo-0 (sem integrador natural), garante a elevação do ganho de malha direta nas baixas frequências, garantindo rejeição às perturbações e seguimento de referência com erro nulo em regime permanente.

A influência da ação Integral pode ser observada nos diagramas de Bode como no exemplo da Fig. 27, em que há o decaimento, a partir da frequência zero, de 20dB/déc, em decorrência do integrador $1/s$ na malha direta, que neste exemplo é $C(s)G(s) = 1/[s(s+1)]$. A questão é aprender como se pode incluir esta ação Integral no projeto em espaço de estados.

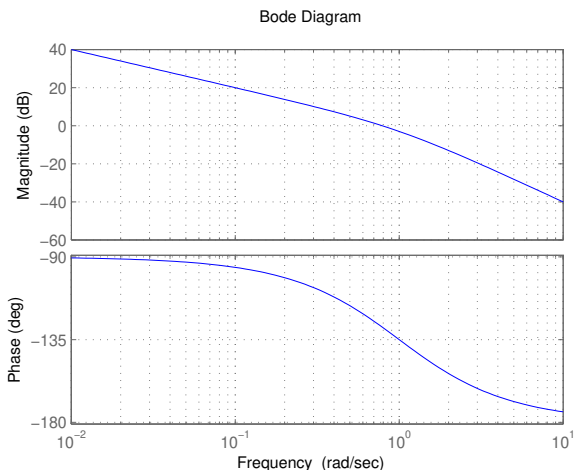


Figura 27 – Diagrama de Bode com ganho infinito na frequência zero e o decaimento de 20dB/déc que atesta a presença da ação Integral na malha direta, $C(s)G(s)$.

Conforme vimos no final do Capítulo 3, a complexidade dos controladores em espaço de estados reside na própria realização do sistema,

já que o controlador em si é baseado simplesmente em uma matriz ou vetor de ganhos K . Por isto, para que se tenha ação Integral em um controlador obtido com base em modelo em espaço de estados, deve-se incluir o integrador ao modelo no espaço de estados. Este procedimento é conhecido na literatura como **augmentação do modelo de estados pela inclusão de integrador**.

Para entender o processo, consideremos um exemplo de um sistema estável e do tipo-0. Isto é, um sistema sem integrador natural, definido pelas seguintes expressões:

$$\begin{aligned} K_p &= 0.9, \quad \omega_n = 2\text{rad/s}, \quad \zeta = 0.2 \\ G(s) &= \frac{Y(s)}{U(s)} = \frac{K_p \omega_n^2}{s^2 + 2\zeta \omega_n s + \omega_n^2} \\ \ddot{y} + 2\zeta \omega_n \dot{y} + \omega_n^2 y &= K_p \omega_n^2 u \\ x_1 &= y, \quad x_2 = \dot{y} \\ \dot{x}_2 &= -\omega_n^2 x_1 - 2\zeta \omega_n x_2 + K_p \omega_n^2 u \end{aligned} \quad (4.1)$$

$$\begin{aligned} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ -\omega_n^2 & -2\zeta \omega_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ K_p \omega_n^2 \end{bmatrix} u \\ y &= \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \end{aligned} \quad (4.2)$$

Na Fig. 28 são mostradas as respostas no tempo e em frequência das representações por função de transferência e no espaço de estados, onde fica evidente a ausência do integrador natural e a confirmação de que se trata de um sistema do tipo-0. Nesta situação, um projeto de controle por realimentação de estados com base nesta realização em espaço de estados, sem integrador, acarretaria solução de controle sem ação Integral e consequente offset não nulo no seguimento de referência. Há então a necessidade de augmentação com integrador.

De uma forma forçada, a augmentação pela adição do integrador seria equivalente a acrescentar $1/s$ à entrada da planta, ficando assim o sistema aumentado, $G_a(s)$, descrito por

$$G_a(s) = \frac{1}{s} G(s) = \frac{Y(s)}{sU(s)} = \frac{1}{s} \left[\frac{K_p \omega_n^2}{s^2 + 2\zeta \omega_n s + \omega_n^2} \right] \quad (4.3)$$

em que $U_a(s) = sU(s)$, que pode ser interpretado como sendo a **variação do sinal de controle** $U(s)$. Isto significa que o modelo aumentado (*aug-*

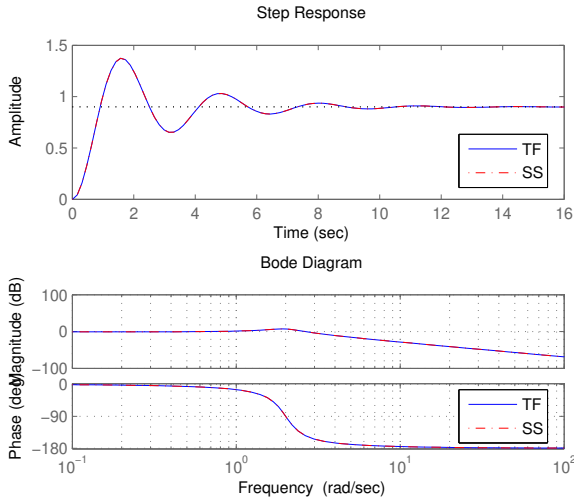


Figura 28 – Resposta no tempo e em frequência de sistema: representação por função de transferência (TF) e no espaço de estados (SS) de um mesmo sistema tipo-0.

mented) de projeto, $G_a(s)$, passa a ser uma *virtualização* cuja entrada agora é a variação da entrada, $U_a(s) = sU(s)$. Nesta configuração a ordem final do modelo aumentado cresce, ficando este na seguinte forma:

$$Y(s) (s^3 + 2\zeta\omega_n s^2 + \omega_n^2 s) = K_p \omega_n^2 \underbrace{U_a(s)}_{sU(s)}. \quad (4.4)$$

Uma possível realização para o modelo de projeto aumentado, mostrado em (4.4), pode ser obtida fazendo-se

$$\begin{aligned} {}^a x_1 &= y, & {}^a x_2 &= \dot{y}, & {}^a x_3 &= \ddot{y}, \\ {}^a \dot{x}_1 &= {}^a x_2 \\ {}^a \dot{x}_2 &= {}^a x_3 \\ {}^a \dot{x}_3 &= -\omega_n^2 ({}^a x_2) - 2\zeta\omega_n ({}^a x_3) + K_p \omega_n^2 \underbrace{(u_a)}_{\dot{u}} \end{aligned} \quad (4.5)$$

Este sistema aumentado exhibe as respostas ao degrau e em frequência, como mostrado na Fig. 29. Observe que o degrau de entrada é, de fato,

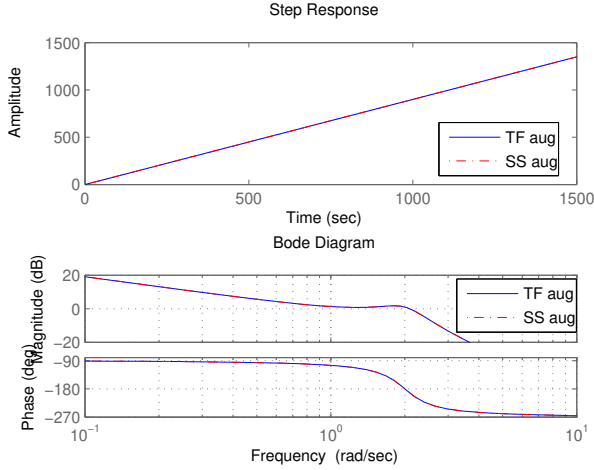


Figura 29 – Resposta no tempo e em frequência de sistema tipo-0 aumentado pela inclusão de integrador: representação por função de transferência (TF aug) e no espaço de estados (SS aug) de um mesmo sistema tipo-0 aumentado pela inclusão do integrador.

integrado ao longo do tempo, tal que a saída do sistema aumentado é uma rampa, já que a integral do degrau é uma rampa. Além disto, o Diagrama de Bode apresenta o decaimento de 20db/déc.

A descrição desse sistema aumentado, no espaço de estados, é mostrada a seguir:

$$\dot{x}_a = \begin{bmatrix} {}^a\dot{x}_1 \\ {}^a\dot{x}_2 \\ {}^a\dot{x}_3 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -\omega_n^2 & -2\zeta\omega_n \end{bmatrix}}_{A_a} x_a + \underbrace{\begin{bmatrix} 0 \\ 0 \\ K_p\omega_n^2 \end{bmatrix}}_{B_a} \underbrace{(u_a)}_{\dot{u}} \quad (4.6)$$

$$y_a = \underbrace{\begin{bmatrix} C & 0 \end{bmatrix}}_{C_a} x_a = \underbrace{\begin{bmatrix} 1 & 0 & 0 \end{bmatrix}}_{C_a} x_a$$

Note que as novas variáveis de estado, do vetor aumentado $x_a^T = [{}^ax_1 \quad {}^ax_2 \quad {}^ax_3]$, não são as mesmas do vetor do **modelo nominal**, não aumentado, $x^T = [x_1 \quad x_2]$, apesar de possuírem uma relação que ficará explícita mais adiante no texto, com a generalização do procedimento de *aumento com integrador*.

O problema de controle servo no espaço de estados é formulado

com base em uma lei de controle mais genérica. Essa lei de controle tem a seguinte forma:

$$u = K(r - x) = \begin{bmatrix} k_1 & k_2 & \cdots & k_n \end{bmatrix} \left(\begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \end{bmatrix} - \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \right) \quad (4.7)$$

onde r é um vetor de referências para as variáveis de estado e que quando $r = 0$, então $u = -Kx$ e também o caso regulatório é contemplado.

Não há como resolver o problema servo para todas as variáveis de estado, ficando o problema servo restrito ao número de variáveis de entrada disponíveis. Portanto, para um sistema SISO como o mostrado em (4.4), apenas um dos sinais do vetor r será não nulo. Como exemplo, consideremos o problema servo para $^a x_1 = y$, tal que o único sinal de referência não nulo seja r_1 , ficando a lei de controle composta de uma parte regulatória e outra de seguimento de referência para a variável de estado $^a x_1$ (a própria saída do sistema):

$$u_a = \begin{bmatrix} k_1 & k_2 & k_3 \end{bmatrix} \left(\begin{bmatrix} r_1 \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} ^a x_1 \\ ^a x_2 \\ ^a x_3 \end{bmatrix} \right) \quad (4.8)$$

$$u_a = \overbrace{k_1 r_1 - K x_a}^{\text{Servo}} \quad (4.9)$$

Regulacao

ou mesmo, usando a simbologia dos capítulos anteriores em que o sinal de referência era y_r ,

$$u_a = \overbrace{k_1 y_r - K x_a}^{\text{Servo}} \quad (4.10)$$

Regulacao

Na situação do projeto com modelo aumentado, veremos que uma peculiaridade deve ser destacada:

- O sinal de controle é $u_a = \dot{u}$, mas a planta real **precisa...** isto mesmo, precisa receber na entrada, o sinal u , já que u_a é uma virtualização matemática. Sendo assim, deve-se aplicar à entrada

da planta real (que por simulação é o modelo nominal), o seguinte sinal:

$$u(t) = \int u_a(t) dt \quad (4.11)$$

Os efeitos da lei de controle do projeto com augmentação sobre o modelo aumentado pode ser avaliado pela simples substituição de (4.10) na equação de estados, que fornece

$$\dot{x}_a = A_a x_a + B_a u_a = A_a x_a + B_a (-K x_a + k_1 y_r), \quad (4.12)$$

ficando então, o sistema aumentado em malha fechada, definido como

$$\begin{aligned} \dot{x}_a &= (A_a - B_a K) x_a + (k_1 B_a) y_r \\ y_a &= C_a x_a \end{aligned} \quad (4.13)$$

A análise do projeto em espaço de estados compreenderá, normalmente, a verificação da alocação dos autovalores da matriz $(A_a - B_a K)$, teste de convergência do sistema em M.F. pela resposta ao degrau, testes de perturbações tipo carga e ruído, além dos testes de resposta em frequência da malha direta, podendo estes serem feitos conforme se destaca a seguir:

- Resposta ao degrau em M.F.: o modelo mostrado em (4.13) pode ser facilmente definido no Matlab ou Scilab para que se verifique a resposta em malha fechada com o ganho K projetado;
- Resposta em frequência da malha direta: faz-se da mesma forma que no caso por função de transferência, pela conexão em série do controlador com a planta, $C(s)G(s)$, que no caso em espaço de estados seria a mera multiplicação do ganho de realimentação K sobre os estados em x_a , sendo isto equivalente à substituição da matriz/vetor C_a pela matriz/vetor K , tal que

$$\begin{aligned} \dot{x}_a &= A_a x_a + B_a u_a \\ z_a &= K x_a \end{aligned} \quad (4.14)$$

A relação com o caso de ganho de malha direta por função de transferência é facilmente verificada ao se converter o sistema em

(4.14) para o domínio s , tal que

$$\frac{Z_a(s)}{U_a(s)} = K(sI - A_a)^{-1}B_a. \quad (4.15)$$

A aplicação dos procedimentos de projeto apresentados sobre o sistema descrito em (4.1) e (4.2), para uma alocação dos autovalores de $(Aa - B_aK)$ em $-1, -1, -1$, escolhidos com base na característica de garantir resposta sobre amortecida e constante de tempo de 1 segundo, resultou no ganho de realimentação dado por

$$K = \begin{bmatrix} 0.2778 & -0.2778 & 0.6111 \end{bmatrix}. \quad (4.16)$$

Os resultados do exemplo desenvolvido, para o teste de resposta ao degrau, são mostrados na Fig. 30, contendo a variação do sinal de controle ($u_a(t)$) e a saída controlada ($y(t)$). Observe que quando a saída estabiliza com offset nulo em regime permanente, a variação de controle tende a zero. No entanto, esta é uma virtualização matemática do modelo aumentado, sendo necessário que $u_a(t)$ seja integrado e aplicado na forma de $u(t)$ à entrada da planta real.

Na Fig. 31 são apresentadas as variáveis de estado do modelo aumentado, consideradas disponíveis para a realimentação, neste exemplo. Observe que apenas $^ax_1(t) = y$ realiza o seguimento de referência do degrau, enquanto que as duas outras variáveis de estado são reguladas a zero.

Por fim, na Fig. 32 mostra-se a análise das margens de estabilidade da malha direta, que se faz com o modelo de estados em M.A. multiplicado pelo ganho de realimentação K . Nota-se que a ação Integral está presente na malha direta, que de fato é confirmada também pelo teste de convergência à entrada degrau na referência $y_r(t)$.

O código para reproduzir esta simulação no Matlab é mostrado na Tabela 16.

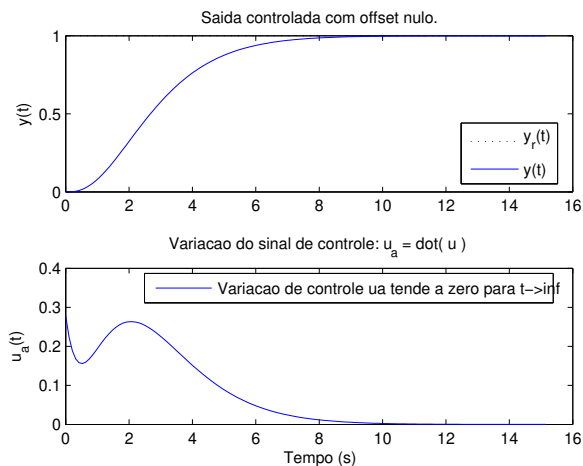


Figura 30 – Variação do sinal de controle e saída controlada com offset nulo.

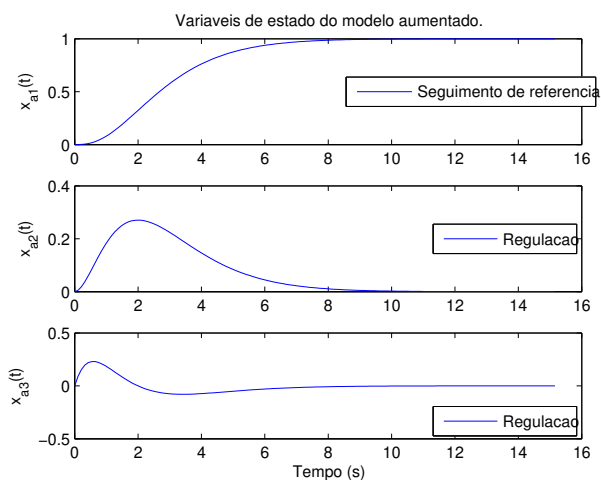


Figura 31 – Variáveis de estado consideradas disponíveis para a realimentação. Destaque para o controle servo na variável $x_1(t)$, enquanto as outras duas sofrem regulação a zero.

Tabela 16 – Código em MATLAB referente ao projeto de controlador no espaço de estados com modelo aumentado com integrador.

```

%% Controle por realimentacao de estados: caso contínuo
clear all; close all; clc;
% Sistema linear contínuo descrito por FT
Kp=0.9; zeta=0.2; wn=2;
Gs = tf(Kp*wn^2,[1 2*zeta*wn wn^2]);
% Realizacao em state space com x1=y e x2=dot(y)
A = [0 1; -wn^2 -2*zeta*wn]; B=[0;Kp*wn^2]; C=[1 0]; D=0;
sys=ss(A,B,C,D);
% Teste dos modelos TF e SS
figure(1);
subplot(211),
step(Gs,sys); legend('TF','SS',0);
subplot(212),
bode(Gs,sys), grid;
legend('TF','SS',0);
%% Aumentacao dos modelos
% Caso TF
Gsa = tf(Kp*wn^2,[1 2*zeta*wn wn^2 0]);
% Caso SS
Aa = [0 1 0; 0 0 1; 0 -wn^2 -2*zeta*wn]; Ba = [0;0;Kp*wn^2];
Ca = [1 0 0]; Da=0;
sys_aug = ss(Aa,Ba,Ca,Da);
% Teste dos modelos TF e SS aumentados
figure(2);
subplot(211),
step(Gsa,sys_aug); legend('TF aug','SS aug',0);
subplot(212),
bode(Gsa,sys_aug), grid;
legend('TF aug','SS aug',0);
%% Projeto do controlador
% Analise de controlabilidade
Co = ctrb(Aa,Ba);
rank(Co)
det(Co)
% Analise dos polos de malha aberta
P=real(eig(Aa)) % polos desejados de malha fechada com base
% na parte real dos de malha aberta
P=[-1 -1 -1]';
Kwacker(Aa,Ba,P)
% Resolvendo a Eq. de Ackermann 'manualmente'
PMF=poly(P); % polinomio desejado de MF
PMF_A= PMF(1)*Aa^3 +PMF(2)*Aa^2 +PMF(3)*Aa +PMF(4)*eye(3);
K=[0 0 1]*inv(Co)*PMF_A
k1=K(1); % elemento do ganho que trata a referencia (K=[k1 k2 k3])
eig(Aa-Ba*K) % autovalores desejados alocados
%% Analise do sistema de controle projetado
% Convergencia pela resposta ao degrau (sistema MF)
Acl = Aa-Ba*K;
Bcl = k1*Ba;
Ccl = Ca;
sys_cl = ss(Acl, Bcl, Ccl, 0); % closed-loop
% Dados para a simulacao
[y,t,xa]=step(sys_cl);
nit = length(t); % numero de iteracoes da simulacao
yr=ones(1,nit); % vetor de referencia
ua=-K*xa'+k1*yr; % sinal de controle usado internamente pelo Matlab
figure(3);
subplot(211),
plot(t,yr,':k',t,y,'b'); ylabel('y(t)');
legend('y_r(t)','y(t)',0);
title('Saída controlada com offset nulo.');
```

```

subplot(212),
plot(t,ua,'b'); ylabel('u_a(t)'); xlabel('Tempo (s)');
legend('Variacao de controle ua tende a zero para t->inf',0);
title('Variacao do sinal de controle: u_a = dot( u ) ');
figure(4); % variaveis de estado
subplot(311),
plot(t,xa(:,1)); ylabel('x_a1(t)');
legend('Seguimento de referencia',0);
title('Variaveis de estado do modelo aumentado.');
```

```

subplot(312),
plot(t,xa(:,2)); ylabel('x_a2(t)');
legend('Regulacao',0);
subplot(313),
plot(t,xa(:,3)); ylabel('x_a3(t)'); xlabel('Tempo (s)');
legend('Regulacao',0);
% Analise em frequencia (caso em MA): malha direta
Amd = Aa; Bmd = Ba; Cmd = K;
sys_md = ss(Amd, Bmd, Cmd, 0);
figure(5);
margin(sys_md);
```

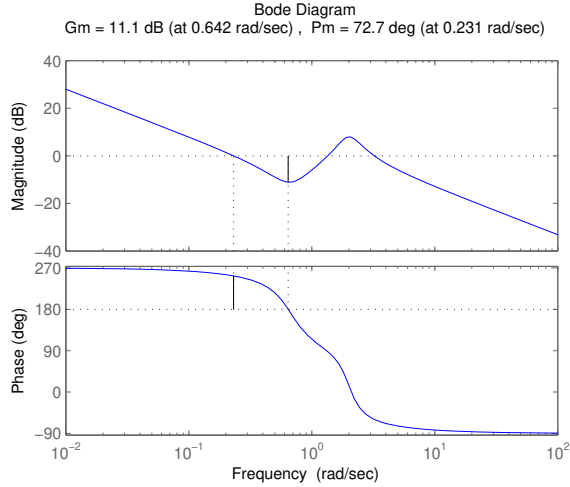


Figura 32 – Análise das margens de ganho e de fase usando a malha direta em espaço de estados.

4.3 GENERALIZAÇÃO DO PROCEDIMENTO DE AUGMENTAÇÃO PELA INCLUSÃO DE INTEGRADOR

4.3.1 Augmentação pela variação do sinal de controle, $u_a(t)$:

Pode-se generalizar o método de augmentação do modelo de estados pela derivação direta do modelo nominal de projeto, isto é, derivando-se

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned} \quad (4.17)$$

chega-se à forma

$$\begin{aligned} \ddot{x} &= A\dot{x} + B\dot{u} \\ \dot{y} &= C\dot{x} \end{aligned} \quad (4.18)$$

Ao se considerar como vetor de estados aumentados, o vetor

$$x_a = \begin{bmatrix} y \\ \dot{x} \end{bmatrix} \quad (4.19)$$

verifica-se que o modelo mostrado a seguir pode ser considerado como um modelo canônico para augmentação para a inclusão de integradores:

$$\underbrace{\begin{bmatrix} \dot{y} \\ \ddot{x} \end{bmatrix}}_{\dot{x}_a} = \underbrace{\begin{bmatrix} 0 & C \\ 0 & A \end{bmatrix}}_{A_a} \underbrace{\begin{bmatrix} y \\ \dot{x} \end{bmatrix}}_{x_a} + \underbrace{\begin{bmatrix} 0 \\ B \end{bmatrix}}_{B_a} \underbrace{\dot{u}}_{u_a} \quad (4.20)$$

$$\underbrace{y}_{y_a} = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{C_a} \begin{bmatrix} y \\ \dot{x} \end{bmatrix}$$

Conforme mencionado no início deste capítulo, há inúmeras formas de realizar a augmentação de modelos no espaço de estados e esta é apenas uma delas. No entanto, há um diferencial nesta forma, que se deve ao fato dela utilizar somente variáveis que de fato são função do modelo nominal. **Isto acarreta garantia de controlabilidade e observabilidade ao modelo aumentado, se o modelo nominal já tiver estas propriedades.**

Mais adiante veremos o caso de tempo discreto, no qual este método é desenvolvido de maneira equivalente para uma variação da sequência de controle, isto é, $\Delta u(k) = u(k) - u(k-1)$.

Deve-se ter muito cuidado com os métodos de augmentação apresentados nos livros clássicos de controle, pois alguns deles acarretam problemas de observabilidade, inviabilizando projetos adequados de *compensadores dinâmicos*, quando o uso do observador de estados é obrigatório. Isto tem sido um sério problema em alguns trabalhos de pesquisa recentes que pude avaliar na área dos MPCs (Model-based Predictive Controllers), no qual o mau condicionamento dos modelos de estado acabam por inviabilizar os sistemas preditores essenciais a esses métodos de projeto de controladores.

O cenário descrito é algo que escapa dos engenheiros e alunos de pós-graduação, muitas vezes, por causa de uma formação de graduação com base em bibliografia que supõe sempre que os estados estão disponíveis. Isto raramente é verdade e portanto, iremos acrescentar no Exemplo 4.1 mostrado a seguir, o estudo dos sistemas estimadores ou observadores de estados com modelos aumentados pela inclusão de integrador e que absorvem as características de convergência da saída estimada à saída medida com *offset* nulo em estado estacionário para quando o sistema observado entra em equilíbrio.

Exemplo 4.1 *Para o sistema subamortecido mostrado seguir, projete um observador de estados, com offset nulo do erro de estimação, que seja capaz de capturar as características dinâmicas da banda passante*

do sistema, mas que rejeite ruído de alta frequência, estabelecendo um problema de alocação dos autovalores de $(A - LC)$, tal que a frequência de crossover seja de no máximo 10 rad/s:

$$\begin{aligned}
 K_p &= 0.9, \quad \omega_n = 2 \text{ rad/s}, \quad \zeta = 0.2 \\
 G(s) &= \frac{Y(s)}{U(s)} = \frac{K_p \omega_n^2}{s^2 + 2\zeta \omega_n s + \omega_n^2} \\
 \ddot{y} + 2\zeta \omega_n \dot{y} + \omega_n^2 y &= K_p \omega_n^2 u \\
 x_1 &= y, \quad x_2 = \dot{y} \\
 \dot{x}_2 &= -\omega_n^2 x_1 - 2\zeta \omega_n x_2 + K_p \omega_n^2 u
 \end{aligned} \tag{4.21}$$

$$\begin{aligned}
 \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ -\omega_n^2 & -2\zeta \omega_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ K_p \omega_n^2 \end{bmatrix} u \\
 y &= \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}
 \end{aligned} \tag{4.22}$$

Teste de observabilidade do modelo nominal de projeto:

$$O_b = \begin{bmatrix} C \\ CA \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{4.23}$$

$$\det(O_b) \neq 0 \tag{4.24}$$

Portanto, o modelo nominal é observável.

Augmentação do modelo pela adição de integrador para garantir offset nulo em relação ao erro de estimação da saída medida:

$$\begin{aligned}
 \dot{x}_a &= \begin{bmatrix} 0 & C \\ \mathbf{0} & A \end{bmatrix} x_a + \begin{bmatrix} 0 \\ B \end{bmatrix} u_a \\
 y_a &= \begin{bmatrix} C & 0 \end{bmatrix} x_a
 \end{aligned} \tag{4.25}$$

Esta forma de augmentação com integrador garante que se o modelo nominal é observável, o modelo aumentado também será. O diagrama de Bode do modelo aumentado é fornecido na Fig. 33.

Utilizando o valor da parte real dos pólos do sistema nominal como referência para a escolha da alocação dos pólos do sistema observador, se estabeleceu uma relação 10 vezes mais veloz, isto é, o modelo nominal possui par complexo conjugado com parte real em -0.4 , enquanto a alocação do observador foi feita para os três autovalores em

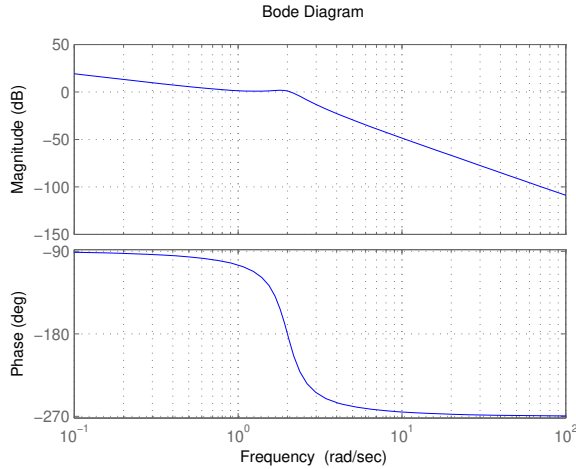


Figura 33 – Diagrama de Bode do sistema aumentado com integrador. A região em torno de 2rad/s mantém o pico de ressonância do modelo nominal e define a banda de interesse de observação.

$-4, -4, -4$.

A solução pelo método de Ackermann, de Observador, forneceu o seguinte resultado:

$$L^T = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} ({}^a O_b^T)^{-1} \Delta(A_a^T) \quad (4.26)$$

$$L^T = \begin{bmatrix} 11,2 & 35,04 & -8,832 \end{bmatrix} \quad (4.27)$$

O teste de convergência é feito com base no sistema observador em malha fechada,

$$\begin{aligned} \frac{d}{dt} \bar{x}_a &= (A_a - LC_a) \bar{x}_a + Ly \\ \bar{y} &= C_a \bar{x}_a \end{aligned} \quad (4.28)$$

em que a partir da saída medida se testa um determinado tipo de entrada conhecida. Note, que testa-se a partir da saída medida $y(t)$ e isto é uma simulação apenas, enquanto que na prática, este é o sinal que vem da saída da planta, no qual o estimador/observador projetado tentará seguir com $\bar{y}(t)$. O resultado para o teste de convergência ao

degrau é mostrado na Fig. 34.

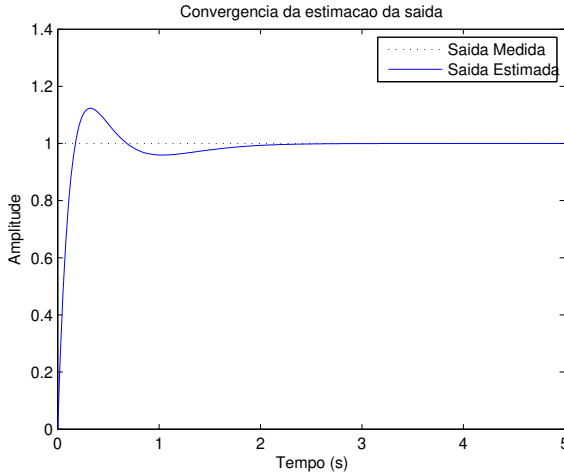


Figura 34 – Análise de convergência do estimador pelo teste de resposta ao degrau, no qual o degrau é simulado como a saída medida $y(t)$, enquanto que $\bar{y}(t)$ segue $y(t)$ com base na dinâmica do observador projetado.

Além do teste de convergência, faz-se o teste de robustez com base na resposta em frequência da malha direta do estimador, pelo teste em malha aberta com o ganho L utilizando-se

$$\begin{aligned}\frac{d}{dt}\bar{x}_a &= A_a\bar{x}_a + Ly \\ \bar{y} &= C_a\bar{x}_a\end{aligned}$$

Este resultado é apresentado na Fig. 35.

Os testes finais relativos ao Observador deste exemplo vão comparar os dados do sistema nominal com os do sistema observador. Devido à complexidade dos testes, por envolverem perturbações de carga e ruído no domínio de tempo contínuo, estes testes foram realizados com o Simulink para simplificar o trabalho. O diagrama de simulação implementado é mostrado na Fig. 36 e o código m-file de apoio, com este projeto no Matlab, é apresentado na Tabela 17.

Os principais resultados dos testes com o observador de estados projetado são mostrados nas figuras de 37 a 42, respectivamente para os casos de estimação de saída e estados sob perturbação de carga (figs.

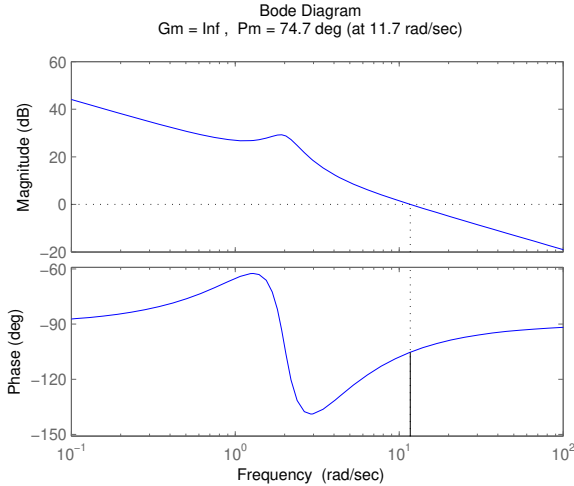


Figura 35 – Diagrama de Bode de malha direta do sistema observador, contendo informações valiosas sobre as margens de ganho e de fase para avaliação da robustez do sistema projetado. Destaque à atenuação de alta frequência, para eliminar ruídos de medição.

37, 38); *estimação de saída e estados sob perturbação de alta frequência* (figs. 39, 40); *estimação de saída e estados sob condições iniciais não nulas* (figs. 41, 42).

Com o conhecimento adquirido sobre controladores por realimentação de estados e também sobre como estimar as variáveis de estado, pode-se então passar ao caso completo que propicia o projeto de compensadores dinâmicos lineares.

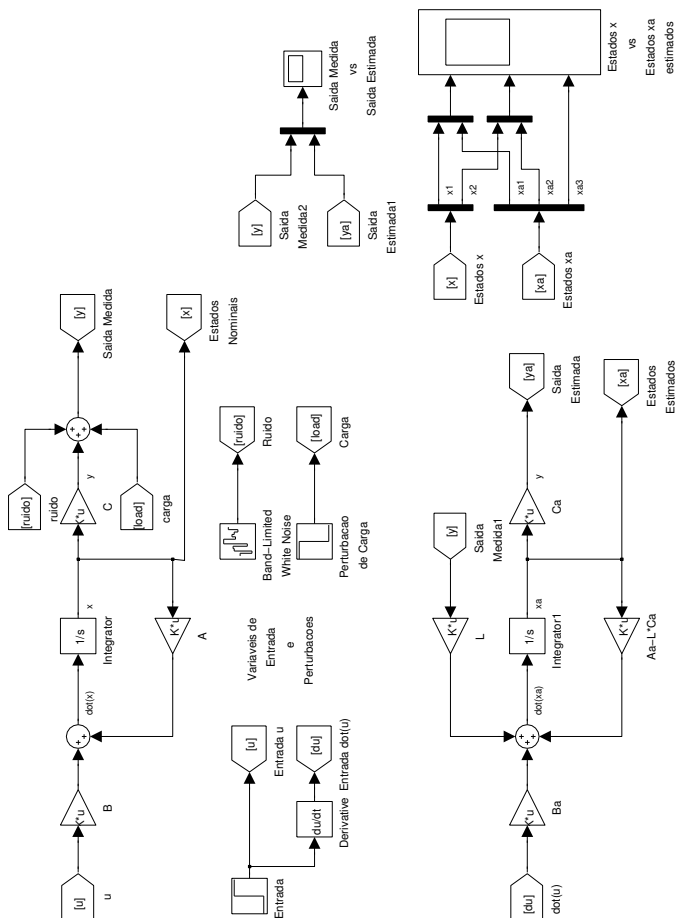


Figura 36 – Diagrama de simulação no Simulink para os testes com o Observador de Estados do Exemplo 4.1, para os casos de estimação de saída, de estados, rejeição de perturbação de baixa e alta frequência.

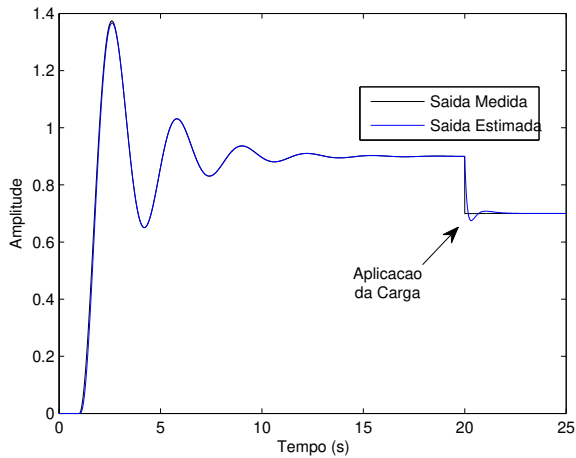


Figura 37 – Estimador de $\bar{y}(t)$ robusto à perturbação de carga, com offset nulo.

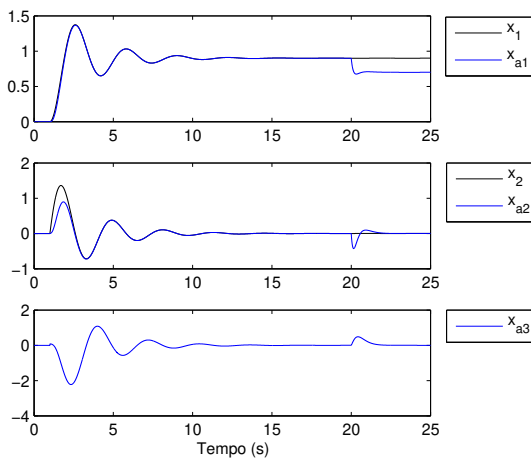


Figura 38 – Estados estimados convergindo aos estados reais (exceto no caso da perturbação de carga, já que esta foi simulada diretamente sobre a saída, propositalmente, por motivo de simplificação da parte matemática da revisão teórica).

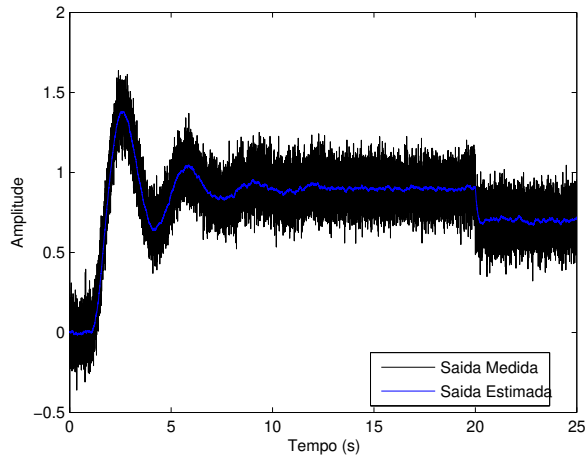


Figura 39 – Estimador robusto às perturbações a partir da frequência de 10rad/s.

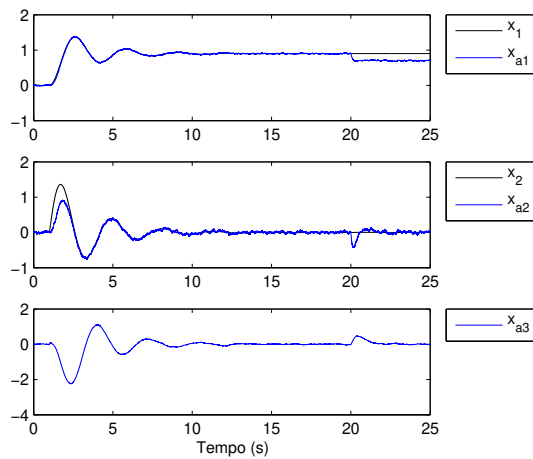


Figura 40 – Estimador robusto às perturbações a partir da frequência de 10rad/s, possibilitando a realimentação de variáveis de estado filtradas para uma aplicação em controle.

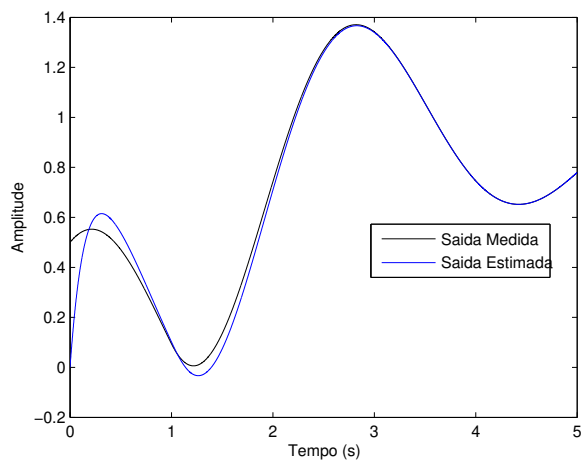


Figura 41 – Estimação da saída mesmo sob condições iniciais não nulas.

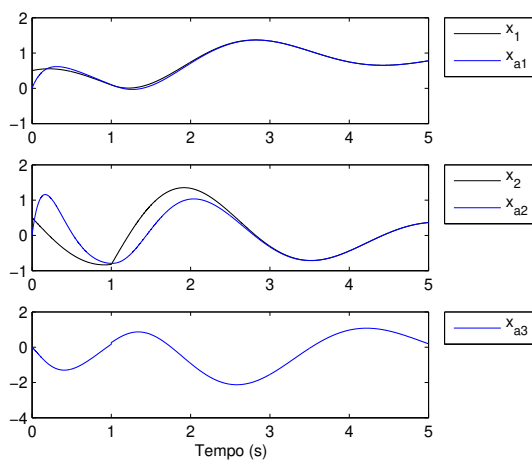


Figura 42 – Estimação dos estados mesmo sob condições iniciais não nulas de x_1 e x_2 .

Tabela 17 – Código em MATLAB referente ao projeto de observador de estados do Exemplo 4.1.

```
% Observador de estados com base em modelo tipo-0
% aumentado por integrador
clear all; close all; clc;

% Sistema linear contínuo descrito por FT
Kp=0.9; zeta=0.2; wn=2;
Gs = tf(Kp*wn^2,[1 2*zeta*wn wn^2]);

% Realizacao em state space com x1=y e x2=dot(y)
A = [0 1; -wn^2 -2*zeta*wn]; B=[0;Kp*wn^2]; C=[1 0]; D=0;
sys=ss(A,B,C,D);

% Modelo de estados aumentado com integrador
Aa = [0 C; [0;0] A]; Ba = [0;B]; Ca=[C 0]; Da=0;
sysa=ss(Aa,Ba,Ca,Da);

% Analise de observabilidade dos modelos nominal e aumentado
disp(['Rank de Co do modelo nominal: ' num2str(rank(observ(A,C)))]);
disp(['Rank de Co do modelo aumentado: ' num2str(rank(observ(Aa,Ca)))]);
disp('Modelos observaveis');

% Analise em M.A. para definir a estrategia do observador
figure(1);
bode(sysa); grid; % resposta em freq, em M.A. sist. aumentado.
% Obs: como o sistema e subamortecido e tem freq. wn=2rad/s,
% esta sera uma freq que deve ser observada no diagrama de
% malha direta do observador. Ou seja, os polos do observador
% deverao operar em frequencia mais rapida que 2rad/s.

% Projeto do Observador de Estados
P = [-4 -4 -4]; % Autovalores desejados de M.F.
L = acker(Aa',Ca',P)' % Dualidade com o caso de controle
eig(Aa-L*Ca)

% Teste de convergencia em malha fechada
Acl = Aa-L*Ca; Bcl = L; Ccl = Ca; Dcl = 0;
syscl=ss(Acl, Bcl, Ccl, Dcl);
[yest,t,xest]=step(syscl,5); % valores estimados/observados
nit=length(t); % numero de iteracoes da simulacao step
y=ones(1,nit); % degrau como valor da saida medida para o teste
figure(2);
plot(t,y,'k',t,yest,'b');
legend('Saida Medida','Saida Estimada');
title('Convergencia da estimacao da saida');
xlabel('Amplitude'); xlabel('Tempo (s)');

% Teste em frequencia da malha direta em M.A.
Amd=Aa; Bmd=L; Cmd=Ca; Dmd=Da;
sysmd=ss(Amd,Bmd,Cmd,Dmd);
figure(3);
margin(sysmd);
```

4.4 COMPENSADOR DINÂMICO E O PRINCÍPIO DA SEPARAÇÃO

A estrutura de um compensador dinâmico linear, no caso regulatório, é apresentada na Fig. 43. Esta é formada pela junção de um sistema de observação de estados com um regulador por realimentação de estados.

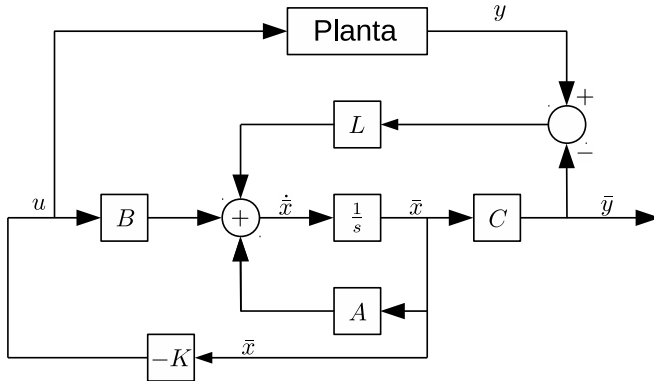


Figura 43 – Diagrama de blocos de um Compensador Dinâmico: realimentação de estados estimados.

A junção entre regulador e observador se dá com base no **Princípio da Separação**, que estabelece que os projetos do regulador e do observador podem ser feitos separadamente, desde que ambos sejam assintoticamente estáveis. De todo o modo, em termos práticos, projetos de reguladores e observadores com margens de estabilidade robusta devem ser priorizados e com alguma sinergia, para que o **Princípio da Separação** seja factível.

A forma sinérgica mencionada diz respeito, principalmente, às regiões de frequência em que cada parte do compensador vai operar. Como exemplo, não faz sentido estabelecer no projeto do observador uma frequência de corte inferior à frequência de corte do projeto do regulador, haja vista que o controlador estará com uma banda passante além da capacidade do sistema de estimação. Nesse sentido, deve-se estabelecer regiões operacionais compatíveis para se obter um projeto de observador que não destrua o **ganho de malha** do regulador e vice-versa.

Do ponto de vista de projeto, pelo Princípio da Separação, pode-

se afirmar que já temos todas as ferramentas para projetar um compensador dinâmico. Portanto, vamos acrescentar apenas o equacionamento para o caso de regulação e de seguimento de trajetória, para facilitar a análise computacional desses sistemas.

4.4.1 Compensador dinâmico de regulação

Assume-se que o sistema nominal de projeto tem a forma

$$\begin{aligned}\dot{x} &= Ax + Bu \\ y &= Cx\end{aligned}\tag{4.29}$$

e que este é do tipo-0, controlável e observável.

O sistema observador de estados é definido pelas seguintes equações:

$$\begin{aligned}\frac{d}{dt}\bar{x} &= (A - LC)\bar{x} + Bu + Ly \\ \bar{y} &= C\bar{x}\end{aligned}\tag{4.30}$$

A regulação por realimentação total de estados estimados modifica a lei de controle para a forma

$$u = -K\bar{x}.\tag{4.31}$$

O problema do compensador pode ser estudado com base em um sistema aumentado pela junção do regulador com o observador, que em termos de variáveis de estado, pode ser compreendido pelo seguinte vetor de estados de compensador:

$$x_c = \begin{bmatrix} x \\ \bar{x} \end{bmatrix}\tag{4.32}$$

Substituindo a lei de controle em (4.31) tanto no modelo nominal como no sistema observador, também pela substituição da saída medida $y = Cx$ na equação do estimador, obtém-se

$$\begin{aligned}\frac{d}{dt}x &= Ax + B(-K\bar{x}) \\ \frac{d}{dt}\bar{x} &= (A - LC)\bar{x} + B(-K\bar{x}) + L(Cx)\end{aligned}\tag{4.33}$$

Estas equações podem ser escritas com base no novo vetor de estados do compensador dinâmico, $x_c^T = [x \quad \bar{x}]$, tal que

$$\begin{aligned} \dot{x}_c &= \begin{bmatrix} A & -BK \\ LC & A - LC - BK \end{bmatrix} x_c \\ y_c &= \begin{bmatrix} y \\ \bar{y} \end{bmatrix} = \begin{bmatrix} C & 0 \\ 0 & C \end{bmatrix} x_c \end{aligned} \quad (4.34)$$

Estas equações representam o compensador dinâmico em malha fechada com o modelo nominal para o caso regulatório.

4.4.2 Compensador dinâmico no caso servo

Para o caso servo com *offset* nulo para o seguimento de referência do tipo degrau, a lei de controle do sistema aumentado é

$$u_a = \dot{u} = -K\bar{x}_a + k_1 y_r \quad (4.35)$$

no qual o ganho k_1 é o primeiro elemento do vetor de ganhos K e possui valor escalar.

De maneira análoga ao caso regulatório, faz-se a substituição da lei de controle servo nas equações do modelo aumentado e do observador de estados, chegando-se assim às seguintes equações:

$$\begin{aligned} \frac{d}{dt} x_a &= A_a x_a + B_a (-K\bar{x}_a + k_1 y_r) \\ \frac{d}{dt} \bar{x}_a &= (A_a - LC_a) \bar{x}_a + B_a (-K\bar{x}_a + k_1 y_r) + L(C_a x_a) \end{aligned} \quad (4.36)$$

Estas equações podem ser reescritas em função de um vetor de estados definido para o caso servo,

$$x_s = \begin{bmatrix} x_a \\ \bar{x}_a \end{bmatrix} \quad (4.37)$$

O sistema completo assume a seguinte estrutura:

$$\begin{aligned} \dot{x}_s &= \begin{bmatrix} A_a & -B_a K \\ LC_a & A_a - LC_a - B_a K \end{bmatrix} x_s + \begin{bmatrix} k_1 B_a \\ k_1 B_a \end{bmatrix} y_r \\ y_s &= \begin{bmatrix} y_a \\ \bar{y}_a \end{bmatrix} = \begin{bmatrix} C_a & 0 \\ 0 & C_a \end{bmatrix} x_s \end{aligned} \quad (4.38)$$

4.5 COMPENSADORES DINÂMICOS DIGITAIS

Para o projeto de compensadores dinâmicos digitais iremos considerar o equivalente discreto via método ZOH de uma realização de estados, tal que as equações de estado e de saída discretas sejam descritas por

$$\begin{aligned}x(k+1) &= A_d x(k) + B_d u(k) \\ y(k) &= C_d x(k)\end{aligned}\tag{4.39}$$

em que A_d , B_d , $C_d = C$ são as matrizes discretas de sistema, de entrada e de saída, respectivamente.

4.5.1 Regulador digital via realimentação de estado estimado

A lei de controle do regulador digital via realimentação de estado estimado é definida por

$$u(k) = -K\bar{x}(k)\tag{4.40}$$

em que $\bar{x}(k)$ é o vetor de estado estimado pelo sistema observador digital cujas equações de estado e saída estimadas são descritas por

$$\begin{aligned}\bar{x}(k+1) &= (A_d - LC_d)\bar{x}(k) + B_d u(k) + Ly(k) \\ \bar{y}(k) &= C_d \bar{x}(k)\end{aligned}\tag{4.41}$$

É importante destacar que estruturalmente não há diferença entre o regulador contínuo e o digital. Este é um forte resultado da inserção da Álgebra Linear na Teoria de Controle, permitindo que os métodos para o cálculo dos ganhos do regulador e do observador, K e L , assim como os métodos de análise de controlabilidade e observabilidade, sejam os mesmos para os casos contínuo e discreto. Portanto, a equação de Ackermann para a solução do problema de alocação é válida para o caso discreto, mudando, somente, a escolha do posicionamento dos autovalores que no caso discreto devem estar contidos no círculo unitário para que a solução seja assintoticamente estável.

4.5.2 Augmentação de sistemas discretos no espaço de estado

Para realizar a inclusão de ação integral discreta (ação incremental) aos sistemas de controle digitais baseados em modelo, se faz necessário aumentar o modelo de projeto

$$\begin{aligned} x(k+1) &= A_d x(k) + B_d u(k) \\ y(k) &= C_d x(k) \end{aligned} \quad (4.42)$$

Um alternativa é trabalhar com a variação discreta do sinal de controle, isto é,

$$\Delta u(k) = u(k) - u(k-1), \quad (4.43)$$

tal que o sinal de controle que é de fato aplicado na entrada do processo é

$$u(k) = u(k-1) + \Delta u(k). \quad (4.44)$$

Sendo assim, para se obter o modelo aumentado pelo operador de diferença discreto, $\Delta = 1 - z^{-1}$, as equações de estado e saída se modificam para

$$\begin{aligned} \Delta x(k+1) &= A_d \Delta x(k) + B_d \Delta u(k) \\ \Delta y(k+1) &= C_d \Delta x(k+1) \end{aligned} \quad (4.45)$$

e um novo vetor de estado aumentado é definido:

$$x_a(k) = \begin{bmatrix} y(k) \\ \Delta x(k) \end{bmatrix}. \quad (4.46)$$

Como $\Delta y(k+1) = y(k+1) - y(k)$, então as equações de saída e de variação discreta de estado podem ser escritas como

$$\begin{aligned} y(k+1) &= y(k) + C_d A_d \Delta x(k) + C_d B_d \Delta u(k) \\ \Delta x(k+1) &= A_d \Delta x(k) + B_d \Delta u(k) \end{aligned} \quad (4.47)$$

estando ambas sujeitas à entrada aumentada $\Delta u(k) = u_a(k)$. Esse sistema aumentado pode então ser reescrito com base na seguinte des-

crição no espaço de estado:

$$\underbrace{\begin{bmatrix} y(k+1) \\ \Delta x(k+1) \end{bmatrix}}_{x_a(k+1)} = \underbrace{\begin{bmatrix} 1 & C_d A_d \\ \mathbf{0} & A_d \end{bmatrix}}_{A_a} \underbrace{\begin{bmatrix} y(k) \\ \Delta x(k) \end{bmatrix}}_{x_a(k)} + \underbrace{\begin{bmatrix} C_d B_d \\ B_d \end{bmatrix}}_{B_a} \underbrace{\Delta u(k)}_{u_a(k)} \quad (4.48)$$

$$y_a(k) = \underbrace{\begin{bmatrix} 1 & \mathbf{0} \end{bmatrix}}_{C_a} \underbrace{\begin{bmatrix} y(k) \\ \Delta x(k) \end{bmatrix}}_{x_a(k)} \quad (4.49)$$

4.5.3 Compensador dinâmico digital no caso servo

Com base no modelo discreto aumentado descrito em (4.48) e (4.49) é possível projetar observador e controlador com base em realimentação de estado observado que garantam *offset* nulo para o caso servo ou de seguimento de referência do tipo degrau. Para tal, se deve lembrar que a lei de controle que será projetada é definida por

$$\Delta u(k) = k_1 y_r(k) - K \bar{x}_a(k), \quad (4.50)$$

em que $\bar{x}_a(k)$ é o vetor de estado aumentado estimado discreto e que o sinal aplicado à entrada do processo controlado será na forma incremental de comando,

$$u(k) = u(k-1) + \Delta u(k). \quad (4.51)$$

O sistema observador digital de estado tem a seguinte estrutura:

$$\begin{aligned} \bar{x}_a(k+1) &= A_a \bar{x}_a(k) + B_a \Delta u(k) + L [y(k) - \bar{y}(k)] \\ \bar{y}(k) &= \bar{y}_a(k) = C_a \bar{x}_a(k) \end{aligned} \quad (4.52)$$

Ao se substituir a lei de controle nas equações (4.48) e (4.52) obtém-se as equações do compensador dinâmico digital em malha fechada, definidas com base no vetor de estado de compensador $x_c^T(k) =$

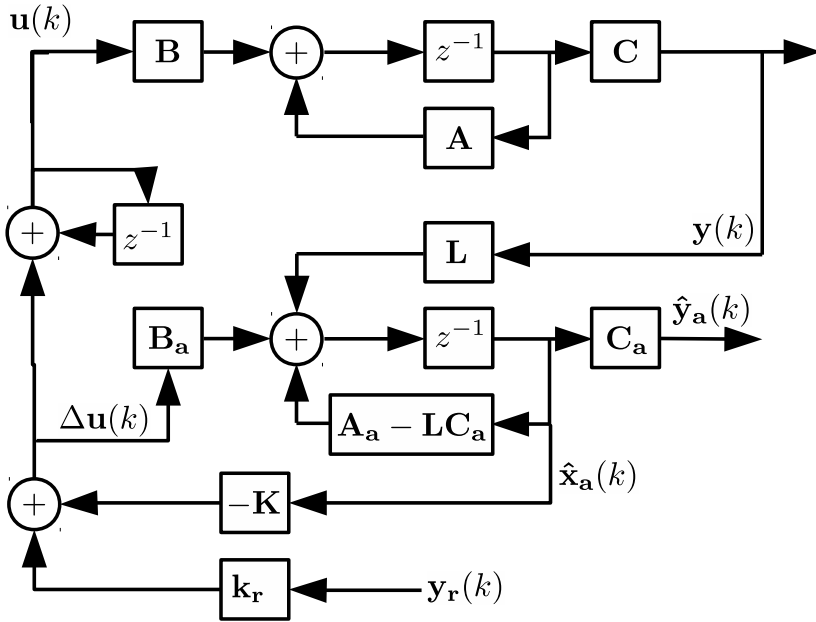


Figura 44 – Compensador dinâmico digital via realimentação de estado estimado por observador. Controlador e observador de estado projetados com base em modelo aumentado com integrador discreto.

$\begin{bmatrix} x_a(k) & \bar{x}_a(k) \end{bmatrix}$, dadas por:

$$\begin{bmatrix} x_a(k+1) \\ \bar{x}_a(k+1) \end{bmatrix} = \begin{bmatrix} A_a & -B_a K \\ LC_a & A_a - LC_a - B_a K \end{bmatrix} \begin{bmatrix} x_a(k) \\ \bar{x}_a(k) \end{bmatrix} + \begin{bmatrix} B_a k_1 \\ B_a k_1 \end{bmatrix} y_r(k) \quad (4.53)$$

$$\begin{bmatrix} y_a(k) \\ \bar{y}_a(k) \end{bmatrix} = \begin{bmatrix} C_a & \mathbf{0} \\ \mathbf{0} & C_a \end{bmatrix} \begin{bmatrix} x_a(k) \\ \bar{x}_a(k) \end{bmatrix} \quad (4.54)$$

5 CAPÍTULO 5

Preliminares da Teoria de Controle Moderno. Regulador Linear Quadrático (LQR). Filtro de Kalman. Regulador Linear Quadrático Gaussiano (LQG). LQG com Recuperação do Ganho de Malha (LQG/LTR).

5.1 INTRODUÇÃO

Neste capítulo se dará continuação ao estudo da Teoria de Sistemas Lineares com foco na Teoria de Controle Moderno, em que as descrições de sistemas dinâmicos no espaço de estado são necessárias às sínteses de controladores e de filtros estimadores. No entanto, nos capítulos anteriores foram negligenciados alguns estudos preliminares sobre os sistemas dinâmicos, como os conceitos de *linearidade* e o procedimento de *linearização local* em torno de um *ponto de equilíbrio*.

Partiremos de exemplos elementares que serão estendidos para conceitos de linearização de sistemas descritos no espaço de estado, e também ao uso de métodos computacionais para a realização das etapas de trimagem (*trim*) por simulação na busca por uma solução ou ponto de equilíbrio aproximado para posterior linearização assistida por computador, por exemplo, usando Matlab/Simulink com as funções *trim* e *linmod*, que apesar de serem tecnologias proprietárias, resolvem problemas de linearização de sistemas dinâmicos cuja complexidade impõe dificuldade de solução não-computacional. Esses conceitos e técnicas de linearização são importantes para que se adquira um melhor entendimento de como as tecnologias de sistemas de controle linear podem ser adequadamente aplicadas aos sistemas não lineares reais.

Além da inserção de novos conceitos preliminares, estenderemos nosso estudo sobre as tecnologias de controle moderno, introduzindo métodos de otimização de sistemas para projetar controladores capazes de minimizar o esforço de controle e a trajetória das variáveis de estado, e também projetar estimadores ótimos de estado capazes de minimizar o erro de estimação enquanto agregam características de filtragem ótima de ruídos nos dados provenientes dos sensores. Essas tecnologias são conhecidas pelos seguintes nomes: Regulador Linear (Ótimo) Quadrático e Estimador ou Filtro de Kalman.

5.2 PRELIMINARES SOBRE SISTEMAS DINÂMICOS LINEARES

5.2.1 Linearidade

Um sistema é dito linear quando este respeita as propriedades da aditividade e da homogeneidade. A análise conjunta dessas propriedades caracteriza o Princípio da Superposição, que estabelece a linearidade de um sistema.

Consideremos um exemplo simples com base no seguinte sistema linear mostrado a seguir:

$$y = au, \quad (5.1)$$

em que y, a, u , são: a saída, um parâmetro escalar e a entrada do sistema, respectivamente. Se aplicarmos uma entrada de valor u_1 , isto é, $u = u_1$, o valor da saída será

$$y_1 = au_1. \quad (5.2)$$

E se aplicarmos uma entrada de valor u_2 , o valor da saída será

$$y_2 = au_2. \quad (5.3)$$

Para que este sistema respeite a propriedade da aditividade, a soma $y_1 + y_2$ deve ser equivalente a quando na entrada u é aplicado o valor de entrada $u_1 + u_2$. Isto é:

$$y = a(u_1 + u_2) = au_1 + au_2 = y_1 + y_2. \quad (5.4)$$

Logo, respeita a propriedade da aditividade.

Consideremos agora o teste de homogeneidade sobre o sistema em (5.1): ao aplicarmos o valor de entrada $u = \alpha u_1$, sendo α um escalar, o sistema respeitará a propriedade da homogeneidade se ao multiplicarmos α por y_1 o resultado for igual a

$$y = a(\alpha u_1), \quad (5.5)$$

que o é, sendo a propriedade da homogeneidade garantida e o Princípio da Superposição é atendido pelo sistema em (5.1).

Apesar do sistema em (5.1) ser um sistema estático, isto é, sem memória e portanto não sendo um sistema dinâmico, ele pode possibilitar o entendimento de um dos conceitos fundamentais de sistemas

dinâmicos lineares, que é o conceito de que quando o sistema é linear e na sua entrada aplica-se um valor nulo, o valor da saída também é nulo.

Um contra-exemplo seria também a utilização de uma equação de reta, porém com um parâmetro constante b como se mostra a seguir:

$$y = au + b. \quad (5.6)$$

Essa reta que não passa pela origem não é linear, pois: aplicando-se $u = u_1$,

$$y_1 = au_1 + b. \quad (5.7)$$

Aplicando-se $u = u_2$,

$$y_2 = au_2 + b. \quad (5.8)$$

Para que o sistema seja linear, $y_1 + y_2$ deve ser igual a

$$y_1 + y_2 = (au_1 + b) + (au_2 + b) = a(u_1 + u_2) + 2b \quad (5.9)$$

para quando na entrada for aplicado $u = u_1 + u_2$. No entanto, se aplicarmos $u = u_1 + u_2$ ao sistema em (5.6) obtemos

$$y = a(u_1 + u_2) + b \neq a(u_1 + u_2) + 2b. \quad (5.10)$$

Como esse sistema falha no teste de aditividade (e falha também no de homogeneidade, apesar de não mostrado), então esse sistema é não linear.

Com este último exemplo fica claro que o parâmetro escalar $b \neq 0$ é o responsável pela não linearidade do sistema, pois faz com que a saída do sistema não passe pela origem para quando a entrada é nula. Desse modo, guardemos essa observação para correlacioná-la ao exemplo de linearização local que estudaremos na seção seguinte, pois nela veremos que o procedimento de linearização local opera deslocando a origem do modelo linearizado para o ponto de equilíbrio ou de solução do sistema.

5.2.2 Linearização local

Uma linearização local de um sistema dinâmico não-linear $y(u)$ pode ser obtida com base em uma solução conhecida $y(u_{eq})$ desse sistema e a sua aplicação em uma expansão em Série de Taylor para obter

uma representação aproximada $L(u)$, dada por:

$$L(u) = y(u_{eq}) + y'(u_{eq}) \frac{(u - u_{eq})^1}{1!} + \dots + y^{(n)}(u_{eq}) \frac{(u - u_{eq})^n}{n!}. \quad (5.11)$$

Normalmente trunca-se essa série em uma aproximação de primeira ordem, tal que

$$L(u) = y(u_{eq}) + y'(u_{eq}) (u - u_{eq}), \quad (5.12)$$

sendo $y'(u)$ a derivada da função $y(u)$.

Consideremos a seguir, novamente, um sistema mais simples para ilustrar a linearização local:

$$y(u) = \sqrt{u}. \quad (5.13)$$

Este sistema não respeita o Princípio da Superposição, sendo então um sistema não linear. Uma aproximação para ele via Série de Taylor truncada em primeira ordem é dada por

$$L(u) = \sqrt{u_{eq}} + \left(\frac{1}{2\sqrt{u_{eq}}} \right) (u - u_{eq}), \quad (5.14)$$

em que a parte linear $L_{lin}(\delta u)$ desse sistema é descrita por

$$L_{lin}(\delta u) = \left(\frac{1}{2\sqrt{u_{eq}}} \right) \delta u, \quad \delta u = (u - u_{eq}). \quad (5.15)$$

Utilizando-se o código para Matlab ou Scilab mostrado na Tabela 18 obtém-se a simulação desse exemplo para o sistema não-linear, para a sua aproximação por Série de Taylor truncada e também apenas para a parcela linear, em que esses resultados da simulação são mostrados na Figura 45.

Neste exemplo da Figura 45 considerou-se a linearização no local em que se conhecia uma solução para o sistema, no valor $u = u_{eq} = 4$. Veja na figura que a aproximação do sistema tem validade apenas na vizinhança do local $u_{eq} = 4$ e que o modelo aproximado não passa pela origem, pois assim como no modelo de reta considerado na seção anterior, há um parâmetro que torna a aproximação não-linear, que é o termo $\sqrt{u_{eq}}$ na equação (5.14). Nela há também uma entrada do tipo variação em torno do local de linearização, isto é: o termo $(u - u_{eq})$. Já o modelo linearizado mostrado em (5.14), passa pela origem e reproduz

o sistema aproximado não-linear mas deslocado devido a ausência do termo $\sqrt{u_{eq}}$. Desse modo, guardemos na memória essas observações, pois veremos mais adiante que os sistemas de controle linear que projetamos demandam o acréscimo desses termos negligenciados no modelo linear para quando o controlador é aplicado na entrada do sistema não-linear real.

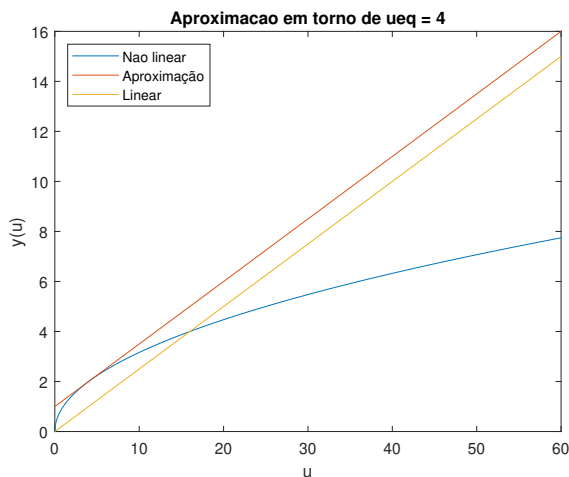


Figura 45 – Exemplo de linearização local do sistema $y(u) = \sqrt{u}$. Local $u = u_{eq} = 4$, cuja solução é conhecida, foi utilizado neste exemplo e é possível notar que somente na vizinhança desse local a aproximação por Série de Taylor truncada e o modelo linear são capazes de representar o sistema não-linear.

Tabela 18 – Código em Matlab para simular o sistema não linear $y(u) = \sqrt{u}$ e sua aproximação $L(u)$.

```
% Exemplo: linearizacao local
% Obter uma aproximacao de y(u) = sqrt(u)
clear all; close all; clc;

% Considerando a funcao nao linear
u = 0:0.01:60;
N = length(u); % Numero de iteracoes pra montar o vetor
                % y(k), k=1,2,...,N, equivalente a y(u).

for k=1:N
    y(k) = sqrt(u(k));
end
plot(u,y); ylabel('y(u)'); xlabel('u'); hold;

% Aproximacao linear: L(u) = f(ueq) +slope*(u - ueq)
% Selecao do ponto em u onde se quer linearizar a funcao
ueq = 4; % vamos obter uma aproximacao linear L(u)
        % em torno de ueq.

% Apoio ao calculo diferencial usando Matlab:
% via toolbox symbolic e funcao diff:
syms usym; % gera uma variavel simbolica usym para determinar a
            % derivada da funcao y(usym) = sqrt(usym)

disp('Inclinacao da reta');
slope = diff( sqrt(usym) )
pretty(slope); % apresenta de uma forma legivel

        % Resultado do slope = 1/( 2*sqrt(ueq) )

for k=1:N
    % L(u) = f(ueq) +slope*(ueq -x)
    L(k) = sqrt(ueq) + ( 1/(2*sqrt(ueq)) )*( u(k) - ueq );
end

plot(u,L);
legend('Nao linear','Aproximação','Location','NorthWest');
title(['Aproximacao em torno de ueq = ' num2str(ueq)]);
```

5.2.3 Linearização local no espaço de estados

Um sistema dinâmico não-linear contínuo pode ser descrito pelas seguintes equações diferenciais (HESPANHA, 2018):

$$\begin{aligned}\dot{x} &= f(x, u), \\ y &= g(x, u),\end{aligned}\tag{5.16}$$

sendo $x \in \mathbb{R}^n$, $u \in \mathbb{R}^{n_u}$, $y \in \mathbb{R}^{n_y}$, em que n é o número de variáveis de estado definidas no vetor x , n_u é o número de entradas no vetor u e n_y

é o número de saídas no vetor y .

A linearização local de um sistema não-linear, como o que é mostrado em (5.16), pode ser realizada com base em um ponto de equilíbrio definido pelo par $(x_{eq}, u_{eq}) \in \mathbb{R}^n \times \mathbb{R}^u$. Por definição, um ponto de equilíbrio é aquele que garante que $\dot{x} = f(x_{eq}, u_{eq}) = 0$. Ou seja, quando não há variação de estado. Nesse caso diz-se que o par (x_{eq}, u_{eq}) é uma solução do sistema (5.16).

Lembremos então do exemplo de linearização local discutido na Seção 5.2.2 sobre um sistema estático $y(u) = \sqrt{u}$. Neste caso, o sistema linearizado adotava uma pequena variação de entrada $\delta u = u - u_{eq}$ em torno do local de linearização u_{eq} . O mesmo princípio é aplicado para sistemas dinâmicos mais complexos como o caso não-linear em (5.16), tal que

$$u(t) - u_{eq} = \delta u(t), \quad \forall t \geq 0, \quad (5.17)$$

sendo $\delta u(t)$ um valor próximo a u_{eq} , que seja considerável a hipótese de que não há mudança no ponto de operação do sistema dinâmico.

Com base nesta pequena variação de entrada $\delta u(t)$ entende-se que pequenas variações se propagam também nas demais equações do sistema dinâmico, conforme:

$$\begin{aligned} x(0) - x_{eq} &= \delta x_{eq}, \quad x(0) : \text{estado inicial}, \\ x(t) - x_{eq} &:= \delta x(t), \quad \forall t \geq 0, \\ y(t) - y_{eq} &:= \delta y(t), \quad \forall t \geq 0. \end{aligned} \quad (5.18)$$

Como o nosso interesse é o de obter o modelo linear baseado nas pequenas variações consideradas, reescrevemos o modelo dinâmico com base em $\delta x(t)$ e $\delta y(t)$, na tradicional forma canônica de sistemas lineares no espaço de estados, dada por:

$$\dot{\delta x}(t) = A\delta x(t) + B\delta u(t), \quad (5.19)$$

$$\delta y(t) = C\delta x(t) + D\delta u(t), \quad (5.20)$$

em que A, B, C, D , são as matrizes Jacobianas obtidas pelo procedimento de expansão em série de Taylor truncada realizado no ponto de equilíbrio ou de solução do sistema.

Começemos pela equação (5.19). A sua obtenção provém da descrição de $\dot{x} = f(x, u)$ (em que o domínio t foi omitido da solução

mostrada a seguir para simplificar a notação):

$$\dot{\delta x} = \dot{x} - \dot{x}_{eq} = f(x, u) - f(x_{eq}, u_{eq}). \quad (5.21)$$

Lembrando que o termo $\dot{x}_{eq} = f(x_{eq}, u_{eq}) = 0$, pois é solução do sistema no ponto de equilíbrio. Esse termo foi mantido propositalmente para destacar como ele é eliminado na etapa de expansão em série de Taylor truncada realizada a seguir.

Aplicando-se o procedimento de expansão por série de Taylor truncada sobre $\dot{x} = f(x, u)$ obtém-se a aproximação $L_f(x, u)$:

$$L_f(x, u) = f(x_{eq}, u_{eq}) + \frac{\partial f(x_{eq}, u_{eq})}{\partial x}(x - x_{eq}) + \frac{\partial f(x_{eq}, u_{eq})}{\partial u}(u - u_{eq}), \quad (5.22)$$

sendo $\delta x = L_f(x, u) - f(x_{eq}, u_{eq})$, $\delta x = x - x_{eq}$ e $\delta u = u - u_{eq}$. Portanto, a equação de estado linearizada é dada por:

$$\dot{\delta x} = \frac{\partial f(x_{eq}, u_{eq})}{\partial x} \delta x + \frac{\partial f(x_{eq}, u_{eq})}{\partial u} \delta u \quad (5.23)$$

Para a equação de saída do sistema não-linear faz-se os mesmos procedimentos de linearização local. Desse modo, considerando obter o modelo de pequena variação de saída, δy , tem-se:

$$\begin{aligned} \delta y &= g(x, u) - y_{eq} \\ \delta y &= L_g(x, y) - g(x_{eq}, u_{eq}) \end{aligned} \quad (5.24)$$

Aplicando-se a expansão em série de Taylor truncada sobre $g(x, u)$ obtém-se

$$L_g(x, u) = g(x_{eq}, u_{eq}) + \frac{\partial g(x_{eq}, u_{eq})}{\partial x}(x - x_{eq}) + \frac{\partial g(x_{eq}, u_{eq})}{\partial u}(u - u_{eq}) \quad (5.25)$$

e portanto a equação de saída do modelo linear é dada por:

$$\delta y = \frac{\partial g(x_{eq}, u_{eq})}{\partial x} \delta x + \frac{\partial g(x_{eq}, u_{eq})}{\partial u} \delta u. \quad (5.26)$$

Por fim, a Teoria de Sistemas Lineares comumente se utiliza da simplificação da notação do modelo linearizado, omitindo a participação

do operador δ e ficando na forma

$$\begin{aligned}\dot{x} &= Ax + Bu, \\ y &= Cx + Du,\end{aligned}\tag{5.27}$$

cujas matrizes Jacobianas são:

$$\begin{aligned}A &:= \frac{\partial f(x_{eq}, u_{eq})}{\partial x}, \quad B := \frac{\partial f(x_{eq}, u_{eq})}{\partial u}, \\ C &:= \frac{\partial g(x_{eq}, u_{eq})}{\partial x}, \quad D := \frac{\partial g(x_{eq}, u_{eq})}{\partial u}.\end{aligned}\tag{5.28}$$

É sempre importante frisar que a Álgebra Linear independe do domínio de tempo, se contínuo ou discreto. Portanto, os mesmos procedimentos aqui adotados, para a linearização de um sistema dinâmico de tempo contínuo no espaço de estado, são aplicáveis a sistemas dinâmicos de tempo discreto. No entanto, é também importante destacar que a digitalização trouxe novas tecnologias que podem proporcionar a linearização de sistemas dinâmicos discretos executando outros procedimentos assistidos por computador, como o método de estimação paramétrica dos Mínimos Quadrados (MQ) ou MQ Recursivo (MQR).

Uma das aplicações de maior pertinência do procedimento de linearização de sistemas no espaço de estados, com relação às tecnologias de controle discutidas neste capítulo, está ligada ao estimador de estado por Filtro de Kalman (FK) que se utiliza da linearização em tempo real para o reprojeto e a adaptação do ganho ótimo do estimador. Essa maneira de implementar o FK é comumente designada pelo nome de Filtro de Kalman Estendido (FKE) (LEWIS et al., 2008). No entanto, métodos equivalentes de promover a linearização em tempo real pelo uso de MQ, MQR e FK têm sido propostos, como é o caso do método *Observer/Kalman filter Identification* (OKID) (VICARIO, 2014; VICARIO et al., 2017).

Dentre os métodos assistidos por computador para promover a linearização de sistemas em tempo real, utilizaremos a solução proprietária fornecida no Matlab/Simulink ainda neste capítulo, e veremos a solução baseada no método OKID no Capítulo 7.

Em relação ao método de linearização local que acabamos de estudar, veremos a seguir dois exemplos. O primeiro deles, o Exemplo 5.1, o método estudado pode ser aplicado diretamente sem qualquer cuidado adicional do ponto de vista da análise da condição de equilíbrio do sistema a ser linearizado, por se tratar de um sistema mais simples e compacto. Já o segundo problema, o Exemplo 5.2, é do em-

blemático pêndulo invertido sobre o carro, o qual sem que seja realizada uma cautelosa análise das condições operacionais e de equilíbrio, seria extremamente difícil a solução manual usando o método aqui estudado, como veremos mais adiante.

Exemplo 5.1 (Baseado na videoaula da Dra. Katherine A. Kim, da National Taiwan University, sobre linearização no espaço de estados disponível no YouTube: [youtube.com/v/41HYQG8BjPA](https://www.youtube.com/v/41HYQG8BjPA)) Para o sistema não-linear mostrado a seguir, em que $u(t)$ é o sinal de entrada e $y(t)$ é o sinal de saída, resolva: (a) Obtenha uma realização de estado não-linear. (b) Determine um ponto de equilíbrio (ou solução) do sistema. (c) Obtenha um modelo linear nesse ponto de equilíbrio.

$$\ddot{y}(t) = 2y(t) + y^2(t)\dot{y}(t) + \dot{y}(t)u(t) - 8 \quad (5.29)$$

Resposta (a): definindo-se $x_1 = y$ e $x_2 = \dot{y}$,

$$\begin{aligned} \dot{x} = f(x, u) &= \begin{cases} f_1(x, u) = \dot{x}_1 = x_2 \\ f_2(x, u) = \dot{x}_2 = 2x_1 + x_1^2 x_2 + x_2 u - 8 \end{cases} \\ y = g(x, u) &= \{g_1(x, u) = y = x_1 \end{aligned}$$

Resposta (b): ponto de equilíbrio $\dot{x} = f(x, u) = 0$:

$$\begin{aligned} \dot{x}_1 = 0 &= x_2 \\ \dot{x}_2 = 0 &= 2x_1 + x_1^2(0) + (0)u - 8 \end{aligned}$$

sendo o ponto de equilíbrio estabelecido quando

$$x_{eq} = \begin{bmatrix} 4 & 0 \end{bmatrix}^T$$

para qualquer valor $u = u_{eq}$. Desse modo assumiu-se $u_{eq} = 0$.

Resposta (c):

$$\begin{aligned} A &:= \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{bmatrix}, \quad B := \begin{bmatrix} \frac{\partial f_1}{\partial u} \\ \frac{\partial f_2}{\partial u} \end{bmatrix} \\ C &:= \begin{bmatrix} \frac{\partial g_1}{\partial x_1} & \frac{\partial g_1}{\partial x_2} \end{bmatrix}, \quad D := \begin{bmatrix} \frac{\partial g_1}{\partial u} \end{bmatrix} \end{aligned}$$

$$A := \begin{bmatrix} 0 & 1 \\ 2 + 2x_1x_2 & x_1^2 + u \end{bmatrix}, \quad B := \begin{bmatrix} 0 \\ x_2 \end{bmatrix}$$

$$C := \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad D := [0]$$

O modelo linear dado pela solução x_{eq} e u_{eq} é:

$$\begin{aligned} \dot{x} &= \begin{bmatrix} 0 & 1 \\ 2 & 16 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \end{bmatrix} u \\ y &= \begin{bmatrix} 1 & 0 \end{bmatrix} x + [0] u \end{aligned}$$

Exemplo 5.2 (Baseado nos Tutoriais de Controle para Matlab e Simulink (CTMS) da University of Michigan sobre o Pêndulo Invertido: ctms.engin.umich.edu) Neste exemplo o foco é a linearização do clássico sistema dinâmico pêndulo invertido sobre o carro.

Na Figura 46 são apresentados os diagramas de corpo livre do pêndulo invertido sobre o carro. A obtenção do modelo do sistema a partir desses diagramas está além do escopo deste capítulo, mas como se trata de um sistema emblemático da Teoria de Controle, a seguir, o desenvolvimento desse modelo tal como mostrado na página do CTMS da Michigan University é transcrito:

$M = 0,5 \text{ kg}$: massa do carro

$m = 0,2 \text{ kg}$: massa do pêndulo

$b = 0,1 \text{ N/m/s}$: coeficiente de fricção do carro

$l = 0,3 \text{ m}$: comprimento até o centro de massa do pêndulo

$I = 0,006 \text{ kg.m}^2$: momento de inércia do pêndulo

F : força aplicada no carro (N)

x_p : posição do carro (m)

θ : ângulo do pêndulo a partir da posição vertical-para-baixo (rad)

Com base no diagrama do lado direito da Figura 46 e somando as forças que atuam na horizontal sobre o carro, tem-se a seguinte equação de movimento:

$$M\ddot{x}_p + b\dot{x}_p + N = F. \quad (5.30)$$

Somando-se também as forças atuantes na horizontal da ação do pêndulo no movimento do carro, tem-se a seguinte força de reação N :

$$N = m\ddot{x}_p + ml\ddot{\theta} \cos \theta - ml\dot{\theta}^2 \sin \theta. \quad (5.31)$$

Substituindo (5.31) em (5.30) obtém-se uma equação de movimento conjunto do pêndulo sobre o carro envolvendo as duas principais variáveis de saída do sistema, a posição do carro x_p em metros e o ângulo do pêndulo θ em radianos, e a variável de entrada ou força de atuação F em Newtons:

$$(M + m)\ddot{x}_p + b\dot{x}_p + ml\ddot{\theta} \cos \theta - ml\dot{\theta}^2 \sin \theta = F. \quad (5.32)$$

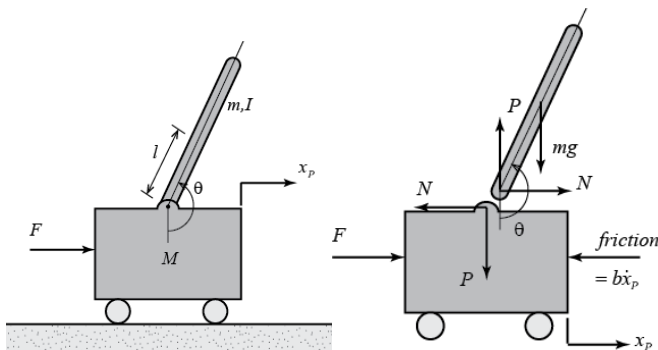


Figura 46 – À esquerda, o diagrama simplificado de um pêndulo invertido sobre o carro. À direita, a análise das forças atuantes no diagrama de corpo livre do sistema (Fonte: CTMS, University of Michigan).

A segunda equação de movimento do sistema se refere à soma das forças perpendiculares ao pêndulo com base na decomposição de P e N representados como vetores cuja origem é o ponto de pivô do pêndulo sobre o carro. Essa equação de movimento é descrita na literatura como

$$P \sin \theta + N \cos \theta = mg \sin \theta + ml\ddot{\theta} + m\ddot{x}_p \cos \theta, \quad (5.33)$$

sendo possível simplificá-la com base na seguinte relação de soma dos momentos ao redor desse ponto de pivô do pêndulo,

$$-Pl \sin \theta - Nl \cos \theta = I\ddot{\theta}. \quad (5.34)$$

Multiplicando-se (5.33) por $-l$ é possível simplificar essa equação para

a seguinte forma que estabelece a segunda equação de movimento do modelo não-linear desse sistema:

$$(I + ml^2) \ddot{\theta} = -mgl \sin \theta - ml\ddot{x}_p \cos \theta. \quad (5.35)$$

Após essa etapa de desenvolvimento do modelo não-linear do pêndulo sobre o carro, é importante dar destaque às duas equações de movimento que serão necessárias para definirmos o modelo não-linear no espaço de estado, sendo elas repetidas a seguir:

$$(M + m)\ddot{x}_p = -b\dot{x}_p - m\ddot{\theta} \cos \theta + m\dot{\theta}^2 \sin \theta + F, \quad (5.36a)$$

$$(I + ml^2) \ddot{\theta} = -mgl \sin \theta - ml\ddot{x}_p \cos \theta. \quad (5.36b)$$

Deste ponto em diante, se seguíssemos o procedimento de linearização no espaço de estado tal como aplicado no exemplo anterior, seria o momento de converter o sistema ao modelo de estado e então buscar uma solução do sistema ou ponto de equilíbrio em que a variação do vetor de estado seria nula, $\dot{x} = 0$, ou mesmo alguma outra possível solução em que o carro tivesse uma velocidade constante e não nula, por exemplo. No entanto, o tratamento matemático seria demasiadamente trabalhoso, pois as equações em (5.36) ainda requerem duas substituições, de \ddot{x}_p e $\ddot{\theta}$, entre as duas equações do modelo, deixando o problema praticamente intratável de ser resolvido à mão. De tal forma, que deste ponto em diante é comum se aplicar uma análise minuciosa das condições angulares do pêndulo com o intuito de simplificar as equações mostradas em (5.36). O interessante é que essa análise e simplificação nos levará diretamente ao modelo linearizado.

Nosso foco neste capítulo, assim como nos demais, é controle de sistemas dinâmicos. Desse modo, nosso objetivo de modelagem com o pêndulo invertido é, logicamente, garantir que o pêndulo fique para cima com ângulo $\theta = \pi$ radianos. Façamos então a análise das condições angulares com base nos diagramas da Figura 46. Assim, as seguintes considerações são propostas:

$$\theta_{eq} = \pi \text{ rad} \quad (5.37a)$$

$$\theta - \theta_{eq} = \delta\theta \quad (5.37b)$$

$$\cos(\theta_{eq} + \delta\theta) \approx -1 \quad (5.37c)$$

$$\sin(\theta_{eq} + \delta\theta) \approx -\delta\theta \quad (5.37d)$$

$$\dot{\theta}^2 = (\delta\dot{\theta})^2 \approx 0 \quad (5.37e)$$

$\delta\theta$ é então uma pequena variação em torno de $\theta_{eq} = \pi$ rad.

Aplicando ao modelo as considerações listadas em (5.37) obtém-se (mantendo-se a notação com θ ao invés de $\delta\theta$):

$$\begin{aligned}(M + m)\ddot{x}_p &= -b\dot{x}_p - ml\ddot{\theta}(-1) + ml(0)(-\theta) + F, \\ (I + ml^2)\ddot{\theta} &= -mlg(-\theta) - ml\ddot{x}_p(-1),\end{aligned}$$

que reorganizadas ficam da seguinte forma:

$$\ddot{x}_p = -\frac{b}{(M + m)}\dot{x}_p + \frac{ml}{(M + m)}\ddot{\theta} + \frac{1}{(M + m)}F, \quad (5.38a)$$

$$\ddot{\theta} = \frac{ml}{(I + ml^2)}\ddot{x}_p + \frac{mgl}{(I + ml^2)}\theta. \quad (5.38b)$$

Observe que esse modelo mostrado em (5.38) é linear. Ou seja, a linearização foi realizada pela mera inspeção das características de funcionamento em torno do ponto de equilíbrio.

Conforme mencionado anteriormente, as duas equações de movimento obtidas para o pêndulo invertido sobre o carro ainda requerem substituições mútuas de \ddot{x}_p e $\ddot{\theta}$. Feitas essas substituições o modelo linear final do sistema assume a forma mostrada a seguir:

$$\ddot{x}_p = -\frac{b(I + ml^2)}{d_{en}}\dot{x}_p + \frac{(ml)^2g}{d_{en}}\theta + \frac{(I + ml^2)}{d_{en}}F \quad (5.39a)$$

$$\ddot{\theta} = -\frac{mlb}{d_{en}}\dot{x}_p + \frac{mlg(M + m)}{d_{en}}\theta + \frac{ml}{d_{en}}F \quad (5.39b)$$

em que esse denominador comum d_{en} é dado por

$$d_{en} = (M + m)(I + ml^2) - (ml)^2.$$

A partir desta etapa do procedimento de modelagem pode-se dar continuidade à obtenção do modelo no espaço de estado para o sistema. Assim, as seguintes variáveis de estado são definidas:

- $x_1 = x_p$: posição do carro (m)
- $x_2 = \dot{x}_p$: velocidade do carro (m/s)
- $x_3 = \theta$: ângulo do pêndulo (rad)
- $x_4 = \dot{\theta}$: velocidade angular do pêndulo (rad/s)

em que assume-se como vetor de entrada, de estado e de saída, respec-

tivamente,

$u = F : \text{força } (N) \text{ aplicada ao carro}$

$$x = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \end{bmatrix}^T$$

$$y = Cx, \quad C = \mathbf{I}_4$$

Escrevendo-se o modelo mostrado em (5.39) com base nas variáveis de estado definidas, tem-se as seguintes equações de estado:

$$\dot{x}_1 = x_2 \quad (5.40a)$$

$$\dot{x}_2 = -\frac{b(I+ml^2)}{d_{en}}x_2 + \frac{(ml)^2g}{d_{en}}x_3 + \frac{(I+ml^2)}{d_{en}}u \quad (5.40b)$$

$$\dot{x}_3 = x_4 \quad (5.40c)$$

$$\dot{x}_4 = -\frac{mlb}{d_{en}}x_2 + \frac{mlg(M+m)}{d_{en}}x_3 + \frac{ml}{d_{en}}u \quad (5.40d)$$

$$(5.40e)$$

de onde as seguintes matrizes Jacobianas são obtidas:

$$A := \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{b(I+ml^2)}{d_{en}} & \frac{(ml)^2g}{d_{en}} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -\frac{mlb}{d_{en}} & \frac{mlg(M+m)}{d_{en}} & 0 \end{bmatrix}, \quad B := \begin{bmatrix} 0 \\ \frac{(I+ml^2)}{d_{en}} \\ 0 \\ \frac{ml}{d_{en}} \end{bmatrix},$$

$$C := \mathbf{I}_4, \quad D := \mathbf{0}_{4 \times 1},$$

sendo \mathbf{I}_4 uma matriz identidade 4×4 e $\mathbf{0}_{4 \times 1}$ um vetor nulo 4×1 .

Exemplo 5.3 Comparando a linearização no espaço de estado do Exemplo 5.1 com o outro caso mais complexo do pêndulo invertido sobre o carro no Exemplo 5.2, se torna importante destacar outras formas automatizadas, assistidas por computador, que permitam linearizar sistemas dinâmicos de maior complexidade matemática no que se refere às relações de não-linearidade e também ao número de variáveis de estado, de entrada e saída envolvidas.

Neste novo exemplo repetiremos a linearização do pêndulo invertido sobre o carro utilizando a tecnologia proprietária da Mathworks para Matlab e Simulink, definida nas funções trim (trimagem) e linmod (linearização de modelo). O procedimento se baseia nos seguintes passos:

1. Definição do modelo não-linear como uma classe de sistema do

Matlab ou Simulink, especificando as variáveis de entrada e as variáveis de estado do modelo.

2. *Utilização da função trim sobre o modelo não-linear, especificando as condições iniciais para as variáveis de estado e de entrada, preferencialmente mais próximas do ponto de equilíbrio ou da solução a qual se deseja determinar na fase de trimagem, tal como:*

$$[x_{eq}, u_{eq}, y_{eq}, dx_{eq}] = \mathbf{trim}('sistema', x0, u0) \quad (5.41)$$

em que o 'sistema' é o nome do sistema declarado no Matlab ou mesmo o nome do arquivo do diagrama de blocos do sistema produzido no Simulink.

3. *O último passo lineariza o modelo no ponto de equilíbrio obtido no passo anterior, fornecendo como resultado as matrizes Jacobianas:*

$$[A, B, C, D] = \mathbf{linmod}('sistema', x_{eq}, u_{eq}) \quad (5.42)$$

Neste exemplo o modelo não-linear mostrado na Equação (5.36), dada por

$$\begin{aligned} (M + m)\ddot{x}_p &= -b\dot{x}_p - ml\ddot{\theta} \cos \theta + ml\dot{\theta}^2 \sin \theta + F, \\ (I + ml^2)\ddot{\theta} &= -mlg \sin \theta - ml\ddot{x}_p \cos \theta, \end{aligned}$$

foi construído no Simulink conforme o diagrama mostrado na Figura 47 e o código mostrado na Tabela 19 foi utilizado para linearizar o sistema e comparar os resultados numéricos com o modelo teórico obtido no Exemplo 5.2.

As matrizes Jacobianas A, B, determinadas manualmente e pelo método automatizado são equivalentes a

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -0,1818 & 2,6727 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -0,4545 & 31,1818 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1,8182 \\ 0 \\ 4,5455 \end{bmatrix}, \quad (5.43)$$

existindo divergência somente a partir da nona casa decimal no elemento 2×3 da matriz A e em casas decimais ainda maiores para os demais elementos. Mesmo assim, é importante frisar que diferentes

versões do Matlab e Simulink poderão melhorar ou degradar esses resultados, já que essas tecnologias são atualizadas frequentemente.

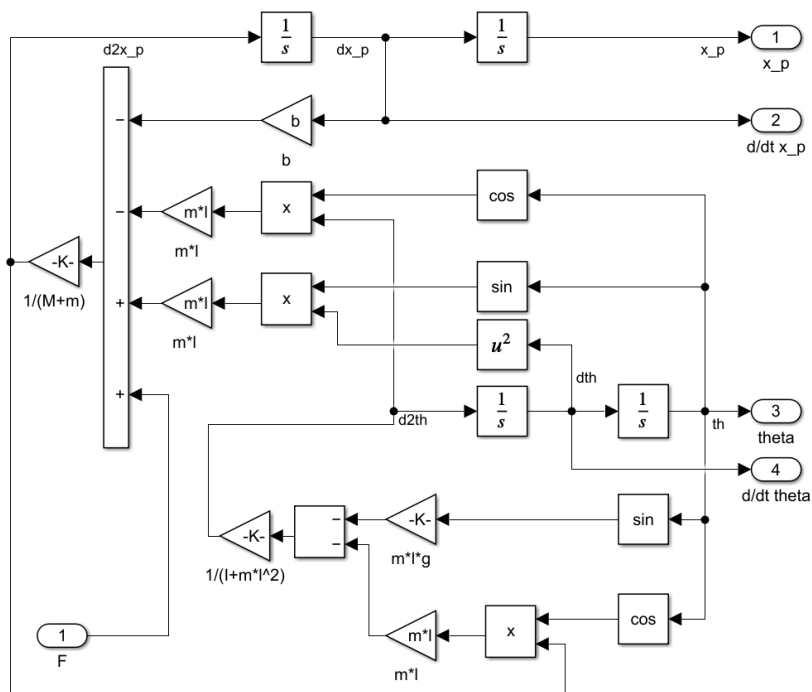


Figura 47 – Diagrama de simulação para Simulink do modelo não-linear da Equação (5.36) do pêndulo invertido sobre o carro.

Tabela 19 – Código em Matlab para executar a linearização no espaço de estado, assistida por computador, usando a tecnologia proprietária do Matlab e Simulink. Aplicação sobre o sistema mostrado no diagrama da Fig. 47.

```

%% Busca solução via método iterativo de trimagem (trim)
x0 = [0 0 pi 0]'; %[xp d/dt_xp theta d/dt_theta]
F0 = 0;

[Xeq,Feq,Yeq,DXeq]=trim('inv_pend_R2011a',x0,F0)

%% Lineariza o modelo usando Xeq e Feq determinados
[A,B,C,D]=linmod('inv_pend_R2011a',Xeq,Feq)

%% Conferindo os valores do modelo linear obtido com aqueles
%% calculados manualmente:
M = 0.5; m = 0.2; b = 0.1; l = 0.3; I = 0.006; g = 9.8;
d_en = (M+m)*(I+m*l^2)-(m*l)*(m*l);

Am = [ 0 1 0 0;
       0 -(b*(I+m*l^2))/d_en ((m*l)^2)*g/d_en 0;
       0 0 0 1;
       0 -(m*l*b)/d_en (m*l*g*(M+m))/d_en 0];
Bm = [0; (I+m*l^2)/d_en ; 0 ; m*l/d_en ];

```

6 CAPÍTULO 6

Conceitos fundamentais em controle preditivo baseado em modelo. Modelos com atraso. Modelos com perturbações determinística e estocástica. Modelagem via Estimador dos Mínimos Quadrados Recursivo. Controle preditivo baseado no Preditor de Smith. Controle de Variância Mínima Generalizado (GMVC). Controle por Matriz Dinâmica (DMC). Controle Preditivo Generalizado (GPC).

6.1 INTRODUÇÃO AO CONTROLE PREDITIVO

Neste capítulo são introduzidas algumas estratégias de Controle Preditivo baseado em Modelo (MPC).

De maneira geral os MPCs utilizam modelos para gerar previsões da saída do sistema controlado a fim de antecipar as ações de controle que manobrarão o sistema para a referência desejada. Geralmente, tal procedimento acarreta menor consumo energético e também melhor desempenho no seguimento de referência.

Os MPCs são populares, principalmente, como estratégias de Controle Ótimo e de Controle Adaptativo, as quais se baseiam em minimizar o esforço de controle e o erro de seguimento de referência, ambos preditos, atualizando, a cada novo período de amostragem, o modelo do processo controlado, reprojando todo o sistema de controle em tempo real. No entanto, há também estratégias MPC não ótimas, como, por exemplo, aquelas baseadas no Preditor de Smith (SILVEIRA et al., 2021), usadas para compensar atrasos de transporte ou tempo morto.

Na Fig. 48, extraída da Wikipedia, tem-se uma ilustração do funcionamento geral da estratégia MPC, do ponto de vista dos principais sinais que nos permitem entender como um MPC se distingue de uma estratégia de controle não preditiva. Na figura entende-se por instante de tempo k como sendo o instante presente. Portanto, tudo após esse instante é previsão e ainda não aconteceu fora do software de controle, sendo mera imaginação artificial. Já olhando-se para antes do instante k tem-se o passado, de fato mensurado, ocorrido.

O parâmetro de maior destaque do MPC é o horizonte de previsão, que na Fig. 48 é representado por p e neste exemplo, tanto a previsão do controle, quanto a previsão da saída, estão a p -passos no futuro. Desse modo, significa que a estratégia MPC otimiza $k + p$ ações de controle futuras visando minimizar o erro de seguimento de

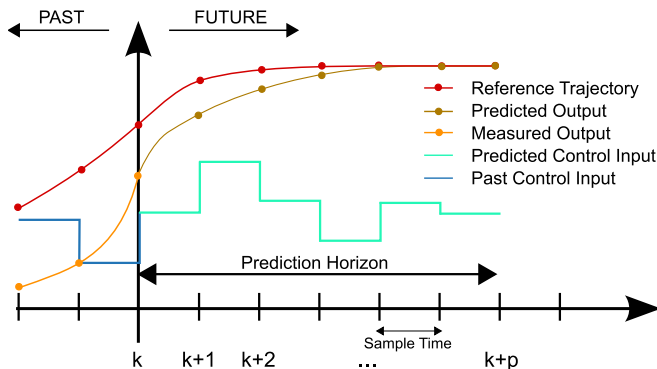


Figura 48 – Ilustração do funcionamento da estratégia MPC (fonte: Wikipedia).

referência através de uma trajetória definida de k até $k + p$ no futuro. Destaca-se assim que outro ponto marcante dos MPCs é a consideração da existência de uma sequência futura de referência, o conhecimento *a priori* de onde se quer estar no instante $k + p$, para todo novo instante $k = 0, 1, \dots, N$, onde N é um instante de tempo discreto final.

Associar relações de causa e efeito ao longo de uma trajetória de previsões, i.e., controle predito e saída predita de k até $k + p$, requer mais memória e esforço computacional. Em alguns casos é possível diminuir tal complexidade e pode-se então definir o problema de predição com base em dois horizontes: o horizonte de predição de saída N_y e o horizonte de predição de controle N_u , tal que $N_u \leq N_y$.

São muitas as técnicas de controle que compõem a família de estratégias MPC, de tal forma que N_y e N_u acabaram por receber diversas recomendações de sintonia ao longo dos anos. De todo o modo, geralmente esses horizontes são sintonizados por tentativa e erro, dada a facilidade a qual se costuma encontrar um par N_y, N_u , que estabiliza a planta controlada com desempenho satisfatório, na maioria dos casos. Adicionalmente, um termo de ponderação do esforço de controle, λ , completa a sintonia dos MPCs, o que nos permite sugerir que o MPC pode ser pensado como uma alternativa a outra família de controladores de três parâmetros de sintonia: o PID.

Este estudo em caráter introdutório sobre MPC se inicia pela apresentação sucinta dos problemas inerentes aos sistemas com atraso de transporte, tempo morto ou latência. Após, são discutidos os efeitos das perturbações estocásticas inerentes aos erros de modelagem e

aos ruídos de medição, seguindo-se ao estudo de como estas perturbações estocásticas podem ser estimadas experimentalmente com base no estimador paramétrico dos Mínimos Quadrados Recursivo (MQR).

Na sequência do capítulo inicia-se o estudo com base na aplicação do Preditor de Smith como estratégia de compensação de atraso, dando ênfase à aplicação de controle PID com o Preditor. Esta etapa é caracterizada pelo emprego de estratégia não ótima de MPC. Após, os seguintes métodos ótimos (e adaptativos) são estudados: o Controle de Variância Mínima Generalizado (GMVC), o Controle por Matriz Dinâmica (DMC) e o Controle Preditivo Generalizado (GPC).

Destaca-se que este capítulo aborda, exclusivamente, os métodos MPC desenvolvidos com base em funções de transferência.

6.2 MODELOS COM ATRASO OU TEMPO MORTO

Considere o sistema dinâmico mostrado a seguir:

$$G_d(s) = G(s)e^{-\theta s} \xrightarrow[T_s]{ZOH} G(z)z^{-\left(\frac{\theta}{T_s}\right)} = G_d(z). \quad (6.1)$$

Este sistema, descrito por função de transferência, apresenta um atraso puro de tempo de θ segundos (ou minutos, horas, etc), tal que o seu equivalente discreto via ZOH apresenta o atraso discreto de θ/T_s amostras. Esta relação de θ/T_s deve resultar em um número inteiro e, quando necessário, costuma-se aproximar pelo inteiro superior, por se tratar da condição de pior caso, de maior atraso.

Assumindo que o sistema é estritamente próprio (i.e., a ordem do polinômio do numerador é maior que a do denominador), o atraso de tempo discreto total será $d = 1 + \theta/T_s$:

$$\frac{Y(z)}{U(z)} = G(z)z^{-\left(\frac{\theta}{T_s}\right)} = \frac{B(z)z^{-1}}{A(z)}z^{-\left(\frac{\theta}{T_s}\right)} = \frac{B(z)}{A(z)}z^{-d}. \quad (6.2)$$

Considerando o operador de atraso de tempo discreto z^{-1} , onde $z^{-1}F(z) := f(k-1)$, o sistema pode também ser representado por

$$y(k) = G(z)u(k-d) = \frac{B(z)}{A(z)}u(k-d), \quad (6.3)$$

onde

$$\begin{aligned} B(z) &= b_0 + b_1 z^{-1} + \cdots + b_{n_b} z^{-n_b}, \\ A(z) &= 1 + a_1 z^{-1} + \cdots + a_{n_a} z^{-n_a}, \end{aligned} \quad (6.4)$$

cujas equações a diferenças é descrita como se mostra a seguir:

$$\begin{aligned} y(k) &= -a_1 y(k-1) - \cdots - a_{n_a} y(k-n_a) \\ &+ b_0 u(k-d) + \cdots + b_{n_b} u(k-d-n_b). \end{aligned} \quad (6.5)$$

Os sistemas dinâmicos com atraso de transporte, tempo morto ou latência, demandam maior atenção para quando se precisa controlá-los em malha fechada. Isto se deve ao fato de que o atraso acarreta em diminuição da curva de fase em frequência, fazendo com que a frequência de inversão de fase, aquela na qual a fase é de -180° , diminua consideravelmente, podendo acarretar severa restrição de banda para o projeto de controle em malha fechada.

Para melhor ilustrar os problemas de estabilidade que precisam ser considerados no caso de sistemas com atraso, considere o exemplo mostrado a seguir.

Exemplo 6.1 *Considere os seguintes sistemas mostrados a seguir, um sem atraso e outro com atraso, respectivamente descritos por:*

$$G(s) = \frac{1}{s+1} \xrightarrow[T_s=0,1s]{ZOH} \frac{0,09516}{1-0,9048z^{-1}} z^{-1} = G(z), \quad (6.6)$$

$$G_d(s) = \frac{1}{s+1} e^{-10s} \xrightarrow[T_s=0,1s]{ZOH} \frac{0,09516}{1-0,9048z^{-1}} z^{-101} = G_d(z). \quad (6.7)$$

Os diagramas de magnitude e de fase em frequência, de $G(e^{j\omega T_s})$ e de $G_d(e^{j\omega T_s})$, mostrados nas figuras 49 e 50, respectivamente, podem ser comparados para destacar que a frequência de inversão de fase reduz, de 31,42 rad/s para 0,285 rad/s, devido ao atraso deste exemplo. No entanto, como não há alteração nas curvas de magnitude, seria necessário considerar uma drástica redução da banda passante da estratégia de controle para se estabelecer margens de ganho e de fase seguras para o controle em malha fechada.

Como forma de exemplificar os efeitos do controle em malha fechada nesses casos sem e com atraso, considere a aplicação de um

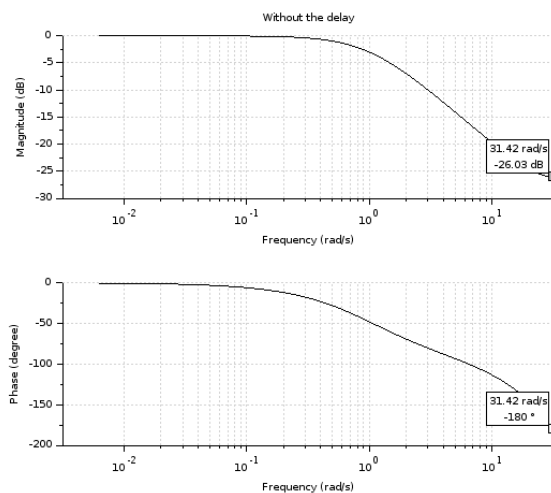


Figura 49 – No caso do sistema **sem atraso**, há forte atenuação energética na frequência de inversão de fase (i.e., fase em -180°).

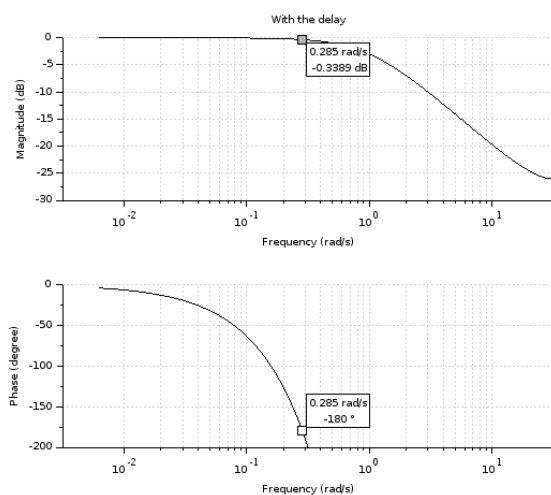


Figura 50 – No caso do sistema **com atraso**, na frequência de inversão de fase (i.e., fase em -180°), a amplitude é elevada, exigindo maior cautela em projeto de controle em malha fechada.

controle proporcional em que a lei de controle é dada por:

$$u(k) = K_p [r(k) - y(k)] . \quad (6.8)$$

Considere também que o ganho $K_p = 1$ é adotado, tal que os diagramas das figuras 49 e 50 representem a análise de estabilidade relativa via resposta em frequência. O resultado da aplicação desse controlador nos dois casos sob análise são mostrados nas figuras 51 e 52.

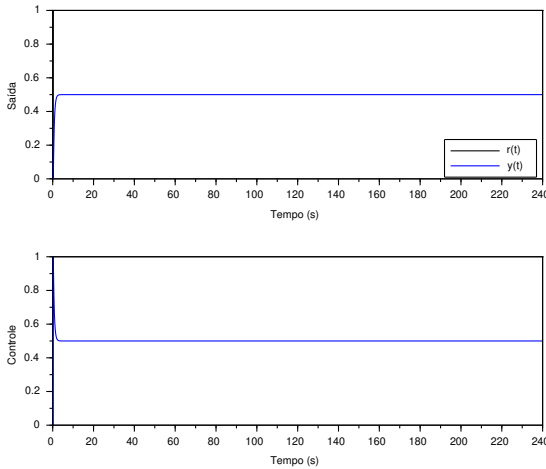


Figura 51 – No caso do sistema **sem atraso**, o controle proporcional com $K_p = 1$ opera de forma estável e sem oscilações.

O controle proporcional com $K_p = 1$ garante margem de ganho de 26,03 dB e margem de fase de 180° . Ou seja, com grandes margens de estabilidade relativa, tal que sua resposta ao degrau em malha fechada não apresenta oscilações, como se observa na Fig. 51.

Já para o caso com atraso, a margem de ganho é de apenas 0,33 dB e a margem de fase é praticamente nula, o que resulta em um sistema em malha fechada oscilatório, como se observa na Fig. 52.

Este exemplo deve servir de alerta para os perigos no controle de sistemas com atraso, destacando a necessidade de tal problema ser levado em consideração quando a duração do atraso não puder ser negligenciada.

O código em Scilab que auxilia a reprodução deste exemplo é mostrado na Tabela 20.

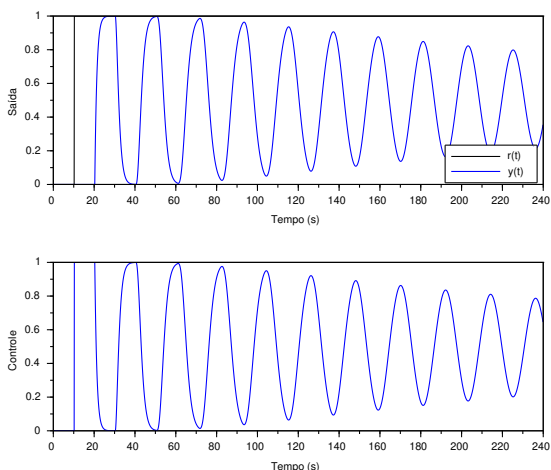


Figura 52 – No caso do sistema **com atraso**, o controle proporcional com $K_p = 1$ opera de maneira oscilatória, o que era esperado devido às margens de estabilidade relativas serem inadequadas à implementação deste controlador no caso com atraso.

Tabela 20 – Código em Scilab para reproduzir o Exemplo 6.1.

```
// Controle Proporcional sobre sistemas sem e com atraso
clear; xdel(winsid()); clc;

s=%s; Gs = syslin('c', 1/(s+1));
theta = 0; // Atraso em segundos

z=%z; Ts = 0.1; // Período de amostragem em segundos
Gz = ss2tf( dscr(Gs, Ts) ); // G(z) Sem atraso

Az = coeff(Gz.den); Az = flipdim(Az,2);
a1 = Az(2);
Bz = coeff(Gz.num); Bz = flipdim(Bz,2);
b0 = Bz(1);

theta_d = (theta/Ts); // Atraso discreto
Gdz = Gz*z^(-theta_d); // Gd(z), com atraso
d = 1+(theta/Ts); // Atraso discreto total

bode(Gz,'rad'); title('Caso sem atraso');
scf(); bode(Gdz,'rad'); title('Caso com atraso');

// Simulando G(z) e Gd(z) no tempo com controle P
tfinal = 240; // tempo final em segundos
N = round(tfinal/Ts); // número total de iterações
r(1:d+2)=0; r(d+3:N)=1; // Degrau de referência
y(1:d+1)=0; // Condição inicial da saída
u(1:d+1)=0; // Condição inicial do controle
for k=d+2:N // Laço de simulação e controle
    y(k) = -a1*y(k-1) + b0*u(k-d);
    u(k)=1*( r(k)-y(k) ); // Lei de controle P
end
t=0:Ts:N*Ts-Ts; // Vetor de tempo para os plots
```

```

scf(); subplot(211); plot(t,r,'k'); plot(t,y,'b');
ylabel('Saída'); xlabel('Tempo (s)');
legend('r(t)', 'y(t)', 4);
subplot(212); plot(t,u,'b');
ylabel('Controle'); xlabel('Tempo (s)');

```

6.3 MODELOS COM PERTURBAÇÕES DETERMINÍSTICA E ESTOCÁSTICA

As técnicas de controle, preditivas ou não, devem ser capazes de garantir a estabilidade das malhas de controle mesmo quando situações adversas ocorrem. Tais situações costumam ser simuladas via modelos de perturbações que podem afetar os sistemas a partir da entrada, da saída e internamente (casos no espaço de estado).

Consideremos a seguir alguns exemplos de modelos de perturbação que permitirão a realização de testes via simulações para avaliar o desempenho das estratégias de controle preditivo que serão estudadas neste capítulo. De maneira introdutória, apenas serão considerados modelos lineares. No entanto, tais formas de se perturbar os sistemas também podem ser empregadas sobre modelos não lineares.

- Modelo com perturbação de carga na saída do processo:

$$Y(z) = \frac{B(z)}{A(z)} z^{-d} U(z) + D_o(z), \quad (6.9)$$

$$A(z)y(k) = B(z)u(k-d) + A(z)d_o(k). \quad (6.10)$$

Este modelo se destaca pela participação do polinômio característico $A(z)$ filtrando o sinal de perturbação de carga na saída, $d_o(k)$. Usualmente aplica-se uma sequência em degrau como sendo $d_o(k)$, permitindo a rápida verificação, via resposta no tempo, se o algoritmo de controle é capaz de eliminar tal perturbação.

- Modelo com ruído na saída do processo:

$$Y(z) = \frac{B(z)}{A(z)} z^{-d} U(z) + V(z), \quad (6.11)$$

$$A(z)y(k) = B(z)u(k-d) + A(z)v(k). \quad (6.12)$$

Do ponto de vista estrutural, este modelo é idêntico ao modelo de carga na saída do processo, mas destaca-se a mudança de variável para retratar uma sequência $v(k)$ que é constituída por sinal de

alta frequência, podendo ser constituído de uma única frequência ou de frequências múltiplas, porém, com média nula, tal que se diferencie do teste de carga.

- Modelo de perturbação de carga na entrada do processo:

$$Y(z) = \frac{B(z)}{A(z)} z^{-d} [U(z) + D_i(z)], \quad (6.13)$$

$$A(z)y(k) = B(z)u(k-d) + B(z)d_i(k-d). \quad (6.14)$$

Este modelo considera que uma perturbação de carga perturba o sinal de entrada do processo. Por exemplo, em um sistema de controle digital pode haver diferença entre o valor calculado pelo software de controle para a ação de controle e o valor de fato gerado pelo conversor digital analógico, pois pode existir erros de quantização.

- Modelo do tipo ruído de processo:

$$Y(z) = \frac{B(z)}{A(z)} z^{-d} U(z) + \frac{1}{A(z)} W(z), \quad (6.15)$$

$$A(z)y(k) = B(z)u(k-d) + w(k). \quad (6.16)$$

Este modelo caracteriza uma perturbação Gaussiana, média nula, tal que $\frac{1}{A(z)}w(k)$ é designado como um processo estocástico, dada a natureza aleatória da sequência $w(k)$. A variância desta sequência, σ_w^2 , simboliza a potência linear da incerteza do modelo do processo, podendo, também, ser relacionada em termos energéticos, como sendo a potência dissipada sobre uma resistência elétrica de 1 ohm.

A variância da perturbação $w(k)$ pode ser estimada experimentalmente, por exemplo, pelo uso de estimação paramétrica via mínimos quadrados recursivo e não recursivo, onde

$$w(k) = y(k) - \hat{y}(k), \quad (6.17)$$

isto é, a diferença entre a saída medida do processo e a saída estimada $\hat{y}(k)$. De posse da sequência $w(k)$, experimentalmente

obtida, calcula-se a variância desta como sendo:

$$\sigma_w^2 = \frac{1}{N} \sum_{i=1}^N [w(k) - \mu_w]^2, \mu_w = \frac{1}{N} \sum_{i=1}^N w(k). \quad (6.18)$$

Observe que N representa o total de amostras analisadas e μ_w é a média da sequência de $w(k)$, que, mesmo experimentalmente, costuma ser desprezível.

Exemplo 6.2 *Considere os seguintes sistemas mostrados a seguir, um sem atraso e outro com atraso, respectivamente descritos por:*

$$G(s) = \frac{4,4}{s^2 + 2s + 4}, \quad (6.19)$$

$$G_d(s) = \frac{4,4}{s^2 + 2s + 4} e^{-2s}. \quad (6.20)$$

Discretize esses sistemas e avalie-os em malha fechada usando um sistema de controle PID digital sintonizado com base na tabela de sintonia obtida pelo método Internal Model Control (IMC), mostrada no Capítulo 2.

*Após obter o desempenho de malha fechada desejado para ambos os sistemas (**usar os mesmos ganhos**), comparando os casos sem e com atraso, acrescente, também, os seguintes testes:*

- (i) *Teste de perturbação de carga na saída;*
- (ii) *Teste de perturbação de carga na entrada;*
- (iii) *Teste de ruído na saída medida (pode usar ruído Gaussiano).*

Considere o seguimento de referência do tipo degrau unitário e, as perturbações de carga na saída e na entrada sendo também do tipo degrau, sendo 10% e 1%, respectivamente, do valor do sinal de referência.

Já para a simulação do ruído de medição, considere a variância do ruído no valor de 0,001. Teste o seguimento de referência para quando a saída estiver apenas sob a perturbação deste ruído (sem as cargas), e avalie o desempenho das malhas de controle dos sistemas sem e com atraso com base nas variâncias do sinal de controle $u(k)$, do sinal variação discreta do sinal de controle, $\Delta u(k)$, e do sinal de erro de seguimento, $e(k) = y_r(k) - y(k)$.

Para resolver este exemplo foi considerado o controlador PID na estrutura paralela, descrita por:

$$C(s) = \frac{U(s)}{E(s)} = K_p + K_i \frac{1}{s} + K_d s. \quad (6.21)$$

Uma síntese digital deste controlador via aproximação Backward Difference, onde

$$s := \frac{1 - z^{-1}}{T_s}, \quad (6.22)$$

foi adotada, se chegando ao seguinte controlador PID digital:

$$C(z) = \frac{U(z)}{E(z)} = \frac{s_0 + s_1 z^{-1} + s_2 z^{-2}}{1 - z^{-1}}. \quad (6.23)$$

A relação entre os ganhos do PID contínuo, K_p , K_i , K_d , e os parâmetros do PID digital é descrita por:

$$\begin{aligned} s_0 &= K_p + K_i T_s + \frac{K_d}{T_s}, \\ s_1 &= -K_p - 2\frac{K_d}{T_s}, \\ s_2 &= \frac{K_d}{T_s}. \end{aligned} \quad (6.24)$$

Conforme solicitou-se neste exemplo, os ganhos do PID foram estabelecidos usando-se o método de sintonia IMC apresentado no Capítulo 2, onde o único parâmetro que o projetista precisa especificar é o valor da constante de tempo de malha fechada τ_{MF} , de maneira que o comportamento de malha fechada seja definido pelo seguinte modelo de referência contínuo:

$$G_{MF}(s) = \frac{Y(s)}{R(s)} = \frac{1}{\tau_{MF}s + 1}, \quad (6.25)$$

onde $R(s)$ é o sinal de referência a ser seguido.

O $\tau_{MF} = 3$ foi selecionado para este exemplo didático, não havendo, neste mero exemplo numérico, qualquer critério de restrição ou sugestão de τ_{MF} . Do mesmo modo, escolheu-se o período de amostragem $T_s = 0.1$ segundo, estando, este, relacionado a uma frequência de amostragem ω_s entre 4 a 10 vezes mais veloz que ω_n do processo con-

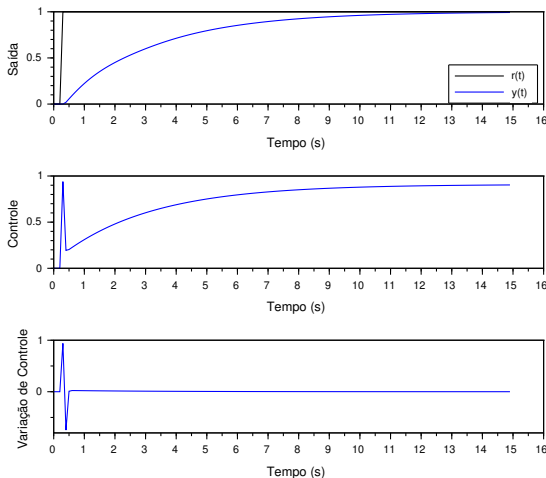


Figura 53 – Controle PID digital sintonizado via método IMC sem considerar o atraso do sistema, mas aplicado no sistema que de fato não contém atraso, resultando na resposta desejada e de acordo com a sintonia selecionada.

trolado. Com este T_s , o atraso discreto total é $d = 21$. Ou seja, a toda ação de controle, somente é possível observar as consequências após 21 amostras ou 2,1 segundos.

Iniciemos a análise deste exemplo considerando que o controlador foi sintonizado sem a consideração da existência do atraso $\theta = 2$ segundos. Nesta situação, este controlador, se aplicado sobre o processo sem atraso, seria obtido o resultado mostrado na Fig. 53, adequado ao que se espera com base no modelo de referência desejado, mostrado em (6.25). No entanto, ao se considerar este sistema de controle com esta mesma sintonia, mas aplicado ao modelo com atraso, é possível verificar no resultado da Fig. 54 que perde-se o desempenho ideal, tal que a saída que deveria ser amortecida se degrada ao caso subamortecido.

Vejamos agora o experimento inverso ao que foi apresentado nas figuras 53 e 54. Neste novo caso, o controlador é sintonizado pelo método IMC e considerando que o processo tem o atraso $\theta = 2$ segundos. Nesta situação é possível verificar nas figuras 55 e 55, respectivas ao caso sem e com atraso, que a sintonia que considera a situação de pior caso é mais robusta, sendo capaz de amortecer a saída do sistema nos casos sem e com atraso.

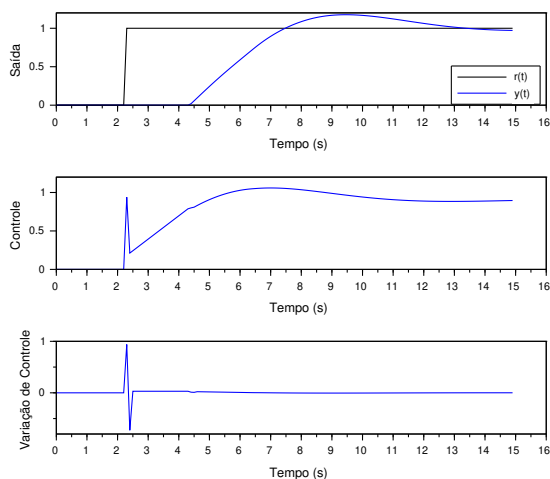


Figura 54 – Controle PID digital sintonizado via método IMC sem considerar o atraso do sistema, mas aplicado sobre o processo com atraso. Observe que a saída controlada se degrada, se tornando subamortecida e divergindo da resposta idealizada devido ao atraso.

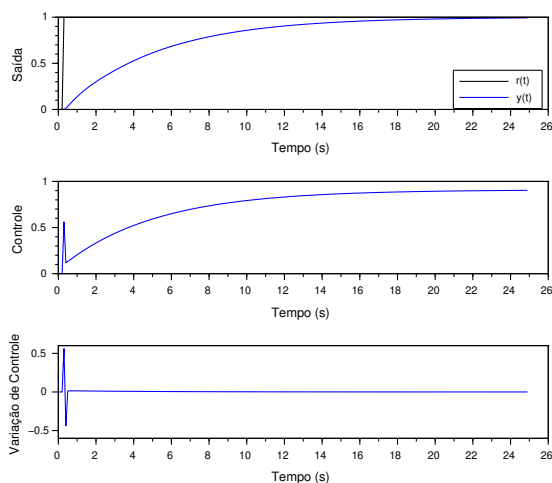


Figura 55 – Controle PID digital sintonizado via método IMC considerando o atraso do sistema e aplicado sobre o processo sem atraso. Nesse caso sem atraso, o controlador tendo sido sintonizado para uma condição mais crítica consegue estabilizar o sistema e evitar oscilações.

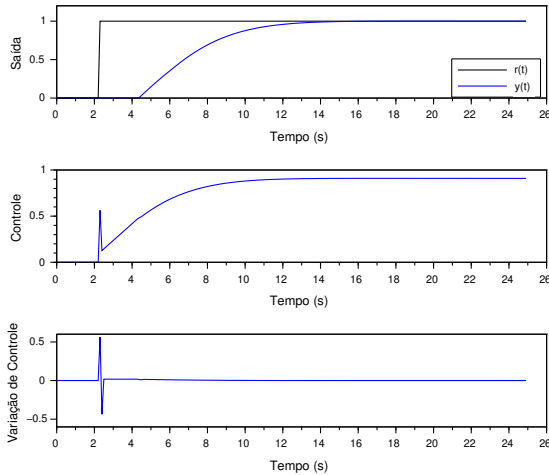


Figura 56 – Controle PID digital sintonizado via método IMC considerando o atraso do sistema e aplicado sobre o processo com atraso. O resultado estabelecido pelo projetista é alcançado.

Com base nos resultados apresentados, a partir deste ponto define-se que a estratégia de sintonia deve ser aquela que considera a condição de pior caso, que é aquela com a participação do atraso. Usando-se esta sintonia aplicou-se então os testes de perturbação de carga na saída, na entrada e também de ruído Gaussiano somado à saída do processo, o que resultou nos gráficos mostrados nas figuras 57, 58, 59, respectivamente.

A reprodução deste exemplo pode ser feita usando-se o código para Scilab apresentado na Tabela 21.

Tabela 21 – Código em Scilab para reproduzir o Exemplo 6.2.

```
// Controle PID sobre sistemas sem e com atraso
clear; xdel(winsid()); clc;

// Modelo do processo do exemplo
wn = 2; tau = 1/wn; zeta = 0.5; Ks = 1.1;
s = %s; Gs = syslin('c', (Ks*wn^2)/(s^2+2*zeta*wn*s+wn^2));
theta = 2; // Atraso em segundos

z = %z; Ts = 0.1; // Período de amostragem em segundos
Gz = ss2tf( dscr(Gs, Ts) ); // G(z) Sem atraso

Az = coeff(Gz.den); Az = flipdim(Az,2);
a1 = Az(2); a2 = Az(3);
Bz = coeff(Gz.num); Bz = flipdim(Bz,2);
```

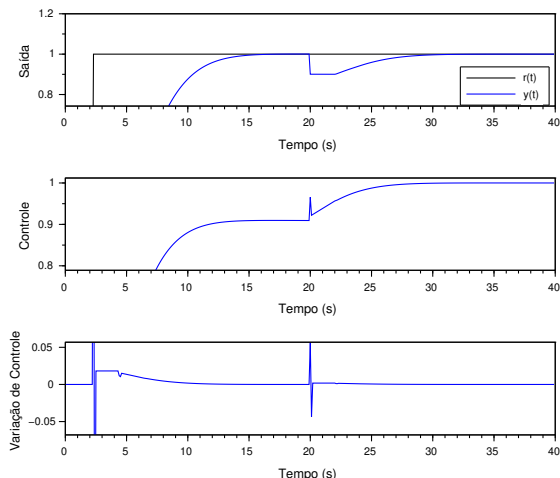



Figura 57 – Teste de perturbação de carga na saída com controle PID digital sintonizado via método IMC considerando o atraso do sistema. Observe que o atraso influencia no tempo de reação e correção da perturbação de carga medida na saída da planta.

```

b0 = Bz(1); b1 = Bz(2);

theta_d = (theta/Ts); // Atraso discreto
Gdz = Gz*z^(-theta_d); // Gd(z), com atraso
d = 1+(theta/Ts); // Atraso discreto total

bode(Gz,'rad'); title('Caso sem atraso');
scf();bode(Gdz,'rad'); title('Caso com atraso');

// Sintonia do PID digital com base no PID contínuo
tau_mf = 3; // cte tempo MF em segundos
Kp = (2*zeta*tau)/(Ks*(theta+tau_mf));
Ki = Kp/(2*zeta*tau);
Kd = (tau*Kp)/(2*zeta);

s0 = Kp*Ki*Ts+Kd/Ts;
s1 = -Kp-2*Kd/Ts;
s2 = Kd/Ts;

// Simulando G(z) e Gd(z) no tempo com controle P
tfinal = 40; // tempo final em segundos
N = round(tfinal/Ts); // número total de iterações

// Ruído de medição
// Distribuição normal/Gaussiana: "nor"
Av = 0; // média
Variancia = 0.001; // Potência Linear
Sd = sqrt(Variancia); // Sd: desvio padrão
v = 0*grand(1, N, 'nor', Av, Sd); // Gaussian noise generation

// Sequência de carga na saída
dout(1:N/2)=0; dout(N/2+1:N)=-0.1*0;

// Sequência de carga na entrada
din(1:N/2)=0; din(N/2+1:N)=0.01*0;

```

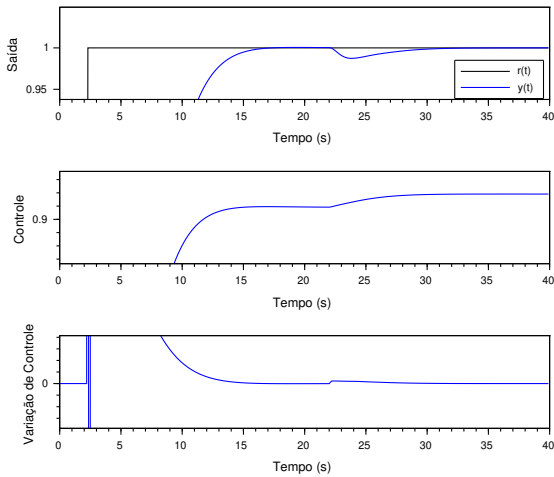


Figura 58 – Teste de perturbação de carga na entrada com controle PID digital sintonizado via método IMC considerando o atraso do sistema. É difícil observar a influência do atraso no tempo de reação e correção da perturbação de carga na entrada da planta, pois tal perturbação se mistura ao sinal de controle. Para detectar tal influência indica-se investigar índices de desempenho no domínio do tempo, como as variâncias do erro de seguimento de referência e do sinal de controle.

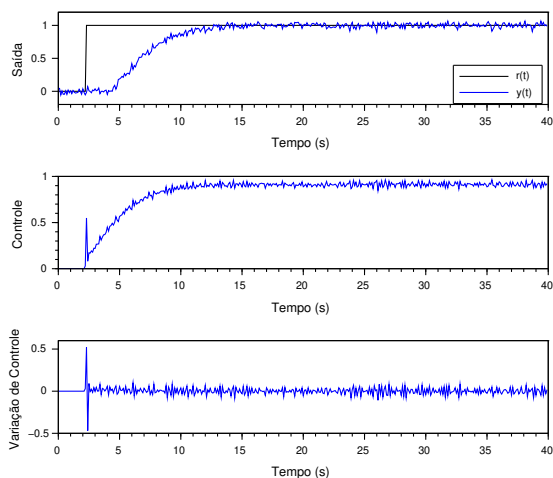


Figura 59 – Teste de perturbação do tipo ruído de medição somado à saída da planta. Teste com controle PID digital sintonizado via método IMC considerando o atraso do sistema. O fator da influência do atraso para o caso com ruído de medição é difícil de se perceber pela mera observação dos gráficos, exigindo que se investigue índices de desempenho no domínio do tempo, como a variância do erro de seguimento de referência e do sinal de controle.

```

// Condições iniciais
r(1:d+2)=0; r(d+3:N)=1; // Degrau de referência
y(1:d+1)=v(1:d+1); // Condição inicial da saída com ruído
u(1:d+1)=0; // Condição inicial do controle
du(1:d+1)=0; // Condição inicial da variação do controle

for k=d+2:N // Laço de simulação e controle
    y(k) = -a1*y(k-1) -a2*y(k-2) +b0*u(k-d) +b1*u(k-d-1) ...
    +v(k)+a1*v(k-1)+a2*v(k-2) ... // ruído
    +1*dout(k)+a1*dout(k-1)+a2*dout(k-2) ... // pert. na saída
    +b0*din(k-d)+b1*din(k-d-1); // pert. na entrada

    // Controle PID
    e(k) = r(k)-y(k);
    u(k)=u(k-1) +s0*e(k)+s1*e(k-1)+s2*e(k-2); // Lei de controle PID
    du(k)=u(k)-u(k-1);

end
t=0:Ts:N*Ts-Ts; // Vetor de tempo para os plots
scf(); subplot(311); plot(t,r,'k'); plot(t,y,'b');
    ylabel('Saída'); xlabel('Tempo (s)');
    legend('r(t)','y(t)',4);
subplot(312); plot(t,u,'b');
    ylabel('Controle'); xlabel('Tempo (s)');
subplot(313); plot(t,du,'b');
    ylabel('Variação de Controle'); xlabel('Tempo (s)');

scf(); plot(t,e,'b'); ylabel('Erro'); xlabel('Tempo (s)');
title('Erro de seguimento de referência');

```

6.4 ESTIMADOR DOS MÍNIMOS QUADRADOS RECURSIVO

No Capítulo 1 discutiu-se o método de estimação paramétrica via Mínimos Quadrados não recursivo, o qual se baseia em uma estrutura de dados de entrada e saída, de N amostras de um processo, que se encontra disponível *a priori*, como segue:

$$\mathbf{Y} = \Phi \hat{\theta} + \mathbf{w} = \hat{\mathbf{Y}} + \mathbf{w}, \quad (6.26)$$

onde \mathbf{Y} , Φ , $\hat{\theta}$, \mathbf{w} , são, respectivamente, o vetor com os dados de saída medidos, a matriz de informação que contém os regressores de saída e de entrada, o vetor de parâmetros estimados e o vetor de erros de estimação ou ruídos de processo.

No caso não recursivo o problema MQ é resolvido em batelada, para todo o conjunto de dados, de uma só vez ao se resolver o seguinte problema:

$$\hat{\theta} = (\Phi^T \Phi)^{-1} \Phi^T \mathbf{Y} = \mathbf{P} \Phi^T \mathbf{Y}. \quad (6.27)$$

A matriz $\mathbf{P} = (\Phi^T \Phi)^{-1}$ é uma matriz simétrica e semidefinida positiva ($\mathbf{P} \geq 0$), chamada de matriz de covariância do erro de estimação paramétrica e que representa a potência do erro de estimação, tal que quanto menor for o seu traço, $\text{tr}(\mathbf{P})$. Quanto menor for a magnitude da soma dos elementos da diagonal principal de \mathbf{P} , melhor.

Destaca-se, no entanto, que a consideração de uma \mathbf{P} melhor ou pior depende de uma análise relativa entre ao menos dois modelos estimados e comparados.

Diferente da solução em batelada, o caso recursivo tem o objetivo de estimar o vetor de parâmetros a cada novo instante de amostragem k , $\hat{\theta}(k)$. Nesse sentido, a solução estática dos Mínimos Quadrados retorna ao problema dinâmico que o originou, descrito por:

$$y(k) = \phi^T(k)\hat{\theta}(k) + w(k), \quad (6.28)$$

$$\phi^T(k) = \begin{bmatrix} -y(k-1) & \cdots & -y(k-n_a) & u(k-d) & \cdots & u(k-d-n_b) \end{bmatrix}. \quad (6.29)$$

No caso recursivo considera-se que $\hat{\theta}(k)$ é também um sistema dinâmico e variante no tempo, dado por:

$$\hat{\theta}(k+1) = I\hat{\theta}(k) + L(k)[y(k) - \hat{y}(k)], \quad (6.30)$$

$$\hat{y}(k) = \phi^T(k)\hat{\theta}(k). \quad (6.31)$$

As equações (6.30) e (6.31) se assemelham a um sistema dinâmico descrito no espaço de estado e do tipo observador de estado. Mas neste caso, as variáveis de estado são os parâmetros do sistema em (6.28).

O objetivo do estimador MQR pode ser definido em se determinar o ganho de estimador $L(k)$, tal que o erro de estimação $w(k) = y(k) - \hat{y}(k)$ seja minimizado. Para que este problema de otimização seja resolvido, considera-se que a seguinte função custo do MQR seja minimizada:

$$J_{MQR} = \frac{1}{2} \sum_{k=0}^{\infty} w^2(k). \quad (6.32)$$

Este problema de otimização, com foco especificamente no estimador paramétrico MQR, foi discutido extensivamente na literatura de Controle e Sistemas, como em Coelho e Coelho (2004), Aguirre (2007), Astrom e Wittenmark (2008), entre outros. O desenvolvimento é um tanto complexo e pode ser estudado nas obras desses grandes autores da área. No entanto, uma contribuição que pode ser feita nesta presente abordagem do estimador MQR é relacioná-lo ao Filtro de Kalman, pois se trata de um caso mais geral da Teoria de Estimação Ótima e Robusta

e que tem como solução particular o estimador MQR.

No Capítulo 5 abordamos o Filtro de Kalman como solução para estimação de estado. É possível aplicar as mesmas equações que calculam a matriz de covariância e o ganho de Kalman para o sistema descrito pelas equações (6.30) e (6.31), tal que, se $\hat{y}(k) \rightarrow y(k)$, então $\hat{y}(k) = \phi^T(k)\hat{\theta}(k) \rightarrow y(k) = \phi^T(k)\theta(k)$. Portanto, $\hat{\theta}(k) \rightarrow \theta(k)$ para quando $k \rightarrow \infty$.

Com base nas equações do ganho de Kalman e de atualização da matriz de covariância adaptativa $P(k)$ mostradas no Capítulo 5, obtém-se o ganho e a matriz de covariância de Kalman para o problema de estimação paramétrica ótimo definido em (6.32) sujeito ao sistema descrito por (6.30) e (6.31):

$$L(k) = P(k)\phi(k)[R(k) + \phi^T(k)P(k)\phi(k)]^{-1}, \quad (6.33)$$

$$P(k+1) = P(k) - L(k)\phi^T(k)P(k) + Q(k). \quad (6.34)$$

As matrizes $R(k)$ e $Q(k)$ são matrizes de covariância respectivas às incertezas de medição e incertezas nos estados, no caso do Filtro de Kalman. Em relação ao estimador MQR, estas são consideradas $R(k) = I$ e $Q(k) = 0$ para todo k .

No caso de sistemas SISO, $R(k)$ é um escalar unitário, representando que a incerteza nas medidas possui variância unitária, $\sigma_w^2 = 1$. Destaca-se, assim, que o que na literatura sobre o estimador MQR é chamado de **fator de esquecimento**, normalmente designado pelo símbolo λ_{ff} e sintonizado por tentativa e erro, corresponde, no caso mais geral que é o Filtro de Kalman, às incertezas de medição na forma de uma matriz de covariância e que pode ser estimada em tempo real, tal como será abordado no Capítulo 7.

O estimador MQR, para sistemas SISO, pode ser resumido pelo seguinte conjunto de equações mostradas a seguir:

$$\phi^T(k) = \begin{bmatrix} -y(k-1) & \cdots & -y(k-n_a) & u(k-d) & \cdots & u(k-d-n_b) \end{bmatrix}, \quad (6.35)$$

$$L(k) = P(k)\phi(k)[1 + \phi^T(k)P(k)\phi(k)]^{-1}, \quad (6.36)$$

$$\hat{y}(k) = \phi^T(k)\hat{\theta}(k), \quad (6.37)$$

$$\hat{\theta}(k+1) = \hat{\theta}(k) + L(k)[y(k) - \hat{y}(k)], \quad (6.38)$$

$$P(k+1) = P(k) - L(k)\phi^T(k)P(k). \quad (6.39)$$

Os parâmetros estimados do sistema podem então ser obtidos diretamente do vetor de parâmetros como se mostra a seguir:

$$\hat{\theta}^T(k) = [\hat{a}_1(k) \quad \cdots \quad \hat{a}_{n_a}(k) \quad \hat{b}_0(k) \quad \cdots \quad \hat{b}_{n_b}(k)]. \quad (6.40)$$

Exemplo 6.3 *Aplicação do estimador MQR para a obtenção de um modelo de segunda ordem do subsistema de velocidade lateral de um drone do tipo quadrirotor: neste exemplo utiliza-se dados de entrada e saída registrados em voo de um quadrirotor modelo AR.Drone 2.0 fabricado pela Parrot. Os dados estão disponíveis no arquivo texto cujo link é mostrado a seguir:*

https://lacos.ufpa.br/plantas/ardrone/datalog_lat_vel.txt

O arquivo com os dados está estruturado em três colunas, referentes ao tempo, à entrada de comando lateral adimensional operando na faixa $[-1; 1]$ e à saída de velocidade de translação lateral dada em metros por segundo. Os dados foram registrados com um período de amostragem $T_s = 0,065$ s com o quadrirotor operando em malha aberta. A estrutura de modelo utilizada é do tipo ARX e descrita por:

$$y(k) = -\hat{a}_1y(k-1) - \hat{a}_2y(k-2) + \hat{b}_0u(k-1) + \hat{b}_1u(k-2) + w(k) \quad (6.41)$$

O algoritmo MQR implementado para este exemplo é apresentado na Tabela 22. O seu uso é direto, tal como descrito nas equações de (6.35) a (6.39). Os únicos ajustes que podem ser feitos no algoritmo são as condições iniciais de $P(k)$ e $\hat{\theta}(k)$. Neste exemplo consideramos uma inicialização sem qualquer conhecimento prévio onde os parâmetros iniciais em $\hat{\theta}(k)$ eram nulos, enquanto que $P(0)$ foi experimentado em duas condições: I e $100I$, caracterizando, respectivamente, um caso de menor magnitude de incerteza inicial e outro de maior magnitude de incerteza inicial.

Os principais resultados deste exemplo são mostrados nas figuras de 60 a 65.

Na Fig. 60 verifica-se como a saída estimada acompanha a saída medida ao longo do tempo. Nesta figura pode-se destacar como a sele-

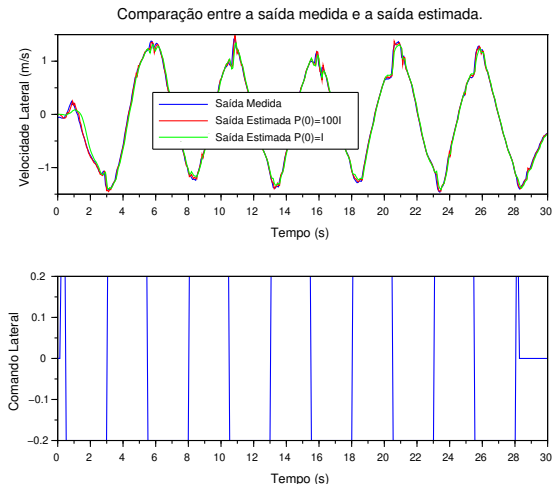


Figura 60 – Convergência das saídas estimadas à saída medida. Curva em cor azul para o caso iniciado com $P(0) = 100I$ e curva na cor verde para o caso assumindo menor incerteza inicial, iniciado com $P(0) = I$.

ção da matriz inicial $P(0)$ influencia na velocidade de convergência do MQR, tal que a curva na cor verde, referente à matriz $P(0) = I$, leva mais tempo para convergir à saída medida nos primeiros instantes do experimento. Maior destaque a esta característica é dado na Fig. 61, onde destacou-se os primeiros instantes da estimação, sendo possível verificar como a curva na cor verde leva mais tempo para convergir à saída medida.

Na adaptação dos parâmetros também é possível verificar a influência da definição inicial da matriz $P(k)$. Tal verificação é destacada na Fig. 62, a qual traz na cor azul as trajetórias dos parâmetros estimados com a inicialização $P(0) = 100I$, de maior incerteza, enquanto que as trajetórias na cor vermelha referem-se ao caso com $P(0) = I$, de menor incerteza inicial.

É importante destacar que apesar das diferentes condições iniciais de $P(k)$ acarretarem diferentes características de convergência à saída medida, ambas, quando $k \rightarrow \infty$, garantem que $\hat{y}(k) \rightarrow y(k)$. No entanto, para qual condição de inicialização de $P(0)$ ou mesmo para qual sistema dinâmico $y(k)$ o estimador MQR poderá falhar é algo que sempre gerou preocupação no uso desse estimador para aplicações em controle adaptativo. Além disso, quando ocorre a convergência do al-

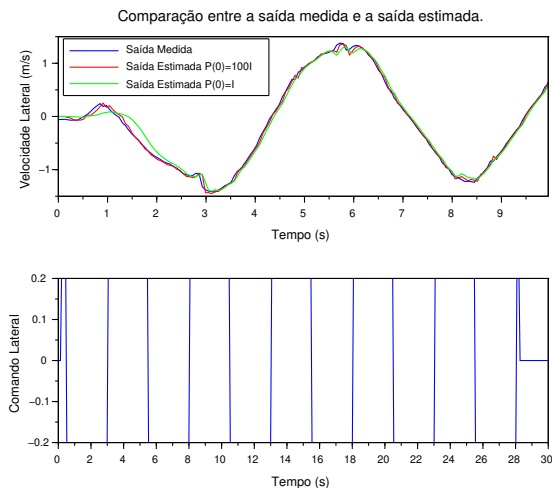


Figura 61 – Convergência das saídas estimadas à saída medida, destacando a fase inicial do experimento de estimação onde pode-se notar a curva na cor verde, caso para $P(0) = I$, assumindo menor incerteza inicial, levando mais tempo para convergir à saída medida.

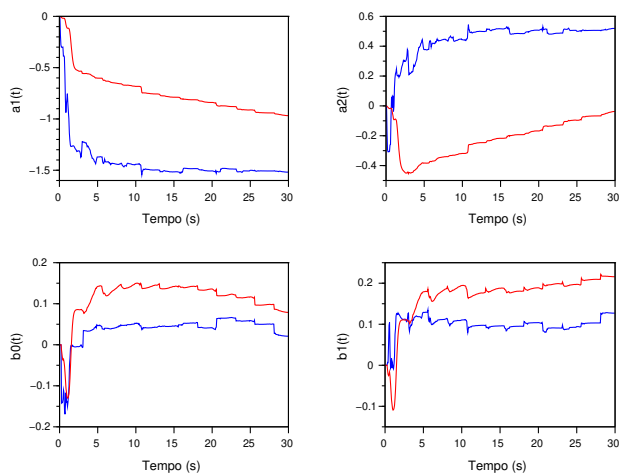


Figura 62 – Trajetória de adaptação dos parâmetros estimados, onde as curvas em azul referem-se ao caso com $P(0) = 100I$ e, as em vermelho, ao caso com $P(0) = I$.

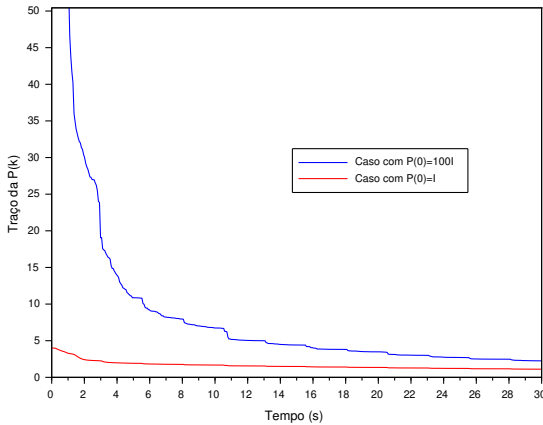


Figura 63 – Traço da matriz $P(k)$ ao longo do experimento. A curva em azul retrata o caso iniciado com $\text{tr}[P(0)] = 400$ e, a curva em vermelho, o caso com $\text{tr}[P(0)] = 4$. Ambos os casos tendem a uma condição de incerteza mínima conforme $k \rightarrow \infty$.

goritmo MQR, a matriz $P(k)$ se adapta ao longo do tempo e quando $k \rightarrow \infty$ as incertezas se reduzem e consequentemente se reduz o traço da Matriz de Covariância, decaindo também a capacidade do MQR de convergir rapidamente no caso de haver alguma mudança abrupta do sistema que se deseja estimar os parâmetros. Esse efeito pode ser destacado como se mostra na Fig. 63.

O resíduo do problema de estimação, representado pelo sinal erro de estimação $w(k)$, é mostrado na Fig. 64. A expectativa teórica é de que este sinal seja Gaussiano e que tenha média nula. Para este caso mostrado na Fig. 64, referente à inicialização com $P(0) = 100I$, a variância calculada foi $\sigma_w^2 = 0,0022770$ e a média foi $\mu_w = 0,0020416$. Ou seja, em um experimento real com um número reduzido de amostras (e.g., $N = 462$, neste caso), esse valor médio pode ser considerado nulo? A resposta a esta pergunta, do ponto de vista de uma aplicação em engenharia, é uma decisão de projeto e que pode ser justificada ao se relativizar com outros experimentos de estimação paramétrica e com a respectiva análise dos resíduos $w(k)$.

Apesar das propriedades dos ruídos de processo estimados para $w(k)$ em experimentos reais serem passíveis de críticas, ainda é possível

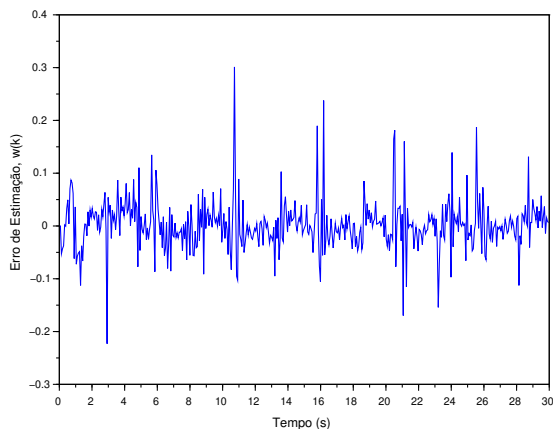


Figura 64 – Sinal $w(k)$ estimado.

justificar que, ao menos não seria possível afirmar que o sinal não seria Gaussiano, quando o seu conteúdo harmônico é investigado e comparado ao conteúdo harmônico observado no experimento com o processo que se quer estimar um modelo. Por exemplo, na Fig. 65 mostra-se a Transformada Rápida de Fourier (FFT) de $w(k)$ e de $y(k)$, respectivamente representadas nas cores azul e vermelho, em que destaca-se como a informação em frequência fornecida por $w(k)$ se apresenta irrelevante e de certo modo normalizada ao longo de todo o espectro de Nyquist, $f_{Nyq} = f_s/2$ (i.e., a metade do espectro amostrado, $f_s = 1/T_s$) se comparada às informações em frequência extraídas de $y(k)$.

Com base no conteúdo harmônico de $w(k)$ apresentado na Fig. 65, não foi possível observar, dentro do espectro de Nyquist, qualquer elevação de amplitude significativa se comparado ao conteúdo harmônico de $y(k)$. Portanto, do ponto de vista de engenharia, visando a obtenção de um modelo ARX para aplicação em sistemas de controle, pode-se assumir que $w(k)$ estimado é Gaussiano e média nula neste exemplo experimental com MQR.

Tabela 22 – Código em Scilab para reproduzir o Exemplo 6.3.

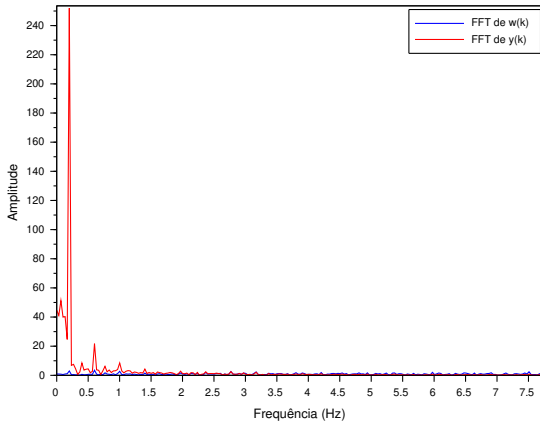


Figura 65 – Transformada Rápida de Fourier de $w(k)$ e de $y(k)$ como forma de relativizar o conteúdo harmônico desses processos, tal que não se possa negar que $w(k)$ não poderia ser considerado um ruído Gaussiano.

```
// Estimador dos Mínimos Quadrados Recursivo
// Aplicação sobre os dados de velocidade lateral
// do quadricóptero AR.Drone 2.0
clear; xdel(winsid()); clc;

dados = fscanfMat('datalog_lat_vel.txt');
// No Matlab: dados = load('datalog_lat_vel.txt');
t = dados(:,1); u = dados(:,2); y = dados(:,3);
N = length(y); // número total de amostras
Ts = t(2)-t(1); // Período de Amostragem

Pini = 100*eye(4,4); P=Pini; // Covariância inicial
theta = [0 0 0 0]'; // Parâmetros iniciais
for k = 3:N
    // Algoritmo MQR
    phi = [-y(k-1) -y(k-2) u(k-1) u(k-2)]'; // Vetor de regressores
    L = P*phi*inv(1+phi'*P*phi); // Ganho do MQR
    yhat(k) = phi'*theta; // Saída estimada
    w(k) = y(k)-yhat(k); // Erro de estimação
    theta = theta + L*(u(k)); // Atualização do vetor de parâmetros
    P = P - L*phi'*P; // Atualização da Matriz de Covariância

    // Obter os parâmetros para análise futura
    a1(k) = theta(1); a2(k) = theta(2);
    b0(k) = theta(3); b1(k) = theta(4);
end

// Plots dos sinais de saída e entrada
subplot(211);
plot(t,y,'b',t,yhat,'r'); legend('Saída Medida','Saída Estimada');
title('Comparação entre a saída medida e a saída estimada. ');
ylabel('Velocidade Lateral (m/s)'); xlabel('Tempo (s)');
subplot(212);
plot(t,u); ylabel('Comando Lateral'); xlabel('Tempo (s)');

// Plots da variação dos parâmetros ao longo do tempo
```

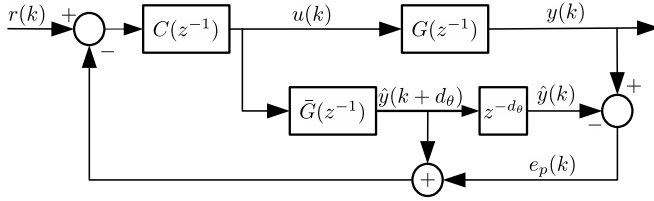


Figura 66 – Diagrama de blocos de uma malha de controle preditivo baseada no Preditor de Smith. O modelo do preditor, $\bar{G}(z^{-1})$, tem comportamento dinâmico aproximado ao do sistema $G(z^{-1})$, porém sem o atraso $d_\theta = \theta/T_s$.

```

scf();
subplot(221); plot(t,a1); ylabel('a1(t)'); xlabel('Tempo (s)');
subplot(222); plot(t,a2); ylabel('a2(t)'); xlabel('Tempo (s)');
subplot(223); plot(t,b0); ylabel('b0(t)'); xlabel('Tempo (s)');
subplot(224); plot(t,b1); ylabel('b1(t)'); xlabel('Tempo (s)');

// Plot do sinal erro de estimação, w(k)
scf(); plot(t,w);
ylabel('Erro de Estimação, w(k)'); xlabel('Tempo (s)');

// Análise via FFT de y(k) versus w(k)
fs=1/Ts; fd=fs*(0:(N-1))/(N-1); // vetor discreto de frequência
fft_y=fft(y); fft_w=fft(w);
scf(); plot(fd,abs(fft_w),'b'); plot(fd,abs(fft_y),'r');
legend('FFT de w(k)', 'FFT de y(k)');
ylabel('Amplitude'); xlabel('Frequência (Hz)');

```

6.5 CONTROLE PREDITIVO BASEADO NO PREDITOR DE SMITH

Nos anos 1950 Otto J. M. Smith propôs uma estrutura de controle preditivo em malha fechada para processos estáveis em malha aberta. Esta estrutura é demonstrada pelo diagrama de blocos apresentado na Figura 66.

O elemento chave na estrutura de Smith foi a simplicidade pela qual um observador livre de atraso, $\bar{G}(z^{-1})$, era utilizado para produzir uma saída predita a d_θ -passos no futuro, $\hat{y}(k+d_\theta)$. Assim, idealmente, o controlador $C(z^{-1})$ trabalharia sempre d -passos à frente, de maneira a guiar $\hat{y}(k+d_\theta)$ na direção da referência $r(k)$. E, se a principal diferença entre o observador e o processo real $G(z^{-1})$ fosse o atraso, então o erro de predição $e_p(k) = y(k) - \hat{y}(k)$ seria nulo, garantindo a compensação dos efeitos do atraso nas margens de estabilidade de malha fechada (SILVEIRA et al., 2021).

O problema pode ser descrito, matematicamente, ao se conside-

rar que o processo controlado é descrito por

$$G(z^{-1}) = \frac{Y(z^{-1})}{U(z^{-1})} = \frac{B(z^{-1})}{A(z^{-1})} z^{-d}, d = 1 + d_\theta \quad (6.42)$$

e que um modelo deste processo, sem o atraso d_θ , pode ser simulado internamente no *software* de controle, sendo este descrito da seguinte forma:

$$\bar{G}(z^{-1}) = \frac{\bar{Y}(z^{-1})}{U(z^{-1})} = \frac{B(z^{-1})}{A(z^{-1})} z^{-1}. \quad (6.43)$$

As equações a diferenças do sistema controlado e do modelo de observador são, respectivamente, descritas como se mostra a seguir:

$$A(z)y(k + d_\theta) = B(z)u(k - 1), \quad (6.44)$$

$$A(z)\bar{y}(k) = B(z)u(k - 1). \quad (6.45)$$

Comparando estas duas equações a diferenças é possível verificar que, quando o modelo do sistema controlado é idealmente conhecido, então a saída do modelo de observador, $\bar{y}(k)$, é equivalente a:

$$\bar{y}(k) = y(k + d_\theta). \quad (6.46)$$

É importante destacar que, ao se considerar um cenário realístico, o modelo do observador jamais será perfeitamente equivalente ao comportamento dinâmico de um sistema real, não linear e variante no tempo. Desse modo, pode-se então representar o sistema observador com polinômios diferentes daqueles do sistema controlado, enfatizando, matematicamente, o fato de que planta e modelo são diferentes. Assim, reescreve-se o observador com base nos novos polinômios e considera-se que a saída é predita e simbolizada como $\hat{y}(k + d_\theta)$:

$$\bar{A}(z^{-1})\hat{y}(k + d_\theta) = \bar{B}(z^{-1})u(k - 1). \quad (6.47)$$

Para completar a estratégia de controle baseada no Preditor de Smith, verifiquemos a composição do erro predito $e_p(k)$. Para isto verifica-se na Fig. 66 que o erro predito é constituído pela seguinte relação:

$$\begin{aligned} e_p(k) &= y(k) - \hat{y}(k + d_\theta)z^{-d_\theta}, \\ e_p(k) &= y(k) - \hat{y}(k). \end{aligned} \quad (6.48)$$

Já o sinal de erro de seguimento de referência que chega na entrada do bloco controlador, na Fig. 66, é constituído pela relação dada por:

$$e(k) = r(k) - [\hat{y}(k + d_\theta) + e_p(k)], \quad (6.49)$$

$$e(k) = r(k) - y(k) + [\hat{y}(k) - \hat{y}(k + d_\theta)]. \quad (6.50)$$

A lei de controle com o Preditor de Smith pode então ser definida como sendo constituída de uma parcela idêntica ao caso sem preditor, mais a nova parcela com a contribuição de Smith, como se mostra a seguir:

$$u(k) = C(z^{-1}) [r(k) - y(k)] + C(z^{-1}) [\hat{y}(k) - \hat{y}(k + d_\theta)]. \quad (6.51)$$

Ou mesmo, uma outra forma de raciocinar sobre o problema é considerar que

$$u(k) = C(z^{-1}) [r(k) - \hat{y}(k + d_\theta)] + C(z^{-1}) [\hat{y}(k) - y(k)], \quad (6.52)$$

tal que, se a saída estimada tender à saída medida, $\hat{y}(k) \rightarrow y(k)$, então a malha de controle vai operar como se não houvesse o atraso, mas isto somente é possível quando $\bar{A}(z^{-1}) \rightarrow A(z^{-1})$ e $\bar{B}(z^{-1}) \rightarrow B(z^{-1})$.

É importante frisar que o uso de métodos de estimação paramétrica que operam em tempo real na malha de controle podem garantir que $\bar{A}(z^{-1}) \rightarrow A(z^{-1})$ e $\bar{B}(z^{-1}) \rightarrow B(z^{-1})$, permitindo que a compensação do atraso se aproxime do desempenho ideal proposto por Smith. Além disso deve ser destacado que, o conhecimento, *a priori*, do valor do atraso d , é necessário para se implementar o Preditor de Smith.

Exemplo 6.4 *Controle PID digital adaptativo com Preditor de Smith aplicado sobre um processo subamortecido e com atraso:*

6.6 CONTROLE DE VARIÂNCIA MÍNIMA GENERALIZADO

O Controle de Variância Mínima Generalizado (GMVC) é um método de projeto que permite agregar funcionalidades do Controle Linear Quadrático, do Controle Estocástico, do Controle Preditivo e do Controle Adaptativo, sendo o GMVC um método de solução particular e mais simples do LQG, realizado com base em modelos polinomiais (SILVEIRA; COELHO, 2011).

O GMVC foi proposto por Clarke e Gawthrop (1975) como uma

extensão ao Regulador de Variância Mínima proposto por Åström (1970). Em ambos os casos, no entanto, a proposta era a de resolver um problema de otimização que garantisse a menor variância (i.e., potência linear) das variáveis de interesse em um determinado problema de controle. Porém, não somente o problema buscava minimizar a variância, mas acrescentava tal minimização de forma preditiva.

Para que possamos entender detalhadamente o problema GMVC, extrapolemos, neste primeiro momento, o modelo de projeto, para um caso mais complexo de sistema auto-regressivo média móvel de entradas exógenas, ou ARMAX, acrônimo este que na língua Inglesa se refere a *Auto-Regressive Moving Average with exogenous inputs*:

$$A(z^{-1})y(k) = B(z^{-1})u(k-d) + C(z^{-1})w(k), \quad (6.53)$$

$$\begin{aligned} A(z^{-1}) &= 1 + a_1 z^{-1} + \cdots + a_{n_a} z^{-n_a}, \\ B(z^{-1}) &= b_0 + b_1 z^{-1} + \cdots + b_{n_b} z^{-n_b}, \\ C(z^{-1}) &= 1 + c_1 z^{-1} + \cdots + c_{n_c} z^{-n_c}. \end{aligned} \quad (6.54)$$

No ARMAX considera-se a participação de um processo estocástico, descrito por $C(z^{-1})w(k)$, influenciando a saída do sistema. Neste caso, o ruído Gaussiano $w(k)$, cuja variância é σ_w^2 e a média é nula, é processado por um filtro $C(z^{-1})$ que exalta e eleva certas componentes dinâmicas que atuam como perturbações no modelo do processo. Nesse sentido pode-se dizer que, se o ruído $w(k)$ é um ruído branco, após este ser processado por $C(z^{-1})$ obtém-se um ruído colorido $\bar{w}(k) = C(z^{-1})w(k)$.

É importante ressaltar que, seja em malha aberta ou em malha fechada, sempre a saída do processo estará sujeita às perturbações do processo estocástico (SILVEIRA et al., 2016), como segue:

$$y(k) = \frac{B(z^{-1})}{A(z^{-1})}u(k-d) + \frac{C(z^{-1})}{A(z^{-1})}w(k). \quad (6.55)$$

O método GMVC se propõe a implementar um sistema de controle, preditivo e generalizado em termos dos filtros que ponderam o sinal de controle $u(k)$, de saída $y(k)$ e de referência $r(k)$, conforme se mostra a seguir:

$$Q(z)u(k) = T(z)r(k+d) - P(z)y(k+d), \quad (6.56)$$

$$\begin{aligned}
Q(z^{-1}) &= q_0 + q_1 z^{-1} + \cdots + q_{n_q} z^{-n_q}, \\
T(z^{-1}) &= t_0 + t_1 z^{-1} + \cdots + t_{n_t} z^{-n_t}, \\
P(z^{-1}) &= p_0 + p_1 z^{-1} + \cdots + p_{n_p} z^{-n_p}.
\end{aligned} \tag{6.57}$$

A estrutura generalizada de controlador mostrada em (6.56) considera a participação da referência e da saída preditas a d -passo à frente. Pressupor que a referência pode ser conhecida *a priori* é factível em muitas aplicações industriais. Caso não se conheça $r(k+d)$ pode-se simplesmente adequar que $r(k+d) = \bar{r}(k)$, onde $\bar{r}(k)$ é uma referência conhecida no instante k . Já a saída predita $y(k+d)$, como esta é função de um processo estocástico, a sua determinação e conhecimento *a priori* não são factíveis, pois

$$y(k+d) = \frac{B(z^{-1})}{A(z^{-1})}u(k) + \frac{C(z^{-1})}{A(z^{-1})}w(k+d) \tag{6.58}$$

em que $w(k+d)$, o ruído no futuro, não é conhecido.

A impossibilidade de se determinar com precisão o valor da saída futura pode ser contornada, ao menos, com uma estratégia de estimação desse valor, por meio da saída predita $\hat{y}(k+d)$, o que consequentemente acarreta erros futuros que vamos considerar sob a influência de uma variável auxiliar $\phi(k+d)$ que modifica a lei de controle generalizada para a seguinte forma:

$$Q(z)u(k) = T(z)r(k+d) - P(z)\hat{y}(k+d) + \phi(k+d). \tag{6.59}$$

Com base na discussão sobre a $\phi(k+d)$, notemos que qualquer que seja a estratégia de controle estabelecida pela escolha dos polinômios generalizados $Q(z^{-1})$, $T(z^{-1})$, $P(z^{-1})$, quanto maior for a influência de $\phi(k+d)$ mais vai se degradar a estratégia de controle idealizada via $Q(z^{-1})$, $T(z^{-1})$, $P(z^{-1})$. Portanto, minimizar a influência de $\phi(k+d)$ é crucial para garantir que o processo estocástico interfira minimamente na malha de controle.

Esta discussão é de extrema relevância do ponto de vista funcional da estratégia GMVC, pois a estratégia não fornece de fato o sistema de controle, mas sim garante que um determinado controlador, descrito por $Q(z^{-1})$, $T(z^{-1})$, $P(z^{-1})$, tenha o seu funcionamento otimizado quando na presença de incertezas de modelagem e ruído de medição, caracterizados pelo processo estocástico $\frac{C(z^{-1})}{A(z^{-1})}w(k)$, que perturba o sistema.

A seleção de $Q(z^{-1})$, $T(z^{-1})$, $P(z^{-1})$, de maneira clássica, pode

ser associada aos projetos de controladores via alocação de polos. No entanto, conforme veremos mais adiante, quanto maior for a complexidade dos polinômios $Q(z^{-1})$, $T(z^{-1})$, $P(z^{-1})$, também quanto maior for o atraso d que se deseja compensar, mais complexo se torna o projeto GMVC. Devido à complexidade, o GMVC passou a ser utilizado mais frequentemente como um controlador em que $Q(z^{-1}) = q_0$ e $T(z^{-1}) = P(z^{-1}) = 1$, chamado por Silveira e Coelho (2011) de GMVC de ordem mínima. Tal simplicidade acarreta limitações de desempenho, quando aplicado em certas classes de processos como processos integradores (tipo-1), mas como permite que o método GMVC seja usado diretamente sem a necessidade de realizar um projeto prévio para determinar $Q(z^{-1})$, $T(z^{-1})$, $P(z^{-1})$, o caso de ordem mínima será abordado exemplos mais adiante neste texto.

Conforme discutiu-se acerca da lei de controle mostrada em (6.59), os efeitos de $\phi(k+d)$ devem ser minimizados. Para isto, reescreve-se (6.59) dando ênfase a $\phi(k+d)$, doravante designada pelo nome *saída generalizada* e descrita por:

$$\phi(k+d) = P(z)\hat{y}(k+d) - T(z)r(k+d) + Q(z)u(k). \quad (6.60)$$

Para minimizar a saída generalizada define-se uma função custo quadrática,

$$\bar{J}_{GMV} = \mathbf{E} [\phi^2(k+d)], \quad (6.61)$$

onde $\mathbf{E}[\cdot]$ representa o operador esperança matemática, e resolve-se o seguinte problema de otimização:

$$\frac{\partial \bar{J}_{GMV}}{\partial u(k)} = 0, \quad (6.62)$$

$$s.a. \ P(z^{-1})y(k+d) = \frac{P(z^{-1})B(z^{-1})}{A(z^{-1})}u(k) + \frac{P(z^{-1})C(z^{-1})}{A(z^{-1})}w(k+d). \quad (6.63)$$

Como o problema de otimização do GMVC está sujeito a $y(k+d)$, mas esta predição depende de se conhecer $w(k+d)$ e cujo valor é aleatório, então recorre-se ao Preditor de Variância Mínima proposto

por Åström (1970):

$$P(z^{-1})\hat{y}(k+d|k) = \frac{P(z^{-1})B(z^{-1})}{A(z^{-1})}u(k) + \frac{F(z^{-1})}{A(z^{-1})}w(k), \quad (6.64)$$

onde $F(z^{-1})$ é um filtro determinado pelo método do projeto e definido mais adiante. Este preditor, representado pela saída predita a d -passos à frente com base em informações presentes até o instante atual k , isto é, $\hat{y}(k+d|k)$, se utiliza da estimação $w(k)$ para tentar prever a sua influência futura na saída $\hat{y}(k+d|k)$.

Åström (1970) propôs que o processo estocástico que perturba o modelo do sistema pode ser separado em duas parcelas, sendo uma que o descreve até o instante presente e outra que o descreve no futuro, como segue:

$$\frac{P(z^{-1})C(z^{-1})}{A(z^{-1})}w(k+d) = \underbrace{\frac{F(z^{-1})}{A(z^{-1})}w(k)}_{\text{Passado e Presente}} + \underbrace{E(z^{-1})w(k+d)}_{\text{Futuro}}. \quad (6.65)$$

Neste modelo proposto são utilizados dois novos polinômios, $F(z^{-1})$ e $E(z^{-1})$, cujos parâmetros são obtidos a partir da identidade polinomial que se define após as seguintes manipulações algébricas:

$$P(z^{-1})C(z^{-1})w(k+d) = z^{-d}F(z^{-1})w(k+d) + A(z^{-1})E(z^{-1})w(k+d). \quad (6.66)$$

Tal identidade polinomial, também chamada de equação Diofantina do Preditor de Variância Mínima, é dada por:

$$P(z^{-1})C(z^{-1}) = z^{-d}F(z^{-1}) + A(z^{-1})E(z^{-1}). \quad (6.67)$$

Os polinômios $E(z^{-1})$ e $F(z^{-1})$ são polinômios generalizados e descritos da seguinte forma:

$$\begin{aligned} E(z^{-1}) &= e_0 + e_1 z^{-1} + \cdots + e_{n_e} z^{-n_e}, \\ F(z^{-1}) &= f_0 + f_1 z^{-1} + \cdots + f_{n_f} z^{-n_f}, \end{aligned} \quad (6.68)$$

$$\begin{aligned} n_e &= d - 1, \\ n_f &= \max[(n_p + n_c), (n_a + n_e)] - d. \end{aligned} \quad (6.69)$$

Assumindo que a partir deste ponto os polinômios $E(z^{-1})$ e $F(z^{-1})$ já são conhecidos, então retorna-se à equação do Preditor de Variância Mínima em (6.64), isto é,

$$P(z^{-1})\hat{y}(k+d|k) = \frac{P(z^{-1})B(z^{-1})}{A(z^{-1})}u(k) + \frac{F(z^{-1})}{A(z^{-1})}w(k),$$

onde apenas a parcela do passado até o presente do processo estocástico é considerada. Portanto supõe-se que o erro de predição está relacionado à parcela no futuro que foi desprezada, sendo esta descrita por:

$$E(z^{-1})w(k+d) = P(z^{-1})[y(k+d) - \hat{y}(k+d|k)]. \quad (6.70)$$

Este erro futuro em relação às medidas no instante k atual pode ser obtido simplesmente deslocando-se esta expressão d -passos para o passado, tal que

$$w(k) = \frac{P(z^{-1})}{E(z^{-1})}[y(k) - \hat{y}(k|k)]. \quad (6.71)$$

Substituindo esta expressão de $w(k)$ na equação do Preditor de Variância Mínima obtém-se:

$$\hat{y}(k+d|k) = \frac{B(z^{-1})E(z^{-1})u(k) + F(z^{-1})y(k)}{z^{-d}F(z^{-1}) + A(z^{-1})E(z^{-1})}. \quad (6.72)$$

Observe, no entanto, que o polinômio característico do Preditor de Variância Mínima é igual à equação Diofantina que é utilizada para se determinar os filtros deste Preditor. Portanto, é possível descrevê-lo, também, da seguinte forma:

$$\hat{y}(k+d|k) = \frac{B(z^{-1})E(z^{-1})u(k) + F(z^{-1})y(k)}{P(z^{-1})C(z^{-1})}. \quad (6.73)$$

Com base na função de transferência do Preditor de Variância Mínima é possível afirmar que este somente será assintoticamente estável se as raízes de $P(z^{-1})C(z^{-1})$ estiverem contidas no círculo unitário no plano z .

Com base no Preditor de Variância Mínima é possível reescrever a saída generalizada predita, mas considerando que esta predição é feita a partir de medidas disponíveis até o instante k atual, isto é:

$$\hat{\phi}(k+d|k) = P(z)\hat{y}(k+d|k) - T(z)r(k+d) + Q(z)u(k). \quad (6.74)$$

Como todas as variáveis envolvidas no problema de otimização GMVC agora estão disponíveis, uma nova função custo é estabelecida, como segue:

$$J_{GMV} = \hat{\phi}^2(k + d|k). \quad (6.75)$$

Substituindo (6.73) em J_{GMV} e resolvendo-se $\frac{\partial J_{GMV}}{\partial u(k)} = 0$ obtém-se a seguinte lei de controle:

$$u(k) = \frac{T(z^{-1})C(z^{-1})r(k + d) - F(z^{-1})y(k)}{B(z^{-1})E(z^{-1}) + C(z^{-1})Q(z^{-1})}. \quad (6.76)$$

Analizando o que foi desenvolvido sobre GMVC do ponto de vista da implementação do método, é possível resumir o projeto com os seguintes passos: (i) definir a estratégia de controle e os polinômios $P(z^{-1})$, $T(z^{-1})$, $Q(z^{-1})$; (ii) resolver a equação Diofantina (6.67) e determinar os polinômios $E(z^{-1})$ e $F(z^{-1})$; (iii) implementar a lei de controle (6.76).

O resumo do método sugere a ideia de que o GMVC é bem simples de se projetar. Porém, o mecanismo de solução da equação Diofantina e também a falta de um direcionamento sobre a escolha de $P(z^{-1})$, $T(z^{-1})$, $Q(z^{-1})$, acabam dificultando o projeto generalizado, tal que veremos que o GMVC de ordem mínima se torna mais convidativo já que dispensa a etapa de definir $P(z^{-1})$, $T(z^{-1})$, $Q(z^{-1})$.

É importante frisar que o GMVC foi proposto como um sistema do tipo *self-tuning* ou auto-ajustável, o qual se baseia no uso de algum método de estimação paramétrica para estimar o modelo do processo e reprojeter o GMVC. Ou mesmo, estimar diretamente os parâmetros do GMVC.

Nesses casos em que o GMVC é um sistema adaptativo pode-se considerar que o estimador paramétrico é capaz de determinar variações de carga e mudanças de ganho estático do processo controlado, tal que o controlador projetado consiga compensar perturbações ou mesmo seguir referências constantes do tipo degrau. No entanto, caso não seja considerada a forma adaptativa do GMVC, se faz necessário o uso de integrador no projeto para a garantia de rejeição de perturbação de carga e erro nulo de seguimento de referência.

Em controle baseado em modelo, de uma forma geral, a estrutura do modelo de projeto define a participação ou não de um projeto com ação de controle integral, pois, a estrutura do controlador, quando a técnica é baseada em modelo, é herdada do modelo do projeto. No caso do GMVC há exceção a esta regra, pois o GMVC atua sobre uma

estrutura de controle $P(z^{-1})$, $T(z^{-1})$, $Q(z^{-1})$, que pode conter ação de controle integral de maneira forçada, ou *ad hoc*, como especificaram Silveira et al. (2012).

Por exemplo, o uso de um polinômio $Q(z^{-1}) = 1 - z^{-1}$ operando sobre $u(k)$ gera uma ação de variação sobre $u(k)$, isto é, $Q(z^{-1})u(k) = u(k) - u(k-1)$. De todo o modo, faremos a distinção entre o projeto com ou sem integrador com base nos modelos de projeto, respectivamente descritos na **forma posicional** ou na **forma de velocidade**, mostradas a seguir:

$$A(z^{-1})y(k) = B(z^{-1})u(k-d) + C(z^{-1})w(k) \quad (6.77)$$

$$\Delta A(z^{-1})y(k) = B(z^{-1})\Delta u(k-d) + C(z^{-1})w(k), \quad (6.78)$$

onde $\Delta = 1 - z^{-1}$ é o operador de diferença.

Ao aumentarmos a descrição de um modelo, da forma posicional para a forma de velocidade, consideramos que o projeto de controle está visando a obtenção de uma lei de variação de controle, $\Delta u(k) = u(k) - u(k-1)$. Desse modo, o GMVC quando é projetado com base na forma de velocidade a sua lei de controle é reescrita da seguinte maneira:

$$\Delta u(k) = \frac{T(z^{-1})C(z^{-1})r(k+d) - F(z^{-1})y(k)}{B(z^{-1})E(z^{-1}) + C(z^{-1})Q(z^{-1})}. \quad (6.79)$$

Observe que a única mudança na lei é que se está calculando a variação de controle $\Delta u(k)$. No entanto, na implementação do controlador, o sinal que é enviado ao processo passa primeiro por um integrador discreto, $1/\Delta$, tal que o sinal de controle aplicado na entrada do processo é dado por

$$u(k) = u(k-1) + \Delta u(k), \quad (6.80)$$

onde $\Delta u(k)$ é o incremento de controle ou a variação de controle.

Deve-se ter cuidado ao optar pelo projeto na forma de velocidade quando o processo que se quer controlar for do tipo-1, isto é, já possuir um integrador em malha aberta. Em alguns casos, mesmo quando o processo não for do tipo-1, mas existir um polo discreto real de dinâmica muito lenta e cujo polo correspondente se aproxime de 1 no plano z , o uso do GMVC baseado em modelo na forma de velocidade por acarretar grande perda de desempenho de malha fechada no que se refere ao tempo de estabilização no seguimento de referência do tipo degrau.

Para melhor ilustrarmos o que foi discutido até então nesta seção

sobre GMVC, seguiremos aos projetos de ordem mínima usando modelos na forma posicional e na forma de velocidade, também chamado de GMVC incremental já que utiliza o incremento de controle.

6.6.1 GMVC de ordem mínima: caso posicional

No GMVC de ordem mínima, no caso posicional, a saída generalizada é simplificada ao se considerar $P(z^{-1}) = T(z^{-1}) = 1$ e $Q(z^{-1}) = q_0$ para plantas integradoras (tipo-1) e $Q(z^{-1}) = q_0(1 - z^{-1})$ para não integradoras (tipo-0), para que seja possível garantir o seguimento de referência do tipo degrau (caso servo) com erro nulo em estado estacionário.

Ressalta-se, no entanto, que para a garantia do erro nulo no caso servo, é imprescindível que a estratégia adaptativa auto-ajustável seja utilizada, pois é a etapa de estimação paramétrica que garante a percepção de mudanças de ganho estático do processo controlado. Caso não seja possível adotar GMVC posicional auto-ajustável, haverá erro no seguimento de referências constantes em regime permanente.

O problema de otimização é descrito como segue:

$$0 = \frac{\partial J_{GMV}}{\partial u(k)} = \frac{\partial}{\partial u(k)} [\hat{y}(k+d|k) - r(k+d) + Q(z^{-1})u(k)]^2, \quad (6.81)$$

sujeito ao modelo na estrutura ARMAX ou posicional, dado por

$$A(z^{-1})y(k) = B(z^{-1})u(k-d) + C(z^{-1})w(k). \quad (6.82)$$

O problema de otimização do GMVC, nesse caso, considera apenas um fator de ponderação do esforço de controle, q_0 . Este fator $q_0 \in \mathbb{R}$ modifica o grau de prioridade da minimização de $u^2(k)$ em relação à minimização de $[\hat{y}(k+d|k) - r(k+d)]^2$. Como $P(z^{-1}) = T(z^{-1}) = 1$, então, qualquer valor de $q_0 > 1$ já priorizará, relativamente, que o esforço de controle tenha prioridade no problema de otimização de variância mínima.

Também, quando $q_0 = 0$ retorna-se ao caso de variância mínima como proposto por Åström (1970), onde não se considerava qualquer economia do ponto de vista da ação de controle, priorizando-se totalmente a minimização de $[\hat{y}(k+d|k) - r(k+d)]^2$ em detrimento de $u^2(k)$. Na maioria dos casos de sistemas reais, o controlador obtido para quando $q_0 = 0$, não é fisicamente realizável, pois as amplitudes de $u(k)$ não são realizáveis por nenhum atuador do mundo real. No en-

tanto, $q_0 = 0$ permite que se investigue qual a solução limite em termos de minimização da variância da saída controlada.

Em algumas aplicações pode ser necessário que $q_0 < 0$ seja utilizado, por exemplo quando o processo controlado possuir ganho estático negativo. Isto é, para quando uma entrada positiva constante acarretar saída negativa em estado estacionário.

A escolha do fator de ponderação do esforço de controle, q_0 , é comumente realizada via tentativa e erro pelo projetista. Destaca-se, no entanto, que algoritmos de busca ou métodos de aprendizado por reforço podem ser empregados para automatizar a seleção de q_0 pelo uso de esforço computacional via sintonia supervisionada com base em alguma condição de desempenho ou de estabilidade. Por exemplo, pode-se varrer um conjunto de valores de q_0 enquanto busca-se por um determinado valor de tempo de assentamento da saída controlada, ou mesmo buscando um determinado valor de margem de fase de maneira a garantir robustez à malha de controle.

De uma maneira geral a seleção de q_0 acaba sendo bastante simples quando a classe de sistema que se quer controlar é, de fato, perfeitamente adequada ao caso GMVC posicional. Dito isto, entenda que tal adequação é referente ao seguinte teste: se com poucas tentativas de q_0 não for possível encontrar um resultado satisfatório, então, é muito provável que o GMVC posicional de ordem mínima não seja a melhor opção para o problema de controle em questão. Desse modo, se o uso da estratégia de variância mínima for imprescindível, então deve-se considerar o caso geral descrito pelos polinômios $Q(z^{-1})$, $P(z^{-1})$, $T(z^{-1})$, que resolvem o problema de controle do ponto de vista determinístico para que seja possível elaborar a estratégia GMVC de controle estocástico correspondente.

A equação Diofantina, no caso posicional, assume a forma

$$C(z^{-1}) = z^{-d}F(z^{-1}) + A(z^{-1})E(z^{-1}) \quad (6.83)$$

e o grau dos polinômios $E(z^{-1})$ e $F(z^{-1})$ são, respectivamente,

$$\begin{aligned} n_e &= d - 1, \\ n_f &= n_a - 1, \end{aligned} \quad (6.84)$$

para quando $n_a \geq n_c$. Observe que houve uma simplificação em relação ao caso geral mostrado em (6.69), onde antes se tinha $n_f = \max[(n_p + n_c), (n_a + n_e)] - d$.

A lei de controle GMVC, no caso posicional, se modifica para a

seguinte forma:

$$u(k) = \frac{C(z^{-1})r(k+d) - F(z^{-1})y(k)}{B(z^{-1})E(z^{-1}) + C(z^{-1})Q(z^{-1})}, \quad (6.85)$$

sendo $Q(z^{-1}) = q_0$ para planta integradora ou $Q(z^{-1}) = q_0(1 - z^{-1})$ para planta não integradora.

O caso posicional carece de um filtro de ponderação de saída, já que $P(z^{-1}) = 1$. Portanto, é preciso redobrar o cuidado com a implementação adaptativa do GMVC no que se refere à estimação de $C(z^{-1})$, pois o Preditor de Variância Mínima no caso posicional assume a seguinte forma:

$$\hat{y}(k+d|k) = \frac{B(z^{-1})E(z^{-1})u(k) + F(z^{-1})y(k)}{C(z^{-1})}. \quad (6.86)$$

Observe que, apesar de não ser necessário implementar explicitamente este Preditor para se realizar no software de controle a lei de controle GMVC, implicitamente o Preditor de Variância Mínima está sendo utilizado e por isto as raízes de $C(z^{-1})$ precisam estar contidas no círculo unitário do plano z .

Uma simplificação adicional ao GMVC posicional e que retira às demandas de análise de estabilidade em relação ao polinômio $C(z^{-1})$ é a adoção do projeto baseado em modelo ARX, isto é, com $C(z^{-1}) = 1$.

$$A(z^{-1})y(k) = B(z^{-1})u(k-d) + w(k). \quad (6.87)$$

Nesse caso assume-se que de fato não há um processo estocástico que exija a modelagem ARMAX, caracterizando que a perturbação proveniente de $w(k)$ é do tipo ruído branco.

Exemplo 6.5 *GMVC posicional aplicado no controle de tensão de um processo eletrônico subamortecido do tipo-0. Portanto, $Q(z^{-1}) = q_0(1 - z^{-1})$.*

Neste exemplo utilizaremos o GMVC de forma adaptativa indireta ou auto-ajustável indireta (indirect self-tuning). Para isto, o estimador dos Mínimos Quadrados Recursivo será utilizado para estimar um modelo ARMAX de segunda ordem do processo controlado, em tempo real, enquanto o GMVC é adaptado com as inovações desse modelo estimado. O fator de ponderação do esforço de controle q_0 será selecionado por tentativa e erro ao longo do projeto e experimentos.

Os parâmetros do modelo do sistema são desconhecidos, a priori, mas uma estrutura ARMAX é sugerida para ser utilizada, sendo esta

descrita por:

$$\begin{aligned} (1 + a_1 z^{-1} + a_2 z^{-2}) y(k) &= (b_0 + b_1 z^{-1}) u(k-1) \\ &+ (1 + c_1 z^{-1} + c_2 z^{-2}) w(k). \end{aligned} \quad (6.88)$$

Os polinômios $E(z^{-1})$ e $F(z^{-1})$ têm grau $n_e = d - 1 = 0$ e $n_f = n_a - 1 = 1$, respectivamente. Portanto, a equação Diofantina que deve ser resolvida é a seguinte:

$$1 + c_1 z^{-1} + c_2 z^{-2} = z^{-1} (f_0 + f_1 z^{-1}) + (1 + a_1 z^{-1} + a_2 z^{-2}) e_0. \quad (6.89)$$

A solução da Diofantina fornece a seguinte atribuição aos parâmetros do GMVC com base no modelo do sistema:

$$\begin{aligned} e_0 &= 1 \\ f_0 &= c_1 - a_1 \\ f_1 &= c_2 - a_2 \end{aligned} \quad (6.90)$$

O sistema GMVC projetado é então implementado com a seguinte lei de controle:

$$\begin{aligned} u(k) &= \frac{(1 + c_1 z^{-1} + c_2 z^{-2}) r(k+1) - (f_0 + f_1 z^{-1}) y(k)}{(b_0 + b_1 z^{-1}) e_0 + (1 + c_1 z^{-1} + c_2 z^{-2}) q_0 (1 - z^{-1})} \\ u(k) &= \frac{(1 + c_1 z^{-1} + c_2 z^{-2}) r(k+1) - (f_0 + f_1 z^{-1}) y(k)}{(b_0 + q_0) + (b_1 + q_0 c_1 - q_0) z^{-1} + (q_0 c_2 - q_0 c_1) z^{-2} + (-q_0 c_2) z^{-3}} \end{aligned} \quad (6.91)$$

Reescrevendo na forma de equação a diferenças obtém-se:

$$\begin{aligned} u(k) &= \frac{1}{(b_0 + q_0)} [r(k+1) + c_1 r(k) + c_2 r(k-1) - f_0 y(k) - f_1 y(k-1) \\ &- (b_1 + q_0 c_1 - q_0) u(k-1) - (q_0 c_2 - q_0 c_1) u(k-2) - (-q_0 c_2) u(k-3)] \end{aligned} \quad (6.92)$$

O estimador MQR para este problema de estimação assume a

seguinte estrutura para a saída estimada do modelo:

$$\hat{y}_{mqr}(k) = \underbrace{\begin{bmatrix} -y_{k-1} & -y_{k-2} & u_{k-1} & u_{k-2} & w_{k-1} & w_{k-2} \end{bmatrix}}_{\phi_{mqr}^T(k)} \underbrace{\begin{bmatrix} \hat{a}_1 \\ \hat{a}_2 \\ \hat{b}_0 \\ \hat{b}_1 \\ \hat{c}_1 \\ \hat{c}_2 \end{bmatrix}}_{\hat{\theta}_{mqr}(k)} \quad (6.93)$$

$$\begin{aligned} w(k) &= y(k) - \hat{y}_{mqr}(k) \\ L(k) &= P(k)\phi_{mqr}(k)[1 + \phi_{mqr}^T(k)P(k)\phi_{mqr}(k)]^{-1} \\ \hat{\theta}_{mqr}(k+1) &= \hat{\theta}_{mqr}(k) + L(k)w(k) \\ P(k+1) &= P(k) - L(k)\phi_{mqr}^T(k)P(k) \end{aligned} \quad (6.94)$$

A realização deste experimento foi feita utilizando-se o sistema de aquisição de dados e controle desenvolvido no LACOS, chamado DaqDuino, disponível no sítio, lacos.ufpa.br, na Internet. O período de amostragem utilizado foi $T_s = 0,05$ s, sendo o estimador MQR iniciado com $P(0) = 100I_{6 \times 6}$ e com $\hat{\theta}(0) = \begin{bmatrix} 1 & \cdots & 1 \end{bmatrix}_{1 \times 6}^T$.

Os resultados para os testes reais de seguimento de referência usando-se q_0 igual a 4, 8 e 12, são apresentados nas figuras 67 a 69, respectivamente. Realizou-se também um teste de seguimento de referência senoidal usando-se $q_0 = 4$, como se mostra na Figura 70.

Para que seja possível reproduzir este exemplo experimental, mas utilizando-se de simulação, a seguir são fornecidos os parâmetros de um modelo linear e invariante no tempo que foi identificado durante a execução do projeto com o processo real:

$$\begin{aligned} A(z^{-1}) &= 1 - 1,2356881z^{-1} + 0,4526202z^{-2} \\ B(z^{-1}) &= 0,0101615 + 0,1947393z^{-1} \\ C(z^{-1}) &= 1 - 0,0398287z^{-1} + 0,0825432z^{-2} \end{aligned} \quad (6.95)$$

E para simular o ruído Gaussiano, $w(k)$, é possível utilizar a variância estimada para o mesmo durante a identificação do modelo do sistema, sendo esta: $\sigma_w^2 = 0,0068327$.

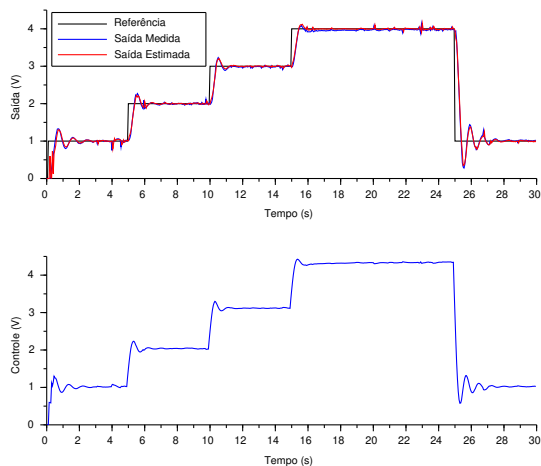


Figura 67 – Teste experimental com GMVC posicional de ordem mínima, com $q_0 = 4$, no controle de tensão de um sistema eletrônico subamortecido.

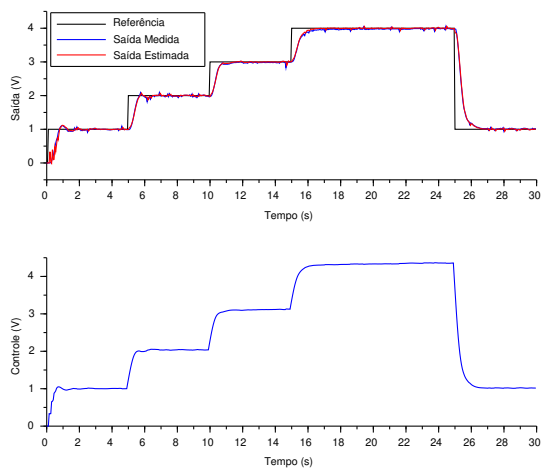


Figura 68 – Teste experimental com GMVC posicional de ordem mínima, com $q_0 = 8$, no controle de tensão de um sistema eletrônico subamortecido.

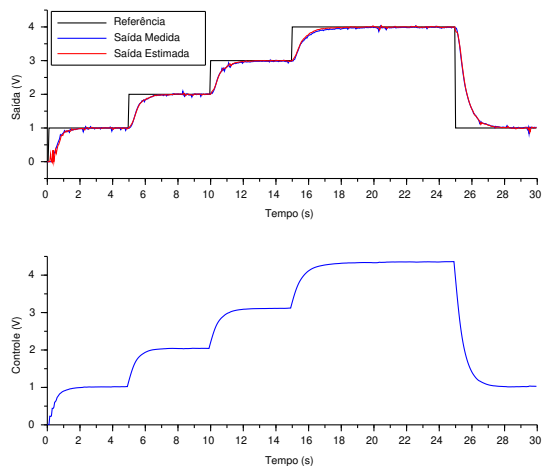


Figura 69 – Teste experimental com GMVC posicional de ordem mínima, com $q_0 = 12$, no controle de tensão de um sistema eletrônico subamortecido.

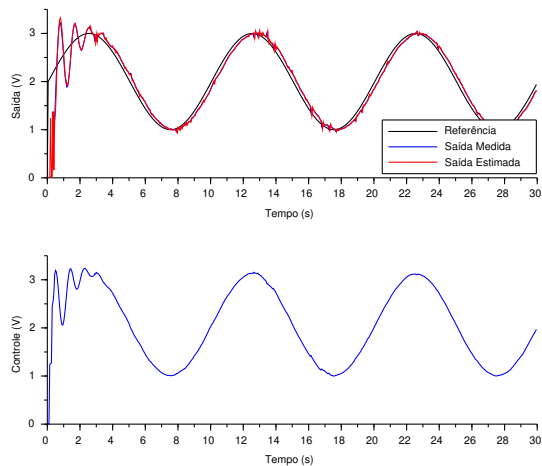


Figura 70 – Teste experimental com GMVC posicional de ordem mínima, com $q_0 = 4$, no controle de tensão de um sistema eletrônico subamortecido usando referência senoidal.

O código, em linguagem Scilab, que reproduz este experimento, mas usando o modelo para simulação, é mostrado na Tabela 23.

Tabela 23 – Código em Scilab para reproduzir o Exemplo 6.5.

```
// GMVC Posicional adaptativo do tipo auto-ajustável indireto
clear; xdel(winsid()); c1c;

// Modelo ARMAX do processo eletrônico subamortecido
Ts = 0.05; // Período de amostragem
Az = [1 -1.2356881 0.4526202];
Bz = [0.0101615 0.1947393];
Cz = [1 -0.0398287 0.0825432];
Var_w = 0.0068327; // Variância do ruído do processo, w(k)

// Sintonia GMVC
q0 = 8;

// Simulação/Experimentação
tfinal = 30; // segundos
N = round(tfinal/Ts); // número de amostras
t = 0:Ts:N*Ts-Ts; // vetor de tempo para os plots

// Sinal de referência
r(1:5)=0; r(6:N/6)=1; r(N/6+1:2*N/6)=2; r(2*N/6+1:3*N/6)=3;
r(3*N/6+1:4*N/6)=4; r(4*N/6+1:5*N/6)=4; r(5*N/6+1:6*N/6+1)=2;
//r=[0 2*sin(2*pi*0.1*t)];

// condições iniciais
y(1:3)=0; u(1:3)=0; we(1:3)=0;
theta = ones(6,1); P = 100*eye(6,6);

// Ruído simulado do processo
w = 0*grand(N, 1, "nor", 0, sqrt(Var_w));

for k=4:N
    y(k)=-Az(2)*y(k-1)-Az(3)*y(k-2)+Bz(1)*u(k-1) ...
        +Bz(2)*u(k-2)+w(k)+Cz(2)*u(k-1)+Cz(3)*u(k-2);

    // Estimador MQR
    phi_mqr = [-y(k-1) -y(k-2) u(k-1) u(k-2) we(k-1) we(k-2)]';
    y_mqr(k) = phi_mqr'*theta;
    we(k) = y(k)-y_mqr(k); // Erro de estimação
    L = P*phi_mqr*inv(1+phi_mqr'*P*phi_mqr);
    theta = theta +L*we(k);
    P = P -L*phi_mqr'*P;

    // Atualização dos parâmetros do modelo e do controlador
    a1 = theta(1); a2 = theta(2);
    b0 = theta(3); b1 = theta(4);
    c1 = theta(5); c2 = theta(6);

    f0 = c1-a1; f1 = c2-a2;

    // GMVC
    u(k)=(1/(b0+q0))*(r(k+1)+c1*r(k)+c2*r(k-1) ...
        -f0*y(k)-f1*y(k-1) ...
        -(b1+q0*c1-q0)*u(k-1)-(q0*c2-q0*c1)*u(k-2)-(-q0*c2)*u(k-3));

    // Saturação do sinal de controle
    if u(k)>5
        u(k)=5;
    elseif u(k)<0
        u(k)=0;
    end

end

fig2=scf();
subplot(211)
plot2d(t,r(1:N),1); plot2d(t,y,2); plot2d(t,y_mqr,5);
ylabel('Saída (V)'); xlabel('Tempo (s)');
legend('Referência','Saída Medida','Saída Estimada',2);
subplot(212)
plot2d(t,u,2); ylabel('Controle (V)'); xlabel('Tempo (s)');
```

6.6.2 GMVC de ordem mínima: caso incremental

O GMVC de ordem mínima, no caso incremental, se baseia em $P(z^{-1}) = T(z^{-1}) = 1$ e $Q(z^{-1}) = q_0$, como no caso posicional, mas o modelo do processo é aumentado à forma de velocidade,

$$\Delta A(z^{-1})y(k) = B(z^{-1})\Delta u(k-d) + C(z^{-1})w(k). \quad (6.96)$$

Nessa forma o projeto se baseia em determinar a variação do sinal de controle, $\Delta u(k) = u(k) - u(k-1)$, que minimiza a função custo

$$J_{GMV} = [\hat{y}(k+d|k) - r(k+d) + Q(z^{-1})\Delta u(k)]^2. \quad (6.97)$$

No GMVC incremental o operador de diferença discreta, $\Delta = 1 - z^{-1}$, não é incluído de maneira forçada no polinômio $Q(z^{-1})$. Ao invés disso, o que se faz é a consideração de que o processo é descrito por uma estrutura ARMAX incremental, chamada de ARIMAX, equivalente à forma descrita anteriormente em (6.96), mas onde se considera que o processo Gaussiano é incremental, pois

$$A(z^{-1})y(k) = B(z^{-1})u(k-d) + C(z^{-1})\frac{w(k)}{\Delta} \quad (6.98)$$

onde $w(k)/\Delta$ é quem caracteriza essa propriedade desse modelo.

Esta forma de velocidade do modelo de projeto, também chamada de ARIMAX ou CARIMA (*controlled auto-regressive integrated moving average*), é bastante utilizada por diversas estratégias MPC, como no Controle por Matriz Dinâmica (DMC) e no Controle Preditivo Generalizado (GPC). É importante frisar que não se trata de nenhuma modificação na forma de realizar o procedimento de identificação do modelo de projeto fazendo-se de fato a utilização da estrutura ARIMAX. Na maioria dos casos a identificação é mesmo aplicada usando-se a estrutura ARMAX, sendo esta, posteriormente, aumentada à forma ARIMAX/CARIMA. A justificativa é que tal estrutura aumentada fornece, de maneira natural, um método de projetar MPCs com ação integral, já que resolve-se a solução ótima para $\Delta u(k)$, mas aplica-se o incremento do controle à entrada do processo, isto é,

$$u(k) = u(k-1) + \Delta u(k). \quad (6.99)$$

Ou seja, a variação de controle $\Delta u(k)$ passa por um sistema integrador discreto $1/\Delta$.

Com a forma aumentada do modelo de projeto modifica-se a equação Diofantina para

$$C(z^{-1}) = z^{-d}F(z^{-1}) + \Delta A(z^{-1})E(z^{-1}), \quad (6.100)$$

fazendo com que o grau do polinômio $F(z^{-1})$ se modifique para

$$n_f = n_a \quad (6.101)$$

enquanto o do polinômio $E(z^{-1})$ permanece $n_e = d - 1$.

A lei referente à variação da ação de controle é dada por:

$$\Delta u(k) = \frac{C(z^{-1})r(k+d) - F(z^{-1})y(k)}{B(z^{-1})E(z^{-1}) + C(z^{-1})q_0}. \quad (6.102)$$

É importante destacar que o Preditor de Variância Mínima no caso incremental e de ordem mínima é equivalente, estruturalmente, ao caso posicional mostrado em (6.86). A diferença ocorre na maneira como o polinômio $F(z^{-1})$ tem seus parâmetros atribuídos, que se modifica pela participação do polinômio aumentado $\Delta A(z^{-1})$.

Exemplo 6.6 *GMVC incremental aplicado no controle de tensão de um processo eletrônico subamortecido do tipo-0.*

Este exemplo é similar ao Exemplo 6.5, para o caso posicional. No entanto, o projeto incremental possui características de robustez interessantes devido a inclusão do integrador na entrada do processo controlado. Tal procedimento, no entanto, acarreta perda de desempenho em termos de velocidade de resposta, já que o integrador acarreta defasamento de 90° nas baixas frequências, já que $z = e^{j\omega T_s}$ tende a 1 quando $\omega \rightarrow 0$ rad/s, fazendo o ganho tender ao infinito nas baixas frequências e a fase tender a -90° , conforme discutiu-se no Capítulo 2 nas seções referentes aos compensadores avanço e atraso e estabilidade relativa com base nas margens de ganho e de fase.

O modelo nominal usado no caso posicional do Exemplo 6.5 é aumentado à seguinte forma:

$$\bar{A}(z)y(k) = B(z)\Delta u(k-d) + C(z)w(k), \quad (6.103)$$

onde $\Delta u(k) = u(k) - u(k-1)$ e

$$\bar{A}(z) = (1 - z^{-1})(1 + a_1 z^{-1} + a_2 z^{-2}) = 1 + \bar{a}_1 z^{-1} + \bar{a}_2 z^{-2} + \bar{a}_3 z^{-3}. \quad (6.104)$$

A solução da equação Diofantina no caso Incremental é dada por:

$$C(z) = z^{-d}F(z) + \bar{A}(z)E(z), \quad (6.105)$$

$$\begin{aligned} n_e &= d - 1, \\ n_f &= n_{\bar{a}} - 1 = n_a, \end{aligned} \quad (6.106)$$

$$\begin{aligned} E(z) &= e_0, \\ F(z) &= f_0 + f_1 z^{-1} + f_2 z^{-2}, \end{aligned} \quad (6.107)$$

$$\begin{aligned} (1 + c_1 z^{-1} + c_2 z^{-2}) &= z^{-1} (f_0 + f_1 z^{-1} + f_2 z^{-2}) \\ &+ (1 + \bar{a}_1 z^{-1} + \bar{a}_2 z^{-2} + \bar{a}_3 z^{-3}) e_0, \end{aligned} \quad (6.108)$$

$$\begin{aligned} e_0 &= 1, \\ f_0 &= c_1 - \bar{a}_1, \\ f_1 &= c_2 - \bar{a}_2, \\ f_2 &= -\bar{a}_3. \end{aligned} \quad (6.109)$$

A lei de controle desta síntese incremental é então definida por:

$$\Delta u(k) = \frac{C(z)r(k+d) - F(z)y(k)}{B(z)E(z) + C(z)q_0}, \quad (6.110)$$

$$\Delta u(k) = \frac{(1 + c_1 z^{-1} + c_2 z^{-2}) r(k+1) - (f_0 + f_1 z^{-1} + f_2 z^{-2}) y(k)}{(b_0 + q_0) + (b_1 + c_1 q_0) z^{-1} + (c_2 q_0) z^{-2}}. \quad (6.111)$$

Aplicando-se a Transformada z inversa sobre esta lei de controle obtém-se a equação a diferenças que implementa este controlador:

$$\begin{aligned} \Delta u(k) &= \frac{1}{(b_0 + q_0)} [r(k+1) + c_1 r(k) + c_2 r(k-1) - f_0 y(k) - f_1 y(k-1) \\ &- f_2 y(k-2) - (b_1 + c_1 q_0) \Delta u(k-1) - (c_2 q_0) \Delta u(k-2)], \end{aligned} \quad (6.112)$$

onde a variação de controle $\Delta u(k)$ ainda deve passar pela ação incremental fazendo-se a seguinte operação para que seja aplicado o sinal $u(k)$ na entrada do processo controlado:

$$u(k) = u(k-1) + \Delta u(k). \quad (6.113)$$

Os resultados experimentais sobre o processo eletrônico subamortecido, para o caso de seguimento de referência em diferentes pontos de operação, são mostrados nas figuras 72 a 74. Estes testes são respectivos à sintonia de q_0 nos seguintes valores: 4, 8, 12 e 8.

Esses resultados são similares aos observados no Exemplo 6.5, do caso GMVC posicional, mas é possível notar algumas diferenças em relação a este novo caso incremental que tem a saída controlada convergindo com maior precisão ao sinal de referência, enquanto que no caso posicional a saída estimada convergia mais rapidamente à referência que a própria saída medida, pois o erro nulo em estado estacionário somente era possível quando o estimador MQR conseguia determinar precisamente o ganho estático do processo controlado. Já no projeto incremental, o seguimento de referências constantes é garantido pela presença da ação integral discreta ou ação incremental.

É interessante observar a diferença, também, na Fig. 71, quando se usou $q_0 = 4$, pois há maior oscilação da saída controlada durante a transição de referência. Tais oscilações são maiores se comparadas as do caso posicional da Fig. 67. Isto é uma desvantagem da ação incremental deste novo projeto, que pode acarretar problemas de ciclos limite oscilatórios em torno da referência no caso de controle de processos do tipo-1 (processos integradores).

O código, em linguagem Scilab, que reproduz este experimento, mas usando o modelo para simulação, é mostrado na Tabela 24.

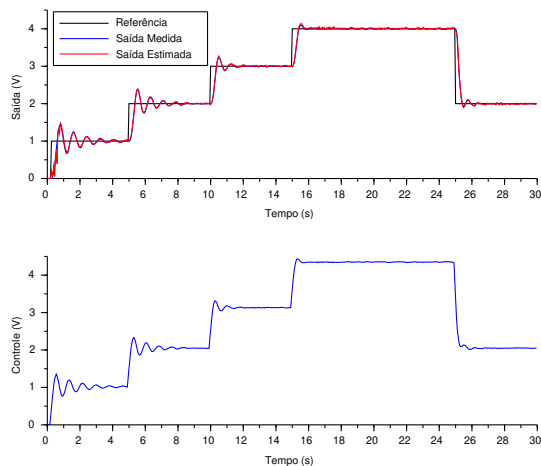


Figura 71 – Teste experimental com GMVC incremental de ordem mínima, com $q_0 = 4$, no controle de tensão de um sistema eletrônico subamortecido.

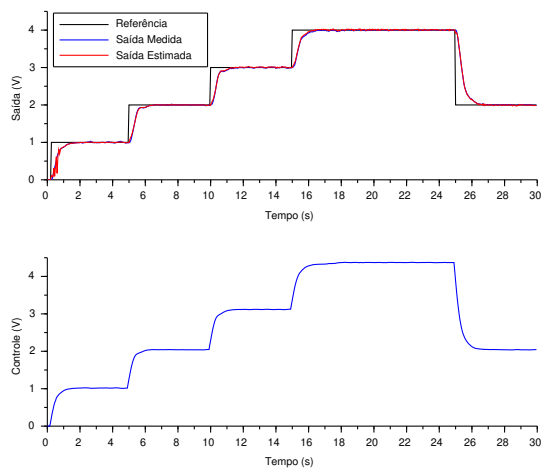


Figura 72 – Teste experimental com GMVC incremental de ordem mínima, com $q_0 = 8$, no controle de tensão de um sistema eletrônico subamortecido.

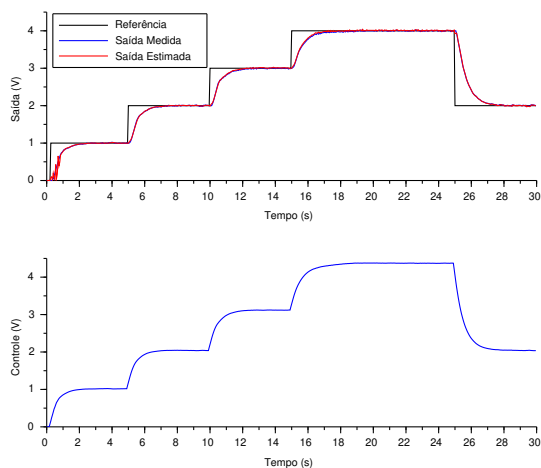


Figura 73 – Teste experimental com GMVC incremental de ordem mínima, com $q_0 = 12$, no controle de tensão de um sistema eletrônico subamortecido.

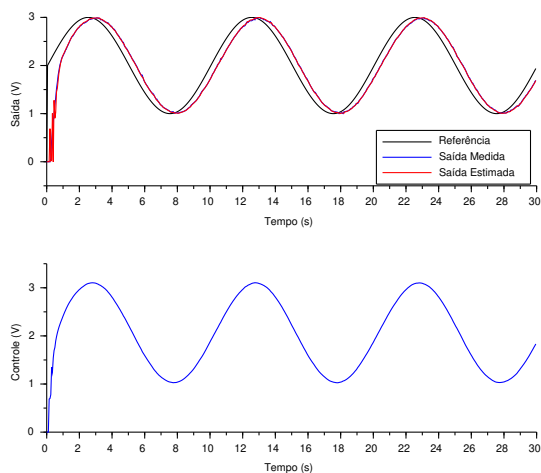


Figura 74 – Teste experimental com GMVC incremental de ordem mínima, com $q_0 = 8$, no controle de tensão de um sistema eletrônico subamortecido usando referência senoidal.

Tabela 24 – Código em Scilab para reproduzir o Exemplo 6.6.

```

// GMVC Incremental adaptativo do tipo auto-ajustável indireto
clear; xdel(winsid()); clc;

// Modelo ARMAX do processo eletrônico subamortecido
Ts = 0.05; // Período de amostragem
Az = [1 -1.2356881 0.4526202];
Bz = [0.0101615 0.1947393];
Cz = [1 -0.0398287 0.0825432];
Var_w = 0.0068327; // Variância do ruído do processo, w(k)

// Sintonia GMVC
q0 = 8;

// Simulação/Experimentação
tfinal = 30; // segundos
N = round(tfinal/Ts); // número de amostras
t = 0:Ts:N*Ts-Ts; // vetor de tempo para os plots

// Sinal de referência
r(1:5)=0; r(6:N/6)=1; r(N/6+1:2*N/6)=2; r(2*N/6+1:3*N/6)=3;
r(3*N/6+1:4*N/6)=4; r(4*N/6+1:5*N/6)=4; r(5*N/6+1:6*N/6+1)=2;
//r=[0 2*sin(2*pi*0.1*t)];

// condições iniciais
y(1:3)=0; u(1:3)=0; we(1:3)=0; du(1:3)=0;
theta = ones(6,1); P = 100*eye(6,6);

// Ruído simulado do processo
w = 1*grand(N, 1, "nor", 0, sqrt(Var_w));

for k=4:N
    y(k)=-Az(2)*y(k-1)-Az(3)*y(k-2)+Bz(1)*u(k-1) ...
        +Bz(2)*u(k-2)+w(k)+Cz(2)*w(k-1)+Cz(3)*w(k-2);

    // Estimador MQR
    phi_mqr = [-y(k-1) -y(k-2) u(k-1) u(k-2) we(k-1) we(k-2)]';
    y_mqr(k) = phi_mqr'*theta;
    we(k) = y(k)-y_mqr(k); // Erro de estimação
    L = P*phi_mqr*inv(1+phi_mqr'*P*phi_mqr);
    theta = theta +L*we(k);
    P = P -L*phi_mqr'*P;

    // Atualização dos parâmetros do modelo e do controlador
    a1 = theta(1); a2 = theta(2);
    b0 = theta(3); b1 = theta(4);
    c1 = theta(5); c2 = theta(6);

    // GMVC
    // Aumentando o modelo de projeto à forma de velocidade
    Abarz = conv([1 -1],[1 a1 a2]);
    a1bar = Abarz(2);
    a2bar = Abarz(3);
    a3bar = Abarz(4);

    f0 = c1-a1bar;
    f1 = c2-a2bar;
    f2 = -a3bar;

    du(k)=(1/(b0+q0))*(r(k+1)+c1*r(k)+c2*r(k-1) ...
        -f0*y(k)-f1*y(k-1)-f2*y(k-2) ...
        -(b1+q0*c1)*du(k-1)-(q0*c2)*du(k-2));

    // Passando o sinal du(k) pelo bloco incremental
    u(k) = u(k-1) +du(k);

    // Saturação do sinal de controle
    if u(k)>5
        u(k)=5;
    elseif u(k)<0
        u(k)=0;
    end
end
end

```

```

fig2=scf();
subplot(211)
plot2d(t,r(1:N),1); plot2d(t,y,2); plot2d(t,y_mqr,5);
ylabel('Saída (V)'); xlabel('Tempo (s)');
legend('Referência','Saída Medida','Saída Estimada',2);
subplot(212)
plot2d(t,u,2); ylabel('Controle (V)'); xlabel('Tempo (s)');

```

6.7 GMVC NO ESPAÇO DE ESTADO

O GMVC no espaço de estado, ou GMVSS, foi proposto por Silveira e Coelho (2011) como uma forma de projetar o GMVC de ordem mínima sem a necessidade de se resolver a equação Diofantina

$$C(z^{-1}) = z^{-d}F(z^{-1}) + A(z^{-1})E(z^{-1}). \quad (6.114)$$

Evitando-se a Diofantina o método GMVSS permitiu a solução para qualquer valor de atraso d , podendo-se até mesmo tratar, no caso adaptativo, d variante no tempo.

O método GMVSS consiste em reescrever o modelo ARMAX (ou ARIMAX) em uma forma canônica observável no espaço de estado, que permite resolver o problema de predição com base no Filtro de Kalman.

O modelo em espaço de estado utilizado no método GMVSS é descrito na seguinte forma:

$$x(k) = Ax(k-1) + Bu(k-d) + \Gamma w(k-1) \quad (6.115)$$

$$y(k) = Cx(k) + w(k), \quad (6.116)$$

em que $x(k)$ é o vetor de estado do processo de n_a variáveis de estado, $u(k)$ é a entrada do processo, $w(k)$ é o ruído do processo e $y(k)$ é a saída medida. As matrizes A , B , Γ , C , são, respectivamente: a matriz dinâmica invertível (não singular) de dimensão $n_a \times n_a$, a matriz de entrada de controle, a matriz de entrada do ruído de incertezas e a matriz de saída. O ruído $w(k)$ tem média nula e variância σ_w^2 .

As matrizes do modelo em espaço de estado são obtidas a partir da equação a diferenças do modelo ARMAX, em que, neste primeiro momento, assumiremos que $n_a = n_b = n_c$, tal que

$$\begin{aligned}
y(k) = & -a_1y(k-1) - \dots - a_{n_a}y(k-n_a) \\
& + b_0u(k-d) + b_1u(k-d-1) + \dots + b_{n_b}u(k-d-n_b) \\
& + w(k) + c_1w(k-1) + \dots + c_{n_c}w(k-n_c).
\end{aligned} \quad (6.117)$$

Assumindo que a variável de estado $x_1(k)$ é dada por

$$x_1(k) = y(k) - w(k), \quad (6.118)$$

pode-se reescrever o modelo ARMAX com base nas seguintes n_a variáveis de estado:

$$\begin{aligned} x_1(k) &= -a_1 x_1(k-1) + x_2(k-1) + b_0 u(k-d) + (c_1 - a_1) w(k-1), \\ x_2(k) &= -a_2 x_1(k-1) + x_3(k-1) + b_1 u(k-d) + (c_1 - a_1) w(k-1), \\ &\vdots \\ x_{n_a}(k) &= -a_{n_a} x_1(k-1) + b_{n_b} u(k-d) + (c_{n_c} - a_{n_a}) w(k-1). \end{aligned} \quad (6.119)$$

Destas equações a diferenças pode-se verificar por inspeção ou aplicando-se o cálculo das matrizes Jacobianas, que o sistema ARMAX é descrito na forma canônica observável no espaço de estado com base nas seguintes matrizes:

$$A = \begin{bmatrix} -a_1 & 1 & 0 & 0 \\ -a_2 & 0 & \ddots & 0 \\ \vdots & 0 & 0 & 1 \\ -a_{n_a} & 0 & \cdots & 0 \end{bmatrix}_{n_a \times n_a}, \quad B = \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_{n_b} \end{bmatrix}_{n_a \times 1} \quad (6.120)$$

$$\Gamma = \begin{bmatrix} c_1 - a_1 \\ c_2 - a_2 \\ \vdots \\ c_{n_c} - a_{n_a} \end{bmatrix}_{n_a \times 1}, \quad C = [1 \quad 0 \quad \cdots \quad 0]_{1 \times n_a} \quad (6.121)$$

Relembrando que o problema de controle GMVC, no caso de ordem mínima, minimiza a função custo

$$J_{GMV} = [\hat{y}(k+d|k) - r(k+d) + q_0 u(k)]^2. \quad (6.122)$$

Desse modo, o problema de prever a saída do modelo de estado fornece o seguinte problema:

$$y(k+d) = Cx(k+d) + w(k+d), \quad (6.123)$$

onde não há como saber qual o valor de $w(k+d)$ devido a sua natureza aleatória. Há também a interferência de valores futuros de $w(k)$ em

$x(k+d)$ quando $d > 1$, como se mostra a seguir:

$$\begin{aligned}
 x(k) &= Ax(k-1) + Bu(k-d) + \Gamma w(k-1) \\
 x(k+1) &= Ax(k) + Bu(k-d+1) + \Gamma w(k) \\
 x(k+2) &= A^2x(k) + ABu(k-d+1) + A\Gamma w(k) \\
 &\quad + Bu(k-d+2) + \Gamma w(k+1) \\
 &\vdots \\
 x(k+d) &= A^dx(k) + \sum_{i=1}^d A^{(d-i)}Bu(k-d+i) \\
 &\quad + \sum_{i=1}^d A^{(d-i)}\Gamma w(k-1+i)
 \end{aligned} \tag{6.124}$$

Substituindo $x(k+d)$ na saída predita em (6.123) obtém-se

$$\begin{aligned}
 y(k+d) &= CA^dx(k) + \sum_{i=1}^d CA^{(d-i)}Bu(k-d+i) \\
 &\quad + \sum_{i=1}^d CA^{(d-i)}\Gamma w(k-1+i) + w(k+d).
 \end{aligned} \tag{6.125}$$

Na equação (6.125), além do problema dos termos futuros e aleatórios de $w(k)$, há também o desconhecimento sobre $x(k)$. No entanto, o estimador ótimo de $x(k)$, no caso de variância mínima para o sistema ARMAX descrito na forma canônica observável mostrada em (6.115), é dado pelo Filtro de Kalman (SILVEIRA; COELHO, 2011):

$$\bar{x}(k) = (A - \Gamma C)\bar{x}(k-1) + Bu(k-d) + \Gamma y(k-1). \tag{6.126}$$

Ainda sobre a equação (6.125), o problema de predição relativo aos termos futuros de $w(k)$ é resolvido a partir da separação do termo

presente dos termos futuros, como segue:

$$\begin{aligned} \sum_{i=1}^d CA^{(d-i)}\Gamma w(k-1+i) + w(k+d) &= \underbrace{CA^{(d-1)}\Gamma w(k)}_{\text{Presente}} \\ &+ \underbrace{\sum_{i=2}^d CA^{(d-i)}\Gamma w(k-1+i) + w(k+d)}_{\text{Futuro}}. \end{aligned} \quad (6.127)$$

A predição estimada de $y(k+d)$ com base apenas em termos passados e até o instante atual k é dada por:

$$\hat{y}(k+d|k) = CA^d \bar{x}(k) + \sum_{i=1}^d CA^{(d-i)}Bu(k-d+i) + CA^{(d-1)}\Gamma w(k), \quad (6.128)$$

onde,

$$w(k) = y(k) - C\bar{x}(k), \quad (6.129)$$

resultando, assim, no Preditor de Variância Mínima do GMVSS:

$$\begin{aligned} \hat{y}(k+d|k) &= [CA^d - CA^{(d-1)}\Gamma C] \bar{x}(k) + CA^{(d-1)}\Gamma y(k) \\ &+ \begin{bmatrix} CA^0 B \\ \vdots \\ CA^{(d-1)} B \end{bmatrix}^T \begin{bmatrix} u(k) \\ \vdots \\ u(k-d+1) \end{bmatrix}, \end{aligned} \quad (6.130)$$

relacionado ao Preditor de Estado de Variância Mínima proposto por Silveira e Coelho (2011):

$$\begin{aligned} \hat{x}(k+d|k) &= [A^d - FC] \bar{x}(k) + Fy(k) \\ &+ \begin{bmatrix} A^0 B \\ \vdots \\ A^{(d-1)} B \end{bmatrix}^T \begin{bmatrix} u(k) \\ \vdots \\ u(k-d+1) \end{bmatrix}, \end{aligned} \quad (6.131)$$

onde o ganho do Preditor de Estado de Variância mínima é dado por

$$F = A^{(d-1)}\Gamma. \quad (6.132)$$

A lei de controle via método GMVSS é obtida substituindo-se (6.130) em J_{GMV} e fazendo-se a minimização desta função custo, como segue:

$$\frac{\partial J_{GMV}}{\partial u(k)} = \frac{\partial \left\{ [\hat{y}(k+d|k) - r(k+N_y) + q_0 u(k)]^2 \right\}}{\partial u(k)} = 0, \quad (6.133)$$

que resulta em

$$\begin{aligned} u(k) = & (CB + q_0)^{-1} \{ r(k+d) \\ & - [CA^d - CFC] \bar{x}(k) - CFy(k) \\ & - \left[\begin{array}{c} CAB \\ \vdots \\ CA^{(d-1)}B \end{array} \right]^T \left[\begin{array}{c} u(k-1) \\ \vdots \\ u(k-d+1) \end{array} \right] \} \end{aligned} \quad (6.134)$$

6.7.1 GMVSS: caso incremental

A lei de controle GMVSS mostrada em (6.134) é referente ao caso posicional. Porém, para se obter o caso incremental a mudança se inicia no modelo do sistema, sendo este considerado ARIMAX,

$$\Delta A(z^{-1})y(k) = B(z^{-1})\Delta u(k-d) + C(z^{-1})w(k), \quad (6.135)$$

$$\Delta A(z^{-1}) = 1 + \bar{a}_1 z^{-1} + \dots + \bar{a}_{n_{\bar{a}}} z^{-n_{\bar{a}}}, \quad (6.136)$$

$$n_{\bar{a}} = n_a + 1, \quad (6.137)$$

estabelecendo, assim, um modelo aumentado à forma de velocidade em espaço de estado definido na estrutura canônica observável,

$$x_a(k) = A_a x_a(k-1) + B_a \Delta u(k-d) + \Gamma_a w(k-1), \quad (6.138)$$

$$y_a(k) = C_a x_a(k) + w(k), \quad (6.139)$$

em que as matrizes aumentadas são descritas por

$$\begin{aligned}
 A_a &= \begin{bmatrix} -\bar{a}_1 & 1 & 0 & 0 \\ -\bar{a}_2 & 0 & \ddots & 0 \\ \vdots & 0 & 0 & 1 \\ -\bar{a}_{n_{\bar{a}}} & 0 & \cdots & 0 \end{bmatrix}_{n_{\bar{a}} \times n_{\bar{a}}}, \quad B_a = \begin{bmatrix} b_0 \\ \vdots \\ b_{n_b} \\ 0 \end{bmatrix}_{n_{\bar{a}} \times 1} \\
 \Gamma_a &= \begin{bmatrix} c_1 - \bar{a}_1 \\ \vdots \\ c_{n_c} - \bar{a}_{n_a} \\ -\bar{a}_{n_{\bar{a}}} \end{bmatrix}_{n_{\bar{a}} \times 1}, \quad C_a = [1 \quad 0 \quad \cdots \quad 0]_{1 \times n_{\bar{a}}}
 \end{aligned} \tag{6.140}$$

A lei de controle GMVSS no caso incremental se modifica para a seguinte forma:

$$\begin{aligned}
 \Delta u(k) &= (C_a B_a + q_0)^{-1} \{r(k+d) \\
 &\quad - [C_a A_a^d - C_a F_a C_a] \bar{x}_a(k) - C_a F_a y(k) \\
 &\quad - \begin{bmatrix} C_a A_a B_a \\ \vdots \\ C_a A_a^{(d-1)} B_a \end{bmatrix}^T \begin{bmatrix} \Delta u(k-1) \\ \vdots \\ \Delta u(k-d+1) \end{bmatrix} \}, \tag{6.141}
 \end{aligned}$$

$$u(k) = u(k-1) + \Delta u(k). \tag{6.142}$$

Destaca-se que, no caso aumentado, o ganho do Preditor de Estado de Variância Mínima se modifica para $F_a = A_a^{(d-1)} \Gamma_a$.

O Filtro de Kalman que estima $x_a(k)$ também se modifica à forma aumentada:

$$\bar{x}_a(k) = (A_a - \Gamma_a C_a) \bar{x}_a(k-1) + B_a \Delta u(k-d) + \Gamma_a y(k-1). \tag{6.143}$$

6.7.2 Obtenção de $E(z^{-1})$ e $F(z^{-1})$ do GMVC via GMVSS

O método GMVSS é uma alternativa ao projeto GMVC de ordem mínima para a compensação de qualquer valor de atraso d sem requerer a solução da equação Diofantina do método original. No entanto, via método GMVSS é possível obter os filtros $E(z^{-1})$ e $F(z^{-1})$ do GMVC ao estabelecer a relação entre o Preditor de Variância Mí-

nima do GMVSS e do GMVC (SILVEIRA; COELHO, 2011).

No GMVC a equação Diofantina se estabelece a partir da seguinte relação sobre o ruído do processo no futuro:

$$\frac{C(z^{-1})}{A(z^{-1})}w(k+d) = \underbrace{\frac{F(z^{-1})}{A(z^{-1})}w(k)}_{\text{Presente}} + \underbrace{E(z^{-1})w(k+d)}_{\text{Futuro}}. \quad (6.144)$$

No caso GMVSS essa relação é equivalente a

$$\begin{aligned} \sum_{i=1}^d CA^{(d-i)}\Gamma w(k-1+i) + w(k+d) &= \underbrace{CA^{(d-1)}\Gamma w(k)}_{\text{Presente}} \\ &+ \underbrace{\sum_{i=2}^d CA^{(d-i)}\Gamma w(k-1+i) + w(k+d)}_{\text{Futuro}}. \end{aligned} \quad (6.145)$$

Desse modo, Silveira e Coelho (2011) demonstraram que o filtro $E(z^{-1})$ do GMVC poderia ser obtido via GMVSS, da seguinte maneira:

$$E(z^{-1}) = 1 + CA^0\Gamma z^{-1} + CA^1\Gamma z^{-2} + \dots + CA^{(d-2)}\Gamma z^{-(d-1)}. \quad (6.146)$$

De maneira similar comparou-se a relação de $F(z^{-1})$, no Preditor de Variância Mínima do GMVC, com o vetor de ganho do Preditor de Estado de Variância Mínima, F . No GMVC a saída do processo controlado, $y(k)$, funciona como uma entrada do Preditor:

$$\hat{y}(k+d|k) = \frac{B(z^{-1})E(z^{-1})}{C(z^{-1})}u(k) + \frac{F(z^{-1})}{C(z^{-1})}y(k). \quad (6.147)$$

O mesmo ocorre com o Preditor de Estado de Variância Mínima no GMVSS, mas com o vetor de ganho $F = A^{(d-1)}\Gamma$:

$$\begin{aligned} \hat{x}(k+d|k) &= [A^d - FC] \bar{x}(k) + Fy(k) \\ &+ \begin{bmatrix} A^0 B \\ \vdots \\ A^{(d-1)} B \end{bmatrix}^T \begin{bmatrix} u(k) \\ \vdots \\ u(k-d+1) \end{bmatrix}. \end{aligned} \quad (6.148)$$

Silveira e Coelho (2011) observaram que os elementos do ganho F ,

$$F^T = [f_0 \quad f_1 \quad \cdots \quad f_{n_a-1}] , \quad (6.149)$$

eram equivalentes aos coeficientes de $F(z^{-1})$, isto é:

$$F(z^{-1}) = f_0 + f_1 z^{-1} + \cdots + f_{n_a-1} z^{-(n_a-1)}. \quad (6.150)$$

Veremos, mais adiante neste capítulo, que este resultado de solução intrínseca da equação Diofantina é interessante tanto para o GMVC como também para auxiliar em uma alternativa para o projeto do GPC.

6.8 CONTROLE PREDITIVO DE VARIÂNCIA MÍNIMA

O método GMVSS reproduz fielmente o GMVC, mas sem a necessidade de resolver a equação Diofantina. Tal contribuição, permite, com maior facilidade, que se resolva o problema de predição para qualquer valor de atraso d .

Utilizando-se dessa flexibilidade de predição do método GMVSS, Silveira et al. (2016) propuseram a extensão do horizonte de predição para além do valor do atraso, assumindo um atraso virtual $N_y > d$ na solução do preditor, em que N_y passou a ser tratado como um parâmetro de sintonia e estabeleceu a técnica como Controle Preditivo de Variância Mínima (PMVC).

Desse modo, seja o processo controlado, representado pelo modelo na forma canônica observável de um modelo ARIMAX, dado por,

$$\begin{aligned} x_a(k) &= A_a x_a(k-1) + B_a \Delta u(k-d) + \Gamma_a w(k-1), \\ y_a(k) &= C_a x_a(k) + w(k) = y(k), \end{aligned} \quad (6.151)$$

cujos valores do atraso é d , a solução preditiva considera o seguinte Preditor de Estado de Variância Mínima:

$$\begin{aligned} \hat{x}_a(k + N_y | k) &= [A_a^{N_y} - F_a C_a] \bar{x}_a(k) + F_a y(k) \\ &+ \begin{bmatrix} A_a^0 B_a \\ \vdots \\ A_a^{(N_y-1)} B_a \end{bmatrix}^T \begin{bmatrix} \Delta u(k) \\ \vdots \\ \Delta u(k - N_y + 1) \end{bmatrix}, \end{aligned} \quad (6.152)$$

$$F_a = A_a^{(N_y-1)} \Gamma_a. \quad (6.153)$$

Este, estabelecido em função do Filtro de Kalman do processo,

$$\bar{x}_a(k) = [A_a - \Gamma_a C_a] \bar{x}_a(k-1) + B_a \Delta u(k-d) + \Gamma_a y(k-1). \quad (6.154)$$

A lei de controle que implementa o controle preditivo de variância mínima para um horizonte de predição N_y e um fator de ponderação do esforço de controle q_0 é descrita por:

$$\begin{aligned} \Delta u(k) = & (C_a B_a + q_0)^{-1} \{ r(k + N_y) \\ & - [C_a A_a^{N_y} - C_a F_a C_a] \bar{x}_a(k) - C_a F_a y(k) \\ & - \begin{bmatrix} C_a A_a B_a \\ \vdots \\ C_a A_a^{(N_y-1)} B_a \end{bmatrix}^T \begin{bmatrix} \Delta u(k-1) \\ \vdots \\ \Delta u(k - N_y + 1) \end{bmatrix} \}, \end{aligned} \quad (6.155)$$

$$u(k) = u(k-1) + \Delta u(k). \quad (6.156)$$

É necessário frisar que este projeto pode também ser utilizado na forma posicional com base no modelo ARMAX, sendo mais indicado para aplicação em processos integradores, tipo-1.

A sintonia de N_y no Controle Preditivo de Variância Mínima é arbitrária, assim como em outros MPCs. Apesar de haver diversas tentativas de se estabelecer como escolher N_y nos MPCs, a prática se mostrou bastante impiedosa com os acadêmicos que arriscaram-se em fornecer as especificações do horizonte de predição. De todo o modo omitir-se de tal tentativa de especificação de N_y também não traz qualquer benefício. Por isto, segue uma discussão, no mínimo, interessante.

6.8.1 Sintonia de N_y no PMVC

No Controle Preditivo de Variância Mínima N_y não compreende uma trajetória de otimização, mas ao invés disso, compreende a minimização instantânea da função custo

$$J_{PMVC} = [\hat{y}(k + N_y|k) - r(k + N_y) + q_0 \Delta u(k)]^2. \quad (6.157)$$

Portanto, a cada instante k dá-se a solução $\Delta u(k)$ tal que J_{PMVC} seja mínimo quando decorridos N_y instantes futuros.

Considera-se resolver, agora, em k , o que resolve o problema de $y(k)$ convergir para $r(k)$ em N_y instantes futuros. Logo, quanto maior

for o valor de N_y , significa que há mais tempo para manobrar o processo controlado de modo a fazer $y(k)$ convergir para $r(k)$ em N_y instantes, exigindo-se menor esforço de controle. Ao contrário, quando $N_y \rightarrow d$, a solução deve ser imediata, gerando maior esforço de controle.

Observa-se, no entanto, que esta explicação sobre N_y é a premissa de todos os MPCs e é a base por trás da ideia de controle com horizonte retrocedente (do inglês, *receding horizon control*): planejar as ações de controle futuras, $\Delta u(k)$, $\Delta u(k+1)$, $\Delta u(k+2)$, \dots , que manobram $y(k)$ na direção de $r(k)$ em N_y instantes, aplicar apenas a ação atual $\Delta u(k)$ ao processo e, no instante seguinte, $k+1$, replanejar todas as ações de controle futuras com base nas novas informações do estado do processo, aplicando, de novo, só a ação presente, replanejando e descartando as previsões, repetidamente.

No Controle Preditivo de Variância Mínima, no entanto, a estratégia é diferente dos seus antecessores, como no Controle por Matriz Dinâmica, DMC, ou no Controle Preditivo Generalizado, GPC. O PMVC não produz ações de controle futuras, $\Delta u(k)$, $\Delta u(k+1)$, $\Delta u(k+2)$, \dots , tal como pode ser observado tanto no preditor da equação (6.152), como na lei de controle em (6.155), onde apenas o passado e o presente de $\Delta u(k)$ são considerados. Portanto, o PMVC ainda planeja o futuro via $\hat{y}(k+N_y|k)$ e replaneja repetidamente, sendo também um controlador de horizonte retrocedente, mas sempre considera que vai resolver o problema de $\hat{y}(k+N_y|k)$ apenas com o que vai fazer no instante presente de $\Delta u(k)$.

Sendo assim, a principal diferença do PMVC para outros MPCs é que as ações de controle passadas, $\Delta u(k)$, $\Delta u(k-1)$, \dots , $\Delta u(k-N_y+1)$, nunca são completamente esquecidas, diferente do que ocorre com o DMC e com o GPC, por exemplo, que as esquecem parcialmente, como veremos mais adiante neste capítulo.

A falta de planejamento de ações de controle futuras na estratégia PMVC acaba por limitar o quão estendido pode ser o seu horizonte N_y , ficando, assim, como uma regra prática, de que o valor do horizonte de predição de saída corresponda, no máximo, ao número de amostras que compreendem o tempo de assentamento t_s do processo controlado, quando este for assintoticamente estável:

$$d \leq N_y \leq \text{int} \left(\frac{t_s}{T_s} \right), \quad (6.158)$$

onde $\text{int}(\cdot)$ é um operador de conversão para número inteiro.

Além de N_y há o parâmetro q_0 para ponderar o esforço de controle. Estes dois parâmetros acabam concorrendo e complicando a sin-

tonia, já que aumentar q_0 , assim como aumentar N_y , minimiza o esforço de controle. O problema é que para cada valor N_y pode existir um novo valor q_0 e vice-versa. Sendo assim, algoritmos de busca e métodos de aprendizado de máquina, precisam, cada vez mais, participar da sintonia de MPCs para auxiliar na escolha desses parâmetros, que, de algum modo, requerem aspectos heurísticos de tentativa e erro supervisionados de maneira inteligente e com base em índices de desempenho e/ou de estabilidade.

O Controle Preditivo de Variância Mínima é uma estratégia ainda em desenvolvimento e conta com abordagens para o caso de sistemas de múltiplas entradas e múltiplas saídas, como apresentado por Silveira et al. (2020). Há também o caso em que o futuro das ações de controle são consideradas, desenvolvido pelo professor Rodrigo Trentini Preuss (IFSC), chamado de Controle Preditivo de Horizonte Irrestrito (do inglês, *Unrestricted Horizon Predictive Control*), UHPC (CASTRO et al., 2022). Estas técnicas são discutidas nos capítulos 7 e 8. Também, o emprego de métodos de aprendizado por reforço, na sintonia do PMVC, do DMC e do GPC, é demonstrado no Capítulo 9.

Exemplo 6.7 *Controle Preditivo de Variância Mínima aplicado no controle de tensão elétrica de um processo eletrônico subamortecido.*

Este é um exemplo de aplicação equivalente ao que foi realizado nos exemplos 6.5 e 6.6, permitindo que se estabeleça uma forma de comparação entre o PMVC e o seu antecessor, GMVC.

Neste exemplo utilizaremos o PMVC de forma adaptativa indireta ou auto-ajustável indireta (indirect self-tuning). Para isto, o estimador dos Mínimos Quadrados Recursivo será utilizado para estimar um modelo ARMAX de segunda ordem do processo controlado, em tempo real, sendo este aumentado à forma ARIMAX e passado à forma canônica observável no espaço de estado para a aplicação do PMVC.

6.9 CONTROLE POR MATRIZ DINÂMICA

O Controle por Matriz Dinâmica, ou DMC – *Dynamic Matrix Control*, foi proposto por Cutler e Ramaker (1980), surgindo como uma solução da indústria, especificamente na *Shell Oil Company* onde estes autores trabalhavam (CAMACHO; BORDONS, 2007).

Uma das principais diferenças na estratégia DMC é a não utilização de modelos paramétricos para a implementação do projeto, mas sim o uso de um modelo baseado na sequência discreta ou sinal discreto da resposta ao degrau do processo que se quer controlar, como

(CAMACHO; BORDONS, 2007):

$$y(k) = \sum_{i=1}^{\infty} g_i \Delta u(k-i), \quad (6.159)$$

em que $\Delta u(k) = u(k) - u(k-1)$ e g_i são equivalentes aos i parâmetros de Markov da resposta ao degrau de

$$y(k) = \frac{B(z^{-1})z^{-1}}{A(z^{-1})}u(k). \quad (6.160)$$

Desse modo, para se obter os parâmetros g_i necessários pela estratégia DMC pode-se aplicar um teste de resposta ao degrau sobre o processo que se quer controlar, armazenando as $y(i) = g_i$ amostras. Isto implica, portanto, que o DMC, na forma em que foi proposto em Cutler e Ramaker (1980), somente pode ser aplicado a processos estáveis.

O DMC é uma estratégia de controle ótimo que se baseia em determinar a sequência de N_u variações de ações de controle (i.e., $\Delta u(k)$, $\Delta u(k+1)$, ..., $\Delta u(k+N_u-1)$) que minimiza a seguinte função custo:

$$J_{DMC} = \sum_{j=1}^{N_y} [\hat{y}(k+j|k) - r(k+j)]^2 + \sum_{j=1}^{N_u} \lambda [\Delta u(k+j-1)]^2. \quad (6.161)$$

O parâmetro λ é um escalar que pondera o esforço de controle de maneira similar ao parâmetro q_0 na estratégia GMVC.

Diferentemente do PMVC, onde exigia-se precisamente a previsão N_y -passos no futuro (i.e., $\hat{y}(k+N_y|k)$), no DMC considera-se uma trajetória de previsões, de $j = 1, \dots, N_y$ (i.e., $\hat{y}(k+1|k)$, ..., $\hat{y}(k+N_y|k)$).

O problema de minimização de J_{DMC} é mais simples de ser resolvido quando a função custo é reescrita na forma vetorial, como segue:

$$J_{DMC} = \mathbf{e}^T \mathbf{e} + \lambda \mathbf{u}^T \mathbf{u}, \quad (6.162)$$

onde o vetor de erros de seguimento de referência no futuro é dado por

$$\mathbf{e} = \begin{bmatrix} \hat{y}(k+1|k) - r(k+1) \\ \hat{y}(k+2|k) - r(k+2) \\ \vdots \\ \hat{y}(k+N_y|k) - r(k+N_y) \end{bmatrix}, \quad (6.163)$$

e o vetor de variações de ações de controle no futuro é descrito por

$$\mathbf{u} = \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \vdots \\ \Delta u(k+N_u-1) \end{bmatrix}. \quad (6.164)$$

Para que seja possível minimizar J_{DMC} , primeiro iremos desenvolver o preditor usado nesta estratégia de controle MPC.

6.9.1 Preditor na estratégia DMC

Assumindo que o modelo do processo é corrompido por uma perturbação aleatória $w(k)$, sendo

$$y(k) = \sum_{i=1}^{\infty} g_i \Delta u(k-i) + w(k) \quad (6.165)$$

e projetando-se este modelo para j -passos no futuro obtém-se a expansão representada na Equação (6.169), mostrada mais à frente no texto. Nela é possível verificar a participação da perturbação aleatória no futuro, $w(k+j)$, e assim como se fez no caso GMVC aplica-se uma solução de estimação, dada por:

$$w(k+j) = y(k+j) - \hat{y}(k+j), \quad (6.166)$$

$$w(k) = y(k) - \hat{y}(k), \quad (6.167)$$

onde $\hat{y}(k) = y(k) - w(k)$ é o próprio modelo do processo, sem perturbação, i.e.,

$$\hat{y}(k) = \sum_{i=1}^{\infty} g_i \Delta u(k-i). \quad (6.168)$$

$$\begin{aligned}
y(k+1) &= \sum_{i=1}^{\infty} g_i \Delta u(k+1-i) + w(k+1) \\
&= \underbrace{\sum_{i=1}^1 g_i \Delta u(k) + w(k+1)}_{\text{Presente e Futuro}} + \underbrace{\sum_{i=2}^{\infty} g_i \Delta u(k+1-i)}_{\text{Passado}} \\
y(k+2) &= \sum_{i=1}^{\infty} g_i \Delta u(k+2-i) + w(k+2) \\
&= \underbrace{\sum_{i=1}^2 g_i \Delta u(k+2-i) + w(k+2)}_{\text{Presente e Futuro}} + \underbrace{\sum_{i=3}^{\infty} g_i \Delta u(k+2-i)}_{\text{Passado}} \\
&\vdots \\
y(k+j) &= \sum_{i=1}^{\infty} g_i \Delta u(k+j-i) + w(k+j) \\
&= \underbrace{\sum_{i=1}^j g_i \Delta u(k+j-i) + w(k+j)}_{\text{Presente e Futuro}} + \underbrace{\sum_{i=j+1}^{\infty} g_i \Delta u(k+j-i)}_{\text{Passado}}
\end{aligned} \tag{6.169}$$

Negligenciando-se a influência futura de $w(k+j)$ no preditor mostrado em (6.169), uma versão estimada é então proposta:

$$\hat{y}(k+j) = \sum_{i=1}^{\infty} g_i \Delta u(k+j-i) + w(k), \tag{6.170}$$

$$\begin{aligned}
\hat{y}(k+j) &= \\
&\underbrace{\sum_{i=1}^j g_i \Delta u(k+j-i)}_{\text{Presente e Futuro}} + \underbrace{\sum_{i=j+1}^{\infty} g_i \Delta u(k+j-i)}_{\text{Passado}} + y(k) - \hat{y}(k)
\end{aligned} \tag{6.171}$$

Substituindo-se o modelo $\hat{y}(k) = \sum_{i=1}^{\infty} g_i \Delta u(k-i)$ na equação do preditor obtém-se a seguinte expressão:

$$\begin{aligned} \hat{y}(k+j) = & \underbrace{\sum_{i=1}^j g_i \Delta u(k+j-i)}_{\text{Presente e Futuro}} + y(k) \\ & + \underbrace{\sum_{i=j+1}^{\infty} g_i \Delta u(k+j-i)}_{\text{Passado}} - \underbrace{\sum_{i=1}^{\infty} g_i \Delta u(k-i)}_{\text{Passado}}. \end{aligned} \quad (6.172)$$

Com base nos termos passados de $\Delta u(k)$ é possível notar que

$$\begin{aligned} \hat{y}(k+j) = & \underbrace{\sum_{i=1}^j g_i \Delta u(k+j-i)}_{\text{Presente e Futuro}} + y(k) \\ & + \underbrace{g_{j+1} \Delta u(k-1) + g_{j+2} \Delta u(k-2) + \cdots + g_{j+\infty} \Delta u(k-\infty)}_{\text{Passado}} \\ & - \underbrace{\sum_{i=1}^{\infty} g_i \Delta u(k-i)}_{\text{Passado}}, \end{aligned} \quad (6.173)$$

podendo-se reescrever a equação do preditor da seguinte maneira:

$$\begin{aligned} \hat{y}(k+j) = & \underbrace{\sum_{i=1}^j g_i \Delta u(k+j-i)}_{\text{Presente e Futuro}} + y(k) + \underbrace{\sum_{i=1}^{\infty} (g_{j+i} - g_i) \Delta u(k-i)}_{\text{Resposta Livre do Modelo}}. \end{aligned} \quad (6.174)$$

A parcela de Resposta Livre do Modelo, por se tratar de um modelo de resposta ao degrau especificamente de processos assintoticamente estáveis, pode ser simplificada para quando

$$i \rightarrow \infty, \quad (6.175)$$

$$g_{j+i} - g_i \approx 0. \quad (6.176)$$

Nesse caso, a Resposta Livre do Modelo tem seu somatório truncado em N (CAMACHO; BORDONS, 2007), como segue:

$$f_r(k+j) = y(k) + \sum_{i=1}^N (g_{j+i} - g_i) \Delta u(k-i). \quad (6.177)$$

Este valor de N é maior do que N_y e sua escolha pode ser feita com base na análise de quando $g_{N_y+N} - g_N \approx 0$. Veremos o porquê disto mais adiante quando desenvolvermos o Modelo de Resposta Livre na forma matricial para uma sequência de predição $j = 1, \dots, N_y$.

Um dos problemas destacados na solução do GMVC e do PMVC era relativo à predição de $y(k+d)$ e $y(k+N_y)$, respectivamente, com base nas medidas disponíveis até o instante presente k , resultando na simbologia $\hat{y}(k+d|k)$ e $\hat{y}(k+N_y|k)$. No DMC há a consideração das variações de ações de controle futuras, $\Delta u(k)$, $\Delta u(k+1)$, \dots , $\Delta u(k+N_u-1)$, mas como essas ações futuras são calculadas no instante k , então assume-se a mesma simbologia ao predito do DMC, conforme:

$$\hat{y}(k+j|k) = \sum_{i=1}^j g_i \Delta u(k+j-i) + f_r(k+j). \quad (6.178)$$

Fazendo-se a projeção desse sistema de predição através da trajetória $j = 1, \dots, N_y$, obtém-se:

$$\begin{aligned} \hat{y}(k+1|k) &= g_1 \Delta u(k) + f_r(k+1) \\ \hat{y}(k+2|k) &= g_2 \Delta u(k) + g_1 \Delta u(k+1) + f_r(k+2) \\ &\vdots \\ \hat{y}(k+N_y|k) &= g_{N_y} \Delta u(k) + g_{N_y-1} \Delta u(k+1) + \dots \\ &\quad + g_1 \Delta u(k+N_y-1) + f_r(k+N_y). \end{aligned} \quad (6.179)$$

Esta projeção de predições pode ser representada por uma forma matricial, como segue:

$$\hat{\mathbf{y}} = \mathbf{G}\mathbf{u} + \mathbf{f}_r, \quad (6.180)$$

$$\begin{bmatrix} \hat{y}(k+1|k) \\ \hat{y}(k+2|k) \\ \hat{y}(k+3|k) \\ \vdots \\ \hat{y}(k+N_y|k) \end{bmatrix} = \begin{bmatrix} g_1 & 0 & 0 & 0 & 0 \\ g_2 & g_1 & 0 & 0 & 0 \\ g_3 & g_2 & g_1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ g_{N_y} & \cdots & g_3 & g_2 & g_1 \end{bmatrix} \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \Delta u(k+2) \\ \vdots \\ \Delta u(k+N_y-1) \end{bmatrix} \\
+ \begin{bmatrix} f_r(k+1) \\ f_r(k+2) \\ f_r(k+3) \\ \vdots \\ f_r(k+N_y) \end{bmatrix}. \quad (6.181)$$

É importante notar que a matriz \mathbf{G} e o vetor \mathbf{u} alteram-se caso o horizonte de predição de controle N_u seja menor que N_y . Se este for o caso reduz-se o número de colunas de \mathbf{G} e o número de linhas de \mathbf{u} para a dimensão N_u , tal que $\mathbf{G}_{N_y \times N_u}$ e $\mathbf{u}_{N_u \times 1}$.

A sequência de predições $j = 1, \dots, N_y$ do Modelo de Resposta Livre,

$$f_r(k+j) = y(k) + \sum_{i=1}^N (g_{j+i} - g_i) \Delta u(k-i),$$

representada pelo vetor \mathbf{f} , pode também ser representada de uma forma matricial com base na sequência de predições mostrada a seguir:

$$\begin{aligned} f_r(k+1) &= y(k) + (g_2 - g_1) \Delta u(k-1) + (g_3 - g_2) \Delta u(k-2) \\ &\quad + \cdots + (g_{1+N} - g_N) \Delta u(k-N) \\ f_r(k+2) &= y(k) + (g_3 - g_1) \Delta u(k-1) + (g_4 - g_2) \Delta u(k-2) \\ &\quad + \cdots + (g_{2+N} - g_N) \Delta u(k-N) \\ f_r(k+3) &= y(k) + (g_4 - g_1) \Delta u(k-1) + (g_5 - g_2) \Delta u(k-2) \\ &\quad + \cdots + (g_{3+N} - g_N) \Delta u(k-N) \\ &\vdots \\ f_r(k+N_y) &= y(k) + (g_{N_y+1} - g_1) \Delta u(k-1) \\ &\quad + (g_{N_y+2} - g_2) \Delta u(k-2) \\ &\quad + \cdots + (g_{N_y+N} - g_N) \Delta u(k-N). \end{aligned} \quad (6.182)$$

Uma generalização matricial do Modelo de Resposta Livre pode ser

descrita por:

$$\begin{bmatrix} f_r(k+1) \\ f_r(k+2) \\ f_r(k+3) \\ \vdots \\ f_r(k+N_y) \end{bmatrix} = \begin{bmatrix} 1 & g_2 - g_1 & g_3 - g_2 & \cdots & g_{1+N} - g_N \\ 1 & g_3 - g_1 & g_4 - g_2 & \cdots & g_{2+N} - g_N \\ 1 & g_4 - g_1 & g_5 - g_2 & \cdots & g_{3+N} - g_N \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 1 & g_{N_y+1} - g_1 & g_{N_y+2} - g_2 & \cdots & g_{N_y+N} - g_N \end{bmatrix} \begin{bmatrix} y(k) \\ \Delta u(k-1) \\ \Delta u(k-2) \\ \vdots \\ \Delta u(k-N) \end{bmatrix} \quad (6.183)$$

$$\mathbf{f}_r = \mathbf{F}\mathbf{x}, \quad (6.184)$$

em que \mathbf{F} e \mathbf{x} tem dimensão $N_y \times (N+1)$ e $(N+1) \times 1$, respectivamente.

A predição na estratégia DMC na forma matricial pode então ser definida como sendo

$$\hat{\mathbf{y}} = \mathbf{G}\mathbf{u} + \mathbf{f}_r = \mathbf{G}\mathbf{u} + \mathbf{F}\mathbf{x}. \quad (6.185)$$

6.9.2 Lei de Controle DMC

Para a obtenção da lei de controle DMC retorna-se à função custo do problema, dada por:

$$J_{DMC} = \mathbf{e}^T \mathbf{e} + \lambda \mathbf{u}^T \mathbf{u}.$$

O vetor erro de seguimento de referência predita é dado por

$$\mathbf{e} = \hat{\mathbf{y}} - \mathbf{r}, \quad (6.186)$$

em que

$$\mathbf{r}^T = [r(k+1) \quad r(k+2) \quad \cdots \quad r(k+N_y)]. \quad (6.187)$$

A função custo do problema DMC pode ser reescrita da seguinte

maneira:

$$J_{DMC} = [\mathbf{Gu} + \mathbf{Fx} - \mathbf{r}]^T [\mathbf{Gu} + \mathbf{Fx} - \mathbf{r}] + \lambda \mathbf{u}^T \mathbf{u}. \quad (6.188)$$

Fazendo-se temporariamente a consideração de que $\mathbf{E} = \mathbf{r} - \mathbf{Fx}$,

$$\begin{aligned} J_{DMC} &= [\mathbf{Gu} - \mathbf{E}]^T [\mathbf{Gu} - \mathbf{E}] + \lambda \mathbf{u}^T \mathbf{u}, \\ J_{DMC} &= \mathbf{u}^T [\mathbf{G}^T \mathbf{G} + \lambda \mathbf{I}] \mathbf{u} - 2\mathbf{u}^T \mathbf{G}^T \mathbf{E} + \mathbf{E}^T \mathbf{E}. \end{aligned} \quad (6.189)$$

A obtenção da lei de controle DMC é feita pela minimização de J_{DMC} em relação à sequência de N_u ações futuras de $\Delta u(k)$, isto é,

$$\min_{\mathbf{u}} J_{DMC} = \frac{\partial J_{DMC}}{\partial \mathbf{u}} = 0, \quad (6.190)$$

que resulta em

$$2 [\mathbf{G}^T \mathbf{G} + \lambda \mathbf{I}] \mathbf{u} = 2 \mathbf{G}^T \mathbf{E},$$

$$\mathbf{u} = [\mathbf{G}^T \mathbf{G} + \lambda \mathbf{I}]^{-1} \mathbf{G}^T [\mathbf{r} - \mathbf{Fx}]. \quad (6.191)$$

O ganho do DMC tem estrutura matricial, definida por:

$$\mathbf{K} = [\mathbf{G}^T \mathbf{G} + \lambda \mathbf{I}]^{-1} \mathbf{G}^T. \quad (6.192)$$

Cada linha desta matriz está relacionada a um determinado momento da sequência de predição das variações da ação de controle contidas em \mathbf{u} , tal que esse ganho está estruturado conforme

$$\mathbf{K}^T = [\mathbf{k}_1 \quad \mathbf{k}_2 \quad \cdots \quad \mathbf{k}_{N_y}] , \quad (6.193)$$

sendo a variação da ação de controle, que será aplicada no instante k , descrita por

$$\Delta u(k) = \mathbf{k}_1 [\mathbf{r} - \mathbf{Fx}]. \quad (6.194)$$

E o sinal de controle aplicado ao processo é dado por

$$u(k) = u(k-1) + \Delta u(k). \quad (6.195)$$

É importante observar que \mathbf{K} tem dimensão $N_u \times N_y$ e, portanto, a dimensão do vetor de ganho \mathbf{k}_1 é $1 \times N_y$.

Exemplo 6.8 *Controle por Matriz Dinâmica aplicado ao sistema eletrônico subamortecido que foi utilizado no Exemplo 6.5 e que pode ser descrito pelos seguintes polinômios do modelo ARMAX estimado:*

$$\begin{aligned} A(z^{-1}) &= 1 - 1,2356881z^{-1} + 0,4526202z^{-2} \\ B(z^{-1}) &= 0,0101615 + 0,1947393z^{-1} \\ C(z^{-1}) &= 1 - 0,0398287z^{-1} + 0,0825432z^{-2} \end{aligned} \quad (6.196)$$

Para simular o ruído Gaussiano, $w(k)$, sugere-se utilizar a variância estimada durante a identificação do modelo do sistema, sendo esta: $\sigma_w^2 = 0,0068327$.

— INCLUIR CÓDIGOS DA SIMULAÇÃO
 — INCLUIR FIGURAS DA SIMULAÇÃO
 — INCLUIR FIGURAS DOS RESULTADOS EXPERIMENTAIS

6.10 CONTROLE PREDITIVO GENERALIZADO

O método conhecido como Controle Preditivo Generalizado, o GPC, foi proposto por Clarke et al. (1987) e pode ser entendido como uma evolução do GMVC pela incorporação de uma trajetória de $N_y \geq d$ predições da saída no critério de desempenho a ser otimizado, utilizando a mesma função custo da estratégia DMC, mas com base em modelo paramétrico igual ao usado na estratégia GMVC. Desse modo, o GPC se estabelece como uma estratégia MPC estocástica, sucessora do GMVC.

Em caráter introdutório, faremos algumas simplificações na abordagem do GPC neste capítulo, tal que o atraso do processo será sempre considerado $d = 1$, conforme feito em Bitmead et al. (1990). Desse modo pode-se definir o problema GPC como sendo

$$\underbrace{\min}_{\underline{\Delta}_u} J_{gpc} = \mathbf{E} \left\{ \sum_{j=1}^{N_y} [y(k+j) - r(k+j)]^2 + \lambda \sum_{j=1}^{N_u} [\Delta u(k+j-1)]^2 \right\}, \quad (6.197)$$

$$\underline{\Delta}_u = \begin{bmatrix} \Delta u(k) & \Delta u(k+1) & \cdots & \Delta u(k+N_u-1) \end{bmatrix}^T, \quad (6.198)$$

sujeito às restrições dinâmicas impostas pelo sistema discreto linear e

variante no tempo, descrito na forma ARIMAX:

$$A_k(z^{-1})y(k) = B_k(z^{-1})u(k-1) + C_k(z^{-1})\frac{w(k)}{\Delta}. \quad (6.199)$$

Em (6.197), $\mathbf{E}[\cdot]$ denota o operador esperança matemática e que se faz necessário dada a natureza do modelo ARIMAX que contém a parcela do ruído colorido, $C_k(z^{-1})\frac{w(k)}{\Delta}$, que agrega a característica estocástica ao problema de otimização do GPC. As escolhas de sintonia, no entanto, são análogas ao caso DMC, constituídas dos horizontes das trajetórias de saída (N_y), de controle (N_u) e do fator de ponderação da variação do sinal de controle (λ).

Como a função custo do GPC é análoga a do DMC, então a obtenção da lei de controle também o será. Portanto, a diferença está na geração da trajetória de predições da saída e que incorpora o Preditor de Variância Mínima como solução, tal que seja possível determinar N_y predições da saída com base em informações conhecidas até o instante atual k , isto é, a saída predita $\hat{y}(k+j|k)$ para $j = 1, \dots, N_y$.

Com base no cenário exposto, a seguir, passaremos ao desenvolvimento do preditor utilizado pelo GPC para depois retornar à solução mostrada em (6.197).

6.10.1 Preditor de variância mínima usado no GPC

O GPC necessita de N_y predições da saída $y(k)$ do modelo ARIMAX mostrado em (6.199), que pode ser reescrito com base na variação da ação de controle, $\Delta u(k) = u(k) - u(k-1)$, como segue:

$$y(k) = \frac{B_k(z^{-1})}{\Delta A_k(z^{-1})}\Delta u(k-1) + \frac{C_k(z^{-1})}{\Delta A_k(z^{-1})}w(k). \quad (6.200)$$

Desse modo, a saída predita a j -passos à frente é dada por:

$$y(k+j) = \frac{B_{k+j}(z^{-1})}{\Delta A_{k+j}(z^{-1})}\Delta u(k+j-1) + \frac{C_{k+j}(z^{-1})}{\Delta A_{k+j}(z^{-1})}w(k+j). \quad (6.201)$$

Tal predição se torna inviável na prática já que não se conhece os polinômios que descrevem a dinâmica do processo e nem a perturbação $w(k)$ a $k+j$ passos no futuro.

A solução para este problema de predição já foi apresentada com

relação ao GMVC, sendo proposta com base no Preditor de Variância Mínima do Åström (1970), onde considera-se apenas o que se conhece sobre o processo controlado até o instante k , como segue:

$$\hat{y}(k+j|k) = \frac{B_k(z^{-1})}{\Delta A_k(z^{-1})} \Delta u(k+j-1) + \frac{F_j(z^{-1})}{\Delta A_k(z^{-1})} w(k), \quad (6.202)$$

onde $F_j(z^{-1})$ está relacionado à separação entre as informações do passado e do presente do ruído, das informações do futuro, conforme

$$\frac{C_k(z^{-1})}{\Delta A_k(z^{-1})} w(k+j) = \underbrace{\frac{F_j(z^{-1})}{\Delta A_k(z^{-1})} w(k)}_{\text{Passado e Presente}} + \underbrace{E_j(z^{-1}) w(k+j)}_{\text{Futuro}}, \quad (6.203)$$

e que apenas se pode estimar tal influência futura do ruído por meio de

$$E_j(z^{-1}) w(k+j) = y(k+j) - \hat{y}(k+j|k), \quad (6.204)$$

$$w(k) = \frac{y(k) - \hat{y}(k|k)}{E_j(z^{-1})}. \quad (6.205)$$

A expressão mostrada em (6.203) dá origem à equação Diofantina do preditor, sendo esta

$$C_k(z^{-1}) = z^{-j} F_j(z^{-1}) + \Delta A_k(z^{-1}) E_j(z^{-1}), \quad (6.206)$$

e cuja solução fornece os polinômios

$$\begin{aligned} E_j(z^{-1}) &= e_{j,0} + e_{j,1} z^{-1} + \cdots + e_{j,(j-1)} z^{-(j-1)} \\ F_j(z^{-1}) &= f_{j,0} + f_{j,1} z^{-1} + \cdots + f_{j,n_a} z^{-n_a} \end{aligned} \quad (6.207)$$

em que n_a é a ordem do polinômio $A_k(z^{-1})$.

Diferentemente do caso GMVC, com o GPC haverá $j = 1, \dots, N_y$ polinômios $E_j(z^{-1})$ e $F_j(z^{-1})$, sendo a ordem destes respectivamente ligadas à escolha de N_y e à ordem do sistema em si, isto é, $n_{e_j} = j - 1$ e $n_{f_j} = n_a, \forall j \in \mathbb{N}$.

Substituindo-se o valor estimado do ruído $w(k)$, mostrado em (6.205), na equação do preditor em (6.202), obtém-se uma expressão de predição dependente apenas da saída medida $y(k)$ e também de informações do passado, presente e futuro de $\Delta u(k)$, dada por:

$$\hat{y}(k+j|k) = \frac{E_j(z^{-1}) B_k(z^{-1}) \Delta u(k+j-1) + F_j(z^{-1}) y(k)}{\Delta A_k(z^{-1}) E_j(z^{-1}) + z^{-j} F_j(z^{-1})}, \quad (6.208)$$

e que tem a própria equação Diofantina como polinômio característico, podendo ser reescrita de maneira mais simples, como segue:

$$\hat{y}(k+j|k) = \frac{E_j(z^{-1})B_k(z^{-1})\Delta u(k+j-1) + F_j(z^{-1})y(k)}{C_k(z^{-1})}. \quad (6.209)$$

Com base no sistema de predição mostrado em (6.212) deve-se enfatizar que o GPC será assintoticamente estável se e somente se as raízes de $C_k(z^{-1})$ tiverem módulo menor do que um. Enfatiza-se também que $C_k(z^{-1})$ é sempre um polinômio mônico, i.e., inicia sempre com um termo unitário (e.g., $C_k(z^{-1}) = 1 + c_1 z^{-1}$).

A prática do GPC ao longo dos anos evitou a abordagem baseada em modelo com ruído colorido (SILVEIRA et al., 2016), sendo, normalmente, adotado um modelo ARIX em que $C_k(z^{-1}) = 1, \forall k \in \mathbb{Z}^{0+}$. Via método GMVSS, aquele usado no PMVC, é possível desenvolver outras formas para tratar o caso de ruído colorido no GPC, mas até o momento da redação deste documento, tal solução ainda não foi amplamente estudada, nem mesmo publicada.

Bitmead et al. (1990) apresentaram uma alternativa não muito elegante, mas tradicional e eficaz para tratar o caso geral de $C_k(z^{-1})$, onde o problema passa a ser resolvido com base em versões filtradas da variação do sinal de controle e da saída, como segue:

$$\frac{\Delta u(k+j-1)}{C_k(z^{-1})} = \Delta u_f(k+j-1), \quad (6.210)$$

$$\frac{y(k)}{C_k(z^{-1})} = y_f(k), \quad (6.211)$$

tal que o preditor seja reescrito como:

$$\hat{y}(k+j|k) = E_j(z^{-1})B_k(z^{-1})\Delta u_f(k+j-1) + F_j(z^{-1})y_f(k). \quad (6.212)$$

Assim como no DMC, neste ponto o preditor precisa ser avaliado em relação às partes que compõem o Modelo de Resposta Livre e o Modelo de Resposta Forçada em que se destacam as ações de controle futuras. Tais termos ainda se encontram misturados na parcela

$$\begin{aligned} G_j(z^{-1}) &= E_j(z^{-1})B_k(z^{-1}) \\ G_j(z^{-1}) &= g_{j,0} + g_{j,1}z^{-1} + \dots + g_{j,(n_{e_j}+n_b)}z^{-(n_{e_j}+n_b)}, \end{aligned} \quad (6.213)$$

onde n_b é a ordem do polinômio $B_k(z^{-1})$.

A maior ordem que o polinômio $G_j(z^{-1})$ pode assumir é quando $j = N_y$, assim como a maior trajetória de ações futuras possíveis é quando $N_u = N_y$. Desse modo a ordem de $G_{N_y}(z^{-1})$ será

$$n_{g_{N_y}} = n_{e_{N_y}} + n_b = N_y - 1 + n_b. \quad (6.214)$$

Este ponto do desenvolvimento do GPC exige grande concentração daqueles que desejam compreender seu projeto com maior profundidade. A partir dele, diversas formas de se obter os polinômios $G_j(z^{-1})$ são relatadas na literatura de Controle e Sistemas, como o método recursivo apresentado por Camacho e Bordons (2007) e o método em espaço de estado relatado por (BITMEAD et al., 1990). No entanto, nestes dois casos não há a consideração da participação do polinômio $C_k(z^{-1})$ para quando a ordem deste é $n_c \geq 1$, o que limita o tratamento estocástico herdado do GMVC (apesar de não haver, de fato, grande contribuição estocástica do GPC, conforme voltaremos a discutir no final deste capítulo).

Diferentemente dos métodos que lidam com $C_k(z^{-1}) = 1$, vamos utilizar uma solução baseada no método GMVSS (SILVEIRA; COELHO, 2011). Para tanto, primeiro façamos a análise de um exemplo de expansão do preditor do GPC, isto é, avaliar a evolução de $\hat{y}(k+j|k)$ para $j = 1, \dots, N_y$. Posteriormente, uma generalização será apresentada e relacionada ao método GMVSS para permitir os cálculos para qualquer sistema ARIMAX.

Exemplo 6.9 *Evolução do Preditor de Variância Mínima do GPC, assumindo a evolução de $k+j$ predições para $j = 1, \dots, N_y$, aplicado sobre um processo ARIMAX de segunda ordem, descrito por:*

$$\Delta A(z^{-1})y(k) = B(z^{-1})\Delta u(k-1) + C(z^{-1})w(k), \quad (6.215)$$

$$\begin{aligned} A(z^{-1}) &= 1 + a_1 z^{-1} + a_2 z^{-2}, \\ B(z^{-1}) &= b_0 + b_1 z^{-1}, \\ C(z^{-1}) &= 1 + c_1 z^{-1} + c_2 z^{-2}, \end{aligned} \quad (6.216)$$

em que na forma aumentada ARIMAX, $\Delta A(z^{-1})$ é equivalente a

$$\bar{A}(z^{-1}) = \Delta A(z^{-1}) = 1 + \bar{a}_1 z^{-1} + \bar{a}_2 z^{-2} + \bar{a}_3 z^{-3}. \quad (6.217)$$

Para se realizar a predição de $\hat{y}(k+j|k)$ para $j = 1, \dots, N_y$, primeiro recordemos como os polinômios $E_j(z^{-1})$ e $F_j(z^{-1})$ são definidos

com base em j e n_a , respectivamente:

$$E_j(z^{-1}) = 1 + e_{j,1}z^{-1} + \cdots + e_{j,j-1}z^{-(j-1)}, \quad (6.218)$$

$$F_j(z^{-1}) = f_{j,0} + f_{j,1}z^{-1} + \cdots + f_{j,n_a}z^{-n_a}. \quad (6.219)$$

Desse modo, o polinômio $G_j(z^{-1})$ para o caso com o polinômio $B(z^{-1})$ cujo $n_b = 1$ é dado por:

$$G_j(z^{-1}) = \left(1 + e_{j,1}z^{-1} + \cdots + e_{j,j-1}z^{-(j-1)}\right) (b_0 + b_1z^{-1}), \quad (6.220)$$

$$G_j(z^{-1}) = g_{j,0} + g_{j,1}z^{-1} + \cdots + g_{j,(j-1-n_b)}z^{-(j-1+n_b)}. \quad (6.221)$$

Portanto, a ordem de $G_j(z^{-1})$ poderá aumentar até $n_g = N_y - 1 + n_b$, onde veremos que até $N_y - 1$ ações de controle futuras comporão a parte de Resposta Forçada (i.e. presente e futuro) enquanto n_b regressões das ações de controle comporão a parte de Resposta Livre (i.e. passado), como segue:

$$\hat{y}(k + j|k) = E_j(z^{-1})B_k(z^{-1})\Delta u_f(k + j - 1) + F_j(z^{-1})y_f(k) \quad (6.222)$$

fazendo $j = 1, \dots, N_y$ obtém-se:

$$\begin{aligned} \hat{y}(k + 1|k) &= \begin{bmatrix} g_{1,0} & g_{1,1}z^{-1} \end{bmatrix} \Delta u_f(k) + F_1(z^{-1})y_f(k) \\ \hat{y}(k + 1|k) &= g_{1,0} \Delta u_f(k) \\ &+ g_{1,1}\Delta u_f(k - 1) + F_1(z^{-1})y_f(k) \end{aligned} \quad (6.223)$$

$$\begin{aligned} \hat{y}(k + 2|k) &= \begin{bmatrix} g_{2,0} & g_{2,1}z^{-1} & g_{2,2}z^{-2} \end{bmatrix} \Delta u_f(k + 1) + F_2(z^{-1})y_f(k) \\ \hat{y}(k + 2|k) &= g_{2,0}\Delta u_f(k + 1) + g_{2,1}\Delta u_f(k) \\ &+ g_{2,2}\Delta u_f(k - 1) + F_2(z^{-1})y_f(k) \end{aligned} \quad (6.224)$$

$$\begin{aligned} \hat{y}(k + 3|k) &= \\ &\begin{bmatrix} g_{3,0} & g_{3,1}z^{-1} & g_{3,2}z^{-2} & g_{3,3}z^{-3} \end{bmatrix} \Delta u_f(k + 2) + F_3(z^{-1})y_f(k) \\ \hat{y}(k + 3|k) &= g_{3,0}\Delta u_f(k + 2) + g_{3,1}\Delta u_f(k + 1) + g_{3,2}\Delta u_f(k) \\ &+ g_{3,3}\Delta u_f(k - 1) + F_3(z^{-1})y_f(k) \end{aligned} \quad (6.225)$$

tal que para quando $j = N_y$ e $n_g = N_y - 1 + n_b = N_y$ obtém-se:

$$\begin{aligned} \hat{y}(k + N_y | k) = & \begin{bmatrix} g_{N_y,0} & g_{N_y,1}z^{-1} & \cdots & g_{N_y,N_y}z^{-N_y} \end{bmatrix} \Delta u_f(k + N_y - 1) \\ & + F_{N_y}(z^{-1})y_f(k) \\ \hat{y}(k + N_y | k) = & g_{N_y,0}\Delta u_f(k + N_y - 1) + g_{N_y,1}\Delta u_f(k + N_y - 2) \\ & + \cdots + g_{N_y,(N_y-2)}\Delta u_f(k + 1) + g_{N_y,(N_y-1)}\Delta u_f(k) \\ & + g_{N_y,N_y}\Delta u_f(k - 1) + F_{N_y}(z^{-1})y_f(k) \end{aligned} \quad (6.226)$$

Observe que para cada predição a j -passos à frente, para $j = 1, \dots, N_y$, o Modelo de Resposta Livre neste caso com $n_b = 1$ foi sempre função de

$$f_r(k + j) = g_{j,j}\Delta u_f(k - 1) + F_j(z^{-1})y_f(k), \quad (6.227)$$

caracterizado pela regressão de $\Delta u_f(k)$ limitada à ordem de $B(z^{-1})$. Portanto, repetindo-se este experimento para $n_b > 1$ é possível confirmar que a resposta livre acessa ações passadas do controlador até o instante $k - n_b$.

Com base no Exemplo 6.10 é possível generalizar o cálculo do Preditor de Variância Mínima adotado para a geração das N_y trajetórias futuras do GPC, sendo este descrito como:

$$\begin{aligned} \hat{y}(k + j | k) = & \begin{bmatrix} g_{j,0} & g_{j,1}z^{-1} & \cdots & g_{j,(j-1+n_b)}z^{-(j-1+n_b)} \end{bmatrix} \Delta u_f(k + j - 1) \\ & + F_j(z^{-1})y_f(k), \end{aligned} \quad (6.228)$$

$$\begin{aligned} \hat{y}(k + j | k) = & g_{j,0}\Delta u_f(k + j - 1) + g_{j,1}\Delta u_f(k + j - 2) + \cdots \\ & + g_{j,(j-2)}\Delta u_f(k + 1) + g_{j,(j-1)}\Delta u_f(k) \\ & + g_{j,j}\Delta u_f(k - 1) + \cdots + g_{j,(j-2+n_b)}\Delta u_f(k - n_b + 1) \\ & + g_{j,(j-1+n_b)}\Delta u_f(k - n_b) + F_j(z^{-1})y_f(k). \end{aligned} \quad (6.229)$$

Para cada $G_j(z^{-1})$, $j = 1, \dots, N_y$, os últimos n_b elementos são relativos aos regressores da ação de controle, enquanto que, de $g_{j,0}$ até $g_{j,(j-1)}$, são usados nas ações de controle futuras ou preditas.

É notório que o cálculo dos parâmetros do preditor do GPC são complexos e executá-los à mão acaba por limitar o número de predi-

ções, a ordem do modelo do processo e também a adoção da versão estocástica com o caso de ruído colorido. Por este motivo, veremos a seguir uma maneira para calcular $E_j(z^{-1})$, $F_j(z^{-1})$ e $G_j(z^{-1})$.

6.10.2 Projeto do preditor do GPC com base no GMVSS

O método GMVSS, proposto em Silveira e Coelho (2011), foi desenvolvido para permitir o projeto de controladores preditivos baseados em modelos ARMAX e ARIMAX sem desprezar o caso de ruído colorido e buscando contornar a solução das equações Diofantinas, que no caso do GPC, são várias que devem ser resolvidas. No entanto, o método GMVSS permite também que se resolvam as tais Diofantinas, podendo fornecer os resultados de $E_j(z^{-1})$, $F_j(z^{-1})$ e $G_j(z^{-1})$. Desse modo, a solução via GMVSS é descrita a seguir.

Seja o sistema ARIMAX descrito por

$$\Delta A(z^{-1})y(k) = B(z^{-1})\Delta u(k-1) + C(z^{-1})w(k), \quad (6.230)$$

$$\begin{aligned} A(z^{-1}) &= 1 + a_1 z^{-1} + \dots + a_{n_a} z^{-n_a} \\ B(z^{-1}) &= b_0 + b_1 z^{-1} + \dots + b_{n_b} z^{-n_b} \\ C(z^{-1}) &= 1 + c_1 z^{-1} + \dots + c_{n_c} z^{-n_c} \end{aligned} \quad (6.231)$$

considerando também que

$$\bar{A}(z^{-1}) = \Delta A(z^{-1}) = 1 + \bar{a}_1 z^{-1} + \dots + \bar{a}_{n_{\bar{a}}} z^{-n_{\bar{a}}} \quad (6.232)$$

e que $n_{\bar{a}} = n_a + 1$.

As matrizes da representação do modelo ARIMAX no espaço de estado são necessárias para a solução, sendo estas descritas da seguinte maneira:

$$\begin{aligned} \mathbf{x}(k) &= \mathbf{A}\mathbf{x}(k-1) + \mathbf{B}\Delta\mathbf{u}(k-1) + \mathbf{\Gamma}\mathbf{w}(k-1) \\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k) + \mathbf{w}(k) \end{aligned} \quad (6.233)$$

em que

$$\mathbf{A} = \begin{bmatrix} -\bar{a}_1 & & \\ \vdots & \mathbf{I}_{n_a} & \\ -\bar{a}_{n_{\bar{a}}} & \mathbf{0}_{1 \times n_a} & \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} b_0 \\ \vdots \\ b_{n_{\bar{a}}} \end{bmatrix} \quad (6.234)$$

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}, \quad \mathbf{\Gamma} = \begin{bmatrix} c_1 - \bar{a}_1 \\ \vdots \\ c_{n_{\bar{a}}} - \bar{a}_{n_{\bar{a}}} \end{bmatrix} \quad (6.235)$$

Com base nessas matrizes é possível calcular $E_j(z^{-1})$, $F_j(z^{-1})$ e $G_j(z^{-1})$ da seguinte forma (SILVEIRA; COELHO, 2011):

$$E_j(z^{-1}) = 1 + \mathbf{CA}^0 \mathbf{\Gamma} z^{-1} + \mathbf{CA}^1 \mathbf{\Gamma} z^{-2} + \cdots + \mathbf{CA}^{(j-2)} \mathbf{\Gamma} z^{-(j-1)}, \quad (6.236)$$

$$F_j(z^{-1}) = \begin{bmatrix} z^0 & z^{-1} & \cdots & z^{-n_a} \end{bmatrix} \mathbf{A}^{(j-1)} \mathbf{\Gamma}, \quad (6.237)$$

$$G_j(z^{-1}) = E_j(z^{-1})B(z^{-1}). \quad (6.238)$$

É importante destacar como a estrutura de $E_j(z^{-1})$, e consequentemente de $G_j(z^{-1})$, vai se comportar conforme $j = 1, \dots, N_y$ vai sendo considerado. Por exemplo:

$$\begin{aligned} E_1(z^{-1}) &= 1 \\ E_2(z^{-1}) &= 1 + \mathbf{C}\mathbf{\Gamma} z^{-1} \\ E_3(z^{-1}) &= 1 + \mathbf{C}\mathbf{\Gamma} z^{-1} + \mathbf{CA}^1 \mathbf{\Gamma} z^{-2} \\ &\vdots \\ E_{N_y}(z^{-1}) &= 1 + \mathbf{C}\mathbf{\Gamma} z^{-1} + \mathbf{CA}^1 \mathbf{\Gamma} z^{-2} + \cdots + \mathbf{CA}^{(N_y-2)} \mathbf{\Gamma} z^{-(N_y-1)} \end{aligned} \quad (6.239)$$

Esta estrutura de termos dos $E_j(z^{-1})$ conforme $j = 1, \dots, N_y$ é referente à parcela que opera diretamente sobre os termos do presente e do futuro da ação de controle, i.e., $\Delta u_f(k)$, \dots , $\Delta u_f(k + N_y - 1)$.

Tal estrutura se expande no caso dos $G_j(z^{-1})$ polinômios, quando $E_j(z^{-1})$ é multiplicado por $B(z^{-1})$. Nesse caso, os n_b termos a mais em relação ao que se tinha na estrutura dos $E_j(z^{-1})$, são justamente a parte que opera sobre os n_b regressores das ações de controle, i.e., $\Delta u_f(k - 1)$, \dots , $\Delta u_f(k - n_b)$, ficando da seguinte forma:

$$G_1(z^{-1}) = g_0 + \underbrace{\bar{g}_{1,0} z^{-1} + \cdots + \bar{g}_{1,n_b} z^{-n_b}}_{n_b \text{ termos}}$$

$$\begin{aligned}
G_2(z^{-1}) &= g_0 + g_1 z^{-1} + \underbrace{\bar{g}_{2,0} z^{-2} + \cdots + \bar{g}_{2,n_b} z^{-(n_b+1)}}_{n_b \text{ termos}} \\
G_3(z^{-1}) &= g_0 + g_1 z^{-1} + g_2 z^{-2} + \underbrace{\bar{g}_{3,0} z^{-3} + \cdots + \bar{g}_{3,n_b} z^{-(n_b+2)}}_{n_b \text{ termos}} \\
&\vdots \\
G_{N_y}(z^{-1}) &= g_0 + g_1 z^{-1} + \cdots + g_{N_y-1} z^{-(N_y-1)} \\
&\quad + \underbrace{\bar{g}_{N_y,0} z^{-N_y} + \cdots + \bar{g}_{N_y,n_b} z^{-(n_b+N_y-1)}}_{n_b \text{ termos}}
\end{aligned} \tag{6.240}$$

A partir dessa demonstração das características dos polinômios $G_j(z^{-1})$ e também dos $F_j(z^{-1})$ pode-se organizar a equação do preditor do GPC, descrita por

$$\hat{y}(k+j|k) = E_j(z^{-1})B_k(z^{-1})\Delta u_f(k+j-1) + F_j(z^{-1})y_f(k), \tag{6.241}$$

em uma forma matricial e vetorial, como segue:

$$\hat{\mathbf{y}} = \mathbf{G} \underline{\Delta_{\mathbf{u}}} + \mathbf{f}_r. \tag{6.242}$$

$$\mathbf{G} = \begin{bmatrix} g_0 & 0 & 0 & \cdots & 0 \\ g_1 & g_0 & 0 & \ddots & \vdots \\ g_2 & g_1 & g_0 & \ddots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ g_{N_y-1} & g_{N_y-2} & g_{N_y-3} & \cdots & g_0 \end{bmatrix} \tag{6.243}$$

$$\hat{\mathbf{y}} = \begin{bmatrix} \hat{y}(k+1|k) \\ \hat{y}(k+2|k) \\ \vdots \\ \hat{y}(k+N_y|k) \end{bmatrix}, \quad \underline{\Delta_{\mathbf{u}}} = \begin{bmatrix} \Delta u_f(k) \\ \Delta u_f(k+1) \\ \vdots \\ \Delta u_f(k+N_y-1) \end{bmatrix} \tag{6.244}$$

O modelo de resposta livre completa o preditor, sendo descrito

da seguinte forma:

$$\mathbf{f}_r = \bar{\mathbf{G}} \underline{\Delta}_{\mathbf{u}} + \mathbf{F} \mathbf{y} = \begin{bmatrix} \bar{\mathbf{G}} & \mathbf{F} \end{bmatrix} \begin{bmatrix} \underline{\Delta}_{\mathbf{u}} \\ \mathbf{y} \end{bmatrix} \quad (6.245)$$

$$\bar{\mathbf{G}} = \begin{bmatrix} \bar{g}_{1,0} & \cdots & \bar{g}_{1,n_b} \\ \bar{g}_{2,0} & \cdots & \bar{g}_{2,n_b} \\ \vdots & \cdots & \vdots \\ \bar{g}_{N_y,0} & \cdots & \bar{g}_{N_y,n_b} \end{bmatrix} \quad (6.246)$$

$$\mathbf{F} = \begin{bmatrix} (\mathbf{A}^0 \mathbf{\Gamma})^T \\ (\mathbf{A}^1 \mathbf{\Gamma})^T \\ \vdots \\ (\mathbf{A}^{N_y-1} \mathbf{\Gamma})^T \end{bmatrix} = \begin{bmatrix} f_{1,0} & \cdots & f_{1,n_a} \\ f_{2,0} & \cdots & f_{2,n_a} \\ \vdots & \cdots & \vdots \\ f_{N_y,0} & \cdots & f_{N_y,n_a} \end{bmatrix} \quad (6.247)$$

$$\underline{\Delta}_{\mathbf{u}} = \begin{bmatrix} \Delta u_f(k-1) \\ \Delta u_f(k-2) \\ \vdots \\ \Delta u_f(k-n_b) \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} y_f(k) \\ y_f(k-1) \\ \vdots \\ y_f(k-n_a) \end{bmatrix} \quad (6.248)$$

Mesmo com todo o detalhamento apresentado e a explícita demonstração de como se obter as matrizes e os vetores que compõem a saída predita na forma matricial e vetorial mostrada em (6.249), para aqueles que implementarão o GPC pela primeira vez em uma prática real mesmo via simulação, ainda pode haver grande dificuldade. Portanto, a seguir apresenta-se um exemplo numérico especificamente para melhor instruir a prática desta etapa de obtenção das matrizes envolvidas no GPC com o auxílio do método GMVSS.

Exemplo 6.10 *Para o modelo de um sistema eletrônico subamortecido, que foi utilizado também no Exemplo 6.5, obtenha as matrizes que permitem o projeto do preditor do GPC como se mostra a seguir:*

$$\hat{\mathbf{y}} = \mathbf{G} \underline{\Delta}_{\mathbf{u}} + \mathbf{f}_r.$$

$$\mathbf{f}_r = \bar{\mathbf{G}} \underline{\Delta_u} + \mathbf{F} \mathbf{y} = \begin{bmatrix} \bar{\mathbf{G}} & \mathbf{F} \end{bmatrix} \begin{bmatrix} \underline{\Delta_u} \\ \mathbf{y} \end{bmatrix}$$

O modelo ARIMAX do processo é descrito pelos polinômios:

$$\begin{aligned} A(z^{-1}) &= 1 - 1,2356881z^{-1} + 0,4526202z^{-2} \\ B(z^{-1}) &= 0,0101615 + 0,1947393z^{-1} \\ C(z^{-1}) &= 1 - 0,0398287z^{-1} + 0,0825432z^{-2} \end{aligned} \quad (6.249)$$

em que $\Delta A(z^{-1}) = 1 + \bar{a}_1 z^{-1} + \bar{a}_2 z^{-2} + \bar{a}_3 z^{-3}$.

Fazendo-se o projeto via método GMVSS, o modelo ARIMAX é convertido à forma canônica observável no espaço de estado, cujas matrizes \mathbf{A} , \mathbf{B} , $\mathbf{\Gamma}$ e \mathbf{C} são mostradas a seguir:

$$\begin{aligned} \mathbf{A} &= \begin{bmatrix} -\bar{a}_1 & 1 & 0 \\ -\bar{a}_2 & 0 & 1 \\ -\bar{a}_3 & 0 & 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} b_0 \\ b_1 \\ 0 \end{bmatrix}, \quad \mathbf{\Gamma} = \begin{bmatrix} c_1 - \bar{a}_1 \\ c_2 - \bar{a}_2 \\ -\bar{a}_3 \end{bmatrix} \\ \mathbf{C} &= \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \end{aligned} \quad (6.250)$$

Uma forma de resolver este exemplo dinamicamente e para qualquer valor de horizonte de predição N_y , inclusive podendo-se adotá-la para o projeto online e adaptativo do GPC, é mostrada na Tabela 25 para uma implementação em Scilab, mas que pode ser facilmente convertida para outras linguagens de programação. No entanto, para enfatizar o método de solução, neste exemplo consideraremos

$$N_y = 3$$

para limitar a dimensão das matrizes e tornar o resultado factível de ser expresso em poucas páginas deste texto. Tal desenvolvimento é mostrado a seguir.

Aplicando-se o método GMVSS obtém-se:

$$\begin{aligned} E_1(z^{-1}) &= 1 \\ E_2(z^{-1}) &= 1 + \mathbf{CA}^0 \mathbf{\Gamma} z^{-1} = 1 + e_1 z^{-1} \\ E_3(z^{-1}) &= 1 + \mathbf{CA}^0 \mathbf{\Gamma} z^{-1} + \mathbf{CA}^1 \mathbf{\Gamma} z^{-2} = 1 + e_1 z^{-1} + e_2 z^{-2} \end{aligned} \quad (6.251)$$

$$\begin{aligned} G_1(z^{-1}) &= E_1(z^{-1})B(z^{-1}) = b_0 + b_1 z^{-1} \\ G_1(z^{-1}) &= g_0 + \bar{g}_{1,0} z^{-1} \end{aligned} \quad (6.252)$$

$$\begin{aligned} G_2(z^{-1}) &= E_2(z^{-1})B(z^{-1}) = b_0 + (e_1b_0 + b_1)z^{-1} + e_1b_1z^{-2} \\ G_2(z^{-1}) &= g_0 + g_1z^{-1} + \bar{g}_{2,0}z^{-2} \end{aligned} \quad (6.253)$$

$$\begin{aligned} G_3(z^{-1}) &= E_3(z^{-1})B(z^{-1}) \\ G_3(z^{-1}) &= b_0 + (e_1b_0 + b_1)z^{-1} + (e_2b_0 + e_1b_1)z^{-2} + e_2b_1z^{-3} \\ G_3(z^{-1}) &= g_0 + g_1z^{-1} + g_2z^{-2} + \bar{g}_{3,0}z^{-3} \end{aligned} \quad (6.254)$$

Observe que os últimos $n_b = 1$ termos de $G_j(z^{-1})$, $j = 1, 2, 3$, respectivamente os termos cujos elementos foram barrados como $\bar{g}_{1,0}$, $\bar{g}_{2,0}$, $\bar{g}_{3,0}$, são aqueles que comporão a matriz $\bar{\mathbf{G}}$ que opera no Modelo de Resposta Livre, \mathbf{f}_r . As matrizes \mathbf{G} e $\bar{\mathbf{G}}$ podem então ser construídas com base nos termos dos $G_j(z^{-1})$, $j = 1, 2, 3$, como segue:

$$\mathbf{G} = \begin{bmatrix} g_0 & 0 & 0 \\ g_1 & g_0 & 0 \\ g_2 & g_1 & g_0 \end{bmatrix}, \quad \bar{\mathbf{G}} = \begin{bmatrix} \bar{g}_{1,0} \\ \bar{g}_{2,0} \\ \bar{g}_{3,0} \end{bmatrix} \quad (6.255)$$

$$\mathbf{G} = \begin{bmatrix} 0,0101 & 0 & 0 \\ 0,2170 & 0,0101 & 0 \\ 0,4611 & 0,2170 & 0,0101 \end{bmatrix}, \quad \bar{\mathbf{G}} = \begin{bmatrix} 0,1947 \\ 0,4276 \\ 0,6433 \end{bmatrix} \quad (6.256)$$

Aplicando-se o método do ganho do preditor do GMVSS é possível obter os polinômios $F_j(z^{-1})$, ou mesmo, diretamente a matriz \mathbf{F} , como segue:

$$\mathbf{F} = \begin{bmatrix} (\mathbf{A}^0\mathbf{\Gamma})^T \\ (\mathbf{A}^1\mathbf{\Gamma})^T \\ (\mathbf{A}^2\mathbf{\Gamma})^T \end{bmatrix} = \begin{bmatrix} f_{1,0} & f_{1,1} & f_{1,2} \\ f_{2,0} & f_{2,1} & f_{2,2} \\ f_{3,0} & f_{3,1} & f_{3,2} \end{bmatrix} \quad (6.257)$$

$$\mathbf{F} = \begin{bmatrix} 2,1958 & -1,6057 & 0,4526 \\ 3,3034 & -3,2546 & -0,9938 \\ 4,1309 & -4,5834 & 1,4952 \end{bmatrix} \quad (6.258)$$

$$\hat{\mathbf{y}} = \mathbf{G}\underline{\Delta_u} + \underbrace{\begin{bmatrix} \bar{\mathbf{G}} & \mathbf{F} \end{bmatrix}}_{\mathbf{f}_r} \begin{bmatrix} \underline{\Delta_u} \\ \mathbf{y} \end{bmatrix} \quad (6.259)$$

$$\begin{bmatrix} \hat{y}(k+1|k) \\ \hat{y}(k+2|k) \\ \hat{y}(k+3|k) \end{bmatrix} = \mathbf{G} \begin{bmatrix} \Delta u_f(k) \\ \Delta u_f(k+1) \\ \Delta u_f(k+2) \end{bmatrix} + \begin{bmatrix} \bar{\mathbf{G}} & \mathbf{F} \end{bmatrix} \begin{bmatrix} \Delta u_f(k-1) \\ y_f(k) \\ y_f(k-1) \\ y_f(k-2) \end{bmatrix} \quad (6.260)$$

Com base nessa estrutura matricial e vetorial da trajetória de predição de $j = 1, 2, 3$ do GPC, é importante reforçar alguns pontos já frisados, mas que em um exemplo mais compacto pode ficar mais claro o entendimento. Por exemplo, a matriz \mathbf{G} deste caso considera $N_u = N_y$ e por isto tem dimensão $N_y \times N_y$, tal que $\underline{\Delta}_{\mathbf{u}}$ vai até $N_y - 1$. No entanto, caso se queira reduzir o horizonte de predição da variação da ação de controle, $N_u < N_y$, basta truncar o número de colunas de \mathbf{G} conforme se mostrou no DMC, tal que

$$\mathbf{G}_{N_y \times N_u}, \underline{\Delta}_{\mathbf{u}} \xrightarrow{N_u \times 1}. \quad (6.261)$$

No Modelo de Resposta Livre, \mathbf{f}_r , reforça-se que os regressores de $\Delta u_f(k)$ estarão limitados a n_b , enquanto que os regressores de $y_f(k)$ estarão limitados a n_a , tal que

$$\begin{bmatrix} \bar{\mathbf{G}} & \mathbf{F} \end{bmatrix}_{N_y \times (n_b + n_a)}, \begin{bmatrix} \underline{\Delta}_{\mathbf{u}} \\ \mathbf{y} \end{bmatrix}_{(n_b + n_a) \times 1} \quad (6.262)$$

Como contraponto em relação a algumas técnicas que estudamos anteriormente, pode-se citar os casos PMVC e DMC, em que no primeiro os regressores de entrada eram limitados a $N_y - 1$ e no segundo eram limitados a $N \rightarrow \infty$. Já em relação aos regressores da saída, PMVC e GPC são equivalentes, enquanto o DMC não utiliza regressores da saída $y_f(k)$. Na verdade, estas versões filtradas, $y_f(k)$ e $\Delta u_f(k)$, pelo polinômio $C(z^{-1})$, somente o PMVC e o GPC utilizam.

Tabela 25 – Código em Scilab para reproduzir o Exemplo GPC.

```

// Projeto GPC com base no GMVSS.
clear; xdel(winsid()); clc;

// Modelo ARIMAX do processo eletrônico subamortecido
Ts = 0.05; // Período de amostragem
Az = [1 -1.2356881 0.4526202];
Bz = [0.0101615 0.1947393];
Cz = [1 -0.0398287 0.0825432];
Var_w = 0.0068327 ; // Variância do ruído do processo, w(k)
DAz = conv([1 -1],Az);

// Conversao ao espaço de estado para aplicação do GMVSS
A = [ -DAz(2:$)' eye(2,2); 0 0]
B = [Bz';0];
C = [1 0 0];
G = [Cz(2)-DAz(2);Cz(3)-DAz(3);0-DAz(4)]; // Matriz Gamma

// Construção das matrizes do preditor
nb = 1; Ny = 3;
E=1; F=[A^0*G]'; Gmat=[];
Gj=conv(E,Bz); Gbar=[Gj($-nb+1:$)];
// Construindo as matrizes Gbar e F:
for j=2:Ny
    E=[E C*A^(j-2)*G]; F=[F; [A^(j-1)*G]'];
    Gj=conv(E,Bz); Gbar=[Gbar; Gj($-nb+1:$)];
end
// Construindo a matriz Gmat com base em Gj completo
Gmat = Gj(1:$-nb)';
for j=2:Ny
    Gmat = [Gmat [zeros(1,j-1) Gj(1:$-nb-j+1)]'];
end
// Alternativa mais simples usando as funções
// para a Matriz de Toeplitz (toeplitz) e
// Matriz Lower Triangular (tril)
Gmat = tril(toeplitz(Gj(1:$-nb)));
disp(['Gbar = ']); disp(Gbar);
disp(['Gmat = ']); disp(Gmat);
disp(['F = ']); disp(F);

```

O preditor do GPC foi detalhadamente discutido e seu projeto, com base no método GMVSS, foi codificado na linguagem Scilab como

mostrado na Tabela 25. Desse modo, agora devemos retornar ao problema de otimização do GPC, mostrado na equação (6.197), onde utilizaremos o sistema preditor desenvolvido para determinar a lei de controle do GPC.

6.10.3 Lei de controle do GPC

O problema de otimização resolvido pelo método GPC foi apresentado nas equações (6.197) a (6.199), reescritas a seguir com base na saída e na variação da entrada filtradas pelo polinômio $C_k(z^{-1})$:

$$\min_{\underline{\Delta u}} J_{gpc} = \mathbf{E} \left\{ \sum_{j=1}^{N_y} [y_f(k+j) - r(k+j)]^2 + \lambda \sum_{j=1}^{N_u} [\Delta u_f(k+j-1)]^2 \right\}, \quad (6.263)$$

$$\underline{\Delta u} = [\Delta u_f(k) \quad \Delta u_f(k+1) \quad \cdots \quad \Delta u_f(k+N_u-1)]^T, \quad (6.264)$$

sujeito a

$$A_k(z^{-1})y(k) = B_k(z^{-1})u(k-1) + C_k(z^{-1})\frac{w(k)}{\Delta}, \quad (6.265)$$

$$\Delta A_k(z^{-1})y_f(k) = B_k(z^{-1})\Delta u_f(k-1) + w(k). \quad (6.266)$$

Como o problema de otimização está sujeito a um sistema dinâmico com perturbações aleatórias, utilizamos o Preditor de Variância Mínima proposto por Åström (1970) para prever a saída estimando a influência futura da perturbação estocástica $w(k)$. Desse modo, vamos reescrever o problema de otimização com base em \hat{y} .

Um outro ponto importante de ser lembrado é que os sinais de entrada e de saída utilizados na estratégia de predição são versões filtradas, como se mostrou em (6.210). Desse modo, o problema de otimização está em função de

$$\Delta u_f(k) = \frac{\Delta u(k)}{C_k(z^{-1})}, \quad (6.267)$$

$$y_f(k) = \frac{y(k)}{C_k(z^{-1})}. \quad (6.268)$$

No entanto, se $C_k(z^{-1}) = 1$, então $\Delta u(k) = \Delta u_f(k)$ e $\Delta y(k) = y_f(k)$.

Assumindo-se em um primeiro momento que $N_y = N_u$ e que toda a trajetória de predição pode ser descrita pelo modelo

$$\hat{\mathbf{y}} = \mathbf{G}\underline{\Delta}_{\mathbf{u}} + \mathbf{f}_{\mathbf{r}}, \quad (6.269)$$

em que o Modelo de Resposta Livre é reescrito como

$$\mathbf{f}_{\mathbf{r}} = \begin{bmatrix} \bar{\mathbf{G}} & \mathbf{F} \end{bmatrix} \begin{bmatrix} \underline{\Delta}_{\mathbf{u}} \\ \mathbf{y} \end{bmatrix} = \mathbf{F}_{\mathbf{R}}\mathbf{x}, \quad (6.270)$$

substitui-se o vetor de saídas preditas,

$$\hat{\mathbf{y}} = \mathbf{G}\underline{\Delta}_{\mathbf{u}} + \mathbf{F}_{\mathbf{R}}\mathbf{x}, \quad (6.271)$$

na função custo do GPC, reescrevendo o problema de otimização com base em

$$\min_{\underline{\Delta}_{\mathbf{u}}} J_{gpc} = [\hat{\mathbf{y}} - \mathbf{r}]^T [\hat{\mathbf{y}} - \mathbf{r}] + \lambda \underline{\Delta}_{\mathbf{u}}^T \underline{\Delta}_{\mathbf{u}}, \quad (6.272)$$

em que a trajetória de referências futuras é dada por

$$\mathbf{r}^T = \begin{bmatrix} r(k+1) & r(k+2) & \cdots & r(k+N_y) \end{bmatrix}. \quad (6.273)$$

A partir deste ponto, praticamente repete-se o que já fora desenvolvido para o método DMC, como segue:

$$J_{gpc} = \left[\mathbf{G}\underline{\Delta}_{\mathbf{u}} + \mathbf{F}_{\mathbf{R}}\mathbf{x} - \mathbf{r} \right]^T \left[\mathbf{G}\underline{\Delta}_{\mathbf{u}} + \mathbf{F}_{\mathbf{R}}\mathbf{x} - \mathbf{r} \right] + \lambda \underline{\Delta}_{\mathbf{u}}^T \underline{\Delta}_{\mathbf{u}} \quad (6.274)$$

$$\mathbf{E} = \mathbf{r} - \mathbf{F}_{\mathbf{R}}\mathbf{x} \quad (6.275)$$

$$J_{gpc} = \left[\mathbf{G}\underline{\Delta}_{\mathbf{u}} - \mathbf{E} \right]^T \left[\mathbf{G}\underline{\Delta}_{\mathbf{u}} - \mathbf{E} \right] + \lambda \underline{\Delta}_{\mathbf{u}}^T \underline{\Delta}_{\mathbf{u}} \quad (6.276)$$

$$J_{gpc} = \underline{\Delta}_{\mathbf{u}}^T \left[\mathbf{G}^T \mathbf{G} + \lambda \mathbf{I} \right] \underline{\Delta}_{\mathbf{u}} - 2 \underline{\Delta}_{\mathbf{u}}^T \mathbf{G}^T \mathbf{E} + \mathbf{E}^T \mathbf{E} \quad (6.277)$$

$$\underbrace{\min_{\underline{\Delta}_{\mathbf{u}}}}_{\underline{\Delta}_{\mathbf{u}}} J_{gpc} = \frac{\partial J_{gpc}}{\partial \underline{\Delta}_{\mathbf{u}}} = 0 \quad (6.278)$$

$$2 [\mathbf{G}^T \mathbf{G} + \lambda \mathbf{I}] \underline{\Delta}_{\mathbf{u}} = 2 \mathbf{G}^T \mathbf{E} \quad (6.279)$$

$$\underline{\Delta}_{\mathbf{u}} = [\mathbf{G}^T \mathbf{G} + \lambda \mathbf{I}]^{-1} \mathbf{G}^T [\mathbf{r} - \mathbf{F}_{\mathbf{R}} \mathbf{x}] \quad (6.280)$$

O ganho do GPC é calculado da mesma forma que no DMC, sendo

$$\mathbf{K} = [\mathbf{G}^T \mathbf{G} + \lambda \mathbf{I}]^{-1} \mathbf{G}^T \quad (6.281)$$

uma matriz de ganhos na forma

$$\mathbf{K}^T = [\mathbf{k}_1 \quad \mathbf{k}_2 \quad \cdots \quad \mathbf{k}_{N_y}] \quad (6.282)$$

Dependendo do horizonte de controle N_u , a matriz \mathbf{G} é truncada à dimensão $N_y \times N_u$, enquanto que a matriz identidade no termo de ponderação do controle, $\lambda \mathbf{I}$, tem sempre a dimensão $N_y \times N_y$.

Como as ações de controle futuras não são implementadas no momento atual, estas são descartadas na estratégia de horizonte retrocedente, sendo apenas a ação de controle atual aplicada na entrada do processo controlado, como segue:

$$\Delta u_f(k) = \mathbf{k}_1 [\mathbf{r} - \mathbf{F}_{\mathbf{R}} \mathbf{x}] \quad (6.283)$$

$$\mathbf{F}_{\mathbf{R}} \mathbf{x} = [\bar{\mathbf{G}} \quad \mathbf{F}] \begin{bmatrix} \underline{\Delta}_{\mathbf{u}} \\ \mathbf{y} \end{bmatrix} = [\bar{\mathbf{G}} \quad \mathbf{F}] \begin{bmatrix} \Delta u_f(k-1) \\ \vdots \\ \Delta u_f(k-n_b) \\ y_f(k) \\ \vdots \\ y_f(k-n_a) \end{bmatrix} \quad (6.284)$$

Passando o sinal variação de controle pelo filtro $C_k(z^{-1})$, tem-se

$$\Delta u(k) = \Delta u_f(k) C_k(z^{-1}). \quad (6.285)$$

Por fim, aplica-se o sinal de controle na entrada do processo controlado:

$$u(k) = u(k-1) + \Delta u(k). \quad (6.286)$$

Exemplo 6.11 *Continuemos de onde paramos no Exemplo 6.10. Até então se tinha para $N_y = 3$ as matrizes \mathbf{G} , $\bar{\mathbf{G}}$ e \mathbf{F} . Sendo assim, assumindo $N_u = N_y$ e $\lambda = 0,1$, as matrizes de ganhos \mathbf{K} e \mathbf{F}_R são:*

$$\mathbf{K} = (\mathbf{G}^T \mathbf{G} + \lambda \mathbf{I})^{-1} \mathbf{G}^T = \begin{bmatrix} \mathbf{k}_1 \\ \mathbf{k}_2 \\ \mathbf{k}_3 \end{bmatrix} \quad (6.287)$$

$$\mathbf{K} = \begin{bmatrix} 0,1015104 & -0,0001046 & -0,0001046 \\ -0,0001046 & 0,1015104 & -0,0001046 \\ -0,0001046 & -0,0001046 & 0,1015104 \end{bmatrix} \quad (6.288)$$

$$\mathbf{F}_R = [\bar{\mathbf{G}} \quad \mathbf{F}]$$

$$\mathbf{F}_R = \begin{bmatrix} 0,1947393 & 2,1958594 & -1,6057651 & 0,4526202 \\ 0,4276201 & 3,3034916 & -3,2546675 & 0,9938903 \\ 0,6433196 & 4,1309095 & -4,583422 & 1,495227 \end{bmatrix} \quad (6.289)$$

De posse de todas as matrizes de ganho necessárias à implementação do GPC, o sistema está pronto para ser embarcado no software de controle.

A cada nova medida $y(k)$ que chega do sensor, uma versão filtrada $y_f(k)$ é produzida, como:

$$y_f(k) = \frac{y(k)}{C_k(z^{-1})} \quad (6.290)$$

$$y_f(k) = -c_1 y_f(k-1) - c_2 y_f(k-2) + y(k) \quad (6.291)$$

Como o modelo do processo tem dimensão definida por $n_a = 2$ e $n_b = 1$, o vetor \mathbf{x} do Modelo de Resposta Livre assume os seguintes regressores:

$$\mathbf{x}^T = [\Delta u_f(k-1) \quad y_f(k) \quad y_f(k-1) \quad y_f(k-2)] \quad (6.292)$$

A lei de variação de controle filtrada é então implementada como

se mostra a seguir:

$$\Delta u_f(k) = \mathbf{k}_1 \left[\begin{pmatrix} r(k+1) \\ r(k+2) \\ r(k+3) \end{pmatrix} - \mathbf{F}_R \mathbf{x} \right] \quad (6.293)$$

Como a entrada do processo controlado é $u(k)$, então não é possível aplicar diretamente a variação do sinal de controle filtrado $\Delta u_f(k)$. Portanto, converte-se o sinal $\Delta u_f(k)$ em $\Delta u(k)$, como segue:

$$\frac{\Delta u(k)}{C_k(z^{-1})} = \Delta u_f(k) \quad (6.294)$$

$$\Delta u(k) = \Delta u_f(k) + c_1 \Delta u_f(k-1) + c_2 \Delta u_f(k-2) \quad (6.295)$$

e, finalmente, aplica-se o sinal de controle à entrada do processo:

$$u(k) = u(k-1) + \Delta u(k) \quad (6.296)$$

O resultado deste exemplo, com base nos gráficos de seguimento de referência e na variação do sinal controle, é mostrado na Figura 75.

Para reproduzir este exemplo em uma simulação, o código em Scilab é apresentado na Tabela 26.

Esta forma de projetar o GPC via método GMVSS e incluindo o caso de ruído colorido para modelos ARIMAX SISO de qualquer ordem, até onde o autor pode pesquisar, é uma forma de projeto inédita. No entanto, no GPC não há, de fato, uma grande contribuição de $C_k(z^{-1})$ no resultado em termos da minimização da potência/variação da variação do sinal de controle, $\Delta u(k)$, como se observa no Controle Preditivo de Horizonte Irrestrito (UHPC, *Unrestricted Horizon Predictive Control*) (TRENTINI et al., 2016; SILVEIRA et al., 2020), que estudaremos no próximo capítulo.

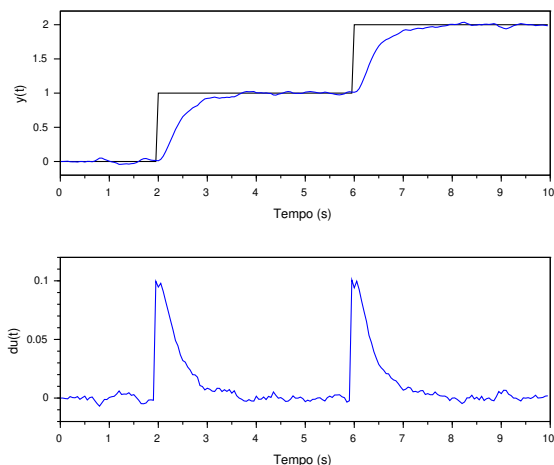


Figura 75 – Simulação do GPC, com $N_y = N_u = 3$ e $\lambda = 0, 1$, aplicado ao modelo do processo eletrônico subamortecido.

Tabela 26 – Código em Scilab para reproduzir a simulação do GPC do Exemplo 6.11.

```
// Projeto GPC com base no GMVSS.
clear; xdel(winsid()); clc;

// Modelo ARIMAX do processo eletrônico subamortecido
Ts = 0.05; // Período de amostragem
Az = [1 -1.2356881 0.4526202];
Bz = [0.0101615 0.1947393];
Cz = [1 -0.0398287 0.0825432];
d = 1; // Atraso discreto
Var_w = 0.0068327 ; // Variância do ruído do processo, w(k)
DAz = conv([1 -1],Az);

// Conversao ao espaço de estado para aplicação do GMVSS
A = [ -DAz(2:$)' [eye(2,2);0 0]]
B = [Bz';0];
C = [1 0 0];
//G = [Cz(2)-DAz(2);Cz(3)-DAz(3);0-DAz(4)]; // Matriz Gamma (ARIMAX)
G = [0-DAz(2);0-DAz(3);0-DAz(4)]; //(ARIX)
```

```

// Construcao das matrizes do preditor
nb = 1; Ny = 3; Nu=Ny;
E=1; F=[A^0*G]'; Gmat=[];
Gj=conv(E,Bz); Gbar=[Gj($-nb+1:$)];
// Construindo as matrizes Gbar e F:
for j=2:Ny
    E=[E C*A^(j-2)*G]; F=[F; [A^(j-1)*G]'];
    Gj=conv(E,Bz); Gbar=[Gbar; Gj($-nb+1:$)];
end
// Construindo a matriz Gu com base em Gj completo
Gmat = Gj(1:$-nb)';
for j=2:Ny
    Gmat = [Gmat [zeros(1,j-1) Gj(1:$-nb-j+1)]];
end
// Alternativa mais simples usando as funções
// para a Matriz de Toeplitz (toeplitz) e
// Matriz Lower Triangular (tril)
Gmat = tril(toeplitz(Gj(1:$-nb)));
disp(['Gbar = ']); disp(Gbar);
disp(['Gmat = ']); disp(Gmat);
disp(['F = ']); disp(F);

// Ganhos do controlador GPC
lambda = 0.1; Gmat = Gmat(d:Ny,d:Nu);
K = inv(Gmat'*Gmat+lambda*eye(Nu-d+1,Nu-d+1))*Gmat';
k1 = K(1,:); FR = [Gbar(d:Ny,:) F(d:Ny,:)];

// Simulação com o controlador GPC
r = [zeros(2/Ts,1);
     ones(4/Ts,1) ;
     2*ones(4/Ts+Ny,1)]; // referência com predição Ny
N = length(r)-Ny;
t = 0:Ts:N*Ts-Ts;
w = 1*grand(1,N, 'nor', 0, Var_w ); // ruído de processo
//w = fscanfMat('w.var');
//fprintfMat('w.var', w)

for k=1:2
    y(k)=0; u(k)=0; du(k)=0; duf(k)=0; yf(k)=0;
end
for k=3:N
    y(k)=-Az(2)*y(k-1)-Az(3)*y(k-2)+Bz(1)*u(k-1)+Bz(2)*u(k-2) ...
        +w(k) +Cz(2)*w(k-1) +Cz(3)*w(k-2);

```

```

// GPC com filtro C(z)
//yf(k) = -Cz(2)*yf(k-1) -Cz(3)*yf(k-2) +y(k); // filtro C(z) em y(k)
yf(k)=y(k); // ARIX
x = [duf(k-1) yf(k) yf(k-1) yf(k-2)]'; // Vetor x da Res-
posta Livre
duf(k)=k1*( r(k+d:k+Ny,1) -FR*x );
//du(k) = duf(k) +Cz(2)*duf(k-1) +Cz(3)*duf(k-2); // duf(k) para du(k)
du(k) = duf(k); // ARIX
u(k)=u(k-1)+du(k); // ação de controle
end
t = 0:Ts:length(y)*Ts-Ts;
subplot(211)
plot(t,r(1:$-Ny),'k'); plot(t,y,'b');
ylabel('y(t)'); xlabel('Tempo (s)')
subplot(212)
plot(t,du,'b'); ylabel('du(t)'); xlabel('Tempo (s)');

```


REFERÊNCIAS

- AGUIRRE, L. A. *Introdução à Identificação de Sistemas – Lineares e Não-Lineares Aplicadas a Sistemas Reais*. 3a. ed. MG, Brasil: Editora da UFMG, 2007.
- ÅSTRÖM, K. J. *Introduction to Stochastic Control Theory*. [S.l.]: Dover Publications, 1970. (Dover Books on Electrical Engineering Series). ISBN 9780486445311.
- ASTROM, K. J.; WITTENMARK, B. *Adaptive Control*. 2nd. ed. Mineola, NY, USA: Dover Publications, 2008.
- ASTROM, K. J.; WITTENMARK, B. *Computer-controlled systems: theory and design*. 3rd. ed. Mineola, NY, USA: Dover Publications, 2011.
- BITMEAD, R. R.; GEVERS, M.; WERTZ, V. Book. *Adaptive optimal control: the thinking man's GPC*. [S.l.]: Prentice Hall, 1990. xi, 244 p. : p.
- CAMACHO, E. F.; BORDONS, C. A. *Model Predictive Control*. 2nd. ed. London: Springer-Verlag, 2007. ISBN 1852336943.
- CASTRO, L. A. M. de; SILVEIRA, A. d. S.; ARAÚJO, R. d. B. Unrestricted horizon predictive control applied to a nonlinear siso system. *International Journal of Dynamics and Control*, 2022. ISSN 2195-2698.
- CLARKE, D.; MOHTADI, C.; TUFFS, P. Generalized predictive control: Part i. the basic algorithm. *Automatica*, v. 23, n. 2, p. 137 – 148, 1987. ISSN 0005-1098.
- CLARKE, D. W.; GAWTHROP, P. J. Self-tuning controller. *Proceedings of the Institution of Electrical Engineers*, v. 122, n. 9, p. 929–934, 1975.
- COELHO, A. A. R.; COELHO, L. d. S. *Identificação de Sistemas Dinâmicos Lineares*. Florianópolis, SC, Brasil: Editora da UFSC, 2004.
- CUTLER, C. R.; RAMAKER, B. L. Dynamic matrix control – a computer control algorithm. In: *Joint Automatic Control Conference, San Francisco, USA*. [S.l.: s.n.], 1980. v. 17.

DORF, R. C.; BISHOP, R. H. *Modern control systems*. 8th. ed. Upper Saddle River, NJ, USA: Addison Wesley Longman, Inc, 1998. ISBN 0-201-30864-9.

FRANKLIN, G. F.; POWELL, J. D.; WORKMAN, M. L. *Digital Control of Dynamic Systems*. 3rd. ed. [S.l.]: Addison-Wesley, 1997.

HESPANHA, J. P. *Linear Systems Theory*. Princeton, New Jersey: Princeton Press, 2018. ISBN13: 9780691179575.

LEMES, A. G.; PRADO, F. D.; KIAMETIS, D.; SILVEIRA, A. S.; COELHO, A. A. R.; MEZA, C. Software livre na educação em engenharia de controle: Um estudo de caso na análise e projeto de controle de um pêndulo amortecido. In: *Congresso Brasileiro de Autômática 2010*. [S.l.: s.n.], 2010. v. 1.

LEWIS, F. L.; XIE, L.; POPA, D. *Optimal and Robust Estimation: With an introduction to stochastic control theory*. 2nd. ed. [S.l.]: CRC Press. Taylor & Francis Group, 2008.

MAYA, P.; LEONARDI, F. *Controle Essencial*. 2a. ed. [S.l.]: Pearson Education do Brazil Ltda., 2014.

MORARI, M.; ZAFIRIOU, E. *Robust Process Control*. 1st. ed. New Jersey, USA: Prentice Hall, 1989.

OGATA, K. *Discrete-time Control Systems*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1987. ISBN 0-132-16102-8.

SEBORG, D. E.; EDGAR, T. F.; MELLICHAMP, D. A. *Process Dynamics and Control*. 2. ed. [S.l.]: John Wiley & Sons, 2003. ISBN 9780471000778.

SILVEIRA, A.; SILVA, A.; COELHO, A.; REAL, J.; SILVA, O. Design and real-time implementation of a wireless autopilot using multivariable predictive generalized minimum variance control in the state-space. *Aerospace Science and Technology*, v. 105, p. 106053, 2020. ISSN 1270-9638.

SILVEIRA, A.; TRENTINI, R.; COELHO, A.; KUTZNER, R.; HOFMANN, L. Generalized minimum variance control under long-range prediction horizon setups. *ISA Transactions*, v. 62, n. 1, p. 325 – 332, 2016.

SILVEIRA, A. S.; COELHO, A. A. R. Generalised minimum variance control state-space design. *Control Theory and Applications, IET*, v. 5, n. 15, p. 1709–1715, 13 2011.

SILVEIRA, A. S.; RODRÍGUEZ, J. E. N.; COELHO, A. A. R. Robust design of a 2-DOF GMV controller: A direct self-tuning and fuzzy scheduling approach. *ISA Transactions*, v. 51, n. 1, p. 13–21, 2012. ISSN 0019-0578.

SILVEIRA, A. S.; SODRÉ, L.; SILVA, A. F.; CONDE, L.; BORGES, J. P.; YURI, S.; KLAUTAU, A. Smith predictor-based adaptive control of network-controlled uavs. In: *XV Simpósio Brasileiro de Automação Inteligente (SBAI 2021)*. [S.l.: s.n.], 2021. v. 1.

STEVENS, B. L.; LEWIS, F. L.; JOHNSON, E. N. *Aircraft Control and Simulation: dynamics, controls design, and autonomous systems*. 3rd. ed. [S.l.]: John Wiley & Sons, Inc., 2016.

TRENTINI, R.; SILVEIRA, A.; KUTZNER, R.; HOFMANN, L. On the unrestricted horizon predictive control ? a fully stochastic model-based predictive approach. In: *2016 European Control Conference (ECC)*. [S.l.: s.n.], 2016. p. 1322–1327.

VICARIO, F. *OKID as a general approach to linear and bilinear system identification*. Tese (Doutorado) — Graduate School of Arts and Sciences, Columbia University, 2014.

VICARIO, F.; PHAN, M. Q.; LONGMAN, R. W.; BETTI, R. Generalized framework of okid for linear state-space model identification. In: BOCK, H. G.; PHU, H. X.; RANNACHER, R.; SCHLÖDER, J. P. (Ed.). *Modeling, Simulation and Optimization of Complex Processes HPSC 2015*. Cham: Springer International Publishing, 2017. p. 249–260. ISBN 978-3-319-67168-0.